

ФИЛИППО МЕНЦЕР
САНТО ФОРТУНАТО
КЛЕЙТОН А. ДЭВИС

НАУКА О СЕТЯХ: ВВОДНЫЙ КУРС



Филиппо Менцер, Санто Фортунато и Клейтон А. Дэвис



Наука о сетях: вводный курс



A First Course in Network Science

FILIPPO MENC SER
SANTO FORTUNATO
CLAYTON A. DAVIS



CAMBRIDGE
UNIVERSITY PRESS

Наука о сетях: вводный курс

ФИЛИППО МЕНЦЕР
САНТО ФОРТУНАТО
КЛЕЙТОН А. ДЭВИС



Москва, 2022

УДК 004.7
ББК 16.263
М50

Менцер Ф., Фортунато С., Дэвис К. А.

М50 Наука о сетях: вводный курс / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2021. – 338 с.: ил.

ISBN 978-5-97060-984-2

В этой книге представлены общие принципы построения и функционирования сетей, связанных с разными областями человеческой деятельности. Рассматриваются концепция малых миров и принцип кластеризации применительно к социальным сетям. Обсуждаются роль хабов, тема устойчивости сетей, направленные и взвешенные сети. Всемирная паутина, «Википедия», цитирование, трафик и Twitter используются для иллюстрации роли направления и веса. В заключение исследуются модели возникновения сетей, методы обнаружения сообществ и динамические сетевые процессы.

Каждая глава включает в себя практические занятия по программированию и упражнения, позволяющие читателям проверить свои знания в области построения и анализа сетей. Учебный материал основан на реальных примерах.

Издание рассчитано на широкий круг читателей, знакомых с основами программирования и желающих изучить основы и приложения науки о сетях.

УДК 004.7
ББК 16.263



This translation of A First Course in Network Science is published by arrangement with Cambridge University Press. Russian-language edition copyright © 2021 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-10847-113-8 (англ.)

ISBN 978-5-97060-984-2 (рус.)

© Filippo Menczer, Santo Fortunato,
and Clayton A. Davis, 2020

© Перевод, оформление, издание,
ДМК Пресс, 2021

Коллин, Массимилиано, Айрис: спасибо.

– Филиппо Менцер



Мойм родителям и брату.

– Санто Фортунато

Лиз, Джина, Мэри Джо и Джей: ваша любовь и поддержка значат для меня все.

– Клейтон Дэвис



Содержание

https://t.me/it_boooks



Предисловие	10
Признательности	16
Введение	17
0.1 Социальные сети.....	18
0.2 Коммуникационные сети	21
0.3 Всемирная паутина и «Википедия»	24
0.4 Интернет.....	26
0.5 Транспортные сети.....	27
0.6 Биологические сети.....	29
0.7 Резюме.....	30
0.8 Дальнейшее чтение.....	31
Упражнения.....	32
1 Сетевые элементы	34
1.1 Базовые определения	34
1.2 Манипулирование сетями в исходном коде	36
1.3 Плотность и разреженность	39
1.4 Подсети	42
1.5 Степень.....	43
1.6 Направленные сети.....	44
1.7 Взвешенные сети	45
1.8 Многослойные и темпоральные сети	46
1.9 Представления сетей	49
1.10 Рисование сетей.....	51
1.11 Резюме.....	52
1.12 Дальнейшее чтение.....	53
Упражнения.....	53
2 Малые миры	58
2.1 Рыбак рыбака видит издалека.....	58
2.2 Пути и расстояния	62
2.3 Соединенность и компоненты	67
2.4 Деревья.....	69

2.5	Отыскание кратчайших путей	71
2.6	Социальное расстояние.....	75
2.7	Шесть степеней сепарации.....	78
2.8	Друг моего друга	81
2.9	Резюме.....	84
2.10	Дальнейшее чтение.....	85
	Упражнения.....	86
3	Хабы.....	94
3.1	Меры центральности	95
3.1.1	Степень	95
3.1.2	Близость.....	95
3.1.3	Промежуточность.....	96
3.2	Распределения значений центральности.....	99
3.3	Парадокс дружбы.....	104
3.4	Ультрамалые миры.....	107
3.5	Устойчивость.....	108
3.6	Разложение ядра	110
3.7	Резюме.....	112
3.8	Дальнейшее чтение.....	113
	Упражнения.....	113
4	Направления и веса	119
4.1	Направленные сети	119
4.2	Всемирная паутина	120
4.2.1	Краткая история Всемирной паутины	121
4.2.2	Как работает Всемирная паутина.....	122
4.2.3	Обходчики Всемирной паутины.....	124
4.2.4	Структура Всемирной паутины	127
4.2.5	Тематическая локальность.....	129
4.3	Метрика PageRank	132
4.4	Взвешенные сети	137
4.5	Информация и дезинформация.....	138
4.6	Сети совместной встречаемости.....	143
4.7	Весовая гетерогенность.....	147
4.7.1	Трафик Всемирной паутины	147
4.7.2	Фильтрация связей	149
4.8	Резюме.....	151
4.9	Дальнейшее чтение.....	153
	Упражнения.....	155
5	Сетевые модели.....	162
5.1	Случайные сети	162
5.1.1	Плотность	165
5.1.2	Степенное распределение.....	166

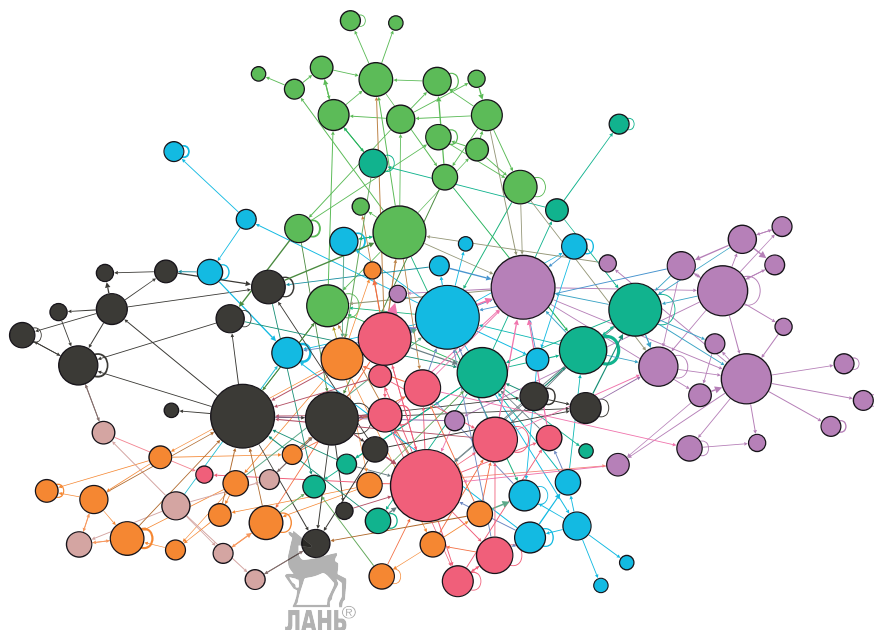
5.1.3	Короткие пути	168
5.1.4	Коэффициент кластеризации.....	169
5.2	Малые миры.....	170
5.3	Конфигурационная модель	174
5.4	Преференциальное прикрепление	177
5.5	Другие преференциальные модели	182
5.5.1	Модель на основе привлекательности	184
5.5.2	Модель на основе приспособленности	185
5.5.3	Модель на основе случайного блуждания	187
5.5.4	Модель на основе копирования.....	190
5.5.5	Модель на основе ранга	191
5.6	Резюме.....	193
5.7	Дальнейшее чтение.....	194
	Упражнения.....	194
6	Сообщества	200
6.1	Базовые определения	203
6.1.1	Переменные сообщества	203
6.1.2	Определения сообщества	205
6.1.3	Разделы.....	207
6.2	Смежные проблемы	209
6.2.1	Деление сети на разделы.....	209
6.2.2	Кластеризация данных	212
6.3	Обнаружение сообществ.....	215
6.3.1	Устранение мостов.....	216
6.3.2	Оптимизация модулярности.....	218
6.3.3	Распространение меток.....	225
6.3.4	Стохастическое блочное моделирование.....	227
6.4	Оценивание методов	230
6.4.1	Искусственные эталоны.....	230
6.4.2	Реально существующие эталоны.....	233
6.4.3	Сходство между разделами.....	234
6.5	Резюме.....	236
6.6	Дальнейшее чтение.....	237
	Упражнения.....	238
7	Динамика.....	244
7.1	Идеи, информация, влияние	246
7.1.1	Пороговые модели	247
7.1.2	Независимо-каскадные модели	250
7.2	Распространение эпидемий.....	252
7.2.1	Модели SIS и SIR	254
7.2.2	Распространение слухов	259
7.3	Динамика мнений	261
7.3.1	Дискретные мнения	262
7.3.2	Непрерывные мнения.....	265

7.3.3 Козволюция сетей и динамика	267
7.4 Поиск	270
7.4.1 Локальный поиск	270
7.4.2 Доступность поиска	273
7.5 Резюме	278
7.6 Дальнейшее чтение	280
Упражнения	281
Приложение А. Руководство по языку Python	288
A.1 Блокнот Jupyter	288
A.2 Условный блок	289
A.3 Списки	290
A.4 Циклы	292
A.5 Кортежи	295
A.6 Словари	297
A.7 Комбинирование типов данных	300
A.7.1 Список кортежей	300
A.7.2 Список словарей	301
A.7.3 Словарь словарей	302
A.7.4 Словарь с кортежными ключами	302
A.7.5 Еще один словарь словарей	303
Приложение В. Модели NetLogo	305
B.1 Модель PageRank	306
B.2 Гигантская компонента	307
B.3 Малые миры	308
B.4 Преференциальное прикрепление	309
B.5 Вирус в сети	310
B.6 Изменение языка	312
Справочные материалы	314
Предметный указатель	331



Предисловие

Сети присутствуют во всех аспектах нашей жизни: круг друзей, коммуникационные и транспортные сети, а также Веб как Всемирная паутина – все это примеры, которые мы воспринимаем внешне, тогда как нейроны в нашем мозге и белки в нашем теле образуют сети, которые определяют наш интеллект и выживание. Когда люди общаются в Facebook или Twitter, покупают что-то в Amazon, ищут в Google или покупают авиабилет, чтобы навестить семью, они используют сети, не осознавая того. Сегодня базовое понимание сетевых процессов требуется в различных сферах деятельности – от технологий до маркетинга, от менеджмента до дизайна, от биологии до искусства и гуманитарных наук. В этом учебнике проводится разведывательный анализ учения о сетях и то, как сети помогают нам понимать сложные шаблоны взаимоотношений, которые формируют наши жизни.



Эта книга тоже является сетью! На приведенном выше рисунке показаны взаимоотношения между главами, разделами и подразделами. Связи представляют и иерархическую структуру книги (как показано в Оглавлении), и перекрестные ссылки между главами, разделами,

рисунками, таблицами, уравнениями и вставками. Цвета узлов представляют главы, а размер узла пропорционален числу соседей.

Зачем нужен «Вводный курс» по науке о сетях?

Это не первая книга о науке о сетях – на самом деле есть несколько отличных книг на выбор, и мы перечислим некоторые из них в главе 1. Мы преподаем эти темы уже в течение нескольких лет в Университете Индианы для широкой аудитории студентов старших курсов по информатике, теории вычислительных машин, науке о данных, теории информации, бизнесу, естественным и социальным наукам. Этот опыт научил нас тому, что студенты стремятся «пачкать свои руки» выполнением черновой работы и писать исходный код, чтобы понимать и использовать сети в интересующих их областях применения, даже если они только учатся программировать и не имеют математического и компьютерного образования за пределами средней школы и курсов начального уровня колледжей. Поэтому мы разработали широкий круг учебно-практических занятий и задач, как теоретических, так и вычислительных, предоставив студентам обилие практических занятий по науке о сетях. Используя такой подход, книга знакомит с сетями широкую аудиторию студентов, не имеющих никаких технических предпосылок, кроме какого-то вводного программирования и готовности учиться на деле. Это делает наш учебник пригодным для «вводного курса» науки о сетях.

Синописис

После проведения обзора сетей, существующих во многих областях человеческих знаний и деятельности, мы поговорим о социальных сетях, которые знакомы студентам больше всего. Это позволяет ввести такие понятия, как маломировое свойство (короткие пути) и кластеризация (треугольники и транзитивность). Указанные темы объясняются с использованием увлекательных учебных занятий, таких как игра *«Шесть степеней Кевина Бейкона»*. Затем мы погрузимся в роль хабов, используя Парадокс дружбы, и обсудим тему устойчивости сетей. Далее мы вводим соответственно направленные и взвешенные сети. Всемирная паутина, «Википедия», цитирование, трафик и Twitter используются для иллюстрации роли направления и веса. Последние три главы охватывают более сложные темы, а именно модели возникновения сетей, методы обнаружения сообществ и динамические процессы, происходящие в сетях.

В каждой главе рассматриваются базовые концепции, необходимые для понимания фундаментального аспекта сетей; избегаются

сложные темы и формализм. Когда это полезно, мы включаем немного математики во вставки, обрамленные рамкой. В них находится чуть-чуть более технический материал, и его можно пропустить без потери базового понимания темы. Но студенты, которые будут следить за этими дополнительными примечаниями, смогут получить более глубокое понимание материала. Каждая глава включает в себя учебно-практические занятия по программированию и упражнения, позволяющие читателям применять и проверять свои знания с помощью практических занятий по строительству и анализу сетей. Указанные учебно-практические занятия основаны на примерах реально существующих сетей, которые используются для иллюстрации концепций на протяжении всей книги. И учебно-практический исходный код, и сетевые данные доступны в репозитории книги на GitHub¹.



Целевая аудитория

С ростом популярности и коммерческого успеха онлайн-социальных сетей многие студенты заинтересованы в том, чтобы узнать немного о том, что находится «под капотом» таких сетей. Данный учебник предназначен для всех этих студентов в основном на уровне бакалавриата, хотя книга, возможно, будет полезна и для вводных курсов аспирантуры в нетехнических областях. Студенты, обучающиеся по таким программам, как наука о данных, информатика, бизнес, теория вычислительных машин, машиностроение, теория информации, биология, физика, статистика и социальные науки, получают пользу от курсов, основанных на этом учебнике. Их интерес будет достаточно велик, чтобы изучить науку о сетях глубже, и, возможно, они выберут карьеру, которая поможет найти им работу в Google, Facebook, Twitter или организовать свой собственный сетевой стартап.



Педагогика

Настоящий курс не требует никакого технического образования в области математики или программирования, что делает эту книгу пригодной для вводных курсов любого уровня, включая курсы сетевой грамотности и грамотности в программировании. Подобного рода курсы могут пропускать математические вставки. Отрабатывая учебно-практические занятия по программированию в коллаборативной вычислительной лаборатории и назначая упражнения по программированию, преподаватели предоставят студентам возможность приобрести технические навыки, достаточные для выполнения задач

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

анализа данных, связанных с сетями. Таков наш подход в Университете Индианы, где мы преподаем материал данной книги в течение двух курсов: первый вводный курс, предназначенный для студентов-второкурсников / младших курсов, которые прошли или проходят курсы конкурентного программирования на Python; и второй курс, предназначенный для студентов младших/старших курсов. Первый курс примерно охватывает материал глав с 0 по 4. Второй курс сосредоточен на главах 5–7 после расширенного обзора и нескольких более продвинутых учебно-практических занятий по предыдущему материалу.

Обширные учебно-практические занятия по программированию и упражнения позволяют преподавателям легко руководить учебным процессом и проводить практические мероприятия, а также позволяют студентам укрепить и проверить свое понимание сетевых концепций. Мероприятия включают учебно-практические занятия по *NetworkX*, широко распространенной библиотеке для сетевой аналитики; и по всем затронутым в книге темам, от базовых упражнений до передовых методов. Например, на одном учебно-практическом занятии студенты знакомятся с шагами извлечения данных социальных сетей из Всемирной паутины. Используя интерфейс прикладного программирования (API) Twitter, студенты смогут анализировать популярные темы, выявлять влиятельных пользователей и реконструировать сети диффузии информации, показывающие процесс онлайн-распространения хештегов. Студенты, которые проходят учебно-практические занятия и выполняют упражнения по программированию, наберутся опыта в строительстве, импортировании/экспортировании, анализировании, манипулировании и визуализировании сетей любого типа.

Учебно-практические занятия основаны на языке Python, т. е. самом популярном языке для написания скриптов/программ. Учебное руководство, в котором рассматриваются главные концепции программирования на Python, включен в приложение А книги. Все учебно-практические занятия доступны онлайн в виде блокнотов Jupyter/IPython. Со временем библиотека NetworkX (и даже язык Python), возможно, эволюционирует, и, возможно, потребуется обновить часть исходного кода книги. Мы будем отмечать такие обновления в репозитории книги на GitHub.

Разумеется, для программирования сетей существуют и другие библиотеки, например *igraph*, *SNAP* и *graph-tool*. Наш выбор библиотеки NetworkX основан на том факте, что она написана на чистом Python, что облегчает отладку для студентов, знакомых с Python. Многие альтернативы имеют интерфейсы Python, но написаны на языке C, что делает их эффективнее, но и сложнее в отладке.

Наконец, в некоторых главах используются интерактивные модели для демонстрации сетевых явлений, таких как гигантские компоненты, малые миры, алгоритм PageRank, предпочтительное прикреплени-

ние и эпидемии. Эти модели работают в NetLogo, популярной симуляционной платформе. Учебно-практический материал по NetLogo и несколько наиболее актуальных моделей представлены в приложении В книги.



Об обложке

Сеть на обложке, сгенерированная Онуром Варолом (Onur Varol) (Феррара и соавт., 2016), изображает диффузию хештега #SB277 в Twitter. Этот хештег относится к калифорнийскому закону 2015 года об обязательствах в отношении вакцинации и освобождении от нее, и указанная сеть изображает обсуждение, которое проходило онлайн среди сторонников и противников указанного законопроекта. Узлы изображают пользователей Twitter, а связи показывают информацию, распространяемую среди пользователей через ретвиты. Размер узла отражает влияние учетной записи (сколько раз пользователь ретвитнул), а цвета узлов иллюстрируют баллы ботов: красные узлы, скорее всего, являются учетными записями ботов, синие узлы, скорее всего, являются людьми.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.



Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.



Признательности

Первоначальная идея этой книги возникла из бесед с нашим бывшим коллегой Алексом Веспиньяни. На протяжении многих лет наши коллеги Сандро Фламмини, Йи Ан и Филиппо Радиччи давали ценные советы. Несколько студентов оказало помощь в преподавании наших курсов науки о сетях в Университете Индианы. Среди них мы хотим отметить Майка Коновера, который первым задумал некоторые упражнения, описанные на страницах данной книги. Мы также благодарны коллегам, которые предоставили отзывы о ранних набросках книги, в особенности Клаудио Кастеллано, Чато Кастильо и несколькими анонимным рецензентам.

Мы благодарим наших замечательных сотрудников, студентов, аспирантов и посетителей: Ану Магитман, Бена Маркинеса, Бруно Гонсалвеса, Ченчен Шао, Дип Тхи Хоанга, Димитара Николова, Эмилио Феррару, Джованни Луку Чампалью, Якоба Раткевича, Джаслин Каур, Хосе Рамаско, Кай-Чен Яна, Кариссу Маккелви, Кадзу Сасахару, Ле-Шин Ву, Лилиан Венг, Луку Марию Алелло, Марка Мейса, Маркуса Якобссона, Михая Аврама, Никола Перра, Онура Варола, Пик-Май Хуэй, Прашанта Ширалкара, Пшемека Грабович, Руж Акавипата, Сяодан Лу, Сяолин Сун, Зохера Качвалу и многих других среди прочих. Это удивительно яркая и веселая группа людей, которые внесли свой значительный вклад в идеи, наборы данных и приводимые к книге иллюстрации.

Наша работа была бы невозможна без поддержки многих преданных своему делу сотрудников Центра исследований сложных сетей и системных исследований, Школы информатики, вычислительной техники и машиностроения и Института науки о сетях Университета Индианы. Прежде всего мы с благодарностью должны отметить Тару Холбрук, Мишель Домпке, Роба Хендерсона, Дейва Кули, Пэтти Мэбри, Энн Маккрейни, Вэла Пентчева, Мэтью Хатчинсона, Чатури Пели Канканамалаге и Бена Серретта. Спасибо также Нику Гиббонсу из издательства Кембриджского университета за его поддержку и отзывы.

Мы благодарны Арику Хагбергу, Питеру Сварту и Дэну Шульту, авторам NetworkX, а также Ури Виленски и Центру связного обучения и компьютерного моделирования Северо-Западного университета за разработку и поддержку NetLogo.

Наконец, мы в огромном долгу перед нашими семьями, которые любят, поддерживают и терпят нас, даже когда мы работаем больше, чем должны.

Введение

Сеть: взаимосвязанная или взаимодействующая цепочка, группа или система.



Вообразите мир, в котором у людей нет друзей. Где дороги никогда не пересекаются. Где компьютеры не связаны между собой. Этот мир без сетей был бы очень грустным и скучным местом, где ничего не происходит, – и даже если бы что-то случилось, никто бы об этом не узнал. Такой мир невообразим, потому что наша жизнь полностью определяется сетями: взаимоотношениями, взаимодействиями, каналами связи и Всемирной паутиной. Биологические сети, управляющие взаимодействиями между генами в наших клетках, определяют наше развитие, нейронные сети в мозге наделяют нас возможностью думать, информационные сети направляют наши знания и культуру, транспортные сети позволяют двигаться, а социальные сети подпирают нашу жизнь.

Сети – это общий, но мощный способ представления и изучения простых и сложных взаимодействий. В этой книге проводится разведывательный анализ учения о сетях и того, как они помогают нам понимать закономерности соединений и взаимоотношений, которые формируют нашу жизнь. По своей сути сеть – это простейшее описание множества взаимосвязанных сущностей, которые мы называем *узлами*, и их соединений, которые мы называем *связями*. Сетевое представление является столь общим и мощным, потому что оно устраняет многие детали конкретной системы и фокусируется на взаимодействиях между ее элементами. Отсюда сети используются для изучения самых разнообразных систем. Узлы могут представлять все виды сущностей: людей, города, компьютеры, веб-сайты, концепции, клетки, гены, виды животных и т. д. Связи представляют взаимоотношения или взаимодействия между этими сущностями: дружеские связи между людьми, рейсы между аэропортами, пакеты, которыми компьютеры обмениваются в интернете, связи между страницами Всемирной паутины, синапсы между нейронами и т. д.

Прежде чем мы представим базовые понятия, определения и номенклатуру сетей, давайте рассмотрим несколько примеров социальных, инфраструктурных, информационных и биологических сетей. Данные для всех представленных здесь примеров доступны в репозитории книги на GitHub¹. Сети, на которых мы сосредоточимся

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

в этой книге, как правило, являются крупными, хотя многому можно научиться, изучая и меньшие системы, такие как социальные сети, созданные на основе опросов или собеседований. В этих случаях имеет смысл детально проинспектировать отдельные узлы и соединения, тогда как анализ крупных сетей, как правило, фокусируется на макроскопических свойствах, классах узлов и связей, типичных проявлений поведения и аномалиях.

https://t.me/it_boooks

0.1. Социальные сети

Социальная сеть – это группа людей, связанных каким-либо типом взаимоотношения. Дружба, сотрудничество, романтика или простое знакомство – все это примеры социальных взаимоотношений, которые соединяют пары людей. Когда мы говорим о социальной сети, мы обычно думаем об определенном типе взаимоотношения. Человек представляется узлом в социальной сети, а взаимоотношение представляется связью между двумя людьми. Таким образом, сеть является представлением взаимоотношения. Это позволяет нам говорить о взаимоотношениях, описывать их и анализировать на уровне, выходящем за рамки пары людей.

Существует много разных типов социальных сетей, и их важно изучать. Медицинские работники анализируют сети сексуальных отношений, чтобы отыскивать способы борьбы с распространением заболеваний, передающихся половым путем. Экономисты изучают сети направления на работу для решения проблемы неравенства и сегрегации на рынках труда. А ученые инспектируют сети соавторства в научных публикациях, чтобы выявлять влиятельных мыслителей и идеи.

В наши дни мы используем веб-сайты онлайн-социальных сетей, чтобы отслеживать социальные связи. Такие платформы, как Facebook и Twitter, позволяют нам поддерживать связь со многими людьми – партнерами, друзьями, коллегами и знакомыми, иногда сотнями, – и комфортно с ними общаться независимо от расстояния. На рис. 0.1 показана сеть знакомств, часть социального графа Facebook. В этой сети узлами являются люди с учетной записью Facebook в университетах США, и соединения могут представлять различные типы взаимоотношения, от настоящей дружбы до простого знакомства. Просто взглянув на визуализацию сети, вы узнаете кое-что о лежащей в ее основе социальной структуре. У некоторых людей связей больше; мы представляем это, делая соответствующие узлы больше и темнее. Это могут быть популярные студенты, преподаватели или администраторы. Мы также замечаем, что сеть примерно поделена на две части. Данные анонимны, поэтому мы не можем сказать наверняка, но возможной интерпретацией будет то, что крупная подсеть

включает в себя в основном студентов старших курсов, а меньшая – в основном аспирантов. Между узлами в двух группах есть соединения, но их не так много, как между узлами внутри каждой группы. Другими словами, студенты старших курсов с большей вероятностью будут дружить с другими студентами, чем с аспирантами. Позже для всех этих наблюдений, которые типичны для большинства социальных сетей, мы введем формальные названия.

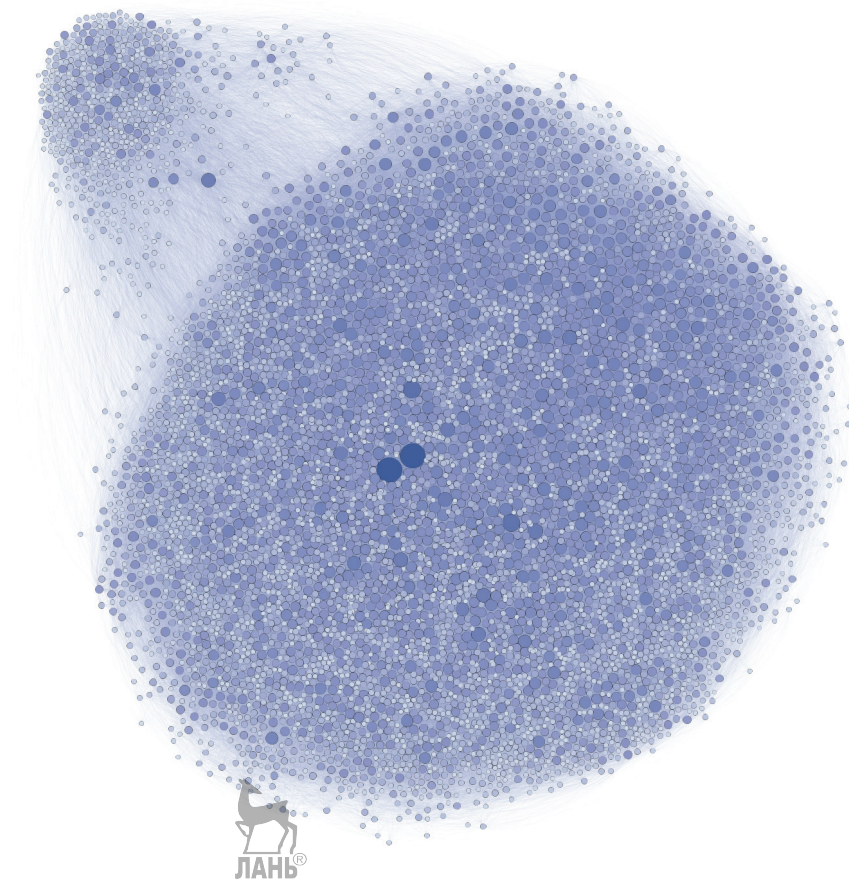


Рис. 0.1 Визуализация сети пользователей Facebook в Северо-Западном университете. Узлы обозначают людей, а связи обозначают соединения между друзьями в Facebook

Доступность данных из онлайн-социальных сетей очень увлекает ученых. Мы можем изучать человеческие взаимодействия в масштабе и в разрешающей способности, которые никогда не были возможны в прошлом: кто с кем дружит, кто на что обращает внимание, кому что нравится, что рекомендуется и как эта информация распространяется по сети. Эти данные предоставляют нам беспрецедентную возможность обнаруживать, отслеживать, добывать и моделировать



то, что делают люди. Подобно тому, как телескоп позволил нам впервые увидеть далекие планеты и звезды, а микроскоп – заглянуть в живые ткани и микроорганизмы, социальные сети позволяют изучать социальные системы и человеческую деятельность. Однако какими бы захватывающими ни были эти возможности для исследователей, они не обходятся без риска злоупотреблений. Онлайн-взаимодействия раскрывают нашу приватную информацию. Мы все слышали истории о том, как работодатели находили неловкие фотографии потенциальных сотрудников или о скандалах, имеющих отношение к хакерам и политическим организациям, собирающим данные о миллионах пользователей. Опасности бывают едва уловимыми. Обладание небольшим объемом информации о большом числе людей может раскрывать гораздо больше, чем предполагалось. Используя данные из Facebook, два студента Массачусетского технологического института обнаружили, что, просто взглянув на пол и сексуальность онлайн-друзей человека, они могут предсказывать, является этот человек геем или нет. Онлайн-социальные сети также облегчают выдачу себя за другого человека и затрудняют ее обнаружение. Выживание информации из социальных сетей (социальный фишинг) – это метод выставления себя за друга жертвы (логически выводимого из данных онлайн-социальной сети), чтобы побуждать жертву раскрывать конфиденциальную информацию. Два студента Университета Индианы продемонстрировали, что таким образом им удалось получить секретные пароли 72 % жертв.

Данные о социальной сети можно извлекать из многочисленных источников. Если мы хотим картировать шаблоны мобильности людей с целью улучшения городских транспортных сетей, то мы можем собирать данные о звонках с мобильных телефонов. Если хотим картировать соавторство среди ученых, то можем извлекать имена из базы данных научных публикаций; два соавтора одной и той же статьи будут связаны друг с другом. (Это не тривиальное упражнение, потому что у нескольких ученых могут быть общие имена.) Если мы хотим картировать сотрудничество между кинозвездами, можем извлекать данные о титрах кинофильмов из интернет-базы данных кинофильмов (Internet Movie Database, IMDb.com). На рис. 0.2 показаны две такие сети. В одном случае на самом деле существует два вида узлов: кинофильмы и актеры/актрисы. Мы проводим связь между актрисой и кинофильмом, в котором она снялась. В другом случае мы фокусируемся на связях между актерами/актрисами, которые снимались в фильмах вместе. Хотя изображенные сети улавливают лишь крошечные части базы данных кинофильмов, мы снова замечаем некоторые четкие закономерности. Более крупные узлы имеют больше соединений, представляющих звезд, которые снимались во многих кинофильмах. Мы также видим, что сети структурированы в несколько плотных групп, связанных с периодами, языками или жанрами кинофильмов: голливудские (синие), европейские (голубые), мексиканские (фиолетовые), китайские (желтые), филиппинские (оранжевые),

турецкие и восточноевропейские (зеленые), индийские (красные), греческие (белые) кинозвезды и кинозвезды фильмов для взрослых (розовые) на рис. 0.2(b). В главе 6 вы узнаете, как обнаружить эти группы и выяснять, чему они посвящены.

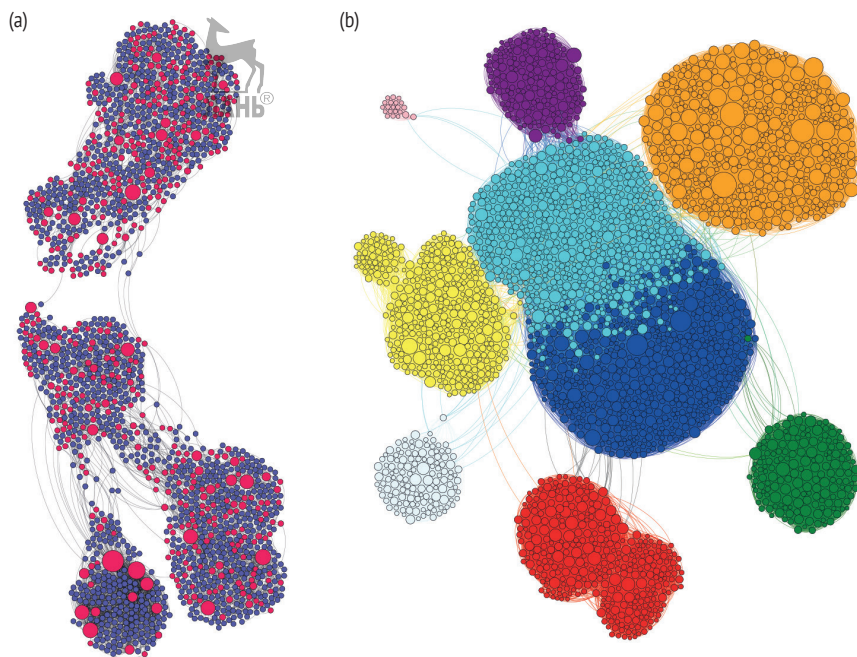


Рис. 0.2 (а) Сеть кинозвезд, основанная на небольшой выборке кинофильмов, актеров и актрис из интернет-базы данных кинофильмов. Узлы представляют кинофильмы (синие) или актеров/актрис (красные). Связь соединяет актера или актрису с фильмом, в котором они снимались. (б) Сеть кинозвезд, основанная на небольшой выборке актеров и актрис из интернет-базы данных кинофильмов, снимавшихся вместе с другой кинозвездой. Связь соединяет двух людей, которые снялись вместе по меньшей мере в одном фильме. Цвета представляют жанры фильмов или языки/страны

0.2. Коммуникационные сети

В сетях Facebook и кинофильмов связи взаимны: вы не можете подружиться с кем-либо на Facebook, если он не согласен, и вы не можете быть снятым в фильме, не будучи упомянутым в титрах. Однако не все социальные сети имеют взаимные связи. Например, Twitter представляет собой популярную социальную сеть со связями, которые не обязательно являются взаимными: Алиса может следить за Бобом без того, чтобы Боб обязательно следил за Алисой. Как следствие, запечатленные в сети Twitter отношения не являются дружбой; вы подписываетесь на кого-то, чтобы увидеть, что он публикует. Когда вы ретвитите сообщение, его видят ваши подписчики. Это хороший

способ широко обмениваться информацией, поэтому Twitter является социальной сетью, в основном направленной на распространение информации, т. е. коммуникационной сетью. Ретвитная сеть на рис. 0.3 иллюстрирует распространение политических сообщений во время выборов в США. Более крупные узлы являются узлами с большим числом исходящих связей, потому что число ретвитов пользователей другими пользователями является способом измерить их влияние. Вы, вероятно, сразу заметили более поразительную закономерность: консервативные пользователи (красные узлы) в основном ретвитят сообщения от других консерваторов, в то время как прогрессивные пользователи (синие узлы) аналогичным образом делятся прогрессивным контентом. На самом деле такие предпочтительные регулярности социальных связей позволяют нам с высокой точностью угадывать политические склонности человека. Это свойство, именуемое *гомофилией*, будет обсуждаться в главе 2; алгоритм определения политических предпочтений по структуре сети будет представлен в главе 6.

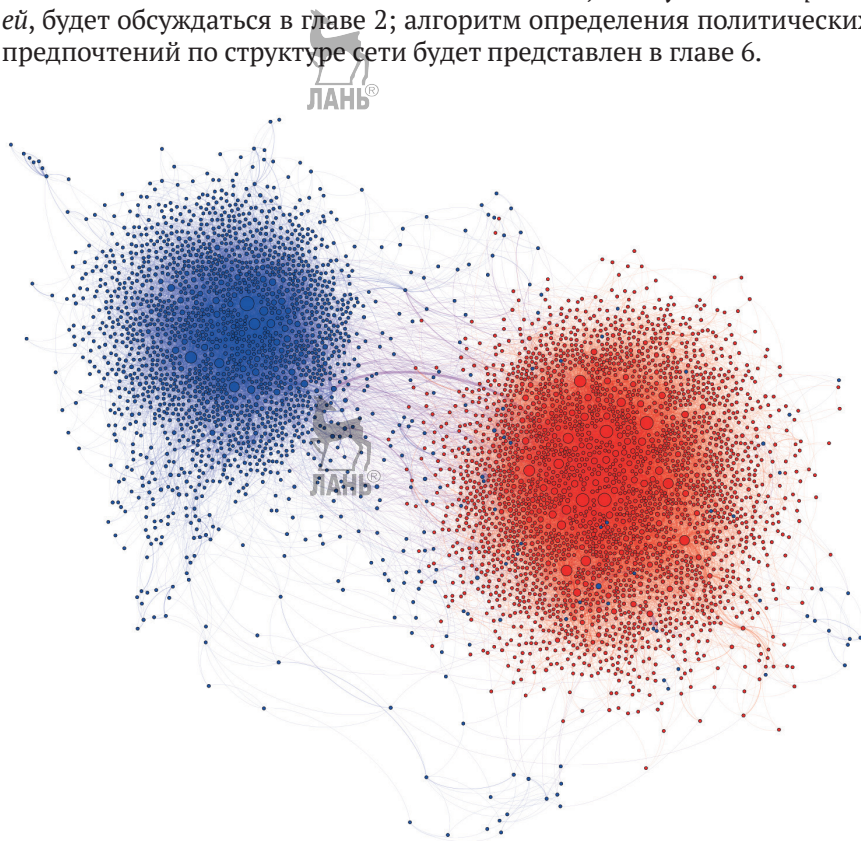


Рис. 0.3 Ретвитная сеть в Twitter среди людей, делящихся постами о политике США. Связи представляют собой ретвиты сообщений, в которых использовались хештеги, такие как #tcot и #p2, связанные соответственно с консервативными (красными) и прогрессивными (синими) сообщениями во время промежуточных выборов в США 2010 года. Когда Боб ретвитит Алису, мы рисуем направленную связь от Алисы к Бобу, чтобы обозначить, что сообщение перешло от нее к нему. Направление связей не показано

Такие сети, как Twitter, позволяют нам отслеживать диффузию хештегов и новостей, наблюдая за тем, как идеи и культурные концепции распространяются от человека к человеку. Но социальные сети также используются для распространения дезинформации, которая неосознанно передается доверчивыми пользователями. Используя поддельные новостные веб-сайты и автоматизированные или полуавтоматические учетные записи, именуемые «социальными ботами», вредоносная организация может дешево и эффективно генерировать и усиливать кампанию по дезинформации в политических целях либо для монетизации трафика с помощью рекламы. В последние годы мы наблюдаем резкий рост подобных видов манипуляций социальными сетями в глобальном масштабе. Если кто-то может контролировать информацию, которую люди видят онлайн, то он может манипулировать их мнением. Во многих странах это явление представляет угрозу демократии, потому что без хорошо информированных избирателей невозможно проводить свободные выборы. Академические исследователи и промышленные инженеры усердно работают над разработкой контрмер. Понимание структуры и динамики сетей, обеспечивающих распространение информации, является важнейшим компонентом этих усилий.

Социальные связи в Twitter создаются до того, как пользователь создает сообщение, которое обычно транслируется всем подписчикам пользователя. В электронной почте, как и в социальных сетях, узлы являются людьми. Однако каждое сообщение предназначено для одного или нескольких конкретных получателей. Связи основаны на обмениваемых сообщениях. Электронная почта не зависит от конкретной платформы; протокол открыт и распространяется, вследствие чего ни одна организация не контролирует весь трафик. Как следствие, электронная почта по-прежнему остается одной из наиболее широко используемых коммуникационных сетей. На рис. 0.4 показан пример сети электронной почты. Опять же, связи направляются от отправителя к получателю электронного письма обозначенными стрелками. Размер и цвет узла представляют два разных признака: число соответственно входящих и исходящих связей. Более крупный узел получает электронные письма от большего числа отправителей, а более темный узел отправляет электронные письма большему числу получателей. Тот факт, что более крупные узлы, как правило, темнее и наоборот, говорит нам о том, что между отправкой и получением электронных писем существует корреляция.



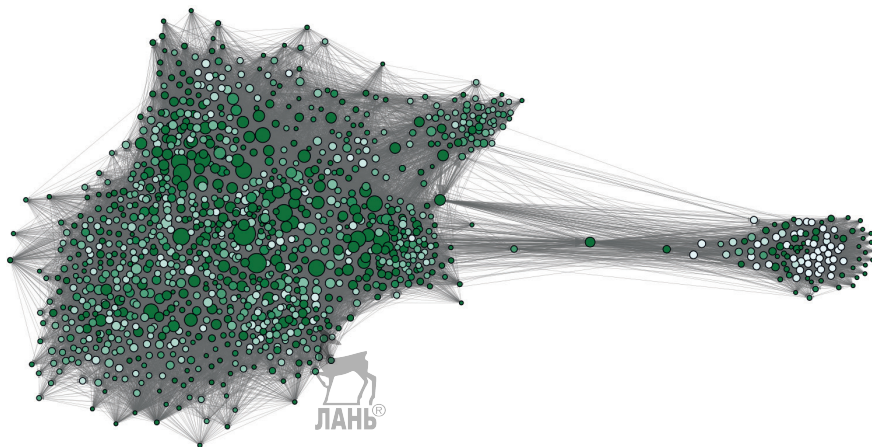


Рис. 0.4 Сеть, опирающаяся на базу данных электронных писем, сгенерированных сотрудниками энергетической компании Enron. Эти данные были получены Федеральной комиссией по регулированию энергетики США в ходе расследования, проведенного после краха компании в 2001 году. По завершении расследования электронные письма были признаны как находящиеся в публичном пространстве и сделаны общедоступными для исторических исследований и академических целей. Показана только небольшая часть центрального ядра сети. Направление связей показано стрелками

0.3. Всемирная паутина и «Википедия»

Всемирная паутина (Веб) – это крупнейшая информационная сеть. Хотя сейчас она используется для предоставления всех видов услуг, изначально это была просто сеть документов (страниц), соединенных «гиперсвязями», или кликабельными связями. В начале 1990-х годов Тим Бернерс-Ли захотел упростить доступ ученых к информации об экспериментах по физике высоких энергий в Европейской организации ядерных исследований (CERN) недалеко от Женевы. Он выдвинул три ключевые идеи: (1) систему именования страниц, Единый локатор ресурсов (Uniform Resource Locator, URL); (2) простой язык для написания документов, именуемый языком разметки гипертекста (HyperText Markup Language, HTML), включая гиперсвязи из одной страницы на другую; и (3) простой протокол, именуемый протоколом передачи гипертекста (HyperText Transfer Protocol, HTTP), для программ-клиентов (браузеров), чтобы общаться с серверами. Благодаря этим трем компонентам родилась Всемирная паутина. Бернерс-Ли даже имплементировал первый веб-сервер и программно-информационное обеспечение для браузера, чтобы скачивать страницы и мультимедиа с серверов, нажимая на связи. На самом деле мы можем видеть здесь участие двух сетей: статический «граф связей», состоящий из моментального снимка веб-страниц и связей в данный момент времени,

и динамическую сеть трафика, возникающую в результате передвижения людей по сети. Перефразируя классическую философскую загадку, если между двумя страницами есть связь, но никто на нее не нажимает, действительно ли она является частью паутины? Ответ, конечно, зависит от того, о какой из двух сетей мы думаем, когда произносим слово «паутина». В последующих главах мы потратим больше времени на разведывание обеих этих информационных сетей.

Всемирная паутина слишком велика, чтобы визуализировать даже малую ее часть осмысленным образом. Давайте сосредоточимся на «Википедии», которая представляет собой сеть страниц (статей) на одном веб-сайте. «Википедия» – это коллаборативная энциклопедия, редактируемая тысячами добровольцев по всему миру, и это одно из самых популярных направлений во Всемирной паутине. Существуют версии «Википедии» на многих языках, поэтому давайте сосредоточимся на английской. Тем не менее английская «Википедия» представляет собой огромную сеть с миллионами статей (и она растет!). Поэтому давайте сосредоточимся лишь на небольшом подмножестве статей по математике, показанном на рис. 0.5. Здесь размер узла представляет метрику *PageRank*, меру значимости, отражающую степень важности статьи на основе других статей, которые имеют с ней связь, – тему нашего обсуждения в главе 4. Например, крупный белый узел посередине – это общая статья по *математике*. Еще одним при-

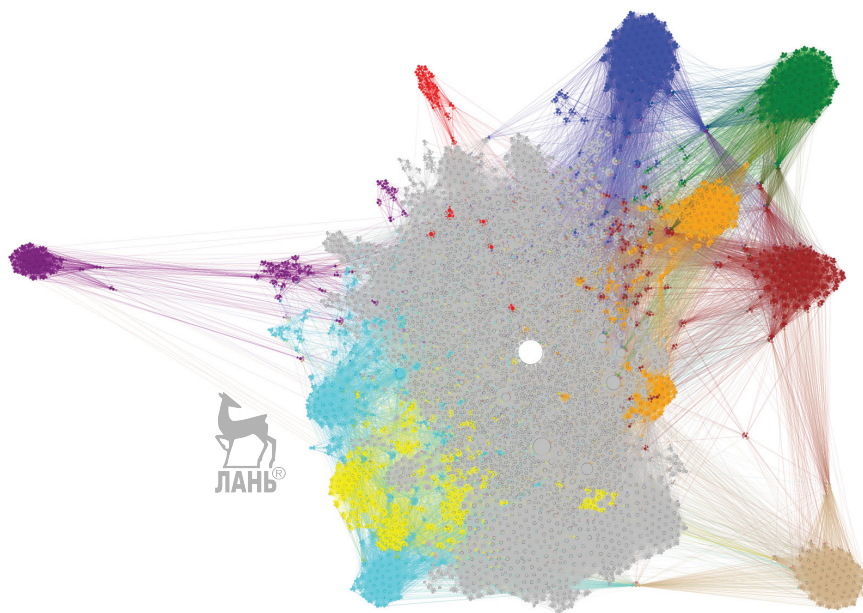


Рис. 0.5 Часть информационной сети «Википедии». Узлы – это статьи о математике. Мы рассматриваем связи только между статьями «Википедии» и игнорируем связи с внешними страницами. Размер узла пропорционален важности статьи, а цвета выделяют сообщества, обсуждаемые в тексте

знаком этой сети является наличие крупного «ядра» (серого цвета) и нескольких малых групп. Эти группы представляют собой тесно связанные группы статей по конкретным темам или разделам математики. Например, статьи об исторических греческих (синих), арабских (зеленых) и индийских (коричневых) математиках; о современных индийских математиках (коричневый); о математике и искусстве (оранжевый), статистике (голубой), теории игр (желтый), математическом программно-информационном обеспечении (фиолетовый) и педагогической теории (красный). Мы также наблюдаем несколько «мостовых» узлов, которые соединяют несколько кластеров. Эти признаки можно найти во многих реально существующих сетях.

0.4. Интернет

Мы часто думаем об интернете как о сети компьютеров и других присоединенных устройств, но в реальности это *сеть сетей*. На самом деле это слово происходит от английского слова *internetworking*, т. е. *межсетевое взаимодействие*, или соединения разных компьютерных сетей через специальные узлы, именуемые *маршрутизаторами*. И по этой причине мы можем наблюдать интернет на многих уровнях: на самом низком уровне у нас аппаратные устройства, которые соединяют отдельные компьютеры в одну локальную или широко-масштабную сеть. Эти сети соединяются маршрутизаторами, поэтому мы можем уменьшать масштаб и думать о сети маршрутизаторов. Если мы еще больше уменьшим масштаб, то обнаружим группы сетей, управляемых провайдером интернет-служб (Internet Service Provider, ISP). Эта организация определяет свою внутреннюю сетевую топологию (способ соединения маршрутизаторов) самостоятельно и поэтому также называется «автономной системой» (AS). Специальные «пограничные» маршрутизаторы соединяют одну автономную систему с другой, образуя то, что мы называем сетью автономных систем.

На рис. 0.6 показана небольшая часть сети интернет-маршрутизаторов. Хотя интернет развивался без централизованного контроля или координации, провайдеры интернет-служб соблюдают локальные правила соединения своих маршрутизаторов. Они стараются обеспечивать наилучшее обслуживание по самой низкой цене. В результате возникают определенные регулярности. Например, та часть интернета, которая обеспечивает наибольший трафик, часто называется «магистралью». Крупные телекоммуникационные компании, управляющие интернет-магистралью, заинтересованы в предотвращении сбоев, поэтому они конструируют свои сети с большой избыточностью. Осюда мы наблюдаем плотное «ядро», в котором крупные маршрутизаторы соединены друг с другом. По мере того как мы продвигаемся к «периферии» интернета – нашим домашним маршрути-

заторам, – сеть становится соединенной все более разреженно. Подобного рода иерархическая *структура ядро–периферия* распространена в многочисленных разных типах сетей и будет обсуждаться в главе 2. В сети маршрутизаторов, изображенной на рис. 0.6, зеленый кластер слева хорошо сепарирован от остальной сети. Вероятно, это обусловлено систематическим смещением в методологии зондирования, используемой для картирования этих сетей: большинство измерений было проведено в Соединенных Штатах, и маршрутизаторы в этом кластере расположены там. Относящейся к этому отличительной особенностью является наличие очень крупных узлов в зеленом кластере, что указывает на маршрутизаторы с большим числом соединений. На самом деле это может быть ошибкой измерения, вызванной тем же систематическим смещением. Ввиду аппаратных ограничений маршрутизатор фактически может иметь только ограниченное число соединений. Пусть это послужит напоминанием о том, что если мы используем ущербный метод сбора данных о сети, то его анализ может привести к неправильным выводам.

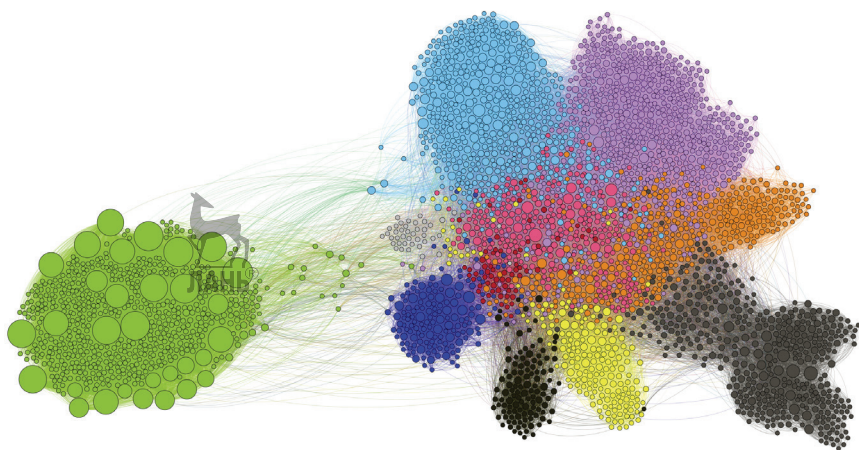


Рис. 0.6 Часть сети интернет-маршрутизаторов. Карта представляет собой снимок, созданный Центром прикладного анализа интернет-данных (Center for Applied Internet Data Analysis, CAIDA.org) с использованием инструментов, которые отправляют малые пакеты данных (зонды) между хостами интернета. Цвета назначаются в соответствии с алгоритмом обнаружения сообществ, который выявляет плотные кластеры, отражающие географическое распределение маршрутизаторов. В главе 6 вы узнаете, как использовать эту методологию для изучения того, что эти кластеры представляют

0.5. Транспортные сети

Еще один важный класс сетей касается различных видов транспортных перевозок. Узлами являются местоположения: города, перекрестки дорог, аэропорты, порты, железнодорожные станции или станции

метро. Однако эти сети сильно отличаются друг от друга. Например, дорожные сети развиваются локально, чтобы минимизировать расстояние, проходимое между близлежащими городами. Это приводит к появлению решетчатых структур, в которых большинство узлов имеет сопоставимое число соединений – к примеру, четырехпутные пересечения. На рис. 0.7 показана сеть авиационных перевозок, которая не имеет решетчатой структуры. Причина в том, что авиакомпании стараются минимизировать число перелетов между пунктом отправления и пунктом назначения, не добавляя дорогостоящих прямых рейсов между аэропортами с низким трафиком. Простое решение состоит в добавлении рейсов, соединяющих аэропорты с существующими хабами, действующими как транспортно-пересадочные узлы. Как следствие, сети авиарейсов имеют структуру «хаб и спица» (hub and spoke): несколько хабов имеет огромные числа связей, тогда как большинство узлов имеет очень мало соединений.

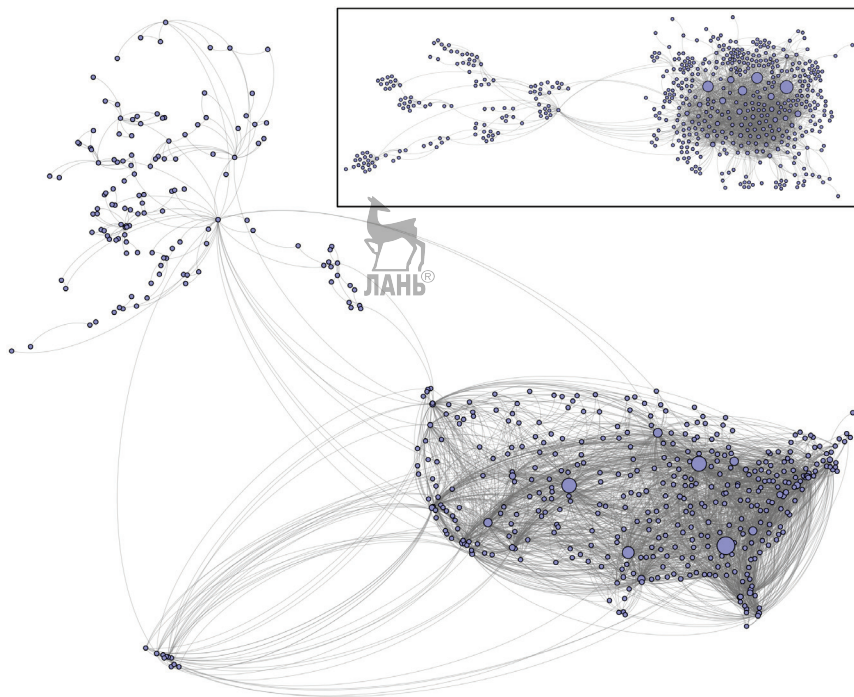


Рис. 0.7 Сеть авиационных перевозок США (данные о рейсах из OpenFlights.org). Узлы расположены в соответствии с географическими координатами соответствующих аэропортов, вследствие чего мы можем различить очертания континентальной части Соединенных Штатов, Аляски и Гавайев. Обратите внимание, что проекция карты делает Аляску больше, чем ее фактический размер, из-за ее широты. Авиационные хабы с большинством соединений (например, Атланта, Чикаго, Денвер) четко узнаваемы. Вставка отображает ту же сеть, но с другой «направляемой силой» компоновкой, обсуждаемой в разделе 1.10

При изучении определенных типов сетей, в особенности относящихся к транспортным перевозкам и коммуникациям, мы можем рассуждать о них с точки зрения их статической структуры или динамических процессов, происходящих в этих сетях. Возьмем, например, сеть авиационных перевозок. Мы могли бы рассматривать карту на рис. 0.7 как множество маршрутов, существующих между аэропортами, независимо от фактического движения по ним; или как транспортную сеть, возникающую в результате передвижения людей между аэропортами. В последнем смысле связи разнообразны, потому что они несут разный объем трафика, а также меняются со временем. Важны как структура, так и динамика сетей. Иногда мы просто улавливаем динамику, представляя трафик через направления и веса связей, как обсуждается в главе 4. В других случаях мы, возможно, захотим изучить фактические процессы, которые позволяют сети расти и меняться с течением времени, или взаимодействия, которые происходят в сети. Главы 5 и 7 посвящены этим темам, касающимся сетевой динамики.

0.6. Биологические сети

В клетках нашего организма специальные молекулы, именуемые белками, взаимодействуют различными способами. Например, когда белок сворачивается, его изменение структуры может регулировать функцию другого белка или активность фермента. Ферменты (сами по себе белки) катализируют биохимические реакции и жизненно важны для обмена веществ, который поддерживает жизнь, собирая энергию для строительства и поддержания белков, составляющих наши ткани и органы. Белки также регулируют клеточную сигнализацию и иммунные реакции. Все эти взаимодействия можно рассматривать как сети: сети белковых взаимодействий, метаболические сети, генно-регуляторные сети и т. д. Эти биологические сети существуют внутри клетки. На более высоком уровне, внутри тела, связи между нервными клетками (синапсами) приводят к возникновению нейронных сетей, которые формируют наш мозг. И на еще более высоком уровне взаимодействуют целые виды животных. Животное одного вида может рассматривать другой вид как пищу, создавая экологическую сеть или пищевую (трофическую) паутину между видами. Когда мы думаем об этой сети, экологический баланс зависит от наличия видов, которые поддерживают друг друга. Удаление узла в такой пищевой паутине – например, когда вид вымирает, – влияет на выживание других частей экосистемной сети. На рис. 0.8 показаны три типа биологических сетей: сеть белковых взаимодействий, нейронная сеть и пищевая паутина. Все они являются важнейшими элементами жизни на нашей планете.

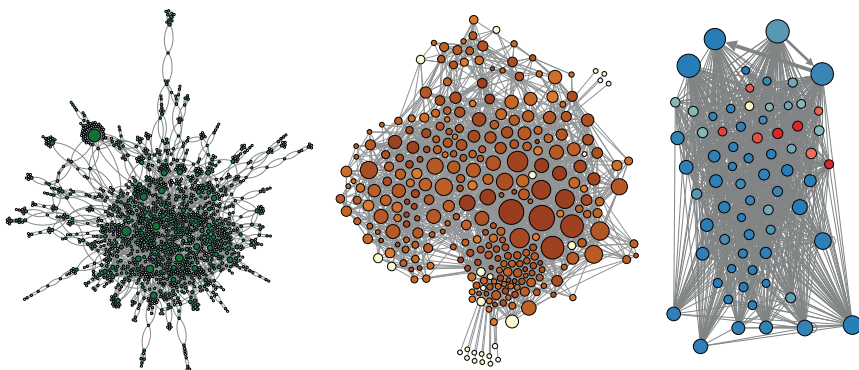


Рис. 0.8 Три биологические сети. Слева: сеть взаимодействия дрожжевых белков. Размер узла пропорционален числу взаимодействующих белков. В центре: нейронная сеть круглого червя *Caenorhabditis elegans*. Крупные и красные узлы представляют нейроны с большим числом соответственно исходящих и входящих синапсов. Справа: пищевая паутина видов в Национальном парке Эверглейдс во Флориде. Направленная связь переходит от жертвы к виду хищника. Вес (ширина) связи представляет собой поток энергии между двумя видами. Размер и цвет узлов представляют соответственно входящие и исходящие связи, вследствие чего крупные синие узлы являются видами в верхней части пищевой цепочки, в то время как малые красные узлы являются видами в нижней части

0.7. Резюме

Сети – это общий способ моделирования и изучения сложных систем со многими взаимодействующими элементами. Мы увидели несколько примеров сетей. Узлы могут представлять самые разные типы объектов, от людей до веб-страниц, от белков до видов животных, от интернет-маршрутизаторов до аэропортов. Узлы могут иметь связанные с ними признаки, помимо меток: географическое местоположение, богатство, активность, число соединений и т. д. Связи также могут представлять много разных видов отношений, от физических до виртуальных, от химических до социальных, от коммуникативных до информационных. Они могут иметь направление (например, веб-связи, так называемые веб-ссылки, и электронная почта)¹ или быть взаимными (например, брак). Все они могут быть одинаковыми или

¹ Обратите внимание, что в науке о сетях все базовые концепции трактуются через призму узлов и связей (а не ссылок) как аналогов терминов «вершины» и «ребра» из теории графов. Отсюда и термин «связь во Всемирной паутине», или «веб-связь» (Web link), и «гиперсвязь» (hyperlink) как связь особого рода, которая связывает страницу с другим ресурсом в паутине. Также следует отметить, что проводится четкое различие между Всемирной паутиной как информационной сетью и интернетом как компьютерной сетью, или сетью маршрутизаторов. – Прим. перев.

иметь разные признаки, такие как сходство, расстояние, трафик, объем, вес и т. д.

0.8. Дальнейшее чтение

Использование сетей для графического представления социальных взаимоотношений между индивидуумами было введено Морено и Дженнингсом (1934), которые назвали эти социальные сети социограммами.

Совсем недавно исследования показали, что онлайн-социальные сети могут выявлять сексуальную ориентацию человека (Джерниган и Мистри, 2009) и способствовать высокоэффективным фишинговым атакам (Джагатик и соавт., 2007). Коновер и соавт. (2011b) показали, что сети диффузии политической информации в Twitter очень поляризованы и сегрегированы. Как следствие, мы можем с высокой точностью предсказывать политические взгляды большинства пользователей, начиная с нескольких меток узлов и распространяя их через соседей по сети (Коновер и соавт., 2011a).

По теме видения, дизайна и истории Всемирной паутины можно прочитать в книге, написанной в соавторстве с его изобретателем (Бернерс-Ли и Фишетти, 2000).

Спринг и соавт. (2002) объясняют принцип применения зондов для измерения топологии интернета. Ахлиоптас и соавт. (2009) показывают, что эти подходы имеют систематическое смещение при взятии выборок. Ученые в области теории вычислительных машин анализируют структуру маршрутизаторов и сетей автономной системы для разработки моделей, именуемых «генераторами топологии», которые будут помогать в дизайне этих сетей (Росси и соавт., 2013). В целях более подробного ознакомления с интернет-сетями мы рекомендуем книгу Пастора-Саторраса и Веспиньяни (2007).

Данные о сети взаимодействий дрожжевых белков взяты из работы Хеонга и соавт. (2001). Данные нейронной сети *C. elegans* взяты из работы Уайта и соавт. (1986). В целях ознакомления с сетями человеческого мозга, или «коннектомом», мы рекомендуем работу Спортса (2012). Экологическая сеть Эверглейдс взята из работы Улановича и Деанджелиса (1998). В целях более подробного ознакомления с пищевыми паутинами мы отсылаем к работам Данна и соавт. (2002) и Мелиана и Баскомпте (2004).

Данные для нескольких примеров реально существующих сетей, показанных в этой книге, предоставлены Сетевым репозиторием (Network Repository, Росси и Ахмед, 2015). Визуализации выполняются с использованием Gephi (Бастиян и соавт., 2009). Алгоритмы компоновки обсуждаются в главе 1.

Упражнения

- 0.1** Рассмотрите дорожную карту на рис. 0.9. Если бы кто-то создавал сетевое представление регулярностей дорожного движения, то какой из следующих ниже вариантов был бы самым лучшим для создания связей сети? (*Подсказка:* ваш ответ на следующий далее вопрос может повлиять на ваш ответ на этот вопрос, и наоборот.)
- Движущиеся по улицам пешеходы.
 - Участки дорог (например, 5-я авеню между 12-й и 13-й улицами).
 - Целые дороги (например, 5-я авеню).
 - Движущиеся по дорогам транспортные средства.

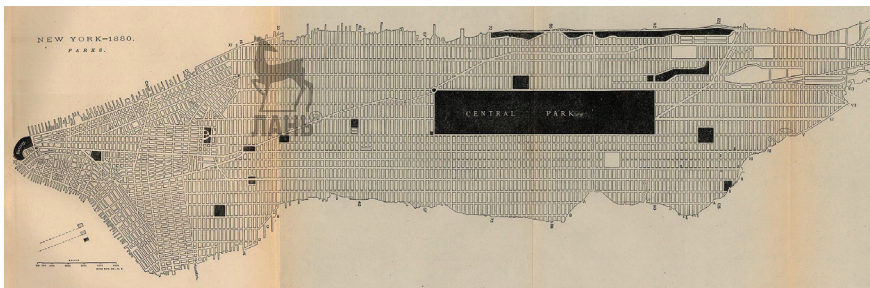


Рис. 0.9 Карта Нью-Йорка в 1880 году. Из Отчета о социальной статистике городов, составленного Джорджем Э. Уорингом-младшим (George E. Waring, Jr.), Бюро переписи населения США, 1886 год. Изображение предоставлено библиотеками Техасского университета

- 0.2** Рассмотрите дорожную карту на рис. 0.9. Какой из следующих ниже вариантов в сетевом представлении шаблонов дорожного движения был бы самым лучшим для создания узлов сети? (*Подсказка:* ваш ответ на предыдущий вопрос может повлиять на ваш ответ на этот вопрос, и наоборот.)
- Городские кварталы (например, квартал между 5–6-й авеню и 12–13-й улицами).
 - Уличные перекрестки (например, 5-я авеню и 12-я улица).
 - Движущиеся по улицам пешеходы.
 - Движущиеся по дорогам транспортные средства.
- 0.3** Рассмотрите сеть авиационных перевозок США, показанную на рис. 0.7. Узлы в этой сети изображают аэропорты. Что могла бы представлять связь между двумя аэропортами?
- 0.4** Сравните сеть авиационных перевозок США на рис. 0.7 с дорожной картой Манхэттена на рис. 0.9. Сеть авиационных перевозок демонстрирует отличительный признак, которого не хватает

у дорожной сети Манхэттена. Какова эта ключевая характеристика?

- a.** Узлы-одиночки без связей.
- b.** Многочисленные маршруты между узлами.
- c.** Узлы с более чем одной присоединенной связью.
- d.** Хабовые узлы со множеством связей.

0.5 Какой тип связи лучше всего отражает взаимоотношение «друг» в социальном графе из Facebook? Направленная или ненаправленная?

0.6 Какой тип связи лучше всего отражает взаимоотношение «подписчик» в социальном графе из Facebook? Направленная или ненаправленная?



https://t.me/it_boooks



Узел: точка в сети или схеме, в которой линии или пути пересекаются или разветвляются.

Рассмотрев несколько примеров реально существующих сетей в главе 0, давайте теперь познакомимся с базовыми определениями и величинами, которые позволяют нам описывать сеть.

1.1. Базовые определения

В очень общих чертах сеть, или граф, представляет собой множество элементов, которые мы называем *узлами*, а также множество соединений между парами узлов, которые мы называем *связями*. Связи обозначают наличие взаимоотношения между элементами, представленными узлами. Как мы видели ранее, связи могут соответствовать социальным, физическим, коммуникационным, географическим, концептуальным, химическим, биологическим или другим взаимодействиям. Мы говорим, что два узла являются *смежными*, *соединены*, или *связаны*, если между ними есть связь. Соединенные узлы также принято называть *соседями*.

Сети обеспечивают общий теоретический каркас, допускающий удобное концептуальное представление взаимоотношений в широком спектре систем; в главе 0 мы увидели несколько примеров таких систем. Изучение сетей имеет давние традиции в математике, информатике, социологии и исследованиях в области коммуникаций. В последнее время сети также интенсивно изучаются в физике и биологии. Разные области, которые имеют отношение к сетям, нередко вводят свою собственную номенклатуру. Например, в некоторых областях сеть называется *графом*, узел называется *вершиной*, а связь – *ребром*. (Время от времени мы будем использовать эти термины.) Строгий язык описания сетей можно найти в теории графов, области математики, которая восходит к новаторской работе Леонарда Эйлера в XVIII веке. Здесь мы не хотим давать строгого введения в теорию графов. Мы в основном заинтересованы в построении словаря и введении набора базовых понятий, которые позволят нам сделать первые шаги в мир сетей. Однако иногда полезно использовать формальную нотацию. В этих случаях мы будем включать формальную нотацию в заштрихованную область, или во вставку, обрамленную

рамкой. Например, более строгое определение сети приведено во вставке 1.1. В последующих главах мы будем вводить дополнительные понятия и определения, необходимые для анализа реально существующих систем.

Вставка 1.1

Определение сети

Сеть G состоит из двух частей, множества из N элементов, именуемых *узлами*, или *вершинами*, и множества из L пар узлов, именуемых *связями*, или *ребрами*. Связь (i, j) соединяет узлы i и j . Сеть может быть направленной или ненаправленной¹. Направленная сеть также называется *орграфом* от термина «ориентированный граф». В направленных сетях связи называются *направленными связями*, и порядок узлов в связи отражает направление: связь (i, j) идет из источникового узла i в целевой узел j . В ненаправленных сетях все связи являются двунаправленными, и порядок расположения двух узлов в связи не имеет значения. Сеть может быть невзвешенной или взвешенной. Во взвешенной сети со связями ассоциированы *веса*: *взвешенная связь* (i, j, w) между узлами i и j имеет вес w . Сеть может быть как направленной, так и взвешенной, и в этом случае она имеет направленные взвешенные связи.

Каждая сеть характеризуется суммарным числом узлов N и суммарным числом связей L . Мы называем N *размером* сети, потому что оно определяет число отдельных элементов, составляющих систему. Чисел узлов и связей недостаточно для определения сети; мы должны указать способ, которым узлы соединяются связями.

Существуют разные типы связей, которые определяют разные классы сетей. В некоторых сетях, таких как Facebook (рис. 0.1), связи не имеют направления, и мы представляем их в виде отрезков. Мы называем такие сети *ненаправленными*. В других случаях, таких как «Википедия» (рис. 0.5), связи направлены, и мы представляем их в виде стрелок. Сети с направленными связями называются *направленными сетями*. Мы расскажем о направленных сетях подробнее в разделе 1.6 и главе 4.

В некоторых случаях, таких как сети авиационных перевозок (рис. 0.7), связи имеют соответствующие веса. Они называются *взвешенными сетями*. Сеть может быть как направленной, так и взвешенной. Сеть электронной почты является примером взвешенной направленной сети, в которой веса и направления связей представляют трафик связи (число сообщений) между узлами. Мы вернемся к взвешенным сетям в разделе 1.7 и главе 4. На рис. 1.1 представлены иллюстрации ненаправленных, направленных и взвешенных сетей.

¹ Понятия направленности и ориентированности связей (и сетей в целом) являются синонимичными. В переводе принят первый вариант. Понятие ориентированности чаще используется в теории графов. – Прим. перев.

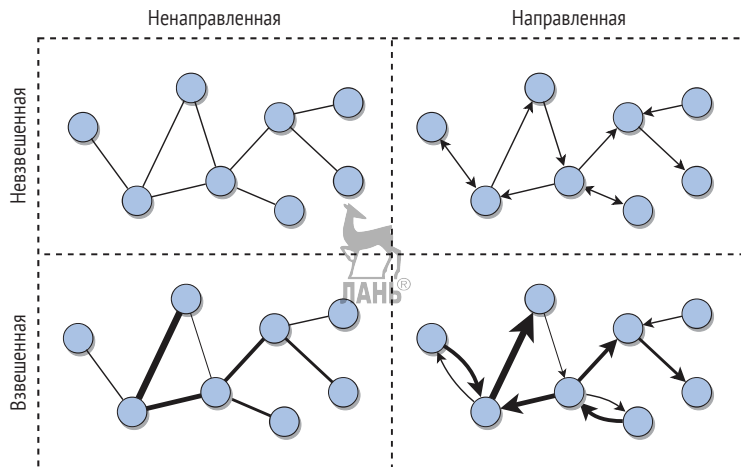


Рис. 1.1 Графические представления ненаправленных, направленных и взвешенных сетей. Круги представляют узлы. Пары смежных узлов соединяются отрезком (связью) или стрелкой (направленной связью). Стрелки указывают направление связей. Толщина связи представляет ее вес во взвешенных сетях

Существует несколько других классов сетей. В сети может быть несколько типов узлов. Например, сеть кинозвезд (рис. 0.2(a)) имеет два типа узлов, представляющих кинофильмы и людей. В этой сети связь соединяет актера или актрису с кинофильмом, но нет никаких связей между людьми или между кинофильмами. Это пример так называемой *двудольной сети*. В двудольной сети есть две группы узлов, таких, при которых связи соединяют узлы только из разных групп, а не узлы из одной и той же группы. Другие примеры двудольных сетей включают сети, которые улавливают взаимоотношения между песнями и исполнителями, между учебными занятиями и студентами, а также между товарами и покупателями. Подробнее о двудольных сетях вы узнаете в главе 4.

Сеть может иметь несколько типов связей, и в этом случае она называется *мультиплексной сетью*. Еще раз используя пример с кинозвездами, мы могли бы вообразить добавление связей между актерами и/или актрисами, которые находятся в супружеской связи друг с другом. В примере с «Википедией» (рис. 0.5) в дополнение к гиперсвязям у нас могут быть взвешенные связи, представляющие клики пользователей «Википедии», и/или ненаправленные связи между статьями, которые имеют общих редакторов. Эти и другие более сложные типы сетей обсуждаются далее в разделе 1.8.

1.2. Манипулирование сетями в исходном коде

Для управления, анализа и визуализации сетей с более чем несколькими узлами и связями нам необходимо использовать программно-

информационные инструменты или писать собственный исходный код. Существует масса инструментов сетевого анализа и визуализации, а также библиотек для работы с сетями на многих языках программирования. На протяжении всей книги мы время от времени будем упоминать пару таких инструментов. Например, визуализации в главе 0 генерируются с помощью приложения под названием *Gephi*. Однако мы считаем, что для практического понимания сетей необходимо «запачкать свои руки» выполнением черновой работы и написать немного исходного кода. Мы исходим из того, что студенты, использующие эту книгу, имеют некоторое представление о Python, популярном языке программирования как среди начинающих, так и среди опытных программистов¹. Чтобы облегчить жизнь, мы будем использовать *NetworkX* (networkx.github.io), пакет Python для создания, управления и изучения структуры, динамики и функций сетей. *NetworkX* предоставляет структуры данных, алгоритмы, меры и генераторы для сетей, а также рудиментарные средства визуализации².

После импортирования библиотеки *NetworkX* мы можем легко создать ненаправленную сеть («Граф») и добавить несколько узлов и связей. Обращение к узлам осуществляется посредством целочисленных идентификаторов, а связи называются ребрами (*edge*):

```
import networkx as nx # всегда сначала следует импортировать NetworkX!
G = nx.Graph()
G.add_node(1)
G.add_node(2)
G.add_edge(1,2)
```

Мы можем добавить несколько узлов или связей одновременно:

```
G.add_nodes_from([3,4,5,...])
G.add_edges_from([(3,4),(3,5),...])
```

Вот как мы получаем списки узлов, связей и соседей данного узла:

```
G.nodes()
G.edges()
G.neighbors(3)
```



¹ Мы предлагаем вводное учебное пособие по Python в приложении А; его также можно скачать из репозитория книги на GitHub по адресу github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

² Мы предлагаем вводное учебное пособие по библиотеке *NetworkX* в репозитории книги на GitHub.

И вот как выполняется перебор узлов или связей в цикле:

```
for n in G.nodes:
    print(n, G.neighbors(n))
for u,v in G.edges:
    print(u, v)
```

Мы можем создать направленную сеть («Орграф») схожим образом:

```
D = nx.DiGraph()
D.add_edge(1,2)
D.add_edge(2,1)
D.add_edges_from([(2,3),(3,4),...])
```



Обратите внимание, что связь между узлом 1 и узлом 2 отличается от связи между узлом 2 и 1, поскольку эта сеть является направленной. Также обратите внимание, что, когда мы добавляем связь, узлы добавляются автоматически, если они еще не существуют. Это очень удобно. Существуют функции для получения размера и числа связей:

```
D.number_of_nodes()
D.number_of_edges()
```

Когда мы запрашиваем соседей узла в направленной сети, мы получаем узлы, соединенные с этим узлом непосредственно входящими и исходящими связями. Помимо этого, есть также функции для получения только тех ребер, которые соответственно ведут к этому узлу либо из этого узла, именуемые предшественниками и приемниками:

```
D.neighbors(2)
D.predecessors(2)
D.successors(2)
```

Наконец, существуют функции для генерирования сетей многих типов. Обычно этим функциям требуются аргументы, задающие число узлов или связей. Ниже приведен исходный код для генерирования нескольких сетей, показанных на рис. 1.2:

```
B = nx.complete_bipartite_graph(4,5)
C = nx.cycle_graph(4)
P = nx.path_graph(5)
S = nx.star_graph(6)
```

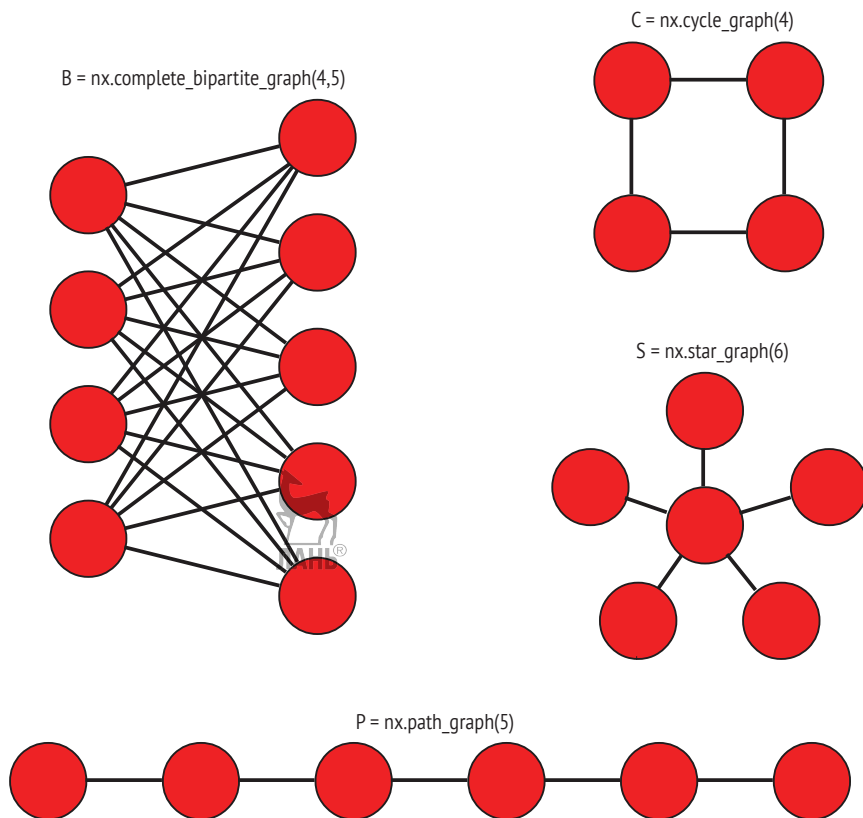


Рис. 1.2 Несколько простых сетей, сгенерированных функциями библиотеки NetworkX: полная двудольная (B), циклическая (C), звезда (S) и путь (P). Понятие *полной* сети представлено в следующем далее разделе

Мы настоятельно рекомендуем вам прочитать учебное руководство по networkx¹ и добавить в закладки его документацию². И помните, что Google и StackOverflow – это ваши друзья в ситуациях, когда вы застреваете!

1.3. Плотность и разреженность

Максимальное число связей в сети ограничено возможным числом отличимых соединений между узлами системы. Следовательно, максимальное число связей определяется числом пар узлов. Сеть с максимальным числом связей, в которой все возможные пары узлов соединены связями, называется *полной сетью*.

¹ См. networkx.github.io/documentation/stable/tutorial.html.

² См. networkx.github.io/documentation/stable/.

Максимальное число связей в ненаправленной сети с N узлами – это число отличимых пар узлов:

$$L_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}. \quad (1.1)$$

Интуитивно каждый узел может соединяться с $N - 1$ другими узлами, и их всего N . Однако это означало бы засчитывать каждую пару дважды, поэтому мы делим на два. В направленной сети каждая пара узлов должна засчитываться дважды, по одному разу для каждого направления, поэтому $L_{\max} = N(N - 1)$. Подсчет возможных пар объектов среди множества из N объектов – это то, с чем мы снова столкнемся в книге чуть позже. У математиков есть название для формулы $\binom{N}{2}$: «из N по два».

Двудольная сеть является *полной*, если каждый узел в одной группе соединен со всеми узлами в другой группе (см. пример В на рис. 1.2). В данном случае $L_{\max} = N_1 \times N_2$, где N_1 и N_2 – это размеры двух групп.

Доля возможных связей, которые существуют фактически, одинаковая с долей пар узлов, которые соединены фактически, называется *плотностью* сети. Полная сеть имеет максимальную плотность, равную единице. Однако фактическое число связей обычно намного меньше максимального, так как большинство пар узлов напрямую друг с другом не соединены. Следовательно, плотность часто намного меньше единицы – на порядки в большинстве реально существующих крупных сетей. Этот признак является важным и помогает в работе со структурой сети. Мы называем его *разреженностью*. Интуитивно чем меньше ребер в сети, тем она разреженнее.

Плотность сети с N узлами и L связями равна:

$$d = L/L_{\max}. \quad (1.2)$$

В ненаправленной сети она задается уравнением

$$d = L/L_{\max} = \frac{L}{N(N-1)}. \quad (1.3)$$

а в направленной сети плотность равна:

$$d = L/L_{\max} = \frac{2L}{N(N-1)}. \quad (1.4)$$

В полной сети, $d = 1$ по определению, поскольку $L = L_{\max}$. В разреженной сети $L \ll L_{\max}$ и, следовательно, $d \ll 1$. Когда сеть становится очень крупной, мы можем наблюдать, как число связей увеличивается как функция от числа узлов. Мы говорим, что сеть разреженная, если число связей растет пропорционально числу узлов ($L \sim N$) или даже медленнее. Если вместо этого число связей растет быстрее, например квадратично вместе с размером сети ($L \sim N^2$), тогда мы говорим, что сеть – плотная.

В качестве иллюстрации важности разреженности сети давайте рассмотрим пример с Facebook. На момент написания этой книги у Facebook было около 2 млрд пользователей ($N \approx 2 \times 10^9$). Если бы эта сеть была полной, то имелось бы $L \approx 10^{18}$ связей – это число с 18 нулями, и нет никакого способа хранить столь много данных! Но, к счастью, социальные сети очень разреженные, и Facebook не является исключением. Каждый пользователь имеет в среднем 1000 друзей или меньше того, вследствие чего плотность приблизительно равна $d \approx 10^{-6}$. Это все же много данных, но Facebook способна ими управлять.

В табл. 1.1 представлены базовые статистические величины, касающиеся размера и плотности сети, примеры которых приведены в главе 0¹. Хотя эти сети сильно отличаются друг от друга, все они являются разреженными.

Таблица 1.1. Базовые статистические величины примеров сетей. Типы сетей могут быть направленными (D) и/или взвешенными (W). Когда метки нет, сеть является ненаправленной и невзвешенной. Для направленных сетей мы показываем среднюю степень-на-входе (которая совпадает со средней степенью-на-выходе)

Сеть	Тип	Узлы (N)	Связи (L)	Плотность	Средняя степень ((k))
Facebook, Северо-Западный университет		10567	488 337	0.009	92.4
IMDB, кинофильмы и кинозвезды		563443	921 160	0.000006	3.3
IMDB, кинозвезды, снимавшиеся вместе	W	252999	1 015 187	0.00003	8.0
Twitter, политика США	DW	18470	48 365	0.0001	2.6
Электронная почта компании Энрон	DW	87273	321 918	0.00004	3.7
Статьи по математике в «Википедии»	D	15220	194 103	0.0008	12.8
Интернет-маршрутизаторы		190914	607 610	0.00003	6.4
Авиационные перевозки в США		546	2781	0.02	10.2
Авиационные перевозки по всему миру		3179	18 617	0.004	11.7
Взаимодействие дрожжевых белков		1870	2277	0.001	2.4
Мозг C. elegans	DW	297	2345	0.03	7.9
Экологическая пищевая паутина Everglades	DW	69	916	0.2	13.3

Библиотека NetworkX позволяет легко измерять плотность направленных и ненаправленных сетей:

¹ Наборы данных для этих сетей доступны в репозитории книги на GitHub: github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

```

nx.density(G)
nx.density(D)
CG = nx.complete_graph(8471) # крупная полная сеть
print(nx.density(CG))        # калькулятор не нужен!

```

1.4. Подсети

Во многих случаях нас интересует подмножество сети, которое само по себе является сетью и называется *подсетью* (или *подграфом*). Подсеть получается путем отбора подмножества узлов и *всех* связей между этими узлами.

На рис. 1.3 представлено несколько иллюстраций подсетей ненаправленных и направленных сетей. Обилие определенных типов подсетей и их свойств имеет важное значение для характеристики реально существующих сетей. В качестве примера клика представляет собой полную подсеть: подмножество узлов, связанных друг с другом. Любая подсеть полной сети является кликой, потому что все пары узлов в сети соединены и, следовательно, все пары узлов в любой подсети соединены тоже.

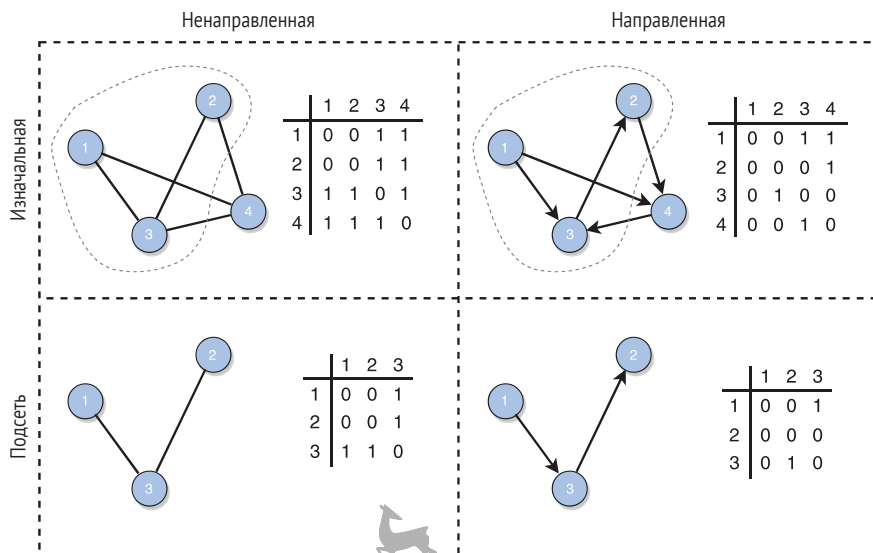


Рис. 1.3 Примеры сетей и подсетей. Мы также показываем представление каждой сети в форме матрицы смежности (см. раздел 1.9)

Особым типом подсети является *эгосеть* узла, которая представляет собой подсеть, состоящую из выбранного узла – именуемого *эго*, – и его соседей. Эгосети часто изучаются в анализе социальных сетей.

Используя библиотеку NetworkX, мы можем сгенерировать подсеть данной сети, указав подмножество узлов:

```
K5 = nx.complete_graph(5)
clique = nx.subgraph(K5, (0,1,2))
```

1.5. Степень

Степень узла – это число его связей или соседей. Мы обозначаем степень узла i через k_i . Рис. 1.4 иллюстрирует степень нескольких узлов в ненаправленной сети. Узел без соседей, например узел а на данном рисунке, имеет нулевую степень ($k = 0$) и называется *узлом-одиночкой*, или *синглетоном*.

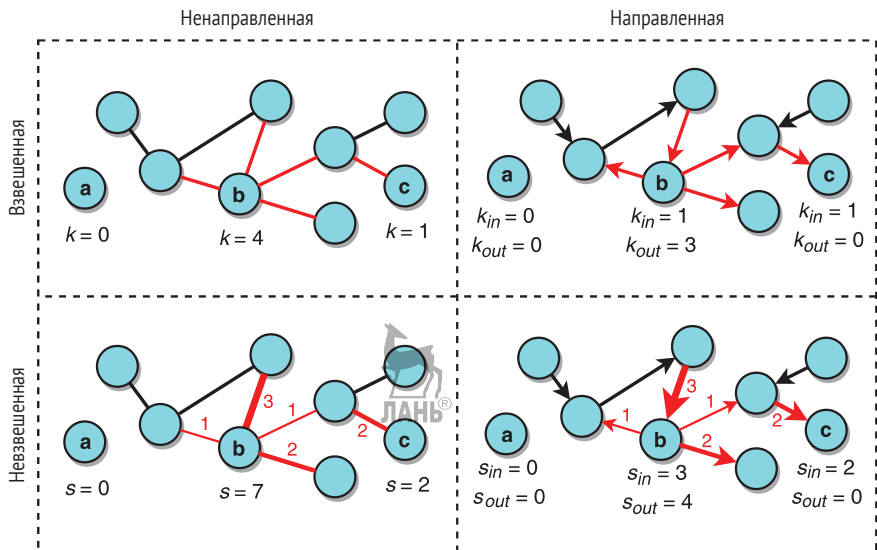


Рис. 1.4 Иллюстрации степени и силы в направленных, ненаправленных, взвешенных и невзвешенных сетях. Связи узлов **a**, **b** и **c** вместе с их весами выделены красным цветом, и показаны их степени, или силы

Средняя степень сети обозначается через $\langle k \rangle$. Это важное свойство и (прямо пропорционально) относится к ее плотности.

Средняя степень сети определяется уравнением

$$\langle k \rangle = \frac{\sum_i k_i}{N}. \quad (1.5)$$

Поскольку каждая связь вносит свой вклад в степень двух узлов в ненаправленной сети, числитель уравнения (1.5) может записываться как $2L$. Из определения плотности для ненаправленной сети [уравнение (1.3)] $2L = dN(N - 1)$. Следовательно,

$$\langle k \rangle = \frac{2L}{N} = \frac{dN(N - 1)}{N} = d(N - 1), \quad (1.6)$$

и наоборот



$$d = \frac{\langle k \rangle}{N - 1}. \quad (1.7)$$

Это имеет смысл: максимально возможная степень узла равна $k_{\max} = N - 1$ и получается, когда узел соединен с каждым другим узлом. Интуитивно плотность – это соотношение между средней и максимальной степенью.

В табл. 1.1 показана средняя степень примеров сетей, показанных в главе 0. В библиотеке NetworkX есть функция, которая возвращает степень данного узла. Без аргументов она возвращает словарь со степенью каждого узла:

<code>G.degree(2)</code>	# возвращает степень узла 2
<code>G.degree()</code>	# возвращает степень всех узлов сети G

В главе 3 мы увидим, что степени отдельных узлов сети являются очень важными свойствами для характеристики структуры сети. До сих пор мы определяли степень в ненаправленных сетях. Далее мы распространим определение на направленные и взвешенные сети.

1.6. Направленные сети

В графическом представлении сети направленная природа связей изображается посредством стрелки, указывающей направление каждой связи. Главное различие между направленными и ненаправленными сетями представлено на рис. 1.1. В ненаправленной сети наличие связи между двумя узлами соединяет смежные узлы в обоих направлениях. С другой стороны, из наличия связи в направленной сети не обязательно вытекает наличие связи в противоположном направлении. Этот факт имеет важные последствия для связности (соединенности) направленной сети, как будет подробнее рассмотрено в главе 2.

Когда мы рассматриваем степень узла в направленной сети, мы должны думать о входящих и исходящих связях отдельно. Число входящих связей, или предшественников, узла i называется *степенью-на-входе* и обозначается через k_i^{int} . Число исходящих связей, или преемников, узла i называется *степенью-на-выходе* и обозначается через k_i^{out} . На рис. 1.4 показаны степень-на-входе и степень-на-выходе нескольких узлов в направленной сети.

Мы уже определили плотность для направленной сети (уравнение (1.4)). Мы можем определить среднюю степень-на-входе и среднюю степень-на-выходе аналогично уравнению (1.5).

В библиотеке NetworkX есть функции, которые возвращают степень-на-входе и степень-на-выходе данного узла. Если сеть является направленной, то функция `degree` возвращает суммарную степень, которая представляет собой сумму степени-на-входе и на-выходе:

```
D.in_degree(4)
D.out_degree(4)
D.degree(4)
```

1.7. Взвешенные сети

В графическом представлении сети взвешенная природа связей изображается посредством отрезков разной ширины, обозначающих вес каждой связи. Нулевой вес эквивалентен отсутствию связи. Главное различие между взвешенными и невзвешенными сетями представлено на рис. 1.1.

Взвешенная сеть может быть направленной или ненаправленной; давайте сначала рассмотрим более простой случай ненаправленной взвешенной сети. Мы можем измерить степень узла в взвешенной сети, игнорируя веса. Тем не менее бывает важно учитывать веса. Поэтому мы можем определить *взвешенную степень*, или *силу* узла, как сумму весов его связей. Схожим образом мы можем определить *силу-на-входе* и *силу-на-выходе* для случая направленной взвешенной сети. Оба случая показаны на рис. 1.4.

Взвешенная степень, или *сила*, узла i в ненаправленной взвешенной сети обозначается через

$$s_i = \sum_j w_{ij}, \quad (1.8)$$



где w_{ij} – это вес связи между узлами i и j . Мы исходим из допущения, что $w_{ij} = 0$, если между i и j нет связи. Степень-на-входе и степень-на-выходе можно обобщить на силу-на-входе и на силу-на-выходе аналогичным образом в направленной взвешенной сети:

$$s_i^{in} = \sum_j w_{ji}, \quad (1.9)$$

$$s_i^{out} = \sum_j w_{ij}, \quad (1.10)$$

где w_{ij} – это вес направленной связи из i в j .

В библиотеке NetworkX как к графам, так и к орграфам могут быть прикреплены атрибуты «веса». При добавлении нескольких взвешенных связей каждая указывается как триплет, где третьим элементом является вес:

```
W = nx.Graph()
W.add_edge(1,2,weight=6)
W.add_weighted_edges_from([(2,3,3),(2,4,5)])
```

Мы можем получить список связей с ассоциированными весовыми данными, например, если нам нужно распечатать связи с крупным весом:

```
for (i,j,w) in W.edges(data='weight'):
    if w > 3:
        print('%d, %d, %d)' % (i,j,w)) # пропустить связь (2,3)
```

Наконец, мы можем получить силу данного узла, используя функцию `degree` и указав весовой атрибут:

```
W.degree(2, weight='weight') # сила узла 2
# равна 6 + 3 + 5 = 14
```

1.8. Многослойные и темпоральные сети

В показанной на рис. 0.7 сети авиационных перевозок в США связи представляют прямые рейсы между аэропортами независимо от того, какие конкретно авиалинии выполняют эти рейсы. Но классифици-

рование рейсов в зависимости от соответствующих им авиалиний полезно в ряде ситуаций. Мы, возможно, захотим предсказывать распространение задержек в расписании по сети авиалинии или исследовать последствия таких задержек для передвижения пассажиров. На самом деле каждая коммерческая авиалиния пытается сначала перепланировать пассажиров на свои собственные рейсы, потому что перебронировать их на рейсы другой компании дорого. Поэтому сеть авиационных перевозок конкретной авиалинии имеет свою собственную идентичность, даже несмотря на то, что она переплетена с сетями других авиалиний. В этих случаях выгодно представлять систему в виде *многослойной сети* (т. е. комбинации слоев), где каждый слой представляет сеть авиационных перевозок конкретной авиалинии: узлами являются аэропорты, связями – рейсы, выполняемые одной и той же компанией.

Если каждый слой в многослойной сети строится на одном и том же множестве узлов, то такая сеть называется *мультиплексной*. Сеть авиационных перевозок является примером мультиплекса. Еще одним примером является социальная сеть, в которой разные слои представляют разные типы социальных отношений. Например, один слой может представлять дружеские привязанности, другой слой – семейные узы, еще один – связи с коллегами и т. д. Узлы в каждом слое представляют одинаковых индивидумов.

Темпоральная сеть – это частный случай мультиплекса. Связи являются динамическими, поскольку соответствующие взаимодействия между узлами происходят в разное время. Узлы тоже могут иметь динамический характер, поскольку они могут появляться и исчезать на разных стадиях эволюции сети. Например, сети пользовательской активности в Twitter являются темпоральными, потому что публикации, ретвиты и упоминания происходят в разное время, что можно определить по их меткам времени. Мы можем разделить временной промежуток темпоральной сети на поочередные интервалы: все узлы и связи, существующие в течение каждого интервала, составляют моментальный *снимок* системы. Каждый снимок может интерпретироваться как один слой мультиплекса, как показано на рис. 1.5.

В многослойной сети существуют *внутрислойные связи*, соединяющие пары узлов в одном слое, и *межслойные связи*, соединяющие пары узлов в разных слоях. В частном случае мультиплексных сетей межслойные связи соединяют каждый узел слоя с его противоположностью в других слоях. Такие связи называются *стыками*, потому что они состыковывают копии одного и того же узла в разных слоях.

Традиционно мультиплексные сети анализировались путем агрегирования данных из разных слоев и последующего изучения результирующей сети. Например, сети на рис. 0.3 и 0.7 представляют агрегации мультиплексных сетей, соответствующих временным интервалам и разным авиалиниям. Агрегированная сеть обычно взвешена, даже если связи мультиплекса не являются таковыми, потому

что обычно одну и ту же пару узлов в разных слоях соединяет несколько связей, которые превращаются в единую взвешенную связь в агрегированной системе. Например, связи на рис. 0.3 взвешены по числу повторных твитов одного пользователя другим. Но агрегация отбрасывает много ценной информации, предоставляемой изначальной многослойной системой. В случае авиационных перевозок слияние сетей, соответствующих разным авиакомпаниям, не позволяет нам изучать переходы пассажиров между такими сетями, что может потребоваться в случае забастовок или технических проблем, затрагивающих конкретную авиалинию.

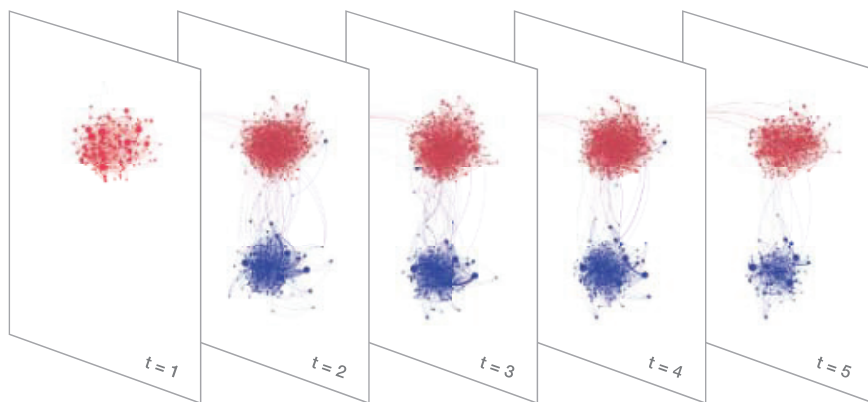


Рис. 1.5 Темпоральная сеть политических ретвитов. Каждый моментальный снимок содержит связи с ретвитами с отметками времени в определенном интервале времени. Агрегируя эти снимки во времени, мы получаем статическую сеть, показанную на рис. 0.3

В общем случае каждый слой может характеризоваться своим собственным множеством узлов и связей. Поэтому слои могут представлять совершенно разные графы, и в результате система представляет собой *сеть сетей*. Здесь межслойные связи могут представлять отношения зависимости между узлами сетей. Давайте рассмотрим электроэнергетическую сеть, которая соединяет электростанции и центры спроса через высоковольтные линии электропередачи. Станции управляются компьютерами, которые проводят мониторинг и управляют производством и передачей электроэнергии. Эти компьютеры соединены через интернет. В свою очередь интернет-маршрутизаторы зависят от электростанций в части их электроснабжения. Поэтому у нас есть система с двумя состыкованными сетями: электросетью и интернетом.

В такой состыкованной системе одна сеть может влиять на другую в целях оптимизирования доставки; по мере надобности сеть может переконфигурироваться для перенаправления электроэнергии. Однако такого рода сеть сетей также может приносить непредсказуемые уязвимости. Проблема с программно-информационным обеспечением или атака могут вывести из строя один или несколько узлов в элект-

росети, а интернет без электричества в регионе тоже может выйти из строя, что приведет к отказам в работе других узлов и в крайнем случае – катастрофическому эффекту домино, именуемому *каскадирующим сбоем*, затрагивающим крупную часть сети. По этим причинам сети сетей являются предметом интенсивного изучения.

Для упрощения в этой книге мы сосредоточимся в основном на сетях с одним типом узла и одним типом связи. В ненаправленной сети мы будем исходить из допущения, что пара узлов соединяется не более одной связью. (Если сеть является направленной, то могут существовать две связи, по одной в каждом направлении, как показано на рис. 1.1.) В дополнение к этому мы не будем рассматривать *само-направленные циклы*, или связи, соединяющие узел с самим собой; мы будем считать, что каждая связь соединяет два отдельных узла.



1.9. Представления сетей

В целях сохранения сети в компьютерном файле или памяти и ее последующего извлечения оттуда нам нужен способ формального представления ее узлов и связей. Существует несколько возможных представлений сетей. Самое простое – это *матрица смежности*, матрица $N \times N$, в которой каждый элемент представляет связь между узлами, индексируемыми соответствующей строкой и столбцом.

