Оглавление

Предисловие

Миром правят связи – повсеместно, от финансовых и коммуникационных систем до социальных и биологических процессов. Выявление скрытого смысла этих связей приводит к прорывным решениям различных задач, таких как выявление мошеннических звонков, оптимизация связей в рабочей группе или прогнозирование каскадных сетевых сбоев.

Поскольку связность мира продолжает нарастать, неудивительно, что возрастает интерес к графовым алгоритмам, потому что они основаны на математике, специально разработанной для изучения взаимосвязей между данными. Анализ графов может раскрыть работу сложных систем и сетей в огромных масштабах – для любой организации.

Мы искренне увлечены полезностью и важностью анализа графов и с удовольствием расшифровываем тонкости внутренней работы сложных сценариев. До недавнего времени применение анализа графов требовало значительного опыта и упорства, потому что инструменты и средства интеграции были трудными в освоении, и лишь немногие знали, как применять графовые алгоритмы к своим задачам. Наша цель – помочь вам преодолеть трудности. Мы написали эту книгу, чтобы исследователи данных начали в полной мере использовать анализ графов, а значит, могли делать новые открытия и быстрее разрабатывать интеллектуальные решения.

О чем эта книга

Эта книга представляет собой практическое руководство по началу работы с графовыми алгоритмами для разработчиков и специалистов по анализу данных, которые имеют опыт использования Apache Spark™ или Neo4j. Хотя в наших примерах алгоритмов используются платформы Spark и Neo4j, эта книга также пригодится для изучения более общих понятий теории графов, независимо от вашего выбора графовых технологий.

Первые две главы содержат введение в теорию, анализ графов и графовые алгоритмы. В третьей главе кратко рассмотрены платформы, используемые в этой книге, прежде чем мы углубимся в следующие три главы, посвященные классическим графовым алгоритмам – нахождению пути, вычислению центральности и выделению сообществ. Мы завершим книгу двумя главами, показывающими, как графовые алгоритмы используются в рабочих процессах – один для общего анализа и другой для машинного обучения.

В начале описания каждой категории алгоритмов есть справочная таблица, которая поможет вам быстро выбрать нужный алгоритм. Для каждого алгоритма вы найдете:

- объяснение того, что делает алгоритм;
- примеры использования алгоритма и ссылки, где вы можете узнать больше;
- примеры кода, демонстрирующие способы реализации алгоритма в Spark и Neo4j.

Условные обозначения, принятые в книге

В книге имеются следующие условные обозначения:

Курсив

Используется для смыслового выделения важных положений, новых терминов, URL-адресов и адресов электронной почты в интернете, имен команд и утилит, а также имен и расширений файлов и каталогов.

`Моноширинный шрифт`

Используется для листингов программ, а также в обычном тексте для обозначения имен переменных, функций, типов, объектов, баз данных, переменных среды, операторов, ключевых слов и других программных конструкций и элементов исходного кода.

`Моноширинный полужирный шрифт`

Используется для обозначения команд или фрагментов текста, которые пользователь должен ввести дословно, без изменений.

`Моноширинный курсив`

Используется для обозначения в исходном коде или в командах шаблонных меток-заполнителей, которые должны быть заменены соответствующими контексту реальными значениями.

Такая пиктограмма обозначает совет или рекомендацию.

Обозначает указание или примечание общего характера.

Эта пиктограмма обозначает предупреждение или особое внимание к потенциально опасным объектам.

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 прямо на нашем сайте **www.dmkpress.com**, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу **dmkpress@gmail.com**, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы 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** или **www.дмк.рф** на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры, для того чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу **dmkpress@gmail.com**, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и O'Reilly очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты **dmkpress@gmail.com** со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Благодарности

Нам очень понравилось работать над этой книгой благодаря помощи наших друзей. Мы особенно благодарны Майклу Хангеру (Michael Hunger) за его руководство, Джиму Уэбберу (Jim Webber) за его бесценные правки и Томазу Братаничу (Tomaz Bratanic) за его увлекательные исследования. Наконец, мы искренне благодарим каталог Yelp, который позволяет нам использовать его богатый набор данных в качестве практических примеров.

Вступительное слово рецензента

Как вы думаете, что общего имеют следующие вещи: анализ маркетинговых взаимосвязей, противодействие отмыванию денег, моделирование поездок клиентов, анализ инцидентов безопасности, анализ литературных источников, обнаружение мошеннических сетей, анализ поисковых узлов в интернете, создание картографических приложений, анализ распространения эпидемий и изучение пьес Уильяма Шекспира? Как вы уже догадались, общим для всех них является использование графов, доказывающих, что Шекспир был прав, когда заявил: «Весь мир – это граф!»

Ну хорошо, великий бард с берегов Эйвона на самом деле не говорил про граф, он сказал «театр». Тем не менее обратите внимание, что приведенные выше примеры включают в себя сущности и отношения между ними, как прямые, так и косвенные. Сущности – вершины графа – могут быть людьми, событиями, объектами, концепциями или местами. Отношения между вершинами – это ребра графа. Но разве сама суть шекспировской пьесы не заключается в изображении героев (вершин) и их отношений (ребер)? Следовательно, Шекспир в своей знаменитой фразе имел полное право сказать про граф.

Что делает графовые алгоритмы и графовые базы данных такими интересными и мощными, так это не простые отношения между двумя сущностями, где *A* связан с *B*. В конце концов, стандартная реляционная модель баз данных использует эти типы отношений уже несколько десятилетий. Что на самом деле делает графы настолько удивительными – это *направленные и транзитивные отношения*. В направленных отношениях *A* может вызвать *B*, но не наоборот. В транзитивных отношениях *A* может быть непосредственно связан с *B*, а *B* может быть напрямую связан с *C*, в то время как *A* не имеет прямого отношения к *C*. Следовательно, A *транзитивно связан* с C.

