ГОУ ВПО «Казанский Государственный Университет
им. В.И.Ульянова-Ленина»
Факультет вычислительной математики и кибернетики
Кафедра экономической кибернетики

Технология *Data Mining*: Интеллектуальный Анализ Данных

Степанов Роман Григорьевич

Казань, 2008

Оглавление

Глава 1

Введение

В результате развития информационных технологий, количество данных, накопленных человечеством в электронном виде, растет быстрыми темпами. Эти данные существует вокруг нас в различных видах: тексты, изображения, аудио, видео, гипертекстовые документы, реляционные базы данных и т.д. Огромное количество данных появилось в результате повсеместного использования сети Интернет, которая значительно облегчила доступ к информации из географически удаленных точек Земли. Однако подавляющая часть доступной информации не несет для конкретного человека какой-либо пользы. Человек не в состоянии переработать такое количество сведений. Возникает проблема извлечения полезной для пользователя информации из большого объема «сырых» данных. Данное руководство посвящено *Технологии Интеллектуального Анализа Данных* (Data Mining) – одной из активно развивающихся областей информационных технологий, предназначенной для выявления полезных знаний из баз данных различной природы.

1.1 Мотивы для создания технологии

Технология интеллектуального анализа данных (*Data mining*) может рассматриваться как результат естественной эволюции информационных технологий. По пути такой эволюции индустрия баз данных занималась разработкой следующих функциональностей: *накопление данных*, *управление данными* (включая хранение и извлечение, а также выполнение транзакций), а также *анализ данных* (включая разработку хранилищ данных и технологию интеллектуального анализа даннных).

С 1960-х годов информационные технологии последовательно

эволюционировали от примитивных систем обработки файлов до сложных, мощных систем управления базами данных. Исследования в области баз данных с 1970-х годов смещались от ранних иерархических и сетевых баз данных к реляционным СУБД, инструментам моделирования данных, а также к вопросам индексирования и организации данных. В дополнение люди получили гибкий и удобный доступ к данным с помощью языков запросов (например, SQL), пользовательские интерфейсы, оптимизированную обработку запросов и управление транзакциями. Эффективные методы онлайновой обработки транзакций (*on-line transaction processing* – **OLTP**) внесли большой вклад в эволюцию и широкое внедрение реляционной технологии в качестве главного инструмента эффективного хранения, извлечения и управления большими объемами данных.

Технология баз данных начиная с середины 1980-х характеризовалась популяризацией, широким внедрением, и концентрацией исследовательских усилий на новые, все более мощные СУБД. Появились новые модели данных, такие как объектно-ориентированные, объектно-реляционные, дедуктивные модели. Возникли предметно-ориентированные СУБД, включая пространственные, временны́е, мультимедийные, научные системы баз данных, базы знаний, базы офисной информации. Рассматривались вопросы, связанные с распределением, диверсификацией и разделением данных. Появились гетерогенные системы баз данных, а также глобальные информационные системы, такие как Всемирная Паутина – World Wide Web (WWW), которые играют ключевую роль в индустрии информационных технологий.

Удивительно быстрый прогресс компьютерных аппаратных средств за последние сорок лет привел к массовому предложению мощных и доступных компьютеров и накопителей данных. Это способствовало всплеску индустрии информационных технологий и сделало огромное количество баз данных и репозиториев информации доступными для управления транзакциями, извлечения данных и анализа данных.

Данные теперь могут храниться в разных типах бах данных. Одна из недавно появившихся архитектур – это *хранилища данных*, репозиторий множества разнородных источников данных, организованных в рамках единой схемы в одном месте, предназначенный для принятия управленческих решений. Технология хранилищ данных включает очистку данных, интеграцию данных, а также *онлайновую аналитическую обработку* (*On-Line Analytical Processing* – **OLAP**), то есть технологию анализа с такими функциональностями, как консолидация, агрегация, подведение итогов, просмотр информации

“под разными углами”. Хотя технология OLAP позволяет проводить многомерный анализ для принятия решений, для более глубокого анализа требуются дополнительные методы, такие как методы классификации данных, кластерного анализа, характеризации изменений данных во времени и т.д.

Избыток данных и недостаток в хороших методах их анализа приводил к ситуации *богатства данными, но бедности информацией*. Быстро растущие объемы накопленных данных быстро превысили способности человека в их обработке. В результате большие базы данных стали «могилами» данных – архивами, которые редко посещаются. Как следствие, важные решения принимаются не на основе информационно-насыщенных баз данных, а на основе интуиции человека, принимающего решения, так как он не имеет подходящих инструментов для извлечения полезных знаний из огромных объемов данных. Технология Интеллектуального Анализа Данных позволяет извлечь полезные знания, важные *паттерны*, способствуя совершенствованию бизнес-стратегий, баз знаний, научных и медицинских исследований.

Интеллектуальным анализом данных мы будем называть процесс определения новых, корректных и потенциально полезных знаний на основе *больших* массивов данных. В англоязычной литературе вместо термина «интеллектуальный анализ данных» обычно используется термин *Data Mining* (дословный перевод – «добыча данных»), а также близкий термин *Knowledge Discovery in Databases* (KDD) – «Обнаружение знаний в больших базах данных».

Извлеченное знание в результате интеллектуального анализа данных мы будем называть термином *паттерн*. Паттерном может быть, например, некоторое нетривиальное утверждение о структуре данных, об имеющихся закономерностях, о зависимости между атрибутами и т.д.

Таким образом, задачей интеллектуального анализа данных является эффективное извлечение осмысленных паттернов из имеющегося массива данных большого размера. Для отсева большого количества возможных малополезных паттернов может вводится функция полезности. В реальности оценка полезности знания имеет субъективный характер, то есть зависит от конкретного пользователя. Можно выделить две главные характеристики «интересного» знания:

- *Неожиданность*. Знание «удивительно» для пользователя и потенциально несет новую информацию.
- *Применимость*. Пользователь может использовать новое знание для достижения своих целей.

Интересные знания, закономерности, высокоуровневая информация, полученные в результате анализа данных, могут быть использованы для принятия решений, контроля за процессами, управления информацией и обработки запросов. Поэтому технология интеллектуального анализа данных рассматривается как одна из самых важных и многообещающих тем для исследований и применения в отрасли информационных технологий.

1.2 Этапы в процессе интеллектуального анализа данных

Традиционно выделяются следующие этапы в процессе интеллектуального анализа данных:

1. **Изучение предметной области**, в результате которого формулируются основные цели анализа.
2. **Сбор данных**.
3. **Предварительная обработка данных**:
   (a) **Очистка данных** – исключение противоречий и случайных "шумов"из исходных данных
   (b) **Интеграция данных** – объединение данных из нескольких возможных источников в одном хранилище
   (c) **Преобразование данных**. На данном этапе данные преобразуются к форме, подходящей для анализа. Часто применяется агрегация данных, дискретизация атрибутов, сжатие данных и сокращение размерности.
4. **Анализ данных**. В рамках данного этапа применяются алгоритмы интеллектуального анализа с целью извлечения паттернов.
5. **Интерпретация найденных паттернов**. Данный этап может включать визуализацию извлеченных паттернов, определение действительно полезных паттернов на основе некоторой функции полезности.
6. **Использование новых знаний**.

1.3 Компоненты систем интеллектуального анализа

Обычно в системах интеллектуального анализа данных выделяются следующие главные компоненты:

1. **База данных, хранилище данных или другой репозиторий информации.** Это может быть одна или несколько баз данных, хранилище данных, электронные таблицы, другие виды репозиториев, над которыми могут быть выполнены очистка и интеграция. Виды баз данных:
   - Реляционные базы данных;
   - Хранилища данных;
   - Транзакционные базы данных;
   - Объектно-ориентированные базы данных;
   - Объектно-реляционные базы данных;
   - Пространственные базы данных (Spatial databases);
   - Временны́е базы данных (Temporal databases);
   - Текстовые базы данных;
   - Мультимедийные базы данных;
   - Разнородные базы данных;
   - Всемирная Паутина.
2. **Сервер базы данных или хранилища данных.** Указанный сервер отвечает за извлечение существенных данных на основании пользовательского запроса.
3. **База знаний.** Это знания о предметной области, которые указывают, как проводить поиск и оценивать полезность результирующих паттернов.
4. **Служба добычи знаний**. Она является неотъемлемой частью системы интеллектуального анализа данных и содержит набор функциональных модулей для таких задач, как характеризация, поиск ассоциаций, классификация, кластерный анализ и анализ отклонений.
5. **Модуль оценки паттернов**. Данный компонент вычисляет меры интереса или полезности паттернов.

6. **Графический пользовательский интерфейс.** Этот модуль отвечает за коммуникации между пользователем и системой интеллектуального анализа данных, визуализацию паттернов в различных формах.