Элемент a_{ij} матрицы смежности представляет связь между узлами i и j . $a_{ij} = 1$, если i и j являются смежными, $a_{ij} = 0$ в противном случае.

На рис. 1.3 мы показываем графические иллюстрации разных ненаправленных и направленных сетей и соответствующих им матриц смежности.

В случае ненаправленных сетей матрица смежности симметрична: мы можем менять местами строки и столбцы, и матрица не меняется. Следовательно, половина матрицы содержит избыточную информацию. В случае направленных сетей матрица смежности не является симметричной. В случае невзвешенных сетей элементы принимают только значения один или ноль, чтобы обозначать соответственно наличие или отсутствие связи. В случае взвешенных сетей матричные элементы могут принимать любые значения, соответствующие весам связей. Мы уже встречались с элементами матрицы смежности для взвешенных сетей (w_{ij} в уравнениях (1.8)–(1.10)).

В библиотеке NetworkX можно получать и распечатывать матрицы смежности и использовать матричное представление для получения и задания атрибутов связи:


```

print(nx.adjacency_matrix(G)) # граф
G.edge[3][4]
G.edge[3][4]['color']='blue'
print(nx.adjacency_matrix(D)) # орграф
D.edge[3][4]
D.edge[4][3]
print(nx.adjacency_matrix(W)) # взвешенный граф
W.edge[2][3]
W.edge[2][3]['weight'] = 2

```

Хотя представление в форме матрицы смежности соответствует математическому формализму сетей, оно неэффективно для хранения реально существующих сетей, которые обычно имеют крупные размеры и разрежены. Требуемое пространство для хранения растет как квадрат размера сети (N^2), но если сеть разрежена, то большая часть этого пространства тратится впустую на хранение нулей (несуществующих связей). В крупных разреженных сетях более компактным представлением сети является *список смежности*, структура данных, в которой хранится список соседей по каждому узлу. Списки смежности эффективно представляют разреженные сети, поскольку игнорируются несуществующие связи; рассматриваются только существующие связи (ненулевые значения матрицы смежности).

Библиотека NetworkX предлагает средства для перебора сетевого списка смежности в цикле и извлечения связей и их атрибутов. Например, вот один из способов распечатать соседей каждого узла:

```

for n,neighbors in G.adjacency():
    for number,link_attributes in neighbors.items():
        print('%d, %d' % (n,number))

```

Третье, не менее эффективное представление сети – это *список ребер*, в котором каждая связь представлена в виде пары соединенных узлов. Нам также может потребоваться перечислить узлы отдельно в случае узлов-одиночек, которые не будут появляться ни в одной из пар. В случае взвешенных сетей каждая связь представляется в виде тройки, где третьим элементом является вес.

В этой книге для хранения сетей мы будем использовать представление в форме списка ребер. Библиотека NetworkX имеет функции для записи и чтения файлов сетей с использованием этого представления. Вы можете просмотреть формат файла списка ребер самостоятельно:

```

nx.write_edgelist(G, "file.edges")
G2 = nx.read_edgelist("file.edges") # G2 такой же, что и G
nx.write_weighted_edgelist(W, "wf.edges") # сохранить веса
with open("wf.edges") as f:

```

```
for line in f:
    print(line)
W2 = nx.read_weighted_edgelist("wf.edges") # W2 такой же, что и W
```

1.10. Рисование сетей

О сети можно узнавать многое, рисуя и изучая ее графическое представление. Для этого требуется *алгоритм компоновки сети*, чтобы размещать каждый узел на плоскости. (Есть также сложные трехмерные компоновки, но в этой книге мы не обсуждаем их.) Существует целый ряд алгоритмов компоновки, служащих для представления разных типов сетей; например, для рисования сети авиационных перевозок на рис. 0.7 мы использовали *географическую компоновку*. Для относительно малых сетей компоновки размещающие узлы вдоль концентрических кругов или слоев могут выявлять важную иерархическую структуру. Наиболее популярным классом алгоритмов компоновки сети являются *алгоритмы компоновки по направлению силы* (*force-directed layout algorithm*), которые используются для визуализации большинства примеров сетей в главе 0. Во вставке на рис. 0.7 тоже используется компоновка по направлению силы.

Цели алгоритма компоновки по направлению силы заключаются в размещении узлов в таком ключе, чтобы соединенные узлы располагались близко друг к другу, все связи имели одинаковую длину, а число пересечений связей минимизировалось. В целях получения представления о принципе работы компоновки по направлению силы вообразите силу, которая отталкивает любые два узла друг от друга, подобно силе между двумя частями с одинаковым электрическим зарядом. Далее вообразите пружину, соединяющую любые два связанных узла, создающую силу притяжения, когда они находятся слишком далеко друг от друга. Алгоритмы компоновки по направлению силы симулируют такую физическую систему, вследствие чего узлы движутся, минимизируя энергию системы: соединенные узлы будут двигаться навстречу друг другу и удаляться от узлов, не соединенных с ними.

Результатом является не только эстетически приятный рисунок, но и – иногда – визуализация наиболее очевидных сообществ в сети, как мы видели в главе 0. Например, поскольку на рис. 0.3 люди в сообществе (прогрессивном или консервативном) тесно соединены друг с другом, они в итоге группируются в компоновке вместе.

Библиотека NetworkX имеет функцию рисования сети, в которой используется элементарный алгоритм компоновки сети:

```
import matplotlib.pyplot
nx.draw(G)
```

Обратите внимание, что для рисования требуется интерфейс графопостроения, такой как Matplotlib. Он достаточно хорошо работает для малых сетей, имеющих, к примеру, менее 100 узлов. Для более крупных сетей существуют более совершенные инструменты визуализации. Примеры в главе 0 визуализированы с помощью алгоритма компоновки *ForceAtlas2* инструмента визуализации Gephi.

1.11. Резюме

Мы представили несколько базовых определений и величин, которые позволяют нам описывать сеть.

1. Сеть состоит из двух множеств элементов: узлов и связей, соединяющих пары узлов.
2. Подсеть – это подмножество сети, включающее несколько ее узлов и все связи между ними.
3. В направленных сетях связи имеют направление. Может существовать связь от узла 1 к узлу 2 и не обязательно из узла 2 в узел 1. В ненаправленных сетях связи являются взаимными.
4. Во взвешенных сетях связи имеют ассоциированные веса, которые представляют атрибуты связи, такие как важность, сходство, расстояние, трафик и т. д. В невзвешенных сетях все связи одинаковы.
5. Многослойные сети имеют разные типы узлов и связей, поделенных на взаимосвязанные слои. Если в каждом слое узлы одинаковы, то такая многослойная сеть называется мультиплексной.
6. Плотность сети – это доля пар соединенных узлов. Сеть является полной, если все пары узлов соединены, вследствие чего плотность равна единице. Большинство реально существующих сетей являются разреженными, а значит они имеют очень малую плотность.
7. Степень узла – это число соседей. В направленных сетях узлы имеют степень-на-входе (in-degree) и степень-на-выходе (out-degree), измеряющие соответственно число входящих и исходящих связей. Если сеть является взвешенной, то сила узла равна сумме весов ее связей. Узлы взвешенных направленных сетей имеют силу-на-входе (in-strength) и силу-на-выходе (out-strength).
8. Списки смежности и списки ребер являются эффективными представлениями, служащими для хранения разреженных сетей.
9. NetworkX – это популярная и удобная программная библиотека для программирования сетей на языке Python.

Определения в этой главе образуют базовый словарь науки о сетях. В следующих главах будут представлены дополнительные величины и свойства, чтобы иметь возможность описывать, анализировать и моделировать реально существующие сети и узнавать то, что они говорят нам о базовых системах и явлениях.



1.12. Дальнейшее чтение

Есть несколько других отличных учебников по науке о сетях, которые выходят за рамки вводного материала этой книги. Калдарелли и Чесса (2016) окунаются чуть-чуть глубже в науку о данных нескольких тематических исследований. Если вы заинтересованы завернуть в сторону физики, то подумайте об учебнике Барабаши (2016); если вы хотите разведать связи с экономикой и социологией, то мы рекомендуем учебник Исли и Клейнберга (2010). Более продвинутым темам по физике, математике и социальным наукам посвящено много книг на выбор (Вассерман и Фауст, 1994; Кальдарелли, 2007; Баррат и соавт., 2008; Коэн и Хавлин, 2010; Боллобас, 2012; Дороговцев и Мендес, 2013; Латори и соавт., 2017; Ньюман, 2018).

Кивеля и соавт. (2014) и Боккалетти и соавт. (2014) представили влиятельные обзоры многослойных сетей. Обзор темпоральных сетей представлен Холме и Сарамяки (2012). Гао и соавт. (2012) анализируют сети сетей. Катастрофический сбой в этих сетях обсуждается Рейсом и соавт. (2014) и Радикки (2015).

Для получения справочной информации о рисовании сетей обратитесь к Ди Баттиста и соавт. (1998). Алгоритмы компоновки сети по направлению силы (также именуемые пружинной компоновкой) были введены Идесом (1984) и усовершенствованы Камадой и Каваи (1989) и Фрухтерманом и Рейнгольдом (1991). Алгоритм компоновки ForceAtlas2, используемый для многих визуализаций в этой книге, был разработан Джакоми и соавт. (2014).



Упражнения

- 1.1 Ознакомьтесь с учебным материалом главы 1 в репозитории книги на GitHub¹.
- 1.2 Рассмотрите сеть с N узлами. При наличии одной связи каково максимальное число узлов, которые может соединять связь? При наличии одного узла каково максимальное число связей, которые могут соединять с этим узлом?
- 1.3 Рассмотрите дорожную карту на рис. 0.9. Решетчатая структура этой сети означает, что большинство узлов имеет одинаковую степень. Какова наиболее распространенная степень узлов в этой сети?
- 1.4 Рассмотрите дорожную карту на рис. 0.9. На Манхэттене много улиц с односторонним движением. Это означает, что хорошая сетевая модель потока дорожного движения, вероятно, будет иметь

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

направленные связи. Взгляните подграф этой сети с решетчатой соединенностью и в котором все улицы имеют одностороннее движение (т. е. каждый узел представляет четырехпутный перекресток двух улиц с односторонним движением). Какова наиболее распространенная степень-на-входе узлов в этом подграфе? Какова наиболее распространенная степень-на-выходе?

- 1.5 Какой объем сети можно использовать для представления объема дорожного движения между каждой парой смежных перекрестков на дорожной карте Манхэттена (рис. 0.9)?
- 1.6 Рассмотрите направленную сеть из N узлов. Теперь рассмотрите суммарную степень-на-входе (т. е. сумму степеней-на-входе по всем узлам в сети). Сравните ее с аналогичной суммарной степенью-на-выходе. Что из перечисленного ниже должно соответствовать действительности для любой такой сети?
 - a. Суммарная степень-на-входе должна быть меньше, чем суммарная степень-на-выходе.
 - b. Суммарная степень-на-входе должна быть больше, чем суммарная степень-на-выходе.
 - c. Суммарная степень-на-входе должна быть равна суммарной степени-на-выходе.
 - d. Ни один из этих пунктов не соответствует действительности во всех случаях.
- 1.7 Рассмотрите ретвитную сеть в Twitter, где пользователи являются узлами, и мы хотим показать, сколько раз данный пользователь ретвитнул другого пользователя. Какой тип связи лучше всего отражает эту связь?
 - a. Ненаправленная невзвешенная.
 - b. Ненаправленная взвешенная.
 - c. Направленная невзвешенная.
 - d. Направленная взвешенная.
- 1.8 Рассмотрите граф совместной встречаемости хештегов в Twitter. В этой сети хештеги являются узлами, и связь между двумя хештегами указывает на частоту появления этих двух хештегов в твитах вместе. Какой тип связи лучше всего отражает эту связь?
 - a. Ненаправленная невзвешенная.
 - b. Ненаправленная взвешенная.
 - c. Направленная невзвешенная.
 - d. Направленная взвешенная.
- 1.9 Рассмотрите сеть, созданную из персонажей истории или пьесы. Узлы – это люди, и между двумя узлами существует связь, если эти персонажи когда-либо вступают в диалог. Какой тип ребра может представлять это отношение? Обоснуйте свой ответ.
 - a. Ненаправленная невзвешенная.
 - b. Ненаправленная взвешенная.

- с. Направленная невзвешенная.
 d. Направленная взвешенная.
- 1.10 Предположим, что мы хотим создать более сложную версию диалоговой сети, которая улавливает то, сколько каждый персонаж говорит и с кем. Какой тип связи лучше всего отражает это отношение?
- a. Ненаправленная невзвешенная.
 b. Ненаправленная взвешенная.
 c. Направленная невзвешенная.
 d. Направленная взвешенная.
- 1.11 Представьте, что в вашей социальной сети есть подсеть, в которой вы и 24 ваших товарища (всего 25 человек) дружите друг с другом. Как называется такая подсеть? И сколько связей содержится в подсети?
- 1.12 Рассмотрите ненаправленную сеть с N узлами. Какое максимальное число связей может существовать в этой сети?
- 1.13 Рассмотрите двудольную сеть из N узлов, N_1 узлов типа 1 и N_2 узлов типа 2 (таких что $N_1 + N_2 = N$). Каково максимальное число связей в этой сети?
- 1.14 Имея полную сеть A с N узлами и двудольную сеть B тоже с N узлами, что из приведенного ниже соответствует действительности для любого $N > 2$?
- a. Сеть A имеет больше связей, чем сеть B .
 b. Сеть A имеет такое же число связей, как и сеть B .
 c. Сеть A имеет меньше связей, чем сеть B .
 d. Ни один из этих пунктов не соответствует действительности по всем таким $N > 2$.
- 1.15 Вспомните, что в полной сети существует связь между каждой парой узлов. Мы знаем, что полная ненаправленная сеть из N узлов имеет $N(N - 1)/2$ ребер. Должна ли любая ненаправленная сеть из N узлов и $N(N - 1)/2$ связей быть с неизбежностью полной? Объясните, почему да или почему нет.
- 1.16 Рассмотрите приведенную ниже матрицу смежности:

$$\begin{array}{c}
 A \ B \ C \ D \ E \ F \\
 \begin{array}{l}
 A \\
 B \\
 C \\
 D \\
 E \\
 F
 \end{array}
 \begin{pmatrix}
 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 \\
 2 & 1 & 3 & 1 & 1 & 0
 \end{pmatrix}
 \end{array}
 \quad (1.11)$$

Запись в i -й строке и j -м столбце указывает вес связи между узлом i и узлом j . Например, запись во второй строке и третьем столбце равна 2, а это значит, что вес связи из узла **B** в узел **C** равен 2. Какую сеть представляет эта матрица?

- a. Ненаправленная невзвешенная.
- б. Ненаправленная взвешенная.
- с. Направленная невзвешенная.
- d. Направленная взвешенная.

- 1.17 Рассмотрите сеть, определенную матрицей смежности в уравнении (1.11). Сколько узлов в этой сети? Сколько связей? Существуют ли какие-либо самонаправленные циклы?
- 1.18 Рассмотрите сеть, определенную матрицей смежности в уравнении (1.11). Существуют ли какие-либо узлы, в которых есть исходящие связи с каждым другим узлом? Если да, то какие узлы? Существуют ли какие-либо узлы, в которых есть входящие связи из каждого другого узла? Если да, то какие узлы?
- 1.19 Рассмотрите сеть, определенную матрицей смежности в уравнении (1.11). Приемник определяется как узел с входящими связями, но без исходящих связей. Какие узлы в сети, если таковые имеются, обладают этим свойством?
- 1.20 Рассмотрите сеть, определенную матрицей смежности в уравнении (1.11). Какова сила-на-входе узла **C**? Какова его сила-на-выходе?
- 1.21 Конвертируйте сеть, определенную матрицей смежности в уравнении (1.11), в ненаправленный невзвешенный граф. (При конвертировании направленного графа в ненаправленный узлы i и j соединяются в ненаправленном графе, если имеется направленная связь из i в j или из j в i или и та и другая.) Возможно, вам захочется распечатать полученную матрицу и/или нарисовать диаграмму сети для справки. Сколько узлов в этой конвертированной сети? Сколько связей?
- 1.22 Рассмотрите невзвешенную, ненаправленную версию сети, определенную матрицей смежности в уравнении (1.11), построенную, как описано в упражнении 1.21. Какова минимальная степень в этой сети? Какова максимальная степень? Какова средняя степень? Какова плотность?
- 1.23 Вообразите две разные ненаправленные сети, каждая с одинаковым числом узлов и связей. Должны ли обе сети иметь одинаковую максимальную и минимальную степень? Объясните, почему да или почему нет. Должны ли они иметь одинаковую среднюю степень? Объясните, почему да или почему нет.

- 1.24** Мы видели, что сеть Facebook невероятно разрежена. Предположим, что у нее примерно 1 млрд пользователей, у каждого из которых в среднем 1000 друзей.
- Предположим, Facebook публикует свой годовой отчет, и он показывает, что, хотя число пользователей в сети осталось прежним, среднее число друзей на одного пользователя увеличилось. Будет ли это означать, что плотность сети увеличилась, уменьшилась или осталась прежней?
 - Предположим вместо этого, что как число пользователей, так и среднее число друзей на одного пользователя удвоились. Будет ли это означать, что плотность сети увеличилась, уменьшилась или осталась прежней?
- 1.25** Netflix хранит данные о предпочтениях клиентов, используя большую двудольную сеть, соединяющую пользователей с кинофильмами, которые они посмотрели и/или оценили. Библиотека кинофильмов Netflix содержит около 100 000 названий, если засчитывать потоковую передачу и отправку DVD по почте. В четвертом квартале 2013 года Netflix сообщила, что у нее около 33 млн пользователей. Предположим, что средняя степень пользователя в этой сети составляет 1000. Примерно сколько связей в этой сети? Считаете ли вы эту сеть разреженной или плотной? Объясните.
- 1.26** Netflix хранит данные о предпочтениях клиентов, используя большую двудольную сеть, соединяющую пользователей с заголовками. Предположим, что с 2013 по 2014 год библиотека Netflix осталась прежнего размера, в то время как число пользователей увеличилось. Далее предположим, что средняя степень пользователя в этой сети осталась неизменной. Увеличилась ли плотность этой сети, уменьшилась или осталась прежней?





https://t.me/it_boooks

Путь: последовательность ребер в сети, которую можно непрерывно отслеживать без повторного отслеживания какого-либо ребра.

Многие типы сетей обладают несколькими фундаментальными признаками. В этой главе мы представим три такие характеристики: сходство между соседями, короткие пути, соединяющие узлы, и треугольники, формируемые общими соседями. *Социальные сети* предоставляют нам знакомые примеры, иллюстрирующие эти признаки. В социальной сети узлы представляют людей, а связи представляют определенный тип социальных отношений, таких как дружба, работа, знакомства или семейные узы. Социальные сети являются наиболее широко изученной категорией сетей; на эту тему существует столетняя литература.

2.1. Рыбак рыбака видит издалека

В социальной сети узлы могут иметь много свойств, таких как возраст, половая идентичность, этническая принадлежность, сексуальные предпочтения, местоположение, интересующие темы и т. д. Зачастую узлы, которые соединены друг с другом в социальной сети, тяготеют к тому, чтобы быть похожими в своих признаках: например, родственники могут жить рядом друг с другом, а друзья могут иметь схожие интересы. Указанное свойство имеет техническое название – *ассортативность*. Рисунок 2.1 иллюстрирует ассортативность на основе узлового признака, представленного цветом. Более яркий, реально существующий пример из Twitter показан на рис. 0.3. Благодаря ассортативности мы можем делать предсказания о качествах человека, инспектируя его соседей. Например, как мы видели в разделе 0.1, исследователи обнаружили, что можно с приемлемой точностью определять сексуальную ориентацию пользователя Facebook и политические предпочтения пользователя Twitter, даже если эти признаки отсутствуют в их профиле, анализируя их круг друзей.

Присутствие ассортативности в социальных сетях может обуславливаться многочисленными факторами. Одна из возможностей заключается в том, что если люди в чем-то похожи, то они с большей вероятностью *выберут* друг друга и установят связь. Это свойство отражено в популярной поговорке: «Рыбак рыбака видит издалека». Его

техническое название – *гомофилия*. Примерами могут служить люди, живущие в географической близости или занимающиеся одним и тем же видом спорта или хобби, – эти индивидуумы, скорее всего, встретятся и станут друзьями. Приложения для завязывания знакомств используют этот вид гомофилии, рекомендуя паросочетания, основанные на общих чертах личности. Обратный механизм заключается в том, что люди, которые являются друзьями, со временем становятся все более похожими в процессе *социального влияния*. Люди – это социальные животные, которые с самого рождения склонны подражать друг другу. Наши идеи, мнения и предпочтения сильно зависят от нашего социального взаимодействия. Отделить причины ассортативности очень трудно – является ли сходство причиной связей, или же связи являются причиной сходства? Нередко эти факторы формируют наши социальные связи одновременно и усиливают друг друга. Мы вернемся к этому вопросу в главе 7.

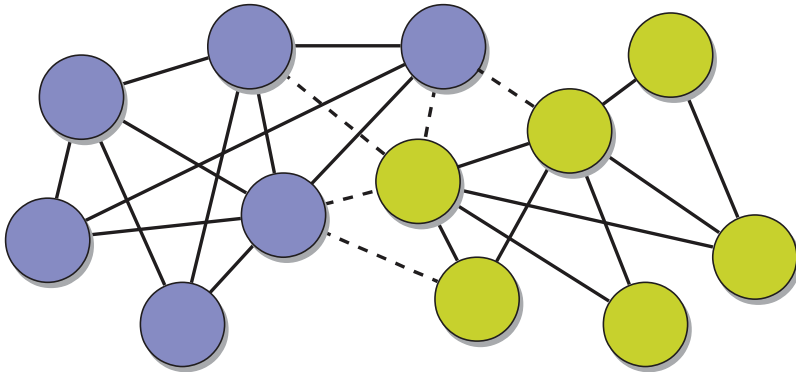


Рис. 2.1 Иллюстрация сетевой ассортативности. Узлы с большей вероятностью будут связаны с другими узлами того же цвета, чем с узлами другого цвета. В частности, большинство связей каждого узла ведет к узлам одинакового цвета, и большинство связей соединяет узлы одинакового цвета. Те несколько связей, которые соединяют узлы разных цветов, показаны пунктирными линиями

Гомофилия обладает и темной стороной. В социальных медиа чрезвычайно легко налаживать связь с людьми, которые разделяют наше мировоззрение, и не дружить или не подписываться на людей с отличающимися мнениями – для этого всего лишь нужно коснуться пальцем экрана. Кроме того, информацией можно делиться и ее потреблять настолько селективно и эффективно, что это позволяет очень эффективно влиять на наши мнения. Указанные механизмы могут приводить к сегрегации и поляризации наших онлайн-сообществ, как видно из рис. 0.3. Когда мы окружены людьми, которые отражают наши собственные взгляды, мы находимся в *эхокамере* – вся информация и мнения, которым мы подвергаемся, отражают наши собственные и подтверждают или усиливают наши идеи, а не оспаривают их. Это бывает опасно тем, что делает нас уязвимыми для ма-

нипуляций посредством дезинформации и социальных ботов, как мы увидим в главе 4.

В библиотеке NetworkX ассортативность сети можно рассчитывать на основе заданного атрибута узла. Имеется две функции для случаев, когда атрибут категориальный, например пол, и когда он числовой, например возраст:

```
assort_a = nx.attribute_assortativity_coefficient(G, category)
assort_n = nx.numeric_assortativity_coefficient(G, quantity)
```

Ассортативность присуща не только социальным сетям; узлы во многих типах сетей обладают свойствами, которые бывают схожими у соседей. Например, узлы в любой сети обладают фундаментальным свойством степени. Ассортативность на основе степени называется *степенной ассортативностью* или *степенной корреляцией*: это происходит, когда высокостепенные узлы, как правило, соединены с другими высокостепенными узлами, тогда как низкостепенные узлы, как правило, в качестве соседей имеют другие низкостепенные узлы. Сети с таким свойством называются *ассортативными*.

Пример ассортативной сети показан на рис. 2.2(а): хабы образуют плотно связанное ядро, тогда как низкостепенные узлы прикреплены друг к другу и/или к узлам ядра свободно. Поэтому мы говорим, что ассортативные сети имеют *структуру ядро–периферия* (подробнее обсуждается в главе 3). Социальные сети часто являются ассортативными. Сети, в которых высокостепенные узлы тяготеют к соединению с низкостепенными узлами и наоборот, называются *дисассортативными*. Пример показан на рис. 2.2(б): хабы расположены в центре звездообразных компонент. Всемирная паутина, интернет, пищевые паутины и другие биологические сети, как правило, являются дисассортативными.

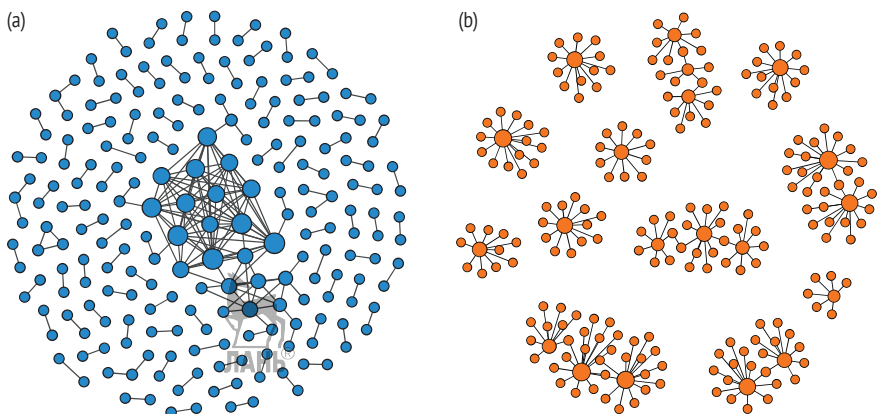


Рис. 2.2 Степенная ассортативность сети, иллюстрируемая (а) ассортативной сетью и (б) дисассортативной сетью

Существует два способа измерения степенной ассортативности сети, оба основаны на измерении *корреляции* между степенями соседних узлов. Мы говорим, что две переменные *положительно (отрицательно) коррелированы*, если более крупные значения одной переменной тяготеют к тому, чтобы соответствовать более крупным (меньшим) значениям другой. Коэффициент корреляции Пирсона является распространенным способом измерения корреляций; он принимает значения в интервале $[-1, +1]$, при этом 0 означает отсутствие корреляции и ± 1 означает идеальную положительную/отрицательную корреляцию.

Одной из мер ассортативности сети является *коэффициент ассортативности*, определяемый как корреляция Пирсона между степенями пар связанных узлов. Используя библиотеку NetworkX, это выглядит вот так:

```
r = nx.degree_assortativity_coefficient(G)
```

Когда коэффициент ассортативности – положителен, сеть является ассортативной, а когда он – отрицателен, сеть является дисассортативной.

Второй метод основан на измерении средней степени соседей узла i :

$$k_{nn}(i) = \frac{1}{k_i} = \sum_j a_{ij} k_j, \quad (2.1)$$

где $a_{ij} = 1$, если i и j являются соседями, и 0 в противном случае. Затем мы определяем функцию k ближайших соседей $\langle k_{nn}(k) \rangle$ для узлов заданной степени k как среднее значение $k_{nn}(k)$ по всем узлам со степенью k . Если $\langle k_{nn}(k) \rangle$ является возрастающей функцией от k , то высокостепенные узлы тяготеют к соединению с высокостепенными узлами, следовательно, сеть является ассортативной; если $\langle k_{nn}(k) \rangle$ уменьшается вместе с k , то сеть является дисассортативной. Используя библиотеку NetworkX, можно рассчитать корреляцию между степенью и ассоциированной с ней соединенностью с соседями:

```
import scipy.stats
knn_dict = nx.k_nearest_neighbors(G)
k, knn = list(knn_dict.keys()), list(knn_dict.values())
r, p_value = scipy.stats.pearsonr(k, knn)
```

Обратите внимание, что для корреляции Пирсона нам нужен пакет `scipy`.

В главе 4 мы узнаем о типе гомофилии, основанной на информационном содержимом, которое имеет большую важность для Всемирной паутины.

2.2. Пути и расстояния

Если можно перейти из *источникового* узла в *целевой*, пройдя по связям в сети, то мы говорим, что между двумя узлами есть *путь*. Путь – это последовательность пройденных связей. Число связей в пути называется *длиной пути*. Между одними и теми же двумя узлами может быть несколько путей. Эти пути могут иметь разную длину и могут разделять или не разделять между собой несколько общих связей. В направленных путях мы должны соблюдать направления связей. *Цикл* – это особый путь, который можно пройти, чтобы вернуться от некоего узла к самому себе. *Простой путь* никогда не проходит по одной и той же связи более одного раза; в этой книге мы сосредоточимся только на простых путях. Поиск путей был самой ранней задачей, изучаемой в науке о сетях (вставка 2.1).

Понятие пути лежит в основе определения *расстояния* между узлами в сети. Естественная мера расстояния между двумя узлами определяется как минимальное число связей, которые необходимо пройти по пути, соединяющему два узла. Такой путь называется *кратчайшим путем*, а его длина – *длиной кратчайшего пути*. Между двумя узлами может иметься несколько кратчайших путей; очевидно, что все они должны иметь одинаковую длину. В разделе 2.5 мы увидим, как отыскивать кратчайший путь между двумя узлами. В некоторых случаях, таких как транспортные сети, можно представить, что связь ассоциирована с географическим расстоянием между смежными узлами. В таких случаях мы можем переопределить длину пути как *сумму расстояний*, ассоциированных со связями вдоль пути; длина пути из Берлина в Рим через Париж равна сумме расстояний из Берлина до Парижа и из Парижа до Рима. Невзвешенная сеть может трактоваться как особый случай, в котором все связи имеют расстояние один.

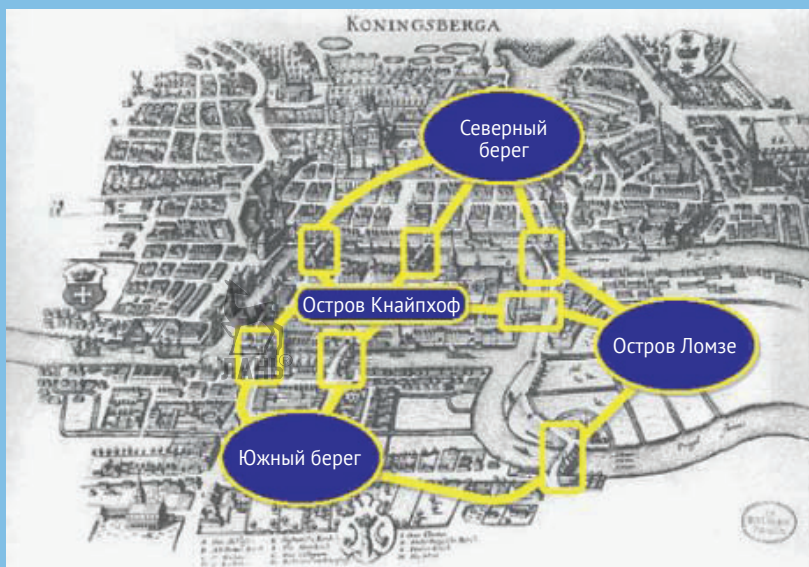
На рис. 2.3 показаны кратчайшие пути между двумя узлами в разных типах сетей, которые зависят от направленности и/или взвешенности сети. В случае ненаправленной невзвешенной сети кратчайший путь – это путь, который минимизирует число проходимых связей, и он одинаков независимо от направления, в котором мы движемся между узлами. Обратите внимание, что между узлами **a** и **b** есть два пути, но тот, который проходит через узел **d**, длиннее на одну связь; кратчайший путь обходит узел **d**, следуя по связи между узлами **e** и **f**. Длина кратчайшего пути равна $\ell_{ab} = 4$. Случай направленной невзвешенной сети отличается тем, что направленные пути должны быть в гармонии с направлением связей вдоль пути. Следовательно, существует только один путь из источника **a** до цели **b**, и он проходит

через **d**. Длина кратчайшего направленного пути равна $\ell_{ab} = 5$. Имейте в виду, что в направленной сети может и не существовать путей между некоторыми парами узлов. Например, если узел имеет только входящие связи, то не существует путей, в которых этот узел является источником. В примере на рис. 2.3 не существует путей из **g** в любой другой узел. Аналогичным образом не существует путей в узлы, которые имеют только исходящие связи, такие как **a**.

Вставка 2.1

Семь мостов Кенигсберга

В 1736 году Леонард Эйлер впервые использовал теорию графов для решения математической задачи. Прусский город Кенигсберг был разделен рекой Прегель на четыре участка суши (Северный и Южный берега, острова Кнайпхоф и Ломзе), соединенные семью мостами. Задача состояла в том, чтобы придумать прогулку по городу, которая пересекала бы каждый мост один и только один раз. Эйлер сформулировал обобщенную версию этой задачи как поиск пути через сеть, где узлы и связи представляют соответственно массивы суши и мосты, и каждая связь должна проходиться ровно один раз.



Эйлер доказал, что такой путь (теперь именуемый в его честь *эйлеровым путем*) существует только в том случае, если все узлы имеют четную степень, кроме источника и цели. Узлы должны иметь четную степень, потому что для каждой входящей связи, прибывающей в узел, должна быть исходящая связь, отправляющаяся из узла. Источник и цель, если они четко различимы, должны иметь нечетную степень, потому что, когда путь начинается (заканчивается), он не «пересекает» узел. Если они совпадают (*эйлеров цикл*), то не может быть узлов с нечетной степенью. Поскольку все четыре узла в сети Кенигсберга имеют нечетную степень, то в таком случае эйлеров путь отсутствовал.

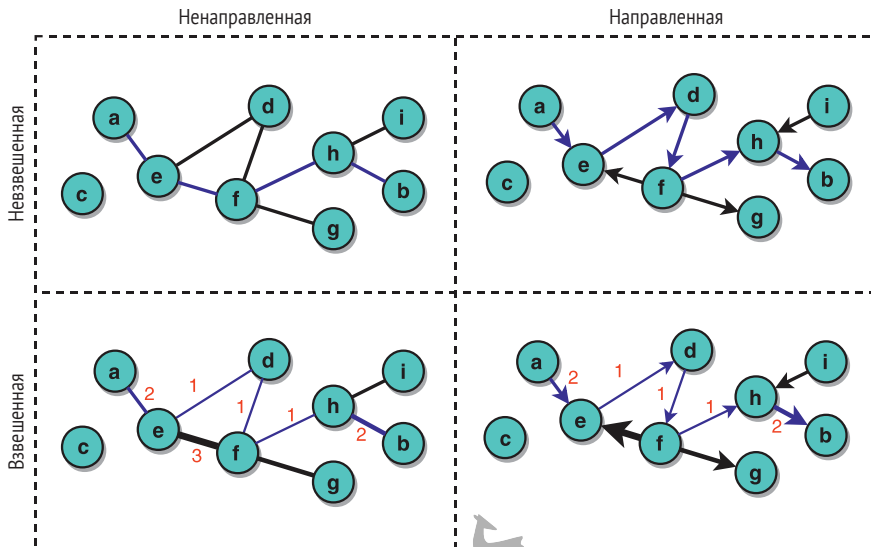


Рис. 2.3 Кратчайший путь в ненаправленной, направленной, невзвешенной и взвешенной сетях. Веса связей представляют расстояния и показаны красным цветом. В каждом случае кратчайший путь между узлами **a** и **b** или из **a** в **b** в направленных случаях выделяется синим цветом. Между узлом **c** и любым другим узлом пути не существует. В направленных сетях кратчайший путь должен быть в гармонии с направлением связей вдоль пути; направленного пути из **b** в **a** не существует

Ненаправленная взвешенная сеть на рис. 2.3 показывает, что происходит, когда мы используем расстояния между связями. В этом случае кратчайший путь между **a** и **b** проходит через **d**: он имеет лишнюю связь, но сумма расстояний между **e** и **f** через **d** равна $1 + 1 = 2$, что меньше расстояния 3, ассоциированного со связью (**e**, **f**). Направленный взвешенный случай прост: кратчайший путь получается путем минимизирования суммы расстояний вдоль пути при соблюдении направлений связей. В обоих примерах взвешенной сети длина кратчайшего пути равна $\ell_{ab} = 7$.

Во многих сетях веса связей выражают меру сходства или интенсивности взаимодействия между двумя соединенными узлами. Затем нас может заинтересовать поиск путей с крупными весами. Распространенный подход заключается в преобразовании весов в расстояния путем взятия обратной величины (единицы, деленной на вес), чтобы крупный вес соответствовал короткому расстоянию. Тогда задача становится эквивалентной отысканию кратчайших путей.

Используя длину кратчайшего пути в качестве меры расстояния между узлами, можно определять агрегатные меры расстояния для всей сети в целом: *средняя длина кратчайшего пути* (или просто *средняя длина пути*) получается путем усреднения длин кратчайших путей по всем парам узлов. В противовес ей *диаметр* сети – это мак-

симальная длина кратчайшего пути между всеми парами узлов (т. е. длина самого длинного кратчайшего пути в сети). Название указанной меры навеяно геометрией, где диаметр – это наибольшее расстояние между любыми двумя точками на окружности.

Формально мы определяем среднюю длину пути ненаправленной невзвешенной сети как

$$\ell = \frac{\sum_{i < j} \ell_{ij}}{\binom{N}{2}} = \frac{2 \sum_{i < j} \ell_{ij}}{N(N-1)}, \quad (2.2)$$

где ℓ_{ij} – это длина кратчайшего пути между узлами i и j , а N – число узлов. Суммирование выполняется по всем парам узлов, и результирующая сумма делится на число пар, чтобы вычислить среднее значение. В случае направленной сети определение будет аналогичным, но расстояние ℓ_{ij} основывается на кратчайшем направленном пути между i и j , и каждая пара узлов рассматривается дважды для путей в обоих направлениях:

$$\langle \ell \rangle = \frac{\sum_{i \neq j} \ell_{ij}}{N(N-1)}. \quad (2.3)$$

Взвешенные случаи аналогичны, причем ℓ_{ij} определяется на основе расстояний между связями. Диаметр сети вычисляется уравнением:

$$\ell_{\max} = \max_{ij} \ell_{ij}. \quad (2.4)$$

Определения средней длины и диаметра пути исходят из допущения, что длина кратчайшего пути определена для каждой пары узлов. Если есть какие-либо пары без пути, то средняя длина и диаметр пути не определены. Например, сети на рис. 2.3 не имеют пути между узлом-одиночкой s и любым другим узлом. Такого рода пропущенные пути можно трактовать как пути с бесконечным расстоянием. С этими случаями можно справиться несколькими способами.

Если необходимо определить среднюю длину пути в сети, где некоторые пути не существуют, в ненаправленных сетях можно использовать следующую ниже формулу:

$$\ell = \frac{\left(\sum_{i < j} \frac{1}{\ell_{ij}} \right)^{-1}}{\binom{N}{2}}. \quad (2.5)$$

Обратите внимание, что, если между i и j пути не существует, $\ell_{ij} = \infty$, и, следовательно, $1/\ell_{ij} = 0$ определена. Ту же хитрость можно использовать в направленных сетях.

В разделе 2.3 мы показываем несколько разных способов расчета расстояния и диаметра сети, когда некоторые пути пропущены.

Для описания типичного расстояния сети можно использовать как среднюю длину пути, так и диаметр. В этой книге мы используем первую. Хотя по определению среднее значение не может превышать максимум, эти два термина иногда используются взаимозаменяемо, поскольку две величины ведут себя одинаково по мере увеличения размера сети.

Библиотека NetworkX имеет функции для определения существования путей, отыскания кратчайших путей и измерения длины пути или средней длины пути в сети. В случае ненаправленной невзвешенной сети из рис. 2.3:

```

nx.has_path(G, 'a', 'c')      # False
nx.has_path(G, 'a', 'b')      # True
nx.shortest_path(G, 'a', 'b') # ['a', 'e', 'f', 'h', 'b']
nx.shortest_path_length(G, 'a', 'b') # 4
nx.shortest_path(G, 'a')      # словарь
nx.shortest_path_length(G, 'a') # словарь
nx.shortest_path(G)           # все пары
nx.shortest_path_length(G)    # все пары
nx.average_shortest_path_length(G) # ошибка
G.remove_node('c')           # сделать граф G связным
nx.average_shortest_path_length(G) # теперь все в порядке

```

Когда указан только источниковый узел, мы получаем словарь со всеми кратчайшими путями или всеми длинами кратчайших путей из источника. Когда ни источник, ни цель не заданы, мы получаем объект с кратчайшими путями для всех пар узлов.

В случае направленных сетей функции те же, но при этом они правильно учитывают направления связей. В случае направленной невзвешенной сети из рис. 2.3:

```

nx.has_path(D, 'b', 'a')      # False
nx.has_path(D, 'a', 'b')      # True
nx.shortest_path(D, 'a', 'b') # ['a', 'e', 'd', 'f', 'h', 'b']

```

В случае взвешенных сетей мы можем хранить ассоциированные со связями расстояния в качестве атрибутов весов. Тогда мы можем указать библиотеке NetworkX интерпретировать веса как расстояния при вычислении длины пути. В случае взвешенной ненаправленной сети из рис. 2.3:

```
nx.shortest_path_length(W, 'a', 'b')          # 4
nx.shortest_path_length(W, 'a', 'b', 'weight') # 7
```

2.3. Соединенность и компоненты

В целях соотнесения структуры и функции сети полезно рассматривать *соединенность* (или в иных случаях *связность*) сети¹. Соединенность определяет многие свойства физической структуры сети. Например, в главе 3 это позволит нам изучить устойчивость сети.

Вспомните из главы 1, что число связей в сети зависит от числа узлов. Это верхняя граница; нижней границы не существует, так как сеть может вообще не иметь связей, как бы это ни было неинтересно. Как мы увидим в главе 5, чем выше плотность, тем больше вероятность того, что сеть связна (т. е. что вы можете добраться до любого узла из любого другого узла, следуя по пути вдоль связей и промежуточных узлов). Чем меньше связей и чем ниже плотность, тем выше вероятность того, что сеть *разъединена*, так как существует несколько узлов или групп узлов, которые недостижимы друг из друга.

В библиотеке NetworkX есть алгоритмы для определения связности сети. Например, все сети на рис. 1.2 связны:

```
K4 = nx.complete_graph(4)
nx.is_connected(K4)          # True
C = nx.cycle_graph(4)
nx.is_connected(C)          # True
P = nx.path_graph(5)
nx.is_connected(P)          # True
S = nx.star_graph(6)
nx.is_connected(S)          # True
```

Если сеть не соединена, мы говорим, что она *разъединена*; она составлена из нескольких *связных компонент* или просто *компонент*. Компонента – это подсеть, содержащая один или несколько узлов, таких, при которых существует путь, соединяющий любую пару этих

¹ Термин «соединенность» (connectedness) синонимичен с термином «связность» и используется в книге взаимозаменяемо, в частности, когда речь идет о связных компонентах. – Прим. перев.

узлов, но не существует пути, соединяющего их с другими компонентами. Самая крупная связная компонента во многих реально существующих сетях включает в себя значительную часть сети и называется *гигантской компонентой*. В связной сети гигантская компонента совпадает со всем графом.

На рис. 2.4 показаны сетевые компоненты, которые определяются по-разному на основе ненаправленных и направленных путей соответственно в ненаправленной и направленной сетях. В ненаправленном случае пример рисунка содержит три компонента. Обратите внимание, что по определению узел-одиночка принадлежит своей собственной компоненте, поскольку он не соединен ни с какими другими узлами. В направленном случае все немного сложнее, потому что при определении возможности достичь некоего узла из другого мы должны обращать внимание на направления связей. Разумеется, мы можем игнорировать направления связей и относиться к связям так, как если бы они были ненаправленными. В этом случае мы называем компоненты *слабо связными*. Направленная сеть на рис. 2.4 состоит из трех слабо связных компонент. Однако не все узлы в слабо связной компоненте могут быть достигнуты друг из друга по *направленным* путям.

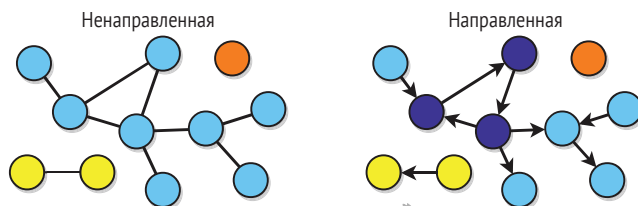


Рис. 2.4 Связные компоненты. Цвета обозначают разные компоненты. В примере ненаправленной сети мы наблюдаем три компонента, одна из которых является узлом-одиночкой, т. е. синглетоном. Светло-голубые узлы составляют гигантскую компоненту. В приведенном примере мы наблюдаем три *слабо* связные компоненты. Самая крупная слабо связная компонента содержит узлы разных оттенков синего; темно-синие узлы составляют самую крупную *сильно* связную компоненту

В *сильно связной* компоненте существует по меньшей мере один направленный путь между каждой парой узлов в обоих направлениях. На рис. 2.4 самая крупная сильно связная компонента содержит три узла; каждый другой узел принадлежит своей собственной сильно связной компоненте. Обратите внимание, что в сильно связной сети или компоненте существует по меньшей мере один направленный цикл из каждого узла. В целях понимания причины давайте рассмотрим любые два узла **a** и **b** в сильно связной сети. Поскольку должен существовать направленный путь из **a** в **b** и один из **b** в **a**, цикл может быть построен путем комбинирования обоих.

Определив связные компоненты, давайте вернемся к вопросу измерения расстояния в сети в случае, когда сеть разъединена. Один из способов состоит в рассмотрении узлов только в гигантской компо-

ненте. Еще один подход заключается в усреднении расстояния между парами узлов только в одной и той же компоненте, но с учетом всех компонент. В целях вычисления диаметра разъединенной сети можно рассчитать диаметр каждой компоненты, а затем взять максимум.

Мы можем выявить множество узлов, из которых можно добраться до сильно связанной компоненты S , но до которой нельзя добраться из S – если бы было можно, то они были бы частью компоненты S . Это множество называется компонентой-на-входе компоненты S . Мы определяем компоненту-на-выходе компоненты S аналогичным образом как множество узлов, до которых можно добраться из S , но из которых нельзя добраться до S .

Мы говорим, что направленная сеть является *сильно связной* (син. – сильно соединенной), если она представляет одну сильно связную компоненту. Направленная сеть является *слабо связной*, если она представляет одну слабо связную компоненту.

Библиотека NetworkX предлагает функции для выявления связных сетевых компонент. Допустим, что G и D – это соответственно ненаправленная и направленная сети из рис. 2.4:

```
nx.is_connected(G)                # False
comps = sorted(nx.connected_components(G), key=len, reverse=True)
nodes_in_giant_comp = comps[0]
GC = nx.subgraph(G, nodes_in_giant_comp)
nx.is_connected(GC)                # True
nx.is_strongly_connected(D)        # False
nx.is_weakly_connected(D)         # False
list(nx.weakly_connected_components(D))
list(nx.strongly_connected_components(D)) # много узлов-одиночек
```

В этом примере мы используем встроенную в Python функцию `sorted()` для перечисления и сортировки результата на выходе из функции `connected_components()`. Мы задаем `key=len` для сортировки по размерам компонент и `reverse=True` для распечатки в убывающем порядке. Тогда первым элементом будет гигантская компонента.

2.4. Деревья

Давайте введем специальный класс ненаправленных связных сетей, таких, в которых удаление любой одной связи будет разъединять сеть на две компоненты. Такие графы называются *деревьями*.

Число связей в дереве равно $L = N - 1$. Для того чтобы убедить себя в том, что это так, следует начать с сети с $N = 2$ узлами, которой нужна $L = 1$ связь, чтобы стать связной. Тогда, добавляя по

одному узлу за раз, мы должны добавлять связь для соединения нового узла с каким-либо существующим узлом. Поэтому, число связей всегда равно числу узлов минус один. Удаление любой связи будет разъединять по меньшей мере один узел.

Деревья обладают и другими интересными свойствами. Они не имеют циклов. Мы можем доказать, что деревья не могут иметь циклов от противного: если бы дерево имело цикл, то мы могли бы удалить по меньшей мере одну связь цикла, не разъединяя его. Следовательно, сеть не была бы деревом – а это противоречие. Поскольку циклов не существует, для любой пары узлов существует только один путь, который их соединяет.

Деревья имеют *иерархическую* структуру. Можно выбрать любой узел в дереве и назвать его *корнем*. Каждый узел в дереве соединен с родительским узлом (ближе к корню) и с одним или несколькими дочерними узлами (дальше от корня). Исключением является корень, у которого нет родительского узла, и так называемые *листья* дерева, у которых нет дочерних узлов. Иерархическая структура деревьев показана на рис. 2.5.

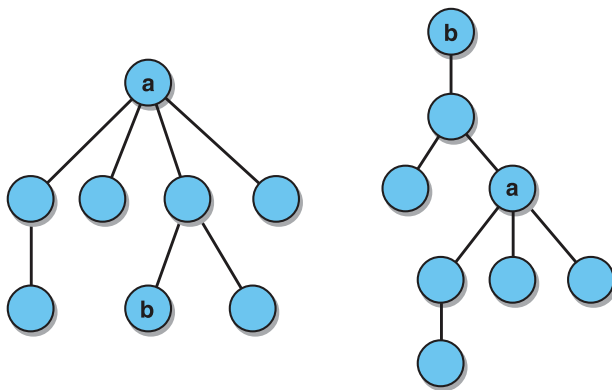


Рис. 2.5 Иерархическая структура деревьев. Одно и то же дерево изображено с двумя разными компоновками, соответственно, с узлами **a** и **b**, взятыми в качестве корней и расположенными вверху. Каждый узел имеет свой родительский узел выше (корень не имеет родительского узла) и свои дочерние узлы ниже (листья находятся внизу и не имеют дочерних узлов)

В библиотеке NetworkX есть алгоритмы, позволяющие определять, является ли сеть деревом или нет. Например, полная сеть с более чем двумя узлами имеет циклы и, следовательно, не является деревом. Звездчатая и путевая сети на рис. 1.2 являются примерами деревьев:

```
K4 = nx.complete_graph(4)
nx.is_tree(K4)           # False
C = nx.cycle_graph(4)
nx.is_tree(C)           # False
P = nx.path_graph(5)
nx.is_tree(P)           # True
S = nx.star_graph(6)
nx.is_tree(S)           # True
```

2.5. Отыскание кратчайших путей

В разделе 2.2 мы обсудили тему кратчайших путей. Но как отыскать кратчайший путь между двумя узлами на практике? Для этого необходимо составить карту и перемещаться по всей сети. Это делается с помощью библиотеки NetworkX и других инструментов сетевого анализа. Как мы увидим в главе 4, это также делается поисковыми машинами с помощью *обходчиков Всемирной паутины*, компьютерных программ, которые автоматически бродят по страницам паутины, отыскивая и сохраняя новые страницы.

Алгоритм или процедура навигации по сети, начинающейся с *источникового* узла и отыскания кратчайшего пути между источником и каждым другим узлом в сети, называется *поиском сперва в ширину*. Его идея состоит в том, что мы посещаем всю «ширину» сети на некотором расстоянии от источника и только потом опускаемся на большую «глубину», дальше от источника. Этот процесс проиллюстрирован на рис. 2.6 для простой ненаправленной сети: начиная с некоторого источникового узла, мы посещаем его соседей (слой 1) и устанавливаем расстояние между этими вершинами от источника равным единице. Затем мы посещаем соседей узлов в слое 1, за исключением уже разведанных узлов (слой 2), и устанавливаем их расстояние равным двум. Затем мы посещаем соседей узлов слоя 2, если они ранее не посещались (слой 3), устанавливая их расстояние равным трем; и т. д. На рис. 2.6 показано, что каждый слой содержит все узлы, расположенные на одинаковом расстоянии от источника. Если сеть связна, то все узлы достигаются, и им назначается расстояние от источника. Указанная процедура аналогична для направленных сетей, таких как Всемирная паутина, за исключением того, что мы достигаем узлов только по направленным путям от источника.

В целях отыскания кратчайшего пути из источника в другие узлы алгоритм поиска сперва в ширину строит направленное *дерево кратчайшего пути*, содержащее те же узлы, что и изначальная сеть, но только подмножество связей. Дерево соотносит кратчайший путь между своим корнем (источниковым узлом) и всеми остальными узлами. Указанный алгоритм показан на рис. 2.7 для направленной сети; подробности его имплементации см. во вставке 2.2.

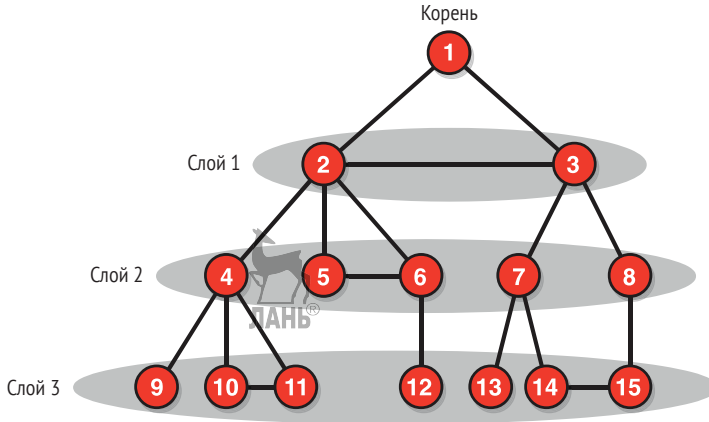


Рис. 2.6 Поиск сперва в ширину. В данном случае в качестве источника выбирается узел **1**. Сначала мы посещаем соседей узла **1**, т. е. узлы **2** и **3**. Это слой 1, включающий все узлы в одном шаге от источника. Затем переходим к их соседям, узлам **4, 5, 6, 7, 8**, которые находятся в двух шагах от источника (слой 2). Наконец, мы достигаем узлов **9, 10, 11, 12, 13, 14, 15** на расстоянии трех шагов от источника (слой 3)

Вставка 2.2

Поиск сперва в ширину

Алгоритм поиска сперва в ширину берет на входе источниковый узел. В целях имплементирования указанного алгоритма каждый узел должен иметь атрибут, используемый для хранения его расстояния от источника. В дополнение к этому мы должны поддерживать очередь узлов, которую называем *границей продвижения*. Очередь – это структура данных, организованная по принципу «первым пришел – первым вышел» (FIFO): узлы извлекаются (исключаются из очереди) в том порядке, в котором они ставятся туда (вставляются в очередь).

Первоначально источниковый узел s ставится в очередь границы продвижения. Его расстояние устанавливается равным $\ell(s, s) = 0$, и для всех остальных узлов расстояние устанавливается равным общепринятому нереалистичному значению, к примеру -1 . Сеть, которая в конечном итоге станет деревом кратчайшего пути, инициализируется без связей.

На каждой итерации мы посещаем следующий узел i в границе продвижения. Узел исключается из очереди. Затем для каждого преемника j узла i (или каждого соседа, если сеть является ненаправленной) мы выполняем три шага, если только у узла j уже не установлено расстояние.

1. Поставить j в очередь границы продвижения.
2. Установить расстояния j от источника равными $\ell_{s,j} = \ell_{s,i} + 1$.
3. Добавить направленную связь ($i \rightarrow j$) в дерево кратчайшего пути.

Указанная процедура заканчивается, когда граница продвижения становится пустой. Если какие-либо узлы остаются с неизвестным расстоянием, то они недоступны из источника; они должны находиться в другой связной компоненте сети.

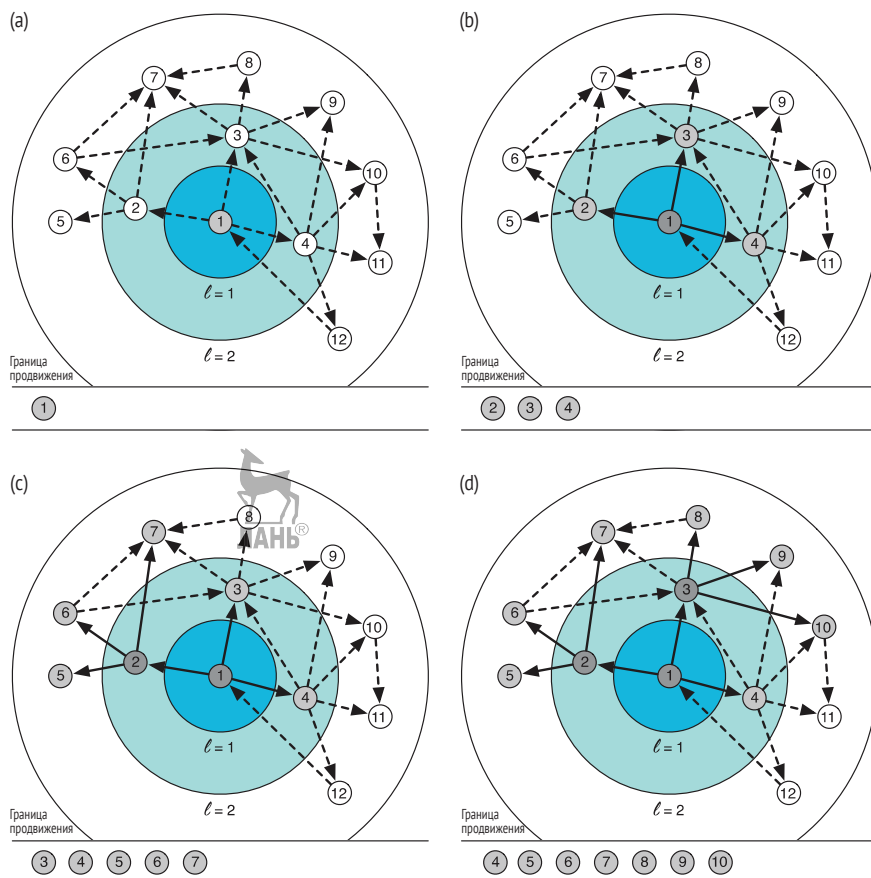


Рис. 2.7 Иллюстрация алгоритма поиска сперва в ширину для обхода направленной сети и отыскания кратчайших путей из источникового узла. Узлы окрашиваются в светло-серый цвет, когда они добавляются в очередь границы продвижения, и темно-серый, когда они удаляются из этой очереди. Связи превращаются из пунктирных линий в сплошные, когда они добавляются в дерево кратчайших путей. (а) Граница продвижения инициализируется источниковым узлом 1. (б) Узел 1 исключается из очереди, а его преемники 2, 3 и 4 ставятся в очередь границы продвижения. (с) Узел 2 исключается из очереди, и его преемники 5, 6 и 7 ставятся в очередь. (д) Узел 3 исключается из очереди, а его преемники 8, 9 и 10 ставятся в очередь. Узел 7 уже находится очереди границы продвижения, поэтому связь с ним из узла 3 игнорируется. Алгоритм поиска сперва в ширину отслеживает посещенные узлы, потому что их расстояния заданы, поэтому они больше не ставятся в очередь границы продвижения. Например, при посещении узла 4 можно игнорировать узел-преемник 3, поскольку его расстояние от источника уже равно единице. На следующем шаге будут посещены все узлы на расстоянии единицы от источника, чтобы иметь возможность посетить узлы на расстоянии два

В результате исполнения алгоритма поиска сперва в ширину всем узлам в той же связной компоненте, что и источниковый узел, назначается расстояние от источника. В целях отыскания кратчайшего пути от источника к любому целевому узлу мы должны следовать по связям в дереве кратчайшего пути назад от целевого узла через предшественников в верхних слоях до тех пор, пока не достигнем ис-

точника. Вспомните, что в дереве имеется один единственный путь к корню; у каждого узла есть один предшественник. И тогда, чтобы получить кратчайший путь от источника к цели, мы должны обратиться путь вспять. В ненаправленной сети он тот же самый, что и путь от цели к источнику, но в направленной сети они могут отличаться.

В примере на рис. 2.7 мы видим дерево кратчайшего пути, поскольку его связи начерчены сплошными линиями. Например, предположим, что нас интересует кратчайший путь из узла 1 в узел 7. Поиск сперва в ширину установил длину этого пути равной $\ell_{1,7} = 2$. В целях отыскания пути мы переходим от 7 к его предшественнику в дереве кратчайшего пути, т. е. узлу 2, а затем к его предшественнику, т. е. корневому узлу 1. Обращая этот путь вспять, мы получаем кратчайший путь $1 \rightarrow 2 \rightarrow 7$. Обратите внимание, что это не единственный кратчайший путь – путь $1 \rightarrow 3 \rightarrow 7$ имеет ту же длину, но алгоритм определяет только один кратчайший путь из источника. Обратите также внимание на то, что в этой направленной сети кратчайший путь из узла 7 в узел 1 не тот же самый; фактически такого пути не существует.

Алгоритм поиска сперва в ширину отыскивает кратчайший путь из одного источника во все остальные узлы невзвешенной сети. Несколько более сложные алгоритмы также существуют для кратчайших путей во взвешенных сетях. Если мы хотим отыскать кратчайший путь между каждой парой узлов, то нам нужно выполнить алгоритм N раз, по одному разу из каждого узла в качестве источника. Это выливается в большие вычислительные затраты. На самом деле в качестве упражнения вам следует попробовать использовать метод библиотеки NetworkX под названием `shortest_path(G)` (или `shortest_path_length(G)`) на некоторых сетях репозитория GitHub книги¹ (см. также табл. 2.1).

Таблица 2.1. Средняя длина пути и коэффициент кластеризации примеров разных сетей. Сети такие же, как в табл. 1.1, их числа узлов и связей также показаны. Веса связей игнорируются. Средняя длина пути измеряется только на гигантской компоненте; в случае направленных сетей мы рассматриваем направленные пути в гигантской сильно связанной компоненте. В целях измерения коэффициента кластеризации в направленных сетях мы игнорируем направления связей

Сеть	Узлы (N)	Связи (L)	Средняя длина пути ($\langle \ell \rangle$)	Коэффициент кластеризации (C)
Facebook, Северо-Западный университет	10567	488 337	2.7	0.24
IMDB, кинофильмы и кинозвезды	563 443	921 160	12.1	0
IMDB, кинозвезды, снимавшиеся вместе	252 999	1 015 187	6.8	0.67
Twitter, политика США	18 470	48 365	5.6	0.03
Электронная почта компании Энрон®	87 273	321 918	3.6	0.12
Статьи по математике в «Википедии»	15 220	194 103	3.9	0.31
Интернет-маршрутизаторы	190 914	607 610	7.0	0.16
Авиационные перевозки в США	546	2781	3.2	0.49
Авиационные перевозки по всему миру	3179	18 617	4.0	0.49
Взаимодействие дрожжевых белков	1870	2277	6.8	0.07
Мозг <i>C. elegans</i>	297	2345	4.0	0.29
Экологическая пищевая паутина Everglades	69	916	2.2	0.55

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

Вы заметите, что для крупных сетей требуется очень много времени; даже если существуют короткие пути, их не обязательно легко отыскать. К счастью, как мы увидим в разделе 7.4, сети часто предоставляют нам подсказки, чтобы иметь возможность эффективно отыскивать целевой узел, следуя эвристическим правилам.



2.6. Социальное расстояние

Средняя длина пути, определенная в разделе 2.2, характеризует близость или отдаленность, с которой по нашим ожиданиям узлы будут находиться в сети. Интуитивно в решетчатой сети, такой как дорожные сети и электросети, пути бывают длинными. Типично ли это для многих реально существующих сетей? Давайте начнем с рассмотрения нескольких социальных сетей, в которых этот вопрос был подробно разведан.

Сети соавторства – это хорошо изученный вид социальной коллаборационной сети, потому что собирать данные об узлах и связях относительно легко. Узлы – это ученые, а связи можно добывать из цифровых библиотек. Когда мы видим публикацию, написанную двумя или более учеными в соавторстве, мы можем сделать вывод о связях между ними в сети.

Пол Эрдеш был известным математиком, который внес важный вклад в науку о сетях, обсуждаемую в главе 5. (Более подробную информацию о его жизни см. во вставке 5.1.) Математики любят изучать свое расстояние в сети соавторства от конкретного узла, соответствующего Эрдешу. Они называют это расстояние своим *числом Эрдеша* (вставка 2.3). У многих математиков очень малое число Эрдеша. Рисунок 2.8 иллюстрирует коллаборационную сеть с участием Эрдеша и его более чем 500 соавторов. На самом деле ученые не просто близко расположены к Эрдешу, они близко расположены ко всем. Это типичная ситуация для коллаборационных сетей: между всеми парами узлов существуют короткие пути. Выберите любых двух ученых, и они будут не очень далеко друг от друга.

Вставка 2.3

Число Эрдеша

Пол Эрдеш был одним из величайших математиков мира. Он также выделяется среди ученых своей удивительной продуктивностью и числом соавторов. Поэтому Эрдеш играет важную роль в связности научно-коллаборационной сети, поскольку через него можно перейти от многих узлов графа ко многим другим. Это настолько важно, что в его честь была определена специальная мера: *число Эрдеша*. Многие ученые с гордостью показывают свое число Эрдеша на своих домашних страницах и в резюме. Это число определяется просто как длина кратчайшего пути в сети соавторства от ученого до Пола Эрдеша. Есть даже онлайн-

вый инструмент, который вычисляет число Эрдеша для математиков (www.ams.org/mathscinet/collaborationDistance.html). Например, Эрдеш был соавтором Фан Чунга, который написал доклад в соавторстве с Алексом Веспиньяни, соавтором двух авторов этой книги, у которых, следовательно, есть число Эрдеша, равное трем. Из-за огромного числа соавторов Пола Эрдеша число ученых с малым числом Эрдеша довольно велико.

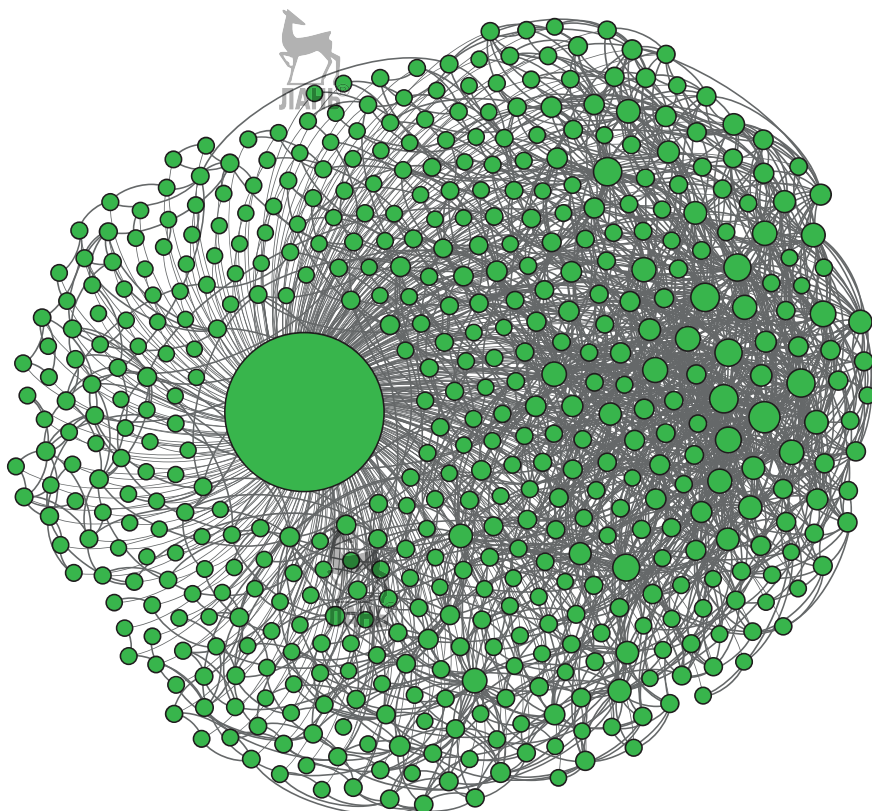


Рис. 2.8 Эгосеть Пола Эрдеша (большой узел в центре) в сети соавторства. Определение эгосетей приводится в разделе 1.4

Оказывается, не только коллаборационные сети, но и практически все социальные сети имеют очень короткие пути между узлами. Вы, вероятно, знаете кого-то, кто знает кого-то, кто знает кого-то... и за несколько шагов вы сможете добраться до любого человека на планете! В целях демонстрации из более знакомой области давайте обратимся к социальной сети, соединяющей кинозвезд. Как мы видели в главе 0, узлы – это актеры и актрисы, и два узла связаны, если они снимались в одном кинофильме вместе. Забавная игра «*Шесть степеней Кевина Бэйкона*» берет свое начало в такой сети. Игра, показанная на рис. 2.9, состоит в отыскании кратчайшего пути, соединяющего произвольную актрису или актера, с Кевином Бэйконом из сети ки-

носозвездий. Например, путь длиной $\ell = 2$ соединяет Мэрилин Монро с Кевином Бейконом. Вы можете играть в эту игру онлайн в *Оракуле Бейкона* (oracleofbacon.org). Веб-сайт черпает данные для строительства сети из интернет-базы данных кинофильмов (IMDB.com). Хотя Кевина Бейкона часто в шутку считают «единственным в своем роде» хабом сети кинозвезд, на самом деле он не особенный; вы можете ввести любую пару актеров/актрис, и Оракул покажет вам кратчайший путь в виде последовательности узлов (кинозвезды) и связи (кинофильмы). Можете ли отыскать две знакомые кинозвезды, сепарированные более чем четырьмя связями? Сыграйте в эту игру и попробуйте!

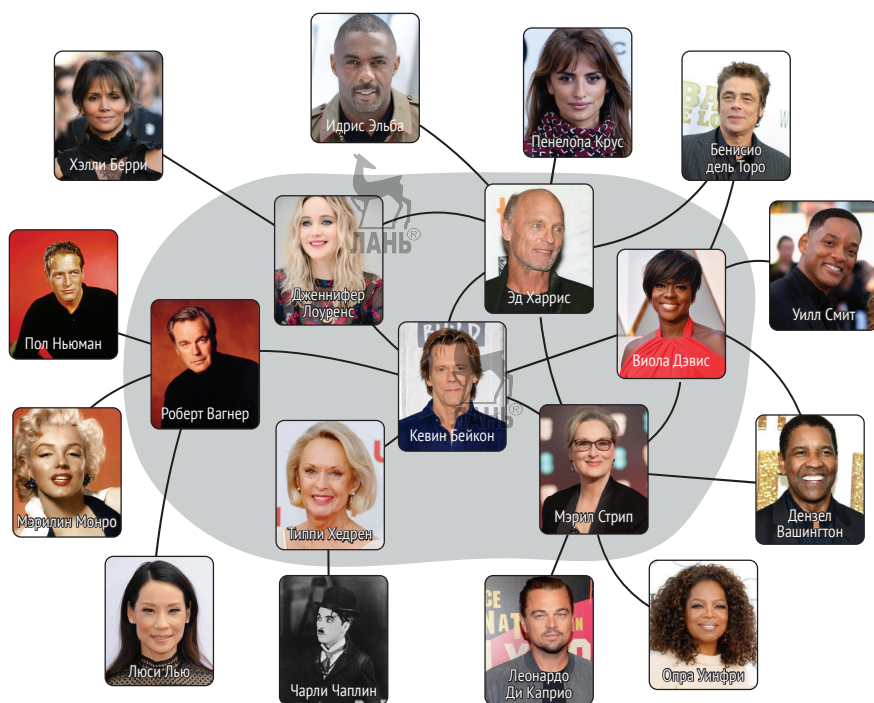


Рис. 2.9 Иллюстрация игры в «Шесть степеней Кевина Бейкона». Несколько узлов, соединенных с Кевином Бейконом в сети кинозвезд, показаны в заштрихованной области вместе со связями между ними. Также включена небольшая выборка узлов с расстоянием $\ell = 2$. Фото сделано силами Getty Images

Число Эрдеша и Оракул Бейкона демонстрируют, что отыскивать длинные пути в реальных сетях непросто. Если подумать, то понятие короткого социального расстояния – то есть что в социальной сети мы все находимся всего в нескольких шагах друг от друга – является знакомым. Сколько раз вы встречали кого-то, а затем удивлялись тому, что обнаруживали общего друга? Низкое ожидание встречи с другом своего друга коренится в нашей интуиции о крохотности нашего круга знакомых по сравнению со всем населением. И все же подобные

вещи случаются достаточно часто, заставляя нас восклицать: «Да уж, мир действительно мал!» *Малый мир* – это популярное представление о том, что социальные расстояния в среднем невелики. Следовательно, число друзей наших друзей там намного-намного больше, чем мы думаем, и отыскание коротких путей в социальной сети в конце концов не является таким уж и странным.

2.7. Шесть степеней сепарации

Название игры «*Шесть степеней Кевина Бейкона*» вдохновлено понятием *шести степеней сепарации*¹. Идея та же, что и в малом мире: любые два человека в мире соединены короткой цепочкой знакомств. Другими словами, социальные сети имеют малый диаметр и еще меньшую среднюю длину пути. Число «шесть» в выражении произошло от венгерского автора Фригьеша Каринти (Frigyes Karinthy) в 1920-х годах, и некоторая заслуга принадлежит также итальянскому изобретателю Гульельмо Маркони (Guglielmo Marconi) за то, что он выдвинул ту же идею 20 лет до этого, в начале 1900-х годов. Однако то, что сделало выражение «шесть степеней» знаменитым, было экспериментом, проведенным психологом Стэнли Милграмом (Stanley Milgram)² в 1960-х годах, который предоставил первые эмпирические данные, доказывающие существование малых миров.

Милграм хотел измерить социальное расстояние между незнакомыми людьми. Поэтому он попросил 160 испытуемых в Небраске и Канзасе переслать письмо знакомому с инструкциями, чтобы письмо в конечном итоге дошло до целевого лица в Массачусетсе. Каждый получатель должен был переслать письмо кому-то известному, кто, скорее всего, знал адресата. Цели достигли только 42 письма (26 %). Однако в этих случаях длина пути была на удивление короткой – от 3 до 12 шагов. На рис. 2.10 показан типичный путь в размере 4 шагов. Средняя длина пути составляла чуть более 6 шагов, что в конечном итоге вдохновило на пьесу под названием «Шесть степеней сепарации», которая в конечном итоге популяризировала понятие малого мира. Эксперимент Милграма был повторен в 2003 году с использованием электронной почты для рекрутирования большего числа испытуемых. Было поставлено 18 целей в 13 странах. Из более чем

¹ Синонимичными ему являются понятия шести степеней разделения или шести степеней рукопожатия. Однако в переводе принят именно такой вариант, поскольку термин «сепарация» используется в качестве антитезы термину «когезия» (см. главу 6). – *Прим. перев.*

² Милграм известен еще одним, очень спорным экспериментом, в котором испытуемым было поручено причинять боль другим людям. Цель состояла в том, чтобы проверить, в какой степени человек способен на аморальные поступки в результате давления со стороны власти.

24 000 начатых цепочек было завершено только 384, средняя длина которых составляла 4 шага. При объяснении большого числа разорванных цепочек авторы оценили медианную длину пути в 5–7 шагов в полном соответствии с «шестью степенями» Милграма. Еще совсем недавно, в 2011 году, исследователи Facebook и Миланского университета проинспектировали 721 млн активных пользователей Facebook, которые были активны в то время (более 10 % мирового населения), с 69 млрд друзей среди них и обнаружили, что средняя длина пути составляла 4.74 шага.

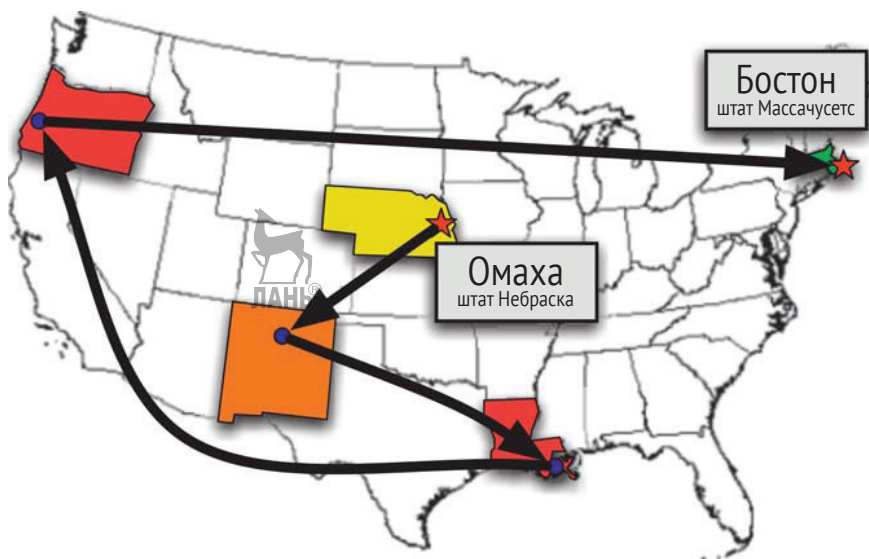


Рис. 2.10 Путь, по которому следует одно из писем в эксперименте Милграма. Испытуемый – источник в Омахе, штат Небраска – отправил письмо знакомому в Санта-Фе, штат Нью-Мексико. Оттуда письмо было переслано людям в Новом Орлеане, Луизиана и в Юджине, штат Орегон, прежде чем оно достигло цели в Бостоне, штат Массачусетс

До сих пор мы называли пути, которые отыскиваем, играя в игру, подобную «Шести степеням Кевина Бейкона» или тем, о которых сообщалось в исследованиях Милграма и других исследователей, «короткими». Но когда можно называть путь *коротким*? По сравнению с чем? Будем ли мы называть путь с 6 шагами *коротким* в сети всего из 10 узлов? Очевидно, что мы должны дать более точное определение, разяснив, что конкретно подразумевается под *короткими* путями, и это определение должно соответствовать размеру сети. На самом деле при рассмотрении сетей (или подсетей) разных размеров имеет больше смысла наблюдать взаимоотношение между средней длиной пути $\langle \ell \rangle$ и размером сети N . Мы говорим, что средняя длина пути *коротка*, когда она растет очень медленно вместе с увеличением размера сети.

Выразить медленный рост математически можно, сказав, что средняя длина пути шкалируется логарифмически вместе с размером сети:

$$\langle \ell \rangle \sim \log N.$$

Логарифм числа a по основанию b , $\log_b a$, – это такая экспонента c , при которой $b^c = a$. Основание $b = 10$ имеет широкое применение; $\log_{10} 10 = 1$, потому что $10^1 = 10$, $\log_{10} 100 = 2$, потому что $10^2 = 100$, $\log_{10} 1000 = 3$ и т. д. Следовательно, логарифм – это функция, которая растет очень медленно.

Это означает, что сеть может иметь десятки миллионов узлов, и все же ее средняя длина пути будет выражаться однозначными цифрами. Более того, сеть может увеличиться в размерах многократно, в то время как средняя длина пути будет увеличиться всего на несколько шагов.

Короткие пути, которые подчиняются такого рода отношениям, можно найти в социальных сетях, включая академическое сотрудничество, сети актеров, сети школьных друзей и онлайн-социальные сети, такие как Facebook. Короткие социальные расстояния бывают полезны, к примеру, когда мы ищем работу. Но короткие пути не являются исключительным признаком социальных сетей. На самом деле отыскание путей – это то, что мы делаем в рутинном порядке во всех видах сетей, например когда мы бронируем продолжительный рейс и стараемся минимизировать число промежуточных остановок. Отыскание путей в сети бывает и увлекательным занятием. *Вики-гонки* (Wikiracing) – это игра с гипертекстовым поиском, предназначенная для работы с «Википедией». Игрок должен переходить от источниковой статьи к целевой статье, обе отбираемых случайно, исключительно путем кликания по связям в каждой статье. Цель состоит в том, чтобы достичь цели за наименьшее число кликов (т. е. отыскать путь в сети с малым числом связей). Существуют командные версии игры и версии с гонкой на время. Можно играть в несколько версий этой игры онлайн, как например на веб-сайте «Вики-игра» (thewikigame.com). Вы будете поражены тем, как быстро можно достигать любой цели, немного попрактиковавшись. Это говорит нам о том, что в «Википедии» есть короткие пути. То же самое верно и для Всемирной паутины, как мы увидим в главе 4.

Как оказалось, короткие пути являются повсеместным признаком почти во всех реально-существующих сетях; решетчатые сети являются одним из немногих исключений. В табл. 2.1 представлена средняя длина пути разных сетей¹. Во всех этих примерах средняя длина

¹ Наборы данных для этих сетей доступны в репозитории книги на GitHub: github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

пути составляет всего несколько шагов. В случае с сетью кинофильмов и кинозвезд пути выглядят длиннее. Однако имейте в виду, что это двудольная сеть, в которой связь соединяет кинофильм и актера/актрису. Если мы рассмотрим сеть киносозвездий, в которой две кинозвезды соединены, если они выступали вместе (как на рис. 2.9), то связи ассоциированы с кинофильмами; в этом случае средняя длина пути сокращается примерно вдвое. В главе 5 мы будем отыскивать короткие пути даже в самых простых сетях, где связи назначаются случайным образом.

2.8. Друг моего друга

В социальной сети если Алиса и Боб оба являются друзьями Чарли, то они также, скорее всего, будут друзьями друг друга. Другими словами, существует большая вероятность того, что друг моего друга также является моим другом. Это приводит к наличию большого числа *треугольников* в сети. Как показано на рис. 2.11(а), треугольник представляет собой триаду (множество из трех узлов), где каждая пара узлов соединена. Способность к установлению соединения между соседями узлов является важным признаком локальной структуры сети, поскольку она отражает то, насколько тесно узлы переплетены или кластеризованы.

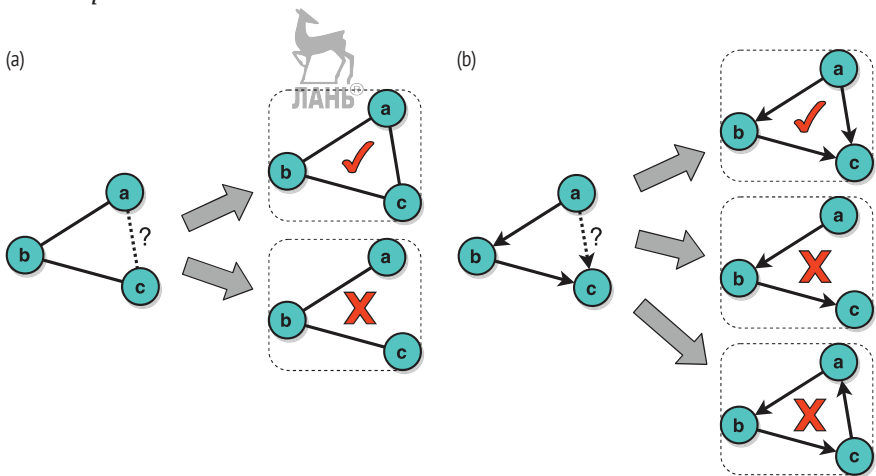


Рис. 2.11 Триады и треугольники. (а) В ненаправленной сети узел **b** имеет соседей **a** и **c**. Они могут образовывать или не образовывать треугольник в зависимости от наличия или отсутствия связи между **a** и **c**. (б) В направленной сети узел **a** связан с **b**, а узел **b** связан с **c**. Укороченная связь из **a** в **c** образует направленный треугольник

Коэффициент кластеризации узла – это доля пар соседей узла, которые соединены друг с другом. Это то же самое, что и соотношение между

числом треугольников, включающих узел, и максимальным числом треугольников, в которых узел *может* участвовать.

Коэффициент кластеризации узла i формально определяется как

$$C(i) = \frac{\tau(i)}{\tau_{\max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i - 1)}, \quad (2.6)$$

где $\tau(i)$ – это число треугольников, включающих i . Максимально возможное число треугольников для i – это число пар, образованных его k_i -соседями. Обратите внимание, что $C(i)$ определено только в том случае, если степень $k_i > 1$ из-за членов k_i и $k_i - 1$ в знаменателе: узел должен иметь по меньшей мере двух соседей, чтобы любой треугольник был возможен.

Коэффициент кластеризации всей сети целиком – это среднее значение коэффициентов кластеризации для ее узлов:

$$C = \frac{\sum_{i:k_i > 1} C(i)}{N_{k > 1}}. \quad (2.7)$$

Узлы со степенью $k < 2$ при расчете среднего коэффициента кластеризации исключаются.

На рис. 2.12 показано, как рассчитать коэффициент кластеризации для нескольких узлов в сети. Узел **a** имеет двух соседей **f** и **g**, которые соединены друг с другом, образуя треугольник. Поэтому его коэффициент кластеризации равен $C(a) = 1/1 = 1$. Узел **b** имеет четырех соседей. Соединены только две из шести пар соседей: (**e**, **c**) и (**c**, **g**). Следовательно, $C(b) = 2/6 = 1/3$. Узел **c** имеет трех соседей, которые образуют два треугольника через связи (**e**, **b**) и (**b**, **g**). Третий возможный треугольник не реализован, потому что связь (**e**, **g**) отсутствует. Следовательно, $C(c) = 2/3$. Наконец, узел **d** имеет одного единственного соседа **e**, следовательно, $C(d)$ не определен.

Наше определение коэффициента кластеризации применимо только к ненаправленным сетям, потому что мы определили только ненаправленные треугольники. Мы могли бы распространить определение на направленные сети, но это зависит от видов треугольников, которые имеют отношение к конкретному случаю. Например, в Twitter нас могут интересовать треугольники, которые сокращают пути, по которым распространяется информация. Рассмотрим сценарий на рис. 2.11(b): если **a** подписан на **b**, а **b** подписан на **c**, то **a**, возможно, заинтересован подписаться на **c**, чтобы получать доступ к сообщениям с напрямую, а не через ретвиты **b**. В таком сценарии мы, возможно,

захотим подсчитывать только направленные треугольники, которые кодируют эти виды сокращений. В данной книге мы работаем с коэффициентом кластеризации только в ненаправленных сетях; в случае же направленных сетей мы можем просто игнорировать направление связей и трактовать их при расчете коэффициента кластеризации так, как если бы они были ненаправленными.

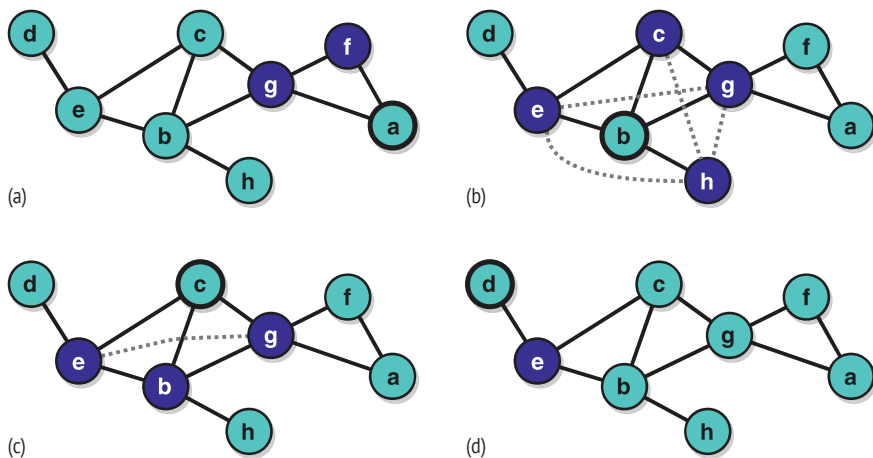


Рис. 2.12 Примеры коэффициента кластеризации. (а) Узел **a** имеет двух соседей **f** и **g**, которые соединены, образуя треугольник. (б) Узел **b** имеет четырех соседей **c**, **e**, **g** и **h**. Две из шести пар соседей соединены, образуя два из шести возможных треугольников. Пропущенные треугольные соединения показаны пунктирными серыми линиями. (в) Узел **c** имеет трех соседей **e**, **b** и **g**, образующих два из трех возможных треугольников. (д) Узел **d** имеет одного единственного соседа **e**, следовательно, возможные треугольники отсутствуют, и коэффициент кластеризации не определен

Усредняя коэффициент кластеризации по всем узлам, мы можем рассчитать коэффициент кластеризации для всей сети целиком. Низкий коэффициент кластеризации (близкий к нулю) означает, что в сети мало треугольников, тогда как высокий коэффициент кластеризации (близкий к единице) означает, что в сети много треугольников. Социальные сети имеют большой коэффициент кластеризации; присутствует значительная часть всех возможных треугольников. Например, сети соавторства, как правило, имеют коэффициент кластеризации выше 0.5. Простой механизм объясняет обилие треугольников в социальных сетях: мы знакомимся с людьми через общие контакты, таким образом замыкая треугольники. Этот механизм, именуемый *триадическим замыканием*, обсуждается далее в главе 5. Онлайн-овые социальные сети делают рекомендации, основываясь на триадическом замыкании. Например, Facebook рекомендует «людей, которых вы, возможно, знаете», основываясь на общих друзьях, а Twitter рекомендует аккаунты, на которые подписаны ваши друзья (на чьи аккаунты подписаны вы). Эти рекомендации приводят к высокой кластеризации.

В табл. 2.1 представлен коэффициент кластеризации для различных сетей. Мы наблюдаем высокую кластеризацию во многих, но не во всех случаях. Сеть кинофильмов и кинозвезд имеет $C = 0$. Это обусловлено тем, что указанная сеть является двудольной, и, следовательно, не может быть треугольников; треугольники потребовали бы связей между парами кинофильмов или кинозвезд, которых нет в двудольной сети. Если мы вместо этого возьмем социальную сеть кинозвезд, то мы находим высокий коэффициент кластеризации. Ретвитная сеть в Twitter тоже имеет низкий $C = 0.03$. В целях понимания причины учтите, что если Боб ретвитит Алису, а Чарли ретвитит Боба, то Твиттер связывает и Боба, и Чарли с изначальным автором, Алисой. Следовательно, каждое ретвитно-каскадное дерево выглядит как звезда. Единственные треугольники возникают из-за пользователей, которые участвуют в нескольких звездах.

Библиотека NetworkX имеет функции для подсчета треугольников и расчета коэффициента кластеризации для узлов и сетей. В настоящее время NetworkX устанавливает коэффициент кластеризации равным нулю для узлов со степенью ниже двух и включает эти узлы в расчет среднего.

<code>nx.triangles(G)</code>	# узел в виде словаря -> число треугольников
<code>nx.clustering(G, node)</code>	# коэффициент кластеризации для узла
<code>nx.clustering(G)</code>	# узел в виде словаря -> # коэффициент кластеризации
<code>nx.average_clustering(G)</code>	# коэффициент кластеризации для сети

2.9. Резюме

В этой главе мы узнали о нескольких признаках сетей: ассортативности, связности (соединенности), коротких путях и кластеризации.

1. Ассортативность – это корреляция между шансом, что два узла соединены, и их сходством. Сходство может измеряться на основе степени, содержимого, местоположения, актуальных интересов или любого другого свойства узла. Ассортативность в социальных сетях может обуславливаться гомофилией, склонностью похожих людей быть связанными; или социальным влиянием, склонностью связанных людей быть похожими.
2. Пути – это последовательности связей, соединяющих узлы в сети. Естественная мера расстояния между двумя узлами определяется как число связей, проходимых кратчайшим соединительным путем. Самый простой способ отыскать короткий путь – использовать алгоритм поиска сперва в ширину. Понятия путей и расстояний могут быть расширены за счет учета направления и веса связей.

3. Дерево – это связная ненаправленная сеть с как можно меньшим числом связей. В деревьях не существует циклов.
4. Связные компоненты – это такие подсети, в которых между любыми двумя узлами в одной и той же компоненте существует путь, но не между двумя узлами в разных компонентах. В направленных сетях мы различаем сильно и слабо связные компоненты, основываясь на соответствии путей направлениям связей.
5. Средняя длина пути сети определяется путем усреднения длин кратчайших путей по всем парам узлов в связной сети. Если сеть не является связной, то обычно рассматриваются пары узлов только в одной и той же компоненте.
6. Большинство реально существующих сетей в среднем имеет очень короткие пути. Это называется маломировым свойством. Популярное представление о том, что социальные сети имеют шесть степеней сепарации, возникло в результате эксперимента Милграма.
7. Локальная кластеризация сети обуславливается наличием треугольников или связных триад. Для узла коэффициент кластеризации измеряет долю треугольников из максимально возможного числа. Для всей сети мы можем усреднить коэффициент кластеризации по всем узлам. Социальные сети имеют высокую кластеризацию из-за треугольников типа «друг моего друга».

2.10. Дальнейшее чтение

Слово «гомофилия» происходит от греческих *homós* (то же самое) и *philia* (дружба). Указанная концепция была сформулирована Лазарсфельдом и соавт. (1954), а наличие различных форм гомофилии наблюдалось во многих исследованиях социальных сетей (Макферсон и соавт., 2001). Айелло и соавт. (2012) обнаружили, что пользователи со схожими интересами с большей вероятностью станут друзьями на различных платформах онлайн-социальных сетей и что сходство между пользователями на основе метаданных их профиля является предсказателем социальных связей. Коэффициенты соединенности и ассортативности на основе k ближайших соседей были введены соответственно Пастором-Саторрасом и соавт. (2001) и Ньюманом (2002).

Исследователи все чаще изучают негативные последствия гомофилии. Знакомство с новостями и информацией через фильтр единомышленников в онлайн-социальных сетях может способствовать появлению кластерных сообществ, где наше внимание сосредоточено на информации, которую мы, скорее всего, уже знаем или с которой согласны. Эти так называемые эхокамеры (Санштайн, 2001) и фильтерные пузыри, как утверждается, являются патологическими послед-

ствиями рекомендательных алгоритмов социальных сетей (Паризер, 2011) и приводят к поляризации (Коновер и соавт., 2011b) и вирусной дезинформации (Лазер и соавт., 2018).

Алгоритмы отыскания кратчайших путей и связанных компонент в сетях имеют сложную историю. Изобретение поиска сперва в ширину приписывается Зузе и Берку в отвергнутой в 1945 году докторской диссертации и независимо Муру (1959). Существует два известных алгоритма отыскания кратчайших путей в взвешенных сетях: один принадлежит Дейкстре (1959), а другой назван в честь изобретателей алгоритмом Беллмана–Форда, опубликованный независимо Шимбелом (1955), Фордом-мл. (1956), Муром (1959) и Беллманом (1958).

Эксперимент Милграма (Трэверс и Милграм, 1969) был повторен Доддсом и соавт. (2003) с использованием электронной почты. Бэкстром и соавт. (2012) обнаружили, что средняя длина кратчайшего пути в сети дружеских связей в Facebook меньше пяти. Ньюман (2001) впервые изучил структуру научно-коллаборационных сетей.

Доступное введение в сети и их маломировую и кластерную структуру предлагает Уоттс (2004). Существование треугольников в сетях также называется транзитивностью (Холланд и Лейнхардт, 1971). Раннее определение коэффициента кластеризации сети было сформулировано Люсом и Перри (1949), тогда как локальное определение, используемое в этой книге, принадлежит Уоттсу и Строгацу (1998).

Концепция триадического замыкания была представлена в основополагающей статье Грановеттера (1973) и обсуждается в главе 5. Изучая данные с социально-медийной платформы, Венг и соавт. (2013a) подтвердили, что триадическое замыкание оказывает сильное влияние на формирование связей, но также обнаружили, что основанные на трафике сокращения являются еще одним ключевым фактором в объяснении новых связей.

Упражнения

- 2.1 Ознакомьтесь с учебным материалом главы 2 в репозитории GitHub книги¹.
- 2.2 Вспомните, что, если не указано иное, длина пути – это число содержащихся в нем связей. Имея два узла в произвольном ненаправленном связном графе, между ними должен существовать какой-то кратчайший путь. Истина или ложь: может существовать несколько таких кратчайших путей.

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

- 2.3 Истина или ложь: для любых двух узлов в (ненаправленном) дереве существует ровно один путь между этими двумя узлами.
- 2.4 Рассмотрите ненаправленную связную сеть с N узлами. Какое минимальное число связей может быть в сети? Если мы не требуем, чтобы сеть была связной, изменится ли это минимальное число связей?
- 2.5 Вспомните, что дерево из N узлов содержит связи. Истина или ложь: любая связная ненаправленная сеть из N узлов и связей должна быть деревом.
- 2.6 Истина или ложь: любая ненаправленная сеть из N узлов по меньшей мере с N связями должна содержать цикл.
- 2.7 Истина или ложь: любая направленная сеть из N узлов по меньшей мере с N связями должна содержать цикл.
- 2.8 Рассмотрите сеть, определенную матрицей смежности в уравнении (1.11). Существуют ли какие-либо циклы в этой сети? Является ли она сильно связной? Слабо связной?
- 2.9 Рассмотрите невзвешенную ненаправленную версию сети, определенную матрицей смежности в уравнении (1.11). Является ли эта сеть деревом?
- 2.10 Рассмотрите невзвешенную ненаправленную версию сети, определенную матрицей смежности в уравнении (1.11). Каков диаметр этой сети?
- 2.11 Если конвертировать слабо связную направленную сеть в ненаправленную сеть, будет ли связна результирующая сеть? Объясните, почему да или почему нет.
- 2.12 Рассмотрите произвольную неполную ненаправленную сеть. Теперь добавьте одну связь. Как изменилось число узлов в гигантской компоненте этой сети в результате добавления?
- а. Оно строго уменьшилось.
 - б. Оно уменьшилось или осталось прежним.
 - в. Оно увеличилось или осталось прежним.
 - г. Оно строго увеличилось.
- 2.13 Рассмотрите взвешенную направленную сеть на рис. 2.13. Что из перечисленного ниже наиболее точно описывает связность этой сети?
- а. Сильно связная.
 - б. Слабо связная.
 - в. Разъединенная (несвязная).
 - г. Ничего из перечисленного выше.

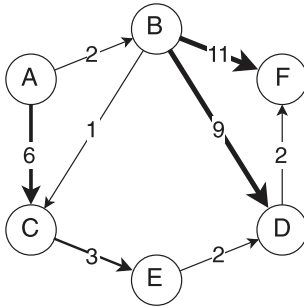


Рис. 2.13 Взвешенная направленная сеть. Числа показывают веса связей

- 2.14** Рассмотрите взвешенную направленную сеть на рис. 2.13. Какова сила-на-входе узла **D**? Какова сила-на-выходе узла **C**? (Вспомните определения из главы 1.)
- 2.15** Сколько узлов находится в самой крупной сильно связной компоненте сети на рис. 2.13?
- 2.16** Рассмотрите сеть на рис. 2.14. Что из перечисленного ниже наиболее точно описывает связность этой сети?
- Сильно связная.
 - Слабо связная.
 - Разъединенная (несвязная).
 - Ничего из перечисленного выше.

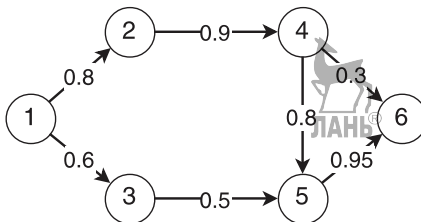


Рис. 2.14 Взвешенная направленная сеть. Числа обозначают веса связей

- 2.17** Веса связей могут представлять все, что касается взаимоотношения между узлами: силу связи, географическое расстояние, напряжение, протекающее по соединительному кабелю, и т. д. При обсуждении длин путей в взвешенном графе необходимо сначала определить принцип, по которому веса увязываются с расстояниями. Длина пути между двумя узлами, таким образом, равна сумме расстояний между связями в этом пути. Простейший случай возникает, когда веса связей представляют расстояние. Взгляните на сеть рис. 2.14 и предположите, что веса связей представляют расстояния. Если использовать эту метрику расстояния, то каков кратчайший путь между узлами 1 и 6?

- 2.18** Распространенным способом определения расстояния между двумя узлами является величина, обратная (или реверсная) весу связи. Взгляните на сеть рис. 2.14 и предположите, что расстояние между двумя соседними узлами определяется как величина, обратная весу связи. Если использовать эту метрику расстояния, то каков кратчайший путь между узлами 1 и 6?
- 2.19** Рассмотрите сеть на рис. 2.15. Какое из следующих ниже значений является наилучшей оценкой диаметра этой сети?
- 2.
 - 4.
 - 10.
 - 20.