Благодаря транзитивным отношениям – особенно когда они многочисленны и разнообразны, с различными паттернами отношений и степеней разделения между объектами – графовая модель раскрывает связи между объектами, которые в противном случае могут показаться не связанными или независимыми в обычной реляционной базе данных. Следовательно, во многих задачах сетевого анализа можно эффективно применять графовую модель.

Рассмотрим пример *маркетинговой атрибуции*[1] (marketing attribution): человек *A* видит маркетинговую кампанию; человек *A* пишет комментарий в социальных сетях; человек *B* связан с человеком *A* и видит комментарий; и впоследствии человек *B* покупает продукт. С точки зрения менеджера маркетинговой кампании, стандартная реляционная модель не может определить атрибуцию, поскольку персонаж *B* не видел кампанию, а персонаж A не купил продукт. Кампания выглядит как провал, но ее фактический успех (и положительная рентабельность инвестиций) обнаруживается алгоритмом анализа графов через транзитивные отношения между маркетинговой кампанией, посредником и конечной покупкой.

Далее рассмотрим пример противодействия отмыванию денег: лица *A* и *C* подозреваются в обороте денег, полученных незаконным путем. Любая прямая сделка между ними – например, транзакция через расчетно-кассовый центр – будет без труда зафиксирована и подвергнута строгому контролю. Однако, если *A* и *C* никогда не совершают взаимные сделки, а вместо этого проводят финансовые операции через надежные, уважаемые и незапятнанные финансовые органы *B*, что может провалить сделку? Алгоритм анализа графов! Графовый движок обнаружит транзитивные отношения между *A* и *C* через посредника *B*.

Отвечая на поисковый запрос, основные поисковые системы интернета используют алгоритмы на основе графов, чтобы найти самый авторитетный сайт во всем интернете для любого заданного набора поисковых слов. В этом случае принципиально важна направленность связей, поскольку авторитетным сайтом в сети является тот, на который ссылается большинство других сайтов.

Сегодня существует особая отрасль анализа данных – *литературный поиск* (literature-based discovery, LBD), технология на основе графов, позволяющая искать открытия в базе знаний тысяч (или даже миллионов) статей исследовательских журналов. Скрытые знания обнаруживаются только через связи между опубликованными результатами исследований, которые могут иметь много этапов переходных отношений. LBD активно применяется для поиска методов лечения рака, где обширная база семантических медицинских знаний о симптомах, диагнозах, методах лечения, взаимодействиях лекарств, генетических маркерах, краткосрочных результатах и долгосрочных последствиях может таить в себе ранее неизвестные или потенциально полезные методы лечения для самых безнадежных случаев. Это невероятно – знание уже может лежать в сети, и нам остается лишь соединить точки, чтобы найти его.

---

[1] Маркетинговая атрибуция – это анализ отдачи от точек взаимодействия с клиентом. Любимая поговорка маркетологов гласит: «Половина денег, которые я трачу на рекламу, выбрасывается впустую. Беда в том, что я не знаю, какая половина». Атрибуция призвана дать ответ на этот вопрос. – *Прим. перев.*

Подобные описания возможностей, реализуемых через построение графов, могут быть даны и для других упомянутых выше примеров – по сути, это примеры сетевого анализа с помощью графовых алгоритмов. Каждый случай глубоко исследует сущности (люди, объекты, события, действия, концепции и места) и их отношения (как причинно-следственные связи, так и простые ассоциации).

Обсуждая выгоды от построения графов, мы должны помнить, что в реальных ситуациях, возможно, самым мощным фактором в графовой модели является *контекст*. Он может включать время, местоположение, связанные события, соседние объекты и многое другое. Включение контекста в граф в виде вершин и ребер может дать впечатляющий толчок прогнозирующей и описательной аналитике.

Цель книги «*Графовые алгоритмы*» Марка Нидхема и Эми Ходлер – расширить наши знания и навыки в прикладном анализе графов, включая алгоритмы, концепции и практическое применение алгоритмов в машинном обучении. Авторы составили полезный и наглядный путеводитель по удивительному миру графов – от базовых концепций до фундаментальных алгоритмов, платформ обработки и практического использования.

Кирк Борн (Kirk Borne),
доктор наук, старший советник по науке и анализу данных,
консалтинговая компания Booz Allen Hamilton, Inc.
март 2019

Глава 1

Введение

«Графы являются одним из объединяющих понятий информатики – абстрактное представление, которое описывает организацию транспортных систем, взаимодействие между людьми и телекоммуникационные сети. То, что с помощью одного формального представления можно смоделировать так много различных структур, является источником огромной силы для образованного программиста».

Стивен С. Скиена[1],
заслуженный профессор кафедры информатики,
университет Стони Брук

В настоящее время наиболее насущные проблемы анализа данных связаны с отношениями, а не просто с размером таблицы дискретных данных. Графовые технологии и анализ графов предоставляют мощные инструменты для работы со связанными данными, которые используются в исследованиях, социальных инициативах и бизнес-решениях, например:

- моделирование динамических сред от финансовых рынков до IT-сервисов;
- прогнозирование распространения эпидемий болезней, а также периодических задержек и сбоев в компьютерных сетях;
- поиск прогностических признаков для машинного обучения систем борьбы с финансовыми преступлениями;
- выявление шаблонов поведения для персонализированного опыта и рекомендаций.

Поскольку данные становятся все более взаимосвязанными, а системы – все более сложными, крайне важно использовать обширные отношения, скрытые в наших данных.

Эта глава содержит введение в анализ графов и графовые алгоритмы. Но прежде чем обсуждать графовые алгоритмы и объяснять разницу между графовыми базами данных и обработкой графов, мы начнем с крат-

[1] The Algorithm Design Manual, by *Steven S. Skiena*, Springer.