1.4 Области применения

Приведем некоторые примеры областей, где большое количество данных хранится в централизованных или распределенных базах данных и требует анализа:

- *Электронные библиотеки*, в которых систематизировано хранятся тексты в различных форматах.
- *Архивы изображений*, состоящие из большого количества изображений в сырой или сжатой форме. К изображениям может прилагаться текст.
- *Базы данных геномных исследований.* Как известно, организм человека состоит из более чем 50000 видов генов и белков в различных сочетаниях. Исследованием и интерпретацией огромных баз данных, возникших в результате расшифровки генома человека, занимается биоинформатика.
- *Медицинские изображения.* Большое количество медицинских сведений имеют вид изображений: ЭКГ, снимки внутренних органов и т.д. Анализ этих изображений имеет большое значение для медицины.
- *Финансовые данные* также являются важной сферой применения методов интеллектуального анализа данных. Эти данные представляют из себя котировки акций, золота, рыночные индексы, процентные ставки, кредитные операции банков, транзакции по кредитным картам, выявленные мошеннические операции, и т.д.
- *Базы данных предприятий* обычно хранят подробные сведения об основных бизнес-операциях организации. Например, сведения о клиентах могут представлять интерес для выработки маркетинговой политики организации, политики удержания клиентов, определения индивидуальных предпочтений клиентов.

- *Телекоммуникационные системы* являются источником таких данных, как история вызовов, сбоев, перегрузок, содержимого траффика, и т.д.

- *Всемирная Паутина* содержит огромный объем разнородной мультимедийной информации различного типа. Ее можно считать самой большой распределенной базой данных, которая когда-либо существовала в мире.

- *Биометрические данные* человека (отпечатки пальцев, снимки лиц, и т.д.) находят все большее применение в системах однозначной идентификации человека. Это порождает необходимость развития методов поиска и анализа в подобных базах данных.

1.5 Виды получаемых паттернов

Мы установили различные типы источников и систем хранения данных, к которым применима технология интеллектуального анализа данных. Теперь определим виды паттернов, которые могут быть получены с помощью данной технологии.

Задачи рассматриваемой технологии могут быть разделены на две категории: *задачи описания* и *задачи предсказания*. В задачах описания требуется описать общие свойства данных. В задачах предсказания требуется проанализировать текущие данные для того, чтобы сделать прогноз.

Ниже описаны виды паттернов, которые могут быть получены, в рамках технологии интеллектуального анализа данных.

Характеризация и дискриминация классов

Данные могут быть ассоциированы с классами. Например, клиенты фирмы могут быть условно разделены на тех, кто покупает часто, и тех, кто покупает редко. полезно бывает описать отдельные классы в общих чертах, кратко, и в то же время точно. Такие описания классов и концепций называются *классовыми описаниями*, и могут быть получены (1) *характеризацией данных*, обобщенным описанием данных в классе, и (2) *дискриминацией данных*, то есть сравнением данного класса с одним или более сопоставляемыми классами, часто называемых контрастирующими классами, а также (3) характеризацией и дискриминацией одновременно.

Пример характеризации классов. Пусть Ваша фирма "Электрон"занимается торговлей компьютерной техники, и Вы хотите характеризовать клиентов, которые покупают у вас на сумму более 20000 рублей в год. Результатом может быть следующая характеристика: указанные клиенты имеют возраст 30-40 лет, работают, имеют высшее образование.

Пример дискриминации классов. Пусть вы хотите сравнить два класса клиентов: тех, кто покупает редко, и тех, кто покупает часто. Вы можете обнаружить, что 80% клиентов из первой группы имеют возраст 20–40 лет и имеют высшее образование, в то время как 60% клиентов из второй группы старше 40 лет и не имеют высшего образования.

Анализ асоциаций

Анализ ассоциаций – это обнаружение *ассоциативных правил*, которые представляют из себя условия на значения атрибутов, которые для заданной выборки объектов часто выполняются вместе. Более формально, ассоциативные правила имеют форму $X \Rightarrow Y$, то есть "$A_1 \wedge \cdots \wedge A_m \Rightarrow B_1 \wedge \cdots \wedge B_n$", где A_i (для $i \in \{1, \ldots, m\}$) и B_j (для $j \in \{1, \ldots, n\}$) являются условиями на значение атрибута. Ассоциативное правило $X \Rightarrow Y$ означает, что "записи базы данных, которые удовлетворяют условиям в X, вероятнее всего также удовлетворяют и условиям в Y ".

Пример. Продолжая пример фирмы "Электрон", ассоциативным правилом, которое может быть получено для внутрифирменной базы данных, является следующее правило:

> Возраст $20 \ldots 29 \wedge$ Доход $12000 \ldots 20000 \Rightarrow$ Покупает MP3-плейер (частота $= 2\%$, достоверность $= 60\%$).

Данное правило означает, что среди всех клиентов фирмы "Электрон" 2% (*частота*) имеют возраст от 20 до 29 лет и купили у фирмы MP3-плейер. При этом человек, обладающий подобным возрастом и доходом, купит MP3-плейер с вероятностью 60% (*достоверность*).

Классификация и прогнозирование

Классификация – это процесс нахождения моделей или функций, которые описывают и различают классы для того, чтобы иметь возможность предсказывать класс произвольного заданного объекта с известными атрибутами, но неизвестной меткой класса. Полученная

модель основывается на анализе **обучающей выборки**, то есть множества объектов. чья метка класса известна.

Пример. Фирме "Электрон" может потребоваться оценить, к какому классу принадлежит новый клиент Айрат Гарипов – к классу тех, кто покупает часто или к классу тех, кто покупает редко. В зависимости от этого к клиенту будет разное отношение. В качестве обучающей выборки выступает множество клиентов, о которых мы уже знаем, к какому классу они принадлежат. Результатом классификации будет являться модель, позволяющая предсказать класс нового клиента. Например, на основе предыдущего опыта взаимоотношений с клиентами, мы могли бы вывести закономерность, что молодые люди покупают часто, а пожилые люди покупают редко.

Полученная модель может быть представлена в различных видах: в виде правил классификации (ЕСЛИ-ТО), деревьев решений, математической формулы, нейронных сетей.

Классификация может быть использована для предсказания метки класса для данного объекта. Однако, во многих приложениях может потребоваться предсказать не метку класса, а некоторое пропущенное или недоступное для наблюдений *значение*. Обычно это касается случаев, когда предсказываемое значение – числового (вещественного) типа. В этом случае говорят не о классификации, а о **прогнозировании**. В задаче прогнозирования часто имеют дело с понятием *трендов* в распределении данных.

Классификации и прогнозированию часто предшествует *анализ релевантности*, который предназначен для идентификации атрибутов, которые не влияют на процесс классификации или прогнозирования. Такие атрибуты могут быть исключены из рассмотрения.

Кластерный анализ

В отличие от классификации и прогнозирования, которые анализируют множество объектов обучающей выборки, имеющей известные метки класса, **кластеризация** или **кластерный анализ** анализирует объекты, у которых метки классов неизвестны. Кластеризация как раз призвана сгенерировать эти метки. Объекты кластеризуются или группируются на основе принципа *максимизации внутриклассовой близости* и *минимизации межклассовой близости*. Таким образом, кластеры объектов формируются так, что объекты одного кластера похожи друг с другом, а объекты разных кластеров непохожи.

Каждый полученный кластер может рассматриваться как класс объектов, который в свою очередь может использоваться в других видах

анализа для получения различных правил и закономерностей.

Существуют также методы **иерархической кластеризации**, которые позволяют организовать множество наблюдений в *иерархию классов*.

Пример. В фирме "Электрон" могут использовать кластерный анализ для выявления однородных групп клиентов. Данные группы могут рассматриваться, как целевые группы при проведении маркетинговых мероприятий.

Эволюционный анализ

Эволюционный анализ данных описывает и моделирует регулярности и тренды для объектов, чье поведение изменяется во времени. Несмотря на то, что здесь могут применяться рассмотренные до этого характеризация и дискриминация, анализ ассоциаций, классификация, кластеризация, у данного вида анализа имеются отличительные черты и свои собственные методы, которые включают анализ временны́х рядов, анализ последовательности и периодичности, поиск близостей.

Пример. Допустим, мы имеем данные о состоянии фондового рынка за последние несколько лет, и хотим инвестировать в акции банковского сектора. Эволюционный анализ имеющихся данных может выявить некоторые закономерности в поведении акций, которые могут помочь предсказать будущие тренды на рынке, что повлияет на наше решение об инвестициях.

1.6 Связь с другими дисциплинами

Технология интеллектуального анализа является междисциплинарной областью исследований. В ней используется множество других технологий: базы данных, теория информации, системы искусственного интеллекта, нейронные сети, теория вероятностей и статистика, хранилища данных, высокопроизводительные вычисления, визуализация данных, распознавание образов и т.д.