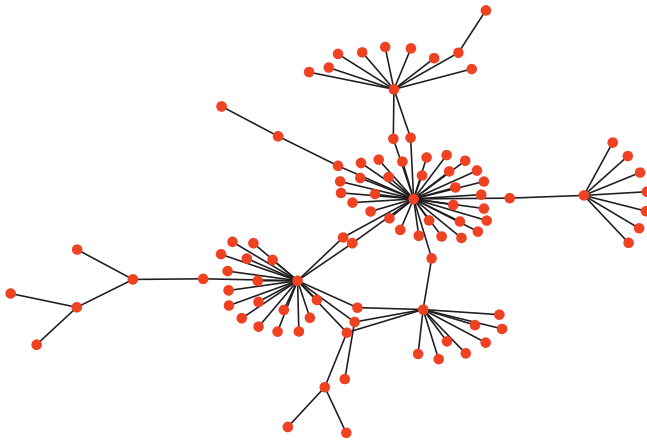


Рис. 2.15 Малая подсеть сети белковых взаимодействий *Drosophila melanogaster* (так называемой плодовой мушки). Каждый узел представляет белок, который взаимодействует с другими белками для выполнения основной работы клетки. Экспериментальные данные показали, что связанные белки образуют прочную межмолекулярную связь для выполнения некоторой биологической функции

- 2.20** Рассмотрите сеть на рис. 2.15. Какое из следующих ниже значений является наилучшей оценкой среднего коэффициента кластеризации для этого графа?
- 0.05.
 - 0.5.
 - 0.75.
 - 0.95.
- 2.21** Будет ли социальная сеть, вероятно, иметь диаметр и коэффициент кластеризации графа на рис. 2.15?
- 2.22** Рассмотрите сеть на рис. 2.16. Что из перечисленного ниже наиболее точно описывает связность этой сети?
- Сильно связная.

- b. Слабо связная.
- c. Разъединенная (несвязная).
- d. Ничего из перечисленного выше.

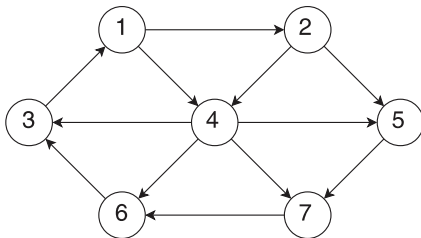


Рис. 2.16 Направленная сеть

- 2.23** Каков диаметр сети на рис. 2.16?
- 2.24** Рассмотрите ненаправленную версию сети на рис. 2.16. Каков диаметр этой сети?
- 2.25** Рассмотрите любой произвольный направленный граф D вместе с его ненаправленной версией G . Истина или ложь: если средняя длина кратчайшего пути и диаметр направленного графа существуют, то они могут быть меньше, чем у ненаправленной версии.
- 2.26** Представьте, что вы строите конкурента библиотеке NetworkX. Вы уже написали метод `shortest_path()` для вычисления кратчайшего пути между двумя узлами, и теперь вы хотите написать функцию для вычисления диаметра сети. Что из следующего лучше всего описывает, как это сделать?
- a Сначала вычислить длину кратчайшего пути между каждой парой узлов. Диаметр – это минимум из этих значений.
 - b Сначала вычислить длину кратчайшего пути между каждой парой узлов. Диаметр – это среднее из этих значений.
 - c Сначала вычислить длину кратчайшего пути между каждой парой узлов. Диаметр – это максимум из этих значений.
 - d Сначала вычислить среднюю длину всех путей между каждой парой узлов. Диаметр – это минимум из этих значений.
- 2.27** Истина или ложь: диаметр сети всегда больше или равен ее средней длине пути.
- 2.28** Какова центральная идея, лежащая в основе понятия «шесть степеней сепарации»?
- a. Социальные сети имеют высокие коэффициенты кластеризации.
 - b. Социальные сети разрежены.

- с. Социальные сети имеют много высокостепенных узлов.
d. Социальные сети имеют малую среднюю длину пути.
- 2.29 У Американского математического общества есть веб-инструмент для отыскания *коллаборационного расстояния* между двумя математиками (см. вставку 2.3). Используйте этот инструмент, чтобы вычислить число Эрдеша для нескольких математиков в вашем учебном заведении или тех, кого вы знаете понаслышке.
- 2.30 Используйте *Оракул Бейкона* (oracleofbacon.org) чтобы измерить расстояние кратчайшего пути в сети кинозвезд среди как можно большего числа пар малоизвестных актеров и актрис, о которых вы можете подумать. Постройте гистограмму, показывающую распределение длин кратчайших путей, а также оцените среднюю длину пути на основе вашей выборки. (Если вы не знакомы с гистограммами, то они определены в следующей главе.)
- 2.31 Играйте в «Вики-игру» (thewikigame.com) до тех пор, пока не сможете успешно завершить несколько раундов. Сообщите среднюю длину (число кликов) обнаруженных путей.
- 2.32 Каков максимальный коэффициент кластеризации для узла в произвольном ненаправленном графе?
- 2.33 Каков максимальный коэффициент кластеризации для узла в дереве?
- 2.34 Вспомните определение эгосети в разделе 1.4. Взгляните на эго-сеть рис. 2.17: каков коэффициент кластеризации для эго?

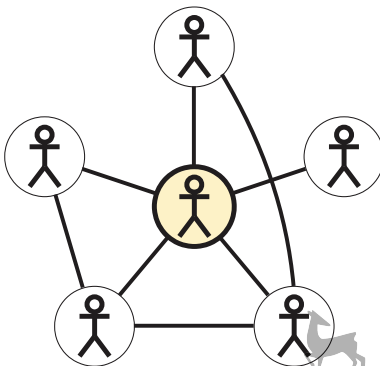


Рис. 2.17 Эгосеть. Эго выделено желтым цветом

- 2.35 Рассмотрим ненаправленную сеть на рис. 2.4. Вычислите длину кратчайшего пути для каждой пары узлов в гигантской компоненте.

- 2.36 Рассмотрите ненаправленную сеть на рис. 2.4. Вычислите такой коэффициент кластеризации для каждого узла, что он был бы определенным.
- 2.37 Рассмотрите пример сети на рис. 2.12. Вычислите длину кратчайшего пути для каждой пары узлов и среднюю длину кратчайшего пути для сети.
- 2.38 Рассмотрите пример сети на рис. 2.12. Вычислите коэффициент кластеризации для каждого узла, такой что он определен, а также для сети.
- 2.39 Если вы используете онлайн-социальную сеть, такую как Facebook или LinkedIn, измерьте свой коэффициент кластеризации в сети. (*Подсказка 1*: если вы используете социальную сеть с направленными связями, такую как Twitter или Instagram, то вы можете трактовать связи как ненаправленные.) (*Подсказка 2*: это может занять продолжительное время; можно сделать оценку, основываясь на небольшой выборке ваших друзей.)
- 2.40 Какие из следующих, казалось бы, противоречивых свойств присущи социальным сетям?
- Социальные сети имеют короткие пути, но крупный диаметр.
 - Социальные сети имеют малый диаметр, но крупную среднюю длину пути.
 - Социальные сети имеют много высокостепенных узлов, но все же разъединены.
 - Социальные сети сильно кластеризованы, но не являются плотными.
- 2.41 Сеть socfb-Northwestern25 в репозитории книги на GitHub представляет собой снимок сети Facebook Северо-Западного университета. Узлы – это анонимные пользователи, а связи – дружеские отношения. Загрузите эту сеть в граф NetworkX, чтобы ответить на следующие ниже вопросы. Убедитесь, что используете надлежащий класс графа для ненаправленной невзвешенной сети.
- Сколько узлов и связей в этой сети?
 - Что из перечисленного ниже лучше всего описывает связность этой сети?
 - Сильно связная.
 - Слабо связная.
 - Связная.
 - Разъединенная (несвязная).
 - Мы хотим получить некоторое представление о средней длине путей в этой сети, но в подобного рода крупных сетях нередко вычислять кратчайший путь между каждой парой узлов обходится вычислительно слишком дорого. Если бы мы хотели вычислить кратчайший путь между каждой парой уз-

лов в этой сети, то сколько бы вычислений кратчайшего пути потребовалось? Другими словами, сколько пар узлов в этой сети? (*Подсказка*: вспомните, что эта сеть является ненаправленной, и мы обычно игнорируем самонаправленные циклы, в особенности при вычислении путей.)

4. В целях экономии времени давайте попробуем подход с взятием выборки. Получить случайную пару узлов можно с помощью следующей ниже инструкции:

```
random.sample(G.nodes, 2)
```

Поскольку эта выборка выполняется без возврата, это предотвращает повторный выбор одного и того же узла. Сделайте это 1000 раз и для каждой такой пары узлов запишите длину кратчайшего пути между ними. Возьмите среднее значение этой выборки, чтобы получить оценку средней длины пути в этой сети. Сообщите свою оценку с прецизионностью до одного десятичного знака.

5. Внесите небольшое изменение в описанную выше процедуру, чтобы оценить диаметр сети. Сообщите приближенный диаметр.
6. Каков средний коэффициент кластеризации для этой сети? Ответьте с прецизионностью по меньшей мере в два знака после точки.
7. Является ли эта сеть ассортативной или дисассортативной? Ответьте на этот вопрос, используя два показанных в этой главе метода. Отличаются ли ответы?





Хаб: центр, вокруг которого вращаются другие вещи или из которого они исходят; концентратор, центр деятельности, власти, торговли, транспорта и т. д.

Если вы путешествовали на самолете, то пересекли важную сеть – сеть авиационных перевозок. На рис. 0.7 мы нанесли на карту сеть авиационных перевозок США: узлы представляют аэропорты, а связи – прямые рейсы между ними. Хотя большинство аэропортов довольно небольшие, в нескольких крупных (например, в Атланте, Чикаго, Денвере) ежедневно совершаются рейсы в сотни или даже тысячи пунктов назначения. В социальных сообществах точно так же существуют индивидуумы, которые гораздо заметнее и влиятельнее, чем другие; а во Всемирной паутине существует несколько очень популярных веб-сайтов, таких как google.com, тогда как большинство веб-сайтов большинству людей неизвестно.

Эти примеры иллюстрируют ключевой признак многих сетей: *гетерогенность* (или *разнородность*). Гетерогенные сети представляют широкую вариабельность свойств и ролей их элементов – узлов и/или связей. Это отражает разнообразие, присутствующее в сложных системах, описываемых сетями. В сетях авиационных перевозок, социальных сетях, Всемирной паутине и многих других сетях очевидным источником гетерогенности является степень узлов: несколько узлов имеет много соединений (Атланта, Google, Обама), тогда как большинство узлов имеет их мало.

Важность узла или связи оценивается путем вычисления его *центральности*. Существует несколько способов измерения центральности сети. В этой главе мы введем несколько важных мер центральности, в частности, для узлов. Как мы обсудим ниже, степень является важной мерой центральности. Высокостепенные узлы называются *хабами*. Как оказалось, хабы отвечают за некоторые поразительные свойства, которые характеризуют широкий спектр сетей.

3.1. Меры центральности

3.1.1. Степень

В главе 1 мы узнали, что степень узла – это число соседей этого узла. В примере сети аэропортов США на рис. 0.7 степень узла (аэропорта) – это число других аэропортов, до которых можно добраться из него прямыми рейсами.

В социальной сети степень узла (индивидуума) – это число социальных связей, соединяющих узел с другими. Например, в сети соавторства, такой как та, что изображена на рис. 2.8, степень – это число соавторов. Высокостепенные узлы в социальных сетях – это люди с многочисленными соединениями – будь то потому, что они общительны либо востребованы, либо просто стремятся к сотрудничеству, эти узлы кажутся в некотором смысле важными. Следовательно, степень является очень естественной мерой центральности в социальных сетях.

Средняя степень сети показывает, насколько в среднем узлы соединены. Как мы увидим позже (раздел 3.2), средняя степень может и не отражать фактическое распределение значений степеней. Это тот случай, когда узлы имеют гетерогенные степени, как во многих реально существующих сетях.

3.1.2. Близость

Измерить центральность узла можно еще одним способом, а именно определив, насколько он «близок» к другим узлам. Это можно сделать путем суммирования расстояний от узла до всех остальных. Если расстояния в среднем невелики, то их сумма невелика, и мы говорим, что узел имеет высокую центральность. Это приводит к определению центральности на основе близости, которая является просто величиной, обратной сумме расстояний узла от всех остальных.

Центральность узла i на основе близости определяется по формуле:

$$g_i = \frac{1}{\sum_{j \neq i} \ell_{ij}}, \quad (3.1)$$

где ℓ_{ij} – это расстояние от i до j , и суммирование выполняется по всем узлам сети, за исключением самого i . Альтернативная формулировка получается путем умножения g_i на константу $N - 1$, т. е. просто на число членов в знаменателе суммы:

$$\tilde{g}_i = (N - 1)g_i = \frac{N - 1}{\sum_{j \neq i} \ell_{ij}} = \frac{1}{\sum_{j \neq i} \frac{\ell_{ij}}{N} - 1}. \quad (3.2)$$

Благодаря этому мы дисконтируем размер графа и делаем меру сопоставимой по разным сетям. Поскольку важность имеет не фактическое значение g_i , а его ранг по сравнению с центральностью других узлов на основе близости, относительная центральность узлов остается такой же, как при использовании уравнения (3.1), поскольку ранг не изменяется, если значения умножаются на константу. Выражение $\sum_{j \neq i} \ell_{ij} / (N - 1)$ представляет собой среднее расстояние от фокального узла i до остальной сети. Таким образом, мы обнаруживаем, что близость может выражаться эквивалентно как величина, обратная среднему расстоянию.

В библиотеке NetworkX есть функция для вычисления центральности на основе близости:

```
nx.closeness centrality(G, node) # центральность узла на основе близости
```

3.1.3. Промежуточность

Многие происходящие в сетях явления основаны на диффузионных процессах (глава 7). Примеры включают передачу информации через социальную сеть, перевозку товаров через порт и распространение эпидемий в сети физических контактов между индивидуумами популяции. Это навело на мысль о третьем понятии центральности, именуемом *промежуточностью*: узел тем более централен, чем чаще он участвует в этих процессах.

Естественно, центральность на основе промежуточности имеет разную имплементацию для каждого отдельного типа диффузии. Самая простая и популярная имплементация рассматривает простой процесс, в котором сигналы передаются от каждого узла к каждому другому узлу по кратчайшим путям. Этот подход часто используется в транспортных сетях для оценивания трафика, обрабатываемого узлами, исходя из того, что число проходящих через узел кратчайших путей является хорошим приближением частоты использования узла. Тогда центральность оценивается путем подсчета числа пересечения узла этими путями. Чем выше это число, тем больше трафика контролируется узлом, который, следовательно, более влиятелен в сети.

Учитывая два узла, в сети может существовать более одного кратчайшего пути между ними, при этом все они будут иметь одинаковую длину. Например, если узлы X и Y не соединены друг с другом, но имеют двух общих соседей S и T , то существуют два отличимых кратчайших пути длиной два, проходящих из X в Y : $X - S - Y$ и $X - T - S$. Обозначим через σ_{hj} суммарное число кратчайших путей из h в j , а через $\sigma_{hj}(i)$ число этих кратчайших путей, проходящих через узел i . Промежуточность узла i определяется по формуле:

$$b_i = \sum_{h \neq j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}. \quad (3.3)$$

В уравнении (3.3) сумма проходит по всем парам вершин h и j , отличным от i и друг от друга. Если ни один кратчайший путь между h и j не пересекает i [$\sigma_{hj}(i) = 0$], то вклад пары (h, j) в промежуточность узла i равен 0. Если все кратчайшие пути между h и j пересекают i [$\sigma_{hj}(i) = \sigma_{hj}$], то вклад равен 1. Если узел является листовым (т. е. у него есть только один сосед), его нельзя пересечь никаким путем. Следовательно, его промежуточность равна нулю. Поскольку потенциальный вклад поступает от всех пар узлов, расстояние между ними растет вместе с размером сети.

Давайте рассмотрим пример на рис. 3.1(а). Для узла 1 единственной парой узлов, которая имеет кратчайший путь, проходящий через этот узел, является (2, 4). Однако между 2 и 4 есть два кратчайших пути одинаковой длины: другой путь проходит через узел 3, а не 1. Следовательно, промежуточность узла 1 равна 1/2. Далее рассмотрим узел 3. Кратчайший путь между тремя парами узлов (1, 5), (2, 5) и (4, 5) будет проходить через 3. Как мы отметили ранее, существует два эквивалентных кратчайших пути между узлами 2 и 4, только один из которых проходит через 3, внося 1/2 в сумму. Итоговая сумма дает центральность на основе промежуточности, равную 3.5 для узла 3. Остальные узлы 2, 4 и 5 не имеют кратчайших путей, проходящих через них кратчайших путей, следовательно, их промежуточность равна нулю.

Узел имеет высокую промежуточность, если он занимает такое особое положение в сети, при котором он является важной станцией для коммуникационных шаблонов, проходящих через сеть. Для того чтобы это случилось, не обязательно иметь много соседей. Как правило, мы наблюдаем корреляцию между степенью узла и его промежуточностью, вследствие которой хорошо соединенные узлы имеют высокую промежуточность и наоборот (рис. 3.1(а)). Однако существует много исключений. Узлы, соединяющие мостом разные участки сети, в типичной ситуации имеют высокую промежуточность, даже если их степень является низкой, как показано на рис. 3.1(б).

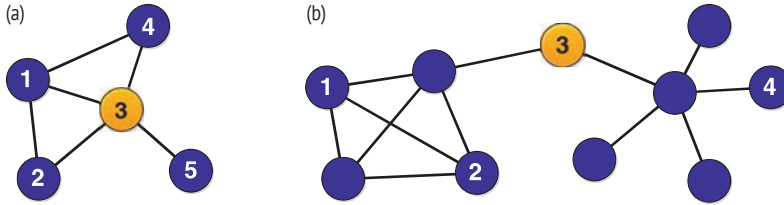


Рис. 3.1 Иллюстрации центральности узла на основе промежуточности. (а) Оранжевый узел имеет высокую степень ($k_3 = 4$), а также высокую промежуточность ($b_3 = 3.5$). (б) Оранжевый узел имеет низкую степень ($k_3 = 2$), но сохраняет сеть связной, действуя как единственный мост между узлами в двух подсетях. Например, кратчайший путь между узлами 1 и 2 не проходит через оранжевый узел, но путь между 1 и 4 проходит. Фактически все кратчайшие пути между четырьмя узлами в одной подсети и пятью узлами в другой подсети проходят через оранжевый узел. Следовательно, его промежуточность равна $b_3 = 4 \times 5 = 20$

Указанная концепция имеет прямое расширение на связи. Центральность связи на основе промежуточности – это доля кратчайших путей среди всех возможных пар узлов, проходящих через эту связь. Связи с очень высокой центральностью на основе промежуточности нередко присоединяются к когезивным участкам сети, именуемым *сообществами*. Следовательно, промежуточность может использоваться для локализации и удаления этих связей, что допускает возможность сепарации и в последствии выявления сообществ (глава 6).

Центральность на основе промежуточности зависит от размера сети. Если мы хотим сравнить центральность узлов или связей в разных сетях, то значения промежуточности должны нормализоваться.

Для промежуточности узлов максимальное число путей, которые могут проходить через узел i , равно числу пар узлов, исключая сам i . Это выражается уравнением $\binom{N-1}{2} = \frac{(N-1)(N-2)}{2}$. Следовательно, нормализованная промежуточность узла i получается путем деления b_i в уравнении (3.3) на этот коэффициент.

Библиотека NetworkX имеет функции для вычисления центральности узлов и связей на основе нормализованной промежуточности:

```
nx.betweenness centrality(G)      # узлы в форме словаря ->
                                # центральность на основе промежуточности
nx.edge_betweenness centrality(G) # связи в форме словаря ->
                                # центральность на основе промежуточности
```

3.2. Распределения значений центральности

До появления онлайн-социальных медиа социальные сети, которые можно было изучать, обычно строились посредством персональных собеседований и опросов, в которых не могло участвовать очень много людей в разумные сроки. Как следствие, сети состояли всего из нескольких десятков узлов. В таких малых сетях имеет смысл различать отдельные узлы и задавать такие вопросы, как «какой самый важный узел сети?». В настоящее время мы обрабатываем гораздо более крупные графы. Например, в социальной сети дружеских связей Facebook участвует два миллиарда индивидуумов, в том числе многие известные люди, такие как известные художники, спортивные знаменитости, политики, бизнесмены и ученые среди прочих. Однако независимо от того, насколько они популярны, каждый из них может быть соединен только с малой порцией всей сети в целом.

В целях более глубокого понимания того, как центральность распределяется среди многочисленных узлов в крупных сетях, нам необходимо использовать *статистический* подход. Благодаря ему мы можем сосредотачиваться на классах узлов и связей, делящих между собой схожие признаки, а не на отдельных элементах сети. Например, мы можем сгруппировать все узлы, имеющие одинаковые значения центральности на основе степени. Статистическое распределение меры центральности говорит нам о том, сколько элементов – узлов или связей – имеет определенное значение центральности для всех возможных значений. На рис. 3.2 показано, например, распределение степени узла в малой сети. В крупных сетях это полезный инструмент для выявления классов элементов: inspectируя распределение, мы можем видеть наличие или отсутствие заметных значений или групп значений и классифицировать элементы соответствующим образом.

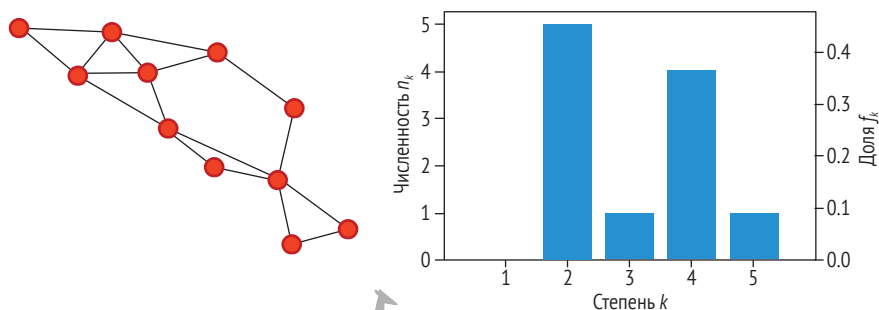


Рис. 3.2 Гистограммное представление степенного распределения малой сети. Сначала генерируется список со степенью каждого узла. Высоты столбиков гистограммы задаются численностью n_k узлов с каждой степенью k . Относительная частота встречаемости f_k определяется как доля всех узлов со степенью k . Также показаны значения f_k

Размах распределения также показывает гетерогенность элементов сети по отношению к определенной интересующей мере центральности: например, если степень узла охватывают многие порядки величины, от единиц до миллионов, то сеть очень гетерогенна по степени. Как мы увидим, такая гетерогенность влияет как на структуру сети, так и на ее функционирование.

Во вставке 3.1 даются определения распределений вероятностей и показано, как их рассчитывать. В целях инспектирования распределений вероятностей мер центральности в реально существующих сетях давайте сосредоточимся на двух системах: сети пользователей Twitter и сети математических статей в «Википедии» (en.wikipedia.org/wiki/Category:Mathematics). В сети Twitter узлы – это пользователи, а прямая связь от пользователя Алисы к пользователю Бобу указывает на то, что Боб ретвитит (часть) контента, первоначально транслируемого Алисой. Узлы «Википедии» – это страницы, а связи – это гиперсвязи, ведущие с одной страницы на другую. Обе сети являются направленными. Рассматриваемая нами сеть Twitter насчитывает 18 470 узлов и 48 365 связей (средняя степень-на-входе 2.6). Сеть «Википедии» насчитывает 15 220 узлов и 194 103 связей (средняя степень-на-входе 12.8). Такие низкие значения средней степени, по сравнению с размером системы, указывают на то, что обе сети разрежены (т. е. очень мало пар узлов соединено связями). Это распространенная черта многих реально существующих сетей (табл. 1.1).

Давайте сосредоточимся на распределении значений центральности на основе степени. На рис. 3.3 мы показываем кумулятивное степенное распределение обеих сетей (вставка 3.1). Кривые охватывают

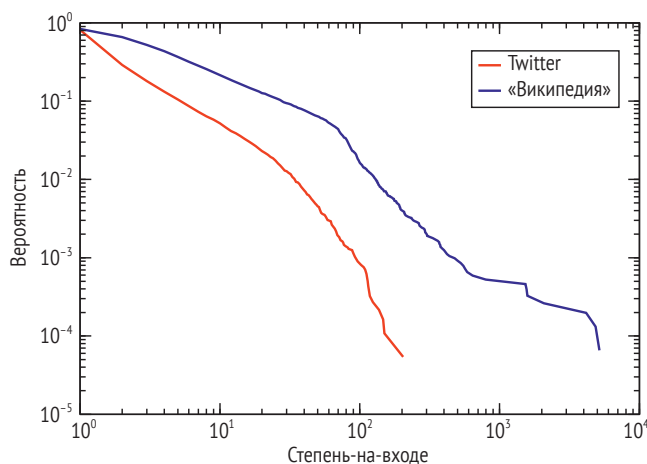


Рис. 3.3 Кумулятивные степенные распределения сетей Twitter и «Википедии», показанные на двойном логарифмическом графике (лог-лог графике). Оба графа являются направленными, поэтому мы показываем распределения значений степени-на-входе. Максимальная степень-на-входе равна 204 для Twitter и 5171 для «Википедии». Кривые построены с использованием логарифмической шкалы, поскольку они охватывают несколько порядков величины

Вставка 3.1

Статистические распределения

Гистограмма или *распределение* величины (например, меры центральности) есть функция, которая подсчитывает численность наблюдений (например, узлов), имеющих разные значения величины. Если интересующая величина является дискретной (например, целое число), то для каждого значения v мы подсчитываем число n_v наблюдений, имеющих это значение. Таким образом, сумма наблюдений n_v по всем значениям равна общему числу наблюдений: $\sum_v n_v = N$. Результат выводится на графике в виде серии поочередных столбиков, по одному для каждого значения, высота которого равна n_v .

В целях сравнения гистограмм разных наборов наблюдений принято делить n_v на суммарное число наблюдений N , порождая *относительную частоту* $f_v = n_v/N$. Сумма всех относительных частот равна 1 независимо от числа наблюдений. Для степени узла нормализация получается путем деления на суммарное число узлов (рис. 3.2). Относительная частота f_v тогда равна доле узлов со степенью v .

В пределе бесконечного числа наблюдений f_v сходится к *вероятности* p_v того, что наблюдение принимает значение v . В этом пределе гистограмма становится *распределением значений вероятности*. Любая реально существующая сеть имеет бесконечное число узлов и связей, поэтому невозможно достичь бесконечного предела, а гистограмма является лишь приближением распределения значений вероятности. Однако если сеть достаточно велика, к примеру, имеет миллионы узлов, то для практических целей мы можем рассматривать ее как распределение вероятностей.

Хотя некоторые меры центральности, такие как степень узла, и принимают целочисленные значения, у других это не так. Например, значения центральности на основе промежуточности не обязательно являются целыми числами. В этих случаях вместо подсчета наблюдений для конкретных значений мы можем подразделять диапазон значений на непересекающиеся интервалы, или *корзины*. Затем мы можем подсчитывать число наблюдений, попадающих в каждую корзину, аналогичным образом. Такой метод группирования в корзины можно использовать всякий раз, когда нас интересуют диапазоны значений, даже если значения являются целыми числами. Например, гистограмма индивидуального благосостояния может подсчитывать численность индивидуумов, которые имеют годовой доход внутри определенных рамок, например 0–50k долл., 50–100k долл., 100–200k долл. и т. д.

Комплементарная кумулятивная функция распределения, или просто *кумулятивное распределение* $P(x)$ переменной величины, дает вероятность того, что наблюдение имеет значение больше x . Для вычисления $P(x)$ мы суммируем относительные частоты (или вероятности) всех значений переменной величины справа от значения x : $P(x) = \sum_{v \geq x} f_v$. Кумулятивное распределение нередко используется, когда диапазон изменчивости очень широк, как в случае с несколькими мерами центральности в реально существующих гетерогенных сетях. Поскольку высокие значения переменных величин встречаются редко, стандартное распределение имеет шумный хвост. Совокупное распределение эффективно усредняет этот шум.

несколько порядков величины. В таких случаях говорят, что распределения являются *широкими* или имеют *тяжелый хвост*, где хвост – это правая часть распределения, достигающая наибольших значений переменной. Когда мера имеет такой широкий диапазон изменчивости, принято использовать кумулятивное распределение. Кроме того, тяжелохвостные распределения эффективнее изображаются на графике в двойной *логарифмической шкале*, или *логарифмически-логарифмической шкале* (вставка 3.2), как мы сделали на рис. 3.3, чтобы иметь возможность урегулировать форму распределения в разных порядках величины.

Вставка 3.2

Логарифмическая шкала

При построении кривой, включающей очень малые и очень большие значения на одной либо обеих осях, разницы между малыми значениями неразличимы. Решение состоит в построении графика в *логарифмической шкале*: вместо использования в качестве координат оси изначальных значений мы используем их *логарифмы*. Благодаря этому большой размах значений, охватывающий многие порядки величины, может быть представлен эффективно: малые разницы усиливаются в диапазоне малых значений, а крупные разницы сжимаются в диапазоне крупных значений. Мы используем логарифмическую шкалу для построения графиков тяжелохвостных распределений мер центральности сети. Поскольку как значения центральности, так и значения вероятностей охватывают несколько порядков величины, логарифмическая шкала используется как по оси x , так и по оси y . Мы называем такие диаграммы *двойными логарифмическими графиками*, или графиками с логарифмической шкалой по обоим координатным осям.

Тяжелохвостные степенные распределения демонстрируют большую гетерогенность значений степеней: в то время как многие узлы имеют всего несколько соседей, некоторые другие имеют много соседей, что придает им заметную роль в сети. Эти узлы называются хабами. Многие природные, социальные, информационные и искусственные сети имеют тяжелохвостное степенное распределение с высокосвязными хабами. Измерить широту степенного распределения можно путем вычисления *параметра гетерогенности*, который сравнивает изменчивость степени по всем узлам со средней степенью.

В целях формального определения параметра гетерогенности k (греческая буква «каппа») степенного распределения сети нам нужно ввести среднюю квадратическую степень $\langle k^2 \rangle$, которая является средним значением квадратов степеней:

$$\langle k^2 \rangle = \frac{k_1^2 + k_2^2 + \dots + k_{N-1}^2 + k_N^2}{N} = \frac{\sum_i k_i^2}{N}. \quad (3.4)$$

Параметр гетерогенности может быть определен как отношение между средней квадратической степенью и квадратом средней степени сети [уравнение (1.5)]:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle^2}. \quad (3.5)$$

В случае нормального или узкого распределения с резким пиком в некотором значении, к примеру k_0 , распределение квадратных степеней концентрируется вокруг k_0^2 . Следовательно $\langle k^2 \rangle \approx k_0^2$ и $\langle k \rangle \approx k_0$, порождая $\kappa \approx 1$. В случае тяжелохвостного распределения с той же средней степенью k_0 $\langle k^2 \rangle$ имеет взрывной рост из-за большой степени хабов, так что $\kappa \gg 1$.

Если степенное распределение концентрируется вокруг типичного значения, то гетерогенности нет, и параметр гетерогенности обычно близок к единице¹. Если вместо этого степенное распределение является широким, то параметр гетерогенности сильно завышен самыми крупными степенями хабов и может принимать крупные значения. Чем больше хабов, тем больше гетерогенность. Как мы увидим, гетерогенность играет ключевую роль в структуре сети и в динамике некоторых протекающих в ней процессов.

Если сеть является направленной, как наши графы «Википедии» и Twitter, то мы должны рассматривать два распределения: распределение значений *степени-на-входе* и распределение значений *степени-на-выходе*, определяемые как вероятность того, что случайно выбранная вершина имеет соответственно заданную степень-на-входе или степень-на-выходе. В этом случае определение хаба может относиться либо к степени-на-входе, либо к степени-на-выходе. Например, веб-страница может иметь много других связанных с ней страниц (высокая степень-на-входе), но сама она может быть связана всего с несколькими страницами (низкая степень-на-выходе), или наоборот. В нескольких направленных сетях эти две меры *коррелированы*, поэтому узлы с большой (малой) степенью-на-входе также имеют большую (малую) степень-на-выходе. Мы вернемся к обсуждению степени в направленных, а также взвешенных сетях в главе 4. В табл. 3.1 приведено несколько базовых чисел, характеризующих степенные распределения в различных сетях².

¹ Альтернативное определение в литературе сравнивает параметр гетерогенности не с 1, а с $\langle \kappa \rangle$.

² Наборы данных для этих сетей доступны в репозитории книги на GitHub: github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

Таблица 3.1. Базовые переменные, характеризующие степенные распределения в примерах различных сетей: средняя степень, максимальная степень и параметр гетерогенности. Сети приводятся такие же, как в табл. 1.1 и 2.1, их число узлов и связей тоже показано. В случае направленных сетей мы сообщаем максимальную степень-на-входе, а параметр гетерогенности вычисляется на распределении значений степени-на-входе

Сеть	Узлы (N)	Связи (L)	Средняя степень ($\langle k \rangle$)	Максимальная степень ($\langle k_{\max} \rangle$)	Параметр гетерогенности (κ)
Facebook, Северо-Западный университет	10 567	488 337	92.4	2105	1.8
IMDB, кинофильмы и кинозвезды	563 443	921 160	3.3	800	5.4
IMDB, кинозвезды, снимавшиеся вместе	252 999	1 015 187	8.0	456	4.6
Twitter, политика США	18 470	48 365	2.6	204	8.3
Электронная почта компании Энрон	87 273	321 918	3.7	1338	17.4
Статьи по математике в «Википедии»	15 220	194 103	12.8	5171	38.2
Интернет-маршрутизаторы	190 914	607 610	6.4	1071	6.0
Авиационные перевозки в США	546	2781	10.2	153	5.3
Авиационные перевозки по всему миру	3179	18 617	11.7	246	5.5
Взаимодействие дрожжевых белков	1870	2277	2.4	56	2.7
Мозг <i>C. elegans</i>	297	2345	7.9	134	2.7
Экологическая пищевая паутина Everglades	69	916	13.3	63	2.2

Конечно, можно анализировать распределения и других свойств, помимо степени. Оказывается, что степень обычно коррелирует с другими мерами центральности. Поэтому хабы обычно по рангу входят в число наиболее центральных узлов по отношению к различным критериям. Есть и исключения. Как мы увидели на рис. 3.1, узел может иметь большую центральность на основе промежуточности, если он соединяет разные области сети, независимо от того, имеет он сам высокую степень или нет.

На рис. 3.4 мы показываем кумулятивные распределения значений промежуточности наших сетей. Как и распределения значений степени, они тоже охватывают несколько порядков величины.

Хабы, когда они присутствуют, являются единственным наиболее важным признаком сети. Они являются столпами его структуры и движущими силами протекающих в нем процессов. В следующих далее разделах мы представим некоторые замечательные последствия наличия хабов.

3.3. Парадокс дружбы

Предположим, что вы ищете человека, у которого наибольшее число друзей среди группы из N человек, по которым у вас есть только справочник телефонных номеров. Если вы просто позвоните по одному из выбранному наугад номеров, то шанс, что вы выбрали правильного человека, составляет $1/N$. Что, если вы спросите его об одном из его друзей? Внешне это может выглядеть, что вы просто отбираете еще

одного индивидуума наугад, как и раньше, и что вероятность того, что этот друг – нужный человек, одинакова. Но это не так. Для того чтобы понять причину, давайте рассмотрим малую социальную сеть на рис. 3.5. Самым соединенным индивидуумом является Том, у которого четверо друзей. Если вы спросите случайного индивидуума, то существует одна возможность из семи, что вы доберетесь до Тома. Однако если вы отберете случайного друга случайного индивидуума, то вероятность того, что вы натолкнетесь на Тома, оказывается равной $5/21 \approx 24\%$, что существенно больше, чем $1/7 \approx 14\%$. Мы приходим к выводу, что легче отыскивать людей через их друзей, чем путем случайного поиска. Но почему?

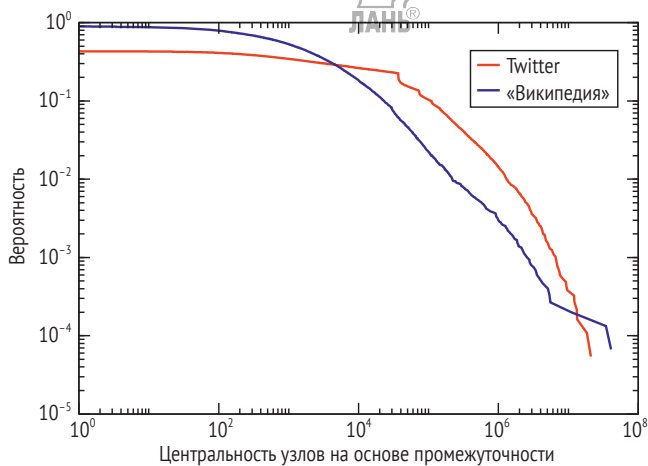


Рис. 3.4 Кумулятивное распределение значений промежуточности как мера центральности узлов для Twitter и «Википедии», показанное на двойном логарифмическом графике. Мы рассматривали обе сети как ненаправленные. В случае «Википедии» мы вычисляли расстояние только для ее гигантской компоненты, которая включает в себя более 98 % узлов. Граф Twitter является связным

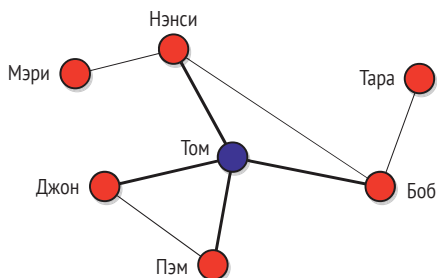


Рис. 3.5 Парадокс дружбы. Путем отбора случайной связи вместо случайного узла Тома можно «отыскать» гораздо легче, чем Мэри, потому что у него четыре друга (Джон, Пэм, Боб и Нэнси), тогда как у нее только один (Нэнси). Во время случайного следования по соединениям гораздо вероятнее натолкнуться на хаб, чем на низкостепенный узел. Это основная причина, по которой у наших друзей в среднем больше друзей, чем у нас

Грубо говоря, если у кого-то много друзей, то у него гораздо больше шансов быть упомянутым кем-то, чем если бы у него было всего несколько. Обращение к чьим-то друзьям на самом деле означает выбор не узлов, а связей. Когда мы предпочитаем узлы, каждый из них имеет одинаковую вероятность быть выбранным независимо от их степени. Когда мы предпочитаем связи, чем крупнее число соседей узла, тем выше вероятность того, что он будет достигнут. В нашей сети, показанной на рис. 3.5, есть четыре возможных канала, ведущих к Тому, и поэтому достигнуть его гораздо проще, чем Мэри или Тары, у которых есть только один друг.

Шансы попасть в хаб увеличиваются, если вы переходите из круга соседей в круг соседей соседей и т. д. Это обусловлено тем, что число прослеживаемых связей с каждым шагом увеличивается, и поэтому становится вероятнее, что одна из них прикреплена к хабу. Это свойство может использоваться в наших интересах. Существует много ситуаций, в которых выявление хабов сети бывает полезным. Например, во время эпидемической вспышки индивидуумы с наибольшим числом контактов являются потенциальными крупными распространителями, и было бы важно их изолировать и/или вакцинировать, чтобы сдержать заболевание. В таком сценарии можно было бы отбирать людей случайно и связываться с некоторыми из их друзей, так как у них более высокая вероятность быть хабами, чем у пула отобранных индивидуумов. Мы вернемся к этой теме в главе 7.

Разница в отборе именно связей, а не узлов имеет еще одно специфическое последствие. Давайте выберем действующее лицо в нашей сети, скажем, Нэнси. У нее есть три друга: Боб, Мэри и Том. У них в общей сложности есть $3 + 1 + 4 = 8$ друзей, что дает в среднем $8/3$. Если мы повторим этот расчет для всех остальных узлов, то обнаружим, что среднее число соседей соседей узла равно $17/6 = 2.83$. Однако средняя степень сети равна $(1 + 3 + 3 + 1 + 4 + 2 + 2)/7 = 16/7 = 2.29$. Такая ситуация является типичной: средняя степень соседей узла больше, чем средняя степень узла. Другими словами, у наших друзей в среднем больше друзей, чем у нас. Это и называется *Парадоксом дружбы*.

Наш пример помогает раскрыть происхождение указанного парадокса. Когда мы вычисляем среднюю степень узла, степень каждого узла появляется в сумме только один раз. С другой стороны, когда мы вычисляем среднюю степень соседей узла и повторяем процедуру для всех узлов, каждый узел будет появляться столько раз, сколько его степень в частичных суммах. В нашем примере степень Тома будет засчитана четыре раза, потому что он находится в списке друзей четырех человек. Это подстегивает значение средней степени соседей, которая в конечном итоге превышает среднюю степень. Таким образом, Парадокс дружбы обусловлен процедурой *отбора*. Два средних значения вычисляются путем отбора степеней узлов по-разному: равномерно для средней степени, пропорционально степени для средней степени соседей.

Чем шире степенное распределение, тем сильнее эффект Парадокса дружбы. Когда все узлы имеют примерно одинаковую степень, два значения похожи друг на друга. В сетях с тяжелохвостными распределениями, как показано на рис. 3.3 (и как в типичных социальных сетях), этот эффект очень заметен из-за сверхсвязных хабов.

3.4. Ультрамалые миры

Хабы сети, как мы только что убедились, не только легко отыскивать; они также пользуются большим спросом. Если мы хотим передать сигнал от одного узла сети к другому по кратчайшему маршруту, то сигнал, скорее всего, будет проходить через один хаб или больше. Вы, вероятно, испытывали это на себе во время своих авиаперелетов: когда вы хотите лететь из аэропорта **A** в аэропорт **B**, а прямого рейса между **A** и **B** нет, то вы вынуждены сделать хотя бы одну пересадку в каком-нибудь хабовом аэропорту **C**. Во многих случаях достаточно одной пересадки, поэтому для поездки из пункта **A** в пункт **B** требуется всего два рейса: $A \rightarrow C$ и $C \rightarrow B$.

В главе 2 мы увидели, что многие реально существующие сети являются *малыми мирами* (т. е. можно переходить из каждого узла в любой другой узел за малое число шагов). В сети с хабами мы ожидаем, что среднее расстояние между любыми двумя узлами будет короче по сравнению с сетью с таким же числом узлов и связей, но без хабов. Фактически сети с широким степенным распределением часто обладают так называемым свойством *ультрамалого мира*, указывающим на то, что расстояния между узлами очень малы. На рис. 3.6 мы по-

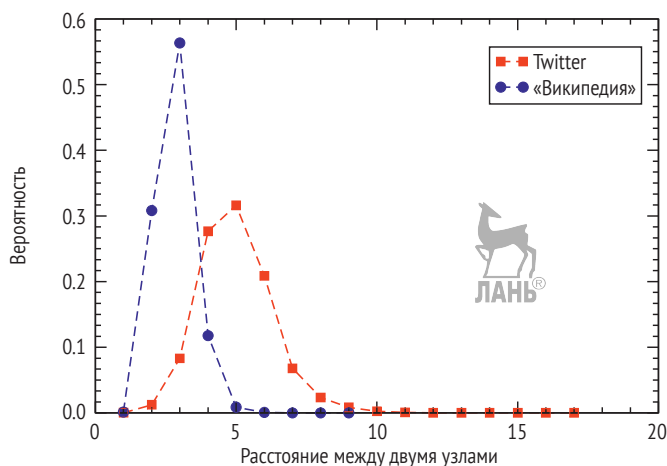


Рис. 3.6 Ультрамалые миры. Распределение значений расстояния между узлами достигает пика при очень низких значениях как для Twitter, так и для «Википедии». Это обусловлено наличием хабов, которые сжимают расстояние между большинством пар узлов, поскольку через них проходят кратчайшие пути. Расстояния вычисляются путем игнорирования направления связей



строили график распределения значений расстояния между любыми двумя узлами для опорных сетей, которые мы использовали: Twitter и «Википедию». Оба распределения имеют сильные пики, поэтому между расстояниями очень мало изменчивости (вариабельности). Пиковые значения чрезвычайно малы по сравнению с размерами систем (пять для Twitter, три для «Википедии»), что указывает на то, что обе сети являются ультрамалыми мирами. Это отличительный признак многих реально существующих сетей.

3.5. Устойчивость

Система является *устойчивой*, или робастной, если отказ некоторых ее компонент не влияет на ее функционирование. Например, самолет продолжает лететь, если один из его двигателей перестает работать. В целом устойчивость зависит от того, какие компоненты отказывают, и от степени повреждения.

Как определить устойчивость сети? Узлы могут описывать широкий спектр сущностей, таких как люди, маршрутизаторы, белки, нейроны, веб-сайты и аэропорты. В таком высокоуровневом представлении не просто определить отказ узла, который зависит от конкретного типа сети. Но, если мы допустим, что узел каким-то образом перестает работать, мы сможем осведомиться о том, как меняется структура и, следовательно, функционирование сети без этого узла и всех его связей.

В главе 2 мы дали определение тому, что для сети означает быть связной – все узлы доступны друг из друга. Мы также увидели, что если сеть не является связной, то она имеет две или более связных компоненты. Связность является важным свойством сети, которое обычно влияет на ее функционирование. Если бы интернет не был связным графом, то было бы невозможно отправлять сигналы (например, электронные письма) между маршрутизаторами, принадлежащими разным компонентам. Следовательно, один из способов определить и измерить устойчивость сети состоит в наблюдении за тем, как удаление узла и его связей влияет на связность системы (рис. 3.7). Если система останется связной, то мы можем допустить, что в какой-то степени она будет продолжать работать нормально. Однако разложение сети на разъединенные части будет сигнализировать о серьезном повреждении, которое может ставить под угрозу ее функционирование.

Стандартный тест устойчивости сетей состоит в проверке влияния на связность сети постепенного удаления все большего числа узлов вместе со всеми смежными им связями. В целях оценивания числа отказов после удаления узла ученые вычисляют относительный размер гигантской компоненты (т. е. отношения числа узлов в гигантской компоненте к числу узлов, изначально присутствующих в сети). Давайте предположим, что первоначальная сеть является связной.

В этом случае гигантская компонента совпадает со всей сетью, поэтому ее относительный размер равен единице. Если удаление подмножества узлов не разлагает ее на разъединенные части, то доля узлов в гигантской компоненте просто уменьшается на долю удаленных узлов. Однако если удаление узла разлагает сеть на две или более связных компонент, то размер гигантской компоненты может существенно уменьшиться. По мере того как доля удаленных узлов будет приближаться к единице, несколько оставшихся узлов, вероятно, будут распределяться между крошечными компонентами, поэтому доля узлов в гигантской компоненте близка к нулю.

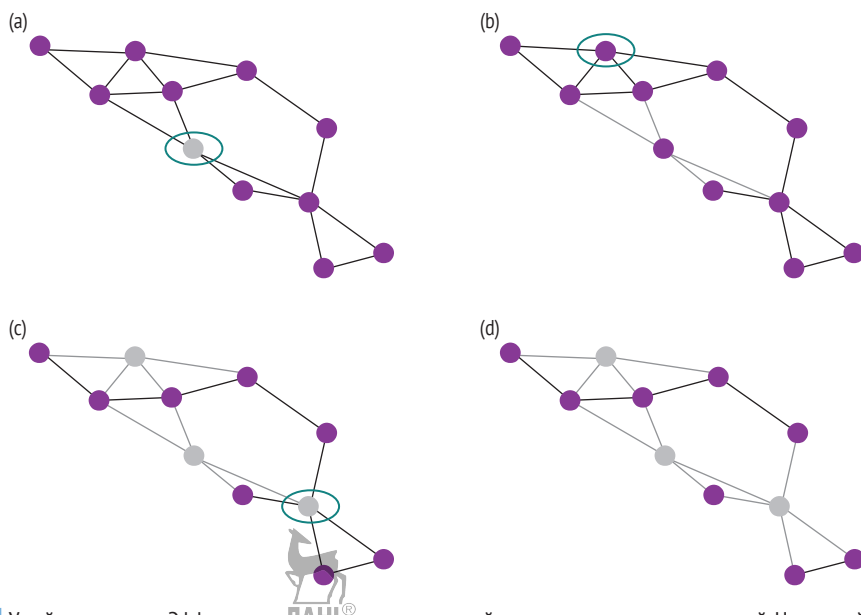


Рис. 3.7 Устойчивость сети. Эффект последовательности удалений узлов и их инцидентных связей. На каждой диаграмме удаленный узел выделен кружком. Удаленные узлы и их инцидентные связи окрашены в серый цвет. После удаления трех узлов (d) сеть распадается на три разъединенные друг от друга компоненты

На рис. 3.8 показаны результаты тестов устойчивости всемирной сети OpenFlights. Когда узлы удаляются случайным образом, процесс симулирует *случайный отказ* элементов сети. Мы наблюдаем, что относительный размер гигантской компоненты уменьшается очень медленно. Это обусловлено наличием хабовых узлов, которые поддерживают структуру связной. До тех пока существует достаточное число хабов, система остается в значительной степени связной. Поскольку мы удаляем узлы случайным образом, вероятность отказа хаба невелика, так как они статистически редки по сравнению с другими узлами. На графике также показано, что происходит, когда узлы удаляются в порядке убывания их степени (т. е. сначала мишенью являются хабы). В этом случае система почти сразу же испытывает серьезные отказы

и полностью фрагментируется, когда устраняется около 20 % узлов. Нацеливание на высокостепенные узлы является примером *атаки*, поскольку цель состоит в нанесении максимального ущерба путем удаления центральных узлов. Мы заключаем, что многие реально существующие сети, имеющие центральные хабы, довольно устойчивы к случайным отказам, но довольно уязвимы для атак.

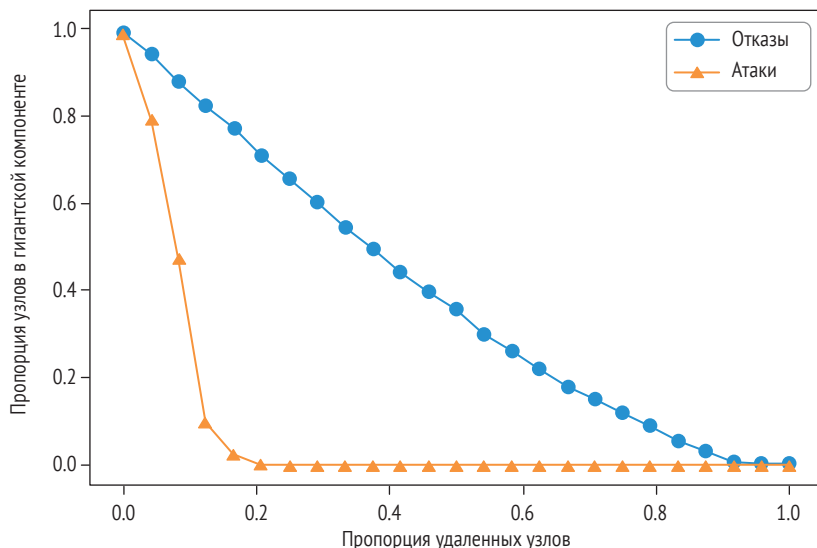


Рис. 3.8 Устойчивость сети. Доля узлов в гигантской компоненте в зависимости от доли узлов, удаленных из всемирной сети OpenFlights. Мы показываем, что произойдет, если узлы удаляются случайным образом (случайные отказы) либо если приоритизируются в зависимости от степени (нацеленные атаки)

3.6. Разложение ядра

Мы кратко упомянули *структуру ядро–периферия* многих сетей в разделе 2.1. При анализе или визуализации крупной сети часто бывает полезно сосредотачиваться на ее более плотной части (ядре).

Степень каждого узла может использоваться для сепарирования сети на четко отличимые части, именуемые *оболочками*, основываясь на их положении в структуре ядро–периферия сети. Низкостепенные внешние оболочки соответствуют периферии. По мере того как они удаляются или отслаиваются, остается все более плотная внутренняя подсеть, *ядро*. Мы начинаем с узлов-одиночек (нуль-степенных узлов), если таковые имеются. Затем удаляем все узлы со степенью один. Как только их не останется, мы начнем удалять узлы со степенью два и т. д. Последняя группа подлежащих удалению узлов – это самое внутреннее ядро.

Формально алгоритм k -ядерного разложения начинается с установления $k = 0$. Затем он продолжается итеративно. Каждая итерация соответствует значению k и состоит из нескольких простых шагов.

1. Рекурсивно удалить все узлы степени k до тех пор, пока их больше не останется.
2. Удаленные узлы составляют k -оболочку, а остальные узлы составляют $(k + 1)$ -ядро, потому что все они имеют степень $k + 1$ или больше.
3. Если в ядре не осталось узлов, то завершить работу; в противном случае увеличить k на единицу для следующей итерации.

Разложение ядра имеет особую пользу на практике для отфильтровки периферийных узлов при визуализации крупных сетей. На самом деле большинство рисунков в главе 0 не изображает все сети целиком, а только их части, полученные путем исключения некоторой периферии. Например, оболочки $k = 1$ и $k = 2$ были удалены в сети политических ретвитов, показанной на рис. 0.3. Этот процесс фильтрации показан на рис. 3.9.

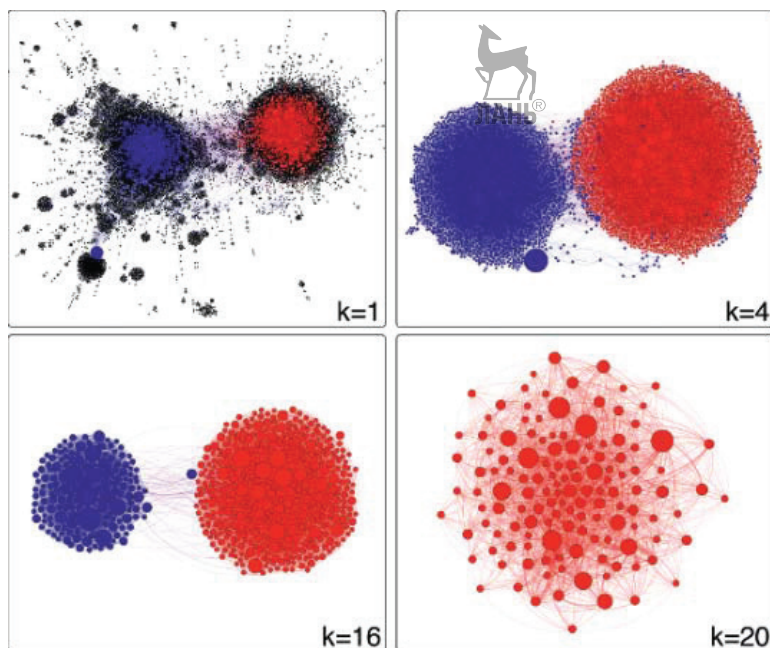



Рис. 3.9 Фильтрация путем k -ядерного разложения. Мы начинаем с полной сети политических ретвитов в Twitter ($k = 1$). По мере увеличения k периферийные узлы удаляются, а оставшееся ядро становится меньше и плотнее. Самое внутреннее ядро содержит только красные узлы, соответствующие консервативным учетным записям; каждый из них имеет по меньшей мере $k = 20$ соседей

Библиотека NetworkX имеет функции для разложения ядра:



```

nx.core_number(G) # вернуть словарь с числом ядра для каждого ядра
nx.k_shell(G,k)   # подсеть, индуцированная узлами в k-оболочке
nx.k_core(G,k)    # подсеть, индуцированная узлами в k-ядре
nx.k_core(G)      # самая внутренняя (с максимальной степенью) ядерная подсеть
  
```

3.7. Резюме

В этой главе мы узнали о разных мерах центральности сетевых узлов и ребер и сосредоточились на степени узлов как важной мере, которая выявляет хабы. Ниже приведено несколько понятий, которые необходимо запомнить.

1. Степень узла определяется как число связей в графе, инцидентных узлу.
2. Промежуточность узла выражает частоту, с которой он пересекается распространяющимися по сетям сигналами, следующими кратчайшими путями.
3. В крупных сетях необходимо использовать статистические инструменты анализа глобальных признаков сети. Гистограмма обеспечивает наглядную иллюстрацию распределения значений данного атрибута узлов или связей (например, степени). Нормализованная гистограмма представляет собой оценку вероятностного распределения значений интересующей меры.
4. Распределения значений мер центральности являются гетерогенными для многих реально существующих сетей (т. е. они охватывают несколько порядков величины). В частности, распределение значений степени нередко имеет утяжеленный хвост. Узлы с крупной степенью называются хабами, или концентраторами.
5. Парадокс дружбы гласит, что в социальной сети у ваших друзей в среднем больше друзей, чем у вас. Это обусловлено с высокой вероятностью выбора хабов среди соседей узла.
6. Хабы играют решающую роль в структуре и динамике сети. Например, они сокращают расстояния между узлами и делают сеть устойчивой к случайным отказам, но уязвимой для целенаправленных атак.
7. Мы можем раскладывать сеть, чтобы выявлять ее структуру ядро-периферия. Это достигается путем итеративной отфильтровки оболочек низкостепенных узлов и сосредоточении на оставшихся, все более и более плотных ядрах.

3.8. Дальнейшее чтение

Центральность на основе близости была введена Бавеласом (1950). Фримен (1977) ввел меру промежуточности, а Брандес (2001) разработал алгоритм, широко принятый на вооружение для ее вычисления. Промежуточность связи, введенная в неопубликованном техническом отчете Антониссе, хорошо описана Гирваном и Ньюманом (2002), которые применили эту меру для обнаружения и удаления связей, соединяющих сетевые сообщества друг с другом, чтобы иметь возможность сепарировать и выявлять последние (раздел 6.3.1). Статистические распределения хорошо представлены в книге Фридмана и соавт. (2007)

Доступное введение в сети и их хабовую структуру предлагается в работе Барабаши (2003). Альберт и соавт. (1999) обнаружили первую крупную сеть с тяжелохвостным степенным распределением, а именно граф Всемирной паутины, Веб-граф. Впоследствии было установлено, что многие другие реально существующие сети обладают тем же свойством (Барабаши, 2016).

Парадокс дружбы был раскрыт Фелдом (1991). Ультрамалые миры были открыты Коэном и его коллегами (Коэн и Хавлин, 2003; Коэн и соавт., 2002, 2003). Первое исследование устойчивости сетей появилось в статье Альберта и соавт. (2000). Коэн и соавт. (2000, 2001) являются авторами классических теоретических исследований по устойчивости.

Применение разложения k -ядра для визуализации сети осуществлялось Батагелем и соавт. (1999), Бауром и соавт. (2004) и Бейро и соавт. (2008).

Упражнения

- 3.1 Ознакомьтесь с учебным материалом главы 3 в репозитории GitHub книги¹.
- 3.2 Предположим, что у вас есть граф со 100 узлами и 200 связями. Какова средняя степень узлов в этой сети?
- 3.3 Рассмотрите сеть, образованную 250 студентами в общежитии. Связи в этой сети представляют отношения соседей по комнате: два узла соединены, если они в настоящее время являются соседями по комнате. В этом общежитии комнаты в основном двухместные, но есть несколько трехместных и четырехместных комнат.

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

1. Является ли этот граф связным?
 2. Какова мода (наиболее частое значение) степенного распределения узлов?
 3. Сколько узлов находится в самой большой клике?
 4. Ожидаете ли вы, что в этом графе будут какие-либо хабы?
- 3.4** Как отыскать узел в сети с наибольшей центральностью на основе степени в библиотеке NetworkX? И как получить степень этого узла?
- 3.5** Предположим, что у вас есть NetworkX-граф G сотрудников. Имена узлов являются идентификаторами сотрудников, и узлы имеют атрибуты для полного имени, отдела, должности и заработной платы. Что из следующего ниже даст вам зарплату для сотрудника с ИД 5567?
- a. `G.node(5567)('salary')`
 - b. `G[5567]['salary']`
 - c. `G.node[5567]['salary']`
 - d. `G(5567)('salary')`
- 3.6** У вас есть NetworkX-граф G , и вы собираетесь нарисовать его с помощью следующей инструкции: `nx.draw(G, node_size=node_size_list)`. Какой из следующих ниже способов является правильным для получения списка `node_size_list`, чтобы узлы получали размеры в соответствии с их степенью?
- a. `node_size_list = [G[n] for n in G.nodes]`
 - b. `node_size_list = G.degree()`
 - c. `node_size_list = [G.degree() for n in G.nodes]`
 - d. `node_size_list = [G.degree(n) for n in G.nodes]`
 - e. `node_size_list = [d for d in G.degree()]`
- 3.7** Академическая коллаборационная сеть, или сеть научного сотрудничества, – это один из видов социальной сети. В такой сети узел со степенью два означает, что:
- a) ученый является соавтором статьи с другим ученым;
 - b) ученый является соавтором публикаций с двумя другими учеными;
 - c) ученый является автором двух публикаций;
 - d) публикация была написана в соавторстве двумя учеными.
- 3.8** Что из перечисленного ниже можно было бы ожидать в отношении степеней узлов социальной сети?
- a. Большинство узлов соединено с одним единственным крупным хабом.
 - b. Можно найти разнообразие степеней.
 - c. Все узлы имеют более или менее одинаковую степень.
 - d. Все узлы имеют очень высокую степень.

- 3.9 Каким свойством должна обладать сеть, чтобы иметь четко определенную центральность на основе близости?
- 3.10 Приведите примеры сетей, таких что:
- 1) узел с наивысшей степенью не является узлом с наибольшей близостью;
 - 2) узел с наибольшей промежуточностью не является узлом с наибольшей близостью.
- 3.11 Рассмотрите сеть на рис. 3.10, чтобы ответить на следующие ниже несколько вопросов. На каждый вопрос, в случае наличия нескольких кандидатов, отвечайте всеми связанными верхними узлами.
1. Какой узел имеет наибольшую центральность на основе степени?
 2. Какой узел имеет наибольшую центральность на основе промежуточности?
 3. Какой узел имеет наибольшую центральность на основе близости?

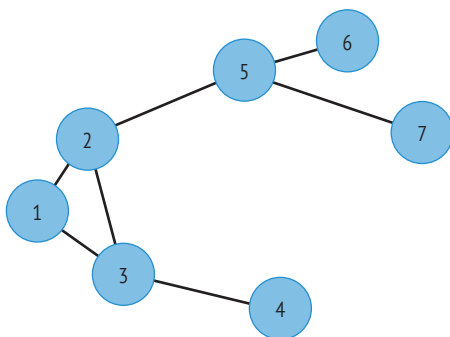


Рис. 3.10 Ненаправленная невзвешенная сеть

- 3.12 Предположим, что мы хотим построить связную сеть с 10 узлами и средней степенью 1.8, такую что параметр гетерогенности является малым. Как выглядит такой граф?
- 3.13 Для каждой из следующих ниже переменных укажите, ожидаете ли вы увидеть тяжелохвостное распределение или нет, и почему.
1. Размер обуви взрослых людей в Великобритании.
 2. Доход домохозяйства в США.
 3. Степень узла в социальной сети Twitter.
 4. Парное расстояние в сети «Википедии».
- 3.14 Если бы рост людей подчинялся тяжелохвостному распределению, то удивились ли вы, увидев на улице человека ростом 30 футов (9 м)?



3.15 График на рис. 3.11 взят из исследования 200 млн веб-страниц и 1.5 млрд связей между ними (Бродер и соавт., 2000). Это двойной логарифмический график числа страниц (ось y) с заданным числом связей (ось x).

1. Приблизительно на скольких страницах есть только одна другая страница, которая связана с ними?
2. Приблизительно на скольких страницах есть связи с 10 другими страницами?
3. Приблизительно на скольких страницах есть связи с 100 другими страницами?

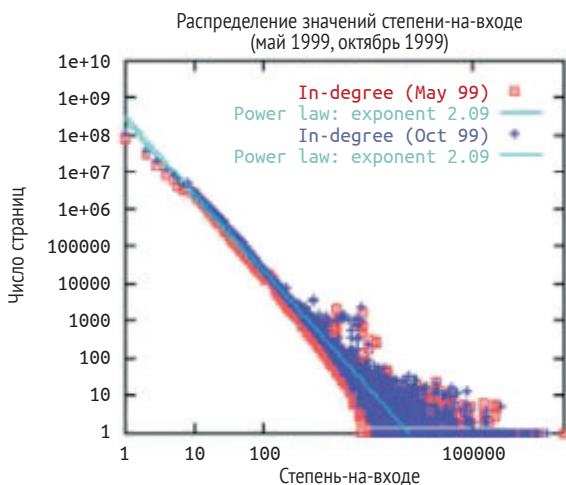


Рис. 3.11 Гистограмма степени-на-входе для Всемирной паутины в двойной логарифмической шкале. Перепечатано из Бродера и соавт. (2000) с разрешения издательства Elsevier

3.16 Рассмотрите социальную сеть, в которой связь представляет собой сексуальные отношения. Прочитайте отчет Лильероса и соавт. (2001) об исследовании такой сети, основанном на выборке из 4781 шведов. (Если у вас нет доступа к журналу через ваше учреждение, то вы можете скачать препринт статьи по адресу <https://arxiv.org/abs/cond-mat/0106507>.) Какова максимальная степень в этой сети? Что она значит? Если рассматривать подсети с узлами, которые соответствуют мужчинам и женщинам, то будут ли они иметь одинаковое распределение значений степени? Почему да или почему нет?

3.17 Обычно слово «хаб» используется в повседневной речи для описания аэропортов, обслуживающих большое число маршрутов (прямые рейсы). Загрузите сеть полетов OpenFlights в США в граф библиотеки NetworkX, чтобы ответить на следующие ниже вопросы.

1. Каково среднее число маршрутов, обслуживаемых каждым аэропортом в этой сети?

2. Назовите пять крупнейших аэропортов по числу маршрутов?
 3. Сколько аэропортов в этой сети обслуживают только один маршрут?
 4. Какой аэропорт имеет самую высокую центральность на основе близости?
 5. Какой аэропорт имеет самую высокую центральность на основе промежуточности?
 6. Вычислите параметр гетерогенности этой сети.
- 3.18** Загрузите математическую сеть «Википедии» в оргграф библиотеки NetworkX, чтобы ответить на следующие ниже вопросы.
1. Вычислите среднюю степень-на-входе и среднюю степень-на-выходе этой сети. Что вы заметили? Почему?
 2. Какой узел имеет самую высокую степень-на-входе?
 3. Какой узел имеет самую высокую степень-на-выходе?
 4. Что больше на этом графе: максимальная степень-на-входе либо максимальная степень-на-выходе? Ожидаете ли вы, что она будет той же самой для других графов Всемирной паутины? Почему?
 5. Вычислите параметр гетерогенности для распределения значений степени-на-входе этого графа.
 6. Вычислите параметр гетерогенности для распределения значений степени-на-выходе этого графа.
- 3.19** Напишите функцию Python, которая принимает граф библиотеки NetworkX и имя узла и возвращает среднюю степень соседей этого узла. Используйте эту функцию, чтобы вычислить эту величину для каждого узла в сети OpenFlights US и возьмите среднее значение. Имеет ли здесь место Парадокс дружбы (т. е. средняя степень ближайших соседей больше, чем средняя степень узла)?
- 3.20** Существуют ли такие сети, в которых среднее число соседей соседей узла совпадает со средней степенью? Если существуют, то каким свойством они должны обладать?
- 3.21** Являются ли сети с тяжелохвостным степенным распределением уязвимее для случайных или целенаправленных атак? А что скажете насчет решетчатых сетей аналогичного размера?
- 3.22** Если кто-то пытается разрушить сеть путем удаления узлов и/или ребер, пытаясь ее разъединить и/или увеличить среднюю длину пути, то очевидной стратегией является атака на хабы. Что из перечисленного является еще одним вредным критерием для отбора мишеней? Объясните свой ответ.
- a. Узлы с высоким коэффициентом кластеризации.
 - b. Узлы с низкой степенью.
 - c. Узлы с высокой центральностью на основе близости.



- d. Узлы/ребра с высокой центральностью на основе промежуточности.

3.23 Рассмотрите два узла одинаковой степени в некоторой сети: один с высоким коэффициентом кластеризации и один с низким коэффициентом кластеризации. Какой из двух вариантов при прочих равных условиях, по вашему мнению, был бы лучшей мишенью, если бы вы стремились разрушить сеть?

3.24 Сеть socfb-Northwestern25 в репозитории книги на GitHub является снимком сети Facebook Северо-Западного университета. Узлы – это анонимные пользователи, а связи – дружеские отношения. Загрузите эту сеть в граф библиотеки NetworkX, чтобы ответить на следующие ниже вопросы. Убедитесь, что вы используете надлежащий класс графа для ненаправленной невзвешенной сети.

1. Какая доля узлов имеет степень 100 или выше?
2. Какова максимальная степень для узлов в этой сети?
3. Пользователи в этой сети анонимизируются путем назначения узлам числовых имен. Какой узел имеет наивысшую степень?
4. Каков 95-й процентиль для степени (т. е. такое значение, при котором 95 % узлов имеют эту степень или меньше)?
5. Какова средняя степень для узлов в этой сети? Округлите до ближайшего целого числа.
6. Какой из следующих ниже контуров лучше всего описывает распределение значений степени в этой сети? Вы можете получить ответ визуально, используя гистограммы или просто статистику.
 - a. Равномерный: степени узлов равномерно распределены между минимумом и максимумом.
 - b. Нормальный: большинство степеней узлов близки к среднему значению, быстро снижаясь в обоих направлениях.
 - c. Правосторонний: большинство степеней узлов относительно малы по сравнению с размахом степеней.
 - d. Левосторонний: большинство степеней узлов относительно велики по сравнению с размахом степеней.



Связь: отношение между двумя вещами или ситуациями, в особенности когда одна вещь влияет на другую.

Многие реально существующие сети являются направленными и взвешенными. Мы видели, например, в главе 0, что пищевые паутины соединяют виды направленными и взвешенными связями, которые представляют направление и объем добычи, потребляемой хищником. Другие знакомые примеры включают «Википедию» и Всемирную паутину в целом, где гиперсвязи взвешиваются на трафик кликов; каждое приложение, в котором мы оцениваем товары и услуги, от книг до кинофильмов и от драйверов до песен; социальные сети, проистекающие из Twitter, где связи друг/подписчик взвешиваются на ретвиты, цитаты, ответы и упоминания; и даже Facebook, где взаимодействие с друзьями взвешивается на комментарии, лайки и перепосты. Многие из этих сетей построены на основе технологий интернета и Всемирной паутины. В этой главе вы познакомитесь с несколькими из этих сетей и протоколов.

4.1. Направленные сети

В сетях, которые мы обсуждали до этого, направление связей не имеет значения. В социальных сетях мы часто исходим из допущения – даже если это не всегда так, – что дружба обладает симметрией: если Алиса – подруга Боба, то Боб в равной степени – друг Алисы. В интернете пакеты перемещаются в обоих направлениях между двумя маршрутизаторами или двумя автономными системами. По большинству дорог автомобили движутся в обе стороны, и на каждый рейс из Нью-Йорка в Рим приходится рейс из Рима в Нью-Йорк. Во многих сетях эти симметрии не соблюдаются; связь имеет определенное направление и может не отвечать взаимностью. Например, если Чарли подписывается на Донну в Twitter, Донна может и не подписываться на Чарли в ответ. Как упоминалось в главе 1, *направленная связь* имеет *источниковый* и *целевой* узел. Направленная связь обычно представлена в виде стрелки, показывающей от источника к цели. Мы называем сеть с направленными связями *направленной сетью*. В то время как в ненаправленной сети каждый узел имеет степень, в направленной

сети каждый узел имеет *степень-на-входе* (число входящих связей) и *степень-на-выходе* (число исходящих связей).

Мы наблюдаем направленные связи чаще всего в коммуникационных и информационных сетях. Знакомые примеры включают электронную почту (см. рис. 0.4) и «Википедию» (см. рис. 0.5). Еще один примечательный пример взят из науки. Ученые всегда опираются на то, что было сделано раньше. Когда они публикуют свои выводы, они ссылаются на относящуюся к теме работу в предыдущих публикациях. Результирующая *сеть цитирования* является прототипическим примером информационной сети. Узлы представляют статьи. Связь из одной статьи в другую, именуемая *цитированием*, указывает на некоторое отношение между содержанием статей: общую методологию, альтернативный подход к решению задачи, предыдущий вывод, который был подтвержден, усовершенствован или, возможно, даже опровергнут. На рис. 4.1 представлена иллюстрация сетей цитирования.

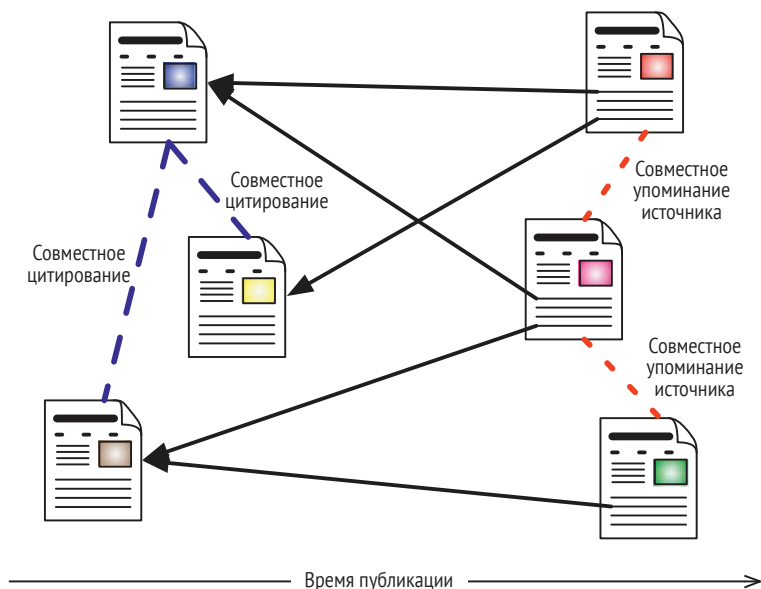


Рис. 4.1 Сеть цитирования. Связи цитирования показаны сплошными черными стрелками. Также показаны, соответственно, пунктирными синими и красными линиями (раздел 4.6) ненаправленные связи *совместного цитирования* и *совместного упоминания источника*, индуцируемые сетью цитирования. Обратите внимание, что в сети цитирования связи всегда должны указывать во времени назад: мы не можем цитировать статью, которая еще не была опубликована

4.2. Всемирная паутина

Мы все знакомы со Всемирной паутиной, так называемым Вебом, где гиперсвязь ведет со страницы **A** на страницу **B**, в то время как стра-

ница **В** может и не содержать связи со страницей **А**. Интересно, что идея двунаправленных связей существовала десятилетиями до того, как была изобретена *Всемирная паутина*. Реализовать двунаправленные связи во Всемирной паутине было сложно технически, поскольку для согласования и хранения информации о связях потребовался бы какой-то центральный орган.



4.2.1. Краткая история Всемирной паутины

В начале 1990-х годов Тим Бернерс-Ли представил модель направленной гиперсвязи, которую было легко имплементировать, потому что любой мог переходить со страницы на другую, не беспокоясь о взаимности из целевой страницы или даже о постоянстве; если целевой страницы не существовало, то пользователь просто сталкивался бы с неработающей связью. Многие люди начали писать страницы паутины, используя гипертекстовый язык Бернерса-Ли для содержимого паутины, и размещать сайты паутины (дословно «площадки»), используя его коммуникационный протокол связи навигаторов (браузеров) и серверов. Так родилась Всемирная паутина, или Веб.

Эта связь была ключом к успеху Всемирной паутины. Каждая страница имела адрес в паутине, именуемый *URL* (Uniform Resource Locator – Единый локатор ресурсов), упрощая переход с одной страницы на другую. В течение примерно 10 лет Всемирная паутина росла по мере того, как многие организации создавали сайты для представления информации и продажи товаров и услуг. Но Всемирная паутина по-прежнему оставалась главным образом сетью, в которой производители информации отличались от ее потребителей. Создание веб-сайта требовало навыков, которыми не обладало большинство людей. Появление онлайн-журналов, именуемых «веб-журналами», или *блогами* (от англ. *web log*), изменило эту ситуацию. Блоги облегчили людям создание простых сайтов в соответствии с шаблонами и создание контента в виде записей в журналах (блогах), размещаемых у сторонних провайдеров. Каждая запись в блоге имела URL-адрес, упрощая переход от одной записи к другой. Блогеры размещали связи, указывающие на блоги друг друга. Блоги быстро стали самым быстрорастущим сегментом Всемирной паутины, даже конкурируя с традиционными СМИ. Самое главное, что многие люди, которые были в основном потребителями информации, также стали ее производителями. Это был один из важных аспектов так называемой революции *Веб 2.0*.

Аналогично тому, как людям стало проще обмениваться информацией через блоги, им также стало легко обмениваться фотографиями, кинофильмами и всевозможными другими медиа через такие сайты, как Flickr и YouTube. Люди также могли делиться связями, публикуя свои закладки на сайтах с тегами, а затем на сайтах социально-сетевого взаимодействия. Соединение через Всемирную паутину стало

повсеместным и знакомым. Естественным следующим шагом было связать людей друг с другом, и это произошло с помощью онлайн-вых социальных сетей, таких как Friendster, Orkut, MySpace, LinkedIn и Facebook. В целях еще большего снижения затрат на создание узлов и связей появилась концепция *микроблога*, позволяющая людям публиковать очень короткие сообщения для трансляции своим друзьям. Это сочетание социальных сетей и блогов, введенное Twitter и вскоре скопированное Facebook, стало настолько популярным, что значительная часть населения Земли теперь является частью Всемирной паутины.

С точки зрения сети понятие узла расширилось за счет представления всего, что имеет URL-адрес, от страниц до людей, от сайтов до мыслей, от фотографий до песен, от кинофильмов до статей и т. д. Понятие связи расширилось аналогичным образом, потому что любой объект может указывать на любой другой: твиты имеют связь, указывающую на записи в блогах, статьи «Википедии» имеют связь, указывающую на другие статьи и внешние страницы, люди соединяются с друзьями и любимыми вещами, карты имеют связь, указывающую на фотографии (и наоборот) и т. д. Таким образом, Всемирная паутина выросла и охватывает практически все аспекты нашей жизни.

4.2.2. Как работает Всемирная паутина

Давайте вернемся к основам Всемирной паутины. В целях более глубокого понимания принципа работы этой вездесущей сети и того, каким образом можно собирать данные о ней, нам нужно получить представление о ее языке и протоколе. Страницы пишутся на некоторой версии *HTML* (HyperText Markup Language – языке разметки гипертекста) с интерактивностью, обеспечиваемой языками написания скриптов, такими как Javascript, которые могут интерпретироваться браузером, или навигатором. Подробности об этих языках выходят за рамки данной книги, за исключением важной концепции гиперсвязей между страницами, которую мы уже обсуждали. HTML предоставляет простой способ кодирования связи, указывающей на другую страницу, с помощью специального *якорного тега* (`<a>`). Например, простой исходный код `новости` создает связь для *якорного текста* «новости», которая при нажатии на нем пользователем приведет к тому, что браузер доставит страницу, находящуюся по адресу npr.org.

Принцип работы доставки или скачивания страниц между клиентом (браузером) и веб-сервером определяется *HTTP-протоколом* (HyperText Transfer Protocol – протоколом передачи гипертекста). Подобно другим интернет-протоколам «клиент–сервер», HTTP-протокол на самом деле довольно прост: он предписывает правила в отношении того, как клиент запрашивает страницу и как сервер отвечает. Рисунок 4.2 иллюстрирует указанный протокол на примере. Для того что-

бы сначала установить соединение с сервером, клиент должен знать *IP-адрес* (Internet Protocol Address – интернет-протокольный адрес) сервера. URL-адрес обычно указывает *уникальное сетевое имя*¹ сервера и *путь к файлу*. Например, в URL-адресе <http://npr.org/> уникальным сетевым именем сервера является `npr.org` и путем к файлу является `/`. Поскольку в данном случае путь является каталогом, сервер будет искать принятое по умолчанию (дефолтное) имя файла, например `index.html` в этом каталоге. В целях получения IP-адреса браузер использует службу под названием DNS (Domain Name Service – Службу доменных имен). Это совсем другой протокол, который переводит уникальное сетевое имя (к примеру, `npr.org`) в соответствующий IP-адрес (к примеру, `216.35.221.76`).

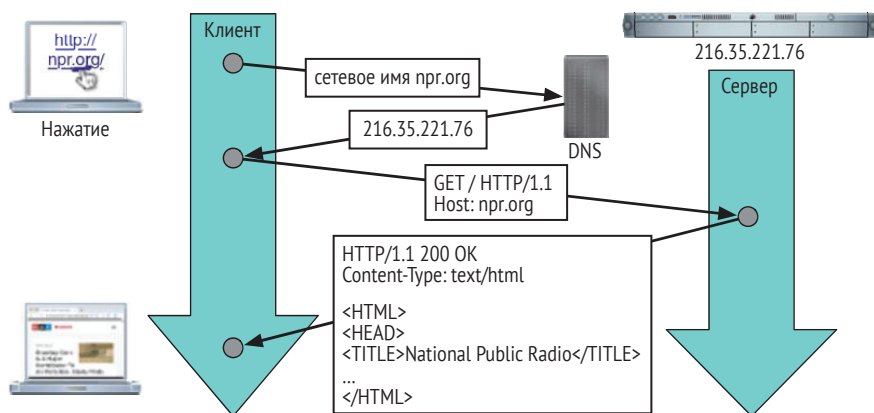


Рис. 4.2 Пример порядка взаимодействия клиента (браузера) и сервера по HTTP-протоколу, чтобы пройти по связи и посетить веб-страницу. Вертикальные стрелки показывают направление времени

Имея на руках IP-адрес, браузер соединяется с сервером. Как только соединение установлено, часть URL-адреса `http://` означает, что браузер переговаривается с сервером по HTTP-протоколу. Для этого браузер отправляет HTTP-запрос на сервер и ожидает HTTP-ответа. Сообщения запроса и ответа имеют формат, состоящий из *заголовка* с последующей пустой строкой, за которой следует необязательное *тело*. Заголовок запроса может состоять всего из нескольких строк. Наиболее распространенным типом запроса является запрос методом GET, который просто запрашивает страницу (путь). В этом случае

¹ Уникальное сетевое имя (hostname), или хост-имя – это назначаемое сетевому устройству символическое имя, которое может использоваться для организации доступа к этому устройству разными способами. Как правило, уникальное сетевое имя – это доменное имя, которое человеку значительно проще читать, помнить и произносить, чем тот же числовой IP-адрес, который тоже идентифицирует сетевое устройство. Обратите внимание, что в слитном написании hostname – это именно сетевое имя, а не пресловутое «имя хоста». – *Прим. перев.*

запрос не имеет тела, поэтому сервер отвечает, как только он получает пустую строку. В других случаях, таких как запрос методом POST, тело содержит дополнительные параметры содержимого. Его можно использовать для отправки входных данных в форму. В дополнение к типу запроса и пути в заголовке должно быть указано уникальное сетевое имя. Это обусловлено тем, что на одном сервере нередко может размещаться большое число веб-сайтов (технология *виртуально-го хостинга*¹). Заголовок ответа может содержать несколько строк информации, таких как тип сервера, дата, число возвращенных байтов и т. д. Самым важным является числовой *код ответа*. Например, код 200 означает «успех», тогда как код 404 означает «ресурс не найден». Тело ответа – это фактическое содержимое запрашиваемого ресурса, обычно HTML-код страницы.

4.2.3. Обходчики Всемирной паутины

Любая программа, которая использует протокол HTTP для запроса содержимого с веб-серверов, является веб-клиентом. Браузер – это привычный инструмент, который мы используем для навигации по Всемирной паутине, позволяющий нам практически передвигаться с одного узла на другой в этой огромной сети сайтов и страниц. *Обходчики Всемирной паутины*, или веб-сканеры (web-crawler), – это программы, которые скачивают веб-страницы автоматически. Поскольку информация во Всемирной паутине разбросана по миллиардам страниц, раздаваемых миллионами серверов по всему земному шару, обходчики предназначены для сбора информации, которую можно анализировать и добывать в централизованном месте. Первичное применение обходчиков лежит в сфере поисковых машин. Всемирная паутина – это динамичная сущность, эволюционирующая быстрыми темпами, следовательно, поисковые машины используют обходчики, чтобы оставаться свежими и предоставлять актуальную информацию по мере добавления, удаления, перемещения и обновления страниц и связей. Поисковая машина берет собираемую обходчиком информацию и создает структуру данных (*индекс*), которая соотносит контент (ключевые слова и фразы) со страницами, которые его содержат. Благодаря этому, когда пользователь отправляет запрос, поисковая машина может быстро получать страницы, содержащие указанные ключевые слова. Еще одна задача поисковой машины состоит в определении метода ранжирования результатов, чтобы пользователи могли отыскивать качественные результаты среди миллионов со-

¹ Виртуальный хостинг (virtual hosting) – это вид размещения большого числа веб-сайтов, при котором они располагаются на одном веб-сервере. Дословно указанный термин означает, что владелец веб-сайта становится виртуальным «хозяином» услуг по организации хранения и работы его веб-сайта на сервере провайдера. – *Прим. перев.*

впадений. Один из ключевых методов использует для этого сетевую структуру Всемирной паутины и представлен в разделе 4.3.

Другие виды использования обходчиков Всемирной паутины включают деловую аналитику, посредством которой организации отслеживают конкурентов и потенциальных сотрудников; цифровые библиотеки и библиометрические системы, чтобы делать научную работу доступнее и оценивать ее влияние; инструменты вебометрии для оценивания влияния учреждений в результате их онлайн-присутствия; и даже вредоносные приложения, например массовый сбор адресов электронной почты спамерами или коллекционирование конфиденциальной информации для выуживания и кражи конфиденциальных данных. Обходчики также задействуются в исследовательских целях, например для восстановления структуры графа связей Всемирной паутины. Поскольку обходчики очень полезны для изучения информационных сетей, давайте получим более глубокое представление о принципе их работы.

Обходчики представляют собой очень сложные программно-информационные системы; основатели поисковика Google Сергей Брин и Лоуренс Пейдж обозначили обходчик Всемирной паутины как самый изощренный, но хрупкий компонент поисковой машины. Но базовую концепцию обходчика понять не трудно. В своей простейшей форме обходчик – это просто алгоритм поиска сперва в ширину (раздел 2.5), работающий на графе связей Всемирной паутины. Он начинает свою работу с набора *затравочных* страниц (URL-адресов), а затем рекурсивно извлекает из них связи, получая еще больше страниц, и т. д. Данное простое описание скрывает технические трудности, такие как узкие места сетевого соединения, планирование запуска повторного посещения страниц, ловушки паука (когда серверы автоматически генерируют бессмысленные URL-адреса), канонические URL-адреса (для принятия решения о том, указывают ли две связи на одну и ту же страницу или нет), устойчивый синтаксический разбор (интерпретирование зачастую неправильного синтаксиса HTML-страниц) и этика работы с отдаленными веб-серверами.

На рис. 4.3 показан логический поток базового обходчика. Обходчик поддерживает очередь непосещенных URL-адресов, именуемую *границей продвижения*. Список инициализируется затравочными URL-адресами, которые обычно представляют собой множество высококачественных страниц, например из предыдущего обхода. На каждой итерации своего главного цикла обходчик подбирает следующий URL-адрес из границы продвижения, доставляет страницу, соответствующую этому URL-адресу, по HTTP-протоколу, разбирает полученную страницу, извлекая ее URL-адреса, добавляет вновь обнаруженные URL-адреса на границу продвижения и, наконец, сохраняет страницу (и другую извлеченную информацию, включая термины индекса и сетевую структуру) в репозитории. Обходчик останавливается, когда граница продвижения пуста, но на практике это случается редко из-за высокой средней степени-на-выходе страниц (порядка 10 или

более связей на страницу по всей Всемирной паутине). Процесс обхода может быть терминирован, когда было пройдено определенное число страниц, либо – как в случае с поисковыми машинами – продолжаться вечно.

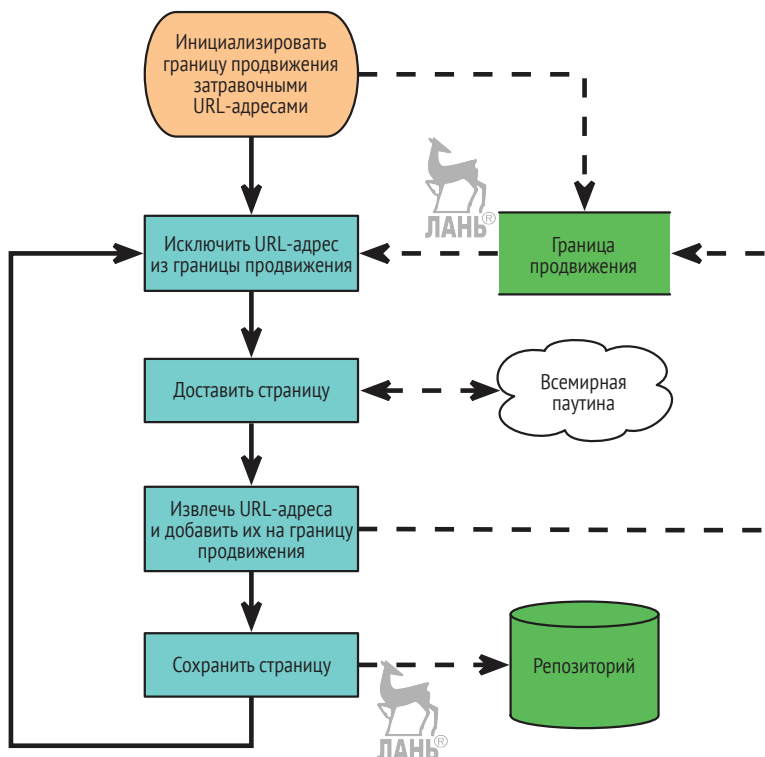


Рис. 4.3 Блок-схема базового обходчика. Главные операции с данными показаны пунктирными стрелками

Граница продвижения типичного обходчика быстро становится огромной, содержащей многие миллионы непосещенных связей. Обходчики нередко задействуют эвристику в попытке приоритезировать связи, которые могут привести к качественному контенту. Поскольку граница продвижения обычно имплементируется в форме очереди с принципом доступа «первым вошел – первым вышел» (FIFO), прежде чем мы посетим любую страницу на расстоянии n от затравочного URL-адреса, мы посещаем все страницы на расстоянии $n - 1$ или меньше. Это неплохая стратегия по той причине, что, как мы увидим ниже, чем дальше мы удаляемся от качественной страницы, тем меньше шансов отыскать качественные страницы. Обходчики, посеянные большими числами хороших страниц, следовательно, могут быстро открывать много новых хороших страниц или заново посещать уже известные страницы, чтобы узнавать, не были ли они обновлены с момента последнего посещения. Эффективные обходчики, которые задействуются поисковыми машинами, используют целый ряд хитростей с целью

оптимизирования процесса обхода и посвящают этой задаче кластеры с тысячами машин, работающих в параллельном режиме круглосуточно. Благодаря этому они обходят и индексируют миллионы страниц в день и поддерживают свежие результаты поиска.

4.2.4. Структура Всемирной паутины

Сеть, составленная из страниц Всемирной паутины и гиперсвязей, называется *графом Всемирной паутины*, или веб-графом. Крупномасштабные обходы выявили несколько интересных фактов о его структуре. Существуют различные (слабо) связанные компоненты. Их размеры, как правило, имеют скошенное (асимметричное) распределение, причем доминирует самая крупная компонента (более 90 % всех страниц) и очень много малых. Внутри гигантской компоненты мы находим самую крупную сильно связную компоненту. Вспомните из раздела 2.3, что *компонента-на-входе* и *компонента-на-выходе* гигантской сильно связной компоненты представляют собой множества страниц, соответственно, с которых гигантская компонента может быть достигнута и которые могут быть достигнуты из этой компоненты, следуя по направленным путям. Когда говорят о Всемирной паутине, указанную компоненту иногда называют структурой *галстук-бабочка* (см. рис. 4.4). Относительные размеры гигантской сильно связной компоненты, компоненты-на-входе и компоненты-на-выходе, варьируются в зависимости от стратегий, используемых обходчиками Всемирной паутины для сбора сетевых данных.

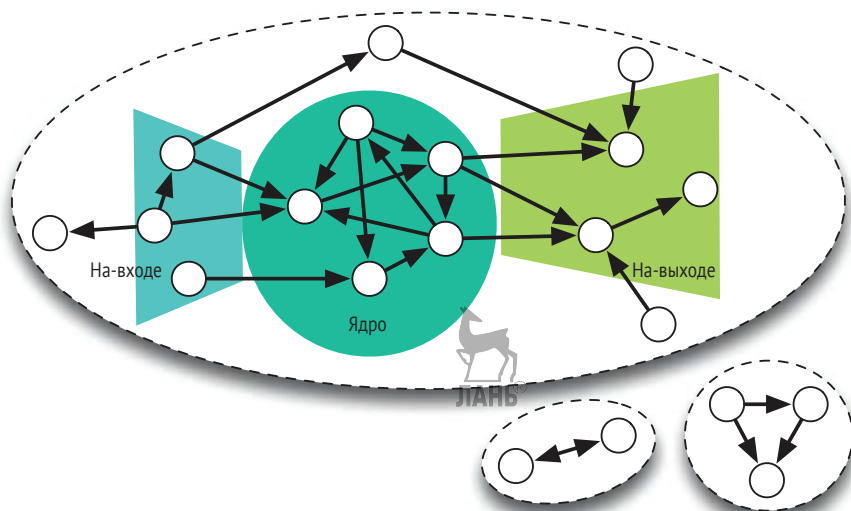


Рис. 4.4 Структура графа Всемирной паутины в форме галстука-бабочки. Компоненты выделены пунктирными овалами. Гигантская компонента имеет гигантскую сильно связную компоненту (иногда именуемую «ядром»), компоненту-на-входе («на входе») и компоненту-на-выходе («на выходе»)

Несколько исследовательских коллективов изучило степенное распределение графа Всемирной паутины, основываясь на крупных обходах. Средняя степень-на-входе (число связей, указывающих на страницу) составляет около 10–30 связей, но стандартное отклонение по меньшей мере на порядок больше, вследствие чего параметр гетерогенности k имеет крупное значение (см. раздел 3.2). Поэтому среднее значение является не самой содержательной мерой. На самом деле распределение значений степени-на-входе имеет утяжеленный хвост, охватывающий несколько порядков величины, как показано на рис. 4.5. Это показывает присутствие огромных страниц-хабов во Всемирной паутине, которые захватывают непропорциональный объем связей и трафика. Скошенное распределение значений степени-на-входе является устойчивым признаком Всемирной паутины, и оно не изменилось с тех пор, как Всемирной паутине было всего несколько лет от роду и ее размер составлял несколько сотен миллионов страниц.

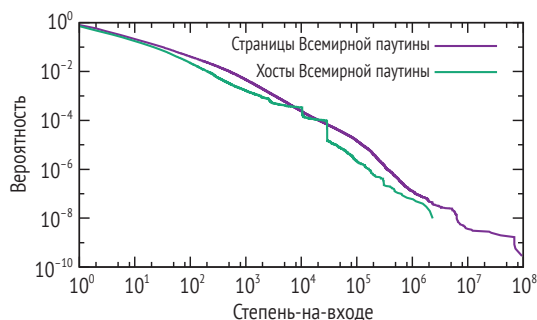


Рис. 4.5 Кумулятивное распределение значений степени-на-входе для графа Всемирной паутины. Эта сеть, основанная на крупном обходе, за август 2012 года насчитывает $N = 3.6$ млрд страниц и $L = 129$ млрд связей. Распределение значений степени-на-входе также показано для *графа хостов*, в котором узлы представляют не отдельные страницы, а целые сайты паутины, а связь указывает, что между страницами двух сайтов имеется по меньшей мере одна гиперсвязь. В обоих случаях мы наблюдаем хабовые узлы с огромным числом входящих связей (сеть из webdatacommons.org, основанная на данных обхода из commoncrawl.org)

Проводить анализ распределения значений степеней-на-выходе труднее. Хотя обходчики отыскивают страницы с тысячами исходящих связей, указанное распределение не охватывает столько порядков величины, сколько оно охватывает для степени-на-входе. Что еще важнее, в то время как страница с большим числом входящих связей обычно является сигнатурой популярности, страница, содержащая слишком много связей, указывающих на другие страницы, обычно является сигнатурой спамного поведения: так называемые *фермы связей*, создаваемые для подстегивания поискового рейтинга сайта. Учтите также и то, что по соображениям эффективности обходчики нередко урезают скачивание очень длинных страниц, вследствие чего меры степени-на-выходе ненадежны.

Исследователи также использовали данные обходов для изучения средней длины пути в графе Всемирной паутины. Его средняя длина пути растет очень медленно вместе с числом узлов: по мере увеличения размера сети на несколько порядков величины кратчайший путь в среднем становится длиннее всего на несколько шагов. Например, самая крупная сильно связанная компонента сети, полученная из данных обходов за 2012 год, используемых для рис. 4.5, имеет $N = 1.8$ млрд страниц, а средняя длина пути составляет менее 13 связей. Эта *ультрамаломировая* структура обусловлена присутствием хабовых страниц, как мы обсуждали в главе 3.

Те же признаки Всемирной паутины – тяжелохвостные распределения значений размера компоненты и степени-на-входе, сигнатура ультрамалого мира – также наблюдались в других информационных сетях, в частности в блогах и в «Википедии».

4.2.5. Тематическая локальность

В разделе 2.1 мы дали определение *гомофилии* как тенденции схожих узлов соединяться друг с другом. Узлы информационных сетей, такие как страницы Всемирной паутины, статьи «Википедии» и исследовательские работы, богаты содержимым – текстовыми атрибутами, которые мы можем использовать для определения и измерения *сходства* между двумя страницами или документами. Основываясь на содержимом, мы можем определять тему страницы или чему она посвящена: две страницы о спорте похожи друг на друга больше, чем одна страница о спорте, а другая о музыке. Следовательно, мы можем выразить гомофилию в информационной сети как способность угадывать, чему посвящена страница, глядя на содержимое соседних страниц. Две страницы по родственным темам могут быть связаны друг с другом или могут иметь короткий путь, который их соединяет. Когда это происходит, мы говорим, что сеть имеет *тематическую локальность*. Причина тематической локальности интуитивно понятна: когда создаются новые страницы, статьи или сообщения в блоге, авторы хотят помочь своим читателям, указывая связи на тематически релевантную информацию. В результате связи кодируют семантическую информацию об узлах.

В целях квантификации тематической локальности мы можем измерить величину вероятности того, что целевая страница на заданном расстоянии от источниковой страницы имеет примерно ту же тему, что и источниковая страница (рис. 4.6). Затем можем сравнить ее с нашим ожиданием случайно наткнуться на страницу по той же теме, что зависит от степени общности темы. Страницы в пределах одной или двух связей от источниковой страницы на порядки с большей вероятностью будут посвящены той же теме, что и источниковая страница, по сравнению со случайными страницами.

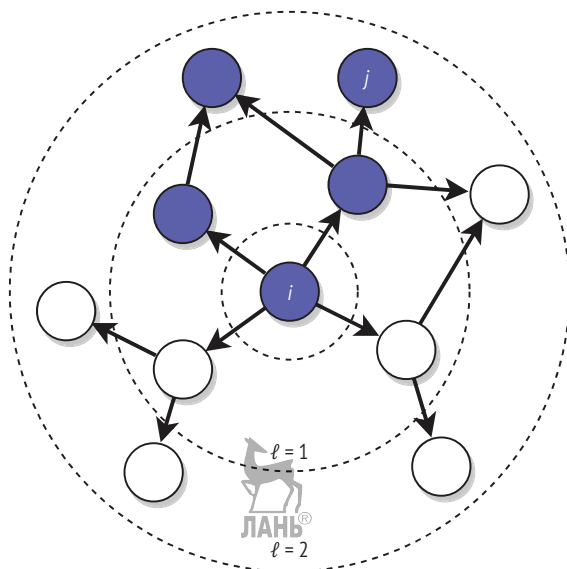


Рис. 4.6 Иллюстрация тематической локальности. Половина страниц на расстоянии $\ell = 1$ от страницы i и одна треть страниц на расстоянии $\ell = 2$ посвящены примерно одной и той же теме (показаны синим цветом)

На практике одним из способов измерения тематической локальности является использование *текстового сходства* в качестве косвенного индикатора для тематической взаимоувязанности. Текстовое сходство основано на совместной встречаемости ключевых слов на двух страницах или в документах. Чем больше ключевых слов используется двумя страницами совместно, тем сильнее подтверждающие данные о том, что эти две страницы посвящены одной и той же теме. Широко используемой мерой текстового сходства является *косинусное сходство*, описанное во вставке 4.1. В целях квантификации тематической локальности можно выполнить обход сперва в ширину до некоторого расстояния от одной или нескольких затравочных страниц по какой-либо теме и измерить сходство между пройденной страницей и затравочной страницей, усредняя по всем затравочным страницам и всем страницам в обходе. Используя эту методологию, мы можем построить график сходства как функцию от расстояния между двумя страницами и заметить, что соседние страницы, как правило, более похожи, чем отдаленные страницы (рис. 4.7). Естественно, когда мы продолжаем навигацию, отдаляясь от страницы, у нас меньше вероятности столкнуться со страницами по родственной теме; это явление называется *тематическим дрейфом*.