кого рассказа о происхождении графов. Затем мы рассмотрим природу современных данных и убедимся, что информация, содержащаяся в связях, гораздо сложнее, чем то, что мы можем выявить с помощью обычных статистических методов. Глава завершится обсуждением вариантов использования графовых алгоритмов.

Что такое графы?

Графы ведут свою историю с 1736 года, когда Леонард Эйлер решил знаменитую задачу «Семи мостов Кенигсберга». Вопрос заключался в том, можно ли посетить все четыре района города, соединенные семью мостами, при этом пересекая каждый мост только один раз.

Размышляя над задачей, Эйлер понял, что для решения имеют значение только связи, и тем самым заложил основы теории графов и ее математики. На рис. 1.1 изображен ход рассуждений Эйлера и один из его оригинальных набросков из статьи «*Solutio problematis ad geometriam situs pertinentis*».

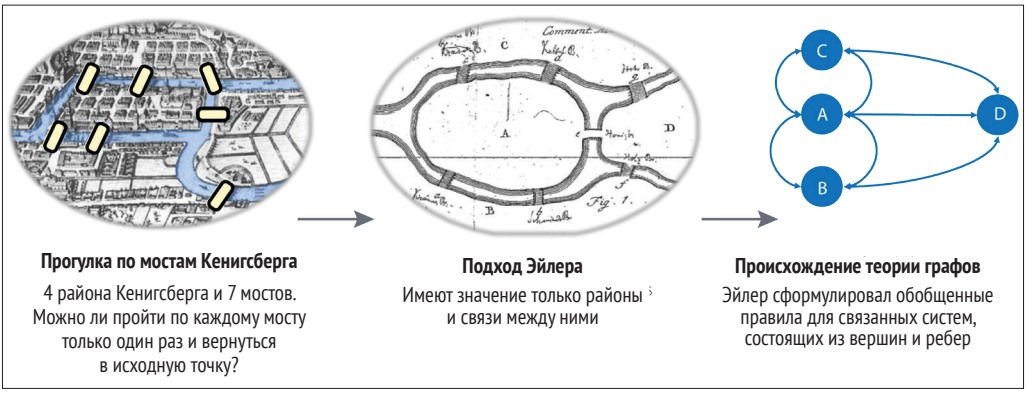

Рис. 1.1. Истоки теории графов. Город Кенигсберг включал два больших острова, соединенных друг с другом и материковой частью города семью мостами. Задача состояла в том, чтобы построить замкнутый маршрут, проходящий по каждому мосту только один раз

Хотя понятие графов зародилось в математике, они также являются удобным и точным способом моделирования и анализа данных. Объекты, составляющие граф, называются *узлами* или *вершинами*, а связи между ними называются *отношениями*, *связями* или *ребрами*. В этой книге мы используем термины *вершины* и *ребра*. Вы можете думать о вершинах как о существительных в предложении, а о ребрах как о глаголах, создающих смысловой контекст, т. е. связи. В английском языке графы, графики и различные диаграммы обозначаются одним словом *graph*, поэтому мы сразу хотим подчеркнуть, что графы, о которых пойдет речь в этой книге, не имеют ничего общего с построением графиков уравнений или диаграмм, изображенных в правой части рис. 1.2.

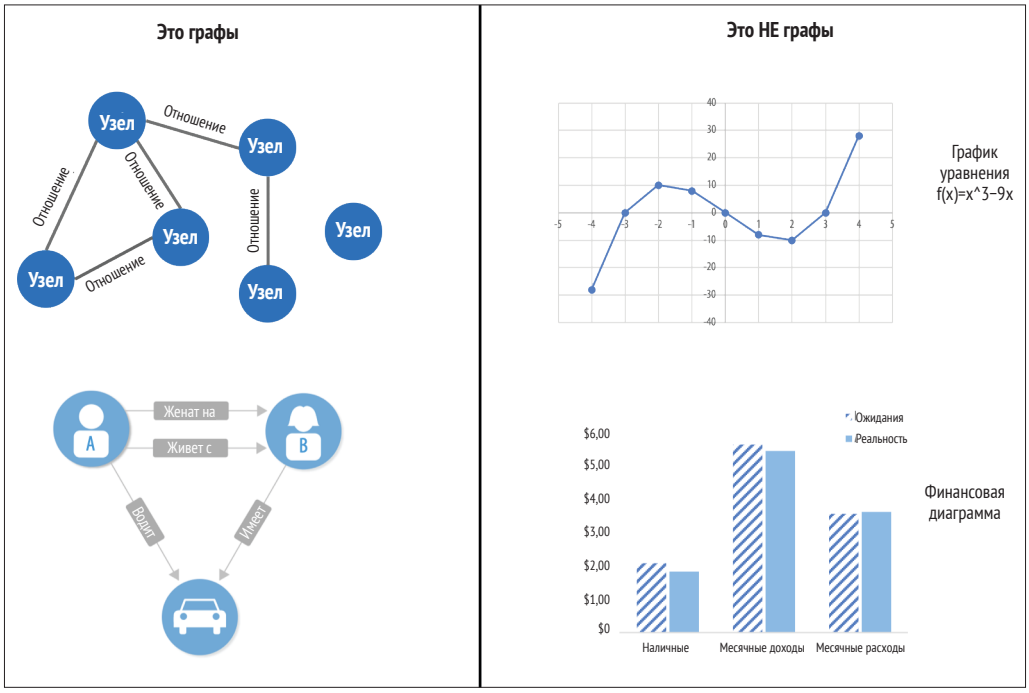

Рис. 1.2. Граф представляет собой схему взаимосвязей, часто изображаемую в виде кружков, называемых вершинами, и соединительных линий, называемых ребрами

Глядя на граф человека в левой части рис. 1.2, мы можем легко составить несколько предложений, которые описывают его как личность. Например, человек *A* живет с человеком *B*, которому принадлежит автомобиль, и человек *A* ведет автомобиль, принадлежащий человеку *B*. Такой подход к моделированию убедителен, поскольку он отлично сопоставляется с реальным миром и, как говорится, «whiteboard friendly», – схему отношений можно запросто набросать маркером на лекционной доске. Это помогает согласовать моделирование и анализ данных.