Отличитиельной чертой технологии интеллектуального анализа данных является то, что особое внимание здесь уделяется *эффективным* и *масштабируемым* методам для обработки *больших* баз данных. При этом масштабируемым считается алгоритм, время которого растет линейно при увеличении размеров базы данных, при заданных системных ресурсах, таких как память и дисковое пространство.

1.7 Упражнения

1.1 Опишите эволюцию в области баз данных по пути к технологии Data Mining.

1.2 Опишите этапы интеллектуального анализа данных.

1.3 Приведите пример, где успех бизнеса зависит от применения технологии интеллектуального анализа данных. Какая функциональность Data Mining при этом используется? При этом можно ли было обойтись запросом к базе данных или простым статистическим анализом?

1.4 Представьте, что вы являетесь разработчиком программного обеспечения в некотором N-ском Университете, и ваша задача – создать систему для интелектуального анализа базы данных, которая содержит следующую информацию о каждом студенте: имя, адрес, год поступления, пройденные курсы и баллы по ним. Опишите архитектуру, которую бы вы выбрали. Каково предназначение каждого компонента данной архитектуры?

1.5 Чем *хранилище данных* отличается от базы данных?

1.6 Опишите каждую функциональность технологии интеллектуального анализа данных из следующего списка: характеризация, дискриминация, анализ ассоциаций, классификация, прогнозирование, кластеризация, эволюционный анализ. Приведите пример для каждой функциональности, используя некоторую реальную базу данных, о которой вы имеете представление.

1.7 В чем различия и сходства между дискриминацией и классификацией? Между характеризацией и кластеризацией? Между классификацией и прогнозированием?

Глава 2

Элементы теории информации

2.1 Энтропия

Для понимания дальнейших тем нам потребуются некоторые сведения из теории информации.

Теорией информации называется наука, изучающая количественные закономерности, связанные с получением, передачей, обработкой и хранением информации. Эта наука была основана Клодом Шенноном в 1948 году. В настоящее время она стала необходимым математическим аппаратом при изучении всевозможных процессов управления.

Получение, обработка, передача и хранение различного рода информации – непременное условие работы любой управляющей системы. Любая информация для того, чтобы быть переданной, должна быть соответственым образом закодирована, то есть переведена на язык специальных символов или сигналов.

Одной из задач теории информации считается задача *сжатия данных*, то есть отыскание наиболее экономных способов кодирования, позволяющих передать заданную информацию с помощью минимального количества символов. Для этого нужно, прежде всего, научиться измерять количественно объем передаваемой или хранимой информации.

Любое сообщение, с которым имеют дело в теории информации, представляет собой совокупность сведений о некоторой физической системе. Например, это может быть сообщение о состоянии котировок на бирже, о нормальном или повышенном количестве брака в цехе, и т.д. и т.п.

Очевидно, если бы состояние физической системы было известно заранее, не было бы смысла передавать сообщение. Сообщение

приобретает смысл только тогда, когда состояние системы заранее неизвестно, случайно. Сведения, полученные о системе, будут тем ценнее и содержательнее, чем больше была неопределенность системы до получения этих сведений («априори»).

Рассмотрим некоторую систему X, которая может принимать конечное множество состояний: $x_1, \ldots, x_n$ с вероятностями $p_1, \ldots, p_n$, где p_i – вероятность того, что система X примет состояние x_i. При этом

$$\sum_{i=1}^{n} p_i = 1. \tag{2.1}$$

Энтропией системы X называется величина

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i. \tag{2.2}$$

Так как $p_i \leq 1$, то величина $H(X)$ неотрицательна.

Концепция «энтропии» была впервые использована физиками как термодинамический параметр для измерения беспорядка или хаоса в термодинамической или молекулярной системе. В статистическом смысле мы можем рассматривать эту величину как меру неопределенности системы.

Энтропия $H(X)$ обладает рядом свойств, оправдывающих ее выбор в качестве характеристики степени неопределенности:

1. Если одно из состояний достоверно, то есть для какого-либо k выполняется $p_k = 1$, то энтропия равна нулю (неопределенность отсутствует). Доказательство следует из того, что

   $$\log_2 1 = 0, \quad \lim_{p \to 0} p \log_2 p = 0.$$

   Так как энтропия может принимать только неотрицательные значения, то в данном случае она достигает своего минимума.

2. При заданном числе состояний n энтропия достигает максимального значения, когда эти состояния равновероятны (максимальная неопределенность). При увеличении n, максимальная энтропия увеличивается.

2.2 Теорема сложения энтропий

Еще одним замечательным свойством энтропии является свойство *аддитивности*. А именно, *энтропия объединения независимых систем равна сумме энтропий этих систем.*

Данное утверждение требует пояснения.

Под объединением двух систем X и Y с возможными состояниями $x_1, \ldots, x_n$; $y_1, \ldots, y_m$ понимается сложная система (X, Y), состояния которой (x_i, y_j) представляют собой все возможные комбинации состояний x_i, y_j систем X и Y. Очевидно, что число возможных состояний системы (X, Y) равно $n \times m$. Обозначим p_{ij} вероятность того, что система (X, Y) будет в состоянии (x_i, y_j).

Найдем энтропию сложной системы:

$$H(X, Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}.$$

Предположим, что системы X и Y независимы, т.е. принимают свои состояния независимо одна от другой. По теореме умножения вероятностей для независимых событий

$$p_{ij} = p_i p_j,$$

откуда

$$H(X, Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_i p_j \log_2 (p_i p_j) =$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{m} p_i p_j \left(\log_2 p_i + \log_2 p_j\right) = -\sum_{i=1}^{n} p_i \log_2 p_i \left(\sum_{j=1}^{m} p_j\right) -$$

$$- \sum_{j=1}^{m} p_j \log_2 p_j \left(\sum_{i=1}^{n} p_i\right) = H(X) + H(Y).$$

Отсюда и следует свойство аддитивности. Доказанное положение называется *теоремой сложения энтропий*. Теорема сложения энтропий может быть легко обобщена на произвольное число независимых систем:

$$H(X_1, \ldots, X_s) = \sum_{k=1}^{s} H(X_k).$$

Если объединяемые системы зависимы, простое сложение энтропий уже неприменимо. В этом случае энтропия сложной системы меньше, чем сумма энтропий ее составных частей.

2.3 Количество информации

Итак, мы выяснили, что энтропия является естественной мерой неопределенности некоторой физической системы. Очевидно, что в результате получения сведений неопределенность системы может быть уменьшена. Чем больше объем получаемых сведений и чем они более содержательны, тем больше будет информация о системе и тем менее неопределенным будет ее состояние. Естественно поэтому *количество информации* измерять уменьшением энтропии той системы, для уточнения состояния которой предназначены сведения.

Оценим информацию $I(X)$, получаемую в результате того, что состояние системы X полностью определилось, т.е. энтропия стала равной 0. Она равна уменьшению энтропии:

$$I(X) = H(X) - 0 = H(X),$$

то есть количество информации, приобретаемое при полном выяснении состояния некоторой физической системы, равна энтропии этой системы.

Количеством информации, содержащей в отдельном сообщении о том, что система находится в состоянии x_i будем называть величину

$$I(x_i) = \log_2 \frac{1}{p_i} = -\log_2 p_i.$$

Это определение согласуется со следующим интуитивным рассуждением: чем меньше вероятность состояния x_i, тем более содержательно сообщение о том, что система в нем находится. Наибольшую информацию несут сообщения о тех событиях, которые априори были наименее вероятны. Например, сообщение о том, что 31-го декабря в г. Казани выпал снег, несет гораздо меньше информации, чем аналогичное по содержанию сообщение, что 31-го июля в г. Казани выпал снег.

Заметим, что величина $I(x_i)$ может быть интерпретирована как количество бит, достаточное для представления сообщения о том, что система находится в состоянии x_i. Легко видеть, что

$$I(X) = \sum_{i=1}^{n} p_i I(x_i),$$

то есть количество информации, приобретаемое при полном выяснении состояния некоторой физической системы равна среднему количеству информации, содержащемся в отдельном сообщении о том, что система находится в состоянии x_i.

2.4 Упражнения

2.1 Докажите второе свойство энтропии.
Подсказка: Найдите максимум функции $-\sum_{i=1}^{n} p_i \log_2 p_i$ по переменным $p_1, \ldots, p_n$ при ограничении (2.1), пользуясь методом неопределенных множителей Лагранжа.

2.2 Пусть рассматривается база данных о физических лицах, содержащая среди прочих два атрибута: *пол* и *служил в армии*, при этом возможные состояния переменной *пол* – "Мужской" и "Женский", а переменной *служил в армии* – "Да" и "Нет". Известны следующие вероятности комбинаций этих состояний:

$$P(\textit{Мужской,Да}) = 0.2, \qquad P(\textit{Женский,Да}) = 0.01.$$

Кроме того, известно, что $P(\textit{Мужской}) = P(\textit{Женский}) = 0.5$. Найдите:

(a) Вероятности $P(\textit{Мужской,Нет})$, $P(\textit{Женский,Нет})$, $P(\textit{Нет})$, $P(\textit{Да})$.