Тематическая локальность – это своего рода отношение между структурой информационных сетей и содержимым узлов – тем, что сеть говорит нам о содержимом, и наоборот. Если мы начинаем с «хороших» затравочных страниц и не отклоняемся слишком далеко, то

мы, скорее всего, отыщем другие качественные страницы. Это одна из причин, по которой применяемые в поисковых машинах обходчики задействуют алгоритмы поиска сперва в ширину. Тематическая локальность также является причиной того, почему вообще имеет смысл осуществлять «навигацию» по Всемирной паутине, – если бы тематической локальности не существовало, то неужели бы вы стали кликать на веб-связи, указывающей на другую страницу?

Вставка 4.1

Косинусное сходство

В информационном поиске и добыче знаний из текстовых данных¹ нередко требуется измерять сходство между двумя документами, веб-страницами, кусками текста или облаками тегов. Давайте представим каждый документ d как высоко-размерный вектор, в котором размерность ассоциирована с каждым термином в словаре: $\vec{d} = \{w_{d,1}, \dots, w_{d,n_t}\}$, где $w_{d,t}$ – это вес термина t в документе d и n_t – суммарное число терминов. Методы глубокого обучения, основанные на искусственных нейронных сетях, приводят к похожим векторным представлениям, за исключением того, что каждая размерность соответствует абстрактному понятию, а не слову или тегу. Существуют различные способы расчета весов. В типичной ситуации они пропорциональны частоте, с которой термин появляется в документе – термин, который встречается часто, считается хорошим описателем. Часто удаляют «шумные слова», которые не имеют смысла, такие как артикли и союзы. Также принято дисконтировать веса терминов, которые встречаются во многих документах, поскольку они обладают низкой различающей способностью. Затем мы можем вычислить сходство между двумя документами d_1 и d_2 , измерив косинус между их векторами:

$$\cos(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{\|\vec{d}_1\| \cdot \|\vec{d}_2\|} = \frac{\sum_t w_{d_1,t} w_{d_2,t}}{\sqrt{\sum_t w_{d_1,t}^2} \sqrt{\sum_t w_{d_2,t}^2}}.$$

Если термины в d_1 также присутствуют в d_2 и наоборот, то косинус близок к единице; если оба документа не содержат общих терминов, то косинус равен нулю. Обратите внимание, что косинус нормализуется нормой или размером каждого вектора:

$$\|\vec{d}\| = \sqrt{\sum_t w_{d,t}^2}.$$

Благодаря этому более длинный документ не будет выглядеть похожим на многие другие только потому, что в нем много терминов: его большая норма уменьшала бы сходство.

¹ Добыча знаний из текстовых данных (text mining) – это процесс обнаружения в исходных данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений. – *Прим. перев.*

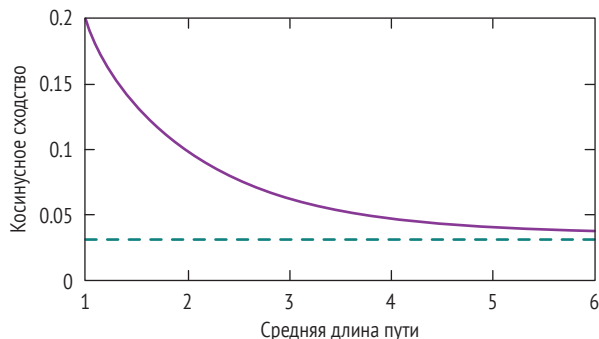


Рис. 4.7 Тематическая локальность во Всемирной паутине, измеряемая с помощью обходов на основе поиска сперва в ширину, начиная со 100 наборов затравочных страниц такого же числа тем. По каждому обходу мы сообщаем среднее косинусное сходство между затравочными страницами и проходимыми страницами как функцию от средней длины пути между ними. Мы наблюдаем сильную тематическую локальность: сходство между затравочными страницами и проходимыми страницами, находящимися на расстоянии одной связи, превышает более чем в шесть раз уровень шума, ожидаемый между случайными страницами (пунктирная линия). По мере того как мы блуждаем все дальше, тематический дрейф иллюстрируется снижением среднего сходства между затравочными страницами и проходимыми страницами в сторону уровня шума

Из структуры сети можно почерпнуть и другие важные подсказки о содержимом страниц. Например, локальная сетевая окрестность страницы может быть сигнатурой спамного контента, например когда единственные связи, указывающие на страницу, происходят от узлов, которые сами не имеют входящих связей. И наоборот, когда на странице много входящих связей со страниц с высокой степенью-на-входе, это может быть наводкой на качество или престижность содержимого страницы; подробнее об этом в следующем разделе.

4.3. Метрика PageRank

Мы узнали, что добытые обходчиком Всемирной паутины страницы обрабатываются поисковой машиной для создания поискового индекса – структуры данных, в которой перечисляются все страницы, содержащие любое данное слово или фразу. Поэтому, когда вы отправляете запрос, поисковая машина способна быстро перечислить список всех соответствующих страниц. Но совпадать с определенным запросом могут миллионы веб-страниц, к примеру, с запросом «социальная сеть», а вы располагаете временем на то, чтоб просмотреть лишь несколько из них. Следовательно, *алгоритмы ранжирования* являются не менее важным компонентом поисковой машины. Если бы результаты сортировались исключительно по сходству содержимого страницы и запроса, то пользователям пришлось бы просматривать большое число страниц низкого качества и даже спам. Но если ре-

зультаты ранжируются также с учетом некоторой важности страницы или меры престижности, то верхние результаты, скорее всего, будут релевантными, интересными и надежными. Поисковые машины начали задействовать меры сетевой центральности в качестве критериев ранжирования в 1998 году, когда Сергей Брин и Ларри Пейдж представили метрику *PageRank* в качестве компонента новой поисковой машины, которую они назвали Google.

Метрика PageRank (дословно ранг страницы) – это алгоритм или процедура для вычисления меры центральности, который ориентирован на улавливание престижности или важности каждого узла; в типичной ситуации он используется в направленных сетях. Это также название, которое мы даем самой мере центральности. Поэтому при применении ко Всемирной паутине указанный алгоритм назначает каждой странице значение метрики PageRank. Применяемый поисковой машиной алгоритм ранжирования может затем использовать это значение в сочетании со многими другими факторами, такими как совпадение между запросом и текстом страницы, для сортировки результатов запроса. Страница с высоким рангом PageRank считается престижной или важной и подстегивается алгоритмом ранжирования: при прочих равных условиях страницы с большим рангом PageRank ранжируются выше. Рассмотрим в качестве примера спамера, который копирует содержимое статьи «Википедии» по социальной сети на страницу блога, заполненную рекламой. С точки зрения содержимого изначальная и плагирированная страница могут выглядеть очень похожими, но страница «Википедии» имеет гораздо более высокий ранг PageRank. Поэтому, когда вы отправите запрос «социальная сеть», вы увидите статью «Википедии» среди верхних результатов, в то время как страница спамера будет скрыта, и вы, вероятно, ее не увидите.

Интуитивная идея метрики PageRank проистекает из представления о людях, которые путешествуют по интернету в случайном порядке, т. е. переходят по случайным связям со страницы на страницу. Этот процесс именуется *случайным блужданием* (или случайным серфингом) по графу Всемирной паутины. Случайное блуждание – это простая модель поведения, проявляемая пользователем во время навигации или поиска; не зная, где может находиться желаемая информация или какая связь приведет к ней, мы принимаем простейшее допущение, что каждая связь на странице имеет равные шансы быть кликнутой. Мы также хотим уловить еще одно поведение пользователя, а именно, что пользователь может в любой момент устать от навигации и начать новый навигационный сеанс. Алгоритм метрики PageRank моделирует это с помощью случайных *прыжков* с текущей страницы на некоторую другую, случайно отобранную среди всех страниц в сети. Процесс случайного прыжка называется *телепортацией*.

Представив себе, что многие люди выполняют эти процессы модифицированного случайного блуждания (серфинг плюс прыжки) в те-

чение длительного времени, можно измерить частоту, с которой будет посещаться каждая страница. Доля случаев, когда мы попадаем на страницу, – это то, что мы называем рангом, *PageRank*, этой страницы. На рис. 4.8 показаны значения ранга PageRank в направленной сети; узлы с ведущими к ним многочисленными путями чаще посещаются случайными пользователями, и, следовательно, имеют высокий ранг PageRank. На практике метрика PageRank вычисляется эффективнее, как описано во вставке 4.2. В приложении В.1 представлена интерактивная демонстрация алгоритма PageRank.

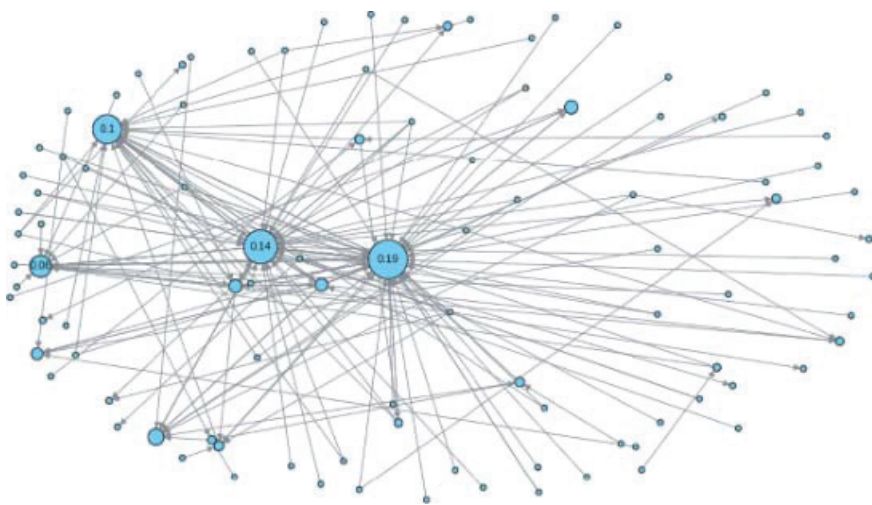


Рис. 4.8 Метрика PageRank в направленной сети. Размер узлов пропорционален их значению метрики PageRank, который показан для нескольких узлов

Вставка 4.2

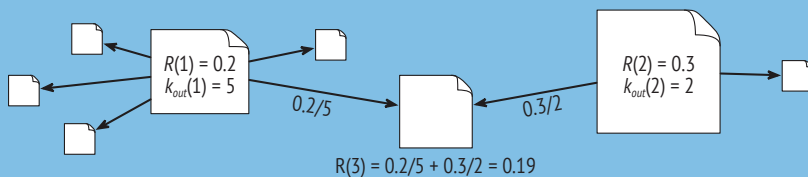
Модель PageRank

Метрика PageRank обычно вычисляется на основе структуры связей графа Всемирной паутины с помощью итеративного подхода, именуемого *степенным методом* (power method). Его идея состоит в том, чтобы вычислять то, как алгоритм PageRank переходит со страницы на страницу. Метрика PageRank R каждого узла инициализируется некоторым значением (к примеру, $R_0 = 1/N$, таким образом, чтобы все значения в сумме равнялись единице). На каждом шаге значение каждого узла уточняется до тех пор, пока процесс не сойдется, т. е. ни одно из значений метрики PageRank не будет изменяться от одного шага к следующему. Мы исходим из допущения, что случайные прыжки происходят с вероятностью, выражаемой параметром α , именуемым *телепортационным фактором*, обычно устанавливаемым равным малому значению $\alpha \approx 0.15$. Вероятность $1 - \alpha$, также именуемая коэффициентом затухания, вместо этого связана с процессом случайного блуждания. Согласно модели PageRank, на каждом шаге пользователь с вероятностью α перепрыгивает на узел, отбираемый случайно среди всех страниц; пользователь с вероятностью $1 - \alpha$ будет продолжать навигацию, следуя по

случайной связи с текущей страницы. Следовательно, ранг PageRank узла i в момент времени t представляет собой сумму двух слагаемых, выражающих два пути, которыми можно попасть на страницу i :

$$R_t(i) = \frac{\alpha}{N} + (1 - \alpha) = \sum_{j \in \text{pred}(i)} \frac{R_{t-1}(j)}{k_{\text{out}}(j)}. \quad (4.1)$$

Первое слагаемое описывает телепортацию на i , которая является одной из N возможных целей прыжка. Второе слагаемое описывает то, как можно пересекать одну из связей, указывающих на i , во время случайного блуждания: суммирование выполняется на множестве предшественников страницы i (т. е. страниц, которые указывают на i). Каждая из этих страниц, j , имеет $k_{\text{out}}(j)$ исходящих связей. Ранг PageRank страницы j разбросан равномерно среди этих исходящих связей, одна из которых ведет к i . На приведенном ниже рисунке иллюстрируется один из шагов указанного процесса разбрасывания, описываемого вторым слагаемым уравнения ($\alpha = 0$) для страницы 3.



Обратите внимание на *рекурсивное* определение: ранг PageRank страницы зависит от соответствующих рангов ее соседей. Ранг PageRank является сохраняющей величиной ($\sum_i R(i) = 1$) по мере того, как он перетекает по связям с одной страницы на соседние, и значит, он не может создаваться или уничтожаться. Оказывается, что, поскольку для $\alpha > 0$ телепортация соединяет все узлы виртуально, ранг PageRank гарантированно сходится, и он это делает относительно быстро, менее чем за 100 шагов или около того, даже в очень крупных сетях. Поэтому степенной метод является гораздо более эффективным способом вычисления метрики PageRank по сравнению с моделированием модифицированных случайных блужданий.

Оказывается, распределение значений метрики PageRank довольно похоже на распределение значений степени-на-входе по Всемирной паутине (рис. 4.9). Почему бы тогда для ранжирования просто не использовать степень-на-входе? Отвечая на этот вопрос, следует учитывать, что не все пути равны. Пути из страниц, которые сами посещаются, нередко обеспечивают более высокий прирост. Другими словами, на важность страницы влияет важность страниц, которые связаны с ней, – что бы вы предпочли: связь, указывающую на вашу домашнюю страницу из блога вашего друга или же из титульной страницы газеты «*Нью-Йорк таймс*»? Это важный аспект, которым метрика PageRank отличается от степени-на-входе. Среди двух страниц с одинаковой степенью-на-входе в игре побеждает та, на которую указывают страницы с более высоким рангом PageRank.

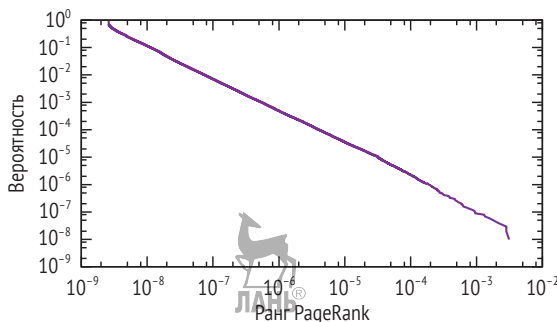


Рис. 4.9 Кумулятивное распределение значений метрики PageRank на графе хостов Всемирной паутины. Значения рангов PageRank нормализуются таким образом, чтобы они в сумме составляли единицу. Сеть получена из тех же данных обхода за 2012 год, которые использовались для рис. 4.5

И конечно же, люди играют в эту игру. Разница между высоким и низким рангом PageRank может означать появление на первой странице результатов поиска или нет, а это в свою очередь может означать выживание или крах бизнеса. Многие компании процветают исключительно благодаря своим хорошим поисковым рейтингам. Поэтому неудивительно, что существует целая индустрия *оптимизации веб-сайтов в поисковых машинах* (search engine optimization, SEO), которая помогает веб-сайтам повышать свой поисковый рейтинг. Большинство SEO-компаний задействует методы, санкционированные поисковыми машинами, такие как принятие описательного текста страницы и усовершенствование приспособленности веб-сайта для навигации. Однако менее scrupulous SEO-агенты могут использовать методы, к которым поисковые машины относятся неодобрительно. Такие методы нередко собирательно именуются «спамдексированием» (spamdexing) или заспамливанием поискового индекса. Один из часто применяемых подходов на основе спамдексирования состоит в создании *ферм-связей*, крупных наборов вымышленных веб-сайтов, которые связаны друг с другом и с целевым веб-сайтом. Такая кликовая (clique) структура предназначена для обмана PageRank-подобных алгоритмов и подстегивания рейтинга целевой страницы. Поисковые машины задействуют изощренные сетевые алгоритмы для борьбы с фермами связей и другими атаками спамдексирования. Когда они обнаруживают такого рода злоупотребления, они могут удалять веб-сайт из поискового индекса.

Библиотека NetworkX предоставляет функцию, которая выполняет алгоритм PageRank в заданной направленной сети и возвращает словарь со значениями рангов PageRank узлов:

```
PR_dict = nx.pagerank(D) # D - это орграф
```

4.4. Взвешенные сети

До сих пор мы сосредотачивались на невзвешенных сетях, в которых связи имеют двоичную природу: два узла либо связаны, либо нет. Однако связи между реально существующими сущностями редко бывают такими черно-белыми. Очень часто связи имеют атрибуты, которые позволяют нам сравнивать две связи и определять, какая из них сильнее. В главе 0 мы рассмотрели несколько примеров взвешенных сетей: ретвитная сеть Twitter, в которой два аккаунта могут ретвитить друг друга любое число раз; сеть электронной почты, в которой пользователи могут отправлять друг другу любое число сообщений; интернет, где данные, передаваемые по физической связи между двумя маршрутизаторами, измеряются числом пакетов или битов; мозговые сети, где синапсы между нейронами передают электрические сигналы возбуждения с разной скоростью; и пищевые паутины, в которых мы можем характеризовать биомассу видов добычи, потребляемой видами хищника.

Даже когда мы думаем о сети как о невзвешенной, это часто является лишь отражением нашей попытки упростить то, как мы моделируем и анализируем лежащие в основе реально существующие отношения. Возьмем, к примеру, сеть Facebook: мы обычно думаем о дружеской связи как о бинарных отношениях. Однако не все дружеские отношения равны; у двух близких друзей может быть много общих контактов, они могут много раз лайкать и комментировать посты друг друга, а также могут изменять и помечать фотографии друг друга. То же самое не относится к двум отдаленным знакомым. Невзвешенная сеть друзей Facebook является всего лишь очень упрощенной моделью фактических отношений. Но не ошибитесь: такие платформы, как Facebook, отслеживают каждое ваше действие и хорошо осведомлены о силе каждой вашей связи. Другие социальные сети аналогичны; например, сеть кинозвезд может иметь взвешенные связи, основанные на числе фильмов, в которых две кинозвезды снимались вместе. Во вставке 5.6 мы увидим, что сила социальных связей уже давно играет важную роль в изучении социальных сетей.

Информационные и транспортные сети предоставляют больше примеров взвешенных направленных сетей: во Всемирной паутине и «Википедии» по некоторым связям кликают гораздо чаще, чем по другим. А в сетях авиационного сообщения некоторые соединения несут больше рейсов и перевозят больше пассажиров, чем другие.

Во всех этих сетях мы используем веса связей для представления важных мер, таких как сообщения, биты, лайки, клики и пассажиры. Вспомните из главы 1, что меры узловой центральности степени, степени-на-входе и степени-на-выходе, расширены на *силу*, *силу-на-входе* и *силу-на-выходе* во взвешенных сетях.



4.5. Информация и дезинформация

Давайте немного углубимся в признаки взвешенных направленных сетей, используя в качестве примера *сети диффузии информации*. В таких сетях узлы представляют людей, а связи представляют части информации – идеи, понятия, новости или проявления поведения, – которые передаются от человека к человеку. Передаваемая единица информации называется *мемом* – изображения с подписями являются лишь одним из видов интернет-мемов. Twitter предоставляет нам данные, которые идеально подходят для наблюдения за тем, как изображения, кинофильмы, веб-связи, фразы, хештеги и другие мемы распространяются онлайн. Каждый из этих мемов однозначно идентифицируется текстовым литералом: URL-адресом для связи во Всемирной паутине либо медиа-сущности либо меткой, предваряемой префиксом в виде знака «диез» (#) для хештега, выражающего понятие или тему. Твит может содержать несколько мемов. Например, сообщение «Уроженцы Индианы – лучшие #GOIU iuhoosiers.com» содержит хештег #GOIU и веб-связь <https://iuhoosiers.com/>.

Используя данные из Twitter, мы можем строить различные виды диффузионных сетей, улавливающие разные способы распространения мема: с помощью ретвитов, цитируемых твитов, упоминаний и ответов¹. Например, если Алиса упоминает Боба в твите, то Боб, скорее всего, увидит твит, и, следовательно, мы можем допустить, что твит распространился от Алисы к Бобу. Аналогичным образом если Алиса отвечает Бобу, то мы можем сделать вывод, что изначальное сообщение было доставлено от Боба Алисе. Для простоты давайте сосредоточимся на ретвитах. Если Алиса подписана на Боба, то она может ретвитнуть сообщение от Боба и тем самым распространять содержащийся в твите мем среди всех своих подписчиков. *Ретвитный каскад* представляет собой направленное дерево, которое улавливает то, как мем распространяется от его создателя ко всем пользователям, которые в конечном итоге подвергнутся его воздействию. Однако, как показано на рис. 4.10, реконструировать каскадное дерево из данных Twitter непросто. Тем не менее мы можем легко наблюдать за всеми пользователями, которые подвергаются воздействию этого мема. Все они соединены с источником, образуя звездную сеть. Если известна базовая сеть подписчиков, то существует возможность восстановить аппроксимацию каскадного дерева.

¹ Интерактивные диффузионные сети из Twitter можно разведывать, используя инструмент из Наблюдательного пункта за социальными медиа (Observatory on Social Media) по адресу: <https://osome.iu.edu/tools>. Вы также можете создавать анимационные фильмы, показывающие, как эти сети разворачиваются с течением времени, используя один из инструментов по адресу: <https://osome.iu.edu/research/open-source>. Общую информацию см. в видеоролике по адресу: <https://osome.iu.edu/videos>.

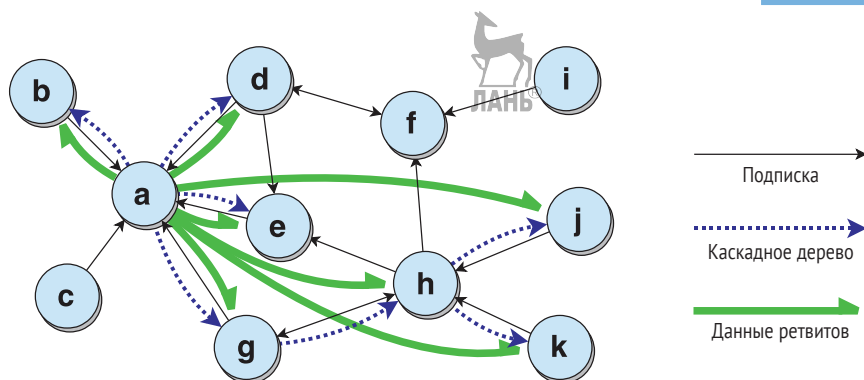


Рис. 4.10 Иллюстрация сетей подписчиков и ретвитов в Twitter. Пользователи обычно видят, а иногда и ретвитят твиты аккаунтов, на которые они подписаны. В этом примере **b** подписан на **a**, а также ретвитит сообщение от **a**. **c** также подписан на **a**, но не ретвитит **a**. Пользователь также может ретвитнуть сообщение, которое он видит, потому что оно было ретвитнуто кем-то, на кого он подписан. Здесь **h** не подписан на **a**, но ретвитит сообщение, исходящее от **a** и которое было ретвитнуто **g**, на которого подписан **h**. Отслеживая эти цепочки ретвитов, теоретически можно реконструировать *дерево ретвитного каскада* с **a** (источником твита) в качестве корневого узла. Однако Twitter не предоставляет данных о каскадах ретвитов. Вместо этого каждый ретвит указывает непосредственно на источник твита. Поэтому каскадное дерево становится звездной сетью с одинаковым корневым узлом

Один и тот же мем может генерировать большое число каскадных деревьев (звезд). Например, несколько пользователей могут делиться одной и той же связью, указывающей на новостную статью, или написать твит с использованием одного и того же хештега. Агрегируя все эти звездные деревья, мы получаем *лес* (множество деревьев), который называем диффузионной сетью.

При строительстве диффузионной сети сначала необходимо сформировать мем или мемы, распространение которых мы хотим анализировать. Нас может интересовать один хештег, скажем, #elections (выборы) или #soccer (футбол); или все связи, указывающие на статьи из источника новостей или набора источников новостей. Например, при расследовании онлайн-дезинформации мы, возможно, захотим отслеживать истории из источников с низким уровнем доверия, которые в рутинном порядке распространяют сфабрикованные новости, мистификации, теории заговора, гиперпартийный контент, клик-наживку или лженауку.

Мы можем трактовать каскадный лес, связанный с каждым хештегом, как слой в многослойной сети, как обсуждалось в разделе 1.8. И, поскольку ретвиты происходят в разное время, это сеть также является темпоральной. Диффузионная сеть на рис. 0.3 была получена путем агрегирования каскадных лесов во времени и по многим популярным хештегам, связанным с политическими разговорами во время промежуточных выборов в США 2010 года.

Построив диффузионную сеть, мы сможем наблюдать несколько признаков. Является ли разброс равномерным по всей сети или

сконцентрированным в плотных и изолированных кластерах узлов? Например, на рис. 0.3 показана поляризованная структура с двумя в основном сегрегированными сообществами: консерваторами и прогрессистами, каждое из которых ретвитит почти исключительно членов одной и той же группы. Эти сообщества иногда называются *эхокамерами*, потому что пользователь подвергается воздействию мнений, в основном подкрепляющих его собственные. На рис. 4.11 показан еще один пример эхокамеры, состоящей из людей, которые уязвимы для политической дезинформации. Обратите внимание, что пользователи, распространяющие дезинформацию, делятся очень малым числом статей из источников, проверяющих факты на достоверность. В главе 6 мы узнаем, как обнаруживать такие сообщества в сетях.

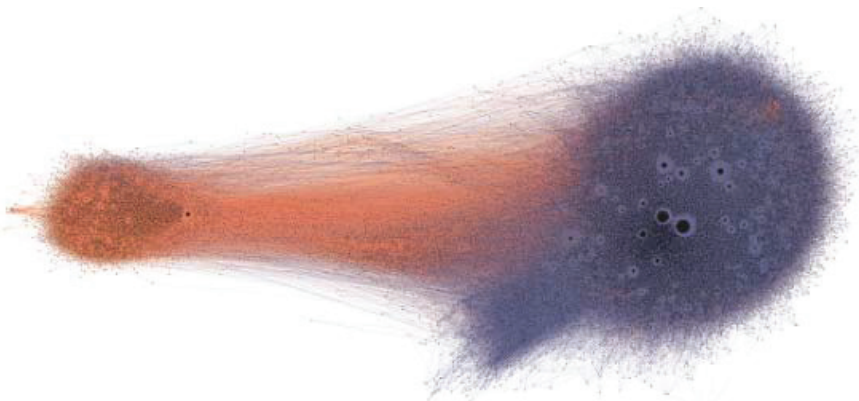


Рис. 4.11 Ретвитная подсеть для статей в преддверии выборов в США в 2016 году. Каждый из $N = 52452$ узлов представляет учетную запись Twitter, а каждая связь представляет ретвит, связывающий со статьей из источника с низким уровнем доверия (фиолетовый) либо источника с проверкой фактов на достоверность (оранжевый). Визуализированная подсеть является $k = 5$ ядром всей ретвитной сети (раздел 3.6). Изображение адаптировано из Шао и соавт. (2018a) по лицензии CC BY 4.0

Еще одним интересным свойством является *вирусность* мема. Самый простой способ его квантифицирования состоит в измерении размера диффузионной сети (т. е. числа пользователей, подвергавшихся воздействию мема). Однако при наличии крупной сети ее структура также показательна. Например, звездная сеть, в которой многие подписчики знаменитости ретвитят мем, может отражать популярность знаменитости больше, чем популярность мема. Однако, исходя из допущения, что мы можем реконструировать каскадные деревья, глубокая сеть с длинными цепочками ретвитов может указывать на более широкую привлекательность сообщения. Как показано на рис. 4.12, дезинформация нередко является более вирусной, чем фактические новостные сообщения.

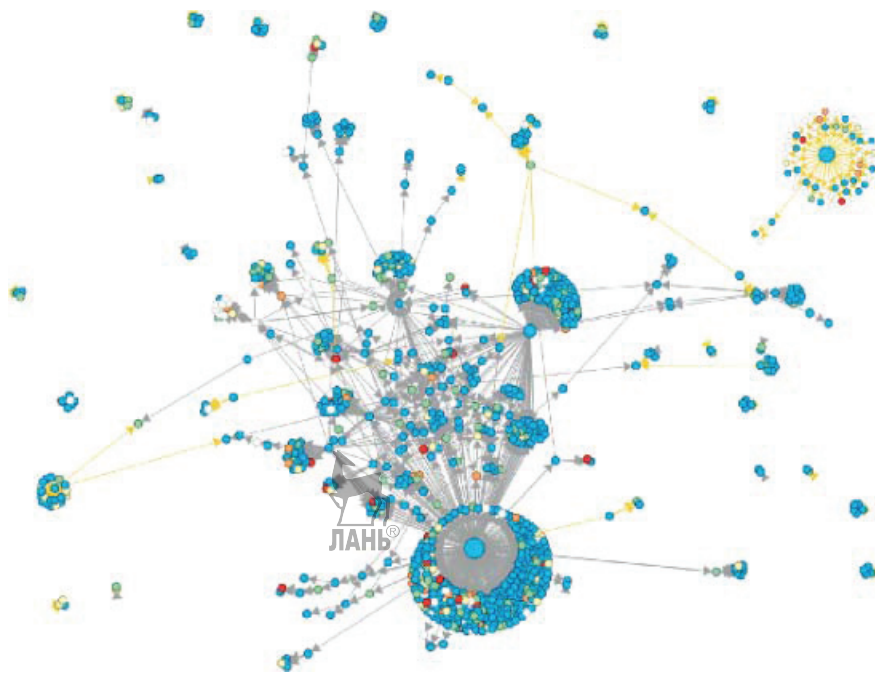


Рис. 4.12 Ретвитная сеть для двух статей о «Белых касках», волонтерской спасательной организации, действовавшей во время гражданской войны в Сирии. «Белые каски» стали мишенью кампании по дезинформации с ложными заявлениями о ее отношениях с террористическими организациями и другими теориями заговора. Серые и желтые связи изображают соответственно диффузию одного из этих ложных утверждений и статью с проверкой фактов на достоверность. Мы легко можем заметить, что распространение дезинформации носит более вирусный характер. Размер узла пропорционален силе-на-выходе, а цвет узла отражает вероятность того, что учетная запись автоматизирована: синие узлы, скорее всего, являются людьми, а красные узлы, скорее всего, являются ботами. Изображение из Noaху, инструмента, который визуализирует распространение дезинформации в Twitter (hoaxy.iuni.iu.edu)

Диффузионные сети могут давать нам представление о шаблонах производства и потребления информации. Во взвешенной и направленной ретвитной сети мера вес связи от Алисы к Бобу указывает на число раз, когда Боб распространял мем, происходящий из сообщений Алисы. Мы можем думать об Алисе как о производителе, а о Бобе как о потребителе информации о меме. Расширяя этот анализ на всю сеть целиком, мы можем использовать силу-на-входе и силу-на-выходе узла для измерения склонности пользователя соответственно к производству и потреблению информации – иметь ретвиты или ретвитить других. Разумеется, пользователь может играть обе роли. Следовательно, мы можем посмотреть на соотношение между силой-на-входе и силой-на-выходе, чтобы классифицировать пользователя: если соотношение намного больше единицы, то пользователь в основном является производителем; и наоборот, соотношение ниже единицы указывает на потребителя.

Независимо от того, фокусируемся ли мы на одном меме или же агрегируем все сообщения, высокое значение силы-на-выходе также может использоваться в качестве индикатора *влияния* в том смысле, что сообщения от пользователя часто ретвитятся. Рисунок 4.12 и изображение на обложке книги иллюстрируют влиятельные узлы в диффузионных сетях, рисуя узлы с размером, пропорциональным их силе-на-выходе. Использование силы-на-выходе в качестве косвенного индикатора влияния контрастирует степени-на-входе сети подписчиков (т. е. числа подписчиков), которая измеряет популярность, но не обязательно влияние; у одного может быть много подписчиков, которые не ретвитят. Сравнивая эти две величины – число ретвитов и число подписчиков, – мы можем лучше понять влияние человека.

Учитывая огромную мощь социальных сетей в информировании, убеждении и влиянии на всех нас, неудивительно, что для манипулирования этими платформами выделяется все больше ресурсов. Купить поддельных подписчиков, чтобы подстегнуть предполагаемую популярность аккаунта в Twitter, довольно легко и дешево. Это равносильно добавлению узлов и связей для повышения степени-на-входе узла в сети подписчиков аналогично тому, как можно создавать поддельные веб-сайты и связи для подстегивания ранга PageRank веб-сайта. Диффузионными сетями также манипулируют посредством *социальных ботов*, обманных фейковых учетных записей, которые выдают себя за пользователей. Боты могут использоваться для создания поддельных твитов и создания видимости предвыборных кампаний, или *астротурфов*¹. Благодаря этому они могут обманывать как пользователей-людей, так и алгоритмы ранжирования, эффективно перехватывая общественное внимание. Ботов также можно использовать для ретвита некоторых сообщений, усиливая их восприятие и подстегивая их распространение. Рисунок 4.12 и изображение на обложке книги иллюстрируют то, как боты используются для усиления распространения дезинформации и манипулирования публичными дебатами. На самом деле боты могут получать значительное влияние. Рисунок 4.13 иллюстрирует еще одну форму сетевых манипуляций с использованием поддельных ответов и упоминаний. Благодаря им можно нацеливаться на влиятельных и популярных пользователей, таких как журналисты и политики, и подвергать их воздействию дезинформации в надежде, что они распространят ее среди своих многочисленных подписчиков. Еще одну сеть диффузии дезинформации, которой манипулируют влиятельные боты, можно увидеть на рис. 7.1 (глава 7)². Хотя наши примеры сосредоточены на Twitter, другие соци-

¹ Изначально астротурф (astrturf) – это искусственное травяное покрытие для спортивных площадок. В черном пиаре это искусственно организованная предвыборная кампания или общественное мнение. – *Прим. перев.*

² Роль ботов в интерактивных диффузионных сетях из Twitter можно разведать с использованием инструмента Ноаху из Наблюдательного пункта за социальными медиа по адресу hoaxy.iuni.iu.edu.

ально-медийные платформы, такие как Facebook, Instagram и WhatsApp, также эксплуатируются для распространения дезинформации.

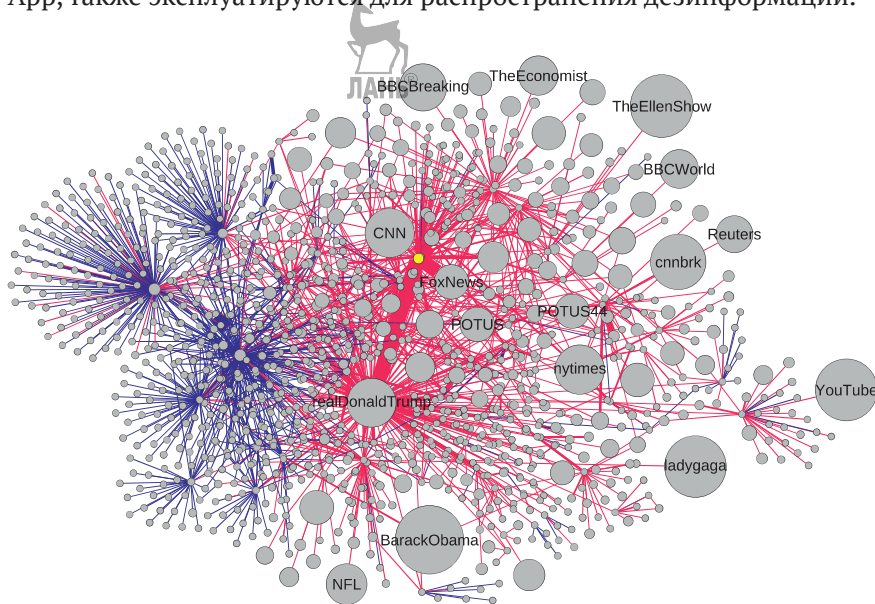


Рис. 4.13 Часть диффузионной сети распространения поддельного новостного сообщения, в котором утверждалось о массовом мошенничестве с избирателями со стороны нелегальных иностранцев на выборах в США в 2016 году. Несмотря на отсутствие подтверждающих данных и опровержение лицами, проверяющими факты на достоверность (фактчекерами), статья была опубликована в Twitter более 18 000 раз. В этой визуализации размер узла представляет число подписчиков учетной записи; связи иллюстрируют то, как статья распространяется с помощью ретвитов или цитируемых твитов (синий), а также ответов или упоминаний (красный); а ширина связи отражает ее вес – число ретвитов, цитирований, ответов и упоминаний между двумя аккаунтами. Малый желтый узел рядом с центром – это бот, систематически публикующий дезинформацию в Twitter в ответах на сообщения, в которых упоминается президент США. Результирующие упоминания генерируют толстую красную связь, соединяющую бота и хештег @realDonaldTrump. Изображение из Шао и соавт. (2018b) по лицензии CC BY 4.0



4.6. Сети совместной встречаемости

В этой главе мы уже обсуждали несколько примеров взвешенных сетей. Информационные, коммуникационные, транспортные, биологические и даже социальные сети часто имеют взвешенные связи. Взвешенные сети могут возникать еще по одному пути, а именно через отношения между более чем одним типом сущности.

Простейшим случаем является направленная сеть, в которой каждая связь имеет источниковый и целевой узел. Вообразите, что все источниковые узлы помещены на одну сторону, а все целевые узлы – на другую (узлы, которые являются как источниками, так и целями, могут дублироваться и появляться с обеих сторон). В качестве кон-

кретного случая сошлемся на сеть цитирования на рис. 4.1. *Связь цитирования* – это связь, соединяющая два разных типа сущностей, цитирующую статью (источник) и цитируемую статью (цель). Теперь мы можем построить новую сеть среди статей в каждой из этих групп. Две статьи *цитируются совместно*, если есть какая-то одна или несколько статей, которые цитируют их обе. Численность их совместных цитирований – это число статей, цитирующих их обе. Схожим образом две статьи имеют *совместные упоминания источников*, если обе из них цитируют какую-то одну или несколько статей совместно. Численность их совместных упоминаний источников – это число статей, цитируемых обоими. Сети *совместного цитирования* и *совместного упоминания источников* являются ненаправленными взвешенными сетями, в которых связи взвешиваются соответственно по численностям совместного цитирования и совместного упоминания источников. Они часто используются для отыскания взаимосвязанных наборов публикаций.

Существует много ситуаций, в которых отношения между двумя различными типами сущностей естественным образом представляются *двудольными сетями*, где каждая связь соединяет два узла разных типов. Один из примеров тому мы обсуждали в главе 0; это отношения между актерами/актрисами и кинофильмами, в которых они снимались. Как показано на рис. 0.2, из этой двудольной сети можно создать сеть между актерами/актрисами, которые снимались в кинофильмах вместе. Связи в сети киносозвездий могут взвешиваться по числу кинофильмов, в которых два человека снялись вместе. Генерирование взвешенных сетей из двудольных сетей в таком ключе, именуемое *проекцией*, является распространенной практикой, и результирующие взвешенные сети называются *сетями совместной встречаемости*, поскольку связи представляют две сущности одного типа, которые «встречаются» вместе в ассоциации с одной или несколькими сущностями еще одного типа. Другие распространенные примеры сетей совместной встречаемости включают студентов, посещающих одни и те же занятия; товары, такие как кинофильмы и книги, приобретаемые одними и теми же покупателями; и страницы, которые понравились пользователям Facebook совместно. Каждый раз, когда вы «лайкаете» или делитесь чем-то на социально-медийной платформе, вы создаете связь между собой и объектом (рис. 4.14(a)). Платформа делится этими связями с друзьями, а также агрегирует их среди миллионов людей, производя массовые сети совместной встречаемости (рис. 4.14(b)), которые можно использовать для генерирования рекомендаций и нацеленной рекламы.

Двудольная сеть, конечно же, может иметь взвешенные связи. Системы рейтингового оценивания имеют веса, которые представляют, к примеру, то, насколько человек получает удовольствие от кинофильма или мобильного приложения. Еще одним источником взвешенных двудольных сетей является *социальное тегирование*: пользователь аннотирует ресурс (идентифицируемый URL-адресом) одной или не-

сколькими метками, или тегами. Сайты обмена, такие как [Flickr.com](https://www.flickr.com) и [YouTube.com](https://www.youtube.com) популяризировали социальное тегирование в отношении изображений, кинофильмов и других медиа. Элементарным конструктом представления в форме социального тегирования является *триплет* (u, r, t) , где пользователь u помечает ресурс r тегом t . Ресурсом может быть медиаобъект, научная публикация, веб-сайт, новостная статья и т. д. В социально-медийной платформе тегирование может подразумеваться. Например, многие пользователи Twitter имеют связи, указывающие на новостные статьи или записи в блогах, а также помечают свои посты хештегами. Из таких твитов можно извлекать триплеты – по одной для каждой пары связь–хештег, с автором поста в качестве пользователя. На рис. 4.15(а) показан набор триплетов. При агрегировании по многочисленным пользователям такой набор называется *фольксономией*, потому что это таксономия, которая возникает благодаря многочисленным людям¹. Фольксономия бывает полезна для поиска или предложения веб-сайтов.

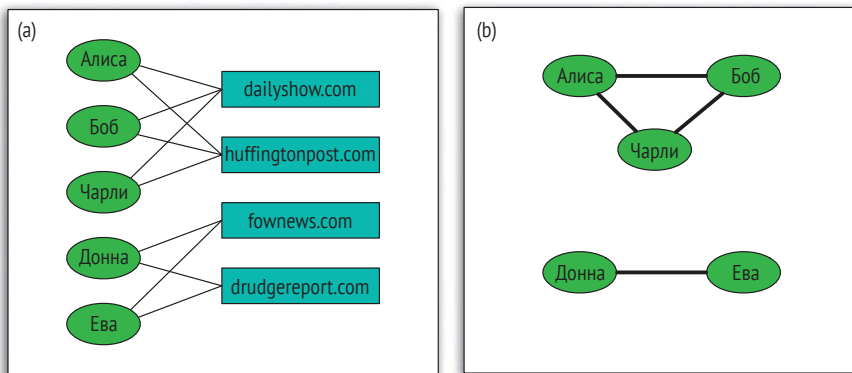


Рис. 4.14 (а) Двудольная сеть, индуцированная отношениями «лайкания». (б) Сеть совместной встречаемости, выведенная из проецирования сети «лайкания» на пользовательские узлы

Из фольксономии мы можем извлекать двудольную сеть, проецируя триплеты на два типа узлов. Результирующие связи соединяют узлы только одного типа (к примеру, теги) с узлами еще одного типа (к примеру, ресурсы). Поэтому мы можем трактовать эти связи как направленные, как показано на рис. 4.15(б). Связи также могут иметь веса, представляющие, к примеру, число пользователей, аннотировавших конкретный ресурс определенным тегом (рис. 4.15(б)). Благодаря этому вместо того, чтобы терять информацию о пользователях, мы закодировали ее в меру надежности связи.

¹ Фольксономия (*folksonomy* от англ. *folks* – народ и *taxonomy* – таксономия) – это практика совместной категоризации информации (ссылок, фото, видеоклипов и т. п.) посредством произвольно выбираемых тегов. – Прим. перев.

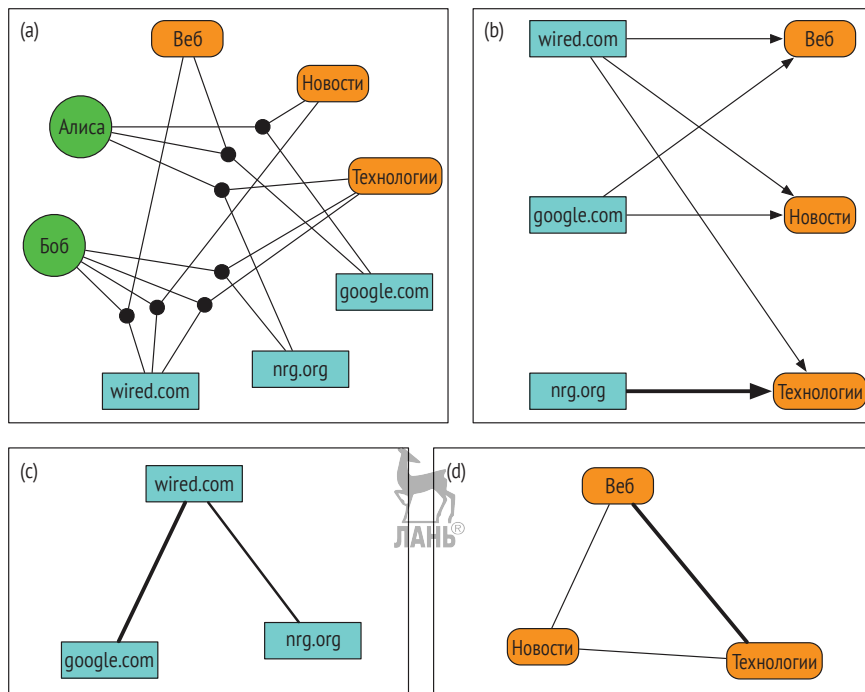


Рис. 4.15 Пример фольксономии и производных двудольных сетей и сетей совместной встречаемости. (а) Два пользователя (Алиса и Боб) аннотируют три ресурса ([npr.org](#), [wired.com](#), [google.com](#)) с использованием трех тегов (новости, веб, технологии), в результате чего получилось семь триплетов. (б) Проецируя фольксономию на ресурсы и теги, мы получаем двудольную сеть. Веса связей соответствуют числам триплетов, или числам пользователей. Связь из [npr.org](#) в сторону новостей имеет более крупный вес, потому что оба пользователя согласны с этой аннотацией. (с) Сеть совместной встречаемости ресурсов. Ресурсы [wired.com](#) и [google.com](#) более похожи потому что они сочетаются с двумя тегами, веб и технологии. (d) Сеть совместной встречаемости тегов. Связь между вебом и технологиями имеет более крупный вес из-за сходства в своих ресурсах: два тега встречаются совместно с двумя ресурсами, [wired.com](#) и [google.com](#)

Из двудольной сети, как обсуждалось ранее, мы можем создавать сети совместной встречаемости путем дальнейшего проецирования одного типа узлов, основываясь на совместных соседях другого типа. Например, на рис. 4.15(с) мы проецируем на сеть ресурсов, а на рис. 4.15(д) мы проецируем на сеть тегов¹. В этих сетях совместной встречаемости направление связей теряется, но мы можем сохранять весовую информацию путем сравнения того, как два тега соединены с ресурсами.



¹ Интерактивные диффузионные сети из Twitter можно разведывать, используя инструмент из Наблюдательного пункта за социальными медиа (Observatory on Social Media) по адресу <https://osome.iu.edu/tools>.



Один из подходов состоит в том, чтобы, к примеру, представлять теги в форме вектора ресурсов $\vec{t} = \{w_{t,1}, \dots, w_{t,n_r}\}$, где $w_{t,r}$ – это число людей, помечающих ресурс r тегом t , и n_r – суммарное число ресурсов. Таким образом, каждый элемент тегового вектора является весом, представляющим ассоциацию ресурса с тегом: веса связей на рис. 4.15(б). Тогда мы можем вычислить *косинусное сходство* между двумя теговыми векторами (см. вставку 4.1) и использовать его в качестве веса связи совместной встречаемости. Если два тега используются для аннотирования ресурсов одинаковыми способами, то их вес является высоким; если они никогда не встречаются совместно, то вес равен нулю, и узлы тегов не связаны.

4.7. Весовая гетерогенность

Во взвешенной сети веса связей могут нести важную информацию о моделируемых сетью процессе или отношениях. Связи с разными весами могут представлять очень разные ассоциации. Для того чтобы разведать это различие, давайте рассмотрим еще один класс взвешенных сетей: сети, которые улавливают различные виды трафика. Транспортные сети включают в себя сети авиационных перевозок и другие транспортные сети, где веса представляют пассажиров или рейсы между аэропортами или автомобилями между перекрестками; интернет, где веса обозначают пакеты или биты данных между маршрутизаторами; и «Википедию», где веса представляют клики (т.е. нажатия) между статьями. Давайте сосредоточимся на случае трафика Всемирной паутины, аналогичного «Википедии», но расширенном на все веб-сайты.

4.7.1. Трафик Всемирной паутины

Данные о трафике Всемирной паутины могут собираться браузерами (или панелями инструментов браузера, или расширениями), которые записывают данные о кликах и передают их на сервер сбора данных. В качестве альтернативы интернет-провайдер может отслеживать пакеты, несущие HTTP/HTTPS-запросы, которые включают целевой хост и страницу, а также источник URL-адрес, именуемый *рефером* (или *направляющим запрос источником*). Оба этих метода сбора данных несколько искажены; в первом случае мы наблюдаем трафик, генерируемый только пользователями браузеров, оснащенных программно-информационным обеспечением для мониторинга трафика, во втором можем видеть пакеты, проходящие только через маршрутизаторы интернет-провайдера. Тем не менее оба метода позволяют собирать очень большие коллекции данных о веб-трафике.

Можно подсчитывать клики между отдельными страницами или, как мы сделаем дальше, рассматривать совокупный трафик на уровне целых веб-сайтов, идентифицируемых по их уникальным сетевым именам, например en.wikipedia.org, google.com и www.indiana.edu.

Инспектирование сетей веб-трафика позволяет нам изучать распределение значений трафика веб-сайта (суммарное число кликов по направлению к веб-сайту), выражаемое силой-на-входе узла, и трафика связей (суммарное число кликов по гиперсвязи), выражаемое весом связи. Утяжеленные хвосты на рис. 4.16 показывают, что оба этих распределения чрезвычайно гетерогенны. Большинство веб-сайтов получает очень мало кликов, тогда как некоторые получают массивный трафик. Схожим образом многие связи почти никогда не кликаются, тогда как другие кликаются постоянно.

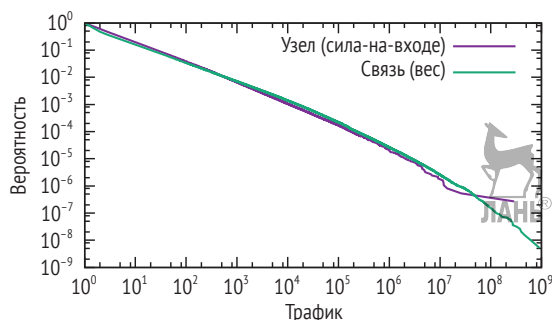


Рис. 4.16 Кумулятивные распределения значений силы-на-входе узлов (трафик веб-сайтов) и веса связей в сети веб-трафика. В период с 2006 по 2007 год в Университете Индианы был собран почти миллиард кликов, представляющих совокупную активность около 100 000 анонимных пользователей, проявляемую при навигации по Всемирной паутине. Результирующая сеть насчитывает около 4 млн сайтов и 11 млн взвешенных направленных связей

Вспомните из раздела 4.3, что идея модели PageRank заключалась в том, чтобы симулировать пользователей, которые осуществляют навигацию по Всемирной паутине. Сравнивая рейтинг сетевых узлов по их силе-на-входе с рейтингом, производимым моделью PageRank, мы можем спросить, способна ли модель PageRank предсказывать трафик по структуре графа связей Всемирной паутины. Другими словами, улавливает ли модель случайного серфера агрегатный шаблон навигации, применяемый фактическими пользователями Всемирной паутины? Как оказалось, ответ – отрицательный: несмотря на аналогичные тяжелохвостные распределения (см. рис. 4.9 и 4.16), корреляция между метрикой PageRank и трафиком является довольно слабой. Следовательно, некоторые из упрощающих допущений модели PageRank, должно быть, нарушаются нашим поведением, проявляемым при навигации по Всемирной паутине.

В целях получения представления о том, какие ингредиенты случайного серфера наименее реалистичны, давайте рассмотрим соотношение между силой-на-выходе и силой-на-входе узлов. За исключением

стартового и финишного узлов навигационных сеансов, указанное соотношение должно быть равно единице, поскольку поток трафика, поступающий в узел, равен потоку, исходящему из узла. Согласно модели PageRank, телепортация не благоприятствует каким-либо узлам; каждый узел с равной вероятностью будет становиться целью случайного прыжка (места, где навигация начинается) и с равной вероятностью будет становиться источником случайного прыжка (места, где навигация заканчивается). Следовательно, даже с телепортацией модель PageRank производит соотношение между силой-на-выходе и силой-на-входе, очень близкое к единице для всех узлов. Тогда мы ожидали бы, что узкое распределение достигнет пика, сосредоточенного вокруг единицы. Но рис. 4.17 рисует совершенно иную картину, показывая огромные колебания, охватывающие многие порядки величины: некоторые узлы с гораздо большей вероятностью будут стартовыми точками навигационных сеансов, а другие с гораздо большей вероятностью будут их финишными точками. Это неудивительно; мы, как правило, начинаем серфинг с нескольких знакомых сайтов, помеченных закладками. Однако большинство сайтов не очень интересно, и поэтому, скорее всего, это будут места, где мы остановимся и отпрыгнем. Мы приходим к выводу, что случайная телепортация является нереалистичным ингредиентом модели PageRank.

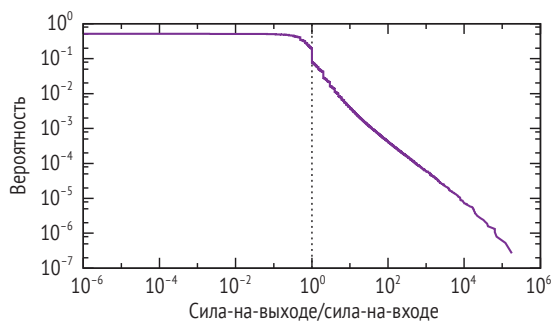


Рис. 4.17 Кумулятивные распределения соотношения между силой-на-выходе узла и силой-на-входе узла в трафике Всемирной паутины, описанном на рис. 4.16. Соотношения сил $s_{out}/s_{in} \ll 1$ указывают на сайты, в которых навигационные сеансы с большей вероятностью будут терминироваться, тогда как $s_{out}/s_{in} \gg 1$ для стартовых точек

4.7.2. Фильтрация связей

Плотные сети трудно визуализировать и изучать: они выглядят как «комки волос», и многие связи не являются значительными. По этим причинам нередко бывает полезно подрезать низковесные связи во взвешенной сети. Это особенно необходимо в сетях совместной встречаемости, поскольку многие низковесные соединения могут вызываться шумом. В качестве примера рассмотрим сеть документов или веб-страниц, в которой связи определяются текстовой похо-

жестью – ключевыми словами из текстового содержимого узлов. Пара не взаимоувязанных по теме или слабо взаимоувязанных документов, скорее всего, будут связаны, потому что у них есть несколько совместных ключевых слов. В подобных случаях нам нужен метод фильтрации таких связей и получения более разреженной сети только с содержательными соединениями.

Самый простой подход к подрезанию сети состоит в удалении всех связей с весом ниже некоторого порога. Этот метод хорошо работает во многих обстановках. Однако существует также много случаев, когда фильтрация, основанная на глобальном пороге, не работает. В целях понимания причины рассмотрим сеть с тяжелохвостным распределением значений весов связей – довольно распространенный сценарий в совместной встречаемости, сетях трафика и других взвешенных сетях (вспомните рис. 4.16). Веса настолько гетерогенны, что невозможно отыскать хороший порог: мы, скорее всего, будем сохранять несколько незначительных связей и/или разъединять много узлов с малой силой. Для малосильных узлов низковесные связи могут быть значительными, даже если связи с тем же весом могут быть незначительными для узлов с гораздо большей силой.

Для того чтобы обойти эту проблему, нам нужно использовать разные пороги для разных узлов. Одним из подходов заключался бы в определении порога относительно степени или силы каждого узла: мы могли бы оставлять только 10 % связей с наибольшим весом по каждому узлу или связи с наибольшим весом, на которые приходится 80 % силы узла. Но даже в этом случае мы не сможем быть уверены в том, что будем удерживать все значительные связи или некоторые из них, которые не являются значительными. Более принципиальный подход заключается в поиске *сетевой магистрали* (т. е. в обнаружении связей, которые несут непропорционально долю силы каждого узла). Это наиболее значительные связи, которые необходимо сохранять. Во вставке 4.3 описано, как это можно сделать. На рис. 4.18 показана магистральная сеть, выделенная из плотной сети.

Вставка 4.3

Сетевая магистраль

В сетях с широким распределением значений весов связей использовать глобальный порог для сокращения связей нецелесообразно. Вместо этого мы можем использовать колебания веса по каждому узлу для выявления связей, которые необходимо сохранять, – те, которые несут большую часть веса. Имея узел i со степенью k_i и силой s_i , давайте оценим связь относительно нулевой модели, в которой веса распределяются случайно на k_i связях, смежных с i , с ограничением, что их сумма равна s_i . Вероятность того, что связь имеет вес w_{ij} или больше в условии этой гипотезы, равна

$$p_{ij} = \left(1 - \frac{w_{ij}}{s_i} \right)^{k_i-1}. \quad (4.2)$$

Поэтому если связь ij имеет вес w_{ij} , то из уравнения (4.2) мы вычисляем вероятность p_{ij} того, что такое значение совместимо с нулевой моделью: если $p_{ij} < \alpha$, где α – это параметр, который представляет желаемый уровень значимости, то связь сохраняется, в противном случае она удаляется. Более низкие значения α приводят к более разреженным сетям, так как сохраняется меньше связей. Поскольку связь соединена с двумя узлами, мы можем получить два значения для p_{ij} , подставив силу и степень любого узла в уравнение 4.2. Затем можем использовать большее или меньшее из этих значений в зависимости от того, насколько агрессивно, больше или меньше, мы хотим выполнять подрезание. Эта процедура фильтрации связей извлекает *сетевую магистраль*, которая должна сохранять существенную структуру и глобальные свойства сети.

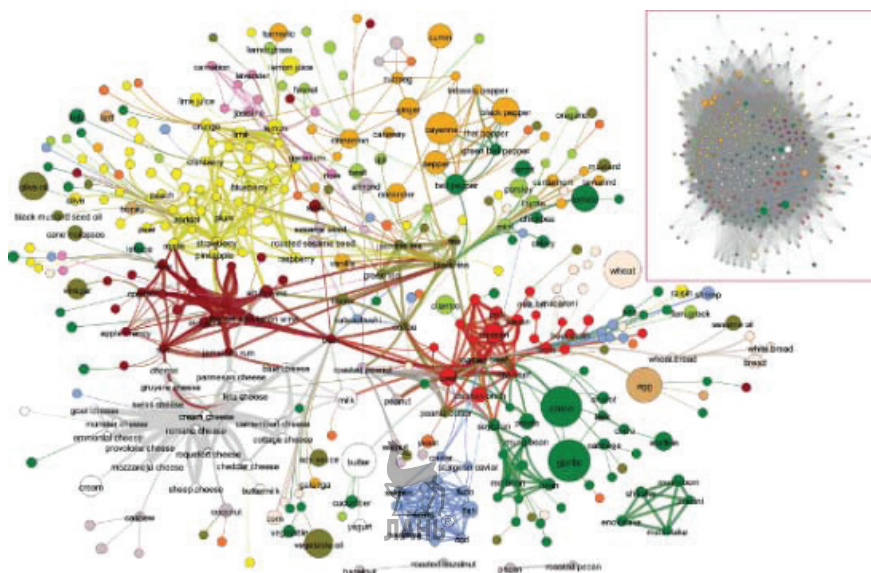


Рис. 4.18 Вкусовая сеть: каждый узел обозначает ингредиент, его цвет указывает на категорию продуктов питания, а его размер отражает распространенность ингредиентов в рецептах. Два ингредиента соединяются, если они имеют общие вкусовые соединения, при этом ширина связи представляет число общих соединений. Полная сеть показана во врезке, тогда как главное изображение визуализирует магистральную сеть с существенными связями, выявленными методом, приведенным во вставке 4.3, с использованием $\alpha = 0.04$. Изображения адаптированы из Ан и соавт. (2011) по лицензии CC BY 4.0

4.8. Резюме

Информационные сети, такие как «Википедия» и Всемирная паутина в целом, имеют направленные связи. То же самое касается и многих биологических сетей, таких как мозг; коммуникационных сетей, включая электронную почту и интернет; сетей транспортных

перевозок, таких как авиаперелеты; и некоторых социальных медиа, в частности Twitter. Связи часто взвешиваются для представления, например, интенсивности взаимодействия или сходства между узлами. Веса сети также могут использоваться для того, чтобы представлять проходящий между узлами трафик: клики, сообщения, пакеты, путешественников, численность ретвитов и т. д. Мы развели признаки направленных и взвешенных сетей, сосредоточив внимание на нескольких случаях.

1. Всемирная паутина образует огромную информационную сеть практически с бесконечным числом страниц, соединенных гиперсвязями. Браузеры используют HTTP-протокол для навигации по связям и скачивания содержимого страниц. Это содержимое обычно выражается на языке HTML, который задает то, как будет представляться насыщенное содержимое – текст и встроенные медиа.
2. Мы изучаем структуру Всемирной паутины и хостовых графов, в которых каждый узел представляет соответственно страницу или площадку во Всемирной паутине (веб-сайт), используя данные, собранные обходчиками паутины – программами, которые осуществляют автоматическую навигацию по веб-страницам и позволяют нам реконструировать крупные выборки сети. Всемирная паутина имеет тяжелохвостное распределение значений степени-на-входе и ультракороткие пути, чему способствуют чрезвычайно популярные хабовые страницы. Она также имеет очень крупную сильно связную компоненту внутри гигантской компоненты.
3. Мы можем представлять документы, такие как веб-страницы, в форме высокоразмерных словарных векторов и использовать косинус между этими векторами для измерения сходства текста между страницами. Благодаря этому мы можем изучать тематическую локальность, отношение между сетевыми соединениями и содержимым страницы. Поскольку авторы, как правило, ссылаются на взаимоувязанные страницы, Всемирная паутина имеет такую кластерную структуру, при которой страницы, расположенные в сети на малом расстоянии друг от друга, будут с большей вероятностью похожими и семантически взаимоувязанными.
4. Метрика PageRank – это мера узловой центральности, основанная на модели случайного блуждания при навигации по Всемирной паутине, модифицированная случайными прыжками. Хотя в реальности люди и не осуществляют навигацию по страницам паутины в такой случайной манере, метрика PageRank обычно применяется для измерения престижности страниц паутины. Алгоритм PageRank работает в любой направленной сети, но особенно важен из-за его роли в ранжировании результатов поисковой машины. Это был ключевой ингредиент поисковика Google, когда он был представлен.

5. Сети диффузии информации возникают, когда мы делимся содержанием в социальных сетях, например, путем ретвита связей, изображений и хештегов. Результирующие каскадные сети позволяют нам отслеживать распространение новостей, идей, убеждений и даже дезинформации. Размер и структура этих графов помогают нам идентифицировать вирусные концепции. Используя силу на-выходе и силу на-входе узла, мы можем характеризовать роли людей, которые производят и потребляют информацию. Узлы с высокой силой, в особенности относительно их степени, сигнализируют об учетных записях, которые являются более активными или влиятельными. Для манипулирования этими сетями могут использоваться социальные боты.
6. Взвешенные сети нередко являются результатом двудольных графов. Вес связи между двумя узлами **a** и **b** одного и того же типа является мерой численности узлов другого типа, которые соединены как с **a**, так и с **b**. Такие сети возникают из отношений совместной встречаемости, таких как совместные цитирования / совместные ссылки на источники, продуктовые рекомендации и сходство слов/тегов.
7. Взвешенные сети, такие как сети, полученные на основе данных о трафике и совместной встречаемости, бывают очень плотными. Поэтому часто бывает необходимо подрезать граф, отфильтровывая низковесные связи. Однако взвешенные сети нередко имеют тяжелохвостные распределения значений весов. В этих случаях использование глобального весового порога изолирует большинство узлов. Определяя локальный весовой порог для выявления статистически значимых связей по каждому узлу, мы можем извлекать магистраль гетерогенной взвешенной сети.

4.9. Дальнейшее чтение

О видении, дизайне и истории Всемирной паутины можно почитать в книге, написанной в соавторстве с ее изобретателем (Бернерс-Ли и Фишетти, 2000). В целях более глубокого ознакомления с принципом работы поисковых машин обратитесь к учебникам по информационному поиску Баззы-Йейтса и Рибейро-Нето (2011) или Мэннинга и соавт. (2008). Лью (2011) рассказывает о том, как добывать данные из сетей связей, контента и пользования Всемирной паутины; в главе 8 этой книги основное внимание уделяется обходчикам Всемирной паутины.

Альберт и соавт. (1999) впервые проанализировали среднюю длину пути Всемирной паутины в 1999 году, основываясь на обходе веб-сайтов Университета Нотр-Дам. В то время считалось, что Всемирная паутина содержит около миллиарда страниц, поэтому авторы экстра-

полировали логарифмическое соответствие между средней длиной пути и размером (под)сети, оценив, что диаметр Всемирной паутины составлял 19 связей. В следующем году Бродер и соавт. (2000) сообщили о первом систематическом исследовании структуры Всемирной паутины. Они измерили среднюю длину пути на гораздо более крупном обходе Всемирной паутины с $N = 10^8$ страницами в грубом соответствии с более ранним предсказанием. Более поздний анализ гораздо более масштабного обхода страниц Всемирной паутины был проведен Мойзелем и соавт. (2015)

Барабаши и Альберт (1999) сообщили о первых подтверждающих данных о тяжелохвостном распределении значений степени-на-входе страниц Всемирной паутины. Бродер и соавт. (2000) позже подтвердили это на основе более крупного обхода. Они (2000) также проанализировали структуру «галстук-бабочка» направленного графа Всемирной паутины. Серрано и соавт. (2007) показали, что относительные размеры самой крупной сильно связной компоненты, компоненты-на-входе и компоненты на-выходе зависят от того или иного обходчика, используемого для реконструкции графа Всемирной паутины.

Дэвисон (2000) измерил тематическую локальность во Всемирной паутине путем сравнения содержимого пар страниц, отбираемых случайно, связанных общим предшественником (сестринскими страницами) и соединенных гиперсвязью. Менцер (2004) расширил этот анализ, выполнив обход сперва в ширину, чтобы проследить затухание содержательного и семантического сходства для страниц в пределах определенного расстояния друг от друга (рис. 4.7).

Идея использования мер сетевой центральности для ранжирования результатов поисковых машин была задумана Маркиори (1997). Год спустя Брин и Пейдж (1998) представили поисковик Google и дали описание того, как метрика PageRank использовалась для ранжирования результатов поиска. Оказывается, такая же мера центральности была предложена 50 годами ранее Сили (1949) как способ зондирования важности человека в социальной сети. Родственная мера авторитетности, основанная на двудольном представлении графа Всемирной паутины, была предложена Кляйнбергом (1999). Фортунато и соавт. (2007) показали, что средний балл метрики PageRank в отношении узлов с равной степенью-на-входе пропорционален степени-на-входе. Обзор математики, лежащей в основе метрики PageRank, смотрите в работе Глейх (2015).

Докинз (2016) ввел понятие мема для обозначения единицы информации, мнения или поведения, которые могут передаваться от человека к человеку. Это было предшественником изображений, хештегов и связи, которые теперь распространяются в социальных медиа. Гозль и соавт. (2015) предложили определение структурной вирусности для мемов и метод реконструирования ретвитных каскадных сетей в Twitter. Изучая эти диффузионные сети, Ша и соавт. (2010) пока-

зали, что наличие высокой степени (много подписчиков) не является единственным фактором, который воздействует на влияние узла.

Сильная поляризация коммуникационных сетей в Twitter была замечена при анализировании диффузионных сетей политических хештегов с сегрегированными консервативным и прогрессивным сообществами (Коновер и соавт., 2011b). Аналогичным образом Шао и соавт. (2018a) обнаружили, что сегрегированные сообщества обмениваются дезинформацией при сопоставлении со статьями, проверяющими факты на достоверность. Сообщество социальной сети с гомогенными мнениями, изолированное от других взглядов, иногда называется эхокамерой (Санштайн, 2001) или фильтренным пузырем (Паризер, 2011).

Раткевич и соавт. (2011) наблюдали самые ранние случаи, когда фальшивые новостные сайты продуцировали дезинформацию через социальные сети. Факторы, влияющие на вирусное распространение дезинформации в социальных сетях, являются предметом тщательного исследования (Лазер и соавт., 2018); они включают новизну (Восути и соавт., 2018) и усиление социальными ботами (Феррара и соавт., 2016; Шао и соавт, 2018b).

Мейс и соавт. (2008) собрали массивные данные о веб-кликах, чтобы реконструировать крупную сеть трафика Всемирной паутины, раскрыв ограничения метрики PageRank как модели серфинга по Всемирной паутине. Более совершенные модели учитывают установку закладок популярных стартовых узлов, возврат к предыдущему состоянию (или вкладки браузера) и тематическую локальность (Мейс и савт., 2010). Мейс и соавт. (2008) также показали, что распределение значений весов связей имеет утяжеленный хвост. Серрано и соавт. (2009) представили метод извлечения магистрали сетей с гетерогенными весами.

Упражнения

4.1 Зайдите на scholar.google.com и отыщите публикации по интересующей вас теме. Выберите две статьи из списка результатов поиска.

1. Какова степень-на-входе каждой из двух статей в сети цитирования?
2. Для каждой из двух статей просмотрите списки, в которых они цитируются в других статьях (нажмите **Cited by...** (Цитируется...)). Рассчитайте совместное цитирование между двумя статьями. (Подсказка: бывает утомительно, если выбрать две статьи со слишком большим числом цитирований.)
3. Какова степень-на-выходе каждой из двух статей в сети цитирования? (Подсказка: для того чтобы ответить на этот во-

прос, документы должны быть в распоряжении для доступа или скачивания.)

4. Скачайте две статьи и проанализируйте списки библиографических ссылок на источники. Рассчитайте взаимное упоминание источников (также именуемое *библиографическим сопряжением*) между двумя документами.

4.2 Зайдите на статью «Википедии» по «network science» (науке о сетях) (en.wikipedia.org/wiki/Network_science).

1. Какова степень-на-выходе этой страницы в сети «Википедии»? (*Подсказка: для упрощения в этом упражнении вы можете сосредоточиться на исходящих связях в разделе «Смотрите также», который обычно содержит несколько связей, указывающих на другие статьи «Википедии»; если этот раздел отсутствует, то вы можете допустить $k_{out} = 0$.*)
2. Посетите узлы-преемники узла «network science» в сети «Википедии» и сообщите, сколько исходящих связей из этой статьи являются взаимными.
3. Постройте эгосеть узла «network science» и отыщите самую крупную сильно связную компоненту. Вспомните, что эгосеть состоит из одного узла (эго), всех его соседей и всех связей между ними (см. рис. 2.8). Определение направленной эгосети является аналогичным путем замены соседей преемниками.
4. Какой узел в эгосети «network science» имеет максимальную степень-на-выходе?
5. Какой узел в эгосети «network science» имеет максимальную степень-на-входе?

4.3 Рассмотрите эгосеть «Википедии» «network science», построенную в предыдущей задаче. Представьте каждый из этих узлов в виде списка категорий – они находятся в нижней части каждой статьи «Википедии»; например Network theory (Теория сетей). Для каждой пары узлов вычислите косинусное сходство между категориальными векторами. (*Подсказка: список – это вектор, в котором вес каждой категории равен единице, а вес любой категории, отсутствующей в списке, равен нулю.*)

1. Какие две статьи из вашей выборки похожи друг на друга больше всего? Каково значение косинусного сходства?
2. Какие две статьи из вашего образца похожи друг на друга меньше всего? Каково значение косинусного сходства?
3. Дают ли ваши измерения подтверждающие данные о тематической локальности? Почему да или почему нет? (*Подсказка: если вы игнорируете направления связей, то любые два узла найдутся либо на расстоянии единицы друг от друга, если они соединены, либо на расстоянии двойки, через эго. Сравните среднее сходство пар статей в этих двух группах.*)

- 4.4** Рассмотрите малую сеть на рис. 4.19. Инициализируйте метрику PageRank каждой страницы значением $R_0 = 1/3$. Примените уравнение (4.1) без телепортации ($\alpha = 0$) для расчета значений метрики PageRank на следующей итерации ($t = 1$). Продолжайте обновлять значения до тех пор, пока они не сойдутся, – допустите, что значения сошлись, когда нет изменений в третьей десятичной цифре метрики PageRank каждого узла. (Подсказка: убедитесь, что при вычислении новых значений вы используете значения из предыдущей итерации; например, используйте первоначальные значения ($t = 0$) при расчете значений $t = 1$.) Сходятся ли значения после скольких итераций? Каковы окончательные значения метрики PageRank?

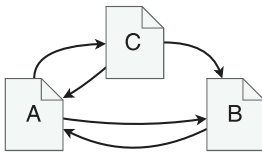



Рис. 4.19 Направленная сеть, представляющая малый веб-сайт с тремя страницами и их гиперсвязями

- 4.5** Повторите предыдущее упражнение с использованием сети на рис. 4.19, но на этот раз используйте телепортационный параметр $\alpha = 0.2$. Каким будет t при схождении, и какими будут значения метрики PageRank?
- 4.6** Зайдите на демонстрационную страницу PageRank по адресу go.iu.edu/pagerank и вставьте ряд узлов (имен), текстовых атрибутов узлов (цвета) и связей. Демостраница вычисляет метрику PageRank, и вы можете использовать ее для измерения популярности каждого узла. Понаблюдайте, как значения меняются с добавлением новых узлов и связей¹.
1. Кто является самым популярным? Что можно сделать, чтобы увеличить свой ранг PageRank?
 2. Поисковая функциональность демостраницы работает как суперупрощенная поисковая машина. Выполните поиск каких-нибудь цветов. Каким образом сходство между вашим запросом и текстовыми атрибутами узла влияет на рейтинг узла? Как метрика PageRank влияет на рейтинг?

¹ *Примечание для преподавателя:* это упражнение выполняется увлекательнее в большой учебной группе. Каждый участник может использовать свой собственный ноутбук, вместо своего имени использовать анонимный псевдоним, вводить свои любимые цвета и связываться со своими друзьями. Как вариант, проведите упражнение в учебной группе с дополнительным зачетом, назначаемым лучшим ученикам по результатам группового занятия. Скажите своим студентам, чтобы они учитывали альянсы, и будьте безжалостны!

- 4.7 Скачайте набор данных «Википедии» (файл graphml) из репозитория книги на GitHub¹ в папке enwiki_math. Используйте библиотеку NetworkX для загрузки файла в виде направленной сети (орграфа), затем выполните алгоритм PageRank для вычисления метрики PageRank каждой статьи.
1. Каковы 10 верхних статей по метрике PageRank?
 2. Сравните 10 верхних статей по метрике PageRank с 10 верхними статьями по степени-на-входе. Являются ли они одинаковыми? Почему да, или почему нет?
- 4.8 Ознакомьтесь с учебным материалом главы 4 в репозитории книги на GitHub.
- 4.9 Протестируйте Парадокс дружбы (обсуждаемый в главе 3) на Twitter. Обратитесь к главе 4 учебного материала в отношении использования API Twitter. Поскольку Twitter является направленной сетью, мы можем сформулировать Парадокс дружбы в терминах степени-на-входе/степени-на-выходе преемников/предшественников узла. Одна из версий состоит в постановке вопроса: «Действительно ли у ваших друзей (людей, на которых вы подписались) в среднем подписчиков больше, чем у вас?» В данном случае мы хотим измерить степень-на-входе ваших узлов-преемников в сети подписчиков. Если вы не являетесь пользователем Twitter, то можете ответить на этот вопрос, используя чей-либо другой дескриптор Twitter, например @clayadavis.
1. Допустим, что user является ответом на запрос к users/show.json о декстрипторе @clayadavis. Какое из следующих ниже значений даст вам число людей, на которых подписан @clayadavis?
 - a. `user['friends']['count']`
 - b. `user['friends_count']`
 - c. `user['followers']['count']`
 - d. `user['followers_count']`
 2. Подтвердите, что Парадокс дружбы соблюдается в вашем случае, рассчитав среднее число подписчиков ваших друзей и сравнив его с вашим числом подписчиков. Если ваша учетная запись имеет больше чем максимальное число друзей, которое может быть возвращено API Twitter одним запросом (200 на момент написания книги), то вам потребуется более одного вызова API для получения полного списка друзей. Обратитесь к учебному материалу в отношении использования курсора для получения нескольких страниц результатов. Каково среднее число подписчиков среди всех ваших друзей?

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

- 
3. Математически Парадокс дружбы делает утверждение о средних значениях: «в среднем у ваших друзей подписчиков больше, чем у вас». Другим было бы утверждение «у большинства ваших друзей подписчиков больше, чем у вас». Это второе утверждение касается медианы, а не среднего значения, и на самом деле является более сильным утверждением (с меньшей вероятностью будет истинным). Является ли оно здесь истинным? Действительно ли у большинства ваших друзей подписчиков больше, чем у вас? В целях ответа на этот вопрос измерьте среднее число подписчиков среди всех ваших друзей.
 4. Каково экранное имя пользователя с наибольшим числом подписчиков среди ваших друзей?
- 4.10 Постройте ретвитную сеть для #RepealThe19th, спорного хештега, использовавшегося во время президентской кампании 2016 года в США для пропаганды отмены 19-й поправки к Конституции США, которая предоставляет женщинам право голоса. Данные для этого упражнения представлены в каталоге datasets репозитория книги на GitHub. Файл с именем `repealthe19th.json.gz` содержит 23 343 твита, включающих указанный хештег. Каждая строка представляет JSON-объект твита. После разбора файла вы должны убедиться, что у вас есть именно такое число твитов; отклонение от этого числа будет указывать на ошибки синтаксического разбора, которые могут повлиять на ваши ответы. Обратитесь к главе 4 учебного материала по использованию этих твитов для строительства ретвитной сети, чтобы ответить на следующие ниже вопросы. Имейте в виду следующее. (i) Направление связи следует информационному потоку: если Алиса ретвитит Боба, то существует связь от Боба к Алисе. (ii) Удалите самонаправленные циклы; это можно сделать после создания сети, либо модифицируйте исходный код создания сети, чтобы не добавлять их вообще. Это определенно повлияет на некоторые ответы.
1. Сколько узлов в ретвитной сети?
 2. Сколько связей в ретвитной сети?
 3. Каково экранное имя узла с наибольшей силой-на-выходе в сети? Какова его сила-на-входе?
 4. Каково экранное имя узла со второй по величине силой-на-выходе в сети?
 5. Каково экранное имя узла с наибольшей силой-на-входе в сети? Какова его сила-на-выходе?
 6. Опишите, что вам говорят значения силы-на-выходе и силы-на-входе этих учетных записей об их онлайн-поведении.
 7. Каков ID самого ретвитируемого твита? Используйте атрибут `id_str`; во время работы с файлами JSON это обычно является хорошей практикой вследствие того, что идентифи-

каторы твитов имеют размер 64 бита. Обратите внимание, что, получив ИД твита, вы можете увидеть его, посетив следующий URL-адрес в своем браузере: https://twitter.com/user/status/<ИД_твита>, заменив <ИД_твита> числовым ИД твита.

8. Сколько узлов в ретвитной сети имеют нулевую силу-на-выходе?
 9. Что из следующего ниже описывает лучше всего, что значит для узла иметь нулевую силу-на-выходе в этой сети? Каждое из утверждений относится только к выборке твитов, которые мы использовали для строительства этой сети.
 - a. Пользователь не производил твитов.
 - b. Пользователь больше никого не ретвитил.
 - c. Пользователь не ретвитился никем.
 - d. У пользователя нет подписчиков.
 - e. Пользователь не подписан на других пользователей.
 10. Что из перечисленного ниже описывает связность этой сети лучше всего?
 - a. Сильно связанная.
 - b. Слабо связанная.
 - c. Связная.
 - d. Несвязная (несоединенная).
 11. Сколько узлов находится в крупнейшей слабо связанной компоненте этой сети?
- 4.11** Постройте ретвитную сеть для хештега, представляющего интересующую вас тему. Обратитесь к учебному материалу главы 4 по использованию API Twitter. Используйте API поиска в Twitter, чтобы получить последние твиты об указанном хештеге. Убедитесь, что в поисковых параметрах у вас есть 'result_type': 'recent'. Извлеките не менее 1000 твитов, соответствующих вашему поисковому запросу; если их не так много, то начните сначала и поищите что-нибудь другое. Поскольку это число превышает максимальное число твитов, которые могут возвращаться из API Twitter с помощью одного поискового запроса (100 на момент написания книги), вам нужно будет задействовать страничную разбивку. Наконец, постройте ретвитную сеть из этих твитов.
1. Начертите полученную ретвитную сеть. Для того чтобы сделать ее информативной, следуйте следующим далее рекомендациям. (i) Размеры узлов должны быть пропорциональными их степени-на-выходе. (Подсказка: обратитесь к упражнению 3.6 о том, как получать последовательность степеней.) (ii) Размер связей должен соответствовать числу твитов между двумя пользователями. (Подсказка: используйте параметр ширины (width) чертежа (draw). Значением этого параметра должен быть список весов ребер в том же порядке, в каком заданы ребра графа) (iii) Узлы-одиночки и самона-

- правленные циклы должны быть удалены. (iv) Используйте свое суждение о том, показывать ли только самую крупную связную компоненту или же все компоненты.
2. Каково экранное имя самого ретвитируемого пользователя? Также сообщите о хештеге, который вы использовали.
 3. В нескольких предложениях расскажите о некоторых интересных наблюдениях об этой ретвитной сети.
- 4.12** Проанализируйте наборы данных взвешенных сетей, имеющиеся в репозитории книги на GitHub, чтобы изучить соотношение между степенью и силой. В случае ненаправленных сетей измерьте коэффициент корреляции Пирсона между степенью и силой по всем узлам. В случае направленных сетей сделайте то же самое для степени-на-входе/на-выходе и силы-на-входе/на-выходе. Обладают ли узлы с высокой степенью также большой силой?
- 4.13** Рассмотрите ретвитную сеть из одного из предыдущих упражнений, где веса связей представляют число ретвитов. Подрежьте сеть, удалив связи с весами ниже порога ω .
1. Вспомните определение плотности для направленной сети из главы 1. Нарисуйте график, показывающий уменьшение плотности сети (по оси y) как функции от весового порога ω (по оси x).
 2. Насколько плотность уменьшается, когда вы применяете порог $\omega = 3$ ретвитов?
 3. Каково значение ω , такое что плотность сокращается ниже половины ее первоначального значения?





Модель: упрощенное описание, в особенности математическое, системы или процесса как вспомогательное средство для проведения расчетов и предсказаний.

Мы видели, что реально существующие сети многих разных типов обладают несколькими общими свойствами.

- Они имеют короткие пути – требуется всего несколько шагов, чтобы перейти от любого узла к любому другому узлу.
- Они имеют много треугольников, что отражается в высоких коэффициентах кластеризации.
- Они имеют гетерогенное распределение значений переменных узлов и связей, таких как степень и веса.

Следующий шаг в нашем исследовании состоит в понимании того, откуда берутся такие свойства. Как узлы выбирают своих соседей? Как генерируются хабы? Как образуются треугольники? В этой главе мы ответим на все эти вопросы.

Подход к изучению происхождения сетевых характеристик может состоять в формулировании *сетевой модели* (т. е. набора инструкций, используемых для сборки сети). Правила модели содержат в себе интуитивные идеи или гипотезы о том, как возникают сетевые признаки. Следуя рецепту модели, мы можем выстраивать сеть и сравнивать ее с реально существующими сетями, чтобы видеть, насколько они похожи или отличаются. Благодаря этому мы сможем узнавать о механизмах, которые порождают реально существующие сети.

Наше изложение проследит историческое развитие науки о сетях, представив классические сетевые модели в порядке их введения. Мы обсудим неспособность каждой модели воспроизводить признаки, наблюдаемые в реально существующих сетях, и подведем к разработке нового класса более реалистичных моделей. Мы представим простые механизмы, которые позволяют нам генерировать модельные графы, обладающие многими базовыми признаками реально существующих сетей.

5.1. Случайные сети

Предположим, у вас есть ряд разъединенных узлов и вы хотите ввести немного связей. Разместить связь между парами узлов можно

самыми разными способами. «Эгалитарный» подход заключается в их размещении между случайно отбираемыми парами узлов. Сеть, построенная в таком ключе, называется *случайной сетью*, или *сетью Эрдеша–Реньи* (вставка 5.1). Для простоты давайте сформулируем сетевую модель в эквивалентной версии, предложенной Гильбертом. Модель Гильберта имеет два параметра: число узлов N и *вероятность связи* p , описывающую то, насколько вероятно, что между любой случайно отобранной парой узлов образуется связь¹.

Вставка 5.1

Пол Эрдеш

Сетевая модель, генерируемая на основе случайного графа Эрдеша–Реньи, названа в честь двух математиков, Пола Эрдеша (Paul Erdős) и Альфреда Реньи (Alfréd Rényi), которые заложили основы теории случайных графов благодаря нескольким новаторским работам, опубликованным совместно в период с 1959 по 1968 годы.

Пол Эрдеш, показанный на рис. 5.1, был интересной личностью. У него не было дома, но он не был бездомным. Он навещал коллег и оставался у них дома до тех пор, пока они вместе работали над какой-нибудь математической задачей. Коллеги были рады принимать его у себя, так как эти визиты были очень продуктивными в профессиональном плане и часто приводили к появлению престижных научных публикаций. После того как теорема была доказана или статья написана, Эрдеш переходил к новой задаче, новому сотруднику и новому дому.

В дополнение к теории графов Эрдеш работал над задачами самых разных видов и сотрудничал более чем с 500 коллегами. Это делает его хабом в математической коллаборационной сети, социальной сети, обсуждаемой в главе 2.

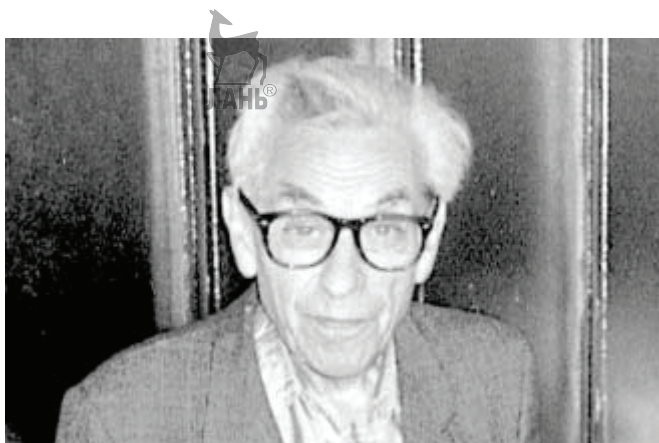


Рис. 5.1 Математик Пол Эрдеш в 1992 году. Изображение адаптировано с commons.wikimedia.org/wiki/File:Erdos_budapest_fall_1992.jpg силами Kmhkmh, используется по лицензии CC BY 3.0

¹ Вероятность связи не следует путать с вероятностью переподсоединения, которая будет представлена в разделе 5.2, хотя мы используем букву p для обоих.

Параметрами сетевой модели, генерируемой на основе случайной сети, сформулированной Гильбертом, являются число узлов N и вероятность связи p . Сеть может быть построена посредством следующей ниже процедуры.

1. Выбрать пару узлов, к примеру, i и j .
2. Сгенерировать случайное число r в интервале от 0 до 1. Если $r < p$, то добавить связь между i и j .
3. Повторить шаги (1) и (2) для всех пар узлов.

Главное различие между этими двумя формулировками заключается в том, что в версии Эрдеша и Реньи число связей сети фиксировано, тогда как в модели Гильберта оно является переменным. Если мы сгенерируем несколько сетей, следуя процедуре, описанной в приведенной выше вставке, все с использованием одинаковых значений для числа узлов и вероятности связи, то в общем случае они будут иметь разное число связей, колеблющееся вокруг среднего значения. Однако когда число узлов достаточно велико, то колебания числа связей малы.

В целях понимания того, как выглядят случайные сети при разных значениях вероятности связи, вообразите очень крупное множество узлов без связей. Естественно, в таком случае система будет полностью фрагментированной на узлы-одиночки (т. е. изолированные узлы, *синглтоны*). Теперь давайте добавим связи в случайном порядке, по одной за раз. Что должно произойти? Очевидно, что будет соединяться все большее число пар узлов, и через них будут формироваться связные подсети. В какой-то момент сеть станет связной, вследствие чего можно будет пройти из любого узла в любой другой узел, двигаясь по связям. Следовательно, должен существовать транзит от конфигураций, в которых все подсети относительно малы, к конфигурации, в которой по меньшей мере одна из подсетей содержит почти все узлы. Естественно ожидать, что подсети будут расти плавно и что указанный транзит будет происходить постепенно. Вместо этого Эрдеш и Реньи обнаружили, что этот транзит является резким и происходит, когда достигается определенная плотность связей. *Гигантская компонента* формируется при $\langle k \rangle = 1$, т. е. когда у каждого узла в среднем есть один сосед.

На рис. 5.2 мы показываем несколько конфигураций графа Эрдеша–Реньи для разных значений средней степени. Самая крупная связная компонента очень мала до транзитной точки и быстро растет со средней степенью в последствии. Остальные узлы поделены между малыми связными подсетями. По мере того как средняя степень становится все больше, гигантская компонента «съедает» все оставшиеся подсети и в итоге включает в себя все узлы: сеть становится связной. В приложении В.2 книги представлена демонстрация появления гигантской компоненты.

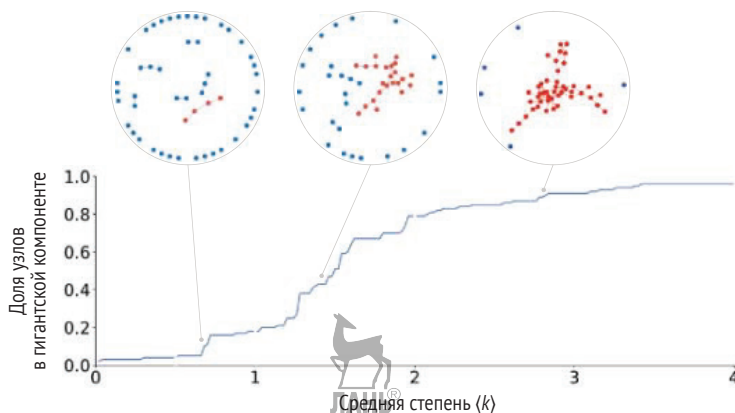


Рис. 5.2 Эволюция случайных сетей для возрастающих значений средней степени $\langle k \rangle$, что соответствует процессу добавления связей в систему по одной за раз. Самая крупная связная компонента, выделенная красным цветом, очень мала, когда средняя степень меньше единицы. Вокруг $\langle k \rangle = 1$ гигантская компонента очень быстро растет за счет других, более мелких компонент

5.1.1. Плотность

Строительство случайной сети для заданного значения вероятности связи похоже на процесс многократного подбрасывания смещенной монеты и подсчета числа раз, когда мы получаем орел или решку. Ожидаемое число орлов пропорционально вероятности того, что монета выпадет орлом, а также пропорционально числу подбрасываний монеты. Схожим образом ожидаемое число связей в случайной сети пропорционально вероятности связи и числу пар узлов.

Предположим, что смещенная монета поворачивается орлом с вероятностью p . Например, если $p = 0.1$, то мы можем ожидать, что в среднем из 10 подбрасываний мы будем получать одного орла и девять решек. Если $p = 0.5$, то мы восстановим знакомую ситуацию, в которой монета справедлива, и мы ожидаем одинакового числа орлов и решек. Если $p = 0$, то монета никогда не поворачивается орлом; если вместо этого $p = 1$, то она никогда не поворачивается решкой. Ожидаемое число орлов из t подбрасываний тогда равно pt (т. е. доле p подбрасываний). В нашей модели, генерируемой на основе случайной сети, число подбрасываний соответствует числу возможных пар из N узлов, т. е.

$\binom{N}{2} = \frac{N(N-1)}{2}$. Следовательно, ожидаемое число связей в случайном графе равно:

$$\langle L \rangle = p \binom{N}{2} = \frac{pN(N-1)}{2}. \quad (5.1)$$

Вспоминая уравнение (1.6), которое выражает среднюю степень сети как удвоенное число связей, деленное на число узлов, мы получаем ожидаемую среднюю степень $\langle k \rangle$ случайной сети:

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1). \quad (5.2)$$

Уравнение (5.2) говорит о том, что ожидаемая средняя степень узла в сети Эрдеша–Реньи является долей p его $N-1$ возможных соседей. Кроме того, подставив ожидаемое число связей в определение в уравнении (1.3) либо подставив ожидаемую среднюю степень в уравнение (1.7), мы находим, что ожидаемая плотность случайной сети равна $\langle d \rangle = p$.

Интуитивно вероятность связи выражает плотность случайной сети: она представляет собой ожидаемое соотношение между ожидаемым и максимальным числом связей. Мы знаем, что реально существующие сети обычно являются разреженными (т. е. они имеют очень малую среднюю степень, по сравнению с суммарным числом узлов, и очень малую плотность). Мы приходим к выводу, что для того, чтобы случайный граф был хорошей моделью реально существующих сетей, вероятность связи должна быть близка к нулю.

5.1.2. Степенное распределение

Предположим, что вы построили случайную сеть. Каково ее степенное распределение? Мы бы в первую очередь хотели бы узнать вероятность того, что узел имеет k соседей. Поскольку в этой модели ни один из узлов не играет какой-либо особой роли, мы можем просто рассмотреть любой узел, к примеру, i , и осведомиться, какова вероятность того, что i не имеет соседей, имеет одного соседа, двух соседей и т. д. Каждый из оставшихся $N-1$ узлов сети может быть соседом узла i . По замыслу решение о размещении или неразмещении связи между i и каждым другим узлом не зависит от наличия (или отсутствия) других связей где-либо. Каждая пара, включающая i , имеет вероятность p быть соединенной независимо от остальной сети.

Мы возвращаемся к задаче о подбрасывании монеты, располагая смещенной монетой и суммарным числом $N-1$ подбрасываний. Наш вопрос превращается в выяснение того, какова вероятность,

что мы получим k орлов в результате $N - 1$ подбрасываний, если вероятность получения орлов в каждом подбрасывании равна p . Эта задача задается биномиальным распределением:

$$p(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}. \quad (5.3)$$

В пределе крупного N и постоянной (не слишком малой) $pN \approx \langle k \rangle$, как и во многих разреженных реально существующих сетях, биномиальное распределение хорошо аппроксимируется колоколообразным распределением со средним значением $\langle k \rangle$ и дисперсией $\langle k \rangle$: средняя степень является хорошим статистическим описателем распределения.



Результирующее распределение вероятностей для степени в случайной сети представляет собой колоколообразную кривую с заметным пиком, сосредоточенным вокруг средней степени $\langle k \rangle$, и быстро затухающим по обе стороны от пика (рис. 5.3(а)). Степень большинства узлов близка к средней степени, и крупные отклонения от нее очень маловероятны.

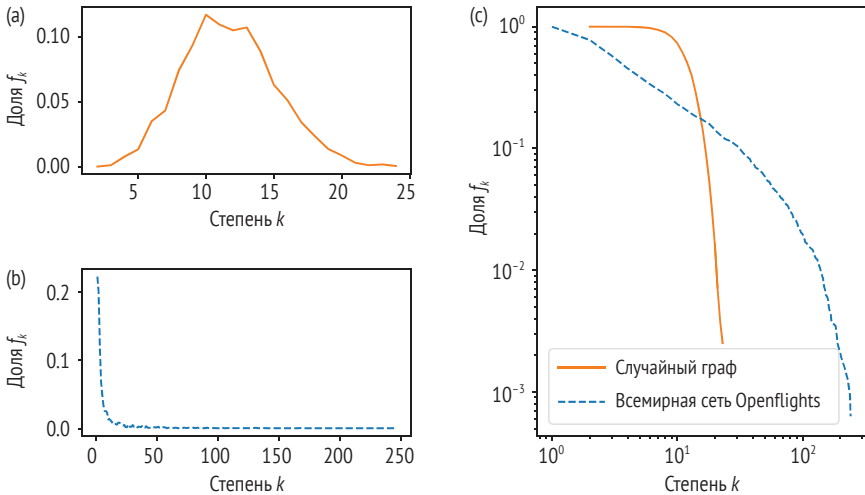


Рис. 5.3 Вероятностное распределение значений степени в случайной сети. (а) Степенное распределение случайного графа Эрдеша-Реньи с одинаковым числом узлов и связей, что и у всемирной сети авиарейсов в нашей коллекции данных: $N = 3179$, $L = 18617$. (б) Степенное распределение для всемирной сети авиарейсов. (в) Сравнение двух распределений в (а) и (б) в двойной логарифмической шкале



В главе 3 мы увидели, что степенное распределение многих реально существующих сетей довольно сильно отличается от такого распре-

деления (раздел 3.2) из-за наличия хабов (т. е. узлов с гораздо более крупной степенью, чем в среднем). На рис. 5.3(b) мы построили график степенного распределения нашей всемирной сети авиаперелетов. Утяжеленный хвост распределения охватывает более двух порядков величины в степени; в то время как у многих узлов есть всего горстка соседей, у некоторых узлов их сотни. На рис. 5.3(c) мы строим график распределения в двойной логарифмической шкале и сравниваем его с распределением на рис. 5.3(a), которое соответствует случайной сети с тем же числом узлов и связей. Очевидно, что модель, генерируемая на основе случайной сети, не обеспечивает хорошего описания распределения: узлы имеют приблизительно одинаковую степень, поэтому хабов нет. Такое расхождение является одной из причин, по которой нам нужны более изощренные сетевые модели.



5.1.3. Короткие пути

Давайте проверим, есть ли у случайных сетей короткие пути. Для разведывания этого вопроса можно использовать простую аргументацию. В предыдущем разделе мы увидели, что узлы имеют приблизительно одинаковую степень. Давайте предположим, что все они имеют степень 10. Если мы начнем с любого узла, то к нему будет прикреплено 10 узлов. У каждого из них также будет по 10 соседей и т. д. Таким образом, число достигаемых узлов растет экспоненциально вместе с числом шагов: за два шага мы сможем достичь 100 узлов, за три шага – 1000, а за несколько шагов мы сможем достичь каждого узла в сети.



Допустим, что сеть является связной и все узлы имеют степень k . В пределах $\ell = 1$ шага мы достигаем k узлов. У каждого из них есть $k - 1$ новых соседей, если исключить корневой узел, с которого начали. Поэтому в пределах $\ell = 2$ шагов мы сможем достичь целых $k(k - 1)$ узлов. Каждый новый сосед в свою очередь имеет вплоть до $k - 1$ новых соседей, поэтому на расстоянии $\ell = 3$ от корня мы находим $k(k - 1)^2$ узлов и т. д. Мы делаем вывод, что на расстоянии ℓ от корня мы находим вплоть до $k(k - 1)^{\ell - 1}$ узлов. Если k не слишком мало, то мы можем аппроксимировать $k - 1 \approx k$, и суммарное число узлов, достигаемых в пределах не более чем ℓ шагов от любого узла, приблизительно равно k^ℓ . (На самом деле эта оценка завышена, потому что в действительности соседи разных узлов иногда будут совпадать, тогда как мы допускаем, что этого никогда не произойдет.) Как далеко от узла мы должны пройти, чтобы добраться до всех других узлов? Диаметр ℓ_{\max} такой, что число узлов, достижимых в пределах не более чем ℓ_{\max} шагов из любого узла, совпадает с суммарным числом узлов N , задается уравнением

$$k^{\ell_{\max}} = N, \quad (5.4)$$

из которого мы получаем:

$$\ell_{\max} = \log_k N = \frac{\log N}{\log k}. \quad (5.5)$$

Оказывается, что приведенное выше уравнение дает хорошее приближение диаметра сети, даже если мы учитываем окрестное наложение и колебания степени вокруг $\langle k \rangle$. Медленный логарифмический рост ℓ_{\max} вместе с N указывает на то, что расстояния внутри сети невелики, даже если размер сети очень велик.