Но построение графов – это только половина дела. Мы ведь захотим обработать их, чтобы выявить скрытый смысл. Тут-то и вступают в дело графовые алгоритмы.

Что такое графовые алгоритмы и анализ графов?

Графовые алгоритмы являются подмножеством инструментария для анализа графов. В свою очередь, *анализ графов* – то, чем мы с вами занимаемся – это использование любого графового подхода для анализа связанных данных. Мы можем использовать разные методы – извлекать данные из графа, использовать обычную статистику, исследовать графы визуально

или включать их в задачи машинного обучения. Графовые запросы на основе шаблонов часто используются для локального анализа данных, тогда как вычислительные графовые алгоритмы обычно используют более глобальный и итеративный анализ. Хотя использование этих типов анализа частично совпадает, мы используем термин *графовые алгоритмы* для обозначения последнего, более вычислительного анализа и использования графов в науке о данных.

Наука о сетях

Наука о сетях (network science) – это академическая область науки, которая прочно опирается на теорию графов и связана с математическими моделями отношений между объектами. Ученые в этой области вынуждены полагаться на графовые алгоритмы и системы управления базами данных из-за размера, связности и сложности данных, с которыми им приходится работать.

Есть много великолепных ресурсов, посвященных науке о сетях и связанных данных. Вот несколько ссылок для самостоятельного изучения:

- http://networksciencebook.com/ – хорошая вводная онлайн-книга;
- https://www.complexityexplorer.org – онлайн-курс Института Санта-Фе;
- https://necsi.edu – различные ресурсы и документы на сайте Института сложных систем Новой Англии.

Графовые алгоритмы предоставляют один из наиболее эффективных подходов к анализу связанных данных, потому что их математические вычисления специально созданы для работы с отношениями. Они описывают действия, которые необходимо предпринять для обработки графа, чтобы выявить его общие качества или конкретные количества. Основываясь на математике теории графов, алгоритмы используют отношения между вершинами, чтобы проанализировать организацию и динамику сложных систем. В науке о сетях эти алгоритмы применяют для выявления скрытой информации, проверки гипотез и прогнозирования поведения.

Графовые алгоритмы обладают широким потенциалом – от предотвращения мошенничества и поиска оптимальных маршрутов телефонных звонков до прогнозирования распространения гриппа. Например, мы можем оценить, какие подстанции подвержены наибольшему риску при перегрузке энергосистемы. Или мы можем обнаружить группировки в графе, способствующие перегрузке транспортной системы.

И в самом деле, в 2010 году в системах воздушных перевозок США произошли два серьезных инцидента, связанных с перегрузкой нескольких аэропортов, которые впоследствии были изучены с использованием ана-

лиза графов. Сетевые ученые П. Флёркин, Дж. Рамаско и В. М. Игалез использовали графовые алгоритмы для выявления событий, вызывающих систематические каскадные задержки, и использовали эту информацию для составления рекомендаций, как описано в их статье[2] «Распространение системной задержки в аэропортах США».

Мартин Гранджин создал визуализацию всемирной сети воздушного транспорта (рис. 1.3) в своей статье[3] «Связанный мир: распутывая сеть воздушного движения». Эта иллюстрация ясно показывает сильно связанную структуру авиатранспортных кластеров. Многие транспортные системы демонстрируют концентрированное распределение связей с четкими узорами типа «ступица и спица», которые влияют на задержки.

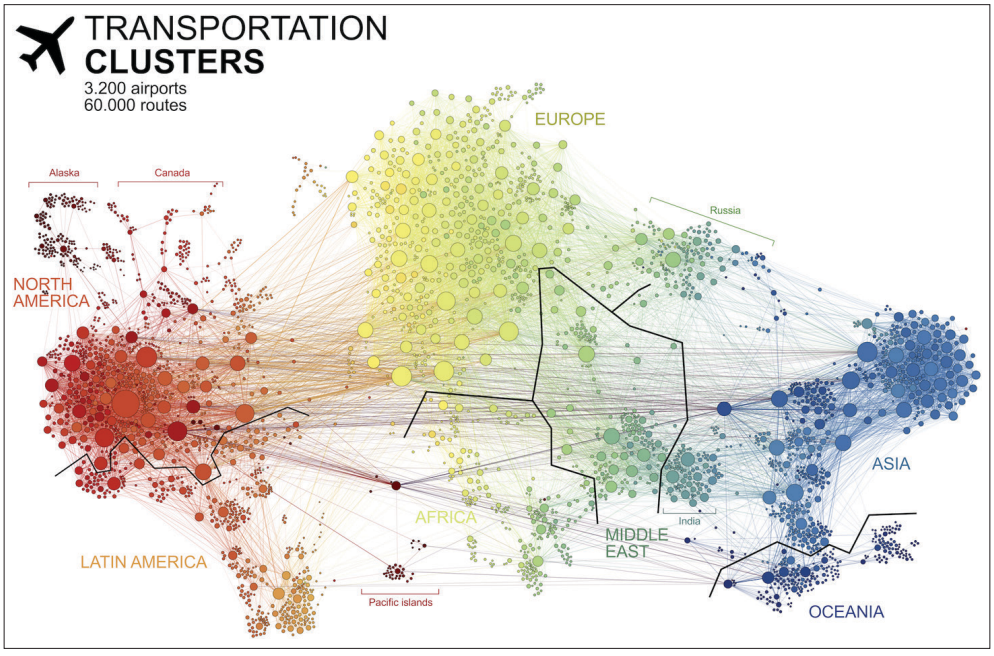

Рис. 1.3. Авиатранспортные сети содержат структуры «ступица и спицы» разной величины. От этих структур зависит перемещение потоков пассажиров и грузов

Графы также помогают раскрыть, как очень маленькие взаимодействия в динамике приводят к глобальным мутациям. Графы связывают воедино микро- и макроуровни, точно отражая, какие сущности взаимодействуют внутри глобальных структур. Эти ассоциации применяются для прогнозирования поведения и определения недостающих связей. Рисунок 1.4 – это сеть взаимодействий видов лугопастбищных угодий, которая использует анализ графов для оценки иерархической организации и взаимодействий видов, а затем предсказывает недостающие взаимосвязи, как подробно

[2] *P. Fleurquin, J. J. Ramasco,* and *V. M. Eguíluz,* «Systemic Delay Propagation in the US Airport Network».