(b) Энтропию простой системы, соответствующей состояниям переменной *Пол*.

(c) Энтропию простой системы, соответствующей состояниям переменной *Служил в армии*.

(d) Энтропию сложной системы, где возможны все комбинации обеих переменных.

Глава 3

Классификация с обучением

3.1 Что такое классификация с обучением?

Пусть имеется набор объектов, каждый из которых принадлежит одному из m классов. В качестве примера можно привести клиентов банка, которые могут быть отнесены к классу добросовестных или недобросовестных заемщиков, а также множество танков на фотоснимке, которые можно разделить на «своих» и «чужих». Задачей *классификации с обучением* является составление правила, по которому для любого объекта можно с большой степенью достоверности определить класс, которому данный объект принадлежит.

Пусть $x_1, \ldots, x_k$ – атрибуты объекта, m – количество классов. В результате классификации должна быть получена некоторая функция $f(x_1, \ldots, x_k)$, значение которой принадлежит $\{1, \ldots, m\}$, и задает номер (*метку*) класса, которому принадлежит объект с атрибутами $x_1, \ldots, x_k$.

В распоряжении у исследователя обычно имеется некоторый набор объектов, у которых метка класса уже известна. Эти объекты могут быть использованы для обучения модели, то есть подбора параметров модели классификации, и для тестирования построенной модели классификации.

Классификация с обучением подразумевает следующие действия:

1. **Подготовка данных.** Имеющийся набор объектов с известными метками классов разбивается на 2 части: *обучающую выборку* и *тестовую выборку*. Желательно, чтобы это разбиение было произведено случайным образом. Чаще всего обучающая выборка имеет размер больше, чем тестовая.

2. **Обучение модели.** Параметры модели классификации подбираются на основе обучающей выборки таким образом,

Рис. 3.1: Дерево решений

чтобы добиться наилучшего соответствия между предсказанными и фактическими метками классов.

3. **Тестирование модели.** Полученная в результате обучения модель проверяется на достоверность. Для этого вычисляется процент неверных результатов классификации объектов из тестовой выборки.

Классификация с обучением имеет множество приложений, например, в таких областях, как кредитование, медицинская диагностика, предсказание доходов, маркетинг. Мы рассмотрим три известных метода классификации с обучением: деревья решений, нейронные сети и метод Naive Bayes.

3.2 Деревья решений

Дерево решений – это дерево, в котором каждой *внутренней вершине* поставлен в соответствие некоторый атрибут, каждая ветвь, выходящая из данной вершины, соответствует одному из возможных значений атрибута, а каждому листу дерева сопоставлен конкретный класс или набор вероятностей классов. Пример дерева решений, позволяющего предсказать, является ли потенциальный клиент добросовестным заемщиком, представлен на рисунке 3.1.

Для того, чтобы классифицировать новый объект, необходимо двигаться по дереву сверху вниз, начиная с корня. При этом на каждом внутреннем узле дерева выбирается та ветвь, которая соответствует фактическому значению соответствующего атрибута. Добравшись до листа дерева, получаем тот класс, которому принадлежит объект согласно классифицирующему правилу.

Основная проблема состоит в том, чтобы построить достаточно хорошее дерево решений. Один из алгоритмов решения этой задачи, известный как алгоритм ID3, представлен на схеме 3.1.

На шаге 3.1 данного алгоритма используется понятие *информационного выигрыша* атрибута. Пусть обучающая выборка S состоит из s объектов, m – это количество рассматриваемых классов, s_i – это число объектов из S, принадлежащих классу с номером i. Количество информации, необходимое для того, чтобы сообщить класс

Алгоритм 3.1 $GenerateDecisionTree(X, A)$. Генерация дерева решений для заданной обучающей выборки

Ввод: Множество X объектов обучающей выборки; набор A дискретных атрибутов объектов.

Вывод: Построенное дерево решений.

1: Создать вершину N;
2: **if** все объекты из X одного класса C **then**
3: Возвращаем вершину N, как лист, соответствующий классу C;
4: **end if**
5: **if** $A = \emptyset$ **then**
6: Возвращаем вершину N, как лист, соответствующий наиболее распространенному в X классу C;
7: **end if**
8: Выбираем среди атрибутов множества A атрибут a с наивысшим *информационным выигрышем*;
9: Сопоставляем вершине N атрибут a;
10: **for all** известные значения $\overline{a}$ атрибута a **do**
11: Создаем ветвь из вершины N, соответствующую условию $a = \overline{a}$;
12: Пусть $\overline{X}$ – множество объектов из X, для которых атрибут a равен $\overline{a}$.
13: **if** $\overline{X} = \emptyset$ **then**
14: Присоединяем к N лист и сопоставляем ему метку самого распространенного в X класса C;
15: **else**
16: {Рекурсивный вызов}
17: Положим $\overline{N} := GenerateDecisionTree(\overline{X}, A \setminus \{a\})$.
18: Присоединяем к созданной ветви из N дерево $\overline{N}$;
19: **end if**
20: **end for**

произвольного объекта, равно

$$I(s_1, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

где p_i – это вероятность того, что произвольный объект принадлежит классу с номером i, оцениваемая величиной

$$p_i = \frac{s_i}{s_1 + \cdots + s_m}.$$

Пусть некоторый заданный атрибут A может иметь ν различных значений $\{a_1, a_2, \ldots, a_\nu\}$. Атрибут A может быть использован для разбиения множества S на ν подмножеств $\{S_1, \ldots, S_\nu\}$, где S_j содержит такие объекты из S, для которых атрибут A имеет значение a_j. Если на шаге 3.1 алгоритма выбрать атрибут A, то подмножества $S_1, \ldots, S_\nu$ соответствуют ветвям, идущим от вершины, содержащей множество S.

Пусть s_{ij} – это количество объектоа класса i в подмножестве S_j. Средняя информация, основанная на разбиении выборки с использованием атрибута A, равна

$$E(A) = \sum_{j=1}^{\nu} \frac{s_{1j} + \cdots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj}). \tag{3.1}$$

Величина $\frac{s_{1j}+\cdots+s_{mj}}{s}$ выступает в качестве веса j-го подмножества и равна числу объектов в подмножестве S_j, деленное на общее число объектов из S. Чем меньше значение (3.1), тем более однородны (в среднем) множества S_j по классовой принадлежности. Заметим, что для заданного множества S_j

$$I(s_{1j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2(p_{ij}),$$

где p_{ij} – вероятность того, что произвольный объект из S_j принадлежит классу i:

$$p_{ij} = \frac{s_{ij}}{|S_j|}.$$

Информационным выигрышем, соответствующим выбору атрибута A в качестве разбивающего множество S, назовем величину

$$Gain(A) = I(s_1, \ldots, s_m) - E(A).$$

Значение $Gain(A)$ может рассматриваться, как среднее сокращение энтропии после того, как стало известным значение атрибута A.

Алгоритм 3.1 на шаге 3.1 вычисляет информационный выигрыш каждого атрибута. Атрибут с наибольшим информационным выигрышем выбирается в качестве разбивающего атрибута для заданного множества S. Создается новая вершина, которая помечается этим атрибутом. Затем для каждого значения этого атрибута создаются ветви дерева, разбивающие S на соответствующие подмножества $S_1, \ldots, S_\nu$. Для каждой созданной ветви процедура повторяется вновь.

В результате работы данного алгоритма получается некоторое дерево решений, которое можно использовать для классификации. Однако часто полученное дерево бывает довольно громоздким, и его желательно упростить. Для этого требуется процедура упрощения дерева. Помимо того, что данная процедура позволяет получитьл более компактный и простой вид дерева решений, она часто позволяет значительно сократить время вычислений.

Существуют два основных подхода к проблеме упрощения дерева решений. Первый подход – упрощение дерева на этапе его создания. В рамках этого подхода на этапе рассмотрения какой-либо вершины может быть принято решение не создавать ветви, выходящие из нее, и не делить соответствующее множество объектов выборки. Это решение может быть принято, если информационный выигрыш от ветвления из данной вершины меньше установленного порога. В результате данная вершина становится листом, который может быть помечен меткой самого представительного класса в соответствующей выборке.

Второй подход предполагает удаление ветвей из уже «выращенного» дерева. Для каждой нелистовой вершины дерева рассматриваются два варианта: когда дерево остается неизменным, и когда из дерева удаляются все ветви, выходящие их данной вершины, а сама вершина становится листом и помечается меткой самого представительного класса для выборки, соответствующей вершине. В обоих вариантах рассчитывается средний процент ошибок классификации и выбирается вариант с наименьшим процентом ошибок.