Тот факт, что максимальное расстояние от любого узла до любого другого узла (диаметр) в случайной сети невелико по сравнению с размером сети, означает, что сети Эрдеша–Реньи действительно имеют короткие пути. Для того чтобы дать представление о том, насколько быстро число достижимых узлов растет вместе с расстоянием от любого узла, давайте рассмотрим всемирную сеть социальных контактов и вообразим, что это случайная сеть. Если мы возьмем $k = 150$, т. е. среднее число регулярных контактов, которые люди могут поддерживать (*число Данбара*), то на расстоянии пять число достижимых людей составит $150^5 \approx 75$ млрд, что в 10 раз больше, чем население земного шара. Таким образом, в принципе, мы могли бы достичь любого индивидуума за пять шагов или меньше, что согласуется с результатом маломирового эксперимента Милграма (глава 2).

5.1.4. Коэффициент кластеризации

Как вы помните из главы 2, коэффициент кластеризации узла измеряет долю пар соседей узла, которые соединены друг с другом. Наличие связи между двумя соседями замыкает треугольник с фокальным узлом, поэтому коэффициент кластеризации также можно интерпретировать как долю треугольников, центрированных в фокальном узле, либо как вероятность замыкания треугольника.

В случайной сети вероятность того, что пара соседей узла соединена, равна p , так как вероятность соединения одинакова для каждой пары узлов независимо от наличия или отсутствия у них общих соседей. Естественно, коэффициенты кластеризации отдельных узлов могут немного отклоняться от p , но среднее значение по всем узлам хорошо аппроксимируется вероятностью p . В разделе 5.1.1 мы заметили, что p является очень малым числом, если мы намереваемся описывать реально существующие разреженные сети посредством модели Эрдеша–Реньи. Из этого следует, что средний коэффициент кластеризации этих сетей очень мал – модель создает треугольники с крайне малой вероятностью. С другой стороны, мы знаем, что ре-

ально существующие социальные сети имеют высокую кластеризацию (раздел 2.8). Поэтому случайные сети либо нереально плотны, либо имеют нереально мало треугольников. Мы приходим к выводу, что, если мы хотим учитывать удивительно высокую долю треугольников, наблюдаемых во многих реально существующих сетях, то нам нужна модель с неким конкретным правилом создания треугольников. Такие модели будут представлены в разделах 5.2 и 5.5.3.

Библиотека NetworkX имеет функции для генерирования случайных графов в соответствии с моделями Эрдеша–Реньи и Гильберта:

```
G = nx.gnm_random_graph(N, L) # случайный граф Эрдеша–Реньи
G = nx.gnp_random_graph(N, p) # случайный граф Гильберта
```

5.2. Малые миры

Как мы увидели в предыдущем разделе, реально существующие сети отличаются от случайных. Сети Эрдеша–Реньи действительно имеют короткие пути, но треугольники там встречаются редко, что приводит к средним значениям коэффициента кластеризации, которые бывают на порядки меньше, чем значения, измеренные в реально существующих сетях.

В конце 1990-х годов Дункан Дж. Уоттс (Duncan J. Watts) и Стивен Г. Стротац (Steven H. Strogatz) представили *маломировую модель*, также именуемую *моделью Уоттса–Стротаца*, которая генерирует сети с обоими признаками – короткими путями и высокой кластеризацией. Их идея состоит в том, чтобы начинать с решетчатой сети, в которой все узлы имеют одинаковое число соседей, например гексагональной решетки на рис. 5.4(а). Такая сеть имеет высокий средний коэффициент кластеризации, так как любая пара поочередных соседей каждого узла соединена, образуя треугольник с узлом.

Внутренние узлы имеют степень $k = 6$ и коэффициент кластеризации $C = 6 / \binom{6}{2} = 6/15 = 2/5$. Пограничные узлы имеют меньшую степень $k = 4, 3, 2$ и даже более высокие коэффициенты кластеризации, соответственно, $C = 3 / \binom{4}{2} = 1/2$, $C = 2 / \binom{3}{2} = 2/3$ и $C = 1 / \binom{2}{1} = 1$. Следовательно, средний коэффициент кластеризации составляет не менее $2/5$ и сходится к $2/5$ в пределе бесконечной решетки ($N \rightarrow \infty$).

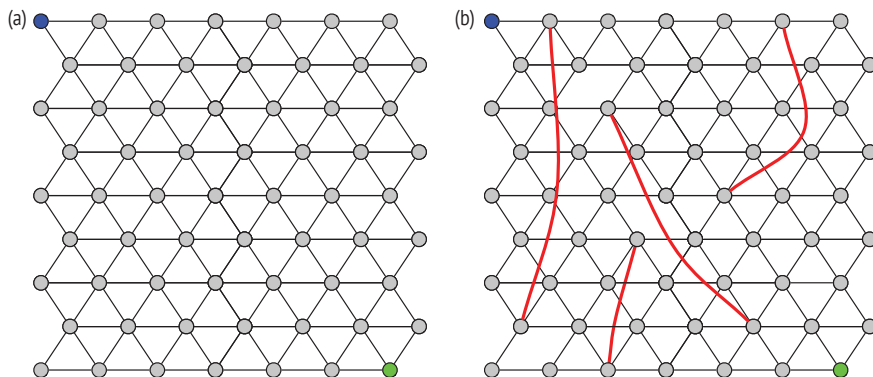


Рис. 5.4 Маломировые сети. (а) Гексагональная решетка, граф, в котором каждый узел имеет шесть соседей (за исключением узлов на границе). В ней существует много треугольников, поэтому узлы имеют высокий коэффициент кластеризации. Путям из одного угла в другой приходится пересекать много связей, поэтому средняя длина кратчайшего пути велика. (б) Четыре связи были переподсоединены к случайно отобранным узлам, которые, как правило, находятся дальше от изначальных конечных точек. Эти связи (красного цвета) являются сокращениями и позволяют нам достигать отдаленных частей сети посредством малого числа прыжков. Например, кратчайший путь из синего узла в зеленый узел проходит с 10 шагов по решетке до шести шагов по одному из сокращений. Поскольку в результате процедуры переподсоединения нарушается лишь несколько треугольников, коэффициент кластеризации остается высоким

С другой стороны, сеть имеет высокую среднюю длину кратчайшего пути. Например, узлы на противоположных сторонах решетки могут быть достигнуты только по путям, пересекающим большое число связей. Однако расстояния между узлами можно значительно сократить, создав несколько *сокращений* – связей, соединяющих части сети, изначально отдаленные друг от друга, как красные связи на рис. 5.4(б). Это можно сделать путем отбора нескольких первоначальных связей в случайном порядке, сохраняя одну из их конечных точек и заменяя другую конечную точку узлом, отбираемым случайно среди всех других узлов. Формально указанная процедура переподсоединения применяется к каждой связи сети с *вероятностью переподсоединения* p^1 . Число переподсоединенных связей пропорционально вероятности переподсоединения.



¹ Обратите внимание, что вероятность переподсоединения в маломировой модели не совпадает с вероятностью связи в модели, генерируемой случайной сетью, несмотря на то что мы используем букву p для обоих. Хотя это немного дезориентирует, мы придерживаемся принятых в сообществе науки о сетях традиций; пожалуйста, не забывайте интерпретировать параметр p , основываясь на контексте обсуждаемой модели.



Ожидаемое число переподсоединенных связей равно pL , где p – это вероятность переподсоединения, а L – суммарное число связей в сети. Частный случай $p = 0$ соответствует первоначальной решетке, а частный случай $p = 1$ порождает случайную сеть.

Если частота переподсоединения очень мала (близка к нулю), мало что происходит. Если она очень велика (близка к единице), то сеть становится случайной сетью а-ля Эрдеш и Реньи, так как, в сущности, все связи переподсоединяются к случайным узлам, что эквивалентно размещению связей между случайно выбранными парами узлов. В этом сценарии большинство треугольников разрушается, и коэффициент кластеризации становится очень малым. Но если p выбрано не слишком малым и не слишком большим, то можно достичь компромисса, когда существует достаточно сокращений, чтобы в среднем значительно уменьшить расстояния, но не так много, чтобы нарушить большинство треугольников. В этом режиме пути являются такими же короткими, как и в случайных сетях, тогда как средний коэффициент кластеризации лишь незначительно уменьшается по отношению к первоначальной конфигурации в форме решетки, вследствие чего он бывает сопоставим с таковым в реально существующих социальных сетях.

На рис. 5.4 мы начинаем с гексагональной решетки, но в качестве первоначальной конфигурации можно использовать любую сеть с высоким коэффициентом кластеризации. В своей новаторской статье Уоттс и Строгац (1998) представили воображаемое кольцо, в котором каждый узел соединен со своими k ближайшими соседями. Они рассмотрели случай $k = 4$, как показано на рис. 5.5(a). В этом случае первоначальный коэффициент кластеризации равен $C = 1/2$, потому что соседи каждого узла образуют три треугольника из шести возможных. Это очень высокий показатель. Можно переподсоединить связь таким образом, чтобы один из прикрепленных узлов сохранял связь, в то время как другой конец связи был прикреплен к случайно выбираемому узлу; это формулировка используется в изначальной модели. В качестве альтернативы можно заменить связь, соединяя два случайных узла независимо от их степени. Еще один вариант заключается в том, что вместо того, чтобы переподсоединять существующие связи, в сеть просто могут добавляться случайные связи.

На рис. 5.5(b) показаны средняя длина кратчайшего пути $\langle \ell \rangle_p$ и коэффициент кластеризации C_p как функция от вероятности переподсоединения p . Существует интервал значений вероятности переподсоединения, лежащий между $p \approx 0.01$ и $p \approx 0.1$ (выделено на рис. 5.5(b)), в котором $\langle \ell \rangle_p \approx \langle \ell \rangle_1$ и $C_p \approx C_0$. Другими словами, получаемая из модели средняя длина кратчайшего пути близка к длине аналогичного пути у эквивалентной случайной сети и намного ниже, чем у решетки. В то же время получаемый из модели коэффициент кластеризации по-прежнему близок к коэффициенту кластеризации у решетки

и намного больше, чем у случайной сети. Следовательно, модель Уоттса–Строгаца действительно способна генерировать – с подходящим объемом случайности – сети, наделенные двумя желаемыми признаками: короткими путями и высокой кластеризацией. Демонстрация указанной способности представлена в приложении В.3 книги.

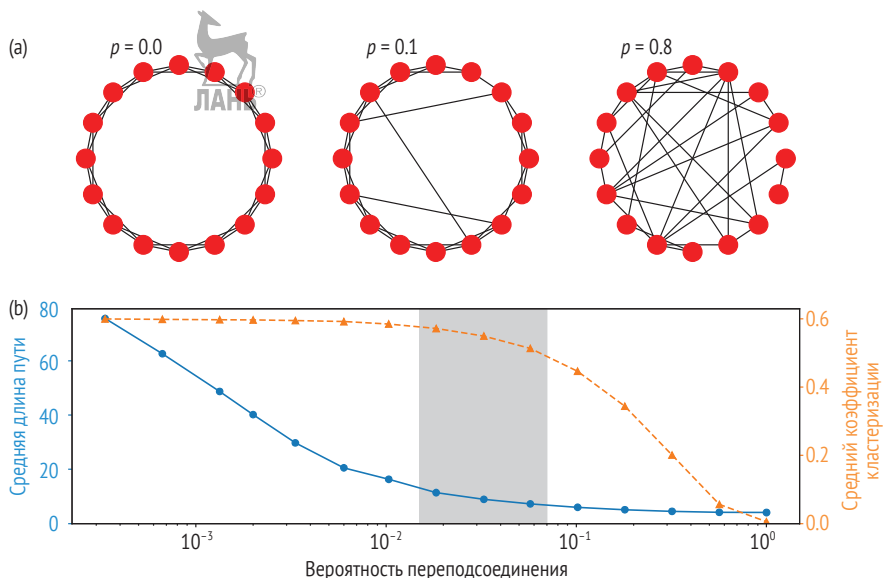


Рис. 5.5

Маломировая модель. (а) В стандартных конфигурациях сетей на основе модели Уоттса–Строгаца мы начинаем с кольцевой решетки (слева), в которой каждый узел соединен со своими четырьмя ближайшими соседями, и постепенно добавляем сокращения, переподсоединяя связи.

(б) Уменьшение средней длины кратчайшего пути и коэффициента кластеризации как функция вероятности переподсоединения p . Крайняя точка $p = 0$ – это решетчатая сеть, как на крайнем левом графе выше, но с $N = 1000$ узлами. Крайняя точка $p = 1$ – это случайная сеть с одинаковым числом узлов и связей. Затененная область выделяет такие значения p , при которых средняя длина пути почти такая же короткая, как у случайной сети, тогда как коэффициент кластеризации по-прежнему почти такой же большой, как у решетки

Однако данная сетевая модель не способна генерировать хабы. Степенное распределение переходит от изначальной решетки, где все узлы имеют одинаковую степень, к распределению, характерному для случайной сети, с одинаковым числом узлов и связей, где степени сконцентрированы вокруг характерного значения (рис. 5.3(а)). Отсюда, при любом значении вероятности переподсоединения p все узлы имеют одинаковую степень; ни один из них не накапливает непропорциональную долю связей. Нам нужен какой-то другой модельный ингредиент, чтобы объяснить появление хабов.

Библиотека NetworkX имеет функцию, которая генерирует графы в соответствии с маломировой моделью Уоттса и Строгаца:

```
G = nx.watts_strogatz_graph(N,k,p) # сеть на основе маломировой модели
```

5.3. Конфигурационная модель

Давайте сосредоточимся на сетях с реалистичным степенным распределением. В разделе 5.4 мы разведем механизмы, ответственные за существование хабов. Но сначала давайте ответим на следующий вопрос: учитывая некоторое степенное распределение, сможем ли мы построить сеть, узлы которой имеют точно такое степенное распределение?

Простой ответ обеспечивается *конфигурационной моделью*. Эта модель на самом деле преследует более амбициозную цель: генерирование сети, узлы которой имеют произвольную степенную последовательность, в которой узел 1 имеет степень k_1 , узел 2 – степень k_2 и т. д. (вставка 5.2). Степенная последовательность может продуцироваться из конкретного распределения, которое мы хотим воспроизвести, либо она может браться из узлов реально существующей сети. После того как мы воспроизведем последовательность степеней всех узлов, мы также должны воспроизвести соответствующее степенное распределение. С другой стороны, многочисленные степенные последовательности соответствуют одинаковому распределению. Например, две сети с четко отличимыми степенными последовательностями (1, 2, 1) и (1, 1, 2) имеют одинаковое степенное распределение.

Вставка 5.2

Степенные последовательности

Степенная последовательность сети – это список степеней их узлов в порядке их меток. Степенная последовательность представляет собой список из N чисел $(k_0, k_1, k_2, \dots, k_{N-1})$, где k_i – это степень узла i . Обратите внимание, что степенная последовательность определяет степенное распределение, но обратное неверно. Каждая перестановка степенной последовательности приводит к одинаковому распределению, так как для распределения не имеет значения, какой узел имеет какую степень, важно только число узлов, которые имеют данную степень.

Предположим, у нас есть множество узлов и их степенная последовательность. Первым шагом является назначение каждому узлу числа *заглушек*, соответствующих степени узла, как показано на рис. 5.6(а). Заглушка (stub) – это просто оборванная связь, имеющая узел в качестве одной из своих конечных точек, но еще не соединенная с соседом. Затем сеть строится с помощью следующих ниже итерационных шагов.

1. Случайно отбирается пара заглушек.
2. Выбранные заглушки соединяются друг с другом, образуя связь между прикрепленными к заглушкам узлами.

Эта процедура повторяется до тех пор, пока все заглушки не будут соединены попарно. Естественно, для того чтобы это произошло,

должно быть четное число заглушек (т. е. сумма степеней в целевой последовательности должна быть четной). Мы видим, почему эта процедура достигает нашей цели: если к узлу прикреплено k заглушек, то у него в конечном итоге будет k соседей. Поскольку число заглушек, прикрепленных к каждому узлу, равно его степени, каждый узел в итоге получает желаемую степень. Как показано на рис. 5.6(b–d), в таком ключе можно создать несколько сетей в зависимости от последовательности пар комбинируемых заглушек. Некоторые исходы бывают нежелательными, если они нарушают ограничения. Например, возможно, кому-то захочется исключить сети с многочисленными связями между двумя узлами (рис. 5.6(c)) либо с самонаправленными циклами (рис. 5.6(d)).

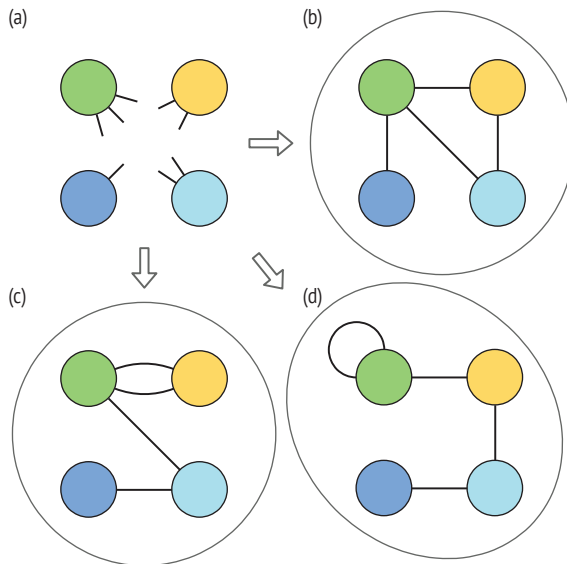


Рис. 5.6 Конфигурационная модель. (a) Мы начинаем с узлов и заглушек, соответствующих данной степенной последовательности. (b–d) Мы можем соединять заглушки по-разному, приводя к разным сетям с данной степенной последовательностью

Способ формирования связей является случайным по конструкции. Таким образом, конфигурационная модель генерирует случайные сети с предписанной степенной последовательностью. Это оказывается очень полезным при анализе сетей. В главе 3 мы увидели, что широкие степенные распределения ответственны за ряд особенных свойств и эффектов. Однако существуют сетевые признаки, которые не зависят только от степенного распределения.

Имея сеть, мы могли бы задаться вопросом, объясняется ли конкретное свойство только степенным распределением. Посредством конфигурационной модели мы можем генерировать рандомизированные, или *перетасованные*, версии сети, имеющие одинаковую

степенную последовательность. Каждая из этих конфигураций представляет собой *рандомизацию* изначальной сети *с сохранением степени*, поскольку она сохраняет степенную последовательность, но все остальное в противном случае является совершенно случайным. Теперь мы можем проверить, присутствует ли интересующий признак в перетасованных сетевых конфигурациях. Если он есть, то этот признак должен быть результатом только степенного распределения; в противном случае в его основе должны стоять другие факторы.

Например, предположим, что признаком, который мы хотели бы расследовать, является средний коэффициент кластеризации: можно ли объяснить кластеризованную структуру реально существующей социальной сети ее степенным распределением? Нам нужно лишь вычислить коэффициент кластеризации достаточного числа случайных конфигураций, вывести среднюю и стандартную ошибку и проверить, совместимо ли значение меры в изначальном графе с оценкой из перетасованных сетей в пределах ошибки. Если это так, то мы делаем вывод о том, что присутствующие в сети треугольники существуют просто из-за степенных ограничений. Если она намного больше, чем случайная оценка, как это обычно бывает, то шаблоны связывания в сети не могут быть случайными; они должны быть результатом какого-то механизма, способствующего образованию треугольников.

Библиотека NetworkX имеет функцию, которая генерирует сеть с предписанной степенной последовательностью посредством конфигурационной модели:

```
G = nx.configuration_model(D) # сеть со степенной последовательностью D
```

Конфигурационная модель генерирует все возможные сети, имеющие данную степенную последовательность, но мы могли бы также наложить другие ограничения. Например, нам было бы интересно разведать все сети, имеющие данное число треугольников. Идея генерирования сетей с конкретными характеристиками привела к разработке широкого класса сетевых моделей, именуемых *экспоненциальными случайными графами* (вставка 5.3).

Вставка 5.3

Экспоненциальные случайные графы

Представляет интерес изучения случайно генерируемых сетей, имеющих некоторые общие количественные признаки, но отличающихся своей детальной структурой. С одной стороны, они представляют потенциальные альтернативы конкретным сетевым конфигурациям, с которыми мы сталкиваемся в реальном мире. С другой стороны, они позволяют нам расследовать взаимную игру между разными структурными свойствами. Например, мы могли бы спросить, какие значения среднего коэффициента кластеризации совместимы с конкретным значением плотности.

Экспоненциальные случайные графы – это классы случайных сетей, которые подвержены ограничениям. Мы определяем класс сетей, основываясь на множестве из M сетевых мер, x_m , $m = 1, \dots, M$. Мы накладываем ограничение для каждой меры x_m : среднее значение по всем сетям класса должно быть, к примеру, $\langle x_m \rangle = x_m^*$. Экспоненциальные случайные графы являются сетями, которые удовлетворяют этим ограничениям при максимизировании случайности. Как оказалось, это позволяет нам определять вероятность $P(G)$ селекции сети G этого класса, имеющей значения мер $x_1(G), x_2(G), \dots, x_M(G)$:

$$P(G) = \frac{e^{H(G)}}{Z} \quad (5.6)$$

с помощью

$$H(G) = \sum_{m=1}^M \beta_m x_m(G), \quad (5.7)$$

где β_m – это параметр, связанный с мерой x_m . Функция Z обеспечивает, чтобы эта вероятность $P(G)$ была вероятностью, такой что $\sum_G P(G) = 1$.

Посредством вероятностей в уравнении (5.6) можно рассчитать среднее значение любой сетевой меры. В частности, мы можем выразить каждое ограничение, устанавливая среднее значение меры x_m равным его желаемому значению:

$$\langle x_m \rangle = \sum_G P(G) x_m(G) = x_m^*, \quad (5.8)$$

где суммирование выполняется по всем сетям класса. В результате мы получаем множество из M уравнений с M переменными, параметрами β_m . Решение этих уравнений дает значения параметров. Оно определяет модель, которую затем можно использовать для расчета среднего значения любой интересующей переменной. Хотя ограничения накладываются на средние значения, оказывается, что для большинства экспоненциальных случайных графов в классе значение любой меры близко к ее среднему значению.

Модель, генерируемая на основе случайной сети, в формулировке Гильберта (раздел 5.1) представляет собой специальную версию экспоненциального случайного графа с единственным ограничением, согласно которому сети должны иметь заданное среднее число связей.

5.4. Преференциальное прикрепление

Разведанные до этого модели являются *статичными*. Под статичностью здесь мы подразумеваем, что все узлы сети существуют с самого начала; мы лишь добавляем (или переподсоединяем) связи между ними. Вместо этого реально существующие сети обычно *динамичны*. Узлы и связи появляются и исчезают. Если мы рассмотрим популярные сети, такие как интернет, Всемирная паутина, Facebook и Twitter,

то заметим, что их размер растет. Узлы также могут исчезать (например, старые маршрутизаторы интернета или старые страницы Всемирной паутины). Но введение новых узлов является более вероятным. Именно по этой причине реалистичные динамические модели обычно содержат в себе ту или иную форму *сетевого роста*. Динамическая процедура начинается с первоначальной конфигурации, обычно очень малой группы узлов. Затем узлы добавляются один за другим. Каждый новый узел прикрепляется к ряду старых узлов на основе некоторого правила, характерного для модели (рис. 5.7).

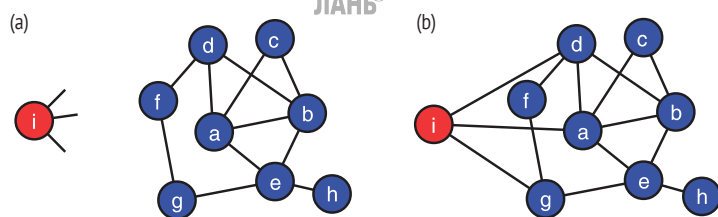


Рис. 5.7 Сетевой рост. Строительство сети обычно происходит динамично, с добавлением новых узлов и соединением со старыми узлами. (a) В систему добавляется новый узел *i* с тремя заглушками. (b) Каждая заглушка прикрепляется к более старому узлу в соответствии с некоторым правилом, и новый узел встраивается в сеть

Еще одним ограничением рассмотренных до этого сетевых моделей является то, что они не могут объяснять существование хабов. Если быть точнее, конфигурационная модель может генерировать хабы, но только путем *априорного* определения степени узлов – это не помогает объяснять, как хабы возникают в реальном мире. Модели, генерируемые на основе случайной сети и на основе малого мира, не порождают хабы. Главная причина заключается в том, что в обоих случаях правила связывания в основном являются эгалитарными – узлы выбирают своих соседей совершенно случайно. По этой причине крайне маловероятно, что какой-либо один узел будет иметь преимущество перед другими узлами и в итоге будет иметь гораздо больше соседей, чем остальные. Если мы хотим восстанавливать хабы, то, следовательно, необходимо ввести механизм, который отдает предпочтение одним узлам перед другими. Такой механизм называется *преференциальным прикреплением* (preferential attachment): чем выше степень узла, тем больше связей он получит.

Лежащая в его основе идея проста. Предположим, что вы создаете новую страницу Всемирной паутины и хотите включить несколько связей, указывающих на другие страницы. Наши знания неизбежно ограничены крошечной частью триллионов страниц во Всемирной паутине. Большинство страниц, о которых нам известно, скорее всего, будут популярными, и в силу этого они имеют входящие связи со стороны многих других страниц. Кстати, это бывает причиной, почему мы их обнаружили: если страница имеет входящие связи со стороны многочисленных документов, то ее с большей вероятностью можно

найти посредством серфинга по Всемирной паутине либо поисковой машины. Следовательно, наш выбор документов для установления связей из нашей новой страницы паутины будет в пользу популярных и сильно связанных страниц. Аналогичным образом, когда мы пишем научную статью и составляем ее библиографию, общепринято ссылаться на статьи, которые часто цитируются другими авторами, поскольку мы, скорее всего, на них натолкнемся, когда будем читать другие статьи и просматривать их библиографические ссылки.

На языке сетей популярный узел – это узел с высокой степенью, указывающей на то, что у него много соседей. Преференциальное прикрепление означает, что высокостепенные узлы имеют высокую вероятность получения новых связей. Такой критерий известен в различных контекстах и под несколькими названиями (вставка 5.4). Наиболее известная модель на основе сетевого роста, содержащая в себе преференциальное прикрепление, была предложена Барабаши и Альбертом в 1999 году и именуется моделью Барабаши–Альберта, или моделью ВА (от англ. *Barabási–Albert*), или *моделью на основе преференциального прикрепления*. Она представляет собой простое сочетание роста и преференциального прикрепления. На каждом шаге добавляется новый узел и присоединяется к нескольким существующим узлам. Вероятность того, что новый узел прикрепляется к старому узлу, пропорциональна степени старого узла. По этой причине узел со степенью 100 получит новую связь в 100 раз с большей вероятностью, чем узел со степенью один. На рис. 5.7, например, в рамках преференциального прикрепления шанс, что узел a получит связь со стороны узла i , будет вдвое выше, чем у узла c .

Мы начинаем с полного графа с m_0 узлами. Каждая итерация алгоритма состоит из двух шагов.

1. В сеть добавляется новый узел i с $m \leq m_0$ прикрепленными к нему новыми связями. Отсюда параметр m представляет собой среднюю степень сети.
2. Каждая новая связь подсоединяется к старому узлу j с вероятностью

$$P(i \leftrightarrow j) = \frac{k_j}{\sum_l k_l}. \quad (5.9)$$

Знаменатель в уравнении (5.9) представляет собой сумму степеней всех узлов (кроме i) и гарантирует, что сумма всех вероятностей равна единице, как и должно быть.

Указанная процедура повторяется до тех пор, пока сеть не достигнет желаемого числа узлов N . Во вставке 5.5 показано, как отбирать узел с желаемой вероятностью на языке Python.

Вставка 5.4

Преференциальное прикрепление

Принцип, лежащий в основе преференциального прикреплении, прост: чем больше вы имеете, тем больше вы получите! Он к тому же еще и древний. Первое известное упоминание можно найти в Евангелии от Матфея (25:29): «Ибо каждому, кто имеет, будет дано больше, и у него будет изобилие; но у того, кто не имеет, даже то, что у него есть, будет отнято». В этой цитате указанный принцип резюмирован в первом предложении, другое – это его симметричная противоположность, утверждающая, что чем меньше кто-то имеет, тем меньше он будет иметь в будущем. Так что *богатые становятся богаче, а бедные – беднее*. Поэтому преференциальное прикрепление также называется *эффектом от Матфея*. Еще одно распространенное его название – *кумулятивное преимущество*.

Первой научной имплементацией этого принципа стала *урновая модель* Пойи, которая работает следующим образом. Урна содержит X белых и Y черных шаров; один шар случайно извлекается из урны и помещается обратно вместе с другим шаром того же цвета. Если шаров X намного больше, чем шаров Y , то вероятнее, что мы извлекаем белый шар, чем черный. Если мы действительно выберем белый шар, то в конце раунда будет $X + 1$ белых шаров и Y черных, давая дополнительное преимущество белым в следующем раунде. Следовательно, число белых шаров будет увеличиваться быстрее, чем число черных шаров.

Преференциальное прикрепление использовалось для объяснения тяжелых степенных распределений значений многих разных величин, таких как число видов в расчете на род цветущих растений, число (отдельных) слов в тексте, численности населения городов, индивидуальное благосостояние, научное производство, статистика цитирования и размер фирмы среди прочего. Сетевые модели на основе преимущественного прикреплении были представлены Джорджем У. Юлом, Гербертом А. Саймоном, Робертом К. Мертоном, Дерекком де Солла Прайсом, Альбертом-Ласло Барабаши и Рекой Альбертом.

Вставка 5.5

Случайный отбор с распределением значений вероятности

Часто бывает необходимо отбирать узлы случайно с вероятностью, пропорциональной некоторой величине. Например, в случае случайной сети мы отбираем узел для прикреплении связей с равномерной вероятностью, а значит, каждый узел имеет одинаковую вероятность быть отобранным. На Python мы можем сделать это с помощью модуля `random`:

```
nodes = [1, 2, 3, 4]
selected_node = random.choice(nodes)
```

В других случаях нам нужно отбирать узлы с разными вероятностями. Например, в случае преференциального прикреплении (раздел 5.4) на каждом шаге нам нужно выбирать узел с вероятностью, пропорциональной его степени. Или в случае модели приспособленности (раздел 5.5.2) селективный отбор взвешивается с помощью какой-то более усложненной функции степени и приспособленности.

К счастью, работать с этими случаями также легко, как и в Python 3.7. Нам нужно лишь предоставить второй аргумент со списком ассоциированных с узлами весов. Предположим, мы хотим отобрать узел в соответствии с его степенью, как в преференциальном прикреплении. Мы можем использовать степени в качестве весов:

```
nodes = [1, 2, 3, 4]
degrees = [3, 1, 2, 2]
selected_node = random.choices(nodes, degrees)
```

Отбор узла 1 ($k = 3$) имеет вероятность в три раза выше, чем узла 2 ($k = 1$). Функция `random.choices()` позволяет выполнять случайный селективный отбор из популяции, основываясь на любом заданном множестве весов. Веса могут быть вероятностями из распределения, но в этом нет необходимости – они не должны обязательно в сумме составлять единицу. И они не обязательно должны быть целыми числами. Позаботьтесь о том, чтобы популяция и весовые последовательности были выровнены: i -й элемент в популяции должен соответствовать i -му элементу в последовательности весов.

По конструкции в начале все узлы имеют одинаковую степень. По мере того как в систему добавляются новые узлы и связи, степени узлов растут. Однако самые старые узлы существуют с самого начала, поэтому они могут получать связи в любое время, в отличие от узлов, которые вступают в игру намного позже. Поэтому степени старейших узлов превышают степени более новых узлов, и по причине преференциального прикреплении это делает первых еще более вероятными для привлечения новых связей в будущем за счет последних.

Такая динамика в стиле «*богатый становится богаче*» создает желаемую гетерогенность в степенном распределении, при этом самые старые узлы становятся сетевыми хабами. На рис. 5.8(а,с) мы показываем сеть, построенную с использованием модели Барабаши–Альберта (BA), вместе с ее степенным распределением. Мы наблюдаем тяжелохвостное распределение, что подтверждает существование хабов. В приложении В.4 книги представлена демонстрация указанной модели.

В этом месте вы, возможно, зададитесь вопросом, достаточно ли для появления хабов иметь рост без преференциального прикреплении. В конце концов, у первоначальных узлов будет больше времени для сбора связей независимо от критерия связывания. Например, предположим, что каждый новый узел может подбирать в качестве соседа любой случайно выбираемый узел независимо от его степени. Как и раньше, мы ожидаем, что чем старше узлы, тем больше их степени. Это верно, но, как хорошо видно на рис. 5.8(б,с), в этом случае значения степеней не сильно отличаются друг от друга, и соответствующее распределение не имеет утяжеленного хвоста. Мы приходим к выводу, что сочетание роста и случайного селективного отбора узлов с ра-

ботой не справляется; необходимо преференциальное прикрепление. И действительно, эмпирические исследования подтвердили, что преференциальное прикрепление влияет на рост многих реально существующих сетей.

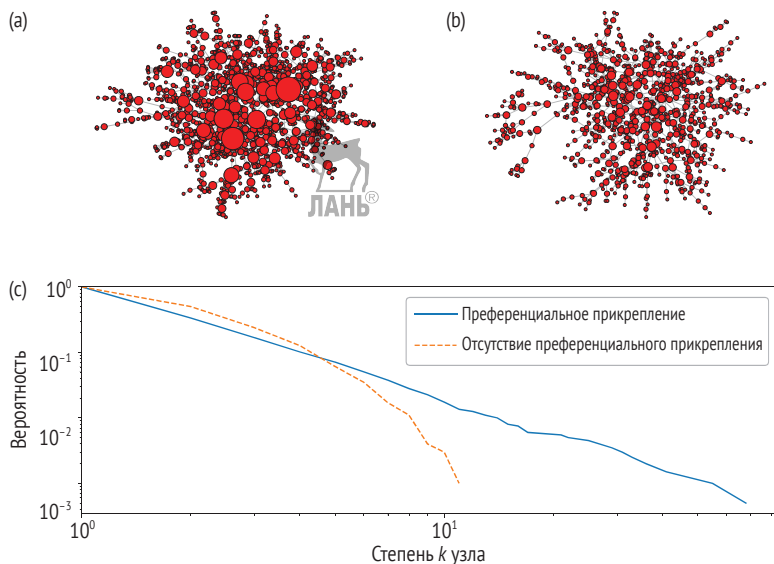


Рис. 5.8 Преференциальное прикрепление. (а) Сеть, сгенерированная с помощью модели Барабаши–Альберта. Она имеет $N = 2000$ узлов и среднюю степень $\langle k \rangle = 2$. Размер узла пропорционален его степени; крупные узлы являются хабами. (б) Сеть, сгенерированная с помощью аналогичной модели роста, но со случайным, а не преференциальным прикреплением. Здесь нет хабов. (с) Кумулятивное степенное распределение сетей в (а) и (б). Модель Барабаши–Альберта генерирует широкое распределение, тогда как отсутствие преференциального прикреплении приводит к более узкому распределению, без хабов

Библиотека NetworkX имеет функцию, которая генерирует графы в соответствии с моделью Барабаши–Альберта:

```
G = nx.barabasi_albert_graph(N,m) # Сеть на основе модели Барабаши-Альберта
```

5.5. Другие преференциальные модели

В модели Барабаши–Альберта используется линейное преференциальное прикрепление, поскольку вероятность связи строго пропорциональна степени целевого узла. Предположим, что мы ослабим это правило и позволим вероятности изменяться в некоторой степени. Мы называем этот подход *нелинейным преференциальным прикреплением*.

Расширение модели Барабаши–Альберта с использованием нелинейного предпочтительного прикрепления идентично изначальной модели Барабаши–Альберта, за исключением того, что уравнение (5.9), выражающее вероятность того, что старый узел j получит связь от нового узла i , задается уравнением

$$P(i \leftrightarrow j) = \frac{k_j^\alpha}{\sum_l k_l}, \quad (5.10)$$

где экспонента α – это параметр. При $\alpha = 1$ мы восстанавливаем стандартную модель Барабаши–Альберта. Что происходит, когда $\alpha \neq 1$? Есть два разных сценария.

1. Если $\alpha < 1$, то вероятность связи не растет со степенью, такой же быстрой, как в модели Барабаши–Альберта, поэтому преимущество высокостепенных узлов над другими не так велико. В результате степенное распределение не имеет утяжеленного хвоста – хабы исчезают!
2. Если $\alpha > 1$, то высокостепенные узлы накапливают новые связи намного быстрее, чем низкостепенные узлы. Как следствие, один из узлов в итоге будет соединен к доле всех остальных узлов. Эффект еще более экстремален, когда $\alpha > 2$, и в этом случае мы наблюдаем эффект «победитель получает все»: один узел может быть соединен со всеми другими узлами, которые имеют приближенно одинаковую низкую степень.

В зависимости от значения экспоненты, выражающей степень (как экспоненциацию) степени (как объема или размаха), мы либо оказываемся без хабов (сублинейное предпочтительное прикрепление), либо с одним единственным суперхабом (суперлинейное предпочтительное прикрепление). В любом случае нелинейное предпочтительное прикрепление отказывается в генерировании хабов, как это наблюдается в реально существующих сетях; линейное предпочтительное прикрепление остается единственным путем. Это проявляет фундаментальную хрупкость модели Барабаши–Альберта, поскольку необходимость строгой пропорциональности между вероятностью связи и степенью выглядит нереалистичной. К счастью, как мы увидим в разделе 5.5.4, существуют естественные механизмы связывания, которые индуцируют линейное предпочтительное прикрепление в неявной форме.

В дополнение к зависимости от линейного предпочтительного прикрепления модель Барабаши–Альберта имеет и другие ограничения.

- Она продуцирует фиксированный шаблон для степенного распределения. Наклон кривой предпочтительного прикрепления

на рис. 5.8(с) одинаков для любого выбора модельных параметров. Реальные степенные распределения могут затухать быстрее или медленнее.

- Хабы являются самыми старыми узлами; новые узлы не способны их преодолевать по степени.
- Она не создает много треугольников. Средний коэффициент кластеризации намного ниже, чем во многих реально существующих сетях.
- Узлы и связи только добавляются; в реально существующих сетях они также могут удаляться.
- Поскольку каждый узел прикрепляется к более старым узлам, сеть состоит из одной единственной связной компоненты. Многие реально существующие сети состоят из нескольких таких компонент.

Далее мы представим более изощренные модели сетевого роста, которые устраняют некоторые из этих ограничений.

5.5.1. Модель на основе привлекательности

Преференциальное прикрепление имеет едва уловимый подвох: что происходит, если у узла нет соседей? Его степень равна нулю, поэтому вероятность того, что он получит связи, тоже равна нулю: у узла никогда не будет соседей! И поэтому если бы мы стартовали с первоначального ядра узлов без соседей, то модель Барабаши–Альберта рухнула бы, так как новые узлы невозможно было бы прикрепить ни к одному из старых узлов. Стандартная первоначальная конфигурация модели Барабаши–Альберта состоит из полного графа, и значит, у каждого узла есть соседи, и эта проблема не возникает, но в идеале модель должна работать с разными вариантами первоначального условия. Если мы рассмотрим направленные сети и допустим, что вероятность связи зависит только от степени-на-входе, то проблема возникает независимо от первоначальной конфигурации. Каждый новый узел изначально имеет нулевую степень-на-входе, так как он приходит с исходящими связями и может принимать входящие связи только из более новых узлов. Поэтому никакие новые узлы не могут получать входящие связи.

К счастью, из этой ситуации есть простой выход. Вместо того чтобы иметь вероятность связи, строго пропорциональную степени, мы можем немного модифицировать правило. Идея, первоначально предложенная Дерекком де Солла Прайсом (Derek de Solla Price) в контексте сетей цитирования, заключается в том, что узел получает связи не только по причине своей степени, но и потому, что он обладает внутренней привлекательностью. В модели на основе привлекательности вероятность связи *пропорциональна сумме степени и постоянной привлекательности*.

Модель на основе привлекательности представляет собой слегка модифицированную версию изначальной модели Барабаши–Альберта, в которой уравнение (5.9), выражающее вероятность того, что старый узел j получит связь от нового узла i , заменяется на формулу

$$P(i \leftrightarrow j) = \frac{A + k_j}{\sum_l (A + k_l)}, \quad (5.11)$$

где A – это параметр привлекательности и может принимать любое положительное значение. Случай $A = 0$ дает модель Барабаши–Альберта.

Для любого значения параметра привлекательности A указанная модель строит сети с тяжелохвостными степенными распределениями. Наклон распределения зависит от A . Благодаря этому модель способна соответствовать степенным распределениям нескольких реально существующих сетей в отличие от модели Барабаши–Альберта.

5.5.2. Модель на основе приспособленности

Как мы видели в разделе 5.4, в модели Барабаши–Альберта хабы также являются самыми старыми узлами. Этот признак нереалистичен. Во Всемирной паутине, например, могут существовать страницы, которые создаются намного позже других, но в итоге становятся популярнее и привлекают больше гиперсвязей. Возьмем, к примеру, поисковик Google. Он был создан в 1998 году, когда уже существовали миллионы сайтов, но в итоге он стал самым популярным хабом Всемирной паутины. Аналогичным образом в научной литературе наиболее цитируемые статьи не являются самыми старыми: время от времени новые новаторские статьи превосходят многие более ранние публикации.

Это происходит потому, что узлы (веб-сайты, газеты, пользователи социальных сетей и т. д.) имеют свою собственную индивидуальную привлекательность, способную увеличивать темп, с которой они накапливают связи, давая им преимущество перед гораздо более старыми узлами. Такая привлекательность лишь частично и косвенно отражается их степенью. Описанный в предыдущем разделе параметр привлекательности модели одинаков для всех узлов, поэтому он не позволяет нам проводить различие между узлами и вводить расхождения в их темпах роста степени. Поэтому в модели на основе привлекательности, как и в модели Барабаши–Альберта, хабы по-прежнему являются самыми старыми узлами.

В целях обеспечения возможности становления новых узлов хабами Бьянкони и Барабаши предложили *модель на основе приспособленности*, в которой каждый узел имеет свою индивидуальную привлекательность, именуемую *приспособленностью* (fitness). Значения приспособленности являются внутренними признаками узлов; они не меняются с течением времени. Вероятность связи пропорциональна произведению степени и приспособленности целевого узла.

Модель на основе приспособленности аналогична модели Барабаши–Альберта, но за каждым узлом i закрепляется значение приспособленности $\eta_i > 0$, генерируемое из некоторого распределения $p(\eta)$. Затем на каждом шаге каждая новая связь со стороны нового узла i подсоединяется к старому узлу j с вероятностью

$$P(i \leftrightarrow j) = \frac{\eta_j k_j}{\sum_l (\eta_l k_l)}. \quad (5.12)$$

Если все узлы имеют одинаковую приспособленность, то модель сводится к модели Барабаши–Альберта, так как константа η является фактором, который уравнивает числитель и знаменатель уравнения (5.12), возвращая стандартное предписание преференциального прикреплении.

Если распределение значений приспособленности имеет *бесконечный носитель* (т. е. η может принимать сколь угодно большие значения), то существует эффект «победитель получает все», когда узел с наивысшей приспособленностью связан с большинством мод. Но если распределение значений приспособленности $p(\eta)$ имеет *конечный носитель* (т. е. η имеет конечное максимальное значение), то степенное распределение модели имеет утяжеленный хвост. Примером этого случая является равномерное распределение в единичном интервале. На языке Python можно черпать значения приспособленности с помощью равномерного распределения, используя функцию `gandom()`.

Модель на основе приспособленности генерирует сети с двумя желаемыми свойствами. Во-первых, до тех пор, пока значения приспособленности ограничены, сеть имеет несколько хабов. Во-вторых, высокая приспособленность позволяет узлу конкурировать со всеми своими сверстниками, независимо от их возраста и статуса. Это обусловлено тем, что узлы увеличивают свою степень с темпом, определяемым их индивидуальной приспособленностью. Поэтому узлы с наибольшими значениями приспособленности в конечном итоге достигают наибольших степеней независимо от того, когда они вводятся в систему.

5.5.3. Модель на основе случайного блуждания

Сети, построенные с использованием модели Барабаши–Альберта, имеют очень низкие коэффициенты кластеризации. В целях понимания причины вспомните, что для того, чтобы иметь много треугольников, необходимо, чтобы связи соединяли пары узлов хотя бы с одним общим соседом. Например, если узлы **b** и **c** оба прикреплены к **a**, то связь между **b** и **c** замкнет треугольник **abc** (рис. 5.9). Однако в модели Барабаши–Альберта вероятность того, что узел получит связь, пропорциональна его степени независимо от того, есть ли у новой пары соседей общий сосед или нет. Вот почему треугольники образуются редко. В целях увеличения производства треугольников необходимо ввести механизм, способствующий созданию связей между узлами с общими соседями.

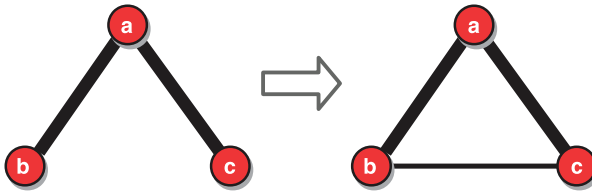


Рис. 5.9 Сильное триадическое замыкание. Индивидуум **a** имеет сильные соединения, обозначенные толстыми связями, с **b** и **c**. В соответствии с принципом сильного триадического замыкания Грановеттера (вставка 5.6) между **b** и **c** должно быть либо в конечном итоге будет, по меньшей мере, слабое соединение

Формирование треугольника путем добавления связи называется *триадическим замыканием* и, возможно, является главным механизмом, объясняющим формирование связей в социальных сетях (вставка 5.6). Это не должно удивлять: многих людей, которых мы знаем, нам представили общие знакомые. Указанный механизм имеет несколько типовых имплементаций. Здесь мы обсуждаем очень интуитивную сетевую модель, именуемую *моделью на основе случайного блуждания*. Ее идея заключается в том, что в дополнение к созданию случайных соединений мы также соединяем с соседями нового соседа – в социальной сети, с друзьями нового друга.

Модель на основе случайного блуждания может начинаться с любой малой сети. Каждая итерация алгоритма состоит из следующих ниже шагов.

1. В сеть добавляется новый узел i с $m > 1$ прикрепленными к нему новыми связями.
2. Первая связь подсоединяется к старому узлу j , выбираемому случайно.

3. Каждая другая связь прикрепляется к случайно отобранному соседу узла j с вероятностью p или к еще одному случайно отобранному узлу с вероятностью $1 - p$.

Параметр p – это вероятность триадического замыкания, потому что, установив связь между i и соседом узла j , к примеру l , мы замыкаем треугольник (i, j, l) . Если $p = 0$, то триадического замыкания нет, и новые узлы выбирают своих соседей совершенно случайно. При $p = 1$ все связи, кроме первой, подсоединяются к соседям первоначально отобранного старого узла, таким образом замыкая треугольники.

Вставка 5.6

Триадическое замыкание и сила слабых уз

В 1973 году социолог Марк С. Грановеттер (Mark S. Granovetter) после долгих переписаний с редакторами опубликовал статью под названием «Сила слабых уз» (The strength of weak ties), которая станет самой цитируемой статьей в социологии. В документе излагается тесное взаимоотношение между тремя фундаментальными признаками социальных сетей: треугольниками, весом связей и сообществами.

Грановеттер ввел принцип *сильного триадического замыкания* в отношении того, как формируются связи в социальных сетях. Если у человека **a** есть прочные (весомые) соединения с двумя индивидуумами **b** и **c**, то очень вероятно, что **b** и **c** являются друзьями или что они в конечном итоге станут друзьями. Тому могут быть разные причины. Если **b** и **c** проводят много времени с **a**, то вполне вероятно, что они в конечном итоге встретятся через **a**. Кроме того, поскольку **a** является хорошим другом обоих, **b** и **c** будут склонны доверять друг другу. Наконец, если **b** и **c** будут продолжать игнорировать друг друга, то это может стать источником стресса для группы. Сильное триадическое замыкание предписывает, что между **b** и **c** должна быть связь, поэтому **a**, **b** и **c** образуют треугольник (рис. 5.9). Этим формализуется взаимосвязь между треугольниками и весами связей.

Социальное сообщество – это круг людей, которые много взаимодействуют друг с другом из-за семейных уз, рабочих отношений и т. д. (Мы обсуждаем сообщества в главе 6.) Грановеттер аргументировал, что связи с большими весами, сигнализирующие о *сильных узлах* между индивидуумами, скорее всего, будут находиться внутри одного и того же сообщества, тогда как низковесные связи (слабые узлы), как правило, лежат между сообществами. Интуиция подсказывает, что люди в разных кругах имеют ограниченный контакт. Грановеттер привел аргументацию в поддержку своей теории. Предположим, что существует сильная связь между двумя индивидуумами **a** и **b**, принадлежащими к разным сообществам. У каждого из них, скорее всего, будут прочные связи с другими членами их собственного сообщества. Давайте предположим, что **a** является близким другом **c**. Из-за сильного триадического замыкания также должна существовать связь между **b** и **c**, потому что **ab** и **ac** являются сильными узлами. Но связь, соединяющая два сообщества, вряд ли может быть стороной треугольников, потому что в противном случае сообщества не были бы хорошо сепарированы, поэтому **ab** должна быть слабой.

Напротив, поскольку между членами одного и того же сообщества существует много прочных уз, внутри сообществ будет много треугольников. Этот аргумент намекает на взаимную игру между сообществами и весами связей, а также между сообществами и треугольниками. Несмотря на их низкий вес, слабые узлы имеют решающее значение для структуры социальных сетей, поскольку они соединяют сообщества друг с другом, обеспечивая распространение информации по сети.

Указанная модель проиллюстрирована на рис. 5.10. Он создает сети с несколькими треугольниками, которые можно варьировать путем регулировки параметра p^1 . Наибольшая плотность треугольников получается при $p = 1$. Кроме того, если p не слишком мал, то модель генерирует тяжелохвостные степенные распределения. Это обусловлено процессом триадического замыкания. Выбирая соседа старого узла, мы просто отбираем связь сети. Как вы, возможно, помните из раздела 3.3, если мы отбираем связь случайно, то вероятность того, что конечная точка этой связи имеет заданную степень, пропорциональна степени. Таким образом, старые узлы будут получать связи с вероятностью, пропорциональной их степеням, точно так же, как при предпочтительном прикреплении.

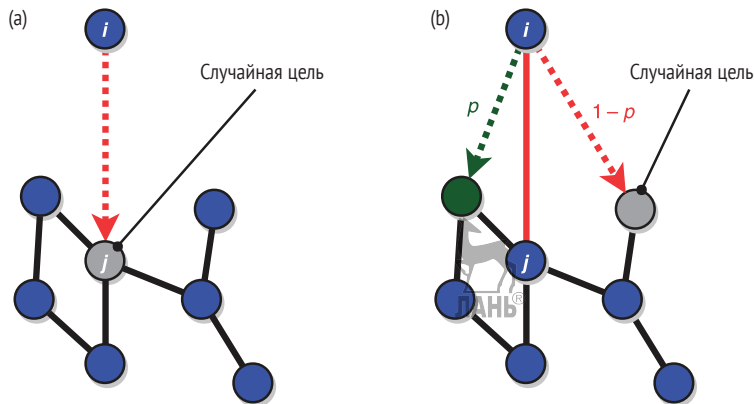


Рис. 5.10 Модель на основе случайного блуждания. (а) Новый узел i прикрепляется к случайно выбранному узлу j . (б) Каждая дополнительная связь, привносимая узлом i , прикрепляется к соседу узла j с вероятностью p , что приводит к образованию треугольников. В противном случае связь прикрепляется к случайно выбираемому узлу

Механизм, используемый в модели случайного блуждания, более интуитивен, чем в модели на основе предпочтительного прикреплении, поскольку он не исходит из допущения, что новые узлы знают о степенях старых узлов. Указанная модель просто разведывает сеть в случайном порядке, и узлы «обнаруживаются» с частотой, пропор-

¹ И снова не путайте этот параметр с параметрами в моделях на основе случайной сети и малого мира, в которых используется одна и та же буква « p ».

циональной их степеням. Другими словами, процесс *триадического замыкания неявно индуцирует преференциальное прикрепление*. С этой точки зрения он в основном эквивалентен *селективному отбору связи* (т. е. выбору связи наугад и прикреплению нового узла к одной из конечных точек связи). Разница в том, что в триадическом замыкании новый узел прикрепляется к обоим конечным точкам отобранной связи, но результирующие степенные распределения в обоих случаях аналогичны. Следовательно, строгая пропорциональность между вероятностью связи и степенью, необходимая для создания широких степенных распределений, может быть обеспечена простыми механизмами, основанными на вариантах случайного выбора.

Наконец, когда p достаточно велик, чтобы имелась достаточно высокая плотность треугольников, модель на основе случайного блуждания создает сети со структурой в форме сообществ, которую мы обсудим в главе 6. Взаимосвязь между треугольниками и сообществами хорошо известна в литературе по науке о сетях (вставка 5.6).

5.5.4. Модель на основе копирования

Из триадического замыкания следует, что в социальных сетях индивидуум копирует контакты кого-то другого. Этот механизм копирования может иметь место и в других контекстах. Вот несколько примеров.

- Дупликация генов – это процесс, посредством которого в ходе молекулярной эволюции создается новый генетический материал. Рассмотрим сеть взаимодействия белок–белок, где каждый узел представляет белок, экспрессируемый геном. Когда ген дублируется, новый узел ген/белок будет взаимодействовать с теми же белками, что и изначальный узел в сети взаимодействия белков. Таким образом, связи, указывающие на эти узлы, копируются.
- Ученые часто обнаруживают новые статьи в разделе библиографических ссылок публикаций, которые они читают, и цитируют их в своих собственных статьях. Делая это, они копируют (некоторые) цитаты из других публикаций.
- Во время онлайн-навигации по Всемирной паутине создатели веб-контента могут обнаруживать релевантные страницы паутины, такие как авторитетные источники или хабовые страницы, которые предоставляют списки ресурсов. Создавая связи, указывающие на эти страницы из недавно созданных страниц, авторы копируют гиперсвязи, ведущие к ним.

Эти сценарии улавливаются рассматриваемой нами сетевой моделью – *моделью на основе копирования*. Она похожа на обсуждавшуюся ранее модель на основе случайного блуждания: новый узел подсоединяется к случайно отбираемому старому узлу, с некоторой

вероятностью, либо к его соседям. Однако в модели на основе копирования нет триадического замыкания; новый узел не прикрепляется одновременно к узлу и (нескольким) его соседям. Поэтому мы получаем сети с хабами, но с малым числом треугольников.

5.5.5. Модель на основе ранга

Из преференциального прикрепления следует, что узлы имеют понимание того, насколько важны другие узлы, поскольку для оценивания вероятности связи и надлежащего распределения связей требуется знание их степеней во время сетевого роста. Мы, возможно, захотим ответить на вопрос, сможем ли мы работать, не зная степеней узлов. В разделах 5.5.3 и 5.5.4 мы увидели, что триадическое замыкание и селективный отбор/копирование связей являются жизнеспособными стратегиями. Здесь мы рассмотрим другой подход.

В реалистичной обстановке более общепринято понимать *относительное значение* вещей, а не их абсолютное значение. Мы с большой уверенностью можем заявить, что Билл Гейтс намного богаче любого из авторов этой книги, даже несмотря на то, что мы игнорируем точную величину его богатства. Утверждение о том, что мы можем ранжировать узлы сети на основе специфической переменной (например, степени или возраста), следовательно, является правдоподобнее, чем утверждение о том, что мы можем точно оценить значения переменной. Эта идея лежит в основе *ранговой модели*.

Мы поддерживаем узлы ранжированными по одному из их свойств, к примеру, по степени. Затем отбираем узлы для получения новых связей с вероятностью, пропорциональной некоторому показателю возведения в степень (как экспоненциации), обратной величине их ранга. Верхний узел будет иметь наибольшую вероятность получения связи, за ним следует узел, занимающий второе место, третье и т. д. То, как вероятность связи затухает вместе с рангом, определяется экспонентным параметром.

Модель на основе ранга может начинаться с любого малого графа с m_0 узлами. Для ранжирования узлов отбирается свойство узла, такое как степень (degree), возраст или некоторая мера приспособленности. Каждая итерация алгоритма состоит из следующих ниже шагов.

1. Все узлы ранжируются на основе интересующего свойства. Узлам назначаются ранги $R = 1, 2$ и т. д. Узел l получает ранг R_l .
2. В сеть добавляется новый узел i с прикрепленными к нему $m \leq m_0$ новыми связями.
3. Каждая новая связь из i подсоединяется к старому узлу j с вероятностью

$$P(i \leftrightarrow j) = \frac{R_j^{-\alpha}}{\sum_l R_l^{-\alpha}}, \quad (5.13)$$

где экспонента $\alpha > 0$ – это параметр.

Узлы, возможно, придется переранжировать на каждой итерации, если свойство ранжирования зависит от связей со стороны новых присоединяемых к сети узлов, как это происходит, например, когда узлы ранжируются по степени.

Верхнеранговый узел (малый ранг) будет получать новую связь с большей вероятностью, чем слаборанговый узел (большой ранг). Если переменной, используемой для ранжирования, является степень, то это означает, что высокостепенные узлы имеют более высокие шансы на привлечение новых связей, чем низкостепенные узлы, как в модели Барабаши–Альберта. Однако фактические значения вероятности связи различны, поскольку они зависят не от степеней, а от рангов.

Как выясняется, модель на основе ранга генерирует сети с тяжелехвостными степенными распределениями для любого свойства, используемого для ранжирования узлов, и любого значения экспонентного параметра. Регулируя экспоненту, можно изменять форму распределения и воспроизводить эмпирические распределения, наблюдаемые во многих реально существующих сетях.

Хабы создаются, даже если узлы имеют частичную информацию о системе, в том смысле, что они знают о существовании только части узлов. Этот факт отражает знакомый сценарий. Вообразите, что вы пишете статью в «Википедии» и хотите установить связи с релевантными новостными статьями. Вы, возможно, воспользуетесь поисковой машиной с целью выявления страниц, на которые устанавливать связь. Поисковая машина представляет страницы, ранжированные по релевантности вашему запросу. Вы ссылаетесь на верхний результат с высокой вероятностью; на второй результат с половиной этой вероятности; на третий с вероятностью в одну треть; и т. д. Возможно, вы даже не потрудитесь заглядывать дальше первой страницы результатов. Ваша статья станет новым узлом «Википедии», связывающимся со старыми узлами в соответствии с процедурой, очень похожей на модель на основе ранга. Это помогает объяснить появление популярных хабов во Всемирной паутине.

5.6. Резюме

Сетевые модели помогают нам понимать базовые механизмы, которые отвечают за характерные структурные признаки, наблюдаемые в реально существующей сети. Базовыми ингредиентами сетевых моделей являются правила, определяющие принцип, по которому узлы прикрепляются друг к другу. Ниже мы перечислим уроки, которые мы извлекли из сетевых моделей, рассмотренных в этой главе.

1. В случайных сетях, генерируемых моделью Эрдеша–Реньи, каждый узел имеет одинаковую вероятность стать соседом любого другого узла. Эти сети имеют короткие пути, но в них очень мало треугольников и нет хабов.
2. Маломировая модель изменяет исходную структуру решетки с высоким средним коэффициентом кластеризации, создавая несколько случайных сокращений между узлами. Несколько сокращений достаточно, чтобы резко сократить расстояния между узлами, индуцируя маломировое свойство, тогда как коэффициент кластеризации остается высоким. Указанная модель не способна создавать хабы.
3. Конфигурационная модель генерирует сети с любой предопределенной степенной последовательностью. Следовательно, структура навязывается «вручную», она не объясняется моделью. Конфигурационная модель нередко используется в качестве базового уровня, чтобы проверять, связано ли какое-либо свойство сети только с ее степенным распределением либо с другими факторами. Это можно делать путем сравнения интересующего свойства в изначальной системе и в рандомизированных сетях с той же самой степенной последовательностью, созданной моделью.
4. Реалистичные сетевые модели включают сетевой рост, при котором узлы и связи добавляются в граф с течением времени. Это соответствует эволюции многих реально существующих сетей, таких как интернет, Всемирная паутина и т. д.
5. Преференциальное прикрепление является ключевым механизмом, объясняющим появление хабов: чем выше степень узла, тем выше вероятность того, что он будет соединен с другими узлами.
6. Модель Барабаши–Альберта с ее сочетанием сетевого роста и преференциального прикрепления порождает сети с тяжелохвостным степенным распределением, следовательно, объясняя появление хабов.
7. Преференциальное прикрепление может индуцироваться в неявной форме простыми процессами с участием вариантов случайного выбора, подобными триадическому замыканию и селективному отбору связей.

8. С целью преодоления ограничений модели Барабаши–Альберта было предложено несколько моделей за счет введения таких ингредиентов, как привлекательность, приспособленность, триадическое замыкание и ранжирование.

5.7. Дальнейшее чтение

Модель на основе случайного графа была представлена в один и тот же год Эрдешем и Реньи (1959) и Гильбертом (1959), хотя аналогичная идея была выдвинута в более ранней статье Соломонова и Рапопорта (1951). Модель Гильберта часто ошибочно приписывается в литературе Эрдешу и Реньи. Мы называем среднее число регулярных контактов, которые люди могут поддерживать, числом Данбара (Dunbar, 1992).

Маломировая модель была разработана Уоттсом и Строгацем (1998). Моллой и Рид (1995) предложили конфигурационную модель. Экспоненциальные случайные графы были введены Холландом и Лейнхардтом (1981).

Барабаши и Альберту обычно приписывают модель на основе преференциального прикрепления, часто именуемую моделью Барабаши–Альберта, или моделью ВА (Барабаши и Альберт, 1999). Другие ученые предлагали эту модель и раньше; ближайшим предшественником была статья Прайса (1976). Нелинейное преференциальное прикрепление было исследовано Крапивским и соавт. (2000) и Крапивским и Реднером (2001). Привлекательность была добавлена к преференциальному прикреплению Дороговцевым и соавт. (2000). Модель на основе приспособленности была предложена Бьянкони и Барабаши (2001).

Грановеттер (1973) является автором новаторской статьи «Сила слабых уз». Модель на основе случайного блуждания была введена Васкесом (2003). Модель на основе копирования была идеей Клейнберга и соавт. (1999), которая возникла во время ранних исследований графа Всемирной паутины. Несколько авторов предложили модели на основе дубликации генов (Вагнер, 1994; Бхан и соавт., 2002; Соле и соавт., 2002; Васкес и соавт., 2003а). Ранговая модель была разработана Фортунато и соавт. (2006).

Упражнения

- 5.1 Ознакомьтесь с учебным материалом главы 5 в репозитории книги на GitHub¹.

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

- 5.2 В чем разница между случайным графом Эрдеша и Реньи и графом Гильберта?
- 5.3 Предположим, что мы хотим построить случайный граф с 1000 узлами и примерно 3000 связями. Дайте значение вероятности связи p , которая может привести к такому исходу.
- 5.4 Предположим, вы создаете случайную сеть с 50 узлами и хотите, чтобы средняя степень узла составляла 10. Какое приближенное значение p вы бы использовали в этом случае?
- 5.5 Имея случайную сеть из 50 узлов и среднюю степень узла $\langle k \rangle = 10$, какое из следующих ниже значений, вероятно, будет ближе всего к средней длине пути этой сети?
- 5.
 - 0.
 - 25.
 - 5.
- 5.6 Рассмотрите семейство случайных сетей со средней степенью $\langle k \rangle = 10$. Сколько узлов нам, вероятно, понадобится для генерирования такой сети со средней длиной пути $\langle \ell \rangle = 3.0$? (Подсказка: если вы используете стратегию «угадать и проверить», убедитесь, что она остается неизменной для различных значений N . Каждое из них будет предусматривать другое значение p .)
- 60.
 - 100.
 - 250.
 - 500.
- 5.7 Постройте случайную сеть с 1000 узлами и $p = 0.002$. Постройте график ее степенного распределения. (Подсказка: мы показываем, как строить распределение в главе 3 книги.) Ответьте на следующие ниже вопросы.
- Какова наибольшая степень сети?
 - Какова мода распределения (т. е. наиболее распространенное значение степени)?
 - Является ли сеть связной? Если нет, то сколько узлов находится в гигантской компоненте?
 - Каков средний коэффициент кластеризации? Сравните его с вероятностью связи p ?
 - Каков диаметр сети?
- 5.8 Рассмотрите следующий процесс. Сначала начните с множества из N узлов и без связей. Это явно разъединенная (несвязная) сеть. Затем, один за другим, добавляйте связь между двумя узлами, которые еще не соединены друг с другом. Продолжайте до тех пор, пока у вас не будет полной сети. Сколько шагов в этом процессе?

- 5.9** Рассмотрите пошаговый процесс из предыдущего вопроса. Предположим, что на каждом шаге этого процесса вы записываете размер самой крупной компоненты. Что из следующего, как правило, верно в отношении последовательности размеров самых крупных компонент при добавлении ребер?
- Медленно увеличивается в начале последовательности, очень быстро увеличивается в конце последовательности.
 - Медленно увеличивается до некоторого порога, очень быстро увеличивается в течение короткого времени, а затем медленно увеличивается после этого.
 - Увеличивается с постоянной скоростью от начала до конца последовательности.
 - Очень быстро увеличивается в начале последовательности, затем уменьшается и медленно увеличивается до самого конца.
 - Увеличивается и уменьшается случайным образом.
- 5.10** Воспроизведите график из рис. 5.2 для сетей с 1000 узлами. (Подсказка: используйте функцию библиотеки NetworkX, которая генерирует случайные сети.) Используйте 25 равно отстоящих значений вероятности связи в интервале $[0, 0.005]$. Для каждого значения сгенерируйте 20 разных сетей, вычислите относительный размер гигантской компоненты и сообщите среднее значение и стандартное отклонение на графике.
- 5.11** В чем причина того, что случайные сети не являются хорошими моделями социальных сетей?
- Случайные сети, как правило, не являются связными.
 - Случайные сети имеют малую среднюю длину кратчайшего пути.
 - Узлы в случайных сетях имеют очень разные степени.
 - Случайные сети имеют низкие коэффициенты кластеризации.
- 5.12** Рассмотрите кольцевую решетку, подобную решетке на рис. 5.5(а), с 100 узлами, каждый из которых соединен со своими четырьмя ближайшими соседями (по два с каждой стороны от него). Каков средний коэффициент кластеризации? Играет ли размер сети важную роль? (Подсказка: учитывая симметрию, достаточно рассчитать коэффициент кластеризации любого узла.)
- 5.13** Для улавливания какого свойства реально существующих социальных сетей, которое не находится в случайных графах Эрдеша–Реньи, полезна модель Уоттса–Строгаца?
- Короткие средние длины пути.
 - Большие средние длины пути.
 - Низкий коэффициент кластеризации.
 - Высокий коэффициент кластеризации.



- 5.14** Воспроизведите график из рис. 5.5(b), рассчитав средний кратчайший путь ($\langle \ell \rangle$) и средний коэффициент кластеризации (C) для сетей Уоттса-Строгаца, построенных для разных значений вероятности переподсоединения p . Возьмите 20 равно отстоящих значений p в интервале между 0 и 1. Для каждого значения p постройте 20 разных сетей и вычислите среднее значение пути $\langle \ell \rangle$ и C . В целях построения графика двух кривых на общей оси y вы можете нормализовать значения, разделив их на соответствующие значения при $p = 0$.
- 5.15** Постройте сети Уоттса-Строгаца с 1000 узлами, $k = 4$, и вот с этими значениями для вероятности переподсоединения: $p = 0.0001$, 0.001 , 0.01 , 0.1 и 1 . Вычислите и сравните их степенные распределения, выведя их на одной и той же диаграмме.
- 5.16** Рассмотрите сеть аэропортов США (USAN). Создайте ее рандомизированную версию (RUSAN), используя конфигурационную модель. Для этого возьмите степенную последовательность сети и примените функцию `configuration_model()` библиотеки `NetworkX`. Выполните следующие ниже задачи.
1. Подтвердите, что степенное распределение идентично распределению USAN.
 2. Сравните средние кратчайшие пути USAN и RUSAN. Как вы интерпретируете разницу в значениях?
 3. Сравните средние коэффициенты кластеризации USAN и RUSAN. Как вы интерпретируете разницу в значениях?
- 5.17** Какие из следующих ниже признаков вы ожидали бы найти в модели Барабаши–Альберта, но не в случайной сети с одинаковым числом узлов и ребер?
- a. Узлы со степенью больше единицы.
 - b. Хабовые узлы со степенью во много раз большей, чем у обычного узла.
 - c. Короткие средние длины пути.
 - d. Большие средние длины пути.
- 5.18** Постройте сеть Барабаши–Альберта с 1000 узлами и $m = 3$. Выполните следующие ниже задачи.
1. Постройте график степенного распределения сети в двойной логарифмической шкале.
 2. Выведите среднюю степень, посмотрите, как она соотносится с m , и интерпретируйте результат.
 3. Рассчитайте средний коэффициент кластеризации.
 4. Убедитесь, что граф является связным.
 5. Рассчитайте средний кратчайший путь.
- 5.19** Постройте случайный граф Эрдеша–Реньи с тем же числом узлов и связей, что и у сети Барабаши–Альберта в предыдущем упражнении.

1. Выведите степенное распределение и сравните его с распределением сети Барабаши–Альберта, нарисовав их на одном и том же графике в двойной логарифмической шкале.
 2. Вычислите средний коэффициент кластеризации и среднюю длину кратчайшего пути и сравните их с соответствующими значениями для сети Барабаши–Альберта. Интерпретируйте результаты.
- 5.20 Модели на основе привлекательности и приспособленности основаны на одинаковой идее о том, что узлы обладают внутренней привлекательностью, не связанной с их степенью. В чем различия между этими двумя моделями?
- 5.21 Предположим, что в модели на основе приспособленности приспособленность узла совпадает с его степенью. Сможете догадаться, какое степенное распределение будет иметь результирующая сеть? (Подсказка: поможет обсуждение темы нелинейного предпочтительного прикрепления в разделе 5.5.)
- 5.22 Укажите причину, по которой сети, генерируемые моделью Барабаши–Альберта, не имеют большого числа треугольников.
- 5.23 Если вы используете онлайн-социальную сеть, такую как Facebook, Instagram или LinkedIn, то рассмотрите свои связи в этой сети: сколько сильных связей и сколько слабых связей?
- 5.24 Селективный отбор связей состоит в выборе связи и подсоединения нового узла к одной из ее конечных точек. Предположим, что мы подсоединяем новый узел к обеим конечным точкам. В чем будет отличие от модели случайного блуждания? И чем будут отличаться сети, генерируемые этими двумя моделями?
- 5.25 Предположим, что мы хотим построить такую сеть, в которой существует релевантное число квадратов (циклов длиной четыре). Основываясь на том, что вы узнали о триадическом замыкании, не могли бы вы предложить механизм, стимулирующий формирование квадратов?
- 5.26 Рассмотрите две версии ранговой модели с разными критериями ранжирования. В первой версии узлы ранжируются по возрасту (времени, прошедшему с момента их добавления в сеть). Во второй версии узлы ранжируются по их степени. Есть ли разница между сетями, генерируемыми этими двумя моделями, и если да, то в чем она заключается?
- 5.27 Сеть socfb-Northwestern25 в репозитории книги на GitHub представляет собой снимок сети Facebook Северо-Западного университета. Узлы – это анонимные пользователи, а связи – это дружеские отношения. Загрузите эту сеть в граф библиотеки

NetworkX; убедитесь, что вы используете надлежащий класс графа для ненаправленной невзвешенной сети. После того как вы измерите число узлов и связей, примените функцию `nx.gnm_random_graph()`, чтобы создать отдельную случайную сеть с тем же числом узлов и связей, что и в графе Facebook. Используйте эту случайную сеть, чтобы ответить на следующие ниже вопросы.

1. Каков 95-й процентиль для степени в случайной сети (т. е. такое значение, при котором 95 % узлов имеют эту степень или меньше)?
2. Мы имеем дело со случайной сетью, поэтому некоторые свойства будут несколько отличаться при каждом ее генерировании. Истина или ложь: при заданных фиксированных параметрах N и L все случайные сети, созданные с помощью функции `gnm_random_graph()`, будут иметь одинаковую среднюю степень.
3. Какой из следующих ниже контуров лучше всего описывает степенное распределение в этой случайной сети?
 - a. Равномерный: степени узлов равномерно распределены между минимальным и максимальным.
 - b. Нормальный: большинство степеней узлов близко к среднему значению, быстро снижаясь в обоих направлениях.
 - c. Правосторонний: большинство степеней узлов относительно малы по сравнению с размахом степеней
 - d. Левосторонний: большинство степеней узлов относительно велико по сравнению с размахом степеней.
4. Оцените среднюю длину кратчайшего пути в этой случайной сети, используя случайную выборку в размере 1000 пар узлов.
5. Каков средний коэффициент кластеризации этой случайной сети? Ответьте с прецизионностью по меньшей мере до двух знаков после запятой.





Кластер: группа, или скопление, похожих вещей или людей, расположенных или оказывающихся близко друг к другу.

Когда вы смотрите на компоновку сети, первое, что можно заметить, – это то, что узлы сгруппированы в *сообщества*, также именуемые *кластерами* или *модулями*, – множества узлов с относительно более высокой плотностью соединений внутри, чем между ними (рис. 6.1).

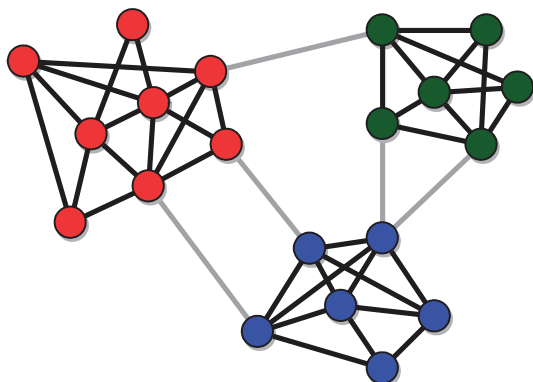


Рис. 6.1 Сеть с тремя сообществами, обозначенными цветами узлов

Сообщества нередко рассказывают нам о принципе организации сети и о функциях, которые она предоставляет. Например, плотные кластеры связанных нейронов в головном мозге часто синхронизированы в своих шаблонах срабатывания. В сети взаимодействия белок-белок группы связанных (взаимодействующих) белков, как правило, ассоциированы с той или иной биологической функцией внутри организма. Иногда мы можем логически выводить роль неизвестного гена в отношении сложного заболевания, глядя на кластер, к которому он принадлежит в генно-регуляторной сети. Во Всемирной паутине, как мы обсуждали в разделе 4.2.5, кластеры страниц с многочисленными гиперсвязями, указывающими друг на друга, обычно идентифицируют тему. А в социальных сетях, как обсуждалось в разделе 2.1, сообщества друзей разделяют между собой важные признаки, такие как политические убеждения. Социальные сообщества могут оказывать значительное влияние на общественное мнение. Например, рассмат-

ривая диффузионную сеть политически заряженных мемов в Twitter (рис. 0.3), мы сразу же отмечаем, что люди поделены на два отдельных сообщества, которые мало взаимодействуют друг с другом.

Аналогичная картина представлена на рис. 6.2, где показана ретвитная сеть политических мемов в Соединенных Штатах. И снова мы наблюдаем два в основном изолированных сообщества. Когда мы проинспектируем нескольких пользователей, становится ясно, что эти два кластера соответствуют политическим правым и левым. Как обсуждалось в разделе 4.5, такой сценарий иногда называется *эхокамерой* или *фильтрным пузырем*, указывая на то, что человек подвергается воздействию людей только со схожими идеями и мнениями. Хотя в реальной жизни совершенно нормально иметь друзей, подобных нам, с помощью онлайн-социальных сетей и социальных медиа легче отфильтровывать разные точки зрения, поскольку нас побуждают общаться с людьми, которые похожи на нас или с которыми у нас уже есть общие друзья. У нас также есть инструменты, позволяющие легко отказываться в общении или игнорировать людей, с которыми мы не согласны, что немного труднее сделать в реальной жизни. Существует теория, что, когда мнения не оспариваются, усиливаются предубеждения, и может возникать поляризация.

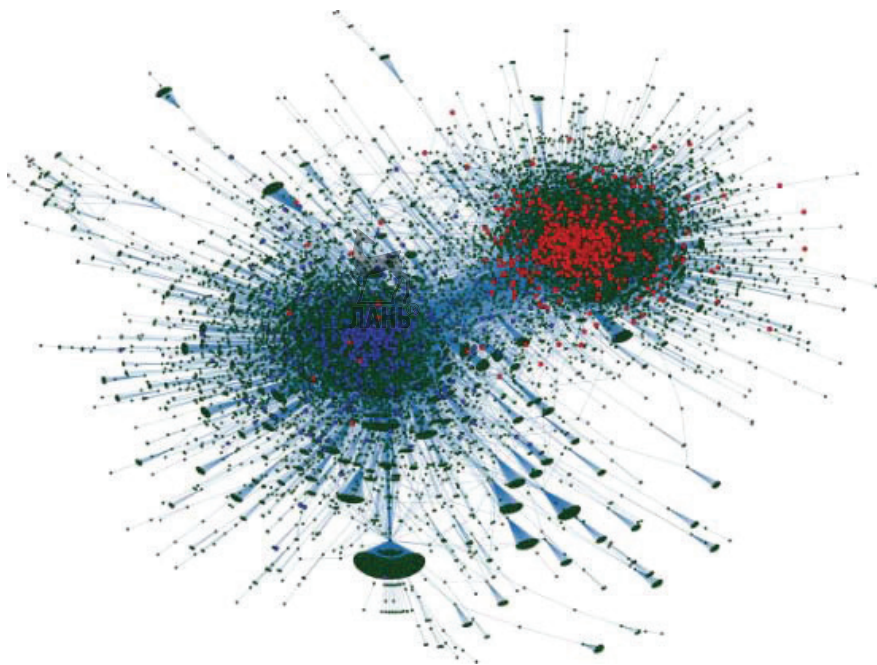


Рис. 6.2 Ретвитная сеть политических хештегов в Twitter до выборов в США 2010 года. Цветные узлы представляют выборку пользователей, классифицированных как либеральные (синий) или консервативные (красный). Связь между двумя узлами указывает на ретвит одного из двух соответствующих пользователей другим

Знание структуры сети, принимающей форму сообществ, позволяет нам классифицировать узлы на основе их положения в своих собственных кластерах. Узлы, которые полностью вложены в кластер, поскольку их соседи принадлежат к одному и тому же кластеру, представляют ядро группы, так как они не смешиваются с членами других групп. Узлы, расположенные на границе сообщества, имеют соседей как внутри, так и за пределами своей группы и служат привратниками между различными частями сети. В силу этого они играют важную роль в диффузионных процессах, происходящих в сети. Если мы хотим остановить распространение слухов или фальшивых новостей в социальной сети, прекратить распространение эпидемии в контактной сети или обеспечить, чтобы критически важная информация доходила до всех сообществ, – вот те узлы, которые нам нужно контролировать.

Учитывая важность сообществ для понимания функций сети и индивидуальных узлов, крайне важно иметь возможность обнаруживать в сети сообщества. Иногда сообщества очень очевидны, и, для того чтобы их увидеть нам лишь нужно скомпоновать узлы на плоскости так, чтобы соединенные узлы располагались близко друг к другу. Именно эта идея лежит в основе популярных *алгоритмов компоновки сети по направлению силы*, описанных в главе 1. Например, на рис. 6.2, поскольку люди в сообществе (либеральном или консервативном) тесно соединены друг с другом, они в итоге группируются вместе в компоновке сети. Однако многие представляющие интерес сети намного больше, чем те, для которых возможно создать содержательные визуализации. И даже во многих малых системах визуализация не помогает выявлять кластеры. Поэтому необходимо разрабатывать алгоритмы, способные обнаруживать сообщества автоматически, исходя из знания сетевой структуры и, возможно, других входных данных (например, желаемого числа кластеров).

Задача выявления сообществ в сетях является поистине междисциплинарной темой, и в силу этого она носит разные названия: *обнаружение сообществ*, *раскрытие сообществ* и *кластеризация*¹ среди прочих. Группирование узлов в сообщества считается задачей *неконтролируемого* классифицирования, поскольку у нас нет точных априорных знаний или примеров того, как должны выглядеть результирующие разделы. И действительно, единого определения сообщества не существует. Естественная интуиция подсказывает, что между узлами в одном сообществе должно быть больше связей, чем между узлами в разных сообществах. Другими словами, плотность связей внутри (между) сообществами должна быть выше (ниже), чем совокупная плотность сети (уравнение (1.3)). Этот критерий может быть математически формализован целым рядом разных способов. Вот почему в научной литературе мы находим много методов кластеризации.

¹ Мы столкнулись с термином «кластеризация», когда вводили коэффициент кластеризации в разделе 2.8; в этой главе мы используем этот термин только для обозначения структуры в форме сообществ.

В этой главе предлагается краткое введение в данную задачу и ее наиболее популярные решения. Мы начнем с введения некоторых базовых элементов: главных переменных, классических определений сообщества и высокоуровневых свойств разделов. Затем обсудим две взаимосвязанные задачи, деление сети и кластеризацию данных, которые внесли в настоящую тему много инструментов и технических приемов. Наконец, мы представим несколько широко принятых алгоритмов, а также стандартные процедуры тестирования технических приемов кластеризации.

6.1. Базовые определения

6.1.1. Переменные сообщества

В типичной ситуации сообщество представляет собой связную подсеть. Рассмотрим сообщество, проиллюстрированное зелеными узлами на рис. 6.3. Пурпурные узлы являются внешними, но соединены с сообществом, тогда как несколько оставшихся узлов сети показаны черным цветом. Синие связи соединяют сообщество с остальной частью сети. Ключевые переменные сообщества, которые мы будем использовать на протяжении всей этой главы, таковы:

- *внутренняя и внешняя степень* узла в сообществе – число соседей соответственно внутри и за пределами сообщества. На рис. 6.3 внутренняя степень зеленого узла – это число прикрепленных к нему черных связей, а внешняя степень – это число синих связей;

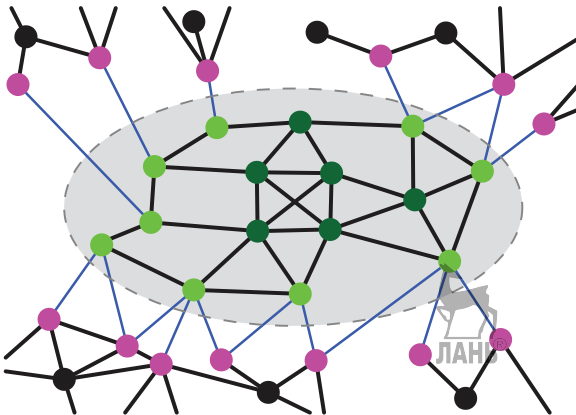


Рис. 6.3 Схематическое изображение сообщества (внутри серого овала) и его ближайших соседей. Перепечатано из Фортунато и Хрика (2016) с разрешения издательства Elsevier

- число *внутренних связей* с сообществом – число связей, соединяющих два узла в сообществе (черные связи в овале на рис. 6.3);
- *степень сообщества* – сумма степеней узлов в сообществе; сумма числа соседей каждого зеленого узла на рис. 6.3;
- *плотность внутренних связей* – соотношение между числом внутренних связей и максимальным числом связей, которые могут существовать между любыми двумя узлами сообщества. Это то же самое, что и плотность, определенная в главе 1, но для подсети, представленной сообществом.

Давайте сформулируем определения и обозначения наших переменных сообщества чуть-чуть формальнее. Предположим, что у вас есть сообщество C .

- Число узлов и внутренних связей в C равно соответственно N_C и L_C .
- *Внутренняя степень* k_i^{int} и *внешняя степень* k_i^{ext} узла i по отношению к сообществу C – это число связей, соединяющих i соответственно с узлами в C и с остальной сетью. Поскольку каждый сосед узла i должен находиться либо внутри, либо снаружи сообщества C , степень узла i равна $k_i = k_i^{int} + k_i^{ext}$. Если $k_i^{ext} = 0$ и $k_i^{int} > 0$, то i имеет соседей только внутри C и является *внутренним узлом* сообщества C (темно-зеленые узлы на рисунке). Если $k_i^{ext} > 0$ и $k_i^{int} > 0$ для узла $i \in C$, то i имеет соседей как внутри, так и снаружи сообщества C и является *пограничным узлом* сообщества C (ярко-зеленые узлы на рисунке). Если $k_i^{int} = 0$, то узел не пересекается с C , так как у него нет соседей внутри сообщества (черные узлы на рисунке).
- *Плотность внутренних связей* задается уравнением

$$\delta_C^{int} = \frac{L_C}{\binom{N_C}{2}} = \frac{2L_C}{N_C(N_C - 1)}. \quad (6.1)$$

Обратите внимание, что оно эквивалентно уравнению (1.3) для узлов и связей, внутренних для C , потому что мы исходим из допущения ненаправленности сети; максимальное число внутренних связей, которые может иметь сообщество с N_C узлами, составляет $\binom{N_C}{2}$.

- *Степень, или объем, сообщества* представляет собой сумму степеней узлов в C :

$$k_C = \sum_{i \in C} k_i. \quad (6.2)$$

Все определения соблюдаются для ненаправленных и невзвешенных сетей. Расширения для взвешенных сетей просты – нам просто нужно заменить степени на силы. Например, внутренняя степень узла становится *внутренней силой*, которая представляет собой сумму весов связей, соединяющих его с узлами в сообществе. В случае направленных сетей нам необходимо различать входящие и исходящие связи. Расширения указанных мер довольно просты в имплементации, но их полезность неясна.



6.1.2. Определения сообщества

На рис. 6.1 представлена традиционная картина структуры сетевого сообщества. Она подчеркивает две вещи: (1) сообщества обладают высокой *когезией*, или сплоченностью (т. е. у них много внутренних связей, поэтому узлы держатся вместе) и (2) сообщества имеют высокую степень *сепарации* (т. е. они соединены друг с другом малым числом связей). Классические определения структур, подобных сообществам, основаны на когезии и взаимной игре когезии и сепарации.

Определения, основанные только на когезии, трактуют сообщество как самостоятельную систему, игнорируя остальную сеть. Наиболее популярным понятием сообщества такого типа является понятие *клики*, определенное в главе 1 как полная подсеть, все узлы которой соединены друг с другом (рис. 6.4). Однако в целом сообщества не так плотны, как клики. Более того, все узлы имеют одинаковую внутреннюю степень внутри клики, тогда как в сообществах реально существующих сетей некоторые узлы важнее, чем другие, что отражается в их шаблонах гетерогенного связывания.

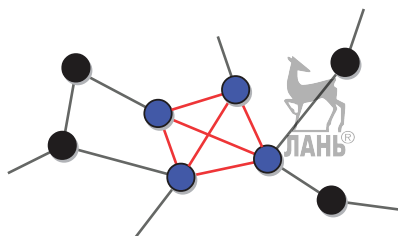


Рис. 6.4 Часть сети, включающая клику из 4 узлов, обозначенную синими узлами и красными связями

В целях более полезного определения сообщества мы должны учитывать как внутреннюю когезию кандидатной подсети, так и ее сепарацию от остальной сети. Популярная идея состоит в том, что сообщество – это такая подсеть, в которой *число внутренних связей больше, чем число внешних связей*. Эта идея вдохновила на следующие ниже определения.

- *Сильное сообщество* – это подсеть, в которой у каждого узла больше соседей в подсети, чем в остальной сети. Другими словами, внутренняя степень каждого узла в сильном сообществе превышает его внешнюю степень.
- *Слабое сообщество* – это подсеть, в которой сумма внутренних степеней всех узлов превышает сумму их внешних степеней.

Сильное сообщество с неизбежностью также является слабым сообществом: если неравенство между внутренней и внешней степенью соблюдается для каждого узла, то оно должно соблюдаться и для суммы по всем узлам. Обратное не является в целом верно: если неравенство между внутренней и внешней степенью соблюдается для суммы, то оно может быть нарушено для одного или нескольких узлов.

Недостатком этих определений является то, что они сепарируют рассматриваемое сообщество от остальной сети, которая рассматривается как единый объект. Но остальная сеть в свою очередь может быть поделена на сообщества. Если подсеть S является надлежащим сообществом, то можно было бы ожидать, что каждый из ее узлов будет теснее прикреплен к другим узлам в S , чем к узлам в любой другой подсети раздела. Такая идея вдохновила на менее строгие определения сильного и слабого сообщества.

- *Сильное сообщество* – это такое сообщество, в котором каждый узел внутри имеет больше соседей, чем в любом другом сообществе.
- *Слабое сообщество* – это такое сообщество, в котором сумма внутренних степеней узлов внутри него превышает сумму их внешних степеней в каждом другом сообществе. Внешняя степень узла в сообществе, отличном от его собственного, – это число соседей в этом сообществе.

Сильное (слабое) сообщество в соответствии с предыдущим определением с неизбежностью также является сильным (слабым) сообществом в соответствии с менее строгим определением. Обратное в целом неверно, как показано на примере на рис. 6.5. В частности, подсеть может быть сильным сообществом в менее строгом смысле, даже если все его узлы имеют внутреннюю степень меньше, чем их соответствующая внешняя степень.

Как мы убедились, традиционные определения сообществ основаны на подсчете связей (внутренних и внешних) различными способами. Но число связей обычно увеличивается вместе с размером сообщества. Следовательно, сравнение внутренних и внешних степеней различных сообществ зависит от их размеров. В идеале мы хотели бы сравнивать *вероятности*: если узлы внутри подсети соединены с большей вероятностью, чем узлы в разных подсетях, то мы бы назвали такую подсеть сообществом. Вероятности устраняют неприятные зависимости от размера сообществ. Но как определить вероятности

связей? Для этого нам нужна модель, описывающая принцип формирования связей в сетях со структурой в форме сообществ. В разделах 6.3.4 и 6.4.1 представлены вероятностные модели для определения и обнаружения сообществ.

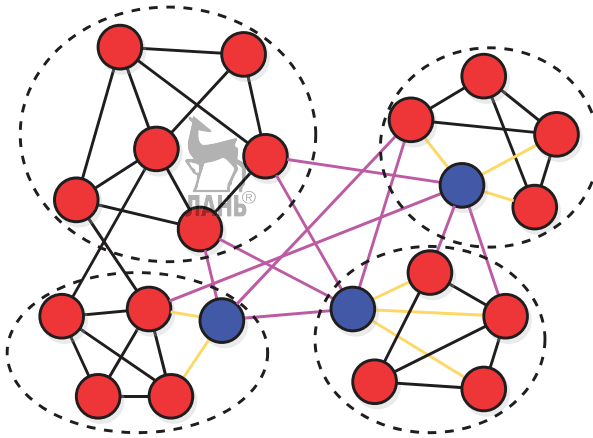


Рис. 6.5 Сильные и слабые сообщества. Четыре подсети, заключенные в пунктирные контуры, являются слабыми сообществами в соответствии с обоими приведенными нами определениями. Они также являются сильными сообществами в соответствии с менее строгим определением, поскольку внутренняя степень каждого узла превышает число связей, соединяющих узел с другими сообществами. Однако три подсети не являются сильными сообществами в более строгом смысле, потому что некоторые узлы (синего цвета) имеют внешнюю степень, большую, чем их внутренняя степень (внутренние и внешние связи этих узлов окрашены соответственно в желтый и пурпурный цвета). Адаптировано по материалам Фортунато и Хрик (2016)

Действительно ли необходимо определение понятия сообщества? На самом деле большинство методов кластеризации сетей не требует точного определения сообщества, как мы увидим в разделе 6.3. Однако предварительное определение критериев для сообществ бывает полезно при проверке надежности окончательных результатов.

6.1.3. Разделы

Раздел – это деление или группировка сети на такие сообщества, в которых каждый узел принадлежит только одному сообществу. Число всех возможных разделов называется *числом Белла* и увеличивается быстрее, чем экспоненциально, вместе с увеличением числа узлов сети. Например, сеть с 15 узлами имеет 1 382 958 545 возможных разделов! Поэтому для сетей с более чем несколькими узлами безнадежно выбирать наилучший раздел сети, просматривая их все. И действительно, алгоритмы кластеризации обычно проводят разведку лишь крошечной части пространства всех разделов, в которых, скорее всего, будут найдены интересные решения.

Сообщества во многих реально существующих сетях *накладываются* друг на друга (т. е. они делят между собой некоторые свои узлы). Например, в социальных сетях индивидуумы могут принадлежать к разным кругам одновременно, таким как семья, друзья и коллеги по работе. На рис. 6.6 показан пример сети с сообществами, накладывающимися друг на друга. Деление сети на накладывающиеся друг на друга сообщества называется *покрытием*. Число возможных покрытий сети намного превышает и без того огромное число разделов из-за многочисленности путей, которыми кластеры могут накладываться друг на друга.

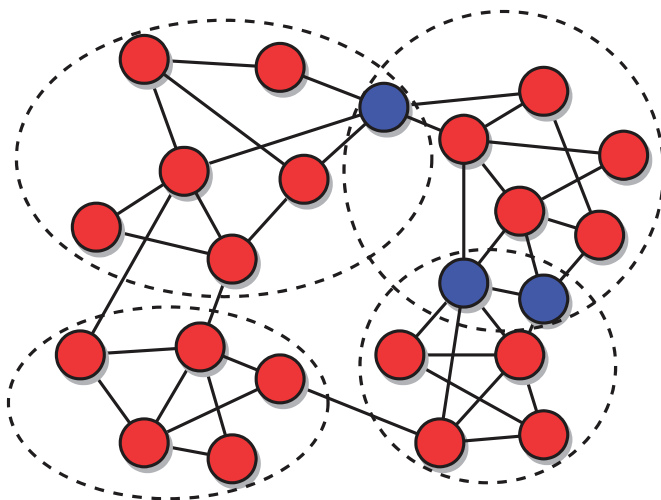


Рис. 6.6 Накладывающиеся друг на друга сообщества. Мы показываем деление сети на четыре сообщества, заключенные в пунктирные контуры. Три из них имеют общие узлы, обозначенные синим цветом. Адаптировано по материалам Фортунато и Хрика (2016)

Разделы бывают *иерархическими* в тех случаях, когда сеть имеет несколько уровней организации в разных масштабах. В этом случае кластеры в свою очередь показывают структуру в форме сообществ с меньшими сообществами внутри, которые опять же могут содержать меньшие сообщества и т. д. (рис. 6.7). Например, в коллаборационной сети сотрудников многонациональной корпорации мы ожидаемо выделим группы сотрудников, работающих в одном и том же филиале, но внутри каждого филиала мы могли бы увидеть дальнейшее деление на отделы. В таких ситуациях каждый уровень иерархии имеет свой смысл, и хороший метод кластеризации должен быть способен раскрывать их все.

Разделы реально существующих сетей нередко являются *гетерогенными*, поскольку некоторые свойства сообщества могут сильно отличаться от одного кластера к другому. Например, часто существует большая разница в размере сообществ. Во Всемирной паутине сообщества примерно соответствуют страницам или веб-сайтам, по-

священным одинаковым или схожим темам. Поскольку некоторые темы являются более общими или популярнее, чем другие, существуют кластеры с миллионами веб-страниц, а также кластеры всего с несколькими сотнями или тысячами страниц. Когезивность тоже очень изменчива. Если мы измерим ее по представленной в разделе 6.1.1 плотности внутренних связей сообщества, то обнаружим, что в нескольких реально существующих сетях ее размер охватывает порядки величины, из чего вытекает, что некоторые кластеры гораздо когезивнее, чем другие. Этот факт может отражать переменную способность групп узлов «привлекать» и связываться друг с другом. Но это также может обуславливаться динамичным характером процесса формирования сообществ: некоторые сообщества развиты полностью, потому что их узлы существуют уже достаточно долгое время, тогда как другие, возможно, все еще развиваются, если многие из их участников были введены совсем недавно.

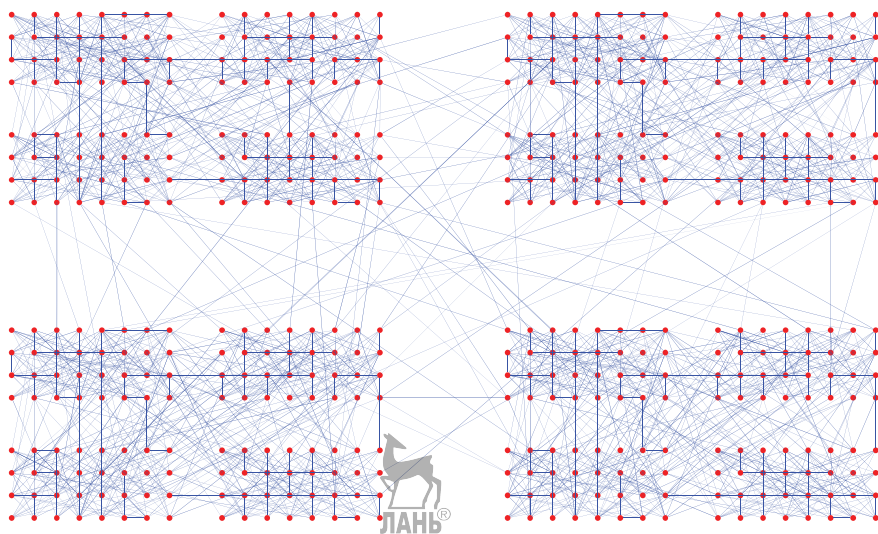


Рис. 6.7 Сеть с иерархическими сообществами. Мы наблюдаем две иерархические структуры в форме сообществ: четыре больших кластера по 128 узлов в каждом и 16 малых кластеров по 32 узла в каждом. Меньшие кластеры полностью входят в состав более крупных

6.2. Смежные проблемы

6.2.1. Деление сети на разделы

Мы видели, что сообщества обычно хорошо сепарированы друг от друга. Выявление хорошо сепарированных подсетей является целью *деления сети на разделы*. Здесь весь фокус внимания сосредоточен на

сепарации независимо от числа связей, которое находится внутри подсетей. Поэтому алгоритмы деления сети на разделы в целом не подходят для обнаружения сообществ. Тем не менее некоторые технические приемы деления используются для обнаружения сообществ, как правило, также в сочетании с другими процедурами. Поэтому полезно ознакомиться с этой задачей.

Деление сети на разделы обусловлено важными практическими задачами. Классическим примером являются параллельные вычисления, когда одна из операционных задач ориентирована на такое распределение операционных задач между процессорами, что число связей обмена данными между группами процессоров, обрабатывающих разные операционные задачи, является низким в целях ускорения вычислений. Деление сети на разделы применяется для решения задач в самых разных областях: решении уравнений в частных производных и разреженных линейных систем уравнений, обработке изображений, гидродинамике, дорожных сетях, сетях мобильной коммуникации, управлении воздушным движением и многих других.

Задача деления состоит в том, чтобы отыскать такое деление сети на заданное число подсетей или кластеров заданного размера, что суммарное число связей, соединяющих узлы в разных подсетях, будет минимизировано. На рис. 6.8 показан пример, в котором требуются два кластера одинакового размера; в данном случае указанная задача также называется *бисекцией графа*, или *рассечением графа пополам*. Множество связей, соединяющих подсети друг с другом, называется *разрезом*, потому что их удаление сепарирует кластеры друг от друга, а их число называется *размером разреза*. Вот почему эта задача также известна в литературе как задача о минимальном разрезе (англ. *minimum cut problem*).