[3] *Martin Grandjean,* «Connected World: Untangling the Air Traffic Network».

описано в статье[4] А. Клаусета, К. Мура и М. Э. Ньюмана «Иерархическая структура и прогнозирование отсутствующих ссылок в сети».

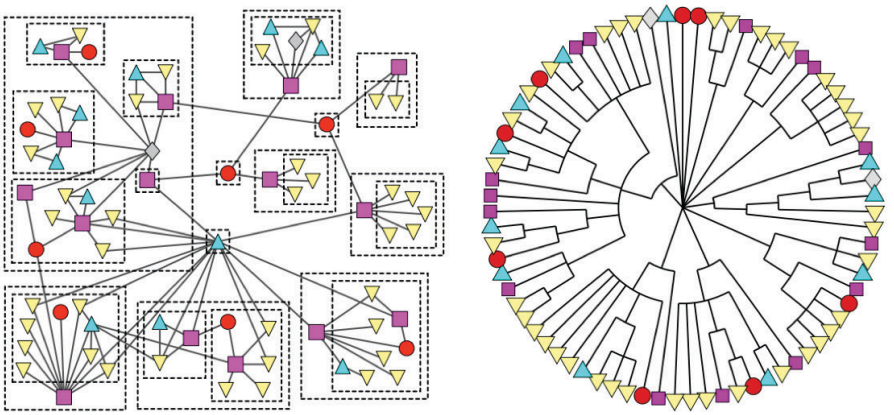

Рис. 1.4. Эта пищевая сеть видов пастбищ использует графы для выявления корреляции мелких взаимодействий с образованием более крупных структур

Обработка графов, базы данных, запросы и алгоритмы

Обработка графов включает методы, с помощью которых выполняются рабочие операции и задачи, связанные с графами. Большинство запросов к графу рассматривает конкретные части графа (например, начальную вершину), и работа обычно сосредоточена на окружающем подграфе. Мы называем этот тип взаимодействия *локальным*, и он подразумевает декларативные обращения к структуре графа, о чем рассказано в книге[5] Яна Робинсона, Джима Вебера и Эмиля Эифрема. Данная разновидность локальной обработки графа часто используется для транзакций в реальном времени и запросов на основе шаблонов.

Говоря о графовых алгоритмах, мы обычно ищем глобальные шаблоны и структуры. Входными данными для алгоритма обычно является *весь* граф, а выходными данными может быть обогащенный граф или какое-либо совокупное значение, такое как оценка. Мы называем такое взаимодействие *глобальным*, и это подразумевает обработку полной структуры графа с использованием вычислительных алгоритмов – зачастую итеративно. Подобный подход раскрывает общий характер сети через ее связи. Организации, как правило, используют графовые алгоритмы для моделирования

[4] *A. Clauset*, *C. Moore*, and *M. E. J. Newman*, «Hierarchical Structure and the Prediction of Missing Links in Network».

[5] *Ян Робинсон*, *Джим Вебер*, *Эмиль Эифрем* «Графовые базы данных. Новые возможности для работы со связанными данными», ДМК Пресс, 2016.

систем и прогнозирования поведения на основе анализа распространения взаимодействий, выявления сообществ, оценки важности компонентов и общей надежности системы.

Эти определения могут частично пересекаться – иногда мы можем использовать алгоритм для ответа на локальный запрос или наоборот, – но, упрощенно говоря, действия над целым графом выполняются вычислительными алгоритмами, а операции с подграфами реализуются через запросы к базам данных.

Традиционно выполнение и анализ транзакций были разделены – противоестественный раскол, основанный на технологических ограничениях. По нашему мнению, анализ графов способствует выполнению более разумных транзакций, что создает новые данные и возможности для последующего анализа. В последнее время появились подходы, преодолевающие упомянутое разделение и способствующие более оперативным решениям.

OLTP и OLAP

Оперативная обработка транзакций (online transaction processing, OLTP) – это, как правило, короткие действия, такие как бронирование билета, пополнение счета, резервирование товара и т. д. OLTP подразумевает массовую обработку запросов с низкой задержкой и высокую целостность данных. Хотя OLTP может включать только небольшое количество записей на транзакцию, системы обрабатывают много транзакций одновременно.

Оперативный анализ данных (online analytical processing, OLAP) облегчает более сложные запросы и анализ исторических данных. Подобный анализ может задействовать несколько источников данных, форматов и типов. Типичными случаями использования OLAP являются обнаружение тенденций, выполнение сценариев типа «*что, если*», прогнозирование и выявление структурных шаблонов. По сравнению с OLTP системы OLAP обрабатывают меньшее количество более длительных транзакций по многим записям. Системы OLAP склонны к быстрому чтению без ожидания подтверждения транзакций, присущего OLTP, поэтому обычной практикой является пакетная обработка.

Однако в последнее время грань между OLTP и OLAP начала стираться. Современные приложения с интенсивным использованием данных теперь сочетают транзакции в реальном времени с аналитикой. Это слияние вызвано современными достижениями в программном обеспечении, такими как более масштабируемое управление транзакциями и добавочная обработка потоков, а также удешевлением быстродействующего оборудования с большой памятью.