3.3 Нейронные сети

В мозге человека имеется порядка 10^{11} *нейронов* – клеток, отвечающих за обработку данных. Нейроны связаны между собой многочисленными соединениями – *аксонами* и *дендритами*. Белое вещество мозга состоит из нейронов, а серое – из аксонов и дендритов. Каждый нейрон получает

Рис. 3.2: Многослойный перцептрон

сигнал через множество своих дендритов, а передает результат его обработки через единственный аксон, разветвляющийся на множество (тысячи) *синапсов*. Таким образом, мозг содержит примерно 10^{15} взаимосвязей.

Такому устройству мозга, как считают нейрофизиологи, человек обязан своим разумом. Искусственные нейронные сети (Artifical Neural Networks) – это искусственная вычислительная система, имитирующая поведение биологических нервных систем.

Задачи, которые могут решаться с помощью искусственных нейронных сетей, включают задачу классификации, кластерный анализ, аппроксимацию функций, задачу прогноза, оптимизации, поиска по содержимому и распознавания образов. Искусственные нейронные сети (ИНС) могут быть представлены, как взвешенные ориентированные графы, в которых вершины соответствуют нейронам, а ориентированные ребра с весами соответствуют связям между выходами нейронов и входами нейронов.

По структуре связей нейронные сети могут разделены на два класса:

1. *Сети прямого распространения*: соответствующий сети граф не имеет петель, то есть обратные связи невозможны. Примерами таких сетей являются *однослойный перцептрон,многослойный перцептрон*, *сети Кохонена*.

2. *Рекуррентные сети (сети обратного распространения)*: возможны циклы, а значит обратные связи. Примером является *сеть Хопфилда*.

Мы рассмотрим применение многослойного перцептрона к задаче классификации. Многослойный перцептрон состоит из нескольких слоев нейронов: *входного слоя*, *выходного слоя* и нескольких *скрытых слоев*. Указанная структура представлена на рисунке 3.2.

Нейронная сеть может рассматриваться, как вычислительная система, которой на вход подается вектор ввода, а результатом вычислений является вектор вывода. При этом каждая компонента вектора ввода подается через соответствующий нейрон входного слоя, а вектор вывода соответствует нейронам выходного слоя.

Все слои нейронной сети пронумерованы последовательно от 0 до m, где номер 0 соответствует входному слою, а номер m – выходному. Обозначим n_k – количество нейронов в слое k.

Нейроны каждого слоя соединены со всеми нейронами смежных слоев. Для каждой пары связанных нейронов определен вес этой связи – величина $w_{ij}^{(k)}$, где i – номер нейрона слоя $k-1$, j – номер нейрона слоя k.

Выходом каждого нейрона является величина $x_i^{(k)}$, где i – номер нейрона слоя k. Она рассчитывается на основе входов нейрона и связей этого нейрона с нейронами предыдущих слоев:

$$x_j^{(k)} = f(S_j^{(k)}),$$

где

$$S_j^{(k)} = \sum_{i=1}^{n_{k-1}} x_i^{(k-1)} w_{ij}^{(k)},$$

$$f(x) = \frac{1}{1+e^{-\alpha x}}.$$

Функция $f(x)$ называется *логистической функцией*, ее применение гарантирует, что величина $x_j^{(k)}$ принадлежит отрезку $[0,1]$. Параметр α выбирается пользователем.

Компоненту вектора ввода с номером i обозначим $x_i^{(0)}$. Считаем, что входные данные преобразованы таким образом, что $x_i^{(0)} \in [0,1]$ для всех i. Выходом нейронной сети в соответствии с используемыми обозначениями является вектор, i-я компонента которого равна $x_i^{(m)}$.

Процесс обучения нейронной сети состоит в том, чтобы подобрать ее веса $w_{ij}^{(k)}$ таким образом, чтобы для обучающей выборки результаты на выходе нейронной сети как можно меньше отличались от требуемых результатов. Мерой ошибки является величина

$$E = \frac{1}{2} \sum_{p=1}^{n_m} (x_p^{(m)} - d_p)^2, \tag{3.2}$$

где d_i – требуемые результаты на выходе. Например, для задачи классификации $d_i = 1$, если рассматриваемый элемент обучающей выборки с атрибутами $x_j^{(0)}$ принадлежит классу i, и $d_i = 0$ в обратном случае.

Алгоритм обучения нейронной сети, который называется *алгоритмом обратного распространения ошибки*, основан на методе градиентного спуска. Это означает, что величины $w_{ij}^{(k)}$ на каждом шаге «немного» сдвигаются в сторону антиградиента функции ошибок E:

$$w_{ij}^{(k)} := w_{ij}^{(k)} + \Delta w_{ij}^{(k)},$$

$$\Delta w_{ij}^{(k)} = -\varepsilon \frac{\partial E}{\partial w_{ij}^{(k)}},$$

где ε – некоторое небольшое положительное число, называемое *скоростью обучения*, обычно лежащее в интервале от 0 до 1. Если ε слишком мало, то процесс обучения занимает слишком много времени, если ε слишком велико, то процесс может быстро «свалиться» к некоторому неадекватному локальному минимуму, или осциллировать между такими локальными минимумами. Часто в качестве ε выбирается величина $1/t$, где t – номер итерации алгоритма.

Рассмотрим вопрос вычисления величины $\frac{\partial E}{\partial w_{ij}^{(k)}}$. Обозначим

$$z_j^{(m)} = f'(S_j^{(m)})(x_j^{(m)} - d_j), \tag{3.3}$$

и определим последовательно для $k = m-1, m-2 \ldots, 1$ величины $z_j^{(k)}$ по формуле

$$z_j^{(k)} = f'(S_j^{(k)}) \sum_{p=1}^{n_{k+1}} z_p^{(k+1)} w_{jp}^{(k+1)}. \tag{3.4}$$

Теорема 1 *Для всех слоев нейронной сети k от 1 до m, всех нейронов i слоя $k-1$, всех нейронов j слоя k выполняется равенство*

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = z_j^{(k)} x_i^{(k-1)}. \tag{3.5}$$

Доказательство: Докажем сначала, что для любого r такого, что $k \le r \le m$, выполняется

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_r} \frac{z_p^{(r)}}{f'(S_p^{(r)})} \frac{\partial x_p^{(r)}}{\partial w_{ij}^{(k)}}. \tag{3.6}$$

Будем доказывать данное утверждение по индукции. Пусть сначала $r = m$. Из (3.2), (3.3) следует, что

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \frac{\partial \frac{1}{2} \sum_{p=1}^{n_m} (x_p^{(m)} - d_p)^2}{\partial w_{ij}^{(k)}} = \\ = \sum_{p=1}^{n_m} (x_p^{(m)} - d_p) \frac{\partial x_p^{(m)}}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_m} \frac{z_p^{(m)}}{f'(S_p^{(m)})} \frac{\partial x_p^{(m)}}{\partial w_{ij}^{(k)}}, \tag{3.7}$$

то есть при $r = m$ соотношение (3.6) выполнено.

Покажем теперь, что если оно выполнено при некотором r, что оно выполняется и для $r' = r - 1$, если $r - 1 \geq k$. Действительно, пусть

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_r} \frac{z_p^{(r)}}{f'(S_p^{(r)})} \frac{\partial x_p^{(r)}}{\partial w_{ij}^{(k)}}.$$

Тогда

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_r} \frac{z_p^{(r)}}{f'(S_p^{(r)})} f'(S_p^{(r)}) \frac{\partial S_p^{(r)}}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_r} z_p^{(r)} \frac{\partial \sum_{q=1}^{n_{r-1}} x_q^{(r-1)} w_{qp}^{(r)}}{\partial w_{ij}^{(k)}} =$$
$$= \sum_{q=1}^{n_{r-1}} \left(\sum_{p=1}^{n_r} z_p^{(r)} w_{qp}^{(r)} \right) \frac{\partial x_q^{(r-1)}}{\partial w_{ij}^{(k)}}.$$

Отсюда и из (3.4), получим

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \sum_{q=1}^{n_{r-1}} \frac{z_q^{(r-1)}}{f'(S_q^{(r-1)})} \frac{\partial x_q^{(r-1)}}{\partial w_{ij}^{(k)}} = \sum_{q=1}^{n_{r'}} \frac{z_q^{(r')}}{f'(S_q^{(r')})} \frac{\partial x_q^{(r')}}{\partial w_{ij}^{(k)}}.$$

Таким образом, соотношение (3.6) доказано для всех $r \geq k$. Следовательно, оно выполняется и для $r = k$. Значит,