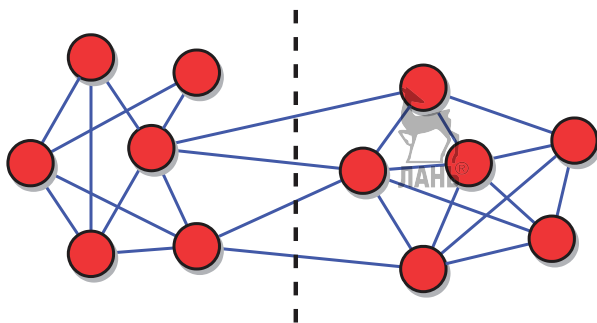


Рис. 6.8 Рассечение графа пополам. Сеть на рисунке имеет 12 узлов. Цель состоит в том, чтобы так разделить его на две части, имеющие одинаковое число узлов, чтоб число связей, соединяющих обе части, было минимальным. Решение обозначено вертикальной пунктирной линией, которая сепарирует две части с минимальным размером разреза из четырех связей

Почему необходимо указывать число кластеров раздела заранее? В конце концов, мы могли бы позволить процедуре деления найти

оптимальное число. Однако это не вариант, потому что она дает тривиальное решение: поскольку мы стремимся минимизировать размер разреза, наилучший возможный раздел состоит из одного кластера, включающего всю сеть целиком, давая нулевой размер разреза. Следующий вопрос заключается в том, почему мы должны указывать размеры кластеров. Причина опять же в том, чтобы избежать тривиальных, неинформативных решений. Например, если сеть имеет один лист (узел со степенью один), то двухчастное деление, состоящее из листа с одной стороны и остальной сети с другой, имеет размер разреза, равный единице, так как существует одна единственная связь, которая сепарирует кластеры. Такое решение нельзя обойти, но оно и не помогает. Главное внимание при делении сети уделяется поиску сбалансированных решений (т. е. разделов, кластеры которых имеют примерно одинаковый размер – как показано на рис. 6.8). В случае двухчастного деления если сеть имеет нечетное число узлов, то в одном кластере будет на один узел больше, чем в другом.

Одним из первых и наиболее популярных методов решения задачи расщепления графа пополам был *алгоритм Кернигана–Лина*. Он основан на очень простой идее: имея первоначальное расщепление сети пополам, мы меняем местами пары узлов между кластерами, например чтобы получить наибольшее уменьшение размера разреза, тогда как размер кластеров не меняется.

Мы начинаем с произвольного деления сети P на два кластера A и B . Например, мы можем отобрать половину узлов в случайном порядке и поместить их в один кластер, а остальные – в другой. Каждая итерация алгоритма состоит из следующих ниже шагов.

1. Для каждой пары узлов i, j , таких что $i \in A$ и $j \in B$, вычислить вариацию в размере разреза между текущим разделом и разделом, полученным путем обмена i и j .
2. Отбирается и обменивается местами пара узлов i^* и j^* , порождающая наибольшее уменьшение размера разреза. Эта пара узлов закрывается на замок; они больше не будут затрагиваться во время этой итерации.
3. Повторять шаги 1 и 2 до тех пор, пока перемена мест незакрытых узлов не приведет к уменьшению размера разреза. Это порождает новый раздел P' , который используется в качестве стартовой конфигурации для следующей итерации.

Процедура заканчивается, когда размер разреза у разделов, полученных после поочередных итераций, будет одинаков, а значит алгоритм не сможет улучшить свой результат. Алгоритм Кернигана–Лина легко расширяется до разделов с более чем двумя кластерами путем обмена узлами между парами кластеров.

Решения, предоставляемые алгоритмом Кернигана–Лина, зависят от выбора первоначальных разделов. Чем хуже качество первоначального раздела – чем больше его размер разреза, – тем хуже окончательное решение и тем больше времени требуется для достижения схождения. В целях получения более качественных исходов мы можем рассматривать несколько случайных разделов и выбирать в качестве первоначального раздела тот, у которого наименьший размер разреза. Еще одним ограничением является то, что алгоритм Кернигана–Лина является жадным алгоритмом, поскольку он пытается минимизировать размер разреза на каждом шаге. Недостатком жадных стратегий является то, что они, скорее всего, будут застревать в локальных оптимумах, таких решениях с субоптимальным размером разреза, что будет приводить любой локальный обмен к менее оптимальным решениям. Более продвинутая версия указанного алгоритма смягчает это ограничение, периодически обмениваясь парой узлов, приводя к увеличению в размере разреза. Принятие таких шагов помогает избегать субоптимальных решений и плотнее приближаться к абсолютному минимуму размера разреза.

Алгоритм Кернигана–Лина широко применяется как технический прием постобработки для улучшения разделов, создаваемых другими методами. Такие разделы могут использоваться в качестве отправных точек для алгоритма, который может возвращать решения с меньшим размером разреза.

Кластеры, выявленные посредством деления сети, хорошо сепарированы, но не обязательно являются когезивными, поэтому они могут не быть хорошими сообществами в соответствии с широко принятым выкоуровневым определением, которое мы дали ранее. Кроме того, для деления сети требуется указывать число кластеров, которые необходимо найти. Хотя это также является признаком ряда методов обнаружения сообществ, было бы предпочтительнее иметь возможность выводить это число непосредственно из данных.

Библиотека NetworkX имеет функцию, которая рассекает сеть пополам с помощью алгоритма Кернигана–Лина:

```
# рассечение пополам с минимальным разрезом: возвращает пару наборов узлов  
partition = nx.community.kernighan_lin_bisection(G)
```

6.2.2. Кластеризация данных

Как мы убедились, сообщества в сетях, как правило, группируют узлы, которые каким-то образом похожи друг на друга: статьи либо веб-сайты, посвященные одним и тем же либо смежным темам, люди, работающие в одной и той же области либо отделе, белки, обладающие одинаковыми либо схожими клеточными функциями, и т. д. Сле-

довательно, обнаружение сообществ является специальной версией гораздо более общей задачи *кластеризации данных* (т. е. такой группировки элементов данных в кластеры на основе некоторого понятия сходства, что элементы в одном кластере похожи друг на друга больше, чем на элементы в других кластерах). Кластеризация данных предлагает ценный набор концепций и инструментов, которые используются регулярно и в сетевой кластеризации.

Существует два основных класса алгоритмов кластеризации данных: *иерархическая кластеризация*, которая обеспечивает вложенную серию разделов, и *раздельная кластеризация*, которая порождает только один раздел. Иерархическая кластеризация применяется при обнаружении сетевых сообществ гораздо чаще, чем раздельная кластеризация, поэтому давайте кратко обсудим ее далее.

Главным ее ингредиентом является *мера сходства* между узлами. Такая мера может выводиться из конкретных свойств узлов. Например, в социальной сети она может указывать на степень близости профилей двух индивидуумов в соответствии с их интересами. Если узлы могут быть встроены в геометрическое пространство, что часто возможно сделать посредством подходящих преобразований, то расстояние между точками, соответствующими паре узлов, можно использовать в качестве меры различия для узлов, такой что точки, которые находятся ближе друг к другу, будут указывать на более похожие узлы. В качестве альтернативы меры сходства могут выводиться только из структуры сети. Классическим примером является *структурная эквивалентность*, которая выражает сходство между окрестностями пары узлов.

Сходство S_{ij}^{SE} пары узлов i и j посредством структурной эквивалентности может быть определено как:

$$S_{ij}^{SE} = \frac{\text{Число соседей, общих среди } i \text{ и } j}{\text{Суммарное число узлов, соседствующих только с } i, \text{ только с } j \text{ либо с обоими}}. \quad (6.3)$$

Например, если соседями узлов i и j являются соответственно (v_1, v_2, v_3) и (v_1, v_2, v_4, v_5) , то $S_{ij}^{SE} = 2/5 = 0.4$, потому что имеется два общих соседа (v_1 и v_2) из пяти отличимых соседей из универсального множества $(v_1, v_2, v_3, v_4, v_5)$. Если пара узлов i, j не имеет общих соседей, то $S_{ij}^{SE} = 0$; если у них есть одинаковый набор соседей, то $S_{ij}^{SE} = 1$. Мы подчеркиваем, что i и j не нуждаются быть соседями: S_{ij}^{SE} может быть вычислено для любой пары узлов.

Следующим шагом является определение сходства между двумя группами узлов. Это можно сделать несколькими способами. Наибо-



лее популярными подходами являются *одиночное соединение*, *полное соединение* и *среднее соединение*. В указанных процедурах сходство между двумя группами определяется с помощью баллов сходства в отношении пар узлов, где каждая пара состоит из одного узла в каждой группе.

Имея меру сходства узлов S и две группы узлов G_1 и G_2 в сети, сходство между G_1 и G_2 может быть вычислено следующим образом. В первую очередь надо измерить сходство S_{ij} всех пар узлов (i, j) , таких что $i \in G_1$ и $j \in G_2$. Сходство $S_{G_1G_2}$ можно определить из этих наборов попарных сходств в соответствии с одним из следующих ниже простых рецептов.

- В одинарном соединении используется максимальное попарное сходство: $S_{G_1G_2} = \max_{i,j} S_{ij}$.
- В полном соединении используется минимальное попарное сходство: $S_{G_1G_2} = \min_{i,j} S_{ij}$.
- В среднем соединении используется среднее попарное сходство: $S_{G_1G_2} = \langle S_{ij} \rangle$.

Технические приемы иерархической кластеризации являются *агломеративными*, если разделы генерируются путем итеративного слияния групп узлов, либо *дивизивными*, если разделы создаются путем итеративного разбиения групп узлов. Здесь мы сосредоточимся на агломеративных процедурах, которые популярны в литературе. Известные примеры дивизивной иерархической кластеризации будут представлены в разделе 6.3.1.

Агломеративная иерархическая кластеризация начинается с тривиального деления на N групп, где каждая группа состоит из одного узла. На каждом шаге пара групп с наибольшим сходством сливаются воедино. Это повторяется до тех пор, пока все узлы не окажутся в одной группе. Поскольку на каждом шаге число групп уменьшается на единицу, процедура дает серию из N разделов, которые можно изобразить посредством *дендрограммы*, или *иерархического дерева*. На рис. 6.9 показана дендрограмма для разделов малой сети. Внизу мы имеем листья дерева, которые являются отдельными узлами, обозначенными их метками. Поднимаясь вверх, пары кластеров сливаются воедино: каждое слияние иллюстрируется горизонтальной линией, соединяющей две вертикальные линии, каждая из которых представляет кластер, узлы которого можно идентифицировать, следуя по вертикальной линии до конца вниз. В целях выделения одного из разделов мы разрезаем дендрограмму горизонтальной линией, как показано на рисунке. Вертикальные линии, разделенные разрезом, указывают на кластеры раздела. Высокие разрезы порождают деления на малое число более крупных групп, тогда как низкие разрезы

порождают деления на большое число более мелких групп. По своей конструкции разделы являются иерархическими: если мы возьмем любые два раздела, то каждый кластер того, который находится в дендрограмме выше, является слиянием кластеров нижнего.

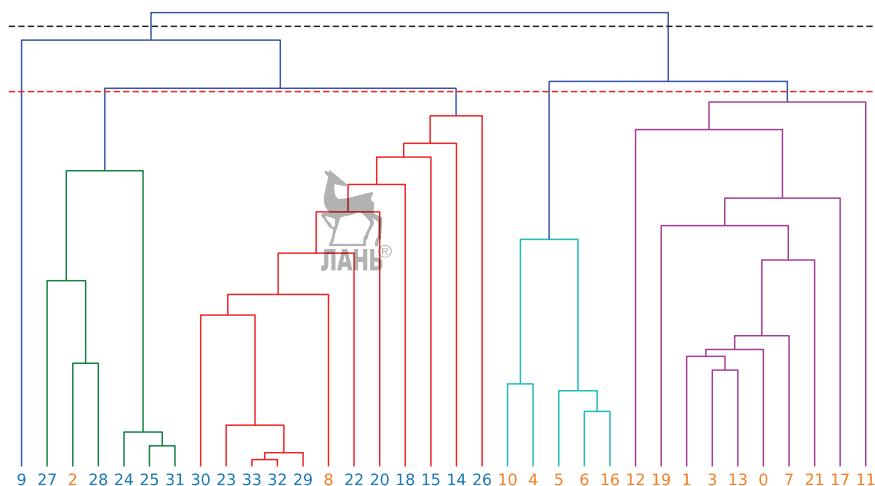


Рис. 6.9 Иерархическая кластеризация, иллюстрируемая дендрограммой иерархических разделов сети клубов карате «Закари» (см. раздел 6.4). Горизонтальные разрезы вычленяют разделы сети. Например, черные и красные пунктирные линии соответствуют делениям соответственно на два и пять кластеров. Каждый кластер включает в себя узлы, «свисающие» из одной из отсеченных ветвей. Цвета представляют реальные и выведенные кластеры, как объясняется в тексте

Иерархическая кластеризация имеет ряд важных ограничений. Во-первых, она предоставляет столько разделов, сколько имеется узлов, без предоставления критерия, который помогает выбирать, какие из них имеют значение для исследуемой сети. Во-вторых, результаты обычно зависят от меры сходства и от критерия, принятого для вычисления сходства групп. Наконец, такого рода алгоритмы довольно медленные, и сети с миллионами или более узлов для них недоступны.

6.3. Обнаружение сообществ

Существует целый ряд методов обнаружения сообществ. Они нередко классифицируются на категории, основываясь на стратегии, используемой для выявления кластеров. Ниже давайте введем четыре популярных подхода: устранение мостов, оптимизация модулярности, распространение меток и стохастическое блочное моделирование. В следующем далее разделе мы покажем, как проводить тестирование результативности алгоритма обнаружения сообществ.

6.3.1. Устранение мостов

Строго говоря, *мост* – это связь, удаление которой разбивает связную сеть на две части. Здесь давайте использовать этот термин свободнее и будем называть мостом любую связь, соединяющую два сообщества. Если бы мы могли отыскивать все мосты, то у нас был бы естественный способ обнаружения кластеров: надо лишь удалить мосты, и сообщества будут отсоединены друг от друга! Тогда задача будет решена путем отыскания связных компонент результирующего разъединенного графа, что является тривиальным. Эта идея лежит в основе популярного алгоритма Гирвана–Ньюмана и нескольких других методов обнаружения сообществ.

Ключевым элементом любого алгоритма, основанного на устранении мостов, является мера, которая позволяет нам выявлять мосты. В случае алгоритма Гирвана–Ньюмана этой мерой является промежуточность связей (раздел 3.1.3). Вспомните, что промежуточность связей в сущности говорит о числе кратчайших путей между парами узлов, которые проходят через связь. Мы ожидаем, что промежуточность связей будет выше для мостов, чем для связей внутри кластеров: многие кратчайшие пути, соединяющие пары узлов в разных сообществах, с неизбежностью проходят через мосты (рис. 6.10). Для сравнения, промежуточность внутренних связей в среднем намного ниже: из-за высокой плотности внутренних связей сообщество пересекается большим числом альтернативных кратчайших путей, поэтому маловероятно, что какой-либо из них является предпочтительным маршрутом.

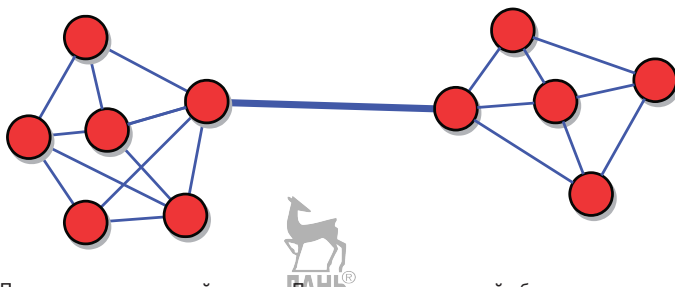


Рис. 6.10 Промежуточность связей и мосты. Промежуточность связей обычно высока, если связь является мостом. На рисунке связь посередине является мостом; она имеет гораздо более высокую промежуточность, чем все остальные, потому что каждый кратчайший путь, соединяющий узлы в двух сообществах, должен проходить через нее

Алгоритм Гирвана–Ньюмана итеративно выявляет и удаляет связь с наибольшей промежуточностью, что приводит к поступательному разложению сети на разъединенные части.

Мы начинаем с вычисления промежуточности для всех связей. Затем каждая итерация алгоритма состоит из двух шагов.

1. Удалить связь с наибольшей промежуточностью; в случае наличия нескольких кандидатов одна из них выбирается случайным образом.
2. Пересчитать расстояние между оставшимися связями.

Процедура заканчивается, когда все связи удалены и узлы изолированы.

На каждой итерации необходимо пересчитывать расстояние между всеми связями. Это имеет решающее значение для получения содержательных результатов, но делает указанный метод очень медленным. В сетях со структурой в форме сильных сообществ, которые быстро распадаются на разьединенные сообщества, шаг пересчета необходимо выполнять только внутри связной компоненты, включая последнюю удаленную связь, так как промежуточность всех остальных связей остается прежней. Этот факт может значительно сокращать вычислительную нагрузку пересчета промежуточности.

Алгоритм Гирвана–Ньюмана обеспечивает набор из N разделов, от одного, состоящего из одного кластера, включающего всю сеть, до того, где каждый узел является узлом-одиночкой (синглетоном) и образует свое собственное сообщество. В каждом разделе кластеры соответствуют связным компонентам сети. Всякий раз, когда удаление связи разбивает одну сетевую компоненту, число кластеров увеличивается на единицу. Все разделы являются иерархическими, потому что удаление связей разбивает кластеры на более мелкие части, которые в свою очередь разбиваются на еще меньшие части и т. д. Это пример дивизивной иерархической кластеризации, противоположной агломеративной иерархической кластеризации (раздел 6.2.2). Что касается агломеративной иерархической кластеризации, то полный набор разделов, предоставляемых этим методом, может быть представлен дендрограммой. На рис. 6.9 мы показываем разделы, обнаруженные методом Гирвана–Ньюмана в сети клубов карате «Закари», которая представляет социальные взаимодействия между членами клуба карате. Она является классическим эталоном для сравнения процедур обнаружения сообществ, который мы описываем в разделе 6.4. Сеть имеет естественное деление на два сообщества, обозначенных разными цветами меток узлов в нижней части дендрограммы. Найденное алгоритмом деление на два кластера (соответствующее черному горизонтальному разрезу) совпадает с естественным делением сети, за исключением узлов 2 и 8, которые классифицированы неправильно.

В библиотеке NetworkX есть функция для алгоритма Гирвана–Ньюмана:

```
# возвращает список иерархических разделов  
partitions = nx.community.girvan_newman(G)ë
```

Поскольку этот метод довольно медленный, он непрактичен для крупных сетей, насчитывающих, к примеру, более 10 000 узлов. Его узким местом является пересчет промежуточности связей. Для решения этой проблемы были предложены более быстрые варианты. Например, вместо точного вычисления промежуточности с использованием всех возможных пар узлов для определения числа кратчайших путей, пересекающих связь, можно получать приближенные значения баллов промежуточности, используя только выборку пар узлов. Имеет значение именно ранжирование связей, а не их точные значения промежуточности. Ученые также предложили альтернативные меры для выявления мостов, которые обходятся не так дорого при вычислении, как промежуточность.

Другой существенный недостаток указанного алгоритма является общим для всех технических приемов иерархической кластеризации (раздел 6.2.2): поскольку разделов сети столько же, сколько и узлов, какие из них являются содержательными, если таковые имеются? В целях ответа на этот вопрос в следующем разделе книги мы введем меру, выражающую качество раздела сети. Такую меру можно использовать для отбора наилучшего раздела из дендрограммы.

6.3.2. Оптимизация модулярности

Как понять, насколько «хорошим» является раздел? Естественный подход состоит в измерении того, насколько подсети раздела подобны сообществу. Например, мы могли бы вычислить плотность внутренних связей каждой подсети и посмотреть, достаточно ли она высока. Но такая стратегия может приводить к неверным результатам. Возьмем, например, случайную сеть (раздел 5.1): мы не ожидаем найти в ней сообщества, потому что узлы соединены друг с другом случайно, поэтому нет никакой группы узлов, которые предпочитают связываться друг с другом, а не связываться с узлами за пределами данной группы. Это относительно устоявшийся принцип обнаружения сообществ: *в случайных сетях нет сообществ!* Такое утверждение не зависит от плотности связей. Следовательно, случайная подсеть не создает хорошего сообщества независимо от плотности ее внутренних связей. Нам нужен более подходящий способ измерения качества раздела сообщества.

Модулярность раздела оценивает сообщества не в абсолютном выражении, а относительно *случайного базового уровня*. Это достигается за счет исключения внутренних связей, которые могут быть отнесены к рандомизированной версии изначальной сети. Проще говоря, модулярность – это разница между числом связей, внутренних для всех кластеров, и значением этого числа, ожидаемым в рандомизированной сети. Базовым уровнем, принятым в этом определении, яв-

ляется обсуждаемая в разделе 5.3 рандомизация с сохранением степени – сети с одинаковым числом узлов и где каждый узел сохраняет степень, которую он имеет в изначальной сети. Если изначальная сеть является случайным графом, то она будет иметь похожие признаки, что и ее рандомизации, и ее модулярность будет низкой. В частности, если число внутренних связей каждого кластера близко к его ожидаемому значению в случайных версиях сети, то структура сети в форме сообществ совместима с такой структурой случайной сети с одинаковой степенной последовательностью, и модулярность является низкой. С другой стороны, если число связей внутри кластеров намного больше, чем ожидаемое случайное значение, то маловероятно, что такая концентрация внутренних связей является результатом случайного процесса, и модулярность может достигать высоких значений (рис. 6.11). Во вставке 6.1 представлено формальное определение модулярности в ненаправленных невзвешенных сетях. Имея деление такой сети на множество сообществ, значение модулярности (уравнение (6.4)) может быть рассчитано с помощью следующей ниже функции библиотеки NetworkX:

```
# возвращает модулярность входного раздела  
modularity = nx.community.quality.modularity(G, partition)
```

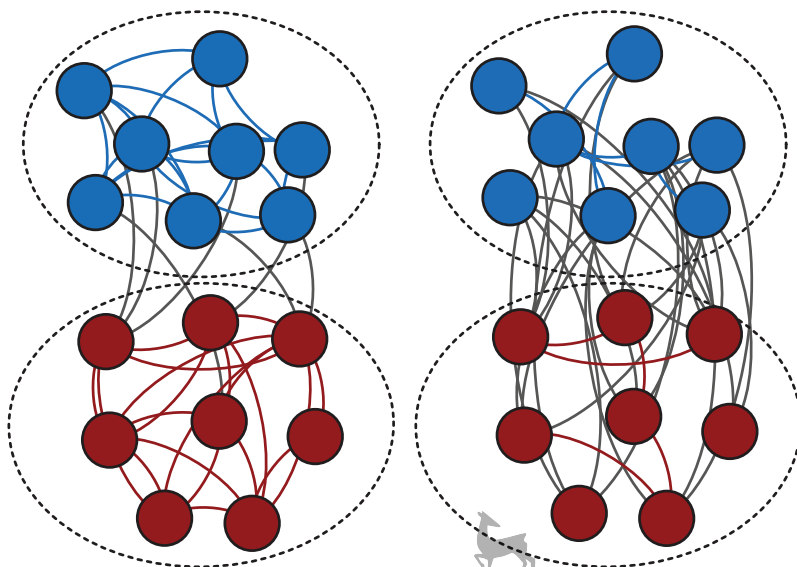


Рис. 6.11 Модулярность сети. Сеть слева имеет видимую структуру в форме сообществ с двумя кластерами, узлы которых выделены синим и красным цветом, и, следовательно, высокую модулярность. На рисунке справа показана рандомизация сети с сохранением степени. В рандомизированной сети меньше внутренних (синих и красных) связей, по сравнению с изначальной сетью, и больше (серых) связей между подсетями: процесс рандомизации разрушил структуру сети в форме сообществ. Таким образом, модулярность одного и того же раздела меньше. Воспроизведено из Фортунато и Хрика (2016) с разрешения издательства Elsevier

Вставка 6.1

Модулярность

Модулярность раздела в ненаправленной невзвешенной сети определяется как:

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right), \quad (6.4)$$

где суммирование выполняется по всем кластерам раздела, L_C – это число внутренних связей в кластере C , k_C – суммарная степень узлов в C (уравнение (6.2)), а L – число связей в сети.

Каждый элемент суммы представляет собой разницу между числом внутренних связей в кластере C и его ожидаемым числом в сетевых рандомизациях с сохранением степени. Рассчитывая ожидаемое число, следует учитывать, что случайные связи формируются путем сочетания пар заглушек, выбираемых случайным образом (раздел 5.3). Суммарное число заглушек, прикрепляемых к узлам в C , равно суммарной степени кластера, которая в каждой рандомизации равна k_C , потому что каждый узел сохраняет свою степень. Вероятность отбора одной из этих заглушек наугад равна $k_C/2L$, потому что это суммарное число заглушек в сети (каждая связь порождает две заглушки). Для того чтобы случайная связь соединила два узла в одном и том же кластере C , две заглушки должны отбираться из C . В близком приближении вероятность случайного выбора пары заглушек из C является произведением вероятностей отбора каждой из них в отдельности: $\frac{k_C}{2L} \cdot \frac{k_C}{2L} = \frac{k_C^2}{4L^2}$. (Этот сценарий аналогичен расчету вероятности получения орлов дважды в двух независимых подбрасываниях справедливой монеты, как $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$.) Наконец, поскольку всего существует L связей и каждая из них соединяет два узла в C с вероятностью $k_C^2/4L^2$, ожидаемое число внутренних связей равно $L \cdot k_C^2/4L^2 = k_C^2/4L$.

Деление любой сети в одном единственном кластере дает $Q = 0$. Это обусловлено тем, что сумма в уравнении (6.4) сводится к одному единственному члену, $L_C = L$, и $k_C = 2L$ является суммарной степенью сети. Далее, $Q < 1$ для любого раздела любой сети, потому что он не может быть больше $(\sum_C L_C)/L$, что не превышает единицы, когда все связи являются внутренними. Модулярность может принимать отрицательные значения. Рассмотрим разбиение на N синглетонов: первый член в каждом слагаемом равен нулю, потому что нет никаких связей, соединяющих узел с самим собой, отсюда Q является суммой отрицательных чисел. Для большинства сетей модулярность имеет нетривиальный максимум: $0 < Q_{\max} < 1$.

Указанное определение может быть распространено непосредственно на разделы в направленных и взвешенных сетях.

Выражение модулярности для разделов направленных сетей таково:

$$Q_d = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^{in} k_C^{out}}{L} \right), \quad (6.5)$$

где L_C – суммарное число направленных связей внутри кластера C и k_C^{in} , k_C^{out} – соответственно суммарная степень-на-входе и степень-на-выходе узлов в C .

Для взвешенных сетей мы имеем

$$Q_w = \frac{1}{W} \sum_C \left(W_C - \frac{s_C^2}{4W} \right), \quad (6.6)$$

где W – это суммарный вес связей сети, W_C – суммарный вес внутренних связей в C и s_C – суммарная сила узлов в C (т. е. сумма сил узлов в C ; уравнение (1.8)).

Для сетей, связи которых являются как направленными, так и взвешенными, мы имеем

$$Q_{dw} = \frac{1}{W} \sum_C \left(W_C - \frac{s_C^{in} s_C^{out}}{W} \right), \quad (6.7)$$

где s_C^{in} , s_C^{out} – это соответственно суммарная сила-на-входе и сила-на-выходе узлов в C .

Модулярность изначально была введена для ранжирования разделов, предоставляемых алгоритмом Гирвана–Ньюмана, чтобы иметь возможность выбирать наилучший. Например, наибольшее значение модулярности в дендрограмме на рис. 6.9 получено путем деления сети клубов карате «Закари» на пять кластеров, соответствующих пунктирному красному разрезу и обозначенных цветами вертикальных линий. Но поскольку модулярность измеряет качество любого раздела, мы не ограничиваемся его использованием в сочетании с другими техническими приемами. Нам лишь нужно отыскать раздел с максимально возможной модулярностью. В этом заключается идея *оптимизации модулярности*, лежащая в основе большого класса алгоритмов кластеризации сетей. Вспомните из раздела 6.1.3, что число возможных разделов сети огромно, поэтому невозможно провести полную разведку пространства разделов даже для малых сетей. Хорошие алгоритмы обычно ограничивают свой поиск малым подмножеством разделов.

Простым методом максимизации модулярности является агрегативный алгоритм, который начинается с раздела, где каждый узел является собственным кластером, затем итеративно сливает воедино пару кластеров, тем самым порождая наибольшее увеличение в модулярности. Данный метод разведывает представленную дендрограммой иерархию разделов. Первоначальное деление на синглтоны имеет отрицательную модулярность, затем модулярность неуклонно увеличивается до тех пор, пока не будет достигнут положительный пик, и, наконец, она снижается до тех пор, пока не достигнет нуля, когда все узлы находятся в одном и том же сообществе. Раздел, соответствующий пиковому значению, является наилучшим решением, найденным алгоритмом. Этот метод является жадным, поскольку на каждом шаге он пытается максимизировать модулярность. В силу этого он, скорее всего, будет застревать на решениях с субоптимальной модулярностью. Указанный алгоритм также тяготеет к генерированию несбалансированных разделов, причем некоторые кластеры намного больше других. Такие несбалансированные разделы часто нереалистичны и значительно замедляют метод. Простые модификации алгоритма, такие как слияние групп сопоставимого размера или более двух групп одновременно, доказали свою эффективность в решении этой проблемы.

В библиотеке NetworkX есть функция для быстрой версии жадной оптимизации модулярности:

```
# возвращает раздел с максимальной модулярностью
partition = nx.community.greedy_modularity_communities(G)
```

Наиболее популярным методом оптимизации модулярности является *алгоритм Лувена*. Это еще одна агрегативная процедура, в которой сообщества итеративно превращаются в суперузлы, как схематично показано на рис. 6.12.

Указанный алгоритм снова начинается с деления на синглтоны. Каждая итерация состоит из двух шагов.

1. Прокручивание узлов в цикле: каждый узел помещается в сообщество соседа, которое дает наибольшее увеличение модулярности ΔQ по отношению к текущему разделу. Все узлы посещаются повторно снова и снова до тех пор, пока больше не станет возможным увеличивать Q , перемещая узел в другое сообщество.
2. Сеть преобразовывается во взвешенную суперсеть, где каждое сообщество в разделе из шага 1 заменяется суперузлом, связи между суперузлами имеют вес, соответствующий числу связей, соединяющих узлы в соответствующих группах, а свя-

зи, соединяющие узлы в одном и том же сообществе, представлены в виде самонаправленного цикла из соответствующего суперузла к самому себе с весом, равным удвоенному числу внутренних связей.

Поскольку мы в конечном счете заботимся о кластеризации фактических узлов, а не суперузлов, модулярность всегда вычисляется по отношению к изначальной сети. Процедура останавливается, когда никакая дальнейшая группировка кластеров в текущем разделе не увеличивает модулярность, и возвращает раздел с наибольшей модулярностью.

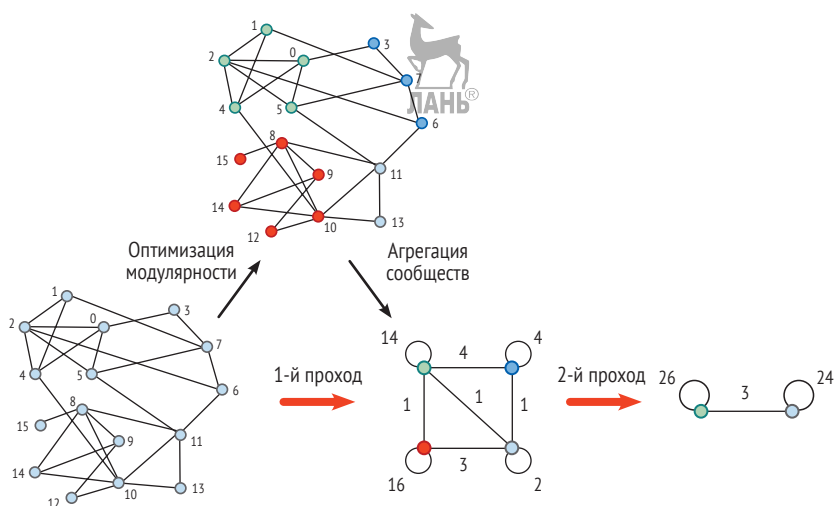


Рис. 6.12

Алгоритм Лувена. Мы показываем две итерации метода, начиная с графика слева. Каждая итерация состоит из двух шагов: сначала каждый узел назначается соседствующему сообществу, порождая наибольший (положительный) прирост модулярности до тех пор, пока модулярность невозможно увеличивать дальше. Далее, сеть преобразовывается в меньший взвешенный граф путем превращения кластеров в суперузлы, каждое множество связей между двумя отдельными сообществами в единую взвешенную связь между соответствующими суперузлами, а внутренние связи в каждом сообществе – в самонаправленный цикл соответствующего суперузла. Веса на суперсвязях позволяют быстрее вычислять изменение модулярности для разделов изначальной сети, соответствующих слиянию групп, представленных суперузлами. Рисунок перепечатан с разрешения Блонделя и соавт. (2008), авторское право (2008) принадлежит издательству Института физики

Алгоритм Лувена является жадным, поэтому ему обычно не удается найти раздел, который плотно приближается к оптимальной модулярности. Более того, результат зависит от порядка посещения узлов. С положительной стороны данный алгоритм является очень быстрым, потому что после первой итерации поочередные преобразования очень быстро сжимают сеть, и обычно генерируется всего несколько разделов. Меньшие сети позволяют выполнять быстрые вычисле-

ния. Поэтому указанный метод широко используется в практических приложениях; например, кластеры, показанные на рис. 0.2(b), были найдены в таком ключе. Алгоритм Лувена может использоваться для обнаружения сообществ в крупных сетях со многими миллионами узлов и связей.

На момент написания этой книги в библиотеке NetworkX еще не было имплементации алгоритма Лувена. Один из них доступен путем импорта модуля `community`¹:

```
# скачать модуль community по адресу
# github.com/taynaud/python-louvain
import community
# возвращает раздел с наибольшей модулярностью
partition_dict = community.best_partition(G)
```

Хотя методы модулярности и популярны, эта мера имеет важные ограничения, которые подрывают ее полезность в практических приложениях. Например, максимальная модулярность, как правило, больше в более крупных сетях. Поэтому ее нельзя использовать для сравнения качества разделов в разных системах. Кроме того, максимальная модулярность разделов случайных сетей может достигать довольно больших значений. Это, возможно, покажется удивительным, поскольку указанная мера определяется относительно случайного базового уровня, поэтому, если сеть сама по себе является случайной, мы ожидаем малых отклонений от базового уровня. Но мера просто вычитает ожидаемое число внутренних связей по каждому сообществу из фактического числа [уравнение (6.4)]; она не учитывает случайные колебания вокруг ожидаемого значения, которые могут увеличивать модулярность. Наконец, максимальная модулярность не обязательно соответствует наилучшему разделу. Это обусловлено тем, что указанная мера имеет внутренний *лимит разрешающей способности*, который не дает ей обнаруживать малые сообщества.

В частности, лимит разрешающей способности зависит от суммарного числа связей в сети: сообщества, степень которых меньше чем $\sqrt{2L}$, практически невидимы для метода и, возможно, будут сливаться воедино с другими кластерами.

На рис. 6.13 показано экстремальное следствие этой проблемы. Поскольку клики являются наиболее когезивными подсетями, которые можно получить, поскольку присутствуют все внутренние связи, сеть на рисунке имеет естественное деление на 16 сообществ, соответствующих кликам. Однако существуют разделы с более крупной модуляр-

¹ См. python-louvain.readthedocs.io.

ностью, такие как восемь кластеров, обозначенных пунктирными контурами на рисунке. Обойти эту проблему можно путем регулировки разрешающей способности метода, введя параметр в определение модулярности [уравнение (6.4)], значение которого определяет масштаб обнаруженных сообществ, от очень малых до очень крупных размеров. Эта стратегия, именуемая *оптимизацией модулярности с несколькими разрешающими способностями*, требует больших вычислительных затрат, поскольку модулярность должна оптимизироваться для нескольких вариантов параметра. Кроме того, необходим критерий для принятия решения о том, какое значение параметра разрешающей способности подходит для данной сети больше всего. Несмотря на эти недостатки, оптимизация модулярности с несколькими разрешающими способностями находит широкое применение в приложениях.

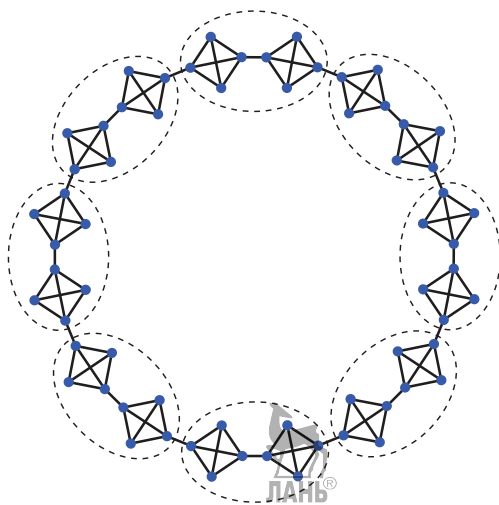


Рис. 6.13 Лимит разрешающей способности оптимизации модулярности. Сеть состоит из групп по четыре узла, образующих кольцевую структуру, каждая из которых соединена с двумя другими одиночными связями. Мы ожидаем, что модулярность достигнет максимума для раздела, сообщества которого являются кликами. Этот раздел имеет модулярность $Q \approx 0.795$. Однако оказывается, что раздел, объединяющий пары клик (обозначенный пунктирными контурами), имеет более высокую модулярность. Перепечатано из Фортунато и Хрик (2016) с разрешения издательства Elsevier

6.3.3. Распространение меток

Алгоритм распространения меток – это простой и быстрый метод обнаружения сообществ, основанный на идее, что соседи обычно принадлежат к одному и тому же сообществу. Это означает, что большинство связей являются внутренними, создавая когезивные и сепарированные сообщества, как описано в разделе 6.1.2. На каждом шаге алгоритм проверяет каждый узел и назначает его сообществу большинства своих соседей. Указанная процедура в конечном ито-

ге сходится к стабильному разделу, где каждый узел имеет такое же членство в сообществе, как и у большинства его соседей.

Метод распространения меток начинается с деления на синглтоны. Каждому узлу дается своя метка. Процедура состоит из двух итерационных шагов.

1. Выполняется виток по всем узлам в случайном порядке: каждый узел берет метку, общую для большинства его соседей. Если уникального большинства нет, то одна из меток большинства подбирается случайно.
2. Если каждый узел имеет метку большинства от своих соседей (стационарное состояние), то нужно остановиться. В противном случае повторить шаг 1.

Сообщества определяются как группы узлов, имеющих одинаковые метки в стационарном состоянии.

Во время этого процесса метки распространяются по сети: большинство меток исчезает, другие же доминируют. Поскольку деление сети изменяется во время каждого витка, для достижения стационарного состояния требуется несколько витков. Однако алгоритм обычно сходится после малого числа итераций, практически не зависящего от размера сети.

В окончательном разделе каждый узел имеет соседей в своем собственном сообществе больше, чем в любом другом. Таким образом, каждый кластер является сильным сообществом в соответствии с менее строгим определением, приведенным в разделе 6.1.2. Проблема заключается в том, что указанный алгоритм не обеспечивает уникального решения; результат зависит от порядка посещения узлов в каждом витке, который устанавливается случайным, чтобы избежать систематического смещения, но равным образом меняется в зависимости от последовательности случайных чисел, используемых при расчете. Разные разделы также являются результатом наличия многочисленных кандидатов, возникающих по ходу, которые могут регулироваться по-разному, опять же в зависимости от последовательности случайных чисел. Несмотря на эти нестабильности, разделы, обнаруживаемые путем распространения меток в реально существующих сетях, как правило, похожи друг на друга. Для получения более надежных результатов можно комбинировать решения, получаемые из разных прогонов процедуры.

Сила данного метода заключается в том, что ему не нужна никакая информация о числе и размере сообществ. У него также нет никаких параметров. Этот метод прост в имплементировании и очень быстр: в таком ключе могут делиться сети с миллионами узлов и связей. Если метки сообщества для некоторых узлов известны, то их можно использовать в качестве затравочных в первоначальном разделе.



В качестве примера, этот метод использовался для назначения цветов узлам в диффузионной сети Twitter, показанной на рис. 0.3.

В библиотеке NetworkX есть функция, выполняющая алгоритм распределения меток:

```
partition = nx.community.asyn_lpa_communities(G)
```

6.3.4. Стохастическое блочное моделирование

Предположим, у вас есть сеть и вы знаете, что она была сгенерирована какой-то моделью, заданной набором параметров. Как только значения параметров фиксированы, модель генерирует класс сетей. Если параметры неизвестны, то мы можем осведомиться, для каких значений модельные сети похожи на наш граф больше всего. Это аналогично ситуации, когда мы помещаем, скажем, прямую линию в набор точек данных и выводим параметр наклона. Например, если мы знаем, что наша сеть является случайным графом, то мы можем спросить, какое значение вероятности связи продуцирует графы, структура которых больше всего похожа на структуру нашей сети.

Если мы заинтересованы в раскрытии структуры сети в форме сообществ, то мы можем подумать о моделях, которые генерируют сети с сообществами. В этом ключе после того как будет найдена наиболее подходящая модель, внедренные в модель сообщества будут наилучшей аппроксимацией кластеров, которые мы хотим обнаружить. Наиболее широко распространенной моделью сетей с сообществами является *стохастическая блочная модель* (stochastic block model, SBM). Ее базовая идея заключается в том, что узлы поделены на группы, и вероятность того, что два узла соединены, определяется группами, к которым они принадлежат.

Формально пусть N узлов сети поделены на q групп $1, \dots, q$. Узел i находится в группе g_i . Вероятность того, что узлы i и j соединены, зависит исключительно от их членств в группах: $P(i \leftrightarrow j) = P_{g_i g_j}$. Следовательно, для любой пары групп g_1 и g_2 вероятность соединения между любым узлом в g_1 и любым узлом в g_2 одинакова. Вероятности образуют матрицу $q \times q$, именуемую *стохастической блочной матрицей* (рис. 6.14). Для направленных графов указанная матрица в общем случае является асимметричной. Здесь мы сосредоточимся на ненаправленных сетях, и поэтому эта матрица является симметричной. Диагональные элементы P_{gg} ($g = 1, \dots, q$) стохастической блочной матрицы представляют собой вероятности того, что узлы в блоке g являются соседями, тогда как внедиагональные элементы дают вероятности связи между разными блоками.

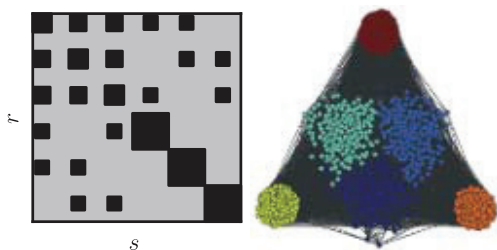


Рис. 6.14 Стохастическая блочная модель. (Слева) Стохастическая блочная матрица указанной модели с шестью блоками. Размер квадратов пропорционален значениям вероятностей связи между соответствующими группами. (Справа) Реализация сети с 1000 узлами, произведенными с использованием вероятностей связи в матрице. Группы идентифицируются по цветам. Рисунок перепечатан из Пейшото (2012) с разрешения Американского физического общества

Если вероятность соединения узлов внутри одной и той же группы выше, чем вероятность связей между узлами в разных группах, то указанная модель дает сети с когезивными и сепарированными сообществами. Но модель может также генерировать другие различные типы групповой структуры.

Если $\forall r, s = 1, \dots, q$ такие что $r \neq s$, мы имеем $p_{rr} > p_{rs}$, тогда мы восстанавливаем структуру в форме сообществ. Для $p_{rr} < p_{rs}$ мы имеем дисассортативную структуру, так как связи между блоками являются более вероятными, чем внутри них. В частном случае $p_{rr} = 0 \forall r$ мы получаем *многораздельные сети*, так как существуют связи только между блоками. Если $q = 2$ и $p_{11} \gg p_{12} \gg p_{22}$, то мы имеем *структуру ядро-периферия*: узлы в первом блоке (ядро) относительно хорошо соединены между собой, а также с периферийным множеством узлов, которые очень мало взаимодействуют между собой. Наконец, если все вероятности равны ($\forall r, s : p_{rs} = p$), то мы восстанавливаем классическую случайную сеть: любые два узла имеют одинаковую вероятность быть соединенными, и, следовательно, групповая структура отсутствует.

Определив модель, мы должны подогнать ее под нашу сеть. Стандартная процедура заключается в максимизировании правдоподобия, понимаемого как вероятная возможность того, что для данного раздела сети стохастическая блочная модель воспроизводит размещение связей между узлами. Такое правдоподобие может быть вычислено аналитически; оно говорит о том, насколько хорошо стохастическая блочная модель с данным разделом имитирует нашу сеть. Последний шаг состоит в отыскании раздела, который дает наибольшее значение указанного правдоподобия. Вставка 6.2 выражает это правдоподобие и представляет жадный алгоритм для отыскания раздела, который ее

максимизирует. Как и все жадные технические приемы, этот метод обеспечивает субоптимальное решение. В целях улучшения результата, помогает выполнение алгоритма несколько раз с разными случайными первоначальными условиями и отбор раздела с наибольшим правдоподобием во всех прогонах.

Вставка 6.2

Подгонка стохастической блочной модели под сеть

Стандартная стохастическая блочная модель плохо описывает групповую структуру большинства реально существующих сетей, поскольку она игнорирует степень гетерогенности. Поэтому мы рассматриваем *откорректированную по степени стохастическую блочную модель* (degree-corrected stochastic block model, DCSBM), в которой степени узлов соответствуют фактическим степеням в сети. Вероятность того, что сеть G воспроизводится откорректированной по степени стохастической блочной моделью на основе заданного деления g узлов сети G на q групп, выражается *логарифмическим правдоподобием*:

$$\mathcal{L}(G|g) = \sum_{r,s=1}^q L_{rs} \log \left(\frac{L_{rs}}{k_r k_s} \right), \quad (6.8)$$

где L_{rs} – это число связей, идущих из группы r в группу s , k_r (k_s) – сумма степеней узлов в r (s), и суммирование выполняется по всем парам групп в q (включая, когда $r = s$).

Максимизация правдоподобия в уравнении (6.8) для деления на q групп может достигаться посредством простого жадного технического приема. Отправной точкой является случайное деление на q кластеров. Каждая итерация алгоритма состоит из двух шагов.

1. Многократно перемещать узел из одной группы в другую, на каждом шаге отбирая перемещение, которое будет увеличивать правдоподобие больше всего (или по меньшей мере ее уменьшит, если увеличение невозможно), при том ограничении, что каждый узел можно перемещать только один раз.
2. Когда все узлы будут перемещены, проинспектировать разделы, через которые система прошла от начала до конца процедуры на шаге 1, и отобрать раздел с наибольшим правдоподобием.

Алгоритм останавливается тогда, когда правдоподобие для двух поочередных итераций одинаково (т. е. его невозможно увеличивать дальше).

Наиболее важным пределом этого подхода является необходимость заранее указывать число групп, которое обычно неизвестно для реально существующих сетей. Это обусловлено тем, что, как выясняется, прямая максимизация правдоподобия по всему набору возможных делений приводит к тривиальному делению на синглтоны. К счастью, есть способы оценивания числа кластеров либо заранее, либо посредством более тонких стохастических блочных моделей.

6.4. Оценивание методов

Как узнать, что метод обнаружения сообществ «хорош»? Как определить, какой из двух методов лучше другого? Такие вопросы являются неоднозначными, потому что, как правило, нет никаких достоверных эмпирических данных о правильном способе деления сети. Распространенный подход к оцениванию алгоритма состоит в проверке его способности отыскивать сообщества в *эталонных графах* (т. е. сетях, о которых известно, что они имеют «естественную» структуру в форме сообществ). Существует два класса эталонов: (i) искусственные сети, созданные с помощью некоторой модели, и (ii) реально существующие сети, в которых сообщества подсказываются историей системы либо атрибутами узлов.

6.4.1. Искусственные эталоны

Стохастические блочные модели (раздел 6.3.4) часто используются для создания искусственных эталонных графов.

В библиотеке NetworkX для этого есть соответствующая функция:

```
# сеть с сообществами с размерами в списке S
# и стохастическая блочная матрица P
G = nx.generators.stochastic_block_model(S, P)
```

Особой версией стохастической блочной модели является *модель на основе внедрения разделов*. Это упрощенная версия изначальной стохастической блочной модели, имеющая только две вероятности связи: вероятность того, что два узла соединены в одном сообществе, и вероятность того, что два узла соединены в разных сообществах.

В разделе 6.3.4 мы увидели, что стандартная стохастическая блочная модель с q группами характеризуется стохастической блочной матрицей $q \times q$, элемент p_{rs} которой выражает вероятность наличия связи между любым узлом в группе r и любым узлом в группе s . В модели на основе внедрения разделов, $p_{rs} = p_{int}$ для $r = s$ и $p_{rs} = p_{ext}$ для $r \neq s$. Если $p_{int} > p_{ext}$, то имеется более высокий шанс того, что два узла соединены, если они находятся в одной группе, чем если они находятся в разных группах, из чего вытекает, что группы являются сообществами. Указанная модель также исходит из допущения, что все сообщества имеют одинаковый размер N/q . Имея значения p_{int} , p_{ext} и q , мы можем вычислить ожидаемые внутренние и внешние степени узла:



$\langle k^{int} \rangle = p_{int} \left(\frac{N}{q} - 1 \right)$ и $\langle k^{ext} \rangle = p_{ext} \frac{N}{q} (q - 1)$, так как каждый из других $\frac{N}{q} - 1$ узлов в группе любого узла i имеют равную вероятность p_{int} стать соседом узла i , и каждый узел в других группах имеет равную вероятность p_{ext} стать соседом узла i . Ожидаемая (суммарная) степень равна $\langle k \rangle = \langle k^{int} \rangle + \langle k^{ext} \rangle = p_{int} \left(\frac{N}{q} - 1 \right) + p_{ext} \frac{N}{q} (q - 1)$.

В библиотеке NetworkX есть функция, которая генерирует сети в соответствии с моделью на основе внедрения разделов:

```
# сеть с q сообществами по nc узлов каждое
# и вероятностями связи p_int и p_ext
G = nx.generators.planted_partition_graph(q, nc, p_int, p_ext)
```

Специфическая имплементация модели на основе внедрения разделов, в которой размер сети, степень узлов, а также число и размер сообществ устанавливаются равными конкретным значениям, называется *эталон* GN и долгое время использовалась научным сообществом в качестве стандартного инструмента подтверждения.

В целях вывода эталона GN мы задаем $N = 128$, $q = 4$, и $\langle k \rangle = 16$. Из этого вытекает, что $31p_{int} + 96p_{ext} = 16$, вследствие чего p_{int} и p_{ext} не являются независимыми параметрами. Как только мы фиксируем p_{int} , значение p_{ext} определяется этим отношением. Зная p_{int} и p_{ext} , сеть строится с помощью процедуры, аналогичной той, которая принята для случайных графов Эрдеша–Реньи (раздел 5.1): мы перебираем все пары узлов в цикле и соединяем каждый с вероятностью p_{int} либо p_{ext} в зависимости от того, находятся узлы в одном сообществе или нет.

Чем выше внешняя степень и чем ниже внутренняя степень, тем труднее обнаруживать сообщества. На рис. 6.15 показаны три сети возрастающей сложности, построенные на базе эталона GN.

При низких значениях $\langle k^{ext} \rangle$ сообщества хорошо сепарированы, и большинство алгоритмов обнаруживает их без проблем. Результативность снижается с увеличением $\langle k^{ext} \rangle$: растущее число узлов в итоге имеет сопоставимое число соседей в разных груп-

пах и может классифицироваться неправильно. До тех пор, пока $p_{int} > p_{ext}$, хороший алгоритм теоретически должен быть способен распознавать сообщества. При $p_{int} = p_{ext} = 16/127$, $\langle k^{ext} \rangle \approx 12$ сообщества должны тогда обнаруживаться для $\langle k^{ext} \rangle$ ниже этого значения. Вместо этого оказывается, что порог обнаруживаемости находится намного ниже, около девятки, из-за случайных колебаний в размещении связей. Для $9 \leq \langle k^{ext} \rangle \leq 12$ эти колебания делают сети практически неотличимыми от случайных графов.

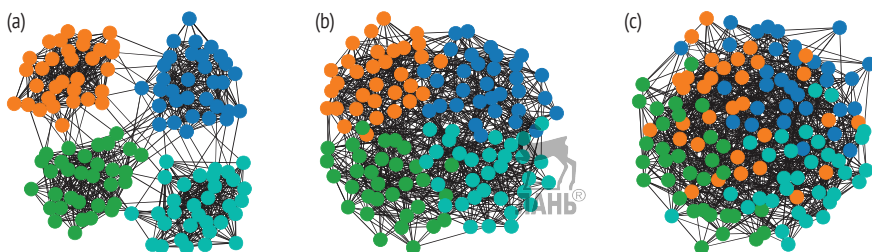


Рис. 6.15 Три сети, построенные на основе эталона GN, с ожидаемыми степенями (а) $\langle k^{ext} \rangle = 1$, $\langle k^{int} \rangle = 15$ (б) $\langle k^{ext} \rangle = 5$, $\langle k^{int} \rangle = 11$ и (с) $\langle k^{ext} \rangle = \langle k^{int} \rangle = 8$. Четыре группы в сети (с) едва различимы; в этом случае методы обнаружения сообществ отказывают в назначении многочисленных узлов правильным группам

Несмотря на свою полезность, эталон GN не является хорошим косвенным индикатором для реально существующих сетей со структурой в форме сообществ. Одна из его проблем заключается в том, что все узлы имеют приблизительно одинаковую степень, тогда как степенное распределение реально существующих сетей обычно является сильно гетерогенным (раздел 3.2). Еще одним лимитом является то, что сообщества в реально существующих сетях обычно имеют совершенно разные размеры, тогда как эталоны, основанные на модели на основе внедрения разделов, порождают группы одинакового размера. Более продвинутый эталон LFR позволяет продуцировать сети с утяжеленными распределениями значений степени и размера сообществ, как показано на рис. 6.16. Эталон LFR теперь регулярно используется для оценивания алгоритмов обнаружения сообществ.

При оценивании методов обнаружения сообществ также важны отрицательные тесты. С этой целью можно использовать сети без структуры в форме сообществ. Случайные сети являются хорошими тому примерами; алгоритм, который обнаруживает сообщества в них, вряд ли будет надежным в приложениях. В этих случаях мы ожидаем получить содержательные деления в виде делений на синглтоны и делений, где вся сеть целиком представляет собой единый кластер. Любое другое деление будет сигнализировать о неспособности метода отличать фактические сообщества от подсетей с высокой концентрацией связей, генерируемых случайными колебаниями.

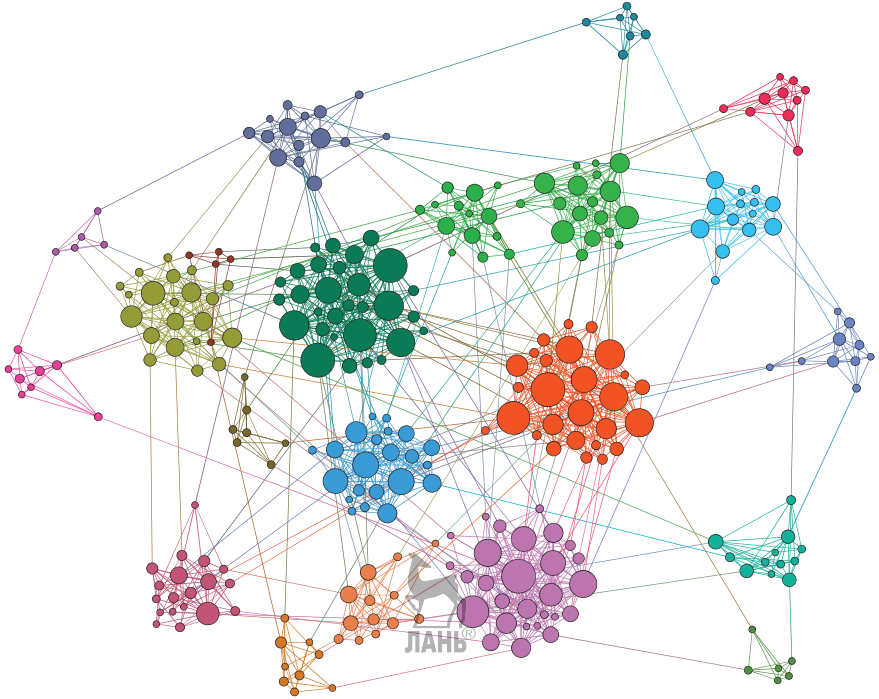


Рис. 6.16 Эталон LFR. Размер узла пропорционален степени. Значения степени узла и размера сообщества широко распределены, чтобы учесть гетерогенность, наблюдаемую в большинстве реально существующих сетей со структурой в форме сообществ

6.4.2. Реально существующие эталоны

Наиболее известным примером реально существующей сети с сообществами является сеть клубов карате «Закари», показанная на рис. 6.17. В нем 34 узла, члены клубов карате в Соединенных Штатах, за которыми велось наблюдение в течение трех лет. Связи соединяют индивидуумов, взаимодействующих за пределами деятельности клуба. В какой-то момент конфликт между инструктором и президентом клуба карате привел к разделению клуба на две отдельные группы, члены которых поддерживали соответственно инструктора и президента. Клубные группы имеют смысл, если основываться на сетевой структуре: большинство участников соединено с одним из двух центров, что свидетельствует об их тесной ассоциации с президентом либо инструктором. Надежный метод кластеризации должен быть способен распознавать двухчастное деление. На самом деле сеть клубов карате «Закари» не представляет какую-то сложную задачу для алгоритмов обнаружения сообществ, многие из которых способны классифицировать узлы правильно.

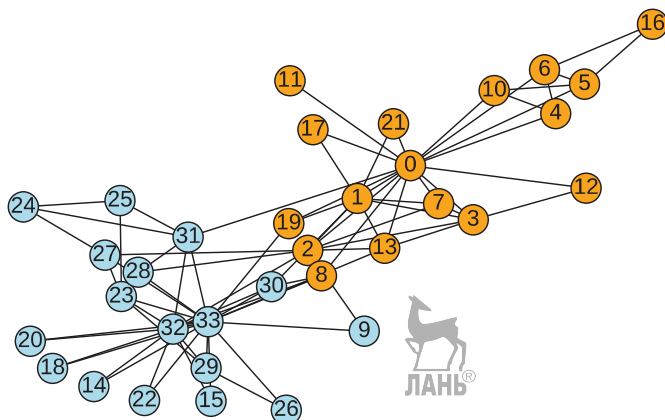


Рис. 6.17 Сеть клубов карате «Закари». Цвета обозначают последователей инструктора клуба (узел 0) и его президента (33), которые в итоге образовали две отдельные группы

В библиотеке NetworkX есть функция, которая возвращает сеть клубов карате «Закари»:

```
G = nx.karate_club_graph()
```

Существуют и многие другие сети, узлы которых могут классифицироваться на основе их атрибутов, предназначенные для тестирования алгоритмов обнаружения сообществ. Например, во многих социальных сетях есть группы, в которые пользователи могут присоединяться; в сетях цитирования статьи могут группироваться в соответствии с местами их публикации; интернет-маршрутизаторы могут классифицироваться по странам и т. д. Такие группы не всегда соответствуют сообществам, найденным методами кластеризации, что ставит вопрос о том, должны ли структурные кластеры соответствовать атрибутивно-ориентированным группам. Ответ зависит от сети и атрибутов.

Обнаружение сообществ в сетях представляет собой способ выявления скрытых атрибутов, которые определяют внешний вид графа. Поэтому, если узлы с одинаковыми или похожими атрибутами сильно друг с другом связаны, то атрибуты могут выявляться с помощью методов обнаружения сообществ. Если же атрибуты не играют роли в построении сети, то они остаются для методов кластеризации невидимыми.

6.4.3. Сходство между разделами

Последним компонентом, необходимым для процедуры подтверждения, является мера *сходства разделов*, показывающая, насколько

результат алгоритма похож на естественное деление сети, принятое в качестве эталона. Существует целый ряд мер сходства между разделами. Их можно классифицировать в соответствии с критериями, используемыми для оценивания сходства.

Популярной мерой является *доля правильно обнаруженных узлов*. Узел классифицируется правильно, если он и по меньшей мере половина других узлов в том же сообществе в обнаруженном разделе находится в том же сообществе в эталонном разделе. Обратите внимание, что если обнаруженный раздел содержит сообщества, полученные путем слияния двух или более групп эталонного раздела, то все узлы этих кластеров считаются классифицированными неправильно по этой мере. Затем число правильно классифицированных узлов делится на число узлов в сети, в результате давая долю от нуля до единицы. Проблема с этой мерой состоит в том, что рецепт для обозначения узлов как классифицированных правильно либо неправильно является несколько произвольным.

Более качественный подход заключается в оценивании сходства между двумя разделами путем вычисления с учетом одного из двух, дополнительного объема информации, необходимого для выведения другого. Это сводится к информации об узлах, которые необходимо переместить между кластерами для транзита из одного раздела в другой. Если разделы похожи, то для перехода из одного в другой требуется мало информации. Чем больше дополнительной информации требуется, тем меньше разделы похожи. Во вставке 6.3 представлена *нормализованная взаимная информация*, мера сходства, основанная на информации. Несмотря на ее широкое распространение, нормализованная взаимная информация имеет некоторые ограничения. Обнаруженные разделы с большим числом кластеров могут давать большие значения, даже если они не обязательно ближе к эталонному разделу. А это может давать неверное представление об относительной результативности алгоритмов. Можно использовать и другие меры сходства разделов, но любая мера имеет свои преимущества и недостатки.

Вставка 6.3

Нормализованная взаимная информация

В некоторых мерах сходства между разделами используются концепции, заимствованные из теории информации. Вероятность того, что случайно выбранный узел принадлежит кластеру x раздела X , задается формулой $P(x) = N_x/N$, где N_x – это размер кластера x . Вероятность того, что случайно выбранный узел принадлежит как кластеру x раздела X , так и кластеру y раздела Y , равно $P(x, y) = N_{xy}/N$, где N_{xy} – это число узлов, которые имеют общие кластеры x и y . Нормализованная взаимная информация о разделах X и Y определяется уравнением

$$\text{NMI}(X, Y) = \frac{2H(X) - 2H(X|Y)}{H(X) + H(Y)}, \quad (6.9)$$

где $H(X) = -\sum_x P(x) \log P(x)$ – это энтропия Шеннона раздела X , а $H(X|Y) = \sum_{xy} P(x,y) \log[P(y)/P(x,y)]$ – условная энтропия раздела X при наличии Y . Суммирование выполняется по всем кластерам x из раздела X и по всем парам кластеров x и y из X и Y . $NMI = 1$, если и только если разделы идентичны, тогда как ожидаемое значение равно нулю, если они независимы, как, например, при сравнении двух случайных разделов.

6.5. Резюме

Сообщества играют ключевую роль в структуре и функциях сетей. Они выявляют сходства между узлами, показывают принцип организации сети, позволяют нам раскрывать роль узлов как в их сообществе, так и в сети в целом, и влияют на динамику процессов, проходящих в сети. Именно по этой причине обнаружение сообществ является центральной задачей в сетевом анализе. Вот что мы узнали в этой главе.

1. Сообщества не являются четко определенными объектами. На высоком уровне они представляют собой когезивные и хорошо сепарированные подсети, поскольку внутри них много связей, а между ними их не так много. Многие алгоритмы кластеризации не требуют точного определения сообщества.
2. Число возможных делений сети на сообщества огромно даже для малого графа, поэтому мы не можем искать их все.
3. В случайных сетях нет сообществ. Их можно использовать для проверки способности алгоритмов обнаружения сообществ отличать сигнал от шума.
4. Процедуры деления сети выполняют поиск хорошо сепарированных подсетей. Они не обязательно являются когезивными, поэтому они могут соответствовать сообществам, а могут и не соответствовать. Несмотря на это ограничение и тот факт, что число кластеров должно указываться на входе в процедуру, инструменты деления сети бывают полезны для обнаружения сообществ.
5. Иерархическая кластеризация группирует узлы на основе их сходства. Она широко используется, но имеет ряд недостатков, и в первую очередь отсутствие критерия отбора содержательных разделов из полной иерархии (дендрограммы), предоставляемой процедурой.
6. Процедура устранения мостов состоит в стирании связей между сообществами, чтобы иметь возможность разъединять и выявлять последние. Как и другие подходы на основе иерархической кластеризации, устранение мостов имеет проблему невозможности ранжировать найденные иерархические разделы, если не предоставлен дополнительный критерий.

7. Процедура оптимизации модулярности выполняет поиск раздела с наибольшим баллом модулярности. Модулярность измеряет качество раздела путем сравнения числа внутренних связей с ожидаемыми в рандомизированных версиях сети. Чем больше балл, тем менее случайны и, следовательно, более содержательны кластеры. Метод Лувена представляет собой метод жадной оптимизации модулярности, который широко используется в приложениях благодаря своей скорости. Одним из ограничений оптимизации модулярности является то, что сети без групповой структуры могут иметь разделы с довольно крупной модулярностью. Еще одно состоит во том, что невозможно отыскивать малые сообщества.
8. Метод распространения меток назначает узлы сообществам таким образом, что у каждого узла больше соседей в его собственном сообществе, чем в любом другом.
9. Стохастические блочные модели генерируют сети с групповой структурой. Сообщества в сети могут быть восстановлены путем подгонки под них стохастической блочной модели. Это делается путем максимизации правдоподобия того, что сеть будет воспроизведена моделью. Этот подход должен учитывать число групп на входе в процедуру, но для его оценивания существуют подходящие процедуры.
10. Алгоритмы обнаружения сообществ могут оцениваться путем тестирования в отношении того, могут ли они восстанавливать известную структуру сообществ эталонных сетей или нет. Популярными искусственными эталонными сетями GN и LFR являются производными от специальных стохастических блочных моделей. Реально существующие сети с групповыми атрибутами бывают полезными или не полезными для тестирования в зависимости от того, являются или нет атрибуты узлов фактором в генезисе структуры в форме сообществ. Меры сходства между разделами используются для оценивания того, насколько близко сообщества, обнаруженные с помощью алгоритма, похожи на сообщества в эталоне.



6.6. Дальнейшее чтение

В отношении подробного изложения настоящей темы мы отсылаем к ряду обзорных статей (Портер и соавт., 2009; Фортунато, 2010; Фортунато и Хрик, 2016). Всестороннее введение в определения сообществ в анализе социальных сетей можно найти в книге Вассермана и Фауста (1994). Мы также рекомендуем прочитать статью Закари об анализе сети клубов карате (Zachary, 1977).

Сильные сообщества были представлены Луччо и Сами (1969). Радикки и соавт. (2004) смягчили концепцию сильного сообщества, дав определение слабым сообществам. Ху и соавт. (2008) предложили менее строгие понятия для сильных и слабых сообществ.

По теме деления сети мы отсылаем к книге Бишота и Сиарри (2013). Алгоритм Кернигана–Лина был первоначально предложен Керниганом и Лином (1970). Джейн и соавт. (1999) и Сюй и Вунш (2008) дают хорошее представление о кластеризации данных.

Алгоритм Гирвана–Ньюмана был представлен Гирваном и Ньюманом (2002). Модулярность была определена Ньюманом и Гирваном (2004), а метод жадной оптимизации модулярности был представлен Ньюманом (2004a). Быстрая версия жадного технического приема Ньюмана была предложена Клаузе и соавт. (2004). Метод Лувена был разработан Блонделем и соавт. (2008).

Гимера и соавт. (2004) обнаружили, что деления случайных графов могут достигать больших баллов модулярности. Лимит разрешающей способности для максимизации модулярности был выявлен Фортунато и Бартелеми (2007). Расширение модулярности на случай взвешенных сетей было предложено Ньюманом (2004b), тогда как Аренас и соавт. (2007) дополнительно расширили определение на сети, которые также являются направленными.

Оптимизация модулярности с несколькими решениями была инициирована Райхардтом и Борнхольдом (2006) и Аренасом и соавт. (2008). Рагхаван и соавт. (2007) разработали метод распространения меток. Стохастические блочные модели были представлены в основополагающих работах Файенберга и Вассермана (1981) и Холланда и соавт. (1983). Современные методы подгонки стохастических блочных моделей под сети и выведения сообществ были представлены Каррером и Ньюманом (2011) и Пейшото (2014).

Модель на основе внедрения разделов была идеей Кондона и Карпа (2001), а основанный на ней сравнительный эталон GN назван по имени авторов, Гирвана и Ньюмана (2002). Сравнительный эталон LFR был разработан Ланчинетти и соавт. (2008) и назван по имени трех авторов. Ланчинетти и Фортунато (2009) провели сравнительный анализ многих алгоритмов на эталоне LFR. Янг и Лесковец (2012) и Хрик и соавт. (2014) обнаружили разрыв между структурными сообществами и атрибутивно-ориентированными сообществами. Мейла (2007) дает хороший обзор мер сходства между разделами.

Доля правильно обнаруженных узлов и нормализованная взаимная информация были введены соответственно Гирваном и Ньюманом (2002) и Фредом и Джейн (2003).

Упражнения

6.1 Ознакомьтесь с учебным материалом главы 6 в репозитории книги на GitHub¹.

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

- 6.2 Сильные сообщества также являются слабыми сообществами, но в целом обратное неверно. Приведите пример слабого сообщества, которое не является сильным сообществом.
- 6.3 Определение слабого сообщества отражает наше наивное ожидание, что внутри сообщества должно быть больше связей, чем за его пределами. Однако для того, чтобы подсеть C была слабым сообществом, вовсе не обязательно, чтобы число внутренних связей L_C превышало число внешних связей k_C^{ext} . Каково фактическое условие? Приведите небольшой пример слабого сообщества C , такого что $L_C < k_C^{ext}$.
- 6.4 Любое деление сети с N узлами на $N - 1$ групп обязательно должно состоять из одной пары узлов и синглтонов. Сколько существует таких разделов?
- 6.5 Предположим, что сеть представляет собой гигантскую клику из N узлов, причем N является четным. Каково решение задачи рассечения графа пополам для этой сети? Каков размер разреза в результате двухчастного деления?
- 6.6 При рассечении графа пополам из минимизации размера разреза между двумя кластерами следует максимальное число связей внутри кластеров. Таким образом, деление сети может выглядеть эквивалентным обнаружению сообществ, когда мы имеем дело с двухчастным делением. Объясните, почему это не так, и приведите пример.
- 6.7 Найдите наилучшее рассечение пополам сети клубов карате «Закари», применив алгоритм Кернигана–Лина. Вы можете использовать функцию `kernighan_lin_bisection()` библиотеки NetworkX. Сравните результирующее двухчастное деление с естественным делением сети и выявите сходства и различия.
- 6.8 В каждой дендрограмме, созданной агломеративной иерархической кластеризацией, горизонтальная линия указывает на слияние двух групп узлов. Если сеть имеет N узлов, каково общее число возможных горизонтальных линий дендрограммы?
- 6.9 Сравните два метода обнаружения сообществ в сети клубов карате «Закари». Во-первых, примените алгоритм Гирвана–Ньюмана, используя функцию `community.girvan_newman()` библиотеки NetworkX. Подтвердите, что раздел P_{NG} в пяти кластерах является разделом с наибольшей модулярностью. (Подсказка: обратитесь к учебному материалу этой главы.) Во-вторых, примените функцию библиотеки NetworkX для жадной оптимизации модулярности, `community.greedy_modularity_communities()`. Сколько сообществ существует в результирующем разделе P_G ? Какой раздел имеет более высокую модулярность, P_{NG} либо P_G ?

- 6.10** Вспомните, что двудольная сеть состоит из двух классов узлов, например A и B , и связи соединяют только узлы в A с узлами в B . Обратитесь к примеру на рис. 6.18, где классы окрашены в красный и синий цвета. Покажите, что модулярность деления двудольной сети в двух группах A и B равна $-1/2$. Это значение также является самым низким, которого может достичь модулярность.

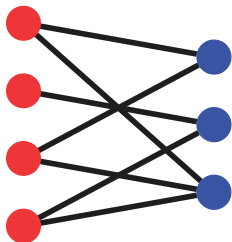


Рис. 6.18 Схематический пример двудольного графа

- 6.11** Предположим, что у вас есть клика с N узлами. Покажите, что любое двухчастное деление клики имеет отрицательную модулярность. Для дополнительной оценки покажите, что модулярность является отрицательной для любого деления клики на более чем один кластер. (Подсказка: пусть две группы имеют соответственно N_A и $N_B = N - N_A$ узлов. Результат соблюдается для любого значения N_A .) Поскольку раздел в одном кластере имеет нулевую модулярность, он также является разделом с максимальной модулярностью: *оптимизация модулярности не разбивает клики!*
- 6.12** Рассчитайте модулярность для двух разделов кольца клик на рис. 6.13: того, в котором каждая клика является сообществом, и того, в котором клики спарены. Подтвердите баллы, указанные в подписи.
- 6.13** Предположим, что A и B — это два из $q > 2$ кластеров сетевого деления со степенями соответственно k_A и k_B . Для простоты допустим, что k_A и k_B приближенно одинаковы: $k_A \approx k_B = k^*$. Обозначим через L_A^{int} , L_B^{int} и L_{AB} число связей соответственно внутри кластера A , внутри кластера B и между A и B . Вычислите разницу в модулярности между этим разделом и разделом, в котором A и B слиты воедино. (Подсказка: поскольку модулярность — это сумма по всем кластерам, вы можете пренебречь вкладами, поступающими из всех кластеров, кроме A и B , которые одинаковы для обоих разделов и уравнивают разницу.) Какое условие на k^* делает раздел со слитыми воедино A и B модулярнее, чем тот, в котором они поделены? Лимит разрешающей способности

модулярности вытекает из этого условия и применяется к парам кластеров, которые соединены по меньшей мере одной связью ($L_{AB} > 0$).

- 6.14** Примените алгоритм распространения меток на сети клубов карате «Закари». Используйте функцию `asyn_lpa_communities()` библиотеки NetworkX. Сравните результат с естественным делением сети.
- 6.15** Предположим, что $q \times q$ -матрица стохастической блочной модели имеет ненулевые элементы только по диагонали. Какой вывод можно сделать о сетях, генерируемых этой моделью?
- 6.16** Напишите программу, которая при заданном на входе значении ожидаемой внешней степени $\langle k^{ext} \rangle$ генерирует эталон GN. Выполните следующие ниже шаги.
1. Назначьте метки узлам в одной и той же группе таким образом, чтобы узлы с 0 по 31 имели метку c_1 , узлы с 32 по 63 имели метку c_2 , узлы с 64 по 95 – метку c_3 , а узлы с 96 по 127 – метку c_4 .
 2. Рассчитайте вероятности $p_{ext} = \langle k^{ext} \rangle / 96$ и $p_{int} = (16 - \langle k^{ext} \rangle) / 31$.
 3. Прокрутите все пары узлов в цикле. Если два узла имеют одинаковую метку (т. е. они находятся в одной группе), то добавьте связь с вероятностью p_{int} , в противном случае – с вероятностью p_{ext} .
- 6.17** Протестируйте алгоритм Лувена на эталоне GN. (Подсказка: можно установить и импортировать модуль `community` из пакета `python-louvain`, как показано в разделе 6.3.2.) Для конструирования эталона примените процедуру, описанную в упражнении 6.16. Используйте следующие значения ожидаемой внешней степени эталона GN: $\langle k^{ext} \rangle = 2, 4, 6, 8, 10, 12, 14$. Для каждого значения сгенерируйте 10 разных конфигураций эталона и примените алгоритм Лувена к каждой из них. Используйте долю правильно обнаруженных узлов (определенных в разделе 6.4), чтобы вычислить сходство каждого раздела с внедренным разделом эталона. Рассчитайте среднее значение и стандартное отклонение сходства по 10 реализациям и постройте график среднего значения как функции от $\langle k^{ext} \rangle$, используя стандартное отклонение для столбцов ошибок. Что вы наблюдаете? Почему?
- 6.18** Создайте случайные сети Эрдеша–Реньи с $N = 1000$ узлами и $L = 5000, 10\,000, 15\,000, 20\,000, 25\,000$ и $30\,000$ связями. Примените алгоритм Лувена к каждой сети. (Подсказка: можно установить и импортировать модуль `community` из пакета `python-louvain`, как показано в разделе 6.3.2.) Проверьте значения модулярности результирующих разделов: близки ли они к нулю?

Постройте график числа сообществ как функции от средней степени случайных сетей: какой вывод можно сделать из его тренда?

- 6.19** Рассмотрите сеть с Нузлами. Вычислите сходство между делением, в котором сеть является единым сообществом, и делением на синглтоны, используя нормализованную взаимную информацию. Зависит ли оно от структуры сети, т. е. дает ли случайная сеть результат, отличный от полной сети?
- 6.20** Иногда сеть слишком велика для быстрого анализа сообщества, и вы, возможно, захотите поработать с подсетью на основе выборки узлов и всех связей между ними. Существует несколько способов отбора узлов. Случайный отбор получается путем рассмотрения каждого узла с одинаковой вероятностью, независимо от структуры сети. Выборка по принципу снежного кома получается, начиная с одного или нескольких узлов, затем добавляя их соседей и так далее, до тех пор, пока не будет включено достаточное число узлов. Это можно делать с помощью алгоритма поиска сперва в ширину, описанного в разделе 2.5. Возьмите один из крупных наборов данных, имеющихся в репозитории книги на GitHub, например сеть киносозвездий IMDB и постройте две подсети с $N = 1000$ узлами, используя эти два метода отбора.
1. Сравните плотности двух подсетей: одинаковы ли они? Почему да, или почему нет?
 2. Сравните средние длины путей двух подсетей: одинаковы ли они? Почему да, или почему нет?
 3. Сравните степенные распределения двух подсетей: одинаковы ли они? Почему да, или почему нет?
- 6.21** Используйте набор данных киносозвездий IMDB, имеющийся в репозитории книги на GitHub, чтобы выполнить анализ сообществ сети, показанных на рис. 0.2(b). Чему соответствуют главенствующие сообщества с точки зрения признаков, общих для актеров и актрис, – жанры? периоды? языки или страны происхождения? Какие алгоритмы работают лучше всего? Дают ли они непротиворечивые результаты? (*Подсказка 1:* сеть является крупной, поэтому вы можете начать с подсети, основанной на выборке узлов, как описано в упражнении 6.20. *Подсказка 2:* можно найти идентификаторы кинозвезд в файлах данных, а затем выполнить поиск по imdb.com, чтобы получить больше информации. Например, ИД Мэрилин Монро равен nm0000054, а ее данные можно найти по адресу imdb.com/name/nm0000054.)
- 6.22** Проанализируйте структуру сообществ электронной почты компании Enron (набор данных имеется в репозитории книги на GitHub, в папке email-Enron). Сможете ли вы выявить модуль

с правой стороны на рис. 0.4? (*Подсказка*: трактуйте сеть как ненаправленную. Она имеет крупный размер, поэтому вам следует сосредоточиться на ядре, как показано на рис. 0.4; используйте функцию `k_core()` библиотеки NetworkX из раздела 3.6 с $k = 43$. Сначала вам нужно будет удалить самонаправленные циклы.)

- 6.23** Проанализируйте структуру сообществ математической сети «Википедии» (набор данных имеется в репозитории книги на GitHub, в папке `enwiki_math`). Обсудите темы разных сообществ, которые вы видите на рис. 0.5. (*Подсказка*: трактуйте сеть как ненаправленную. Так как она имеет крупный размер, следуйте подсказке из упражнения 6.22, чтобы сосредоточиться на ядре $k = 30$. Вы можете найти названия статей, используя атрибут узла «title» (заголовок) после прочтения файла `enwiki_math.graphml`.)
- 6.24** Проанализируйте структуру сообществ сети интернет-маршрутизаторов (набор данных имеется в репозитории книги на GitHub, в папке `tech-RL-caida`). Алгоритм Лувена использовался для обнаружения сообществ, выделенных разными цветами на рис. 0.6. Расследуйте сообщество с крупнейшими хабами. Какова его средняя степень? (*Подсказка*: эта сеть имеет очень крупный размер, поэтому вы можете сначала выполнить разложение ядра, как описано в предыдущих упражнениях. Найдите подходящее значение k , чтобы в сетевом ядре осталась пара тысяч узлов.)





Динамика: силы или свойства, которые стимулируют рост, развитие или изменения внутри системы или процесса.

За четыре дня до выборов в США в 2016 году некий сайт теорий заговора опубликовал ложное новостное сообщение, в котором утверждалось, что сотрудники одного из кандидатов в президенты занимались сатанинскими ритуалами. Фальшивые новости распространялись вирусным образом в Twitter, в основном среди сторонников оппозиционного кандидата, которые приняли сфабрикованные истории, укрепляющие их убеждения, как факт. Автоматизированные учетные записи, которые называются социальными ботами, также способствовали распространению указанного сообщения, расширяя его охват. Это был лишь один пример из тысяч ложных новостей, распространявшихся во время избирательной кампании, – настоящая эпидемия, которая повлияла на мнения и, по словам некоторых экспертов, могла даже повлиять на результаты выборов.

Ученые изучают факторы, которые делают людей и социально-медийные платформы уязвимыми для такого рода манипуляций. Ученые-сетевики, в частности, изучают эти явления, потому что структура онлайн-социальных сетей играет ключевую роль в вирусной природе определенных сообщений. Например, на рис. 7.1 показана часть диффузионной сети упомянутого выше поддельного новостного репортажа. Мы сразу замечаем, что некоторые узлы, в том числе социальные боты, были особенно влиятельными.

Распространение дезинформации – это особый случай диффузии информации, один из классов динамических процессов, происходящих в сетях. В этой главе рассматривается несколько других важных типов сетевых процессов в дополнение к диффузии информации: эпидемия, формирование мнений и поиск. В каждом случае мы фокусируемся на динамике, т. е. на том, что происходит в сети с течением времени – как информация и заболевания передаются по связям, как атрибуты узлов подвергаются влиянию со стороны их взаимодействий и как можно проводить поиск или навигацию по сетям. Мы представим несколько *моделей*, которые отражают эти виды динамики.

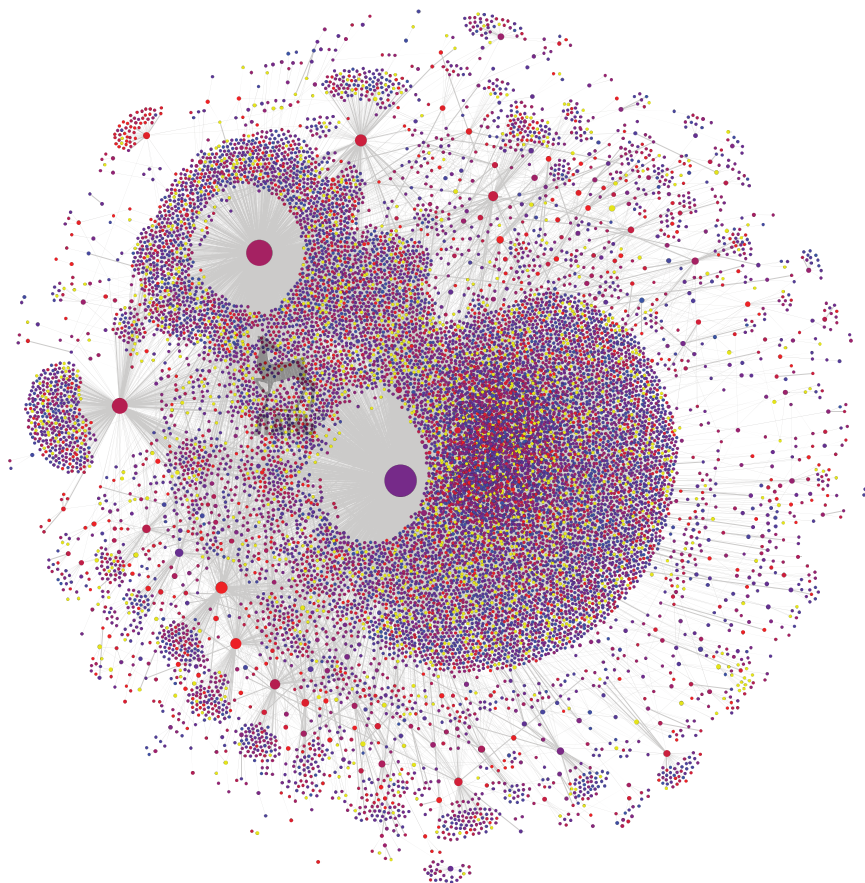


Рис. 7.1 Ядро сети диффузии вирусного сфабрикованного новостного репортажа под названием «Духовная кулинария: председатель кампании Хиллари Клинтон практикует странный оккультный ритуал», опубликованного сайтом теорий заговора InfoWars за четыре дня до выборов в США в 2016 году и более чем в 30 000 твитах. Узлы представляют собой учетные записи Twitter. Связь между двумя узлами указывает на то, что один из соответствующих аккаунтов ретвитнул сообщение другого, содержащее статью. Размер узла указывает на влияние учетной записи, измеряемое числом раз, когда делящаяся этой статьей учетная запись была ретвитнута (сила-на-выходе). Цвет узла отражает правдоподобие (понимаемое как шанс возникновения) того, что учетная запись автоматизирована, от синего (вероятно, человек) до красного (вероятно, бот); желтые узлы невозможно оценить, поскольку они были приостановлены. Вспомните из главы 4, что Twitter не предоставляет данные для восстановления ретвитного дерева; все ретвиты указывают на изначальный твит. Показанная здесь ретвитная сеть комбинирует в себе несколько каскадов (каждый из которых представляет звездную сеть, происходящую из другого твита), все из которых имеют одну и ту же общую статью. Изображение предоставлено Шао и соавт. (2018b); интерактивная версия этой сети доступна в интернете (iunetsci.github.io/HoaxyBots/)

7.1. Идеи, информация, влияние

Сети играют центральную роль в том, как идеи и информация распространяются в социальном сообществе. Мы нередко узнаем что-то новое через друзей: например, можем узнать о новой модели смартфона, потому что наш лучший друг только что его купил, или узнать последние новости из области внешней политики США, потому что подруга нам об этом рассказывает или переслала статью, которую она только что прочитала.

Действительно многое из того, что мы делаем, прямо или косвенно определяется нашими социальными контактами. Социальное влияние является решающим фактором, когда мы перенимаем поведение, принимаем решение, внедряем инновации или формируем наши культурные, политические и религиозные взгляды. Поэтому моделирование того, как влияние, идеи и информация распространяются в социальных сетях, является ключевым приложением науки о сетях. Эти процессы распространения также называются *социальным заражением*, потому что они напоминают болезнь, которая передается через контакты между индивидуумами. На самом деле, как мы увидим в разделе 7.2.2, социальное заражение часто моделируется как распространение эпидемии.

В любой модели распространения влияния мы исходим из допущения, что первоначально активируется определенное число узлов (*влиятелей*), представляющих, что они приняли новую идею, инновацию, поведение и т. д. Затем каждый неактивный узел активируется (или нет) в соответствии с некоторым правилом, которое зависит от присутствия активных соседей и от других условий и параметров, как показано на рис. 7.2. Исходом этого процесса является генерация *каскадов влияния*, активация в последовательности подмножества узлов в сети. Каскады могут варьироваться от нескольких узлов до *глобальных каскадов*, охватывающих значительную часть сети. Иногда несколько узлов в итоге влияет на всю сеть. В разделе 4.5 мы обсудили структуру каскадных сетей; в целях ознакомления с тем, как эти каскады разворачиваются с течением времени, давайте обсудим два главных класса моделей социального заражения, основанных на *порогах* и *независимых каскадах*.

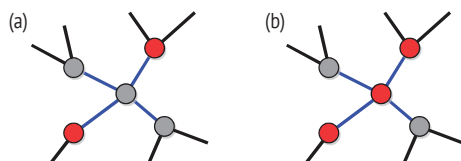


Рис. 7.2 Диффузия социального влияния в сетях. (а) Центральный (серый) узел неактивен и имеет двух активных (красных) и двух неактивных (серых) соседей. (б) Узел становится активным из-за влияния его активных соседей

7.1.1. Пороговые модели

Принцип пороговых моделей очень прост: узел может быть активирован только в том случае, если влияние, оказываемое на него его активными соседями, превышает определенное значение. В самой базовой версии *линейной пороговой модели* влияние на узел определяется как сумма его активных соседей, в которой вклад каждого соседа определяется весом связи, соединяющей его с узлом: чем сильнее связь, тем выше влияние соседа. Если влияние превышает определенный для узла порог, то узел становится активным, а значит, он принимает идею, информацию или поведение.

В линейной пороговой модели влияние на узел i выражается формулой:

$$I(i) = \sum_{j: \text{active}} w_{ji}. \quad (7.1)$$

В уравнении (7.1) сумма включает только активных соседей узла i ; если узел j не является соседом, то не существует никакой связи, которая соединяла бы его с i , а $w_{ji} = 0$. Условие для активации узла i таково:

$$I(i) \geq \theta_i, \quad (7.2)$$

где θ_i – это конкретный порог узла i , который назначается узлу до того, как процесс начнется. Такой порог указывает на склонность индивидуума подвергаться влиянию, которая обычно варьируется от одного индивидуума к другому. Если граф невзвешенный, то уравнение (7.2) сводится к

$$n_i^{on} \geq \theta_i, \quad (7.3)$$

где n_i^{on} – это число активных соседей узла i . В этом случае если число активных соседей превышает порог узла, то узел активируется, в противном случае он остается неактивным. Если все узлы имеют одинаковый порог θ , то уравнение (7.3) превращается в простое условие о том, что любой неактивный узел должен иметь по меньшей мере θ активных соседей, чтобы стать активным.

Модель работает следующим образом. Сначала мы выбираем нашу сеть, которая может происходить из реально существующих данных или из модели генерирования графов, подобной тем, которые были представлены в главе 5. Для простоты давайте допустим, что граф не

является взвешенным. Далее мы назначаем всем узлам порог, например путем генерирования случайных чисел в некотором интервале. Затем активируется заданное число узлов; опять же, они могут отбираться случайно. Наконец, мы проходим итеративные этапы, в ходе которых неактивные узлы могут стать активными, основываясь на активации их соседей.

Каждая итерация модельной динамики состоит из следующих ниже операций.

1. Все активные узлы остаются активными.
2. Каждый неактивный узел активируется, если число активных соседей равно или превышает его порог.

Эти шаги повторяются до тех пор, пока другие узлы не будут активированы.

В моделях сетевой динамики порядок рассмотрения узлов не должен влиять на результат. Обеспечить это при имплементировании правил обновления узлов можно двумя способами. В *асинхронных* имплементациях узлы оцениваются в разной случайной последовательности на каждой итерации. Это делается для того, чтобы избежать систематических смещений, которые могут возникать в результате постоянного следования одной и той же последовательности. В *синхронных* имплементациях новое состояние активации каждого узла на каждой итерации определяется с использованием значений активации других узлов из предыдущей итерации; затем все узлы обновляются в конце итерации. Порядок в данном случае не имеет значения.

Был предложен целый ряд вариаций линейной пороговой модели. В *дробно-пороговой модели* мы рассматриваем дробь, а не число активных соседей. Поэтому, для того чтобы в указанной модели можно было бы активировать узел с порогом, к примеру, $1/2$, по меньшей мере половина его соседей должна быть активна. На рис. 7.3 показано то, как динамика модели разворачивается на простой сети: активация одного узла запускает каскад, который в конечном итоге приводит к активации всех остальных узлов.

В дробно-пороговой модели условие активации таково:

$$\frac{n_i^{on}}{k_i} \geq \theta_i, \quad (7.4)$$

где k_i – это степень узла i . Соотношение в левой части уравнения (7.4) представляет долю активных соседей узла i . Если все

узлы имеют одинаковый порог θ , то указанное условие состоит в том, что для активации неактивного узла необходимо, чтобы по меньшей мере доля θ активных соседей была активирована.

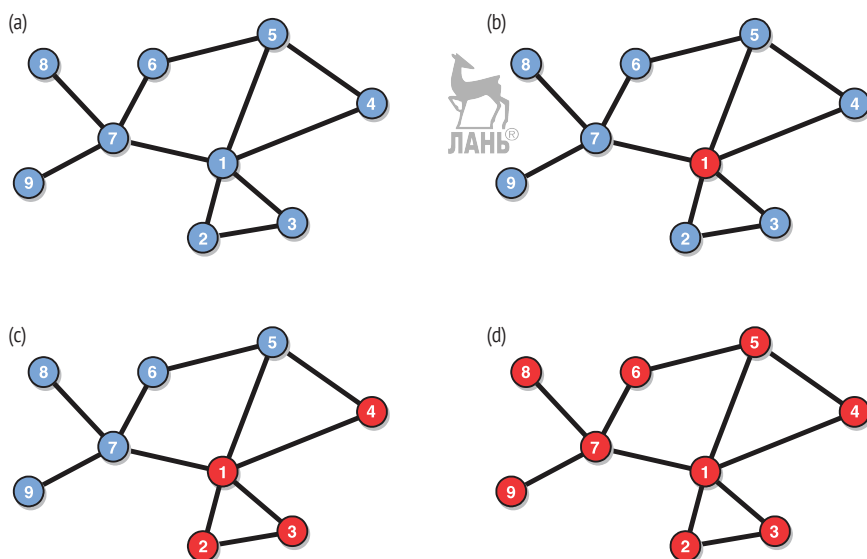


Рис. 7.3 Дробно-пороговая модель диффузии влияния. Порог активации равен $1/2$ для всех узлов. (a) Изначально все узлы неактивны. (b) Узел 1 активирован. (c) Узлы 2, 3 и 4 имеют двух соседей, и один из них равен 1, который активен, поэтому они активируются. (d) После активации узла 4 узел 5 имеет двух активных соседей из трех и становится активным (так как $2/3 \geq 1/2$). Аналогичным образом впоследствии активируются узлы 6, 7, 8 и 9

Если сеть является разреженной, то, будет ли запущен глобальный каскад или нет, все зависит от ее структуры. Ключевыми движущими силами являются *уязвимые узлы* (т. е. узлы, которые могут быть активированы одним активным соседом).

Из уравнения (7.4) мы видим, что узел является уязвимым, если $k_i \leq 1/\theta$, т. е. если его степень ниже или на уровне величины, обратной его порогу.

Для того чтобы иметь глобальные каскады, число уязвимых узлов должно быть достаточно крупным. Хабы обычно являются очень эффективными влиятелями: чем больше число соседей, тем больше вероятность того, что некоторые из них имеют достаточно низкую степень для того, чтобы быть уязвимыми. Однако быть хабом не всегда является достаточным условием для влияния. Положение влиятеля

в сети также имеет важность: каскад на периферии сети вряд ли справится с прокладыванием себе пути через ядро.

Еще одним аспектом сетевой структуры, который играет важную роль в размере каскада, является плотность и сепарация между сообществами. Распространение облегчается внутри плотных сообществ, но сдерживается между сообществами. Кластерные границы действуют как стены, потому что узел вряд ли будет иметь нескольких активных соседей в разных сообществах.

Знание структуры сети обеспечивает нам возможность контролировать размер каскадов. В примере на рис. 7.3 если первоначальным влиятелем является узел 7, то его соседи 6, 8 и 9 станут активными, но каскад на этом останавливается, потому что доли активных соседей узлов 1 и 5 равны соответственно $1/5$ и $1/3$, оба ниже $1/2$. Однако, если нам также удастся поочередно активировать, к примеру, узел 2, то узел 3 тоже станет активным, а узлы 2, 3 и 7 активируют узел 1, позволив каскаду распространиться на всю сеть целиком. Таким образом, в данном случае воздействие на узел 2 «деблокирует» каскад. И действительно, успех товара или идеи часто зависит от выявления ключевых индивидуумов, которых необходимо убедить их купить. Этот вопрос занимает центральное место в вирусном маркетинге, где социальные сети используются для продвижения товаров. В приложении В.6 книги представлена демонстрация мелко-пороговой модели.

7.1.2. Независимо-каскадные модели

Пороговые модели основаны на концепции *давления со стороны сверстников*: чем больше наших контактов делят между собой одну идею или владеют товаром, тем больше вероятность того, что мы сами ее или его примем. Это похоже на то, как если бы наши активные социальные соседи работали вместе, чтобы нас убедить. Но социальное влияние часто происходит один на один: нас можно убедить принять товар или убеждение, если об этом с энтузиазмом говорит один единственный друг. Каждый из наших других контактов будет иметь свое собственное влияние, если только мы уже не купили товар или идею. Независимые каскадные модели фокусируются на таких межузловых взаимодействиях.

Установочная настройка модели такая же, как и для пороговых моделей, в том смысле, что выбирается или строится сеть, и некоторые узлы активируются. Как только узел становится активным, у него есть один шанс «убедить» каждого своего неактивного соседа; каждый сосед активируется с некоторой *вероятностью влияния*. Если узлу не удастся активировать своего друга, то он уже не сможет повторить попытку. Однако друга все еще может убедить другой активный сосед. Процесс, показанный на рис. 7.4, продолжается до тех пор, пока дальнейшие активации не прекратятся.

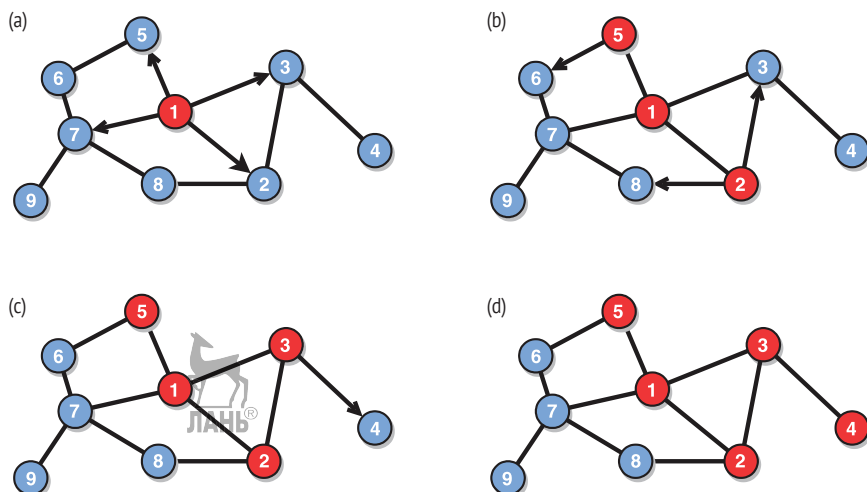


Рис. 7.4 Независимо-каскадная модель. Вероятность влияния устанавливается равной $1/2$ для всех пар узлов, поэтому успех каждого взаимодействия определяется подбрасыванием справедливой монеты. Стрелки указывают на то, кто на кого пытается повлиять. (a) Узел 1 активирован и пытается повлиять на своих неактивных соседей 2, 3, 5 и 7. (b) Узлы 2 и 5 становятся активными и оказывают свое влияние на 3, 6 и 8. (c) Узел 3 активируется и пытается убедить 4. (d) Узел 4 становится активным, и каскад останавливается

В простейшей версии независимо-каскадных моделей активный узел i имеет вероятность p_{ij} убедить своего неактивного соседа j . Такая вероятность обычно зависит только от конкретной пары влиятель-сосед, поэтому на исход каждого взаимодействия не влияет то, что происходит с другими парами. В асинхронных имплементациях если j имеет несколько активных соседей, то их попытки активации следуют в произвольном порядке во избежание систематического смещения. Вероятности влияния p_{ij} и p_{ji} могут отличаться, потому что в общем случае каждый узел имеет свою способность убеждать и восприимчивость к убеждению. Таким образом, для i бывает легче влиять на j , чем наоборот. Вероятность p_{ij} может интерпретироваться как вес связи, указывающей из i в j .

Очевидно, что чем больше число активных соседей неактивного целевого узла, тем больше число попыток повлиять на узел и тем больше вероятность того, что он будет активирован. Следовательно, пороговые модели и независимо-каскадные модели взаимосвязаны, но существуют важные различия. Пороговые модели сосредоточены на цели, которая активируется, если удовлетворяется пороговое условие. Независимо-каскадные модели сосредоточены на влиятеле, который убеждает своих неактивных соседей с заданной вероятностью.

В дополнение к этому пороговые модели обычно являются *детерминированными*. Активация любого узла зависит от соблюдения или несоблюдения порогового условия; случайность не играет никакой роли. Из этого следует, что если мы начнем с одного и того же первоначального множества активных узлов и будем активировать узлы синхронно, то может существовать только один исход. Независимо-каскадные модели вместо этого являются *вероятностными*: развитие динамики зависит от случайности. В примере на рис. 7.4 при первоначальной активации узла 1 могут запускаться разные каскады. В независимо-каскадной модели мы можем «деблокировать» каскад, активировав дальнейшие узлы, выбранные соответствующим образом, как мы увидели в разделе 7.1.1 в случае линейной пороговой модели. Однако из-за вероятностного характера модели трудно делать предсказания относительно будущего продвижения каскада, даже когда структура сети известна.