Объединение аналитики и транзакций позволяет проводить непрерывный анализ данных как естественную часть регулярных операций. Совре-

менная аналитика обеспечивает возможность выработки рекомендаций и решений по мере сбора данных – с компьютеров в кассовых терминалах (point-of-sale, POS), производственных систем или устройств интернета вещей (internet of things, IoT) – т. е. в режиме реального времени. Эта тенденция проявилась несколько лет назад и описывается неуклюжим длинным термином *транслитеральная и гибридная обработка и анализ транзакций* (translytics and hybrid transactional and analytical processing, HTAP). На рис. 1.5 показано, как для объединения различных типов обработки применяется защищенная репликация баз данных.

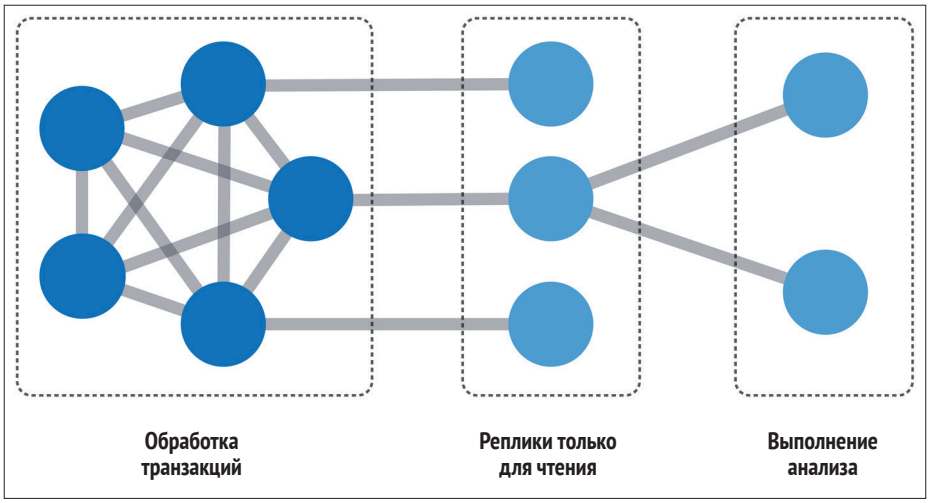

Рис. 1.5. Гибридная платформа поддерживает обработку запросов с низкой задержкой и высокую целостность данных, необходимые для транзакций, и в то же время выполняет сложный анализ больших объемов данных

Как сказано в докладе исследовательской и консалтинговой компании Gartner[6]:

> *«[HTAP] потенциально может изменить способ выполнения некоторых бизнес-процессов, поскольку расширенная аналитика в реальном времени (например, планирование, прогнозирование и анализ «что, если») становится неотъемлемой частью самого процесса, а не отдельной операцией, выполняемой постфактум. Это позволит создать новые способы принятия решений в режиме реального времени. В конечном счете HTAP станет ключевой архитектурой для интеллектуальных бизнес-процессов».*

Поскольку OLTP и OLAP становятся более интегрированными и начинают поддерживать общие функции, больше нет нужды использовать разные форматы данных или разные системы для наших рабочих задач – мы

[6] https://www.gartner.com/imagesrv/media-products/pdf/Kx/KX-1-3CZ44RH.pdf

можем упростить нашу архитектуру, используя одну и ту же платформу для того и другого. Это означает, что наши аналитические запросы могут получать данные в реальном времени, а мы сможем упростить итеративный процесс анализа.

Почему мы должны изучать графовые алгоритмы?

Графовые алгоритмы применяются для углубленного понимания связанных данных. Мы видим отношения в различных реальных системах – от взаимодействия белков до социальных сетей, от систем связи до электросетей, от розничных продаж до планирования миссий на Марсе. Понимание сетей и связей внутри них несет в себе невероятный потенциал для открытий и инноваций.

Графовые алгоритмы уникально подходят для изучения структур и выявления шаблонов в наборах данных, которые тесно связаны между собой. Нет ничего более очевидного, чем большие данные, снабженные наглядными связями. Каждую секунду во всем мире собирается, объединяется и динамически обновляется ошеломляющий объем информации. Именно графовые алгоритмы помогают разобраться в гигантских объемах связанных данных с помощью более сложного анализа, использующего отношения и – что особенно важно сегодня – улучшают контекстную информацию для искусственного интеллекта.

Поскольку наши данные становятся все более связанными, нам важно вовремя понимать их взаимосвязи и взаимозависимости. Ученые, изучающие рост сетей, отмечают, что со временем вероятность подключения увеличивается, но не равномерно. Одной из теорий о том, как динамика роста влияет на структуру сети, является механизм *предпочтительного присоединения* (preferential attachment). Эта идея, показанная на рис. 1.6, описывает склонность узла сети связываться с теми узлами, которые уже имеют много соединений[7].

В своей книге[8] «Синхронизация: как порядок возникает из хаоса во Вселенной, природе и повседневной жизни» Стивен Строгац приводит примеры и объясняет различные способы самоорганизации реальных систем. Независимо от объяснения причин многие исследователи считают, что развитие сетей неотделимо от их конечных форм и иерархий. Сети передачи данных склонны к образованию очень плотных групп, причем сложность сети растет вместе с размером данных. Сегодня мы наблюдаем такую кластеризацию связей в большинстве реальных сетей, от интернета до социальных сетей, таких как игровое сообщество (рис. 1.7).

---

[7] Наглядный пример – нарастающее число новых подписчиков у знаменитых блогеров. – *Прим. перев.*

[8] «Sync: How Order Emerges from Chaos in the Universe, Nature, and Daily Life», *Steven Strogatz*, *Hachette*.