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_k} \frac{z_p^{(k)}}{f'(S_p^{(k)})} \frac{\partial x_p^{(k)}}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_k} \frac{z_p^{(k)}}{f'(S_p^{(k)})} f'(S_p^{(k)}) \frac{\partial S_p^{(k)}}{\partial w_{ij}^{(k)}} =$$
$$= \sum_{p=1}^{n_k} z_p^{(k)} \frac{\partial \sum_{q=1}^{n_{k-1}} x_q^{k-1} w_{qp}^{(k)}}{\partial w_{ij}^{(k)}} = \sum_{p=1}^{n_k} z_p^{(k)} \sum_{q=1}^{n_{k-1}} x_q^{k-1} \frac{\partial w_{qp}^{(k)}}{\partial w_{ij}^{(k)}} =$$
$$= \sum_{p=1}^{n_k} z_p^{(k)} \sum_{q=1}^{n_{k-1}} x_q^{(k-1)} \delta_{iq} \delta_{jp} = z_j^{(k)} x_i^{(k-1)},$$

что и требовалось доказать. □

Нетрудно показать, что $f'(x) = \alpha f(x)(1 - f(x))$, поэтому

$$f'(S_j^{(k)}) = \alpha f(S_j^{(k)})(1 - f(S_j^{(k)})) = \alpha x_j^{(k)}(1 - x_j^{(k)}).$$

Следовательно (3.3),(3.4) могут быть записаны в более удобном для вычислений виде:

$$z_j^{(m)} = \alpha x_j^{(m)}(1 - x_j^{(m)})(x_j^{(m)} - d_j), \tag{3.8}$$

Алгоритм 3.2 Алгоритм обратного распространения ошибки. Обучение нейронной сети для заданной обучающей выборки.

Ввод: Множество объектов обучающей выборки; скорость обучения ε; параметр α; конфигурация нейронной сети: количество слоев m, количество нейронов n_k в каждом слое k.

Вывод: Обученная нейронная сеть для классификации объектов.

Случайным образом инициализируются начальные значения весов $w_{ij}^{(k)}$;
while не выполнено условие выхода **do**
  Выбираем случайным образом объект из обучающей выборки. Запомним его атрибуты в массиве входов нейронной сети $x_i^{(0)}$, а в массиве значений d_i закодируем номер класса выбранного объекта.
  {Прямой ход алгоритма:}
  for $k = 1$ to m **do**
    for $j = 1$ to n_k **do**
      $S_j^{(k)} = \sum_{i=1}^{n_{k-1}} x_i^{(k-1)} w_{ij}^{(k)}$;
      $x_j^{(k)} = 1/(1 + e^{-\alpha S_j^{(k)}})$;
    end for
  end for
  {Обратный ход алгоритма:}
  for $j = 1$ to n_m **do**
    $z_j^{(m)} = \alpha x_j^{(m)}(1 - x_j^{(m)})(x_j^{(m)} - d_j)$.
  end for
  for $k = m - 1$ to 1 **do**
    for $j = 1$ to n_k **do**
      $z_j^{(k)} = \alpha x_j^{(k)}(1 - x_j^{(k)}) \sum_{p=1}^{n_{k+1}} z_p^{(k+1)} w_{jp}^{(k+1)}$;
    end for
  end for
  {Изменение весов:}
  for $k = 1$ to m **do**
    for $j = 1$ to n_k **do**
      $\Delta w_{ij}^{(k)} = -\varepsilon z_j^{(k)} x_i^{(k-1)}$;
      $w_{ij}^{(k)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$;
    end for
  end for
end while

$$z_j^{(k)} = \alpha x_j^{(k)}(1 - x_j^{(k)}) \sum_{p=1}^{n_{k+1}} z_p^{(k+1)} w_{jp}^{(k+1)}. \tag{3.9}$$

Алгоритм обучения нейронной сети представлен на схеме 3.2

Условием окончания обучения может быть, например, истечение времени, отведенного на обучение, или то, что процент неверно классифицированных объектов обучающей выборки не превысил заданной величины.

Топология нейронной сети (количество слоев, количество нейронов в каждом слое) обычно выбирается эмпирически, и строгих указаний для такого выбора не имеется.

Обучение нейронной сети занимает обычно продолжительное время, поэтому она может применяться только в тех областях, где это приемлемо. Другим существенным недостатком нейронных сетей является то, что результаты обучения плохо интерпретируемы, так как для человека трудно интерпретировать символическое значение весов.

К преимуществам использования нейронных сетей относится то, что они универсальны для разных видов данных, и дают хорошие результаты даже при наличии «зашумленности» в выборке. Данные факторы говорят в пользу использования нейронных сетей в задачах классификации.

3.4 Байесовская классификация

Метод Байесовской классификации является статистическим методом. Он позволяют предсказать вероятность принадлежности объекта к заданному классу.

Метод Байесовской классификации основан на теореме Байеса, приведенной ниже. Достоинствами метода являются как точность, так и скорость при работе с большими массивами данных.

Пусть X – некоторый объект, класс которого неизвестен. Пусть H – гипотеза. заключающаяся в том, что X принадлежит к классу C. Для проблемы классификации мы хотим определить $P(H|X)$, вероятность выполнения гипотезы H при наблюдаемых данных X.

На языке теории вероятностей $P(H|X)$ – это вероятность *a posteriori* наступления H при условии X. Например, рассмотрим в качестве множества объектов фрукты, описываемых в базе данных цветом и формой. Предположим, что X – красного цвета и круглой формы, а H – гипотеза, что X – это яблоко. Тогда $P(H|X)$ – степень достоверности того, что X – это яблоко при том, что мы видим, что X – красное и круглое.

В то же время $P(H)$ – это вероятность *a priori* наступления H. Для нашего примера $P(H)$ – это вероятность, что произвольно взятый объект из нашей базы данных будет являться яблоком. Вероятность a posteriori $P(H|X)$ базируется на большем количестве информации, чем вероятность a priori $P(H)$, которая не зависит от X.

Аналогично, $P(X|H)$ – это вероятность a posteriori наступления X при условии H. То есть это вероятность, что X – круглой формы и красного цвета при том, что мы знаем, что X – яблоко.

Теорема Байеса гласит, что

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}. \tag{3.10}$$

Рассмотрим так называемый *наивный метод Байесовской классификации*, как очень простой и эффективный при больших размерах базы данных. В нем предполагается, что все атрибуты независимы друг от друга.

Пусть любой объект задан с помощью n атрибутов, то есть объект X может быть представлен в виде вектора $X = (x_1, \ldots, x_n)$. Предполагаем для простоты, что все атрибуты категориальные, то есть могут принимать лишь конечное число значений. Пусть m – это количество классов.

Мы должны для произвольного заданного объекта X с неизвестной меткой класса, определить вероятности его вхождения в классы $1, \ldots, m$. Класс, которому соответствует наибольшая вероятность, и будет оценкой по методу Байесовской классификации.

Ясно, что искомая вероятность вхождения X в класс с номером i равна $P(H_i|X)$, где H_i – это гипотеза, что объект X относится к классу i. По теореме Байеса (3.10)

$$P(H_i|X) = \frac{P(X|H_i)P(H_i)}{P(X)}.$$

Вычисление $P(X|H_i)$ в общем случае – очень сложная задача. Но если считать, что все атрибуты независимы, то данная задача упрощается, так как в этом случае

$$P(X) = \prod_{k=1}^{n} P(x_k),$$

$$P(X|H_i) = \prod_{k=1}^{n} P(x_k|H_i),$$

где $P(x_k)$ – вероятность a priori того, что значение атрибута с номером k равно x_k, а $P(x_k|H_i)$ – вероятность a posteriori того, что для объекта, принадлежащего классу i, значение атрибута с номером k равно x_k.

Величины $P(x_k), P(x_k|H_i)$ могут быть вычислены на основе обучающей выборки следующим образом:

$$P(x_k|H_i) = \frac{s_{ik}(x_k)}{s_i},$$

$$P(x_k) = \frac{\sum_{i=1}^{m} s_{ik}(x_k)}{\sum_{i=1}^{m} s_i},$$

где $s_{ik}(x_k)$ – количество записей в обучающей выборке, принадлежащих классу i, таких. что значение атрибута с номером k равно x_k; s_i – количество всех записей, принадлежащих классу i.

Теоретически, метод Байесовской классификации имеет минимальную степень ошибок по сравнению с другими классификаторами. Однако на практике это не всегда верно, так как условие независимости атрибутов – слишком сильное условие. Кроме того, часто необходимых статистических данных не хватает для выполнения классификации. Тем не менее, различные эмпирические исследования и сравнения данного метода с деревьями решений и с нейронными сетями показывают, что в ряде областей наивный метод Байесовской классификации вполне применим.

3.5 Упражнения

3.1 Опишите вкратце основные шаги в построении деревьев решений.

3.2 В чем польза от упрощения дерева решений?

3.3 Почему наивный метод Байесовской классификации называется *наивным*? Назовите основные идеи данного метода.