Нельзя ожидать, что очень простые описанные нами модели будут воспроизводить реальную динамику социального заражения. Однако более сложные вариации этих моделей способны улавливать важные признаки многих реально существующих явлений. Одним из примеров является вероятностная версия пороговой модели, в которой шансы на активацию растут с увеличением числа активных соседей. Это похоже на независимо-каскадную модель, но контакты с активными соседями не являются независимыми друг от друга. Такой механизм моделирует процессы так называемого *сложного заражения*: каждый новый человек, знакомящий нас с товаром или идеей, оказывает большее влияние, чем предыдущие, на то, чтобы мы его приняли или в нее поверили.

7.2. Распространение эпидемий

В середине XIV века человечество пережило одно из величайших бедствий в своей истории: черную смерть. Также именуемая Великой чумой, она, как полагают, была вызвана бактерией *Yersinia pestis*, переносимой блохами, живущими на черных крысах, которые регулярно путешествовали на борту торговых судов. По всей видимости, она началась в Центральной Азии и распространилась по всей Европе между 1346 и 1353 годами (рис. 7.5). По оценкам, Черная смерть унесла жизни 30–60 % населения Европы.

Хотя потенциально разрушительные последствия инфекционных заболеваний были эффективно смягчены значительным улучшением условий жизни людей и прогрессом в области медицины и биологии, в особенности за последнее столетие, скорость их распространения значительно возросла благодаря технологическим достижениям в области транспортировки людей. В средние века самым эффективным средством передвижения были лошади по суше и корабли по морю,

и требовались месяцы на то, чтобы добраться до отдаленного пункта назначения. В наши дни перелет через континенты занимает всего несколько часов. Человек, заразившийся Эболой в Африке, может легко отправиться в Европу, Азию или Америку и распространить там болезнь, все еще не подозревая об этом. В последние годы мир неоднократно сталкивался с подобными чрезвычайными ситуациями.

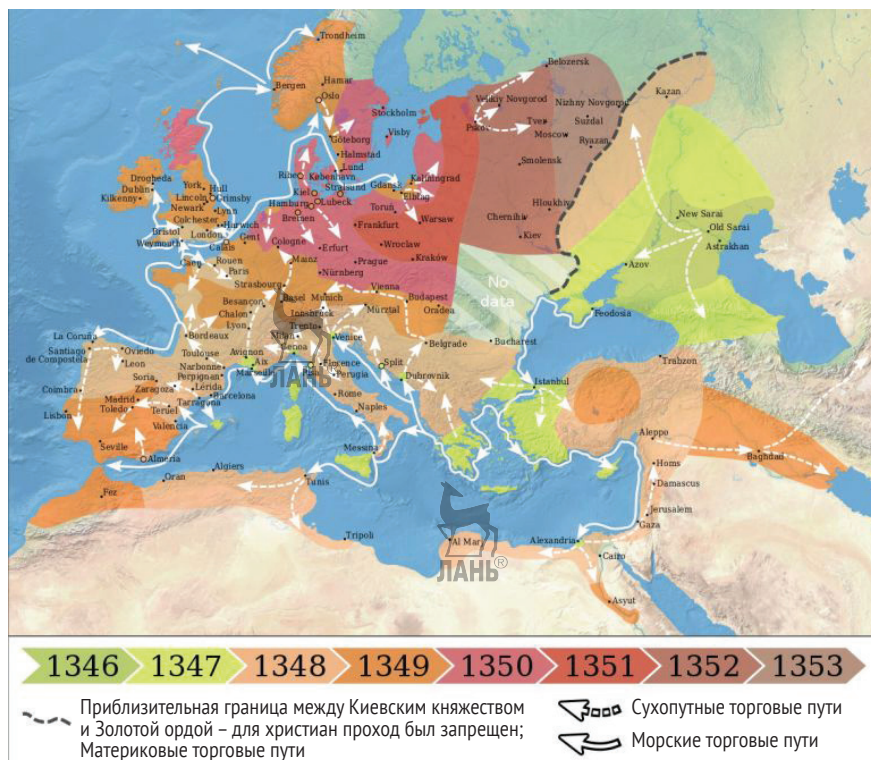


Рис. 7.5 Черная смерть достигла Европы в 1346 году и в течение нескольких лет распространилась по всему континенту. На карте показаны регионы, пораженные болезнью с течением времени, а также вероятные маршруты ее миграции. Изображение принадлежит Flappieff, используется по лицензии CC-BY-SA 4.0 (commons.wikimedia.org/wiki/File:1346-1353_spread_of_the_Black_Death_in_Europe_map.svg)

Кроме того, технологии создали новые формы эпидемий. Компьютерные вирусы и другие вредоносные программы распространяются через интернет, ставя под угрозу работу миллионов устройств. Вирусы мобильных телефонов могут легко передаваться через Bluetooth или службы мультимедийных сообщений (MMS). Онлайн-социальные сети стали благодатной почвой для диффузии слухов, мистификаций, фальшивых новостей, теорий заговора и лженауки. Процессы распространения информации имеют много общего с эпидемиями инфекционных заболеваний.

Эпидемии распространяются в *контактных сетях*, таких как сети физических контактов (рис. 7.6), транспорт (рис. 0.7), интернет (рис. 0.6), электронная почта (рис. 0.4), онлайн-овые социальные сети (рис. 0.1 и 0.3) и мобильная телефонная связь. Многие такие сети характеризуются наличием хабов (обсуждаемых в главе 3), которые играют центральную роль в этом процессе. В оставшейся части этого раздела мы рассмотрим классические модели распространения эпидемий и укажем на ключевые различия в динамике, когда они развиваются в сетях.

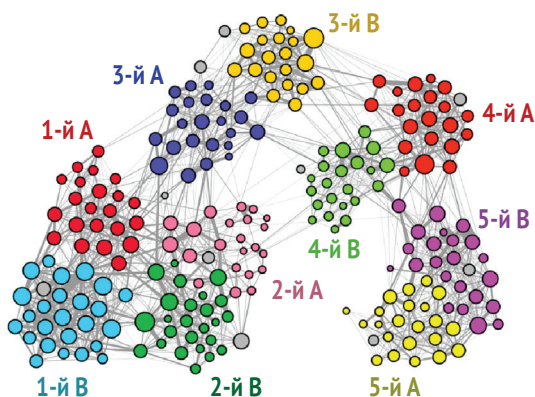


Рис. 7.6 Контактная сеть в начальной школе. Связи указывают на непосредственную близость между детьми и учителями во французской школе, отслеживаемую устройствами радиочастотной идентификации. Цвета обозначают детей в одном классе и градации; учителя показаны серым цветом. Узлы с более высокой степенью имеют больший размер, контакты большей продолжительности представлены более толстыми связями. Хотя каждый ребенок по прошествии достаточно долгого времени в конечном итоге взаимодействует со всеми своими одноклассниками, некоторые из них также взаимодействуют с детьми из других классов. Этот тип сети может предлагать вмешательства, направленные на сдерживание или смягчение распространения инфекционных заболеваний в школах. Изображение перепечатано из Стеле и соавт. (2011) по лицензии CC-BY-4.0

7.2.1. Модели SIS и SIR

Классические эпидемические модели разделяют популяцию на разные отделения, соответствующие разным стадиям заболевания. Двумя ключевыми отделениями являются восприимчивые (S от англ. *susceptible*) и инфицированные (I от англ. *infected*). Восприимчивые индивидуумы могут заразиться болезнью, инфицированные индивидуумы уже заразились ею и могут передавать ее восприимчивым индивидуумам. В зависимости от того, какое заболевание мы рассматриваем, могут потребоваться дополнительные отделения. В *модели восприимчивый–инфицированный–восприимчивый* (SIS) инфицированные индивидуумы снова становятся восприимчивыми, когда они выздоравливают от болезни, поэтому они могут заразиться ею снова

(рис. 7.7). Данная модель применима к заболеваниям, которые не обеспечивают длительного иммунитета, таким как обычная простуда.

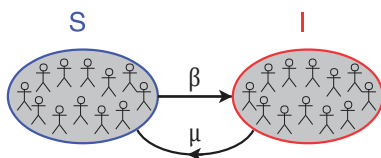


Рис. 7.7 Отделения и переходы в модели SIS. Каждый восприимчивый индивидум заболевает с вероятностью β после каждого контакта с инфицированным индивидумом. На каждом временном шаге у каждого инфицированного индивидума есть вероятность μ выздороветь от болезни и снова стать восприимчивым. Индивидумы могут инфицироваться несколько раз

Модель SIS начинается либо с реально существующей контактной сети, реконструированной на основе эмпирических данных, либо с искусственной сети, сгенерированной с помощью какой-либо модели, подобной тем, которые были представлены в главе 5. Далее мы исходим из допущения, что несколько узлов инфицировано в соответствии с неким критерием (например, случайно). Все остальные узлы являются восприимчивыми. В ходе динамики указанной модели восприимчивые индивидумы заражаются болезнью с определенной вероятностью, именуемой *частотой инфицирования*, при каждой встрече с инфицированным индивидумом. Инфицированные люди на каждом временном шаге выздоравливают от болезни, превращаясь в восприимчивых, с некоторой вероятностью, именуемой *частотой выздоровления*.

На каждой итерации динамики SIS мы посещаем все узлы. Для каждого узла i :

- 1) если i является восприимчивым, то прокрутить его соседей в цикле. Для каждого инфицированного соседа узел i заражается с вероятностью β ;
- 2) если i инфицирован, то i становится восприимчивым с вероятностью μ .

Как и в других моделях распространения, узлы могут посещаться асинхронно в случайном порядке либо синхронно. Частота инфицирования β и частота выздоровления μ являются ключевыми параметрами модели.

Динамика производит ряд переходов из S в I и из I в S, которые при подходящих условиях могут продолжаться бесконечно.

Еще одной классической моделью является *модель восприимчивый–инфицированный–выздоровевший* (SIR). Когда инфицированные

индивидуумы выздоравливают от болезни, они перемещаются в третье отделение *выздоровевших* (R от англ. *recovered*) людей и больше не могут быть инфицированы (рис. 7.8). Эта модель применима к заболеваниям, которые обеспечивают длительный иммунитет, таким как корь, эпидемический паротит, краснуха и т. д. Обратите внимание, что смерть – это особый случай выздоровления от смертельных заболеваний, потому что умершие люди не заражают других. Динамика инфицирования и выздоровления в точности соответствует описанной выше модели SIS с теми же параметрами частоты инфицирования и выздоровления. Единственное отличие состоит в том, что, когда инфицированный индивидуум выздоравливает, он перемещается в состояние R , а не обратно в состояние I ; он не будет играть никакой дальнейшей роли в динамике. В конечном итоге распространение модели SIR прекращается, когда инфицированных индивидуумов больше нет.

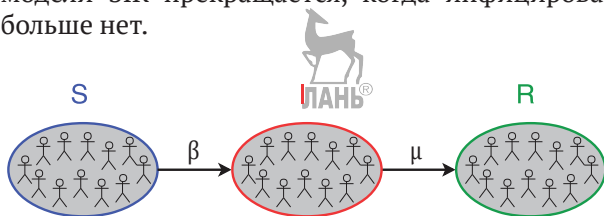


Рис. 7.8 Отделения и переходы в модели SIR. Каждый восприимчивый индивидуум заболевает с вероятностью β после каждого контакта с инфицированным индивидуумом. На каждом временном шаге у каждого инфицированного индивидуума есть вероятность μ выздороветь (или умереть) от болезни

На рис. 7.9 мы видим характерную эволюцию моделей SIS и SIR, построенную по доле популяции, заразившейся этим заболеванием, как функции от времени. Первоначально инфицировано всего несколько человек, и распространение эпидемии происходит нерегулярно и медленно. За этим следует фаза экспоненциального роста, которая может быстро повлиять на большое число людей. Наконец, процесс достигает стационарного состояния, в котором болезнь либо эндемична (т. е. с течением времени поражает стабильную часть населения), либо ликвидирована.

Классические эпидемиологические модели можно упростить, выполнив *аппроксимацию гомогенного перемешивания*, которая заключается в допущении о том, что каждый индивидуум может контактировать с любым другим. По этой причине все индивидуумы в одном и том же отделении ведут себя одинаково, и для динамики модели имеют значение только относительные пропорции людей в различных отделениях. Это эквивалентно допущению о том, что индивидуумы являются узлами полного графа, где каждый связан со всеми остальными. Такое упрощающее допущение бывает оправданным для малой популяции, такой как жители маленькой деревушки, где все люди соприкасаются друг с другом. Но в реальных, крупномас-

штабных эпидемиях индивидуумы могут заражаться только от людей, с которыми они вступают в контакт. Поэтому крайне важно, насколько это возможно, реконструировать фактическую сеть контактов.

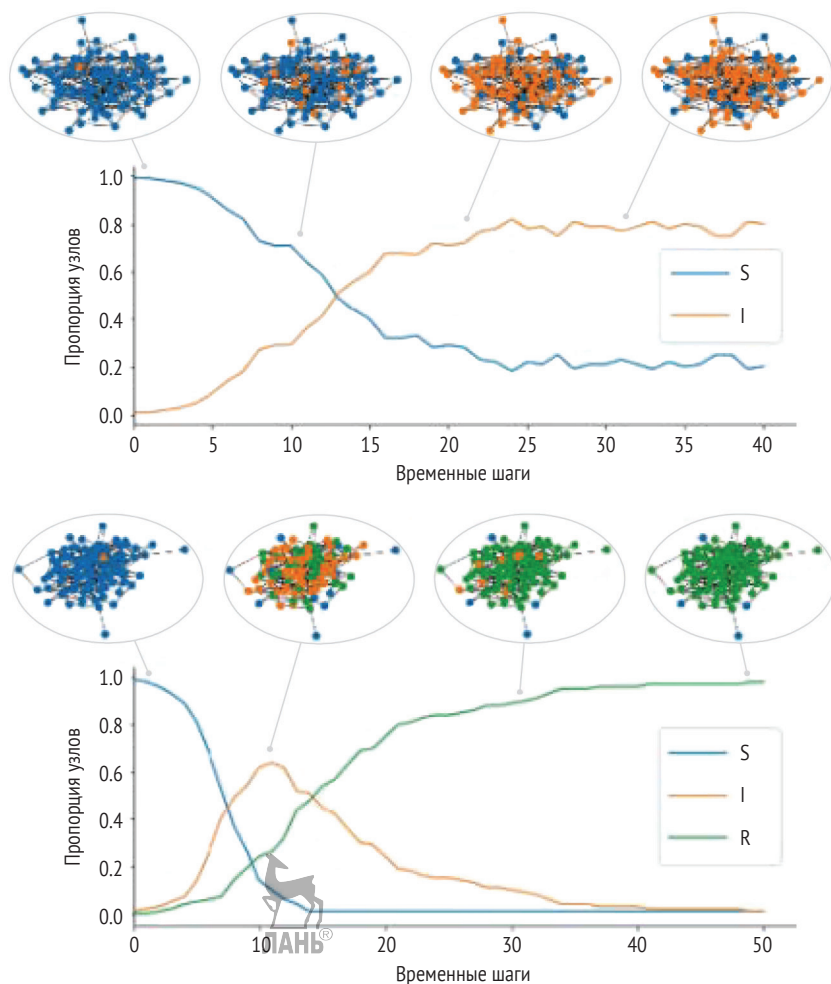


Рис. 7.9 Схематическая эволюция (а) динамики моделей SIS и (б) SIR. Доля инфицированных индивидуумов строится по сравнению с временем после вспышки эпидемии. После первоначальной фазы, характеризующейся низкой долей инфицированных людей, эпидемия быстро разрастается до тех пор, пока заболевание не поразит некую долю населения. Заключительная фаза зависит от модели: в случае модели SIS инфицированные стабилизируются вокруг постоянной доли (которая также может быть очень малой или даже нулевой), сигнализируя об эндемическом состоянии. В случае модели SIR доля инфицированных всегда снижается до нуля по мере выздоровления индивидуумов

На каждой итерации модели появляются вновь инфицированные индивидуумы, именуемые *вторичными инфекциями*, помимо больных индивидуумов, которые выздоравливают после болезни. Для рас-

пространения эпидемии вторичных инфекций должно быть больше, чем выздоровевших людей, потому что только таким образом число инфицированных людей может расти. На гомогенных сетях, где все узлы имеют одинаковую степень, имея в виду, что каждый индивидуум вступает в контакт примерно с одинаковым числом людей, это условие приводит к *пороговому эффекту*. Мы можем определить *базовое репродукционное число* как среднее число вновь инфицированных людей, генерируемое инфицированным индивидуумом в течение его инфекционного периода. Указанная величина зависит от частоты инфицирования, частоты выздоровления и средней степени. Если она превышает порог, то эпидемия может затронуть значительную часть популяции; в противном случае она будет быстро поглощена без серьезных последствий.

Давайте допустим наличие гомогенной контактной сети, все узлы которой имеют степень, приближенно равную среднему значению $\langle k \rangle$. Согласно динамике моделией SIS и SIR каждый больной человек инфицирует восприимчивого соседа с вероятностью β . На ранних стадиях эпидемии инфицировано всего несколько человек, поэтому можно допустить, что каждый из них контактирует с наиболее восприимчивыми индивидуумами. Каждый инфицированный человек на каждой итерации может передавать болезнь примерно $\langle k \rangle$ индивидуумам. Следовательно, среднее число инфекций, вызванных одним человеком после одной итерации на ранней стадии процесса распространения, равно $\beta \langle k \rangle$. С другой стороны, на каждой итерации каждый больной индивидуум выздоравливает с вероятностью μ . Поэтому если есть I инфицированных индивидуумов, то после одной итерации в среднем будет $I_{\text{sec}} = \beta \langle k \rangle I$ вторичных инфекций, тогда как ожидается, что $I_{\text{rec}} = \mu I$ людей выздоровеют. Для распространения эпидемии мы должны иметь $I_{\text{sec}} > I_{\text{rec}}$, которое приводит к условию *эпидемического порога*:

$$\beta \langle k \rangle I > \mu I \Rightarrow R_0 = \frac{\beta}{\mu} \langle k \rangle > 1. \quad (7.5)$$

Переменная $R_0 = \beta \langle k \rangle / \mu$ – это *базовое репродукционное число*. Уравнение (7.5) гласит, что если $R_0 < 1$, то первоначальная вспышка угасает за короткое время, затрагивая лишь несколько индивидуумов. Если $R_0 > 1$, то эпидемия может продолжить распространяться.

Для того чтобы эпидемия затронула значительную часть популяции, каждый инфицированный человек должен передавать заболевание более чем одному другому индивидууму. Это условие является

необходимым, но недостаточным: в определенных ситуациях эпидемия может и не иметь серьезных последствий, даже если базовое репродукционное число превышает единицу; такие факторы, как карантинные политики или структура сетевых сообществ, могут предотвращать распространение эпидемии. В общем случае чем выше базовое репродукционное число, тем более инфекционным является заболевание. Например, это число превышает 10 для кори и колеблется вокруг двух для Эболы.

В приложении B.5 книги представлена демонстрация моделей SIS и SIR на гомогенных сетях. Но, как мы увидели, реально существующие контактные сети не являются гомогенными. Наличие хабов существенно меняет сценарий. Если есть узлы с очень большой степенью, то *фактически нет порога*: значительную часть популяции могут в итоге затрагивать даже болезни с низким уровнем инфицирования и/или высокой частотой выздоровления! На самом деле, даже если вероятность инфицирования этой болезнью невелика, довольно легко инфицировать один или несколько хабов, которые очень подвержены риску из-за большого числа контактов. После заражения хабы становятся опасными распространителями среди своих многочисленных восприимчивых контактов, которые будут распространять инфекцию дальше к своим контактам и т. д.

Из-за роли хабов при столкновении с реальными чрезвычайными ситуациями, связанными с эпидемией, эффективные стратегии сдерживания должны направляться на вакцинацию либо изоляцию людей со многими контактами. Например, работники секс-бизнеса являются основными объектами кампаний вакцинирования от инфекций, передаваемых половым путем. Во многих случаях не очевидно, как определять хабы контактной сети. В разделе 3.3 предлагается один такой способ. Следуя по связям сети, мы увеличиваем вероятность натолкнуться на хабы (каламбур). Поэтому, вместо того чтобы вакцинировать случайную выборку популяции, следует вакцинировать их друзей!

7.2.2. Распространение слухов

Социальное заражение можно описать естественным образом как распространение эпидемии. Рассмотренные нами модели социального заражения по сути дела имеют общие черты с моделями SIS и SIR, в особенности это касается независимо-каскадных моделей раздела 7.1.2.

Вариант модели SIR может использоваться для описания распространения слухов в сообществе. Как и модель SIR, указанная *модель распространения слухов* имеет три отделения: невежи (S), распространители (I) и подавители (R). Последние – это люди, которые знают о слухе, но не способствуют его распространению. Основная идея заключается в том, что люди занимаются диффузией слуха до тех пор,

пока они находят людей, которые о нем не знают, в противном случае они теряют интерес и прекращают распространять слух.

Модель распространения слуха начинается с сети контактов, которая может являться реально существующей или искусственной сетью, сгенерированной какой-либо моделью, подобной тем, которые мы увидели в главе 5. Все узлы являются невежами, за исключением некоторых, которые превращены в распространителей слуха в соответствии с неким критерием; они могут отбираться случайно. В динамике модели, когда распространитель приближается к невеже, передается слух, и невежа становится распространителем с *вероятностью передачи*. Когда распространитель встречается подавителя, распространитель становится подавителем с *вероятностью остановки*. Когда встречаются два распространителя, они оба превращаются в подавителей с одинаковой вероятностью остановки. Рисунок 7.10 иллюстрирует эти переходы. Если невежа встречается подавителя, то ничего не случается.

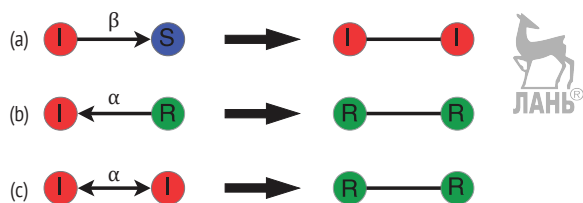


Рис. 7.10 Модель распространения слуха. (а) Слух распространяется только в том случае, если распространитель (I) встречается с невежей (S). В этом случае невежа становится распространителем с вероятностью β . (б) Если распространитель встречается подавителя (R), то распространитель становится подавителем с вероятностью α . (с) Если встречаются два распространителя, то они оба становятся подавителями с вероятностью α

На каждой итерации динамики модели распространения слуха все узлы посещаются синхронно либо асинхронно в случайном порядке. Для каждого узла i :

- 1) если i является невежей, то прокрутить его соседей в цикле. Для каждого распространяющего соседа узел i становится распространителем с вероятностью β ;
- 2) если i является распространителем, то прокрутить его соседей в цикле.
 - (i) Для каждого подавляющего соседа узел i становится подавителем с вероятностью α .
 - (ii) Для каждого распространяющего соседа узел i и сосед оба становятся подавителями с вероятностью α .

Вероятность передачи β и вероятность остановки α являются двумя ключевыми параметрами модели.

Важным отличием от модели SIR является то, что здесь переход из I в R происходит не спонтанно (в том смысле, что больной человек выздоравливает от болезни), а зависит от взаимодействия между индивидуумами. Как и в модели SIR, начиная с нескольких распространителей, в конечном итоге все индивидуумы будут либо невестами, либо подавителями, так как в этом случае динамика не может приводить к каким-либо изменениям. Число подавителей в окончательном состоянии также является числом людей, которые узнали об этом слухе.

Модель распространения слуха не имеет порогового эффекта даже в гомогенных сетях. Слух может доходить до большого числа людей, даже если вероятность передачи невелика. В гетерогенных сетях по-прежнему нет порога, и окончательное число осведомленных о слухе людей ниже, чем в гомогенных сетях с равным числом узлов и связей. Это обусловлено тем, что слух достигает хабов на ранних стадиях процесса, и те быстро становятся подавителями из-за их многочисленных взаимодействий с другими индивидуумами, некоторые из которых могут быть осведомлены о слухе. Как только хабы превращаются в подавителей, процесс диффузии замедляется.

7.3. Динамика мнений

У нас есть мнения обо всех и обо всем. Мнения управляют нашим поведением, влияют на наш выбор, влияют на наши планы. Политика, проводимая правительствами во всем мире, диктуется мнениями о торговле, конфликтах, иммиграции, пандемиях, окружающей среде и т. д. Динамика мнений – это процессы, которые определяют то, как мнения формируются и распространяются в обществе. С появлением интернета и социальных сетей человечество наделило себя невероятно мощными инструментами по распространению мнений и даже манипулированию ими. Мнения распространяются в таких сетях, как друзья в Facebook и подписчики в Twitter. Следовательно, сетевые модели помогают нам понимать принципы, по которым мнения распространяются.

Модели динамики мнений аналогичны моделям распространения влияния, рассмотренным в предыдущем разделе, но у них есть некоторые отличительные признаки. Мнение можно представить в виде числа или множества чисел. Модели обычно делятся на две категории в зависимости от того, какие мнения в них используются: *дискретные* (целочисленные) либо *непрерывные* (вещественные). Далее мы представим простые модели в обоих классах. Мы также обсудим взаимную игру между структурой сети и динамикой, поскольку в нескольких реалистичных сценариях структура сети влияет на происходящие в ней процессы, но динамика в свою очередь может изменять и структуру.

7.3.1. Дискретные мнения

Люди иногда сталкиваются с предельным числом позиций по конкретному вопросу, часто всего с двумя позициями: справа/слева, Android/iPhone, покупка/продажа и т. д. В таких случаях мнение представляется целочисленным атрибутом или *состоянием* каждого узла. Для простоты давайте рассмотрим случай только двоичных мнений.

Модель характеризуется набором правил, которые определяют то, как меняется мнение узла в зависимости от мнений его соседей. Динамика обычно следует приведенным ниже шагам.

1. В первоначальной конфигурации мнения случайно распределяются между узлами сети. Из этого вытекает, что изначально существует примерно одинаковое число людей, придерживающихся того или иного мнения (*несогласие*).
2. Правило обновления мнений применяется снова и снова ко всем узлам. Итерация состоит из прокручивания всех узлов в цикле. В типичной ситуации узлы обновляются асинхронно в случайном порядке в целях облегчения схождения.
3. Есть два возможных исхода:
 - (i) система достигает устойчивого состояния, когда ни один узел больше не меняет своего мнения. Такое состояние бывает *консенсусом*, когда все узлы придерживаются одного и того же мнения, или *поляризацией*, когда некоторые узлы придерживаются одного мнения, а остальные – другого;
 - (ii) система не достигает стационарного состояния в том смысле, что на каждой итерации некоторые (или все) узлы продолжают менять свое мнение. Тем не менее некоторые признаки конфигурации мнений, например средние значения некоторых переменных, могут в долгосрочной перспективе стабилизироваться.

В этих моделях можно вычислять и отслеживать несколько стандартных переменных.

- *Среднее мнение* – это среднее арифметическое мнений по всем узлам. Если мы начинаем со случайного распределения двух мнений, нуля и единицы, то среднее значение составит около 0.5, так как половина узлов будет иметь мнение ноль, а другая половина – мнение один. Среднее мнение обычно меняется в динамике, и его значение можно отслеживать после каждой итерации. Если система достигает стационарного состояния, то среднее значение сходится к точному значению. Если стационарное состояние является консенсусным, то оно равно либо нулю, либо единице в зависимости от того, какое мнение доминирует.
- *Вероятность выхода* оценивает величину вероятности того, что сеть достигнет консенсуса в отношении мнения один, в зависимости от доли узлов с мнением один в начальной конфигурации.

В качестве иллюстрации предположим, что мы выполняем модельную динамику 100 раз, начиная со 100 разных случайных конфигураций. В каждой первоначальной конфигурации мы назначаем мнение один каждому узлу с вероятностью 0.4, вследствие чего приблизительно 40 % узлов будут иметь мнение один. Вообразите, что все прогоны приводят к консенсусу, 30 из которых – к консенсусному мнению один. Значение вероятности выхода для первоначальной вероятности 0.4 мнения один тогда равно $30/100 = 0.3$.

Две простые дискретные модели динамики мнений заимствованы из статистической физики: *модель на основе большинства* и *модель на основе избирателя*. В первом случае динамика основана на правиле большинства: каждый узел принимает мнение большинства своих соседей, как показано на рис. 7.11. Если число соседей является четным и имеется равное их число среди обоих мнений, то мы подбрасываем монету, чтобы решить, какое мнение будет принято узлом. Это, в сущности, эквивалентно дробно-пороговой модели, представленной в разделе 7.1.1, с порогом $1/2$. Разница заключается в интерпретации: здесь мы думаем о двух соперничающих мнениях, а не о распространении одной идеи.

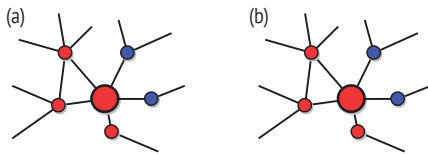


Рис. 7.11 Модель динамики мнений на основе большинства. (а) Узел, подлежащий обновлению (большая окружность), имеет мнение один (красный). Узел имеет пять соседей: у трех есть мнение один, у двух других мнение ноль (синий). (б) Узел принимает мнение большинства, поэтому он остается красным

Консенсус – это стабильное состояние, в котором все узлы придерживаются одного и того же мнения, и ничто не может измениться. Но есть и другие стабильные состояния: если узел придерживается мнения большинства своих соседей, как показано на рис. 7.11, то его мнение не изменится. Такое условие локального большинства часто достигается всеми узлами сети, порождая стабильные конфигурации, в которых сосуществуют оба мнения. В большинстве сетей, которые мы видели в этой книге, как и во всех модельных сетях главы 5, динамика большинства никогда не достигает консенсуса; сеть попадает в ловушку в состояниях с сосуществованием мнений. Консенсус достигается только на одномерных и двумерных решетках. По сути дела, на двумерной квадратной решетке консенсус достигается примерно в двух третях прогонов. Если мы вычислим вероятность выхода для прогонов, которые приводят к консенсусу, то мы получим характерный шаговый профиль, показанный на рис. 7.12(а): для достижения консенсуса по любому мнению это мнение должно иметь большинство в первоначальной конфигурации.

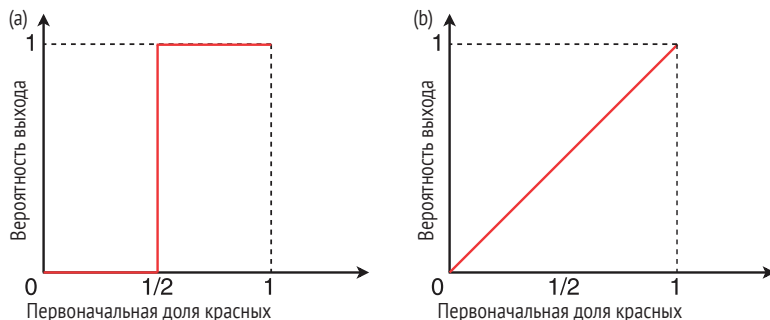


Рис. 7.12 Вероятность выхода. (а) Модель на основе большинства на решетчатой сети. Шаговая функция указывает на то, что первоначальная доля любого мнений из двух будет определять возможность достижения системой консенсуса по этому мнению: если динамика приводит к консенсусу и более половины узлов имеют мнение один (ноль) в первоначальной конфигурации, то имеется консенсус по мнению один (ноль). Эта диаграмма может быть построена только для одномерных или двумерных решеток, так как в противном случае динамика никогда не будет приводить к консенсусу. (б) Модель на основе избирателя. Диагональная функция указывает на то, что первоначальная доля мнений один также является вероятностью достижения консенсуса по мнению один. В отличие от динамики большинства в модели на основе избирателя можно достигать консенсуса по любому мнению из двух, даже если первоначально такое мнение было у менее половины узлов

В показанной на рис. 7.13 модели на основе избирателя каждый узел принимает мнение случайно выбранного соседа, каким бы он ни был. Демонстрация моделей как на основе большинства, так и на основе избирателя представлена в приложении В.6. Консенсус является единственным стабильным состоянием динамики модели на основе избирателя, поэтому он является неизбежной окончательной конфигурацией системы в любой связанной сети. На самом деле до тех пор, пока сосуществуют разные мнения, соседи с разными мнениями всегда могут влиять друг на друга. Вероятность выхода модели на основе избирателя совпадает с долей первоначальных узлов с мнением один, поэтому эта функция является диагональной на рис. 7.12(б). В отличие от модели на основе большинства здесь исход динамики является неопределенным. Например, предположим, что в первоначальной конфигурации 30 % узлов имеет мнение один. Тогда мы ожидаем, что в 30 % прогонов все узлы в итоге будут иметь мнение один, но мы не можем сказать заранее, к консенсусу по какому мнению приведет конкретный прогон: один либо ноль.

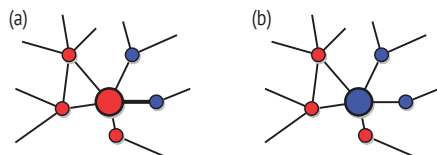


Рис. 7.13 Модель на основе избирателя. Окрестность обновляемого узла (большая окружность) такая же, как на рис. 7.11. (а) Выбирается случайный сосед (синий узел, прикрепленный к толстой связи). (б) Центральный узел принимает мнение своего соседа

Модели на основе избирателя имеют много вариаций. Ее распространенные модификации таковы:

- присутствие *фанатичных сторонников*, которые никогда не меняют своего мнения. Если все они придерживаются одного и того же мнения, то они будут поддерживать консенсус в отношении этого мнения, в противном случае консенсус никогда не будет достигнут;
- рассмотрение более двух мнений. В этом случае взаимодействие бывает ограничено только между узлами с достаточно близкими мнениями. Например, у одного может быть три мнения (1, 2 и 3), таких что взаимодействовать могут только соседние мнения (1 и 2, 2 и 3, но не 1 и 3). Мы подробно обсуждаем такой принцип в разделе 7.3.2. Неконсенсусные конфигурации с невзаимодействующими мнениями стабильны в любой сети;
- возможность узлов спонтанно менять свое мнение, например, с некоторой вероятностью на каждой итерации в зависимости от динамики избирателя.

Аналогичные модификации могут применяться и к другим дискретным моделям динамики мнений.

7.3.2. Непрерывные мнения

Существуют ситуации, в которых мнение индивидуума может плавно варьироваться от одной крайности к другой из целого ряда возможных вариантов. Например, оно может выражать восприятие произведения искусства, которое может постоянно варьироваться от неприязни (0) до энтузиазма (10). Либо мы, возможно, захотим смоделировать политический расклад по спектру от очень прогрессивного (–1) до очень консервативного (+1). В таких случаях мнения лучше представлять вещественными непрерывными числами.

Как и в моделях дискретных мнений, случайные мнения обычно назначаются сетевым узлам в первоначальной конфигурации. Это может осуществляться путем генерирования случайных чисел в нужном диапазоне. Затем значения мнений меняются по мере их многократного обновления. Если в какой-то момент наибольшее отклонение любого мнения становится меньше заранее определенного порога, то мы можем остановить симуляцию, потому что система в конечном итоге достигнет стационарного состояния. Типичными стационарными состояниями являются *консенсус*, *поляризация* либо *фрагментация* в зависимости от того, сосредоточены ли мнения соответственно вокруг одного, двух или более значений. В пределах бесконечного времени симуляции каждый узел будет иметь ровно одно из немногих сохранившихся мнений.

Мы воображаем людей, ведущих конструктивную дискуссию по какой-либо теме с возможностью влиять на мнение друг друга, в особен-

ности когда их позиции достаточно близки друг к другу. Индивидуум вряд ли сможет убедить другого, если у последнего противоположная точка зрения. Это простое наблюдение вдохновило на *принцип ограниченной уверенности*: два мнения могут влиять друг на друга только в том случае, если их разница меньше заданной величины, которая называется *границей уверенности*, или *толерантностью*.

Изначальная модель на основе ограниченной уверенности имеет правило обновления, которое состоит в выборе узла и одного из его соседей. Если их мнения расходятся меньше, чем граница уверенности, то они оба «движутся» навстречу друг другу на некоторую относительную величину, определяемую параметром схождения. В противном случае мнения не меняются.



В модели на основе ограниченной уверенности на итерации t каждый узел i имеет мнение $o_i(t)$, которое является вещественным числом, к примеру между нулем и единицей. Итерация состоит из витка по всем узлам, синхронно либо в случайном порядке. На итерации $t + 1$, когда дело доходит до узла i , мы выбираем одного из его соседей наугад, к примеру, j . Если

$$|o_i(t) - o_j(t)| < \epsilon, \quad (7.6)$$

где ϵ – это граница уверенности, то значения мнений обновляются до

$$o_i(t+1) = o_i(t) + \mu |o_j(t) - o_i(t)|, \quad (7.7)$$

$$o_j(t+1) = o_j(t) + \mu |o_i(t) - o_j(t)|, \quad (7.8)$$

где $\mu > 1$ – это параметр схождения. Если $\mu = 1/2$, то мнения сходятся к их среднему значению, имея в виду, что оба индивидуума занимают общую промежуточную позицию. Если $\mu = 1$, то мнения меняются в том смысле, что i принимает мнение узла j , и наоборот. Обычно μ колеблется вот в этом интервале $(0, 1/2]$.

Если мы сложим уравнения (7.7) и (7.8) бок о бок и разделим на два, то мы увидим, что вторые члены с правых сторон уравновешивают друг друга. Мы приходим к выводу, что среднее мнение узлов i и j одинаково до и после обновления: *среднее мнение системы сохраняется* в динамике ограниченной уверенности! Если первоначальные мнения взяты случайно из интервала $[0, 1]$, то их среднее значение равно $1/2$ (возможно, с малыми отклонениями). Таким образом, если система в конечном итоге достигнет консенсуса, то мнения всех узлов будут скапливаться вокруг $1/2$.

Начиная со случайной первоначальной конфигурации мнений, динамика всегда приводит к стационарному состоянию на любой сети.

Параметр схождения влияет только на число итераций, необходимых для достижения схождения. Число кластеров мнений в стационарном состоянии зависит от границы уверенности и структуры сети. Чем ниже граница уверенности, тем больше число кластеров окончательных мнений.

Для $\epsilon > 1/2$ система всегда достигает консенсуса на любой сети, при этом мнения центрируются вокруг $1/2$.

ЛАНЬ®

Модели на основе на ограниченной уверенности имеют целый ряд вариаций. Распространенные модификации таковы:

- использование индивидуальных значений границы уверенности, чтобы учесть тот факт, что не всех можно убедить так же легко, как и всех остальных. В некоторых расширениях граница уверенности у узла состыкована с мнением индивидуума. Например, если мнение близко к экстремумам интервала, то граница уверенности невелика, потому что убедить экстремистов труднее, чем большинство людей;
- возможность ситуаций, когда индивидуумы меняют свое мнение спонтанно. Как и в модели на основе избирателя и других моделях, это может имплементироваться путем предоставления узлам возможности изменять свое мнение с некоторой вероятностью на каждой итерации.

ЛАНЬ®

7.3.3. Козволюция сетей и динамика

В разделе 2.1 мы увидели, что ассортативность встречается во многих реально существующих графах, в особенности в социальных сетях: узлы похожи на своих соседей. Мы также обсудили два возможных механизма, которые за это ответственны: социальное влияние (соседи становятся все более похожими) и *селекция* или *гомофилия* (похожие узлы становятся соседями). Вполне вероятно, что оба механизма ответственны за наблюдаемую ассортативность. Например, если мы постоянно обсуждаем какой-либо вопрос с одним из наших знакомых, то мы можем либо попытаться найти компромисс, либо нам будет лучше, если мы будем общаться с кем-то другим, кто разделяет нашу точку зрения. Это часто происходит в социальных сетях, где люди «раздружаются» или «отписываются» от контактов с отличающимися взглядами. В обсуждавшихся до сих пор моделях динамики мнений сеть является фиксированной. Поэтому мы не допускаем селекции, потому что узлы с очень похожими мнениями не имеют возможности стать соседями, если они уже не являются таковыми. Точно так же соседи с очень несхожими мнениями не могут стать разъединенными. Узлы могут влиять только на мнения друг друга. Реалистичная модель

должна допускать взаимную игру как влияния, так и селекции. Это привело к разработке *коэволюционных моделей*, в которых изменение мнений может приводить к изменениям в структуре сети, что в свою очередь может влиять на мнения и т. д. По сути, мнения и сети *адаптируются* друг к другу.

В одной коэволюционной модели мнения дискретны и могут принимать два или более значений. В начале мнения распределяются по узлам случайно. Динамика состоит из чередующихся шагов селекции и влияния с относительной частотой, определяемой параметром. Посредством селекции узлы устанавливают связи с другими узлами с тем же мнением. Посредством влияния узлы принимают мнение своих соседей. На рис. 7.14 показаны шаги модели по селективному отбору и оказанию влияния.

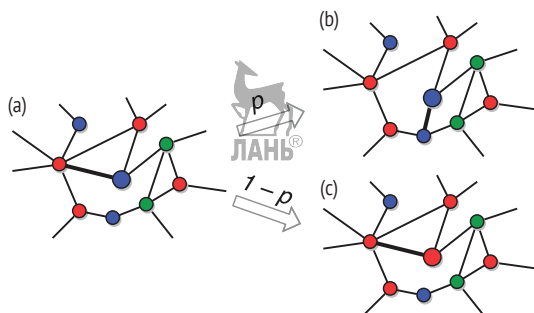


Рис. 7.14 Коэволюция мнений и сетей. Мнения обозначаются цветами. (а) Выбирается узел (большая синяя окружность посередине) вместе с одним из его соседей (красный узел, прикрепленный к толстой связи). (б) С вероятностью p узел заменяет своего соседа узлом, имеющим одинаковое мнение. (с) С вероятностью $p - 1$ узел принимает мнение соседа

Каждая итерация коэволюционной модели требует витка по узлам, синхронно либо в случайном порядке. Когда мы экзаменуем узел i , мы отбираем случайного соседа j с мнением, отличным от i :

- 1) с вероятностью p связь между i и j переподсоединяется, меняя связь с i на связь со случайно отобранным не-соседом, придерживающимся того же мнения, что и i (*селекция*);
- 2) в противном случае (с вероятностью $p - 1$) i принимает мнение j (*влияние*).

Вероятность селекции, p , является единственным параметром модели.

Поскольку и селекция, и влияние имеют тенденцию уменьшать число пар соседних узлов с разными мнениями, сеть в конечном ито-

ге достигает состояния, в котором все пары соседей придерживаются одного и того же мнения. Это означает, что сеть будет поделена на множество отдельных компонент, разьединенных друг от друга, при этом все члены каждой компоненты придерживаются одного и того же мнения, которое в разных компонентах может отличаться. Отсюда мы будем наблюдать сегрегацию на гомогенные сообщества мнений, как показано на рис. 7.15. Такой сценарий является стабильным состоянием: изменений во мнениях либо структуре сети больше не происходит, и динамика прекращается.

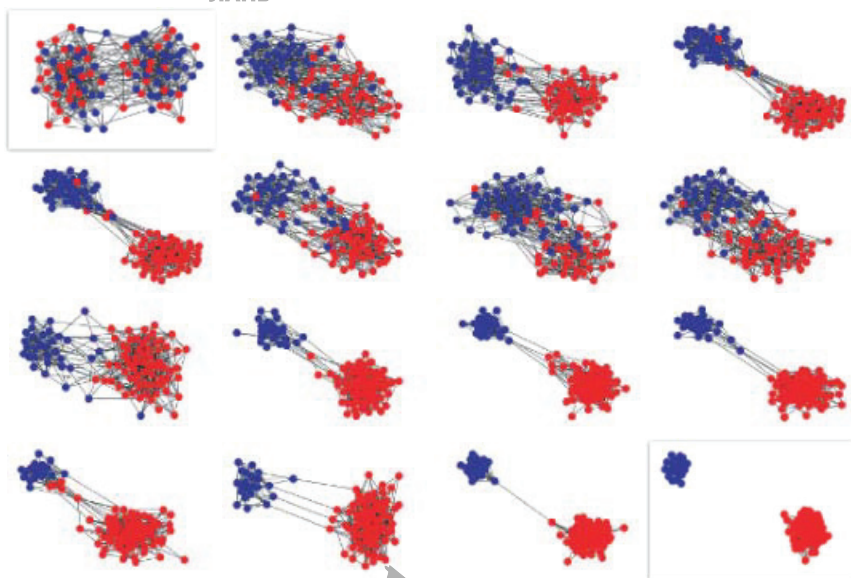


Рис. 7.15 Динамика коэволюционной модели на сети с двумя сообществами. Первоначально (вверху слева) два мнения распределяются между узлами случайно. Вероятность селекции равна $p = 0.7$. В конечном счете (внизу справа) сеть сегрегируется на две разьединенные компоненты с гомогенными мнениями

Когда вероятность селекции близка к нулю, влияние доминирует, и структура сети практически не меняется. Система в основном будет гомогенизировать мнения внутри связанных компонент первоначальной сети. Когда вероятность селекции близка к единице, селекция доминирует, и мнения практически не влияют друг на друга. Здесь окончательными компонентами системы являются группы узлов с одинаковым мнением, что и в первоначальной конфигурации.

Давайте посмотрим, что происходит, когда число мнений велико. Если мы начнем со случайной сети со средней степенью больше единицы, то мы знаем, что она имеет гигантскую компоненту (раздел 5.1), поэтому в случае вероятности селекции, близкой к нулю, в долгосрочной перспективе будет существовать гигантское сообщество, придерживающееся мнения большинства, и много малых сооб-

ществ с разными мнениями. В случае вероятности селекции, близкой к единице, вместо этого динамика связи разобьет сеть на множество мелких компонент, каждая из которых будет состоять в основном из узлов, которым первоначально было назначено одно из отличимых мнений. Оказывается, существует резкий транзит между сценарием с мнением крупного большинства и сценарием с многочисленными меньшими сообществами мнений сопоставимого размера. Этот транзит происходит в пороговом значении вероятности селекции.

Модели, в которых люди со схожими мнениями тяготеют к тому, чтобы собираться вместе, помогают нам изучать появление эхокамер в социальных медиа, как описано в разделе 4.5 и проиллюстрировано на рис. 6.2.

7.4. Поиск

Одним из наиболее распространенных видов деятельности, которые мы выполняем при взаимодействии с сетями, является *поиск*. Предположим, вы хотите отыскать какой-то ресурс, расположенный на каком-то узле сети. Это может быть веб-сайт с информацией по интересующей теме, кинофильм, хранящийся в одноранговой сети, или деловой контакт в социальной сети – в отличие от целевого человека в маломировом эксперименте Милграма (раздел 2.7). В целях решения этих задач нам необходимо разработать стратегии проведения эффективной разведки сети до тех пор, пока не будет достигнут нужный узел. Обычно такая стратегия начинается с узла происхождения и продолжается посещением соседей, соседей соседей и т. д. Чем эффективнее стратегия, тем быстрее вы сможете достичь цели. В этом разделе представлено несколько распространенных подходов к поиску. В частности, мы подчеркнем принципы использования особых свойств реально существующих сетей в целях ускорения процесса поиска.

7.4.1. Локальный поиск

Представленный в главе 2 поиск сперва в ширину является попыткой поиска по всей сети, посещая каждый узел, по меньшей мере, в связанных компонентах, где известны некоторые затравочные узлы. Этот тип подхода на основе *исчерпывающего поиска* решает задачу в некоторых случаях, в особенности когда сеть мала либо когда доступны огромные вычислительные ресурсы и ресурсы хранения, о чем свидетельствуют обходчики Всемирной паутины, поддерживающие запросы поисковых машин. Но нередко эффективнее или даже нужнее выполнять локальный поиск в сети (т. е. выполнять целенаправленные обходы по конкретным поисковым запросам, разведывая только малую часть сети). Например, вас может заинтересовать очень специ-

фический или новый веб-контент, которого нет в индексе поисковой машины. В этих случаях процесс поиска должен задействовать некие эвристические методы, сортируя узлы сети, которые будут с наибольшей вероятностью содержать требуемую информацию.

Еще один сценарий, в котором необходим локальный поиск, – это когда вы хотите скачать только что выпущенную песню из *одноранговой* сети (т. е. сети между равноправными узлами), которая представляет собой множество персональных компьютеров, соединенных напрямую друг к другу для обмена файлами. В таких системах отсутствует центральный сервер, который мог бы хранить местоположение каждого файла. Это выгодно, потому что функционирование всей системы невозможно нарушить из-за отказа какого-либо одного узла – к примеру, из-за судебного иска или атаки с отказом в обслуживании (DoS-атаки). Ее недостатком является то, что местоположение нужного файла неизвестно. Поэтому всякий раз, когда пользователи ищут файл, запросы отправляются на компьютеры других пользователей, соединенных в одноранговой сети. Если на компьютере запрошенного файла нет, то запрос переадресовывается одному или нескольким соседям и т. д.

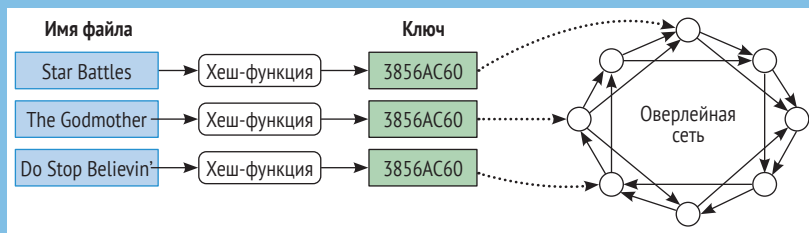
В принципе, поиск сперва в ширину тоже может использоваться для локального поиска. Начиная с источника, мы могли бы посещать все узлы первого уровня и проверять, является ли какой-либо из них целевым узлом. Если нет, то каждый из них переадресовывает запрос всем своим соседям и т. д. до тех пор, пока не будет достигнут нужный узел. Запросы, уже полученные от других соседей, игнорируются. В одной из самых ранних одноранговых сетей, именуемой Gnutella, этот подход как раз и использовался. Но стратегия поиска сперва в ширину не является эффективной. Прежде всего в ней не используются преимущества структуры сети. По сути дела, компьютеры в сети Gnutella заваливались запросами и тратили всю свою пропускную способность на управление этим трафиком. Именно поэтому Gnutella в конечном итоге была заменена современными одноранговыми сетями, такими как BitTorrent, в которых задействуются специальные сетевые структуры, разработанные для эффективных поисковых алгоритмов (вставка 7.1).

Сетевая структура эксплуатируется за счет опоры на наличие хабов. Алгоритм локального поиска, основанный на этой идее, исходит из допущения, что каждый узел знает степень всех своих соседей, а также хранящиеся в них данные, поэтому вся информация, доступная узлам, является локальной. Когда сосед целевого узла получит запрос, он ответит: «Я не тот узел, который ты ищешь, а вот мой сосед – да!» – и отправит адрес целевого узла. Каждый запрашиваемый узел, начиная с источника, перенаправляет запрос своему соседу с наибольшей степенью, если только он или любой из его соседей не является целевым. Процесс повторяется до тех пор, пока сообщение не будет получено соседом цели (рис. 7.16). Поскольку узлы во время указанной процедуры могут посещаться несколько раз, передавшие запрос узлы помечаются, вследствие чего ни один из них не запрашивается более одного раза.

Вставка 7.1

Поиск в одноранговых сетях

Одноранговые сети используются для обмена файлами и имеют структуру, разработанную для эффективного поиска совместных файлов. Это достигается за счет комбинации *распределенной хеш-таблицы*, которая соотносит файлы с одноранговыми компьютерами, и *оверлейной сети*, соединяющей эти одноранговые узлы.



Когда файл необходимо сохранить, для файла создается уникальный *ключ*. Это делается с помощью *хеш-функции*, алгоритма, который производит уникальную сигнатуру из произвольных данных. Указанный ключ соотносит с конкретным узлом в сети, вследствие чего файл может быть перемаршрутизирован на этот узел. Аналогичным образом при поиске файла ключ используется для переадресации запроса по сети до тех пор, пока он не достигнет узла, на котором находится файл с этим ключом. Каждый узел поддерживает множество связей, указывающих на своих соседей – *маршрутную таблицу*, которая используется для переадресации сообщений через оверлейную сеть. Распределенная хеш-таблица одноранговой сети, имеющей ту или иную конструкцию, кодирует правила для поддержания структуры сети таким образом, чтобы поиск был быстрым. В частности, для любого ключа каждый узел либо знает целевой узел, которому принадлежит этот ключ, либо имеет связь, указывающую на узел, который находится ближе к цели. Благодаря этому свойству можно задействовать простой алгоритм жадной маршрутизации для переадресации сообщения ближайшему к цели соседу. Еще одним важным свойством одноранговой сети является то, что любой компьютер в любое время может присоединиться к сети либо ее покинуть. Когда одноранговый узел покидает или новый одноранговый узел присоединяется, необходимо обновлять только соседние одноранговые узлы; остальная часть сети остается нетронутой.

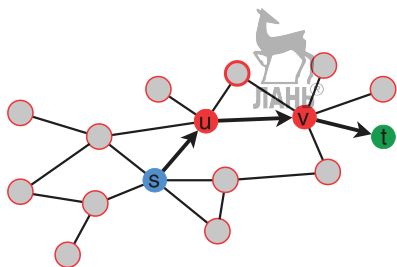


Рис. 7.16 Модель локального поиска в сетях. Источником является *s*, а целью – *t*. Источник передает запрос своему соседу с наивысшей степенью (*u*), который пересылает его своему соседу с наивысшей степенью (*v*). Поскольку цель является соседом *v*, поиск заканчивается

В разделе 3.3 мы увидели, что соседи случайно отобранного узла в среднем будут хабами с большей вероятностью, чем сам узел. В частности, при разведывании соседей с большей степенью шанс того, что любой из их соседей является **главным хабом**, выше. Следовательно, алгоритм быстро достигает узла с наибольшей степенью. После проверки верхнего хаба он помечается, и в будущем он будет избегаться. Тогда следующий хаб, скорее всего, будет вторым по степени и т. д. В сущности, после быстрой переходной фазы, в которой посещаемые узлы поступательно имеют более крупную степень, разведывание следует обратному порядку последовательности степеней сети, начиная с узла с наибольшей степенью и следуя вниз. Число запрашиваемых узлов, которые являются соседями хабов, растет очень быстро, и цель достигается за малое число шагов.

Хотя движимый хабами локальный поиск и увеличивает число шагов, необходимых для завершения поиска, число узлов, которые приходится запрашивать, в среднем примерно такое же, как и при использовании поиска сперва в ширину. Это обусловлено тем, что целевой узел в принципе может находиться где угодно, поэтому в обоих случаях требуется много проверок. Меньшее число шагов алгоритма локального поиска компенсируется тем фактом, что на каждом шаге проверяется больше соседей, поскольку узлы, проходимые во время указанной процедуры, имеют большую степень. Однако если каждый узел знает информационное содержимое своих соседей, то ему на самом деле не нужно запрашивать ни один из них, что значительно сокращает накладные расходы на обмен данными между узлами. Это требует наличия хабов, которые хранили бы огромный объем данных, что невозможно в очень крупных сетях.

7.4.2. Доступность поиска

Мы увидели несколько стратегий поиска в сетях. Но все ли сети «доступны для поиска»? Можем ли мы выполнять поиск по любому графу и ожидать результатов за достаточно короткое время? Короткий ответ – нет, но существует несколько важных исключений, которые мы обсудим далее.

В целях проведения разведки свойств *доступности поиска* в сети вспомните представленный в разделе 2.7 маломирной эксперимент Милграма. Указанный эксперимент преподает нам два урока. Во-первых, это хорошо знакомое наблюдение о том, что большинство пар людей в социальной сети соединено, как мы уже убедились, короткими цепочками знакомств. Во-вторых, люди на удивление эффективно находят эти цепочки. Это непросто: участники знали только свои контакты, а также имя и местонахождение целевого человека. Они должны были доверять своему инстинкту в выборе друга, которому следует пересылать письмо, надеясь приблизить письмо к цели. Большинство участников старалось отправлять письмо так, чтобы оно достигало

района Бостона, где проживал целевой человек, как можно быстрее. А это задействует гомофилию сети (обсуждавшуюся в разделе 2.1), в частности *географическую гомофилию*: два человека с большей вероятностью будут знать друг друга, если они живут поблизости. Тем не менее, в принципе, письмо, оказавшись в Бостоне, могло бы там задержаться надолго, передаваясь от одного человеку другому, прежде чем наконец не достигнет цели. Успешные участники использовали некую дополнительную интуицию в отношении структуры сети, чтобы отыскивать цель за несколько шагов. Они использовали разные типы гомофилии, основанные на профессиях, к примеру: адвокат, скорее всего, знает другого адвоката. Это тесно соотносится с тематическим местоположением во Всемирной паутине (раздел 4.2.5).

Существует возможность проведения анализа условий, которым сеть должна удовлетворять, чтобы быть доступной для поиска, используя эвристику, основанную на описанных выше типах гомофилии, – соединение с узлом, который географически или локально близок к цели. Давайте сначала сосредоточимся на *доступности географического поиска*. Оказывается, существуют узкие условия, которые делают сеть географически доступной для поиска. В целях иллюстрации рассмотрим специальную структуру, напоминающую маломировые сети, генерируемые моделью, рассмотренной в разделе 5.2. Мы начинаем с квадратной решетки, которая служит для целей вложения социальной сети в географическое пространство, например размещения людей на карте. Каждый узел соединен со своими ближайшими соседями, образуя решетчатую сеть. Затем мы добавляем сокращения между парами узлов решетки (рис. 7.17). В отличие от маломировой модели (рис. 5.4(b)) сокращения не соединяют пары узлов с равной вероятностью; скорее, наоборот, вероятность связи уменьшается вместе с географическим расстоянием между узлами в решетке. Это сделано для учета географической гомофилии, эмпирического наблюдения о том, что большинство отношений в реально существующих социальных сетях происходит между людьми, находящимися в географической близости друг от друга.

Давайте допустим, что каждый индивидуум точно знает географическое положение своих соседей, а также положение цели. Следовательно, каждый индивидуум может точно определить соседа, который географически находится к цели ближе всего. Для простоты давайте далее допустим, что источниковый и целевой узлы выбраны случайно и что люди следуют *алгоритму жадного поиска*, вдохновленному экспериментом Милграма: каждый узел переадресовывает сообщение по связи, которая максимально приближает его к цели. Мы можем определить *время доставки* как число раз, когда сообщение передается между узлами до тех пор, пока оно не достигнет цели. Как оказалось, время доставки является очень коротким только в том случае, если вероятность быстрого доступа падает правильным образом как функция от географического расстояния между узлами.

В случае двумерной решетки, как показано на рис. 7.17, вероятность сокращения должна уменьшаться как обратная величина квадрата расстояния. Например, связь между двумя узлами, расположенными на расстоянии двух шагов друг от друга, должна быть в четыре раза вероятнее, чем связь, соединяющая два узла, которые находятся в два раза дальше (на расстоянии четырех шагов).

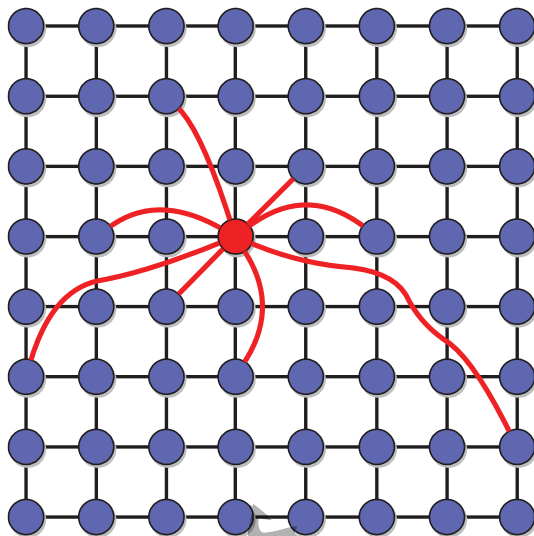


Рис. 7.17 Географическая социальная сеть. Квадратная решетка представляет географическую область, в которой живут люди (узлы). Каждый узел связан со своими четырьмя ближайшими соседями. Сокращения между узлами добавляются в пользу пар индивидуумов, живущих близко друг к другу. На рисунке показаны сокращения только красного узла

Если вероятность сокращения падает быстрее вместе с расстоянием между узлами, то не хватает дальних связей, поэтому человек обречен идти по большому числу локальных связей, прежде чем достичь цели. Если вероятность сокращения падает медленнее, то существует слишком много дальних связей. В этом сценарии существует много коротких путей, но их трудно отыскать, подобно поиску иголки в стоге сена. В обоих случаях процесс поиска не очень эффективен, и алгоритму жадного поиска требуется много времени, чтобы отыскать цель.

Хотя условие доступности географического поиска в сети в этом сценарии довольно узкое, оно не совсем нереально. Если во Всемирной паутине мы заменим понятие географической гомофилии понятием тематической локальности, то мы сможем эмпирически измерить вероятность того, что две страницы связаны как функцию от их тематического расстояния. Вообразите, что решетка на рис. 7.17 представляет тематический ландшафт, а близлежащие точки представляют узлы по теме веб-страницы. На практике мы можем

измерить сходство между двумя страницами, посмотрев на их содержимое (вспомните вставку 4.1). Малые значения сходства могут быть соотнесены с большими расстояниями, и наоборот. Оказывается, что близлежащие (похожие) страницы с высокой вероятностью будут иметь общих соседей или будут связаны, тогда как для отдаленных (непохожих) страниц снижение вероятности связи совместимо с условием доступности географического поиска. Следовательно, Всемирная паутина представляет собой особый случай сети с доступностью поиска, что обнадеживает, поскольку означает, что мы можем отыскивать интересную информацию, переходя по связям. Если бы это было не так, то навигация во Всемирной паутине была бы безнадёжной.

Сетевая модель, используемая для разведывания доступности географического поиска, во многих отношениях является нереалистичной. Люди не расположены и не соединены как узлы сети. Что еще важнее, география является лишь одним из многих возможных атрибутов узлов сети, в которой выполняется поиск. Два человека в социальной сети могут иметь одну и ту же работу, заниматься одним и тем же хобби, посещать одну и ту же школу и т. д. Давайте обобщим понятие доступности поиска на *доступность тематического поиска*, где любой атрибут узлов может отражаться в сетевой гомофилии и, таким образом, облегчать процесс поиска. Например, как упоминалось ранее, занятие целевого узла было полезной информацией в эксперименте Милграма.

Мы можем сгруппировать узлы сети иерархически на основе их тематических атрибутов: вершина иерархии представляет наиболее общую категорию, которая по мере продвижения вниз разбивается на более мелкие, все более конкретные тематические категории до тех пор, пока мы не достигнем самых малых групп, которые можем идентифицировать. Результирующая иерархическая диаграмма представляет собой *дерево тематических расстояний*, показанное на рис. 7.18. Дерево тематических расстояний может использоваться для организации статей «Википедии» о науке. На вершине (ниже корня) будут находиться формальные, физические, медико-биологические, социальные и прикладные науки. На уровне ниже мы найдем такие дисциплины, как математика, логика, биология, химия, физика, психология, экономика, социология, машиностроение, информатика и т. д. Более конкретные области, такие как молекулярная биология, статистическая физика, машинное обучение и наука о сетях, будут размещены на более низких уровнях. Тематическое дерево можно использовать аналогичным образом для классифицирования людей в социальной сети. В верхней части будет находиться население всего земного шара, а нижние группы могут представлять географическое деление населения на континенты, страны, города и районы. Разные социальные атрибуты (например, профессии, хобби, школы, религии) приводят к разным делениям и деревьям.

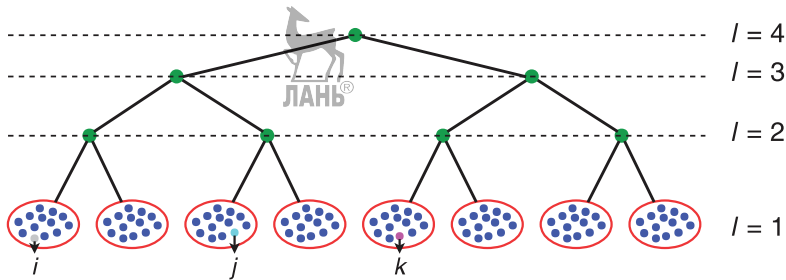


Рис. 7.18 Дерево тематических расстояний. Расстояние между узлами i и j равно трем, потому что ближайший общий предок групп, к которым принадлежат i и j , находится на третьем уровне (зеленая точка слева от пунктирной линии $l = 3$). Аналогичным образом тематическое расстояние между i и k или j и k равно $l = 4$, потому что ближайшим общим предком является корень

Дерево тематических расстояний – это мысленный конструкт, который позволяет нам оценивать *тематическое расстояние* между узлами (рис. 7.18). Если два индивидуума принадлежат к одной и той же наименьшей идентифицируемой группе, то их тематическое расстояние равно единице. Так было бы, к примеру, в случае двух профессоров, работающих на одном факультете в Университете Индианы в Блумингтоне. В противном случае их группы в конечном итоге сольются, когда мы поднимемся вверх по иерархическому дереву. Это происходит, когда мы наталкиваемся на категорию их *ближайшего предка* в дереве, которая представляет наиболее конкретный атрибут общий для узлов. В этом случае тематическое расстояние задается числом уровней в дереве снизу вверх до ближайшего общего предка. Например, на схематической диаграмме на рис. 7.18 индивидуумы i и j могут быть двумя профессорами, работающими на разных факультетах разных университетов Индианы, поэтому их тематическое расстояние равно трем, потому что работа над отдельными темами и в разных местах добавляет две степени сепарации.

Давайте придерживаться сценария социальной сети и допустим, что люди могут оценивать свое тематическое расстояние от кого бы то ни было. Это менее строгая гипотеза, чем в географической модели, где индивидуумы знают точное местоположение друг друга. Давайте далее допустим, что дерево тематических расстояний так отражает гомофилию социальной сети, что вероятность связи между двумя узлами уменьшается по мере увеличения их тематического расстояния в соответствии с функцией затухания. С помощью алгоритма жадного поиска (т. е. позволяя каждому человеку переадресовывать сообщение соседу с наименьшим тематическим расстоянием от цели) можно показать, что существует специальная функция тематического затухания, которая обеспечивает эффективный поиск. В этом случае поиск выполняется за малое число шагов.

Условие для доступности тематического поиска в сети, выраженное соотношением между тематическим расстоянием и вероятностью связи, является довольно строгим. Тем не менее оно социологически

правдоподобно и помогает нам понимать успешные цепочки в эксперименте Милграма. Кроме того, существует возможность измерять затухание вероятности того, что две веб-страницы связаны, вместе с увеличением их тематического расстояния, анализируя страницы, классифицированные в тематическом каталоге Всемирной паутины. Оказывается, что граф Всемирной паутины также соответствует условию доступности тематического поиска, подтверждая, что он доступен для поиска путем навигации.



7.5. Резюме

Сети – это средства диффузии идей, мнений и влияния. Равным образом они способствуют вредным процессам распространения, таким как диффузия инфекций, дезинформации и слухов. Раскрытие принципов, по которому эти явления разворачиваются, помогает нам повышать эффективность первого и защищаться от второго. Поиск в сетях имеет решающее значение для получения информации, но затруднен, когда структура сети и хранящееся на узлах содержимое неизвестны. В этой главе мы рассмотрели простые модели, описывающие эти процессы, и извлекли следующие ключевые уроки.

1. В пороговых моделях диффузии влияния узел/индивидуум подвержен комбинированному эффекту со стороны всех влиятелей, находящихся по соседству: когда этот эффект превышает порог, узел подвергается воздействию. В независимо-каскадных моделях узел/индивидуум «убеждается» каждым находящимся по соседству влиятелем с определенной вероятностью. Наиболее эффективные влиятели имеют высокую степень и центральное положение в сети.
2. В модели восприимчивый–инфицированный–восприимчивый (SIS) распространения эпидемии, когда инфицированные индивидуумы выздоравливают, они снова становятся восприимчивыми, поэтому они могут заразиться болезнью несколько раз. В модели восприимчивый–выздоровевший–восприимчивый (SIR), когда инфицированные индивидуумы выздоравливают, они больше не могут быть инфицированы, поэтому они не играют дальнейшей роли в динамике.
3. Если в контактных сетях есть хабы, то распространение болезни в соответствии с динамикой как SIR, так и SIS может затрагивать значительную часть популяции, даже если вероятность заражения невелика, потому что хабы могут легко заразиться и превращаться в опасных распространителей.
4. Модель распространения слуха похожа на модель SIR, но процесс «выздоровления», который соответствует решению не распространять слух дальше, является следствием встреч между индиви-

дуумами, которые о слухе знают, вместо того чтобы происходить спонтанно для каждого индивидуума. Слух может достигать значительной части любой сети, даже при низкой вероятности передачи.

5. В модели мнений на основе большинства узел принимает мнение большинства своих соседей. В конечном состоянии сосуществуют разные мнения. Консенсус достигается только на одномерных и двумерных решетках; в таких случаях консенсусное мнение является мнением большинства в первоначальной конфигурации.
6. В модели на основе избирателя узел принимает мнение случайно отобранного соседа. Динамика приводит к консенсусу во всех сетях. Консенсус по мнению достигается с вероятностью, соответствующей доле узлов, придерживающихся этого мнения в первоначальной конфигурации.
7. В моделях непрерывной динамики мнений, базирующихся на ограниченной уверенности, два мнения могут влиять друг на друга только в том случае, если их разница меньше, чем параметр ограниченной уверенности. Окончательное число кластеров мнений зависит от значения границы уверенности и структуры сети. При достаточно большой границе уверенности динамика ведет от случайных первоначальных мнений к консенсусу на любой сети.
8. Козволюционные модели сочетают процессы селекции и социального влияния. Мы представили модель, в которой узел может либо принимать мнение соседа, либо селективно отбирать нового соседа с таким же мнением. В окончательном состоянии система сегрегирована на гомогенные сообщества мнений, отсоединенные друг от друга.
9. Для исчерпывающего поиска в сети, подобного тем, которые выполняются обходчиками Всемирной паутины, стандартным подходом является поиск сперва в ширину, тот же алгоритм, который используется для вычисления расстояний и поиска кратчайших путей между узлами. Он бывает неосуществим в случае крупных сетей, вследствие чего становится необходимым локальный эвристический поиск. Одной из эвристик локального поиска является переадресация запроса на соседние узлы с наибольшей степенью, чтобы иметь возможность быстро добираться до крупнейших хабов и эксплуатировать их большое число соседей для отыскания цели за малое число шагов.
10. Некоторые сети доступны для поиска, поскольку можно отыскивать короткие пути, соединяющие источник с целью. Это бывает обусловлено особым географическим распределением связей между узлами либо иерархической организацией узлов в соответствии с их содержимым или атрибутами. Оценивая расстояние между двумя узлами в иерархии, можно определять ближайшего к цели соседа.



7.6. Дальнейшее чтение

Большинство общих книг по науке о сетях, подобных тем, которые рекомендованы в разделе 1.12, содержат обширные разделы, посвященные динамическим процессам. Книга Баррата и соавт. (2008) посвящена этой теме и подробно охватывает большинство моделей, представленных в этой главе.

Наука, стоящая за распространением дезинформации, является новой областью исследований (Лэйзер и соавт., 2018). Изучение сетей диффузии информации (Шао и соавт., 2018a) имеет решающее значение для целей оказания помощи в понимании принципов манипулирования социальными сетями, например посредством социальных ботов (Шао и соавт., 2018b).

Пороговые модели были представлены в классической статье Грановеттера (1978), тогда как независимо-каскадная модель появилась сравнительно недавно (Гольденберг и соавт., 2001). Уоттс (2002) предложил устанавливать порог на долю соседей, а не их число. Кемпе и соавт. (2003) рассмотрели проблему выявления множества влиятелей, которые могут генерировать самые большие каскады. Кицак и соавт. (2010) показали, что хабы не обязательно являются наиболее эффективными источниками влияния. Чентола и Мэйси (2007) провели разведывательный анализ сложного заражения в распространении коллективного поведения. Венг и соавт. (2013b) показали, что сообщества влияют на вирусное распространение мемов в социальных медиа, и что размеры каскадов можно предсказывать, основываясь на числе сообществ, которые вовлечены на ранних стадиях распространения.

Книга Андерсона и Мэй (1992) является хорошим справочником для классического моделирования эпидемий. Пастор-Саторрас и соавт. (2015) опубликовали всеобъемлющий обзор эпидемических процессов в сетях. Штеле и соавт. (2011) реконструировали сеть личных взаимодействий между детьми и учителями в школе посредством устройств радиочастотной идентификации. Отсутствие эпидемического порога на сетях с хабами было впервые выявлено Пастором-Саторрасом и Веспиньяни (2001). Коэн и соавт. (2003) продемонстрировали, что иммунизирование знакомств случайно отбираемых индивидуумов является эффективной стратегией, если контактные сети имеют тяжелохвостное степенное распределение. Кристакис и Фаулер (2010) показали, что мониторинг друзей случайно отбираемых индивидуумов позволяет выявлять эпидемические вспышки на ранней стадии. Модель распространения слуха была впервые представлена Дейли и Кендаллом (1964).

Кастеллано и соавт. (2009) провели обзор динамики мнений и другие модели социальной динамики с точки зрения статистической физики. Модель на основе большинства была впервые введена в контексте спиновых моделей в статистической физике (Глаубер, 1963). Еще одна модель, основанная на концепции большинства, не обсуждае-

мая в этой главе, называется моделью правила большинства (Галам, 2002; Крапивский и Реднер, 2003). Модель на основе избирателя была предложена для описания территориальной конкуренции между видами (Клиффорд и Садбери, 1973). Мобилия и соавт. (2007) изучили роль фанатичных сторонников в модели на основе избирателя.

Васкес и соавтр. (2003b) разработали модель ограниченного избирателя, в которой могут взаимодействовать только схожие мнения. Принцип ограниченной уверенности восходит к теории социального сравнения Фестингера (1954). Изначальная модель мнений, базирующаяся на ограниченной уверенности, была представлена Деффуантом и соавт. (2000). Первые модели коэволюции сетевой динамики и структуры были предложены Холмом и Ньюманом (2006) и Гилом и Занеттом (2006).

Адамик и соавт. (2001) предложили стратегию локального поиска, которая основывается на наличии хабов в сети. Географическая сеть и сравнительный анализ доступности поиска в сети были представлены Клейнбергом (2000). Анализ доступности поиска на основе тематических иерархий и расстояний был независимо предложен Клейнбергом (2002) и Уоттсом и соавт. (2002). Менцер (2002) показал, что граф Всемирной паутины удовлетворяет версиям доступности географического и тематического поиска.

Упражнения

- 7.1 Ознакомьтесь с учебным материалом главы 7 в репозитории книги на GitHub¹. Он предлагает занятие, которое упрощает кодирование и выполнение симуляций моделей сетевой динамики.
- 7.2 Рассмотрите пример на рис. 7.19. Будет ли, в соответствии с линейной пороговой моделью, активирован узел 1, если его порог равен 4? Что, если он равен 5? Изменятся ли ответы на эти вопросы, если мы изменим веса связей, соединяющих узел 1 с его неактивными соседями?

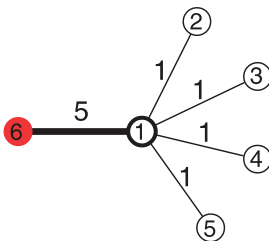


Рис. 7.19 Взвешенная сеть влияния. Узел 1 имеет только одного активного соседа (6)

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

- 7.3 Кто-то дает вам сеть, в которой несколько ее узлов активированы. Он утверждает, что вам никогда не удастся активировать все узлы, независимо от модели распространения влияния, которую вы используете. На чем основывается такая его уверенность?
- 7.4 Примените дробно-пороговую модель к сети на рис. 7.20. Порог равен $1/2$ для всех узлов. Какой узел мы должны активировать, чтобы получить самый крупный каскад? Является ли это решение уникальным? Какое минимальное число первоначальных влиятелей необходимо для активации всей сети?

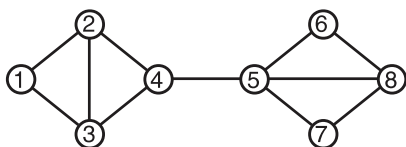


Рис. 7.20 Сеть влияния. Каждый узел имеет порог $1/2$

- 7.5 Вы рассматриваете независимо-каскадную модель на сети. Два активных узла s и t имеют соответственно степень 4 и 10. Они могут убедить своих соседей с вероятностью $1/2$ (s) и $1/5$ (t). Какой узел в среднем будет влиять на большее число соседей, s или t ?
- 7.6 Рассмотрите сеть на рис. 7.21: вероятности влияния являются симметричными; например, вероятность того, что узел 1 убедит узел 2, равна вероятности того, что узел 2 убедит узел 1. Используйте независимо-каскадную модель, чтобы предсказать число узлов, которые будут активными в конце, в среднем, путем активирования узла 2 в самом начале.

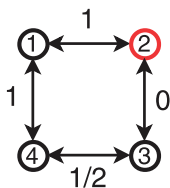


Рис. 7.21 Сеть с симметричными вероятностями влияния, показанная рядом со связями. Узел 2 является активным

- 7.7 Модели заражения, такие как SIS и SIR, взяты из эпидемиологии, но, как оказалось, они могут довольно хорошо моделировать другие процессы распространения на сетях. Какой из следующих ниже процессов лучше всего может быть описан моделью SIS на сети?
- Распространение токсичного газа по воздуху в географическом регионе.

- b. Распространение нефтяного пятна по поверхности водоема.
 - c. Последствия сбоя электростанции в энергосистеме США.
 - d. Принятие конкретного смартфона членами сообщества.
- 7.8 Игра «Пандемия II» (pandemic2.org) основана на тщательно проработанной модели SIR. Поиграйте в игру и напишите краткий отчет о том, как различные аспекты игры соответствуют механизмам модели SIR. Обсудите принятые в игре ключевые упрощающие допущения. Опишите, как различные варианты игры влияют на модельные параметры.
- 7.9 Рассмотрите динамику модели SIS на популяции. Предположим, что доля f популяции никогда не болеет и что такие иммунно-стойкие индивидуумы случайно распределены в однородной контактной сети (все узлы имеют одинаковую степень). Будет ли риск распространения эпидемии больше или меньше, чем в чистой модели SIS, для которой $f = 0$? Изменится ли ответ, если вместо нее мы рассмотрим модель SIR? (Подсказка: используйте условие уравнивания (7.5).)
- 7.10 Происходит вспышка эпидемии, и после быстрого подтверждения выясняется, что базовое репродукционное число равно $R_0 = 2.5$, поэтому мы движемся в сторону распространения эпидемии (допустим, что контактная сеть является гомогенной). Власти призывают население ограничивать свои контакты с другими людьми, чтобы в среднем каждый индивидуум общался примерно с половиной обычного числа людей. Предположим, что врачи способны разрабатывать лекарства, которые могут значительно увеличивать частоту выздоровления μ . На сколько должна увеличиться μ , чтобы можно было остановить эпидемию?
- 7.11 Просимулируйте динамику SIR в случайной сети с $N = 1000$ узлами и вероятностью связи $p = 0.01$. Первоначально заражаются 10 случайно выбранных узлов. Вероятность выздоровления составляет $\mu = 0.5$. Выполните динамику для вот этих значений вероятности заражения: $\beta = 0.02, 0.05, 0.1, 0.2$. При каждом прогоне сохраняйте число одновременно инфицированных людей после каждой итерации и вычисляйте максимальное значение. Проинтерпретируйте результаты. Сколько итераций необходимо для достижения максимума? Наблюдаете ли вы крупную вспышку заболевания? Почему да, или почему нет? (Подсказка: не стесняйтесь вносить изменения в исходный код учебного материала этой главы, чтобы выполнить симуляцию.)
- 7.12 В сообществе есть три типа людей: разочарованные (S), агрессивные (I) и мирные (R). Когда разочарованный индивидуум встречается с агрессивным, он становится агрессивным с веро-

ятностью β . Когда агрессивный индивидиум встречается с мирным, он становится мирным с вероятностью α . Когда два агрессивных индивидуума встречаются друг с другом, они начинают спорить. Но с вероятностью α они через некоторое время понимают, что борьба бесполезна, и поэтому они оба становятся мирными. Можно ли предотвратить значительное распространение агрессивного поведения малым значением β ?

- 7.13** В сети, показанной на рис. 7.22, каждый узел имеет одно из двух возможных мнений. Активная связь соединяет узлы в разных состояниях мнения. Эти связи называются активными, потому что теоретически любая конечная точка имеет шанс убедить другую принять ее мнение, в зависимости от конкретных правил модели. Сколько здесь активных связей?

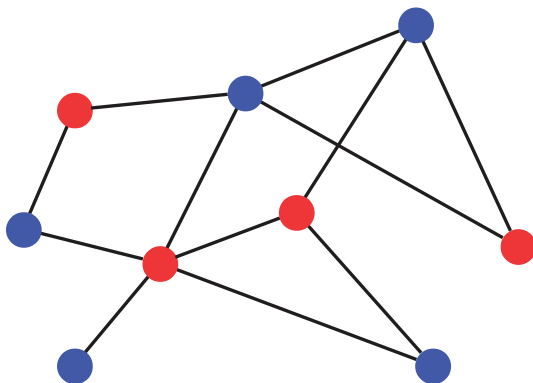


Рис. 7.22 Сеть с узлами, окрашенными в красный либо синий цвет в соответствии с их мнением

- 7.14** Во время симулирования динамического процесса на сети существует несколько способов асинхронного подбора следующего обновляемого узла (узлов). В типичной ситуации узлы отбираются в случайной последовательности. Еще одной стратегией был бы отбор одной конечной точки случайно отобранной связи. Как вы думаете, повлияет ли это каким-либо образом на динамику? Почему да или почему нет?
- 7.15** Просимулируйте динамику мнений большинства на квадратной решетке с $N = 20 \times 20 = 400$ узлами. Первоначально назначьте каждое из двух мнений половине выбранных случайно узлов. Исполняйте 100 прогонов с разными первоначальными случайными назначениями до тех пор, пока система не перейдет в стационарное состояние. Сколько прогонов приведет к консенсусу? Постройте гистограмму доли мнения один в неконсенсусных стационарных состояниях. Постройте гистограмму доли активных связей в этих конфигурациях. (Активные связи определе-

ны в упражнении 7.13. Доля активных связей – это соотношение между числом активных связей и суммарным числом связей в сети.) (Подсказка: если для выполнения симуляций вы используете исходный код из учебного материала этой главы, чтобы гарантировать схождение к стационарному состоянию, то вам необходимо написать функцию `state_transition()` таким образом, чтобы узлы обновлялись асинхронно и в случайном порядке. Вам также необходимо задать функцию условия остановки, чтобы завершать симуляцию при достижении стационарного состояния.)

- 7.16** Вычислите вероятность выхода в модели мнений на основе большинства на квадратной решетке с $N = 20 \times 20 = 400$ узлами. Пусть первоначальная доля узлов с мнением один равна $p = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.9$. Исполняйте 20 прогонов с разными первоначальными случайными величинами для каждого значения p до тех пор, пока система не перейдет в стационарное состояние. Рассматривайте только те прогоны, которые ведут к консенсусу, и для каждого p вычисляйте долю тех прогонов, для которых консенсус равняется мнению один, которая является вероятностью выхода для этого значения p . Постройте график результата как функцию от p . (Подсказка: не стесняйтесь вносить изменения в исходный код учебного материала этой главы для выполнения симуляций, как описано в предыдущем упражнении.)
- 7.17** Вычислите и постройте график вероятности выхода модели на основе избирателя на квадратной решетке. Используйте те же параметры, что и в упражнении 7.16. (Подсказка: не стесняйтесь вносить изменения в исходный код учебного материала этой главы для выполнения симуляций, как описано в предыдущих упражнениях.)
- 7.18** Рассмотрите модель динамики мнений с ограниченной уверенностью на полной сети. Поскольку все узлы соединены друг с другом, любые два узла могут влиять на мнение друг друга, если они находятся достаточно близко. Математическая аргументация показывает, что если первоначальные мнения случайно распределены в интервале $[0, 1]$, то число окончательных кластеров мнений в этом случае примерно равно $\frac{1}{2\epsilon}$, где ϵ – это граница уверенности. Если вы склонны к математике, то сможете ли дать интуитивно понятное объяснение этой аргументации?
- 7.19** Просимулируйте модель динамики мнений с ограниченной уверенностью на полной сети с $N = 1000$ узлами. Первоначальные мнения – это случайные числа от нуля до единицы. Рассмотрите три разных значения границы уверенности: $\epsilon = 0.125, 0.25, 0.5$.

Для каждого используйте разные значения параметра схождения, к примеру $\mu = 0.1, 0.3, 0.5$. Выполняйте каждую симуляцию до тех пор, пока каждое мнение не будет варьироваться менее чем на 1 % между поочередными итерациями, и постройте гистограмму окончательных мнений. Зависит ли от ϵ число окончательных кластеров мнений? Почему да, или почему нет? Зависит ли оно от μ ? Почему да, или почему нет? (Подсказка: не стесняйтесь вносить изменения в исходный код учебного материала этой главы для выполнения симуляций.)

- 7.20** Просимулируйте модель динамики мнений с ограниченной уверенностью на случайной сети с $N = 1000$ узлами и вероятностью связи $p = 0.01$. Первоначальная конфигурация мнений генерируется путем назначения каждому узлу случайного числа между нулем и единицей. Установите параметр $\mu = 1/2$ и разведайте разные значения границы уверенности ϵ . Выполняйте каждую симуляцию до тех пор, пока каждое мнение не будет изменяться менее чем на 1 % между поочередными итерациями. Каков порог ϵ_c , такой что для $\epsilon > \epsilon_c$ мы имеем единый кластер мнений (консенсус) в окончательной конфигурации? Теперь просимулируйте модель на маломировой сети с $N = 1000$ узлами, $k = 4$, и вероятностью переподсоединения $p = 0.01$. Какова ϵ_c в данном случае? (Подсказка: не стесняйтесь вносить изменения в исходный код учебного материала этой главы для выполнения симуляций.)
- 7.21** Рассмотрите коэволюционную модель всего с двумя мнениями, изначально распределенными случайно между узлами. Сколько, по вашему мнению, будет сообществ мнений, когда будет доминировать селекция (p находится близко к 1)? Каков их размер, приближенно? (Подсказка: можно допустить, что сеть является не слишком разреженной.)
- 7.22** В коэволюционной модели компонента влияния следует правило модели на основе избирателя в том смысле, что узел принимает мнение случайного соседа. Давайте посмотрим, что произойдет, если мы перейдем к динамике большинства. Новая модель работает следующим образом: для данного узла с вероятностью p она переподсоединяет одну из своих связей к несоседнему узлу с тем же мнением, что и раньше; с вероятностью $p - 1$ она принимает мнение большинства в своей окрестности. Опишите окончательные конфигурации, которые вы ожидаете наблюдать, когда система достигнет стабильного состояния в экстремальных случаях вероятности p , близких к нулю и единице.
- 7.23** Постройте маломировые сети с $N = 1000$ узлами, $k = 4$, и вот эти значения вероятности переподсоединения: $p = 0.001, 0.01$,

0.1, 1. Выберите случайно источниковый узел s и целевой узел t . Примените алгоритм жадного поиска, при котором сообщение передается ближайшему к цели соседу по кольцу, и вычислите число шагов, необходимых для доставки сообщения из s в t для каждого значения p . Проинтерпретируйте результаты. (Подсказка: для каждого p усредняйте свои измерения по нескольким прогонам с разными случайными парами узлов.)

- 7.24 Дерево тематических расстояний на рис. 7.18 очень стилизовано и нереалистично. Реальные деревья тематических расстояний, как правило, являются асимметричными, как показано на рис. 7.23. Каково тематическое расстояние между двумя отмеченными индивидуумами?

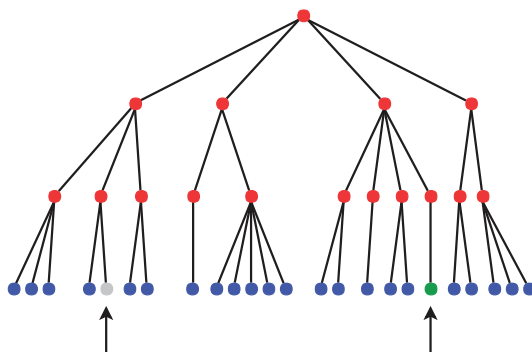


Рис. 7.23 Дерево тематических расстояний



Приложение А. Руководство по языку Python



В этом учебном руководстве демонстрируются возможности языка программирования Python и блокнотов среды Jupyter Notebook, которые используются в примерах и учебных пособиях учебника. Особое внимание уделяется наиболее важным типам данных, используемым в рабочих процессах анализа данных, а также распространенным идиомам и шаблонам, применяемым в этих вариантах использования. Настоящее приложение к книге может быть особенно полезно для читателей, более опытных в языках программирования, отличных от Python.

Данное учебное руководство, а также учебные материалы по каждой главе доступны в виде блокнотов среды Jupyter Notebook в репозитории книги на GitHub¹.

А.1. Блокнот Jupyter

Jupyter Notebook — это веб-приложение с открытым исходным кодом. Существуют бесплатные среды Jupyter, которые не требуют настройки и работают полностью в облаке, такие как блокноты среды совместной разработки Google Colaboratory². Вы также можете установить Jupyter Notebook вместе с Anaconda³, бесплатным дистрибутивом, включающим в свой состав язык Python, среду Jupyter, библиотеку NetworkX и другие часто используемые пакеты для научных вычислений и обработки данных.

Даже те, кто хорошо разбирается в языке Python, возможно, раньше не пользовались приложением Jupyter Notebook. Его главная идея заключается в возможности смешивать текст и исходный код, подобно презентации в данной книге, и исполнять этот исходный код в «ячейках». Нажимая на ячейке и кликая сочетания клавиш **Shift+Enter**, вы исполняете ячейку и переходите к следующей ячейке. Сочетание **Ctrl+Enter** исполняет ячейку, но не переходит к следующей ячейке. Вы можете исполнять несколько ячеек одновременно, используя различные опции в меню **Ячейка** приложения Jupyter.

Представленные в этом приложении фрагменты исходного кода организованы как ячейки приложения Jupyter Notebook. Каждый

¹ См. github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

² См. colab.research.google.com.

³ См. www.anaconda.com.

выделенный раздел является ячейкой; результат исполнения, если таковой имеется, исходного кода ячейки печатается под серым разделителем, как показано в приведенном ниже простом примере:

```
print('Привет из Jupyter')
Привет из Jupyter
```

В блокнотах Jupyter имеется два разных способа инспектирования переменных. Как всегда, полезна функция `print()` языка Python:

```
my_str = 'Привет'
my_int = 16
```

```
print(my_str)
print(my_int)
```

```
Привет
16
```



Мы также можем просто исполнить ячейку с именем переменной:

```
my_str
'Привет'
```

Вся разница между двумя подходами заключается в том, что инструкции `print()` могут выводить в ячейке несколько элементов, тогда как последний подход будет показывать только последнюю названную переменную. Обратите внимание:

```
my_str
my_int
16
```

В отличие от первого примера с использованием `print()` здесь выводится только последнее значение.

А.2. Условный блок

Причудливое словосочетание «условный блок» предназначено для инструкции `if`. Если вы когда-либо занимались программированием, то вы, несомненно, знаете о конструкции `if-then-else` (если-то-иначе). В языке Python она пишется следующим образом:



```
number_of_apples = 5

if number_of_apples < 1:
    print('У тебя нет яблок')
elif number_of_apples == 1:
    print('У тебя есть одно яблоко')
elif number_of_apples < 4:
    print('У тебя есть несколько яблок')
else:
    print('У тебя много яблок!')

У тебя много яблок!
```

Вы можете изменить `number_of_apples` и исполнить приведенную выше ячейку повторно, чтобы получить на выходе другой возможный результат.

А.3. Списки

Одним из наиболее универсальных и распространенных типов данных языка Python является *список* (list). Это *упорядоченная, мутируемая* коллекция *не уникальных* элементов данных.

Под упорядочением мы подразумеваем, что элементы адресуются по их *индексу* в коллекции:

```
student_names = ['Алиса', 'Боб', 'Кэрол', 'Дейв']
student_names[1]

'Боб'
```

Индексы в Python начинаются с нуля, поэтому голова списка имеет индекс 0:

```
student_names[0]

'Алиса'
```

Получить последний элемент в списке можно, используя отрицательную индексацию:

```
student_names[-1]

'Дейв'
```

Списки также можно *разрезать*, чтобы получать подмножество элементов списка:

```
student_names[0:2]
```

```
['Алиса', 'Боб']
```

```
student_names[1:3]
```

```
['Боб', 'Кэрол']
```

При нарезке от начала либо до конца списка индекс можно опускать:

```
student_names[:2]
```

```
['Алиса', 'Боб']
```

```
student_names[2:]
```

```
['Кэрол', 'Дейв']
```

Под мутируемостью мы подразумеваем, что список может быть изменен путем добавления или удаления элементов. Элементы чаще всего добавляются в конец списка, используя для этого метод `.append()`:

```
student_names.append('Esther')
```

```
student_names
```

```
['Алиса', 'Боб', 'Кэрол', 'Дейв', 'Эстер']
```

Но можно добавлять элементы и в любой произвольный индекс, используя для этого метод `.insert()`:

```
student_names.insert(2, 'Xavier')
```

```
student_names
```

```
['Алиса', 'Боб', 'Ксавье', 'Кэрол', 'Дейв', 'Эстер']
```

Удалять элементы можно с помощью ключевого слова `del`:

```
del student_names[2]
```

```
student_names
```

```
['Алиса', 'Боб', 'Кэрол', 'Дейв', 'Эстер']
```

Элементы списка необязательно должны быть уникальными. Ничто не мешает нам добавлять одно и то же имя в этот список неоднократно:

```
student_names.append('Esther')
```

```
student_names
```

```
['Алиса', 'Боб', 'Кэрол', 'Дейв', 'Эстер', 'Эстер']
```

Если вам нужна *коллекция*, в которой обеспечивается уникальность, то следует обратиться к словарям либо множествам.

Коллекция относится к типу данных, состоящему из более одного значения. Списки – это всего лишь один из видов коллекций, но есть и другие, такие как кортежи, множества и словари.

При именовании переменных, содержащих списки, полезно использовать существительные во множественном числе, такие как `student_names` (имена студентов) в приведенном выше примере. С другой стороны, одиночные значения следует именовать существительными единственного числа, как `my_str` в первом разделе. Это помогает вам и другим пользователям, читающим ваш исходный код, четко определять, какие переменные являются коллекциями, а какие – отдельными элементами, а также помогает при написании циклов, как показано в следующем далее разделе.

А.4. Циклы

Если вы пришли из другого языка программирования, то вы, вероятно, знаете о более чем одном типе цикла. В языке Python мы особо сосредотачиваемся на одном типе цикла: цикле `for`. Цикл `for` перебирает коллекцию элементов, исполняя свой исходный код для каждого элемента:

```
student_names = ['Алиса', 'Боб', 'Кэрол', 'Дейв']
```

```
for student_name in student_names:  
    print('Привет ' + student_name + '!')
```

```
Привет Алиса!  
Привет Боб!  
Привет Кэрол!  
Привет Дейв!
```

Обратите внимание на правило именования, используемое в конструкции `for-in`:

```
for student_name in student_names:
```

Используя существительное во множественном числе для коллекции `student_names`, мы автоматически получаем хорошее имя для каждого отдельного элемента в коллекции: `student_name`. Указанное правило именования используется везде, где это возможно, в учебных материалах данной книги, поскольку читателю становится ясно, какая именно переменная является «переменной цикла», которая меняет свое значение между итерациями тела цикла.

Распространенным типом программных задач при работе с данными является *задача фильтрации*. В абстрактном виде эта задача предусматривает прокручивание коллекции в цикле, проверку каждого элемента на наличие некоторого критерия, а затем добавление элементов, удовлетворяющих этому критерию, в еще одну коллекцию.

В следующем ниже примере мы создадим из списка `student_names` список только «длинных» имен. Длинные имена – это имена, содержащие более четырех символов. В учебных материалах этой книги вы часто будете встречать и писать исходный код, который выглядит следующим образом:

```
# Инициализировать пустой список и добавьте в него
# имена студентов, содержащие более четырех символов
long_names = []
for student_name in student_names:
    # Это наш критерий
    if len(student_name) > 4:
        long_names.append(student_name)

long_names
['Алиса', 'Кэрол']
```

Циклы могут быть «вложенными» друг в друга. Это нередко происходит в ситуациях, когда мы хотим сопоставить элементы из одной коллекции с элементами из той же или другой коллекции. Давайте создадим список всех возможных пар студентов:

```
student_names = ['Алиса', 'Боб', 'Кэрол', 'Дэйв']

student_pairs = []
for student_name_0 in student_names:
    for student_name_1 in student_names:
        student_pairs.append(
            (student_name_0, student_name_1)
        )

student_pairs
[('Алиса', 'Алиса'),
 ('Алиса', 'Боб'),
 ('Алиса', 'Кэрол'),
 ('Алиса', 'Дэйв'),
 ('Боб', 'Алиса'),
 ('Боб', 'Боб'),
 ('Боб', 'Кэрол'),
 ('Боб', 'Дэйв'),
 ('Кэрол', 'Алиса'),
 ('Кэрол', 'Боб'),
```



```
('Кэрол', 'Кэрол'),  
( 'Кэрол', 'Дэйв'),  
( 'Дэйв', 'Алиса'),  
( 'Дэйв', 'Боб'),  
( 'Дэйв', 'Кэрол'),  
( 'Dave', 'Dave')]
```



Обратите внимание, что вместо простого добавления имен в список `student_pairs` мы добавляем кортежи (`student_name_0`, `student_name_1`). Это означает, что каждый элемент в списке состоит из двух кортежей:

```
student_pairs[0]  
( 'Алиса', 'Алиса')
```

Подробнее о кортежах мы поговорим в следующем разделе. Следует обратить внимание на еще один момент, а именно, то, что мы включаем пары с одним и тем же студентом. Предположим, мы хотим их исключить. Это можно сделать, добавив инструкцию `if` во второй цикл `for`, чтобы отфильтровать эти повторы:

```
student_names = ['Алиса', 'Боб', 'Кэрол', 'Дейв']
```

```
student_pairs = []  
for student_name_0 in student_names:  
    for student_name_1 in student_names:  
        # Это критерий, который мы добавили  
        if student_name_0 != student_name_1:  
            student_pairs.append(  
                (student_name_0, student_name_1)  
            )
```

```
student_pairs
```

```
[('Алиса', 'Боб'),  
( 'Алиса', 'Кэрол'),  
( 'Алиса', 'Дейв'),  
( 'Боб', 'Алиса'),  
( 'Боб', 'Кэрол'),  
( 'Боб', 'Дейв'),  
( 'Кэрол', 'Алиса'),  
( 'Кэрол', 'Боб'),  
( 'Кэрол', 'Дейв'),  
( 'Дейв', 'Алиса'),  
( 'Дейв', 'Боб'),  
( 'Дейв', 'Кэрол')]
```



И теперь в списке нет повторов.

А.5. Кортежи

Даже опытные пользователи Python нередко путают разницу между кортежами и списками, поэтому обязательно прочитайте этот короткий раздел, даже если у вас есть некоторый опыт.

Кортежи внешне похожи на списки, поскольку они представляют собой упорядоченные коллекции неуникальных элементов:

```
student_grade = ('Алиса', 'испанский язык', '5-')
student_grade
('Алиса', 'испанский язык', '5-')
```

```
student_grade[0]
'Алиса'
```

Вся разница в том, что, в отличие от списков, кортежи *не мутируемы*. Каждая следующая ниже ячейка должна вызывать исключение:

```
student_grade.append('IU Блумингтон')
Traceback (most recent call last):
  <ipython-input-24-782d93a0b0cf> in <module>()
    ----> 1 student_grade.append('IU Блумингтон')

AttributeError: 'tuple' object has no attribute 'append'
```

```
del student_grade[2]
Traceback (most recent call last):
  <ipython-input-25-f8ded3b186ff> in <module>()
    ----> 1 del student_grade[2]

TypeError: 'tuple' object doesn't support item deletion
```

```
student_grade[2] = '3'
Traceback (most recent call last):
  <ipython-input-26-c9fd9c464431> in <module>()
    ----> 1 student_grade[2] = '3'

TypeError: 'tuple' object does not support item assignment
```

Указанная немутуируемость делает кортежи полезными, когда индексы имеют важное значение. В этом примере индекс имеет семантический смысл: индекс 0 – это имя студента, индекс 1 – название курса, а индекс 2 – его курсовая оценка. Невозможность вставлять или

добавлять элементы в кортеж означает, что мы уверены, что, к примеру, название курса не переместится в другую позицию.