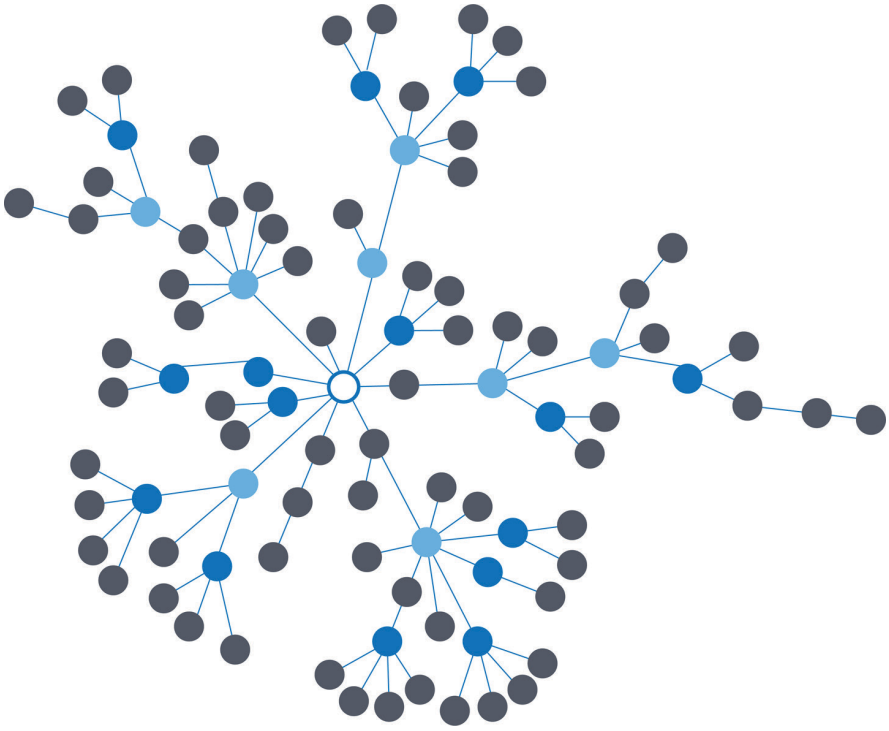

Рис. 1.6. Предпочтительное присоединение – это явление, когда чем больше связан узел сети, тем больше вероятность того, что он получит новые связи. Это приводит к возникновению неравномерности концентрации и выраженных кластеров

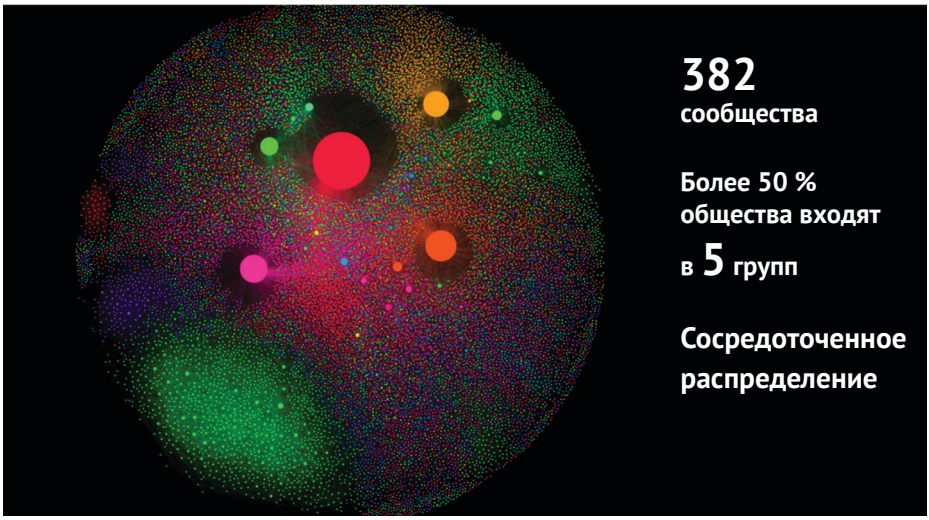

Рис. 1.7. Этот анализ игрового сообщества показывает наибольшую концентрацию связей лишь у пяти из 382 сообществ

Анализ сетевого сообщества, проиллюстрированный на рис. 1.7, был выполнен Франческо Д'Орацио из компании Pulsar с целью изучить виральность контента и стратегии распространения информации. Д'Орацио обнаружил корреляцию между распределением концентрации связей и скоростью распространения фрагмента контента в сообществе.

Эта картина разительно отличается от того, что предсказывает модель среднего распределения, где большинство узлов имеет приблизительно одинаковое количество связей. Например, если бы Всемирная паутина подчинялась среднему распределению связей, у большинства страниц было бы примерно одинаковое количество входящих и исходящих ссылок. Модели среднего распределения утверждают, что большинство узлов одинаково связаны, но многие типы графов и реальные сети демонстрируют кластеризацию. Интернет, так же как и транспортные маршруты или социальные отношения, описывается степенным распределением, т. е. несколько узлов сети буквально облеплены связями, а остальные узлы довольствуются небольшим количеством связей.

Степенной закон

Степенной закон (power law, также называемый *законом подобия*) описывает отношения между двумя величинами, когда одна величина изменяется как степень другой. Например, площадь куба связана с длиной его сторон степенью 3. Хорошо известным примером является распределение Парето, или «правило 80/20», первоначально использовавшееся для описания ситуации, когда 20 % населения контролирует 80 % богатства. Мы встречаем различные степенные законы в мире природы и сетях.

Попытка «усреднить» сеть, как правило, неприемлема для исследования взаимосвязей или прогнозирования, поскольку реальным сетям присуще неравномерное распределение узлов и взаимосвязей. Более того, эта неравномерность сама по себе несет важную информацию. На рис. 1.8 мы можем наблюдать, как использование среднего значения характеристик для данных, которые являются неравномерными, приведет к неверным результатам.

Поскольку сильно связанные данные не соответствуют среднему распределению, сетевые ученые используют анализ графов для поиска и интерпретации структур и распределений отношений в реальных данных.