3.4 Пусть имеется база данных о сотрудниках. В результате выполнения запроса к базе данных, получена следующая таблица, в которой для каждой строки поле *Количество* содержит количество записей в исходной базе данных, имеющих соответствующие значения для столбцов *Подразделение*, *Статус*, *Возраст* и *Оклад*.

Подразделение	*Статус*	*Возраст*	*Оклад*	*Количество*
Отдел продаж	старший	31...35	21...24 тыс.	30
Отдел продаж	младший	26...30	5...8 тыс.	40
Отдел продаж	младший	31...35	9...12 тыс.	40
Отдел производства	младший	21...25	21...24 тыс.	20
Отдел производства	старший	31...35	37...40 тыс.	5
Отдел производства	младший	26...30	21...24 тыс.	3
Отдел производства	старший	41...45	37...40 тыс.	3
Отдел маркетинга	старший	36...40	21...24 тыс.	10
Отдел маркетинга	младший	31...35	17...20 тыс.	4
Канцелярия	старший	46...50	13...16 тыс.	4
Канцелярия	младший	26...30	5...8 тыс.	6

Пусть метка класса содержится в столбце *Статус*.

(a) Как модифицировать алгоритм ID3 для учета *Количества* каждого полученного кортежа (т.е. каждой строки вышеописанной таблицы)?

(b) Используйте ваш модифицированный алгоритм ID3 для построения дерева решений, используя данную таблицу.

(c) Классифицируйте объект со значениями «Отдел производства», «21...24 тыс.», «26...30» для атрибутов *Подразделение*, *Оклад* и *Возраст*, используя наивный метод Байесовской классификации.

(d) Постройте многослойный перцептрон для данных из указанной таблицы. Пометьте каждый нейрон из входного и выходного слоя.

(e) Используя построенный многослойный перцептрон, выполните одну итерацию обучения нейронной сети по методу обратного распространения ошибки, если для данной итерации в качестве обучающего элемента выбран элемент “(Отдел продаж, 31…35, 21…24 тыс.)”. Как изменятся исходные веса нейронной сети, если скорость обучения равна ε?

Глава 4

Поиск ассоциативных правил

4.1 Определения

Ассоциативные правила – это связи между логическими атрибутами объектов. Данная глава посвящена методам поиска интересных с точки зрения исследователя ассоциативных правил в больших наборах данных. Рассматривыемые методы применимы к данным произвольной природы.

Типичным примером области, в которой поиск ассоциативных правил имеет важное значение, является *анализ рыночной корзины*. Рассмотрим, например, некоторый супермаркет, в котором продается множество товаров. Покупатели выбирают необходимые им товары, складывают их в корзину и затем оплачивают.

Продавца интересуют ассоциации между различными товарами, которые покупатель складывает в корзину. Например, интерес может представлять вопрос, какова вероятность того, что покупатель, купивший хлеб, приобетет с ним и молоко? Какие товары обычно покупаются вместе?

Такая информация может помочь продавцам выработать маркетинговую или рекламную стратегию. Например, это поможет:

- эффективно расположить товары на территории супермаркета;
- разработать систему скидок на одни товары для того, чтобы стимулировать продажи других товаров;
- выбрать идею рекламы на товар, используя то, что этот товар обычно используется совместно с другим товаром.

Другими областями анализа данных, где используются методы поиска ассоциативных правил, являются: выявление мошеннических

операций по кредитным картам, страховым случаям; определение причин сбоев в телекоммуникационных системах; анализ ДНК живых организмов; обработка данных социологических исследований. и т.д.

Рассмотрим базу данных, состоящую из клиентских *транзакций*, где каждая транзакция характеризуется множеством элементов. Ассоциативное правило формулируется обычно в виде:

$$X \Rightarrow Y, \quad \text{частота} = s\%, \quad \text{достоверность} = c\%,$$

где X и Y – некоторые множества элементов, s, c – числа от 0 до 100. Данное правило означает, что $c\%$ транзакций, содержащих элементы X, содержат и элементы Y; при этом $s\%$ всех транзакций содержат одновременно X и Y. Например, правило

$$\{\text{Сыр}\} \Rightarrow \{\text{Масло, Хлеб}\}, \quad \text{частота} = 3\%, \quad \text{достоверность} = 60\%$$

означает, что 60% транзакций, содержащих сыр, содержат масло и хлеб, и доля транзакций, которые содержат сыр, масло и хлеб, равна 3%.

Более строго, *частота* и *достоверность* правила $X \Rightarrow Y$ определяются следующим образом:

$$\text{Частота}(X \Rightarrow Y) = P(X \cup Y),$$

$$\text{Достоверность}(X \Rightarrow Y) = P(Y|X) = P(X \cup Y)/P(X),$$

где $P(A)$ – доля транзакций, содержащих A.

Кроме частоты и достоверности, еще одной характеристикой ассоциативного правила является его *интерес*:

$$\text{Интерес}(X \Rightarrow Y) = P(Y|X) - P(Y)$$

Ассоциативное правило будет считаться *допустимым*, если оно удовлетворяет ограничениям на *минимальную частоту* и *минимальную достоверность*, которые выбирает пользователь. Наша цель – найти все допустимые ассоциативные правила для исходного множества транзакций.

Определение. Набор элементов X мы будем называть *часто встречающимся*, если его частота $P(X)$ удовлетворяет неравенству $P(X) \geq s$.

Процесс поиска ассоциативных правил состоит из следующих этапов:

1. Определение всех часто встречающихся наборов (ЧВН).
2. Генерация ассоциативных правил, используя найденные ЧВН.

При этом основную сложность представляет первый этап. Мы рассмотрим алгоритм определения часто встречающихся наборов, известный под названием *'A priori'*.

4.2 Алгоритм *A priori*

Пусть задано минимальное значение частоты s.

Рассматриваемый алгоритм использует следующее утверждение, называемое также *свойством A priori*:

Утверждение 1 *Любой набор, содержащийся в некотором часто встречающемся наборе, является часто встречающимся. Другими словами, если* $Y \subseteq X$ *и* $P(X) \geq c$, *то* $P(Y) \geq c$.

Определение. *k-набором* будем называть набор, состоящий из k элементов.

Обозначим через L_k множество всех часто встречающихся k-наборов. Объединение L_k по всем k дает, как нетрудно убедиться, все искомое множество ЧВН. Построение L_k выполняется по шагам. Сначала находится L_1, то есть множество одноэлементных ЧВН. Затем для каждого фиксированного $k \geq 2$, используя найденное множество L_{k-1}, определяется L_k. Процесс завершается, как только k станет больше максимального количества элементов.

Определение L_k при известном L_{k-1} выполняется в два шага: сперва генерируется множество наборов-кандидатов C_k, затем из этого множества исключаются лишние элементы. Полученное таким образом множество и будет равно L_k.

1. Генерация множества кандидатов.

Множество кандидатов C_k составляется путем *слияний* всех *допустимых пар* $l_1, l_2 \in L_{k-1}$. Необходимо дать определение, что такое допустимая пара, и что такое их слияние.

Пусть $l_1, l_2 \in L_{k-1}$ – два набора из множества L_{k-1}. Обозначим через $l_i[j]$ j-й элемень в наборе l_i. Например, $l_1[k-2]$ – это предпоследний элемент в l_1. Предполагается, что на исходном множестве элементов задано некоторое отношение порядка '<'(например, по номерам элементов), и в наборе l_i элементы отсортированы в соответствии с данным отношением порядка.

Пара l_1, l_2 – *допустимая для слияния*, если

$$(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1]).$$

Условие $(l_1[k-1] < l_2[k-1])$ гарантирует, что дубликатов в множестве C_k не будет. *Слиянием* $u(l_1, l_2)$ допустимых наборов l_1, l_2 будет набор, состоящий из элементов $l_1[1], l_1[2], \ldots, l_1[k-1], l_2[k-1]$.

2. Сокращение.

Множество кандидатов C_k содержит все наборы из L_k, но содержит

также и лишние наборы, не являющиеся часто встречающимися. Чтобы получить L_k, осталось лишь исключить такие наборы.

Для этого необходимо для каждого набора из C_k посчитать количество его повторений в базе данных, и исключить этот набор, если число повторений меньше заданного порога.

Но такой подсчет – довольно трудоемкая процедура, так как C_k может иметь очень большой размер. Поэтому рекомендуется сначала произвести его предварительную очистку следующим образом. Пусть l – некоторый набор из C_k (следовательно, он состоит из k элементов). Если l – ЧВН, то в соответствии со свойством A priori, все поднаборы l, состоящие из $k-1$ элементов, должны быть также ЧВН, т.е. принадлежать множеству L_{k-1}. Поэтому, если хотя бы один набор, полученный из l удалением одного элемента, не принадлежит L_{k-1}, то l не может являться ЧВН и должен быть исключен из C_k. Для организации быстрого поиска в L_{k-1} могут быть использованы хеш-деревья всех ЧВН.