Немутируемость кортежей делает их полезными и для *распаковки*. В простейшем случае распаковка кортежей позволяет выполнять следующее:

```
student_grade = ('Алиса', 'испанский язык', '5-')
student_name, subject, grade = student_grade

print(student_name)
print(subject)
print(grade)
```

```
Алиса
испанский язык
5-
```

Распаковка кортежей наиболее полезна при использовании с циклами. Рассмотрим следующий ниже фрагмент кода, который поздравляет студентов с получением хороших оценок:

```
student_grades = [
    ('Алиса', 'Испанский язык', '5'),
    ('Боб', 'Французский язык', '3'),
    ('Кэрл', 'Итальянский язык', '4+'),
    ('Дейв', 'Итальянский язык', '5-'),
]

for student_name, subject, grade in student_grades:
    if grade.startswith('A'):
        print(student_name, ', поздравляем тебя',
              'с получением оценки', grade,
              'по предмету', '', subject, '')
```

```
Алиса, поздравляем тебя с получением оценки 5 по предмету "испанский язык"
Дейв, поздравляем тебя с получением оценки 5- по предмету "итальянский язык"
```

Сравните это с тем же исходным кодом, используя индексы:

```
for student_grade in student_grades:
    if student_grade[2].startswith('A'):
        print(student_grade[0], ', поздравляем тебя',
              'с получением оценки', student_grade[2],
              'по предмету', '', student_grade[1], '')
```

```
Алиса, поздравляем тебя с получением оценки 5 по предмету "испанский язык"
Дейв, поздравляем тебя с получением оценки 5- по предмету "итальянский язык"
```

Распаковка кортежей позволяет нам легко обращаться к этим структурированным данным по семантическим именам вместо того, что-

бы все время держать индексы в своей памяти. Второй пример, хотя и функционально идентичен, сложнее писать и еще труднее читать.



А.6. Словари

Следующий тип коллекции отличается от двух предыдущих, но является одним из самых мощных инструментов в языке Python: *словарь*. Словарь представляет собой *неупорядоченную, мутируемую* коллекцию *уникальных* элементов. В других языках программирования словари называются *таблицами соответствия, сопоставлениями, хеш-таблицами* или *ассоциативными массивами*.

Под *неупорядоченностью* мы подразумеваем, что к элементам словаря не обращаются по их положению или индексу в коллекции. Вместо этого элементы словаря имеют ключи, каждый из которых ассоциирован со значением. Вот очень простой пример:

```
foreign_languages = {  
    'Алиса': 'испанский язык',  
    'Боб': 'французский язык',  
    'Кэрол': 'итальянский язык',  
    'Дейв': 'итальянский язык',  
}
```

Здесь имена студентов являются ключами, а языки – значениями. Поэтому, чтобы увидеть язык Кэрол, вместо индекса мы используем ключ – ее имя:

```
foreign_languages['Кэрол']  
'Итальянский язык'
```

Попытка получить значение для ключа, которого в словаре нет, приводит к ошибке ключа:

```
foreign_languages['Зик']  
Traceback (most recent call last):  
  <ipython-input-32-1ff8fc89736a> in <module>()  
----> 1 foreign_languages['Зик']  
  
KeyError: 'Зик'
```

Проверить наличие конкретного ключа в словаре можно с помощью ключевого слова `in`:

```
'Зик' in foreign_languages
```

```
False
```

```
'Алиса' in foreign_languages
```

```
True
```

Обратите внимание, что ключи чувствительны к регистру:

```
'алиса' in foreign_languages
```

```
False
```

Записи в словаре можно добавлять, удалять и изменять:

```
# Добавить несуществующую запись  
foreign_languages['Эстер'] = 'Французский язык'
```

```
foreign_languages
```

```
{'Алиса': 'испанский язык',  
'Боб': 'французский язык',  
'Кэрол': 'итальянский язык',  
'Дейв': 'итальянский язык',  
'Эстер': 'французский язык'}
```

```
# Удалить существующую запись
```

```
del foreign_languages['Боб']
```

```
foreign_languages
```

```
{'Алиса': 'испанский язык',  
'Кэрол': 'итальянский язык',  
'Дейв': 'итальянский язык',  
'Эстер': 'французский язык'}
```

```
# Изменить существующую запись
```

```
foreign_languages['Эстер'] = 'Итальянский язык'
```

```
foreign_languages
```

```
{'Алиса': 'испанский язык',  
'Кэрол': 'итальянский язык',  
'Дейв': 'итальянский язык',  
'Эстер': 'итальянский язык'}
```

Обратите внимание, что синтаксис добавления несуществующей записи и изменения существующей записи одинаков. Присвоение значения ключу в словаре добавляет ключ, если он не существует, либо обновляет значение ключа, если он существует. Как следствие,

ключи с неизбежностью являются уникальными – в словаре не может быть более одного ключа с одинаковым именем.

Записи в словаре можно перебирать в цикле. Вот один из способов выполнить эту задачу:

```
for student, language in foreign_languages.items():  
    print(student, 'изучает', language)
```

```
Алиса изучает испанский язык  
Кэрол изучает итальянский язык  
Дейв изучает итальянский язык  
Эстер изучает итальянский язык
```

В `foreign_languages` у нас есть парные данные – каждое имя связано с предметом. Кроме того, словари нередко используются для хранения нескольких разных элементов данных об одной сущности. В целях иллюстрации этого тонкого различия давайте взглянем на один элемент из `student_grades`:

```
student_grade = ('Алиса', 'испанский язык', '5')
```

Здесь мы знаем, что элементами в каждом из этих кортежей является имя, предмет и оценка:

```
student_name, subject, grade = student_grades[0]  
print(student_name,  
      'имеет оценку', grade,  
      'по предмету', subject, '')
```

```
Алиса имеет оценку 5 по предмету "испанский язык"
```

Вместо этого мы могли бы представить указанные выше данные в виде словаря. Словарь с информацией, описывающей один элемент, часто называют *записью* (record):

```
record = {  
    'name': 'Алиса',  
    'subject': 'испанский язык',  
    'grade': '5',  
}  
print(record['name'],  
      'имеет оценку', record['grade'],  
      'по предмету', record['subject'], '')
```

```
Алиса имеет оценку 5 по предмету "испанский язык"
```

Хотя результирующий исходный код немного длиннее, здесь нет абсолютно никакой двусмысленности в отношении сопоставления

индексов и того, что конкретно представляет каждое значение. Такая структура также полезна в ситуациях, когда некоторые поля могут быть необязательными.

А.7. Комбинирование типов данных

В большинстве этих простых примеров мы работали с коллекциями простых значений, таких как строковые литералы и числа, однако анализ данных часто предусматривает работу со сложными данными, в которых с каждым интересующим элементом данных ассоциировано несколько типов данных. Такие сложные данные часто представляются в виде коллекций коллекций, например списка словарей.

Выбор надлежащих типов данных для решаемой задачи облегчит вам написание бездефектного исходного кода и облегчит чтение вашего кода другими, но определение наилучших типов данных является навыком, приобретаемым благодаря опыту. Ниже показано несколько часто используемых комбинированных типов данных, но этот список вряд ли будет исчерпывающим.

А.7.1. Список кортежей

На самом деле мы уже видели его раньше. Рассмотрим данные `student_grades` из предыдущего примера по распаковке кортежа:

```
student_grades = [  
    ('Алиса', 'Испанский язык', '5'),  
    ('Боб', 'Французский язык', '3'),  
    ('Кэрол', 'Итальянский язык', '4+'),  
    ('Дейв', 'Итальянский язык', '5-'),  
]
```

Это список кортежей:

```
student_grades[1]  
( 'Боб', 'Французский язык', '3')
```

И мы можем работать с отдельными кортежами:

```
student_grades[1][2]  
'3'
```


А.7.2. Список словарей

В разделе, посвященном словарям, мы провели разведку того, как словарь нередко используется для хранения записи об одной сущности. Давайте конвертируем список кортежей `student_grades` в список записей `student_grade_records`:

```
student_grade_records = []
for student_name, subject, grade in student_grades:
    record = {
        'name': student_name,
        'subject': subject,
        'grade': grade,
    }
    student_grade_records.append(record)

student_grade_records

[{'name': 'Алиса', 'subject': 'испанский язык', 'grade': '5'},
 {'name': 'Боб', 'subject': 'французский язык', 'grade': '3'},
 {'name': 'Кэрол', 'subject': 'итальянский язык', 'grade': '4+'},
 {'name': 'Дейв', 'subject': 'итальянский язык', 'grade': '5-'}]
Теперь каждый элемент в списке является словарем:
student_grade_records[1]
{'name': 'Боб', 'subject': 'французский язык', 'grade': '3'}
```

И мы можем работать с отдельными записями:

```
student_grade_records[1]['grade']
'3'
```

Этот список словарей часто используется для представления данных из базы данных или API. Давайте воспользуемся этими данными для написания нашего исходного кода, поздравляющего учащихся с хорошими оценками, как мы сделали в разделе о распаковке кортежей:

```
for record in student_grade_records:
    if record['grade'].startswith('A'):
        print(record['name'], ', поздравляем тебя',
              'с получением оценки', record['grade'],
              'по предмету', '', record['subject'], ', ''')
Алиса, поздравляем тебя с получением оценки 5 по предмету "испанский язык"
Дейв, поздравляем тебя с получением оценки 5- по предмету "итальянский язык"
```

А.7.3. Словарь словарей

Список словарей очень полезен при работе с неуникальными данными; в предыдущем примере у каждого студента может быть несколько оценок по другим учебным предметам. Но иногда мы хотим обращаться к данным по определенному имени или ключу. В этом случае мы можем использовать словарь, значениями которого являются записи (т. е. другие словари).

Давайте снова воспользуемся данными из `student_grades`, но допустим, что нам просто нужна оценка по языку, чтобы можно было использовать имя студента в качестве ключа:

```
foreign_language_grades = {}
for student_name, subject, grade in student_grades:
    record = {
        'subject': subject,
        'grade': grade,
    }
    foreign_language_grades[student_name] = record

foreign_language_grades
{'Алиса': {'subject': 'испанский язык', 'grade': '5'},
 'Боб': {'subject': 'французский язык', 'grade': '3'},
 'Кэрол': {'subject': 'итальянский язык', 'grade': '4+'},
 'Дейв': {'subject': 'итальянский язык', 'grade': '5-'}}
```

Теперь мы можем обращаться к ним по имени студента:

```
foreign_language_grades['Алиса']
{'subject': 'испанский язык', 'grade': '5'}
```

И можем получать индивидуальные данные, которые нас интересуют:

```
foreign_language_grades['Алиса']['grade']
'5'
```

А.7.4. Словарь с кортежными ключами

Иногда бывает полезно использовать ключевые словари на нескольких компонентах данных. В словарях может использоваться любой немутуруемый объект в качестве ключа, который включает в свой со-

став кортежи. Продолжая наш пример с оценками студентов, мы, возможно, захотим, чтобы ключами были имя студента и предмет:

```
course_grades = {}
for student_name, subject, grade in student_grades:
    course_grades[student_name, subject] = grade

course_grades
{('Алиса', 'испанский язык'): '5',
 ('Боб', 'французский язык'): '3',
 ('Кэрол', 'итальянский язык'): '4+',
 ('Дейв', 'итальянский язык'): '5-'}
```

Теперь мы можем представить все оценки студента:

```
course_grades['Alice', 'Math'] = 'A'
course_grades['Alice', 'History'] = 'B'
course_grades
{('Алиса', 'испанский язык'): '5',
 ('Боб', 'французский язык'): '3',
 ('Кэрол', 'итальянский язык'): '4+',
 ('Дейв', 'итальянский язык'): '5-',
 ('Алиса', 'математика'): '5',
 ('Алиса', 'история'): '4'}
```

A.7.5. Еще один словарь словарей

Нередко возникает потребность получать пары предмет–оценка в виде таблицы успеваемости по конкретному студенту. Мы можем создать словарь с именами учащихся в качестве ключей, и значениями – в качестве словарей пар предмет–оценка. В этом случае нам нужно выполнить небольшую проверку – этот шаг прокомментирован ниже:

```
report_cards = {}
for student_name, subject, grade in student_grades:
    # Если по студенту табель успеваемости отсутствует,
    # то нам нужно создать пустой
    if student_name not in report_cards:
        report_cards[student_name] = {}
    report_cards[student_name][subject] = grade
report_cards
{'Алиса': {'испанский язык': '5'},
 'Боб': {'французский язык': '3'},
 'Кэрол': {'итальянский язык': '4+'},
 'Дейв': {'итальянский язык': '5-'}}
```

Преимущество этой дополнительной работы состоит в том, что теперь мы можем легко получать несколько оценок по одному студенту:

```
report_cards['Алиса']['математика'] = '5'
report_cards['Алиса']['история'] = '4'
report_cards
{'Алиса': {'испанский язык': '5', 'математика': '5', 'история': '4'},
 'Боб': {'французский язык': '3'},
 'Кэрол': {'итальянский язык': '4+'},
 'Дейв': {'итальянский язык': '5-'}}
```

И мы можем легко получать «табель успеваемости» студента:

```
report_cards['Алиса']
{'испанский язык': '5', 'математика': '5', 'история': '4'}
```





Приложение В. Модели NetLogo

NetLogo – это мультиагентная программируемая среда моделирования. Она разработана и поддерживается Центром связанного обучения и компьютерного моделирования при Северо-Западном университете (Виленски, 1999) и свободно доступен для скачивания в качестве настольного приложения¹ или для выполнения во Всемирной паутине² – мы рекомендуем настольную версию.

NetLogo поставляется с большой библиотекой образцовых моделей, включая несколько сетей. Эти предварительно написанные модели позволяют вам экспериментировать без необходимости кодировать всю модель целиком. Играя с различными начальными конфигурациями и параметрами, вы можете наблюдать, как они влияют на динамику и исходы модели. Благодаря этому вы сможете получать более глубокое понимание основополагающих правил и возникающих сетевых явлений.

После загрузки модели из библиотеки (используя меню **File** (Файл) в настольном приложении или поле поиска в веб-версии) вы увидите три панели: вкладки **Interface** (Интерфейс), **Info** (Информация) и **Code** (Исходный код). Во вкладке **Информация** представляется модель, объясняются правила ее использования, и предлагаются варианты проведения ее разведывательного анализа.

Давайте кратко рассмотрим ключевые элементы интерфейса модели приложения NetLogo: кнопки, переключатели, ползунки и мониторы. Эти элементы позволяют вам взаимодействовать с моделью. Кнопки используются для настройки, запуска и остановки модели. Ползунки и переключатели изменяют настройки модели. Мониторы и графики выводят данные на экран. В целях запуска модели вам сначала нужно ее настроить с помощью кнопки **setup** (настроить). Затем вы можете прокручивать модель по одному шагу за раз либо просматривать итерации с помощью кнопки **go** (перейти). Ползунок позволяет управлять скоростью выполнения. Переключатели и ползунки предоставляют вам доступ к настройкам и параметрам модели, чтобы иметь возможность проводить разведку разных сценариев или гипотез. Эта проекция позволяет видеть, что конкретно происходит с моделируемой сетью. Графики и мониторы показывают изменение ключевых статистических величин модели с течением времени. Графики имеют легенды для интерпретирования смысла диаграмм. Данные графиков можно экспортировать в электронную таблицу.

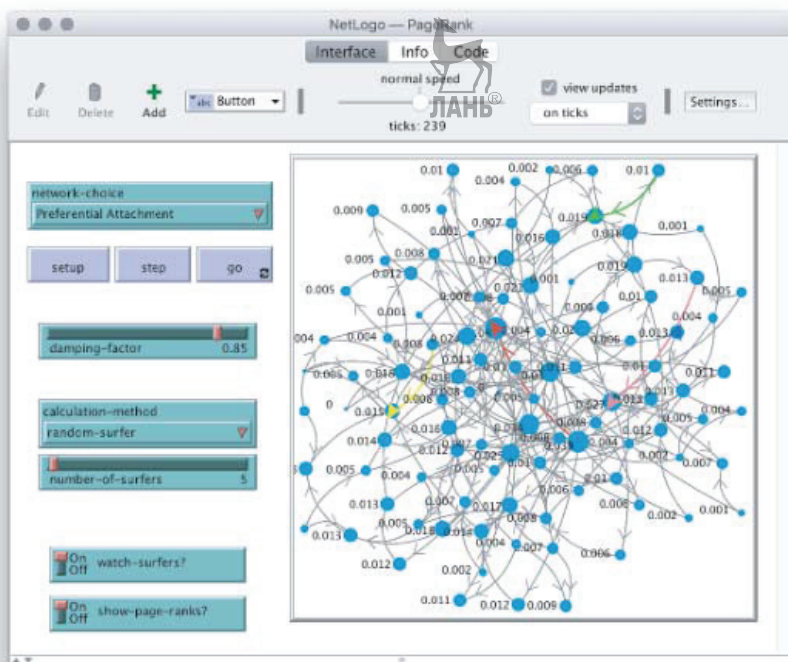
¹ См. ccl.northwestern.edu/netlogo/.

² См. www.netlogoweb.org.

Хотя существует возможность обращаться к исходному коду модели (на языке программирования NetLogo) и его модифицировать через вкладку code (и даже писать свои собственные модели), здесь мы сосредоточимся на выполнении нескольких библиотечных моделей, которые наиболее соответствуют материалу данного учебника.

В.1. Модель PageRank

Модель PageRank (Стоундал и Виленски, 2009) проиллюстрирована на рис. В.1. Метрика PageRank обсуждается в разделе 4.3. Указанная модель демонстрирует агентную имплементацию модели на основе случайного блуждания (random-surfer, т. е. случайный серфер) и имплементацию модели на основе степенного метода (diffusion) для вычисления модели PageRank. Имплементация на основе случайного блуждания имеет параметр, который позволяет указывать число «серферов». Вы сможете увидеть, как эти агенты перемещаются или переходят со страницы на страницу. Обратите внимание на разницу в скорости между этими двумя методами.



Данная модель показывает, как метрика PageRank обновляется для каждого узла на каждой итерации. Размер каждого узла примерно пропорционален его рангу PageRank. Выбор сети включает в себя две простые образцовые сети и более крупную сеть с более широким распределением значений степени-на-входе, сгенерированную посредством преференциального прикрепления. В одной из образцовых сетей некоторые узлы не имеют входящих связей, но все же в итоге заканчиваются с ненулевым рангом PageRank. Поиграйте с коэффициентом демпфирования, чтобы увидеть, как он влияет на эти значения. Проведите разведку того, что происходит, когда коэффициент демпфирования близок к нулю или единице.

В.2. Гигантская компонента

Модель «Гигантская компонента» (Виленски, 2005а) проиллюстрирована на рис. В.2. Она демонстрирует темп, с которым гигантская компонента возникает в случайных сетях при увеличении средней степени, как обсуждалось в разделе 5.1. Изначально вероятность связи, средняя степень и плотность равны нулю; связей нет, и каждый узел является узлом-одиночкой (синглетоном). На каждом шаге добавляется связь между двумя случайными узлами, которые еще не со-

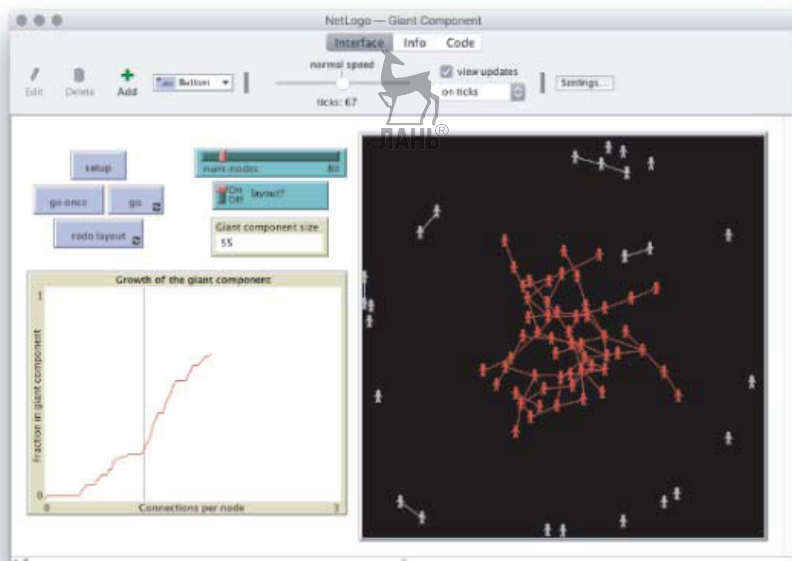


Рис. В.2 Экранный снимок модели «Гигантская компонента» в NetLogo. Указанная модель лицензирована в соответствии с CC BY-NC-SA 3.0 и воспроизводится с разрешения

единены связью. По мере выполнения модели формируются компоненты – вначале малые, затем постепенно увеличивающиеся по мере того, как новые связи приводят к слиянию отдельных компонент. Гигантская компонента выделена красным цветом.

Единственным параметром модели является размер сети. График показывает рост гигантской компоненты с течением времени как функции от средней степени. Вертикальная линия на графике показывает точку, где средняя степень равна единице. Обратите внимание, как скорость роста гигантской компоненты увеличивается после этой критической точки: сеть претерпевает транзит от фрагментированной фазы с большим числом малых компонент к главному образом связной фазе с гигантской компонентой и несколькими оставшимися малыми компонентами. Сравните поведение модели в нескольких прогонах с одинаковым размером сети и с разными размерами.

В.3. Малые миры

Модель «Малые миры» (Виленски, 2005с) проиллюстрирована на рис. В.3. Она имплементирует маломировую модель, обсуждаемую в разделе 5.2, показывая, как генерировать сети с короткой средней

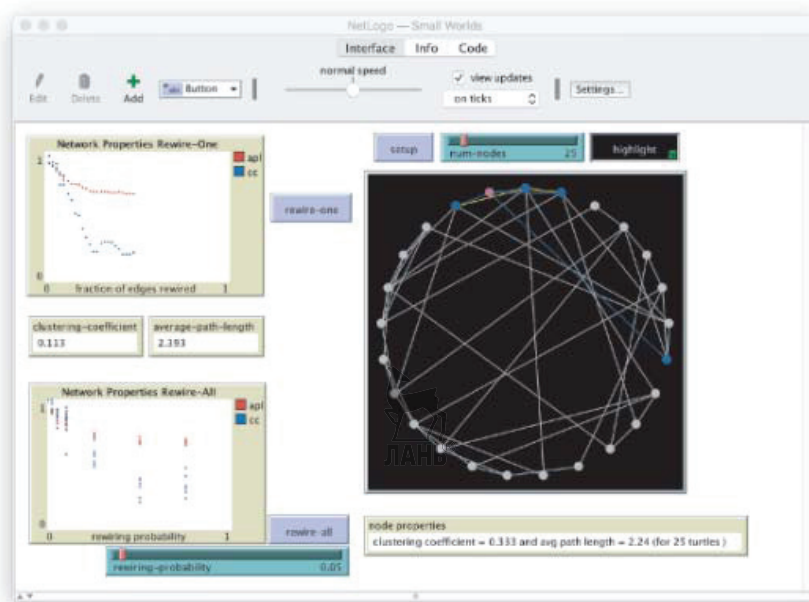


Рис. В.3 Экранный снимок модели «Малые миры» в NetLogo. Указанная модель лицензирована в соответствии с CC BY-NC-SA 3.0 и воспроизводится с разрешения

длиной пути и высоким коэффициентом кластеризации. Параметр определяет число узлов. После настройки исходной решетчатой сети вы можете переподсоединять по одной связи за раз и понаблюдать за уменьшением средней длины пути и коэффициента кластеризации как функции от доли переподсоединенных связей (верхний график). Второй режим состоит в установлении параметра вероятности переподсоединения, а затем переподсоединении всех связей сразу с этой вероятностью. Окончательная средняя длина пути и коэффициент кластеризации выводятся на графике в сопоставлении с вероятностью переподсоединения (нижний график).

Поиграйте с различными вероятностями переподсоединения и наблюдайте за трендами средней длины пути и значениями коэффициента кластеризации. Обратите внимание, что в определенных диапазонах вероятности переподсоединения средняя длина пути уменьшается быстрее, чем коэффициент кластеризации. По сути дела, существует диапазон значений, для которых (нормализованная) средняя длина пути намного меньше (нормализованного) коэффициента кластеризации. Сети в этом диапазоне считаются малыми мирами. Выявите приближенный диапазон и попытайтесь получить малый мир, переподсоединяя по одной связи за раз. Развейдите вопрос, зависят ли тренды от числа узлов в сети.

V.4. Преференциальное прикреплeние

Модель «Преференциальное прикреплeние» (Виленски, 2005b) проиллюстрирована на рис. V.4. Она демонстрирует процесс возникновения хабов посредством преференциального прикреплeния, как описано в разделе 5.4. Указанная модель начинается с двух узлов, соединенных связью. На каждом шаге добавляется новый узел и соединяется с одним существующим узлом. Последний отбирается случайно, но с некоторым смещением: вероятность селекции пропорциональна степени узла. Обратите внимание, что, поскольку для каждого нового узла добавляется только одна связь, эта имплементация модели порождает деревья (см. раздел 2.4).

Используйте кнопку *resize nodes* (изменить размер узлов), которая делает размер узла пропорциональным степени, чтобы понаблюдать за возникновением хабов. Обратите внимание, что более старые или новые узлы с большей вероятностью будут становиться основными хабами. Вы можете изучить степенное распределение сети, посмотрев на графики. Верхний график представляет собой гистограмму степеней узлов. На нижнем графике показано то же распределение, но обе оси находятся в логарифмической шкале. Дайте модели поработать некоторое время, затем опишите контур степенного распределения на двойном логарифмическом графике. Сравните его с распределе-

нием на рис. 5.8(с). Ускорьте модель (вы можете отключить переключитесь layout (компоновка) и снять флажок view updates (просмотр обновлений), чтобы дать сформироваться крупной сети. Инспектируйте увеличение ширины степенного распределения по мере приобретения сетью крупного размера. Дайте объяснение.

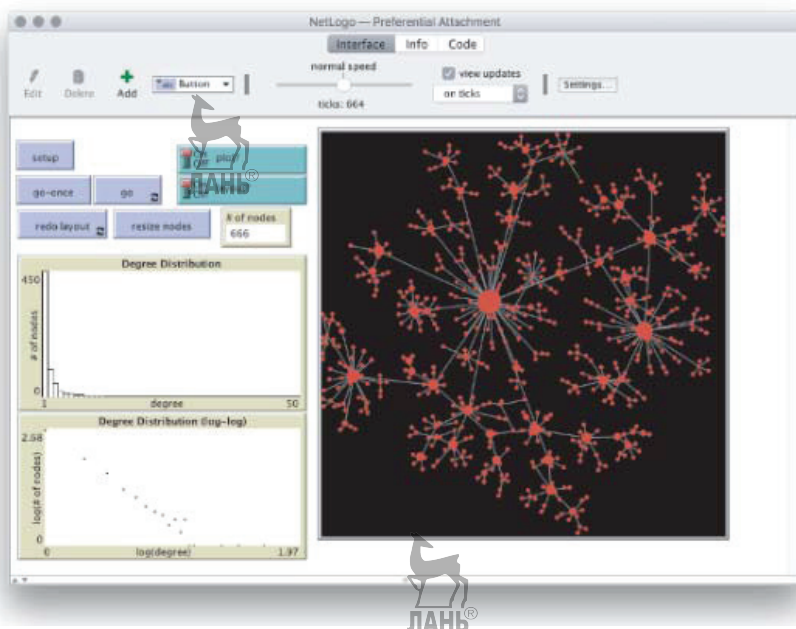


Рис. В.4 Экранный снимок модели «Преференциальное прикрепление» в NetLogo. Указанная модель лицензирована в соответствии с CC BY-NC-SA 3.0 и воспроизводится с разрешения

В.5. Вирус в сети

Модель «Вирус в сети» (Стоундал и Виленски, 2008) объединяет модели SIS и SIR распространения эпидемий, обсуждаемые в разделе 7.2.1. Параметр gain-resistance-chance (шанс сопротивления усилению) смешивает две модели: когда он равен нулю, модель соответствует SIS (как показано на рис. В.5), а когда он равен единице, модель соответствует SIR. Указанная модель перестает работать, как только вирус полностью затухнет. На графике показано число узлов в трех состояниях (S, I, R) с течением времени. Связи между сопротивляющимися узлами и их соседями затемнены, так как они больше не могут распространять вирус. Сеть имеет гомогенную степень и географическую гомофилию: шанс быть связанными имеют только те узлы, которые находятся рядом друг с другом (на основе евклидова расстояния). Другие параметры модели включают среднюю степень сети, частоту

инфицирования ($\text{virus-spread-chance}$, т. е. шанс распространения вируса) и частоту выздоровления (вероятность выздоровления).

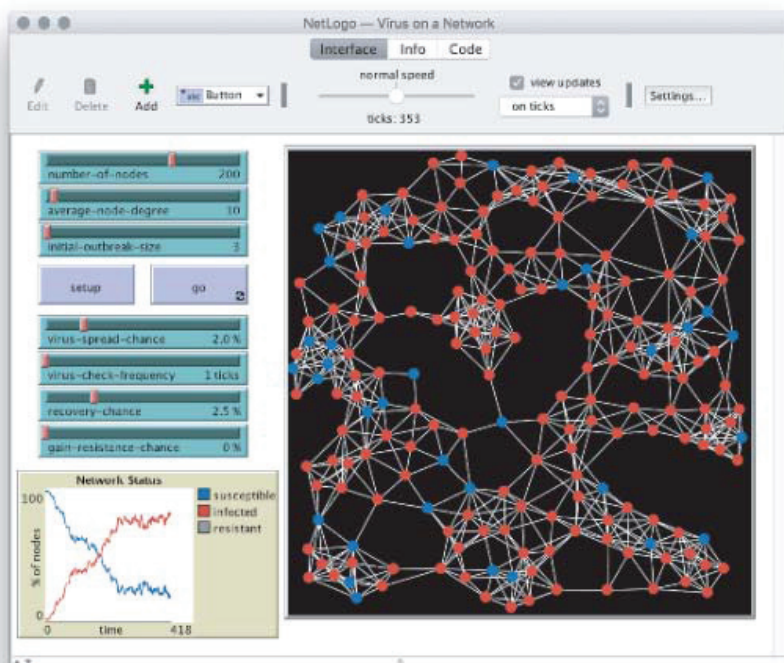


Рис. В.5 Экранный снимок модели «Вирус в сети» в NetLogo. Указанная модель лицензирована в соответствии с CC BY-NC-SA 3.0 и воспроизводится с разрешения

Выполните модель с экстремальными значениями параметра $\text{gain-resistance-chance}$, чтобы воспроизвести динамику моделей SIS и SIR. В модели SIS наблюдайте за влиянием частот инфицирования и выздоровления на баланс между популяциями S и I. В модели SIR разведайте влияние частоты инфицирования, частоты выздоровления и средней степени на максимальное число инфицированных узлов. Сначала поварьируйте каждый параметр, держа остальные постоянными, и наблюдайте за тем, что происходит. Затем поиграйте с разными комбинациями этих трех параметров, чтобы воспроизвести поведение, объясненное эпидемическим порогом в уравнении (7.5). Дайте объяснение тому, какие условия позволяют эпидемиям распространяться по большей части сети. Когда вирус затухает, не заразив всю популяцию, проверьте уцелевшие узлы. Некоторые кластеры узлов могут никогда не заразиться даже при превышении эпидемического порога. Опишите ключевые структурные характеристики узлов, связей и сообществ, которые способствуют или препятствуют распространению эпидемии.

В.6. Изменение языка

Модель «Изменение языка» (Трутман и Виленски, 2007) сочетает в себе модели динамики мнений на основе избирателя и на основе большинства (раздел 7.3.1), а также дробно-пороговую модель социального заражения (Раздел 7.1.1). Параметр `update-algorithm` (алгоритм обновления) позволяет отбирать одну из этих моделей: алгоритм `individual` (индивидуум) соответствует модели на основе избирателя, показанной на рис. В.6. Алгоритм `threshold` (порог) соответствует дробно-пороговой модели, и в этом случае вы можете установить параметр порога (`threshold-val`); когда он равен 0.5, модель эквивалентна модели на основе большинства. Указанная модель работает на малом дереве, основанном на преференциальном прикреплении (до 100 узлов). Состояния, или мнения, называются грамматиками (`grammars`) ноль (черный) и один (белый). Еще одним ключевым параметром является первоначальная доля узлов в состоянии один (`percent-grammar-1`). Обязательно отключите `sink-state-1` (состояние стока), иначе узлы не смогут вернуться в нулевое состояние, после того как они примут состояние один.

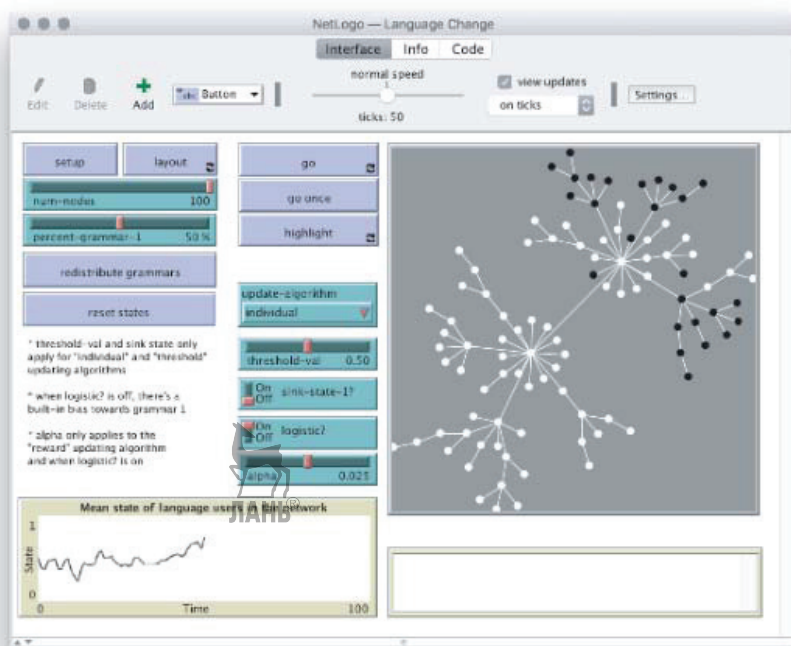


Рис. В.6 Экранный снимок модели «Изменение языка» в NetLogo. Указанная модель лицензирована в соответствии с CC BY-NC-SA 3.0 и воспроизводится с разрешения

В пороговой модели обратите внимание на роль хабов как влиятелей. Затем установите порог равным 0.5 и сравните, как сеть сходится к стационарному состоянию в моделях на основе избирателя и на основе большинства. Понаблюдайте за тем, когда стационарное состояние является состоянием консенсуса или поляризации с сосуществующими состояниями. Отчитайтесь о том, какая модель чаще порождает консенсус. Развейте вопрос, влияют ли на исход другие условия, такие как число узлов и первоначальная конфигурация. Изучите вероятность достижения консенсуса в белом состоянии как функции от первоначальной доли белых узлов. Обсудите, соответствует ли эта вероятность выходу поведению на рис. 7.12 для этих двух моделей.





Справочные материалы

- Achlioptas, D., Clauset, A., Kempe, D., and Moore, C. 2009. On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs. *Journal of the ACM*, 56 (4), 21 (Ахлиоптас, Д., Клаузе, А., Кемпе, Д. и Мур, С. О систематическом смещении отбора маршрутов трасс: или о степенных распределениях по экспоненциальному закону в регулярных графах).
- Adamic, L. A., Lukose, R. M., Puniyani, A. R., and Huberman, B. A. 2001. Search in power-law networks. *Physical Review E*, 64 (4), 046135. Адамик, Л. А., Лукозе, Р. М., Пунияни, А. Р. и Губерман, Б. А. 2001 (Адамик, Л. А., Лукозе, Р. М., Пунияни, А. Р. и Губерман, Б. А. Поиск в сетях, построенных на экспоненциальном законе).
- Ahn, Y.-Y., Ahnert, S. E., Bagrow, J. P., and Barabási, A.-L. 2011. Flavor network and the principles of food pairing. *Scientific Reports*, 1, 196 (Ан, У.-У., Анерт, С. Е., Багров, Дж. П. и Барабаши, А.-Л. Сеть вкусовых ингредиентов и принципы сочетания продуктов питания).
- Aiello, L., Barrat, A., Schifanella, R., Cattuto, C., Markines, B., and Menczer, F. 2012. Friendship prediction and homophily in social media. *ACM Transactions on the Web*, 6 (2), 9 (Айелло, Л., Баррат, А., Шифанелла, Р., Камтутто, С., Маркинес, Б. и Менцер, Ф. Предсказание дружбы и гомофилия в социальных сетях).
- Albert, R., Jeong, H., and Barabási, A.-L. 1999. Internet: Diameter of the world-wide web. *Nature*, 401 (6749), 130 (Альберт, Р., Чонг, Х. и Барабаши, А.-Л. Интернет: диаметр Всемирной паутины).
- Albert, R., Jeong, H., and Barabási, A.-L. 2000. Error and attack tolerance of complex networks. *Nature*, 406 (6794), 378–382 (Альберт, Р., Чонг, Х. и Барабаши, А.-Л. Устойчивость сложных сетей к ошибкам и атакам).
- Anderson, R. M., and May, R. M. 1992. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press: Oxford (Андерсон, Р. М. и Май, Р. М. Инфекционные заболевания человека: динамика и контроль).
- Arenas, A., Duch, J., Fernández, A., and Gómez, S. 2007. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9 (6), 176 (Аренас, А., Дуч, Дж., Фернандес, А. и Гомес, С. Сокращение размеров сложных сетей с сохранением модулярности).
- Arenas, A., Fernández, A., and Gómez, S. 2008. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10 (5), 053039 (Аренас, А., Фернандес, А. и Гомес, С. Анализ структуры сложных сетей на разных уровнях разрешающей способности).

- Backstrom, L., Boldi, P., Rosa, M., Ugander, J., and Vigna, S. 2012. Four degrees of separation. In Proceedings of the 4th Annual ACM Web Science Conference (WebSci'12), pp. 33–42 (*Бэкстрем, Л., Болди, П., Роза, М., Угандер, Дж. и Винья, С. Четыре степени сепарации*).
- Baeza-Yates, R., and Ribeiro-Neto, B. 2011. Modern Information Retrieval: The Concepts and Technology Behind Search, 2nd edn. ACM Press Books Addison-Wesley: New York (*Баеза-Йейтс, Р. и Рибейро-Нето, Б. Современный информационный поиск: концепции и технологии, лежащие в основе поиска, 2-е изд.*).
- Barabási, A.-L. 2003. Linked: How Everything is Connected to Everything Else and What it Means for Business, Science, and Everyday Life. Basic Books: New York (*Барабаш, А.-Л. Связано: как все соединено со всем остальным и что это значит для бизнеса, науки и повседневной жизни. Базовые книги*).
- Barabási, A.-L. 2016. Network Science. Cambridge University Press: Cambridge (*Барабаш, А.-Л. Наука о сетях*).
- Barabási, A.-L., and Albert, R. 1999. Emergence of scaling in random networks. *Science*, 286 (5439), 509–512 (*Барабаш, А.-Л. и Альберт, Р. Возникновение масштабирования в случайных сетях*).
- Barrat, A., Barthélemy, M., and Vespignani, A. 2008. Dynamical Processes on Complex Networks. Cambridge University Press: Cambridge (*Баррат, А., Бартелеми, М. и Веспиньяни, А. Динамические процессы на сложных сетях*).
- Bastian, M., Heymann, S., Jacomy, M., et al. 2009. Gephi: An open source software for exploring and manipulating networks. In Proceedings of the Third AAAI International Conference on Web and Social Media (ICWSM), pp. 361–362 (*Бастиян М., Хейманн С., Джакоми М. и соавт. Gephi: Программно-информационное обеспечение с открытым исходным кодом для разведывания сетей и управления ими*).
- Batagelj, V., Mrvar, A., and Zaveršnik, M. 1999. Partitioning approach to visualization of large graphs. In International Symposium on Graph Drawing, pp. 90–97 (*Батагель, В., Мрвар, А. и Завершник, М. Подход на основе разделения для визуализации крупных графов*).
- Baur, M., Brandes, U., Gaertler, M., and Wagner, D. 2004. Drawing the AS graph in 2.5 dimensions. In International Symposium on Graph Drawing, pp. 43–48 (*Баур, М., Брандес, У., Гертлер, М. и Вагнер, Д. Рисование графа автономной системы в 2.5-размерностях*).
- Bavelas, A. 1950. Communication patterns in task-oriented groups. *Journal of the Acoustical Society of America*, 22 (6), 725–730 (*Бавелас, А. Шаблоны общения в задачно-ориентированных группах*).
- Beiró, M. G., Alvarez-Hamelin, J. I., and Busch, J. R. 2008. A low complexity visualization tool that helps to perform complex systems analysis. *New Journal of Physics*, 10 (12), 125003 (*Бейро, М. Г., Альварес-Хамелин, Дж. И. и Буш, Дж. Р. Низкосложностный визуализационный инструмент, который помогает выполнять анализ сложных систем*).
- Bellman, R. 1958. On a routing problem. *Quarterly of Applied Mathematics*, 16, 87–90 (*Беллман, Р. Задача о маршрутизации*).

- Berners-Lee, T., and Fischetti, M. 2000. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. HarperCollins: New York (*Бернерс-Ли, Т. и Фишетти, М. Плетение паутины: изначальный дизайн и конечная судьба Всемирной паутины от ее изобретателя*).
- Bhan, A., Galas, D. J., and Dewey, T. G. 2002. A duplication growth model of gene expression networks. *Bioinformatics*, 18 (11), 1486–1493 (*Бхан, А., Галас, Д. Дж. и Дьюи, Т. Г. Модель дубликационного роста в сетях генной экспрессии*).
- Bianconi, G., and Barabási, A.-L. 2001. Bose–Einstein condensation in complex networks. *Physical Review Letters*, 86 (24), 5632–5635 (*Бьянкони, Г. и Барабаши, А.-Л. Конденсация Бозе–Эйнштейна в сложных сетях*).
- Bichot, C.-E., and Siarry, P. 2013. Graph Partitioning. Wiley: Hoboken, NJ (*Бишот, С.-Е. и Сиарри, П. Деление графа на разделы*).
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, P10008 (*Блондель, В. Д., Гийом, Ж.-Л., Ламбьотт, Р. и Лефевр, Э. Быстрое развертывание сообществ в крупных сетях*).
- Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardenes, J., Romance, M., et al. 2014. The structure and dynamics of multilayer networks. *Physics Reports*, 544 (1), 1–122 (*Боккалетти, С., Бьянкони, Г., Криадо, Р., Дель Генуи, К. И., Гомес-Гарденес, Дж., Романтика, М. и соавт. Структура и динамика многослойных сетей*).
- Bollobás, B. 2012. Graph Theory: An Introductory Course. Springer: New York (*Боллобас, Б., Теория графов: вводный курс*).
- Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25 (2), 163–177 (*Брандес, У. Более быстрый алгоритм для центральности на основе промежуточности*).
- Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30 (1–7), 107–117 (*Брин, С. и Пейдж, Л. Анатомия крупномасштабной гипертекстовой поисковой машины для Всемирной паутины*).
- Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. 2000. Graph structure in the web. *Computer Networks*, 33 (1–6), 309–320 (*Бродер, А., Кумар, Р., Магхул, Ф., Рагхаван, П., Раджагопалан, С., Стата, Р. и соавт. Структура графов во Всемирной паутине*).
- Caldarelli, G. 2007. Scale-Free Networks. Oxford University Press: Oxford (*Кальдарелли, Г. Безмасштабные сети*).
- Caldarelli, G., and Chessa, A. 2016. Data Science and Complex Networks: Real Case Studies with Python. Oxford University Press: Oxford (*Кальдарелли, Г. и Чесса, А. Наука о данных и сложные сети: исследования реальных случаев применения с использованием Python*).
- Castellano, C., Fortunato, S., and Loreto, V. 2009. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81 (2), 591–646 (*Кастеллано, С., Фортунато, С. и Лорето, В. Статистическая физика социальной динамики*).

- Centola, D., and Macy, M. 2007. Complex contagions and the weakness of long ties. *American Journal of Sociology*, 113 (3), 702–734 (Чентола, Д. и Мэйси, М. Сложные инфекции и слабость длительных уз).
- Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. P. 2010. Measuring user influence in Twitter: The million follower fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pp. 10–17 (Ча, М., Хаддади, Х., Беневенуто, Ф. и Гуммади, К. П. Измерение влияния на пользователей в Твиттере: заблуждение с миллионом подписчиков).
- Christakis, N. A., and Fowler, J. H. 2010. Social network sensors for early detection of contagious outbreaks. *PloS ONE*, 5 (9), e12948 (Кристакис, Н. А. и Фаулер, Дж. Х. Датчики социальных сетей для раннего выявления вспышек инфекционных заболеваний).
- Clauset, A., Newman, M. E. J., and Moore, C. 2004. Finding community structure in very large networks. *Physical Review E*, 70 (6), 066111 (Клаузет, А., Ньюман, М. Э. Дж. и Мур, С. Поиск структуры сообществ в очень крупных сетях).
- Clifford, P., and Sudbury, A. 1973. A model for spatial conflict. *Biometrika*, 60 (3), 581–588 (Клиффорд, П. и Садбери, А. Модель пространственного конфликта).
- Cohen, R., and Havlin, S. 2003. Scale-free networks are ultrasmall. *Physical Review Letters*, 90 (5), 058701 (Коэн, Р. и Хавлин, С. Безмасштабные сети являются ультрамалыми).
- Cohen, R., and Havlin, S. 2010. *Complex Networks: Structure, Robustness and Function*. Cambridge University Press: Cambridge (Коэн, Р. и Хавлин, С. Сложные сети: структура, устойчивость и функционирование).
- Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S. 2000. Resilience of the Internet to random breakdowns. *Physical Review Letters*, 85 (21), 4626–4628 (Коэн Р., Эрез К., Бен-Авраам Д. и Хавлин С. Отказоустойчивость интернета к случайным сбоям).
- Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S. 2001. Breakdown of the Internet under intentional attack. *Physical Review Letters*, 86 (16), 3682–3685 (Коэн Р., Эрез К., Бен-Авраам Д. и Хавлин С. Сбой в работе интернета в результате преднамеренной атаки).
- Cohen, R., Havlin, S., and Ben-Avraham, D. 2002. Structural properties of scale-free networks. *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley: Weinheim (Коэн, Р., Хавлин, С. и Бен-Авраам, Д. Структурные свойства безмасштабных сетей. Справочник по графам и сетям: от генома до интернета).
- Cohen, R., Havlin, S., and Ben-Avraham, D. 2003. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91 (24), 247901 (Коэн, Р., Хавлин, С. и Бен-Авраам, Д. Эффективные стратегии иммунизации для компьютерных сетей и популяций).
- Condon, A., and Karp, R. M. 2001. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18,

- 116–140 (Кондон, А. и Карп, Р. М. Алгоритмы деления графов на разделимые на модели на основе внедрения разделов. Случайные структуры и алгоритмы).
- Conover, M., Gonçalves, B., Ratkiewicz, J., Flammini, A., and Menczer, F. 2011a. Predicting the political alignment of Twitter users. In *Proceedings of the 3rd IEEE Conference on Social Computing (SocialCom)*, pp. 192–199 (Коновер, М., Гонсалвес, Б., Раткевич, Дж., Фламмини, А. и Менцер, Ф. Предсказание политического расклада пользователей Twitter).
- Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Flammini, A., and Menczer, F. 2011b. Political polarization on Twitter. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pp. 89–96 (Коновер, М., Раткевич, Дж., Франсиско, М., Гонсалвес, Б., Фламмини, А. и Менцер, Ф. Политическая поляризация в Твиттере).
- Daley, D. J., and Kendall, D. G. 1964. Epidemics and rumours. *Nature*, 204 (4963), 1118 (Дейли, Д. Дж. и Кендалл, Д. Г. Эпидемии и слухи).
- Davison, B. D. 2000. Topical locality in the web. In *Proceedings of the 23rd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 272–279 (Дэвисон, Б. Д. Тематическая локальность во Всемирной паутине).
- Dawkins, R. 2016. *The Selfish Gene: 40th Anniversary Edition*, 4th edn. Oxford University Press: Oxford (Докинз, Р. Эгоистичный ген).
- Deffuant, G., Neau, D., Amblard, F., and Weisbuch, G. 2000. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3 (01n04), 87–98 (Деффуант, Г., Нео, Д., Амблард, Ф. и Вайсбуч, Г. Смешивание мнений между взаимодействующими агентами).
- Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. G. 1998. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall: Upper Saddle River, NJ (Ди Баттиста, Г., Ид, П., Тамассия, Р. и Толлис, И. Г. Рисование графов: алгоритмы визуализации графов).
- Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 (1), 269–271 (Дейкстра, Э. У. Примечание по поводу двух задач в связи с графами).
- Dodds, P. S., Muhamad, R., and Watts, D. J. 2003. An experimental study of search in global social networks. *Science*, 301 (5634), 827–829 (Доддс, П. С., Мухамад Р. и Уоттс, Д. Дж. Экспериментальное исследование поиска в глобальных социальных сетях).
- Dorogovtsev, S. N., and Mendes, J. F. F. 2013. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press: Oxford (Дороговцев, С. Н. и Мендес, Дж. Ф. Ф. Эволюция сетей: от биологических сетей к интернету и WWW).
- Dorogovtsev, S. N., Mendes, J. F. F., and Samukhin, A. N. 2000. Structure of growing networks with preferential linking. *Physical Review Letters*, 85 (21), 4633–4636 (Дороговцев С. Н., Мендес Дж. Ф. Ф. и Самухин А. Н. Структура растущих сетей с преференциальным связыванием).

- Dunbar, R. I. M. 1992. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22 (6), 469–493 (Данбар, Р. И. М. Размер неокортекса как ограничение на размер группы у приматов).
- Dunne, J. A., Williams, R. J., and Martinez, N. D. 2002. Food-web structure and network theory: The role of connectance and size. *Proceedings of the National Academy of Sciences of the USA*, 99 (20), 12917–12922 (Данн, Дж. А., Уильямс, Р. Дж. и Мартинес, Н. Д. Структура пищевой паутины и теория сетей: роль связи и размера).
- Eades, P. 1984. A heuristic for graph drawing. *Congressus Numerantium*, 42, 149–160 (Идз, П. Эвристика для рисования графов).
- Easley, D., and Kleinberg, J. 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press: Cambridge (Исли, Д. и Клейнберг, Дж. Сети, толпы и рынки: рассуждения о высокосвязном мире).
- Erdős, P., and Rényi, A. 1959. On random graphs. I. *Publicationes Mathematicae Debrecen*, 6, 290–297 (Эрдеш, П. и Рени, А. О случайных графах).
- Feld, S. L. 1991. Why your friends have more friends than you do. *American Journal of Sociology*, 96 (6), 1464–1477 (Фелд, С. Л. Почему у твоих друзей больше друзей, чем у тебя).
- Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. 2016. The rise of social bots. *Communications of the ACM*, 59 (7), 96–104 (Феррара, Э., Варол, О., Дэвис, С., Менцер, Ф. и Фламмини, А. Появление социальных ботов).
- Festinger, L. 1954. A theory of social comparison processes. *Human Relations*, 7 (2), 117–140 (Фестингер, Л. Теория процессов социального сравнения).
- Fienberg, S. E., and Wasserman, S. 1981. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12, 156–192 (Файенберг, С. Е. и Вассерман, С. Категориальный анализ данных одиночных социометрических отношений).
- Ford Jr., L. R. 1956. *Network Flow Theory*. Technical Report Paper P-923. RAND Corporation (Форд-мл., Л. Р. Теория Сетевых Поток).
- Fortunato, S. 2010. Community detection in graphs. *Physics Reports*, 486 (3–5), 75–174 (Фортунаато, С. Обнаружение сообществ в графах).
- Fortunato, S., and Barthélemy, M. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the USA*, 104 (1), 36–41 (Фортунаато, С. и Бартелими, М. Лимит разрешающей способности в обнаружении сообществ).
- Fortunato, S., and Hric, D. 2016. Community detection in networks: A user guide. *Physics Reports*, 659, 1–44 (Фортунаато, С. и Хрик, Д. Обнаружение сообществ в сетях: руководство пользователя).
- Fortunato, S., Flammini, A., and Menczer, F. 2006. Scale-free network growth by ranking. *Physical Review Letters*, 96 (21), 218701 (Фортунаато, С., Фламмини, А. и Менцер, Ф. Рост безмасштабной сети путем ранжирования).

- Fortunato, S., Boguñá, M., Flammini, A., and Menczer, F. 2007. On local estimations of PageRank: A mean field approach. *Internet Mathematics*, 4 (2–3), 245–266 (Фортуато, С., Богуна, М., Фламмини, А. и Менцер, Ф. О локальных оцениваниях модели PageRank: подход на основе среднего поля).
- Fred, A. L. N., and Jain, A. K. 2003. Robust data clustering. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 128–136 (Фред, А. Л. Н. и Джейн, А. К. Устойчивая кластеризация данных).
- Freedman, D., Pisani, R., and Purves, R. 2007. *Statistics*. W. W. Norton & Co.: New York (Фридман, Д., Пизани, Р. и Пурвес, Р. Статистика).
- Freeman, L. C. 1977. A set of measures of centrality based on betweenness. *Sociometry*, 40 (1), 35–41 (Фримен, Л. С. Набор мер центральности, основанных на промежуточности).
- Fruchterman, T. M. J., and Reingold, E. M. 1991. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21 (11), 1129–1164 (Фрухтерман, Т. М. Дж. и Рейнгольд, Э. М. Рисование графов путем размещения по направлению силы).
- Galam, S. 2002. Minority opinion spreading in random geometry. *The European Physical Journal B: Condensed Matter and Complex Systems*, 25 (4), 403–406 (Галам, С. Распространение мнения меньшинства в случайной геометрии).
- Gao, J., Buldyrev, S. V., Stanley, H. E., and Havlin, S. 2012. Networks formed from interdependent networks. *Nature Physics*, 8 (1), 40–48 (Гао, Дж., Булдырев, С. В., Стэнли, Х. Е. и Хавлин, С. Сети, сформированные из взаимозависимых сетей).
- Gil, S., and Zanette, D. H. 2006. Coevolution of agents and networks: Opinion spreading and community disconnection. *Physics Letters A*, 356 (2), 89–94 (Гил, С. и Занетт, Д. Х. Коэволюция агентов и сетей: распространение мнений и разъединение сообществ).
- Gilbert, E. N. 1959. Random graphs. *Annals of Mathematical Statistics*, 30 (4), 1141–1144 (Гильберт, Э. Н. Случайные графы).
- Girvan, M., and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the USA*, 99 (12), 7821–7826 (Гирван, М. и Ньюман, М. Э. Дж. Структура в форме сообществ в социальных и биологических сетях).
- Glauber, R. J. 1963. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4 (2), 294–307 (Глаубер, Р. Дж. Зависимая от времени статистика модели Изинга).
- Gleich, D. F. 2015. PageRank beyond the Web. *SIAM Review*, 57 (3), 321–363 (Глейх, Д. Ф. PageRank за пределами Всемирной паутины).
- Goel, S., Anderson, A., Hofman, J., and Watts, D. J. 2015. The structural virality of online diffusion. *Management Science*, 62 (1), 180–196 (Гоэль, С., Андерсон, А., Хофман, Дж. и Уоттс, Д. Дж. Структурная вирусность онлайн-диффузии).
- Goldenberg, J., Libai, B., and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Market-*

- ing Letters, 12 (3), 211–223 (Голденберг, Дж., Либай, Б. и Мюллер, Э. Разговор о сети: сложные системы смотрят на основополагающий процесс сарафанного радио).
- Granovetter, M. 1973. The strength of weak ties. *American Journal of Sociology*, 78 (6), 1360–1380 (Грановеттер, М. Сила слабых уз).
- Granovetter, M. 1978. Threshold models of collective behavior. *American Journal of Sociology*, 83 (6), 1420–1443 (Грановеттер, М. Пороговые модели коллективного поведения).
- Guimerà, R., Sales-Pardo, M., and Amaral, L. A. 2004. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70 (2), 025101(R) (Гимера, Р., Салес-Пардо, М. и Амарал, Л. А. Модулярность из флуктуаций в случайных графах и сложных сетях).
- Holland, P. W., and Leinhardt, S. 1971. Transitivity in structural models of small groups. *Comparative Group Studies*, 2 (2), 107–124 (Холланд, П. У. и Лейнхардт, С. Транзитивность в структурных моделях малых групп).
- Holland, P. W., and Leinhardt, S. 1981. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76 (373), 33–50 (Холланд, П. У. и Лейнхардт, С. Экспоненциальное семейство распределений вероятностей для направленных графов).
- Holland, P., Laskey, K. B., and Leinhardt, S. 1983. Stochastic blockmodels: First steps. *Social Networks*, 5 (2), 109–137 (Холланд, П., Ласки, К. Б. и Лейнхардт, С. Стохастические блок-модели: первые шаги).
- Holme, P., and Newman, M. E. J. 2006. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74 (5), 056108 (Холм, П. и Ньюман, М. Э. Дж. Неравновесный фазовый переход в коэволюции сетей и мнений).
- Holme, P., and Saramäki, J. 2012. Temporal networks. *Physics Reports*, 519 (3), 97–125 (Холме, П. и Сарамяки, Дж. Темпоральные сети).
- Hric, D., Darst, R. K., and Fortunato, S. 2014. Community detection in networks: Structural communities versus ground truth. *Physical Review E*, 90 (6), 062805 (Хрик, Д., Дарст, Р. К. и Фортунато, С. Обнаружение сообществ в сетях: структурные сообщества в сравнении с достоверными эмпирическими данными).
- Hu, Y., Chen, H., Zhang, P., Li, M., Di, Z., and Fan, Y. 2008. Comparative definition of community and corresponding identifying algorithm. *Physical Review E*, 78 (2), 026121 (Ху, Ю., Чэнь, Х., Чжан, П., Ли, М., Ди, З. и Фан, Ю. Сравнительное определение сообщества и соответствующий алгоритм выявления).
- Jacomy, M., Venturini, T., Heymann, S., and Bastian, M. 2014. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS ONE*, 9 (6), e98679 (Джакоми, М., Вентурины, Т., Хейманн, С. и Бастиян, М. ForceAtlas2, алгоритм непрерывной компоновки графов для удобной визуализации сети, разработанный для программы Gephi).

- Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. 2007. Social phishing. *Communications of the ACM*, 50 (10), 94–100 (Джагатиц, Т. Н., Джонсон, Н. А., Якобссон, М. и Менцер, Ф. Социальный фишинг).
- Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data clustering: A review. *ACM Computing Surveys*, 31 (3), 264–323 (Джейн, А. К., Мерти, М. Н. и Флинн, П. Дж. Кластеризация данных: обзор).
- Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. N. 2001. Lethality and centrality in protein networks. *Nature*, 411 (6833), 41–42 (Чонг, Х., Мейсон, С. П., Барабаши, А.-Л. и Олтвай, З. Н. Летальность и центральность в белковых сетях).
- Jernigan, C., and Mistree, B. F. T. 2009. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14 (10) (Джерниган, К. и Мистри, Б. Ф. Т. Гейдар: Дружба в Facebook разоблачает сексуальную ориентацию).
- Kamada, T., and Kawai, S. 1989. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31 (1), 7–15 (Камада, Т. и Каваи, С. Алгоритм строительства общих ненаправленных графов).
- Karrer, B., and Newman, M. E. J. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83 (1), 016107 (Каррер, Б. и Ньюман, М. Э. Дж. Стохастические блок-модели и структура в форме сообщества в сетях).
- Kempe, D., Kleinberg, J., and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146 (Кемпе, Д., Кляйнберг, Дж. и Тардос, Э. Максимизирование распространения влияния через социальную сеть).
- Kernighan, B. W., and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49 (2), 291–307 (Керниган, Б. У. и Лин, С. Эффективная эвристическая процедура для деления графов).
- Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. 2010. Identification of influential spreaders in complex networks. *Nature Physics*, 6 (11), 888–893 (Кицак, М., Галлос, Л. К., Хавлин, С., Лильерос, Ф., Мучник, Л., Стенли, Х. Е. и Макс, Х. А. Выявление влиятельных распространителей в сложных сетях).
- Kivelä, M., Arenas, A., Barthélemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. 2014. Multilayer networks. *Journal of Complex Networks*, 2 (3), 203–271 (Кивеля, М., Арена, А., Бартелеми, М., Глисон, Дж. П., Морено, Ю. и Портер, М. А. Многослойные сети).
- Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5), 604–632 (Кляйнберг, Дж. М. Авторитетные источники в гиперсвязанной среде).
- Kleinberg, J. M. 2000. Navigation in a small world. *Nature*, 406 (6798), 845 (Кляйнберг, Дж. М. Навигация в малом мире).

- Kleinberg, J. M. 2002. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems: Proceedings of the First 12 Conferences*, pp. 431–438 (Кляйнберг, Дж. М. Маломировые явления и динамика информации).
- Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. 1999. The web as a graph: Measurements, models, and methods. In *Computing and Combinatorics: Proceedings of the 5th Annual International Conference*, pp. 1–17 (Клейнберг, Дж. М., Кумар, Р., Рагхаван, П., Раджагопалан, С. и Томкинс, А. С. Паутина в форме графа: измерения, модели и методы).
- Krapivsky, P. L., and Redner, S. 2001. Organization of growing random networks. *Physical Review E*, 63 (6), 066123 (Крапивский П. Л. и Реднер С. Организация растущих случайных сетей).
- Krapivsky, P. L., and Redner, S. 2003. Dynamics of majority rule in two-state interacting spin systems. *Physical Review Letters*, 90 (23), 238701 (Крапивский П. Л. и Реднер С. Динамика правила большинства во взаимодействующих спиновых системах с двумя состояниями).
- Krapivsky, P. L., Redner, S., and Leyvraz, F. 2000. Connectivity of growing random networks. *Physical Review Letters*, 85 (21), 4629–4632 (Крапивский П. Л., Реднер С. и Лейвраз Ф. Соединенность растущих случайных сетей).
- Lancichinetti, A., and Fortunato, S. 2009. Community detection algorithms: A comparative analysis. *Physical Review E*, 80 (5), 056117 (Ланчинетти, А. и Фортунато, С. Алгоритмы обнаружения сообществ: сравнительный анализ).
- Lancichinetti, A., Fortunato, S., and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78 (4), 046110 (Ланчинетти, А., Фортунато, С. и Радикки, Ф. Эталонные графы для тестирования алгоритмов обнаружения сообществ).
- Latora, V., Nicosia, V., and Russo, G. 2017. *Complex Networks: Principles, Methods and Applications*. Cambridge University Press: Cambridge (Латора, В., Никосия, В. и Руссо, Г. Сложные сети: принципы, методы и приложения).
- Lazarsfeld, P. F., Merton, R. K., et al. 1954. Friendship as a social process: A substantive and methodological analysis. *Freedom and Control in Modern Society*, 18 (1), 18–66 (Лазарсфельд, П. Ф., Мертон, Р. К. и соавт. Дружба как социальный процесс: содержательный и методологический анализ).
- Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., et al. 2018. The science of fake news. *Science*, 359 (6380), 1094–1096 (Лазер, Д. М. Дж., Баум, М. А., Бенклер, Ю., Беринский, А. Дж., Гринхилл, К. М., Менцер, Ф. и соавт. Наука о фальшивых новостях).
- Liljeros, F., Edling, C. R., Amaral, L. A. N., Stanley, H. E., and Åberg, Y. 2001. The web of human sexual contacts. *Nature*, 411, 907–908 (Лильерос, Ф., Эдлинг, К. Р., Амарал, Л. А. Н., Стенли, Х. Е. и Обер, Ю. Паутина человеческих сексуальных контактов).

- Liu, B. 2011. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2nd edn. Springer: New York (Лю, Б. Добыча знаний из данных Всемирной паутины: разведка гиперсвязей, содержимого и данных об использовании, 2-е изд).
- Luccio, F., and Sami, M. 1969. On the decomposition of networks into minimally interconnected networks. *IEEE Transactions on Circuit Theory*, 16 (2), 184–188 (Луччо, Ф. и Сами, М. О разложении сетей на минимально взаимосвязанные сети).
- Luce, R. D., and Perry, A. D. 1949. A method of matrix analysis of group structure. *Psychometrika*, 14 (2), 95–116 (Люс, Р. Д., и Перри, А. Д. Метод матричного анализа групповой структуры).
- Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press: Cambridge (Мэннинг, К. Д., Рагхаван, П. и Шютце, Х. Введение в информационный поиск).
- Marchiori, M. 1997. The quest for correct information on the web: Hyper search engines. *Computer Networks and ISDN Systems*, 29 (8–13), 1225–1235 (Маркиори, М. В поисках правильной информации в паутине: гиперпоисковые машины).
- McPherson, M., Smith-Lovin, L., and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27 (1), 415–444 (Макферсон, М., Смит-Ловин, Л. и Кук, Дж. М. Птицы одного полета: гомофилия в социальных сетях).
- Meilă, M. 2007. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98 (5), 873–895 (Мейла, М. Сравнение кластеризаций – расстояние на основе информации).
- Meiss, M., Menczer, F., Fortunato, S., Flammini, A., and Vespignani, A. 2008. Ranking web sites with real user traffic. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 65–75 (Мейс, М., Менцер, Ф., Фортунато, С., Фламмини, А. и Веспиньяни, А. Ранжирование веб-сайтов с реальным пользовательским трафиком).
- Meiss, M., Gonçalves, B., Ramasco, J., Flammini, A., and Menczer, F. 2010. Modeling traffic on the web graph. In *Proceedings of the 7th Workshop on Algorithms and Models for the Web Graph (WAW)*, pp. 50–61 (Мейс, М., Гонсалвес, Б., Рамаско, Дж., Фламмини, А. и Менцер, Ф. Моделирование трафика на графе Всемирной паутины).
- Melián, C. J., and Bascompte, J. 2004. Food web cohesion. *Ecology*, 85 (2), 352–358 (Мелиан, К. Дж. и Баскомпте, Дж. Когезия пищевой паутины).
- Menczer, F. 2002. Growing and navigating the small world web by local content. *Proceedings of the National Academy of Sciences of the USA*, 99 (22), 14014–14019 (Менцер, Ф. Развитие и навигация в малой Всемирной паутине посредством локального контента).
- Menczer, F. 2004. Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55 (14),

- 1261–1269 (Менцер, Ф. Лексическая и семантическая кластеризация по связям в паутине).
- Meusel, R., Vigna, S., Lehmberg, O., and Bizer, C. 2015. The graph structure in the web — analyzed on different aggregation levels. *Journal of Web Science*, 1 (1), 33–47 (Мейзель, Р., Винья, С., Лемберг, О. и Бизер, С. Графовая структура в паутине – анализ на разных уровнях агрегации).
- Mobilia, M., Petersen, A., and Redner, S. 2007. On the role of zealotry in the voter model. *Journal of Statistical Mechanics: Theory and Experiment*, P08029 (Мобилия, М., Петерсен, А. и Реднер, С. О роли фанатизма в модели на основе избирателя).
- Molloy, M., and Reed, B. 1995. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6 (2–3), 161–179 (Моллой, М. и Рид, Б. Критическая точка для случайных графов с заданной степенной последовательностью).
- Moore, E. F. 1959. The shortest path through a maze. In *Proceedings of the International Symposium on Switching Theory 1957, Part II*, pp. 285–292 (Мур, Э. Ф. Кратчайший путь через лабиринт).
- Moreno, J. L., and Jennings, H. H. 1934. *Who Shall Survive? Nervous and Mental Disease Publishing Co.: New York* (Морено, Дж. Л. и Дженнингс, Х. Х. Кто Выживет?).
- Newman, M. E. J. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the USA*, 98 (2), 404–409 (Ньюман, М. Э. Дж. Структура научно-коллаборационных сетей).
- Newman, M. E. J. 2002. Assortative mixing in networks. *Physical Review Letters*, 89 (20), 208701 (Ньюман, М. Э. Дж. Ассортативное смешивание в сетях).
- Newman, M. E. J. 2004a. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69 (6), 066133 (Ньюман, М. Э. Дж. Быстрый алгоритм определения структуры в форме сообщества в сетях).
- Newman, M. E. J. 2004b. Analysis of weighted networks. *Physical Review E*, 70 (5), 056131 (Ньюман, М. Э. Дж. Анализ взвешенных сетей).
- Newman, M. 2018. *Networks*, 2nd edn. Oxford University Press: Oxford (Ньюман, М. Сети. 2-е изд.).
- Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E*, 69 (2), 026113 (Ньюман, М. Э. Дж. и Гирван, М. Отыскание и оценивание структуры в форме сообщества в сетях).
- Pariser, E. 2011. *The Filter Bubble: What the Internet is Hiding From You*. Penguin: Harmondsworth (Паризер, Э. Фильтерный пузырь: что интернет от вас скрывает).
- Pastor-Satorras, R., and Vespignani, A. 2001. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86 (14), 3200–3203 (Пастор-Саторрас, Р. и Веспиньяни, А. Распространение эпидемий в безмасштабных сетях).
- Pastor-Satorras, R., and Vespignani, A. 2007. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University

- Press: Cambridge (*Пастор-Саторрас, Р. и Веспиньяни, А. Эволюция и структура интернета: подход на основе статистической физики*).
- Pastor-Satorras, R., Vázquez, A., and Vespignani, A. 2001. Dynamical and correlation properties of the Internet. *Physical Review Letters*, 87 (25), 258701 (*Пастор-Саторрас, Р., Васкес, А. и Веспиньяни, А. Динамические и корреляционные свойства интернета*).
- Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A. 2015. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87 (3), 925–979 (*Пастор-Саторрас, Р., Кастеллано, С., Ван Мигем, П. и Веспиньяни, А. Эпидемические процессы в сложных сетях*).
- Peixoto, T. P. 2012. Entropy of stochastic blockmodel ensembles. *Physical Review E*, 85 (5), 056122 (*Пейшото, Т. П. Энтропия стохастических блокмоделей*).
- Peixoto, T. P. 2014. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4 (1), 011047 (*Пейшото, Т. П. Иерархические блочные структуры и отбор моделей с высокой разрешающей способностью в крупных сетях*).
- Porter, M. A., Onnela, J.-P., and Mucha, P. J. 2009. Communities in networks. *Notices of the American Mathematical Society*, 56 (9), 1082–1097 (*Портер, М. А., Оннела, Дж.-П. и Муха, П. Дж. Сообщества в сетях*).
- Price, D. D. 1976. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society of Information Science*, 27 (5), 292–306 (*Прайс, Д. Д. Общая теория библиометрических и других процессов кумулятивного преимущества*).
- Radicchi, F. 2015. Percolation in real interdependent networks. *Nature Physics*, 11 (7), 597–602 (*Радикки, Ф. Просачивание в реально существующих взаимозависимых сетях*).
- Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. 2004. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the USA*, 101 (9), 2658–2663 (*Радикки, Ф., Кастеллано, С., Чеккони, Ф., Лорето, В. и Паризи, Д. Определение и выявление сообществ в сетях*).
- Raghavan, U. N., Albert, R., and Kumara, S. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76 (3), 036106 (*Разгаван, У. Н., Альберт Р. и Кумара, С. Почти линейно-временной алгоритм обнаружения структур в форме сообществ в крупномасштабных сетях*).
- Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Flammini, A., and Menczer, F. 2011. Detecting and tracking political abuse in social media. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pp. 297–304 (*Раткевич, Дж., Коновер, М., Мейс, М., Гонсалвес, Б., Фламмини, А. и Менцер, Ф. Выявление и отслеживание политических злоупотреблений в социальных сетях*).
- Reichardt, J., and Bornholdt, S. 2006. Statistical mechanics of community detection. *Physical Review E*, 74 (1), 016110 (*Рейхардт, Дж., и Борнхольдт, С. Статистическая механика обнаружения сообществ*).

- Reis, S. D. S., Hu, Y., Babino, A., Andrade Jr., J. S., Canals, S., Sigman, M., and Makse, H. A. 2014. Avoiding catastrophic failure in correlated networks of networks. *Nature Physics*, 10 (10), 762–767 (Рейс, С. Д. С., Ху, Ю., Бабино, А., Андраде-мл., Дж. С., Каналы, С., Сигман, М. и Макс, Х. А. Предотвращение катастрофических сбоев во взаимосвязанных сетях сетей).
- Rossi, R. A., and Ahmed, N. K. 2015. The network data repository with interactive graph analytics and visualization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 4292–4293 (Росси, Р. А. и Ахмед, Н. К. Сетевое хранилище данных с интерактивной графовой аналитикой и визуализацией).
- Rossi, R. A., Fahmy, S., and Talukder, N. 2013. A multi-level approach for evaluating Internet topology generators. In *Proceedings of the IFIP Networking Conference*, pp. 1–9 (Росси, Р. А., Фахми, С. и Талукдер, Н. Многоуровневый подход для оценивания генераторов топологии интернета).
- Seeley, J. R. 1949. The net of reciprocal influence: A problem in treating sociometric data. *Canadian Journal of Experimental Psychology*, 3 (4), 234–240 (Сили, Дж. Р. Сеть взаимного влияния: проблема трактовки социометрических данных).
- Serrano, M., Maguitman, A., Boguñá, M., Fortunato, S., and Vespignani, A. 2007. Decoding the structure of the WWW: A comparative analysis of Web crawls. *ACM Transactions on the Web*, 1 (2), 10 (Серрано, М., Магитман, А., Богуна, М., Фортунато, С. и Веспиньяни, А. Расшифровка структуры WWW: сравнительный анализ обходов паутины).
- Serrano, M. Á., Boguñá, M., and Vespignani, A. 2009. Extracting the multi-scale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences of the USA*, 106 (16), 6483–6488 (Серрано, М. А., Богуна, М. и Веспиньяни, А. Извлечение многомасштабной магистралей сложных взвешенных сетей).
- Shao, C., Hui, P.-M., Wang, L., Jiang, X., Flammini, A., Menczer, F., and Ciampaglia, G. L. 2018a. Anatomy of an online misinformation network. *PLoS ONE*, 13 (4), e0196087 (Шао, С., Хуэй, П.-М., Ван, Л., Цзян, Х., Фламмини, А., Менцер, Ф. и Чампалья, Г. Л. Анатомия онлайн-сети дезинформационной сети).
- Shao, C., Ciampaglia, G. L., Varol, O., Yang, K., Flammini, A., and Menczer, F. 2018b. The spread of low-credibility content by social bots. *Nature Communications*, 9, 4787 (Шао, С., Чампалья, Г. Л., Варол, О., Ян, К., Фламмини, А. и Менцер, Ф. Распространение контента с низким уровнем доверия социальными ботами).
- Shimbel, A. 1955. Structure in communication nets. In *Proceedings of the Symposium on Information Networks*, pp. 199–203 (Шимбел, А. Структура в коммуникационных сетях).
- Solé, R. V., Pastor-Satorras, R., Smith, E., and Kepler, T. B. 2002. A model of large-scale proteome evolution. *Advances in Complex Systems*, 5 (01), 43–54 (Соле, Р. В., Пастор-Саторрас, Р., Смит, Э. и Кеплер, Т. Б. Модель крупномасштабной эволюции протеома).

- Solomonoff, R., and Rapoport, A. 1951. Connectivity of random nets. The Bulletin of Mathematical Biophysics, 13 (2), 107–117 (Соломонов Р. и Рапопорт А. Соединенность случайных сетей).
- Sporns, O. 2012. Discovering the Human Connectome. MIT Press: Boston, MA (Спорнс, О. Обнаружение человеческого коннектома).
- Spring, N., Mahajan, R., and Wetherall, D. 2002. Measuring ISP topologies with Rocketfuel. In ACM SIGCOMM Computer Communication Review, pp. 133–145 (Спринг, Н., Махаджан, Р. и Уэзеролл, Д. Измерение топологии ISP с помощью ракетного топлива).
- Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., et al. 2011. High-resolution measurements of face-to-face contact patterns in a primary school. PLoS ONE, 6 (8), e23176 (Штеле, Дж., Вуарин, Н., Баррат, А., Камтутто, С., Изелла, Л., Пинтон, Дж.-Ф. и соавт. Измерения с высокой разрешающей способностью моделей межличностных контактов в начальной школе).
- Stonedahl, F., and Wilensky, U. 2008. NetLogo Virus on a Network Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/VirusonaNetwork> (Стоундал, Ф. и Виленски, У. Модель «Вирус в сети» в приложении NetLogo).
- Stonedahl, F., and Wilensky, U. 2009. NetLogo PageRank Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/PageRank> (Стоундал, Ф. и Виленски, У. Модель «PageRank» в приложении NetLogo).
- Sunstein, C. R. 2001. Echo Chambers: Bush v. Gore, Impeachment, and Beyond. Princeton University Press: Princeton, NJ (Санштейн, С. Р. Эхо-камеры: Буш против Гора, импичмент и далее).
- Travers, J., and Milgram, S. 1969. An experimental study of the small world problem. Sociometry, 32 (4), 425–443 (Трэверс, Дж. и Милграм, С. Экспериментальное исследование задачи о малом мире).
- Troutman, C., and Wilensky, U. 2007. NetLogo Language Change Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/LanguageChange> (Трутман, С. и Виленски, У. Модель «Изменение языка» в приложении NetLogo).
- Ulanowicz, R. E., and DeAngelis, D. L. 1998. Network analysis of trophic dynamics in South Florida ecosystems. FY97: The Florida Bay Ecosystem, 20688–20038 (Уланович, Р. Е. и ДеАнджелис, Д. Л. Сетевой анализ трофической динамики в экосистемах Южной Флориды).
- Vázquez, A. 2003. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. Physical Review E, 67 (5), 056104 (Васкес, А. Выраживание сети с локальными правилами: преференциальное прикрепление, иерархия кластеризации и степенные корреляции).
- Vázquez, A., Flammini, A., Maritan, A., and Vespignani, A. 2003a. Modeling of protein interaction networks. Complexus, 1 (1), 38–44 (Васкес, А.,

- Фламмини, А., Маритан, А. и Веспиньяни, А. Моделирование сетей белкового взаимодействия).
- Vázquez, F., Krapivsky, P. L., and Redner, S. 2003b. Constrained opinion dynamics: Freezing and slow evolution. *Journal of Physics A: Mathematical and General*, 36 (3), L61–L68 (Васкес Ф., Крапивский П. Л. и Реднер С. Ограниченная динамика мнений: замораживание и медленная эволюция).
- Vosoughi, S., Roy, D., and Aral, S. 2018. The spread of true and false news online. *Science*, 359 (6380), 1146–1151 (Восуги, С., Рой, Д. и Арал, С. Распространение истинных и ложных новостей в онлайн-режиме).
- Wagner, A. 1994. Evolution of gene networks by gene duplications: A mathematical model and its implications on genome organization. *Proceedings of the National Academy of Sciences of the USA*, 91 (10), 4387–4391 (Вагнер, А. Эволюция генных сетей путем дупликации генов: математическая модель и ее влияние на организацию генома).
- Wasserman, S., and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press: Cambridge (Вассерман, С. и Фауст, К. Социально-сетевой анализ: методы и приложения).
- Watts, D. J. 2002. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences of the USA*, 99 (9), 5766–5771 (Уоттс, Д. Дж. Простая модель глобальных каскадов на случайных сетях).
- Watts, D. J. 2004. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Co.: New York (Уоттс, Д. Дж. Шесть степеней: наука подсоединенной эпохи).
- Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of ‘small-world’ networks. *Nature*, 393 (6684), 440–442 (Уоттс, Д. Дж. и Стрോഗатц, С. Х. Коллективная динамика «маломировых» сетей).
- Watts, D. J., Dodds, P. S., and Newman, M. E. J. 2002. Identity and search in social networks. *Science*, 296 (5571), 1302–1305 (Уоттс, Д. Дж., Доддс, П. С. и Ньюман, М. Э. Дж. Идентичность и поиск в социальных сетях).
- Weng, L., Ratkiewicz, J., Perra, N., Gonçalves, B., Castillo, C., Bonchi, F., et al. 2013a. The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 356–364 (Венг, Л., Раткевич, Дж., Перра, Н., Гонсалвес, Б., Кастильо, С., Бончи, Ф. и соавт. Роль информационной диффузии в эволюции социальных сетей).
- Weng, L., Menczer, F., and Ahn, Y.-Y. 2013b. Virality prediction and community structure in social networks. *Scientific Reports*, 3, 2522 (Венг, Л., Менцер, Ф. и Ан, Ю.-Ю. Предсказание вирусности и структура в форме сообществ в социальных сетях).
- White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. 1986. The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London Series B, Biological Science*, 314 (1165), 1–340 (Уайт, Дж. Дж., Саутгейт, Э.,

- Томсон, Дж. Н. и Бреннер, С. Строение нервной системы нематоды *Caenorhabditis elegans*).
- Wilensky, U. 1999. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/> (Виленски, У. Приложение NetLogo).
- Wilensky, U. 2005a. NetLogo Giant Component Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/GiantComponent> (Виленски, У. Модель «Гигантская компонента» в приложении NetLogo).
- Wilensky, U. 2005b. NetLogo Preferential Attachment Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/PreferentialAttachment> (Виленски, У. Модель «Преференциальное прикреплениe» в приложении NetLogo).
- Wilensky, U. 2005c. NetLogo Small Worlds Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/SmallWorlds> (Виленски, У. Модель «Малые Миры» в приложении NetLogo).
- Xu, R, and Wunsch, D. 2008. Clustering. Wiley: Piscataway, NJ (Сюй, Р и Вунш, Д. Кластеризация).
- Yang, J., and Leskovec, J. 2012. Defining and evaluating network communities based on ground-truth. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS '12), pp. 3:1–3:8 (Янг, Дж. и Лесковец, Дж. Определение и оценивание сетевых сообществ на основе достоверных эмпирических данных).
- Zachary, W. W. 1977. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33 (4), 452–473 (Закари, У. У. Модель информационного потока для конфликтов и раскола в малых группах).



Предметный указатель



F

Facebook, 12, 20, 35, 41, 58, 79, 83, 99, 119, 122, 137, 143, 144, 177
соцсеть, 18, 21

G

Gephi, инструмент визуализации графов, 52
Google, 10, 12, 39, 125, 133, 152, 185

H

HTML, 24, 122
HTTP, 24, 122

I

IMDB, интернет-база данных, 20
Instagram, 143

M

Matplotlib, графопостроительная библиотека, 52

N

NetworkX, пакет, 37
манипулирование сетями, 36

P

PageRank
случайное блуждание, 133
степенной метод, 134
телепортация, 133
PageRank, метрика, 25, 132

S

scipy, пакет, 61
StackOverflow, 39

T

Twitter, 12, 18, 23, 47, 58, 82, 83, 100, 108, 119, 122, 138, 141, 177
граф, 103
ретвитная сеть, 137
соцсеть, 21, 23
API, 13

U

URL, 121, 123, 125

W

WhatsApp, 143

A

Алгоритм
Гирвана–Ньюмана, 216
жадного поиска, 274
жадной маршрутизации, 272
Кернигана–Лина, 211
компоновки по направлению силы, 51
компоновки сети, 51
Лувена, 222
обнаружения сообществ, 27, 215
определения дерева, 70
определения связности сети, 67
поиска сперва в ширину, 71, 72
разложения ядра, 111
распространения меток, 225
рекомендательный, патологические последствия, 86
Алгоритм ранжирования, 132
Аппроксимация гомогенного перемешивания, 256
Ассортативность, 58, 228, 267
степенная, 60
Астротурф, 142

Атрибут

- весовой, 46
- категориальный, 60
- связи, 49, 52
- узла, 60, 72
- числовой, 60

**Б**

- Бисекция графа, 210
 - разрез, 210
- Близость, мера центральности, 95
- Бот социальный, 142

В

- Веб, 24
 - граф, 125
 - обходчик, 71, 124
 - трафик, 147
- Вероятность
 - влияния, 250
 - выхода, 262
 - остановки, 260
 - передачи, 260
 - селекции, 268
- Вес, 30, 131, 137
- Взаимодействие межсетевое, 26
- Взаимоотношение, 17, 18
- Взаимоотношение социальное, 31
- Викигонки, онлайнная игра, 80
- Википедия, 24, 35, 100, 108
 - граф, 103
- Вирусность мема, 140
- Влияние, 142
 - каскады, 246
 - социальное, 59, 267
- Влиятель, 246, 249
- Возможность возникновения, 228
- Время доставки, 274
- Встречаемость совместная, 143
- Выуживание конфиденциальной информации, 20

**Г**

- Галстук-бабочка, 127
- Гетерогенность, 94, 100
 - весовая, 147
- Гистограмма, 101
- Гомофилия, 22, 59, 129, 267
 - географическая, 274
- Граница продвижения, 125
 - очередь, 73

- Граница уверенности, 266
- Грановеттер, Марк С., 188
- Граф
 - Всемирной паутины, 125
 - случайный экспоненциальный, 176
 - эталонный, 230

Д

- Давление со стороны сверстников, 250
- Дезинформация, 138
 - распространение, 140
- Деление сети на разделы, 210
 - размер разреза, 210
- Дендрограмма, 214
- Дерево, 69
 - иерархическое, кластеризация, 214
 - корень, 70
 - лист, 70
 - ретвитного каскада, 139
 - тематических расстояний, 276
- Диаметр, 64
- Диффузия дезинформации, астротурф, 142
- Доступность
 - географического поиска, 274
 - поиска, 273
 - тематического поиска, 276
- Дрейф тематический, 130
- Друг моего друга, 81

З

- Заглушка, 174
- Замыкание триадическое, 83, 86
 - сильное, 187
- Заражение
 - сложное, 252
 - социальное, 246

И

- Имплементация асинхронная, 251
- Индикатор влияния косвенный, 142
- Инструмент сетевого анализа, 71
- Интенсивность взаимодействия, 64
- Интернет, 26
 - межсетевое взаимодействие, 26
- Интернет-база данных кинфильмов, 20
- Инфекция вторичная, 257
- Информация, 138
 - нормализованная взаимная, 235
 - шаблоны производств и потребления, 141

Источник, проверяющий факты
на достоверность, 140

К

Каринти, Фригьеш, 78

Касакад

влияния, 246

глобальный, 246

независимый, 246

ретвитный, 84, 138

Категория тематическая, 276

Кластер, 27

Кластеризация, 81

данных, 212

дендрограмма, 214

иерархическая, 213

иерархическая агломеративная, 214

иерархическая дивизивная, 214

одиночное соединение, 213

полное соединение, 213

раздельная, 213

среднее соединение, 213

Клика, 42, 205

Ключ уникальный, 272

Когезия, 98, 205, 209, 212, 224

Компонента, 67

гигантская, 68, 69, 164

связная, 67

сильно связная, 68

слабо связная, 68

Компонента-на-входе, 69, 127

Компонента-на-выходе, 69, 127

Компоновка, 200

алгоритм, 51

по направлению силы, 51, 202

Консенсус, 262, 265

Корреляция Пирсона, измерение

ассортативности, 61

Козволюция, модель, 268

Коэффициент

ассортативности

корреляция Присона, 61

средняя степень узла, 61

k ближайших соседей, 61

кластеризации, 81, 169

корреляции Пирсона, 61

Л

Ландшафт тематический, 275

Лес, 139

Лимит разрешающей способности, 224



Логарифм числа, 80

Локальность тематическая, 129, 275

Магистраль сетевая, 150

Манипулирование сетями в исходном
коде, 36

Маркони, Гульельмо, 78

Маршрутизатор сетевой, 26

Материал для дальнейшего изучения, 31,
53, 85, 113, 153, 194, 237, 280

Матрица

смежности, 42, 49

стохастическая блочная, 227

Машина поисковая, 133

Мем, 138

вирусность, 140

Мера

различия, 213

расстояния агрегатная, 64

сходства, 213

центральности

близость, 95

промежуточность, 96

степень, 95

Метрика PageRank, 132

Милграм, Стэнли, 78

Мир

малый, 58, 78, 170

ультрамалый, 107, 129

Мнение

адаптация, 268

дискретное, 262

непрерывное, 265

Мнение среднее, 262

Моделирование стохастическое

блочное, 227

Модель, 244

PageRank, 134

SIR, 254

SIS, 254

Барабаси–Альберта, 179

восприимчивый–инфицированный–

восприимчивый (SIS), 254

восприимчивый–инфицированный–

выздоровевший, 255

Гилберта, 163

коэволюционная, 268

на основе

большинства, 263

внедрения разделов, 230

избирателя, 263
 копирования, 190
 ограниченной уверенности, 266
 привлекательности, 184
 приспособленности, 185
 ранга, 191
 случайного блуждания, 187
 независимо-каскадная, 250
 вероятностность, 252
 пороговая, 247
 давление со стороны
 сверстников, 250
 детерминированность, 252
 пороговая линейная, 247
 распространения слуха, 259
 сетевая, 162
 конфигурационная, 174
 маломирная, 170
 случайная сеть, 162
 стохастическая блочная, 227
 откорректированная по степени, 229
 урновая Пойи, 180
 Эрдеша–Рени, 163
 Модулярность, 218, 220
 лимит разрешающей способности, 224
 несколько разрешающих
 способностей, 225
 оптимизация, 218
 Мост, 98, 216

Н

Наложение, 208
 Направление, 119
 Несогласие, 262

О

Обнаружение сообществ, 215
 оптимизация модулярности, 218
 распространение меток, 225
 стохастическое блочное
 моделирование, 227
 устранение мостов, 216
 Обходчик Всемирной паутины, 71, 124
 Объем сообщества, 204
 Оптимизация модулярности, 218
 с несколькими разрешающими
 способностями, 225
 Оптимум локальный, 212
 Оракул Бейкона, игровое онлайн-овое
 приложение, 77
 Орграф, 38

Отыскание кратчайшего пути, 71
 Очередь, структура данных FIFO, 72

П

Пара влиятель–сосед, 251
 Парадокс дружбы, 104
 Параметр гетерогенности, 100, 102
 Паутина Всемирная, 24, 120
 краткая история, 121
 принцип работы, 122
 трафик, 147
 Плотность, 39, 67, 165
 внутренних связей, 204
 внутренняя, 204
 Победитель получает все, 183, 186
 Подграф, 42
 Подсеть, 42
 Подход эгалитарный, 163
 Поиск, 270
 исчерпывающий, 270
 локальный, 270
 сперва в ширину, 71, 270
 граница продвижения, 73
 имплементация, 72
 Покрытие, 208
 Покупка фейковых подписчиков, 142
 Поляризация, 262, 265
 Порог, 246
 эпидемический, 258
 Последовательность степенная, 174
 Почта электронная, 23
 Правдоподобие логарифмическое, 229
 Прайс, Дерек де Солла, 184
 Представление сетей, 49
 Преимущество кумулятивное, 180
 Признак сети, 25, 40, 58, 94
 Признак узла, 30
 Прикрепление преференциальное
 модель, 177
 нелинейное, 182
 принцип, 180
 эффект Мэттью, 180
 Принцип Грановеттера
 сильное триадическое замыкание, 187
 Принцип ограниченной уверенности, 266
 Промежуточность
 мера центральности, 96
 распределение, 101
 связь, 98
 узел, 96
 Пузырь фильтерный, 85, 201

**Путь, 62**

- длина, 62
- короткий, 168
- кратчайший, 62, 97
- дерево, 71
- длина, 62
- средняя длина, 64
- простой, 62
- средняя длина, 64
- эйлеров, 63

Р

- Раздел, 207
- Разложение ядра, 110
- Размер сети, 35
- Разреженность, 39, 41
- Разрез минимальный, 210
- Рандомизация с сохранением степени, 219
- Распределение
 - биномиальное, 167
 - значений вероятности, 167
 - колоколообразное, 167
 - кумулятивное, 101
 - промежуточностей, 101
 - статистическое, 101
 - степенное, 102, 166
 - центральности, 99
- Распространение
 - дезинформации
 - вирусность, 140
 - покупка фейковых подписчиков, 142
 - меток, 225
- Расстояние, 62
 - коллаборационное, 91
 - социальное, 75
 - тематическое, 275
 - дерево, 276
- Рени, Альфред, 163
- Рисование сетей, 51
- Рыбак рыбака видит издалека, 58

С

- Самоцикл (самонаправленный цикл), 49
- Связность, 67
- Связь, 17, 30
 - взвешенная, 30
 - направленная, 119
 - направленная (ориентированная), 30
 - отбор, 189
 - совместного упоминания
 - источника, 120

- совместного цитирования, 120
- ферма, 136
- цитирования, 144
- Селекция, 267
- Семь мостов Кенигсберга, 63
- Сеть
 - авиационных перевозок, 27, 94
 - белковых взаимодействий, 29
 - биологическая, 29
 - взвешенная, 30, 45, 137
 - «Википедия», 24
 - Всемирная паутина, 24
 - генно-регуляторная, 29
 - двудольная, 36, 144
 - динамика, 29
 - диссортативная, 60
 - диффузии дезинформации, 142
 - диффузии информации, 138
 - диффузионная, 138, 139, 141
 - дорожная, 27
 - иерархическая, 70
 - Интернет, 26
 - информационная, 24
 - клубов каратэ «Закари», 215, 217, 221, 233
 - коллаборационная, 75, 163
 - коммуникационная, 21
 - контактная, 254
 - магистраль, 150
 - маршрутизаторов, 26
 - метаболическая, 29
 - многораздельная, 228
 - многослойная, 46
 - мультиплексная, 36, 47
 - стыки, 47, 49
 - направленная, 44, 119
 - направленная (ориентированная), 30
 - невзвешенная, 30
 - нейронная, 29
 - ненаправленная, 30
 - несвязная, 67
 - оверлейная, 272
 - одноранговая, 270
 - определение понятия, 35
 - полная, 39
 - размер, 30, 35
 - ретвитная, 22, 84, 111, 137, 140
 - решетчатая, 80
 - рост, 169, 178
 - связная, 67
 - сеть сетей, 26, 48
 - сильно связная, 68
 - слабо связная, 69

случайная, 162
соавторства, 75, 95
совместного упоминания
источников, 144
совместного цитирования, 144
совместной встречаемости, 143
социальная, 18
структура, 29
темпоральная, 47
транспортная, 27
цитирования, 120
эгосеть, 42, 76
экологическая, 29
Сигнатура
популярности страницы, 128
ультрамалого мира, 129
Сила, 45, 137
сила-на-входе, 45, 137, 141
сила-на-выходе, 45, 137, 141
Синглетон, 68
Система автономная, 26
Содержимое информационное, 62
Создание видимости предвыборных
кампаний, астротурф, 142
Сокращение, 171, 172, 173, 274
Сообщество, 94, 200
обнаружение, 202
объем, 204
слабое, 206
степень, 203, 204
Сосед, 31
Состояние
стационарное, 226, 262
узла, 262
Спамдексирование, 136
Список ребер, 50
Список смежности, 50
Способность к установлению
соединения, 81
Степень, 43, 112, 119
ассортативность, 60
внешняя, 203, 204
внутренняя, 203, 204
корреляция, 60
мера центральности, 95
сообщества, 204
средняя, 43
Степень-на-входе, 45, 120
распределение, 100
Степень-на-выходе, 45, 120
распределение, 100
Сторонник фанатичный, 265

Строгац, Стивен Г., 170
Структура
в форме сообщества, 190, 202, 207, 208,
219, 227, 230, 232
Всемирной паутины, 127
диссортативная, 228
локальная, 81
решетчатая, 28
ядро-периферия, 27, 228
Стык, 47
Сходство, 129, 212
косинусное, 131
между разделами, 234
текстовое, 129
Схождение, 212

Т

Таблица маршрутная в одноранговой
сети, 272
Тегирование социальное, 144
Телепортация PageRank, 133
Тест отрицательный, 232
Толерантность, 266
Трафик, 96, 128, 147
Всемирная паутина, 147
Треугольник, 81
Триада, 81

У

Уверенность ограниченная, 266, 267
модель, 267
Узел, 17, 30
дочерний, 70
отбор, 180
родительский, 70
смежный, 34
уязвимый, 249
Узел-одиночка, 43, 65
Уоттс, Дункан Дж., 170
Упоминание источника совместное, 144
Уровень базовый случайный, 218
Устойчивость, 108
атаки, 109
отказы, 108
Устранение мостов, 216

Ф

Ферма связей, 128
Фильтрация связей, 149
Фишинг социальный, 20
Фольксономия, 145

Фрагментация, 265

Функция

тематического затухания, 277

хеш, 272

Функция распределения, 101



Х

Хаб, 60, 94

Хеш-таблица распределенная, 272

Хеш-функция, 272

Ц

Центральность

на основе близости, 95

на основе промежуточности, 96

на основе степени, 95

Цикл, 62, 68, 70

самонаправленный, 49, 175

эйлеров, 63

Цитирование совместное, 120

Ч

Частота

выздоровления, 255

инфицирования, 255

Число

базовое репродукционное, 258

Белла, 207

Данбара, 169

Эрдеша, 75

Чунг, Фан, 76

Ш

Шесть степеней Кевина Бейкона, 76

Шесть степеней сепарации, пьеса, 78

Шкала логарифмическая, 80, 102

Э

Эйлер, Леонард, 34, 63

Эквивалентность структурная, 213

Энциклопедия онлайн, 24

Эпидемия, распространение, 252

Эрдеш, Пол, 75, 163

Эталон, 230

LFR, 232

Эффект

от Матфея, 180

пороговый, 258

Эхокамера, 59, 85, 140, 201, 270

Я

Ядро-периферия

оболочка, 110

разложение, 111

структура, 27, 60, 110

ядро, 60, 110



Книги издательства «ДМК ПРЕСС»
можно купить оптом и в розницу
в книготорговой компании «Галактика»
(представляет интересы издательств
«ДМК ПРЕСС», «СОЛОН ПРЕСС», «КТК Галактика»).

Адрес: г. Москва, пр. Андропова, 38;

тел.: **(499) 782-38-89**, электронная почта: **books@aliants-kniga.ru**.

При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.

Желательно также указать свой телефон и электронный адрес.

Эти книги вы можете заказать и в интернет-магазине: **<http://www.galaktika-dmk.com/>**.



Филиппо Менцер, Санто Фортунато и Клейтон А. Дэвис

Наука о сетях: вводный курс

Главный редактор *Мовчан Д. А.*
dmkpress@gmail.com

Зам. главного редактора *Сенченкова Е. А.*

Перевод *Логунов А. В.*

Корректор *Абросимова Л. А.*

Верстка *Чаннова А. А.*

Дизайн обложки *Бурмистрова Е. А.*

Гарнитура РТ Serif. Печать цифровая.

Усл. печ. л. 27,46. Тираж 200 экз.

Веб-сайт издательства: **www.dmkpress.com**

Сети присутствуют во всех аспектах нашей жизни. Круг друзей, коммуникационные и транспортные сети, а также Всемирная паутина – внешние сети для общения. Нейроны и белки в нашем мозге – внутренние сети, определяющие интеллект и способность к выживанию.

Данная книга знакомит с основами науки о сетях, необходимой в самых разных сферах знаний и деятельности – от менеджмента до маркетинга, от биологии до машиностроения. Читатели разовьют важные практические навыки и способность писать исходный код для использования сетей в интересующих их областях, даже если они только учатся программировать на Python. Такой интуитивный и прямой подход делает издание идеальным для самообразования.



В этой книге вы найдете:

- обзор сетей разных видов – от социальных до биологических;
- описание базовых сетевых элементов;
- рассмотрение роли сообществ и методов кластеризации;
- изучение Всемирной паутины как частного примера сети;
- практические упражнения для построения и анализа сетей.

Филиппо Менцер – профессор информатики и вычислительной техники в Университете Индианы, Блумингтон. Его исследования посвящены науке о сетях, вычислительному обществоведению с акцентом на противодействие манипуляциям в социальных сетях.

Санто Фортунато – директор Института науки о сетях, профессор информатики Университета Индианы, основатель и председатель Международной конференции по вычислительному обществоведению. Его интересы сосредоточены в области науки о сетях, в том числе на выявлении сетевых сообществ.

Клейтон А. Дэвис имеет степень доктора философии по информатике, а также степень бакалавра и магистра математики в Университете Индианы. Специализируется на разработке платформ больших данных для анализа социальных сетей и алгоритмах машинного обучения для борьбы с онлайн-злоупотреблениями.



Интернет-магазин:
www.dmkpress.com

Оптовая продажа:
КТК «Галактика»
books@aliants-kniga.ru

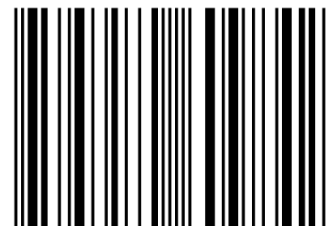


CAMBRIDGE
UNIVERSITY PRESS



www.dmk.pф

ISBN 978-5-97060-984-2



9 785970 609842 >