> *В природе не существует известных нам сетей, которые можно было бы описать случайной моделью.*
>
> Альберт-Ласло Барабаши,
> директор Центра по изучению сложных сетей при Северо-Восточном университете, автор многочисленных книг в области науки о сетях.

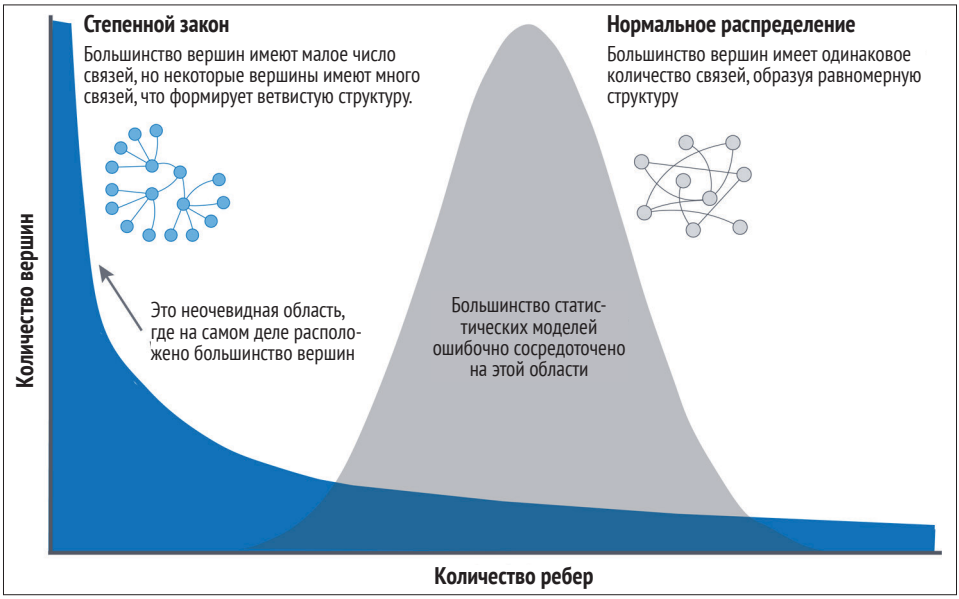

Рис. 1.8. Реальные сети имеют неравномерное распределение узлов и связей, представленных в крайнем случае степенным распределением. Среднее распределение предполагает, что большинство узлов имеет приблизительно одинаковое количество связей

Проблема для большинства исследователей заключается в том, что данные с плотными и неравномерными связями трудно анализировать с помощью традиционных инструментов. В данных наверняка есть какая-то структура, но ее трудно найти. Заманчиво использовать усредненный подход к запутанным данным, но это спрячет закономерности, а результаты вашего анализа будут отражать что угодно, только не реальные группы. Например, если вы усредните демографическую информацию обо всех ваших клиентах и предложите опыт, основанный исключительно на средних показателях, вы гарантированно пропустите большинство сообществ! Ведь сообщества, как правило, объединяются через связывающие факторы, такие как возраст и род занятий, семейное положение и место проживания.

Кроме того, с помощью моментального снимка данных практически невозможно отследить динамическое поведение, особенно в отношении внезапных событий и всплесков. Например, если вы рассматриваете социальную группу с растущими отношениями, вполне естественно, что ее члены будут интенсивно общаться между собой. Это может привести к тому, что наступит переломный этап общения, который приведет к образованию прочной коалиции или, наоборот, к формированию и поляризации подгрупп, например на выборах. Для прогнозирования развития сетей применяются весьма сложные методы, но мы можем сделать вывод

о возможном поведении, если будем видеть структуры и взаимодействия в наших данных. В данном случае анализ графов может спрогнозировать устойчивость социальной группы, поскольку сосредоточен именно на отношениях.

Где и когда применяется анализ графов?

На самом абстрактном уровне анализ графов применяется для прогнозирования поведения и предписания действий для динамических групп. Для этого требуется понимание отношений и структуры внутри группы. Графовые алгоритмы достигают этого понимания путем изучения общей природы сетей через их соединения. Благодаря такому подходу вы в результате сможете понять топологию связанных систем и смоделировать их процессы.

Существует три основных блока вопросов, от которых зависит, уместно ли использование анализа графов и графовых алгоритмов в вашей задаче (рис. 1.9).

Рис. 1.9. Типы вопросов, на которые отвечает анализ графов

Итак, вот несколько типов задач, в которых используются графовые алгоритмы. Похожи ли на них ваши рабочие задачи?

- изучение путей распространения заболевания или каскадного транспортного сбоя;
- выявление наиболее уязвимых или вредоносных компонентов при сетевой атаке;
- определение наименее затратного или самого быстрого способа маршрутизации информации или ресурсов;
- предсказание недостающие связей в ваших данных;
- выявление прямого и косвенного влияния в сложной системе;
- обнаружение невидимых иерархий и зависимостей;
- прогнозирование, будут ли группы объединяться или распадаться;

- поиск перегруженных или недогруженных узлов в сетях;
- выявление сообщества на основе поведения и создание персональных рекомендаций;
- уменьшение количества ложных срабатываний при обнаружении мошенничества и аномалий;
- извлечение дополнительных признаков для машинного обучения.

Заключение

В этой главе мы говорили о том, что современные данные, как правило, образуют сильно связанные структуры. В научной среде для анализа групповой динамики и взаимоотношений давно применяются надежные инструменты, однако подобные решения не всегда являются обычным делом в бизнесе. Оценивая передовые методы анализа, мы должны учитывать природу наших данных и хорошо понимать атрибуты сообщества при прогнозировании сложного поведения. Если наши данные представляют собой сеть, нам следует избегать соблазна снизить число параметров путем усреднения. Вместо этого мы должны использовать правильные инструменты, которые соответствуют нашим данным и ожиданиям от анализа.

В следующей главе мы рассмотрим связанные с графами понятия и термины.