4.3 Генерация ассоциативных правил

Как только множество ЧВН для рассматриваемой базы данных определено, генерация ассоциативных правил не составляет труда.

Пусть задано минимальное значение достоверности c.

Напомним, что ассоциативное правило должно удовлетворять ограничению

$$\text{Достоверность}(X \Rightarrow Y) > c,$$

где

$$\text{Достоверность}(X \Rightarrow Y) = count(X \cup Y)/count(X),$$

где $count(X \cup Y)$ – число транзакций, содержащих набор $X \cup Y$, а $count(X)$ – число транзакций, содержащих набор X.

Ассоциативные правила генерируются следующим образом:

- Для каждого непустого ЧВН Z рассматриваем все *непустые* подмножества.
- Для каждого непустого подмножества $X \subset Z$ выводим правило $X \Rightarrow Y$, где $Y = Z \setminus X$, если $\frac{count(Z)}{count(X)} \geq c$.

Так как правила генерируются на основе ЧВН, то все они автоматически удовлетворяют ограничению на частоту. ЧВН вместе с подсчитанным количеством транзакций могут храниться в хеш-таблицах, что обеспечит к ним быстрый доступ.

4.4 Упражнения

4.1 Докажите следующие утверждения:

(a) Любое непустое подмножество часто встречающегося набора элементов, само является часто встречающимся набором.

(b) Частота любого непустого подмножества s' набора элементов s не меньше частоты s.

(c) Пусть l, s – два часто встречающихся набора элементов, таких что $s \subset l$. Тогда достоверность правила $s' \Rightarrow (l - s')$ не больше, чем достоверность правила $s \Rightarrow (l - s)$, где $s' \subset s$.

4.2 Пусть база данных содержит данные о четырех покупках:

ID транзакции	*Дата*	*Приобретенные товары*
100	15.10.2003	{K,A,D,B}
200	15.10.2003	{D,A,C,E,B}
300	19.10.2003	{C,A,B,E}
400	22.10.2003	{B,A,D}

Пусть минимальная частота равна 60%, а минимальная достоверность равна 80%.

(a) Найдите все часто встречающиеся наборы элементов с помощью алгоритма A priori.

(b) Определите все ассоциативные правила.

4.3 Предположим, что в крупной торговой компании имеется база данных о покупках в четырех местах. В каждом из этих четырех мест ведется своя база данных о покупках, при этом формат данных во всех местах одинаков. Предложите эффективный алгоритм для поиска ассоциативных правил. Ваш алгоритм не должен предусматривать передачу всех данных в одно место и не должен приводить к слишком большим объемам сетевых коммуникаций.

4.4 Предположим, что найденные часто встречающиеся наборы элементов для большой базы данных, скажем DB, для заданного минимального уровня частоты сохранены в определенном месте. Предложите эффективные способы поиска ассоциативных правил при том же минимальном уровне частоты, если к DB было добавлено небольшое множество новых транзакций ΔDB.

4.5 Приведите пример, когда элементы в ассоциативном правиле являются *отрицательно коррелированными*.

Глава 5

Кластерный анализ (классификация без обучения)

5.1 Определения

Пусть имеется множество объектов для анализа. Пусть при этом, в отличие от случая классификации с обучением, метка класса для объектов не задана. *Кластерный анализ* – это процесс группировки данных в классы или *кластеры* таким образом, что объекты одного кластера имеют высокую схожесть друг с другом и высокую степень отличия от объектов других классов. Степень различия между объектами определяется на основе значений атрибутов, описывающих объект.

Кластерный анализ, или *кластеризация*, – важная деятельность человека. Еще в раннем детстве человек учится различать кошек и собак или животных и растений.

Кластеризация широко применяется во многих областях, включая статистику, биологию, машинное обучение, распознавание образов, маркетинг, политологию, социологию и т.д. В бизнесе кластеризация может помочь маркетологам определить группы потребителей на основе имеющейся базы данных покупок. Кластеризация может помочь идентифицировать земельные площади одинакового использования в базе данных наблюдений земной поверхности. Страховые компании могут определить группы держателей полисов автомобильного страхования. Политические партии на основе опросов могут выявить

Рис. 5.1: Разбиение на кластеры

группы избирателей.

В отличие от классификации с обучением, кластерный анализ не имеет дело с предопределенными классами. Нет также и обучающей выборки, то есть примеров с известными метками классов. Поэтому кластерный анализ часто называют *классификацией без обучения*.

Этапы кластерного анализа следующие:

1. выявление подходящих классов;

2. описание каждого полученного класса.

На первом шаге применяются формальные алгоритмы кластеризации, рассматриваемые далее в этой главе.

Второй шаг предназначен для получения характерных свойств каждого класса. При этом часто применяются рассмотренные ранее алгоритмы классификации с обучением, используя метки классов, полученные на первом шаге.

5.2 Типы данных в кластерном анализе

В данном разделе мы изучим типы данных, возникающих в кластерном анализе, и то, как предварительно их обработать для такого анализа. Будем предполагать, что набор данных для кластеризации содержит n объектов, которые могут соответствовать людям, домам, документам, странам и т.д. Основные алгоритмы кластеризации обычно оперируют над следующими двумя структурами данных:

- **Матрица данных** представляет n объектов, таких как люди, с p *переменными* или *атрибутами*, такими как возраст, рост, вес, пол и т.д. Данная структура имеет форму реляционной таблицы, или матрицы размерностью $n \times p$ (n объектов на p переменных):

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- **Матрица различий** хранит коллекцию различий между всеми парами n объектов. Она может быть представлена таблицей $n \times n$:

$$\begin{pmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{pmatrix},$$

  где $d(i,j)$ – измеренное различие или расстояние между объектами i и j. Подразумевается, что $d(i,j)$ – неотрицательное число, близкое к нулю, когда объекты i, j очень близки. Чем больше $d(i,j)$, тем больше различия между i и j. При этом считаем $d(i,i) = 0, d(i,j) = d(j,i)$.

Многие алгоритмы кластеризации оперируют с матрицей различий. Если данные изначально представлены в виде матрицы данных, то перед применением этих алгоритмов сначала необходимо вычислить матрицу различий. Далее мы рассмотрим способы вычисления матрицы различий для объектов, описываемых *вещественнозначными*, *бинарными*, *номинальными* и *порядковыми* переменными, а также комбинацией этих видов переменных.

Вещественнозначные переменные

Вещественноначные переменные (interval-scaled variables) – это количественные измерения каких-либо свойств. Например, вес, рост, продолжительность, координаты по вертикали и по горизонтали, температура и т.д.

Используемый масштаб измерений при этом может оказывать влияние на результаты кластерного измерения. Например, переход в единицах измерения от метров на сантиметры, или от килограмм на фунты может привести к получению совсем другой кластерной структуры. Для избежания такой зависимости от выбора единиц измерения данные должны быть неким образом стандартизированы.

Для стандартизации необходимо преобразовать исходные измерения в безразмерные величины. Это можно сделать следующим образом:

1. Вычислить среднее абсолютное смещение, s_f:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|),$$

где $x_{1f}, \ldots, x_{nf}$ – n измерений переменной f, а m_f – среднее значение f, то есть $m_f = 1/n(x_{1f} + \cdots + x_{nf})$.

2. Вычислить стандартизированное измерение, или *z-оценку*:

$$z_{if} = \frac{x_{if} - m_f}{s_f}.$$

Среднее абсолютное смещение s_f более устойчиво при наличии шумов, чем среднеквадратическое отклонение σ_f, так как при вычислении s_f величины $|x_{2f} - m_f|$ не возводятся в квадрат и влияние "выбросов" уменьшается. В качестве меры смещений от среднего можно также использовать ввеличину *медианного абсолютного смещения*.

После стандартизации можно вычислять матрицу различий. Мерой различий для веществennозначных переменных обычно выбирается расстояние между парами объектов. Самыми популярными расстояниями являются:

1. Евклидово расстояние:

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + \cdots + |x_{ip} - x_{jp}|^2}.$$

2. Манхеттеновское расстояние (или расстояние в городских кварталах):

$$d(i,j) = |x_{i1} - x_{j1}| + \cdots + |x_{ip} - x_{jp}|.$$

3. Расстояние Минковского – обобщение двух вышеперечисленных расстояний:

$$d(i,j) = \left(|x_{i1} - x_{j1}|^q + \cdots + |x_{ip} - x_{jp}|^q\right)^{1/q},$$

где $q > 0$.

Стандартизацией измерений можно добиться того, чтобы все переменные имели равные веса. Это в частности бывает полезно в тех случаях, когда у исследователя нет предварительного знания данных. Однако в некоторых приложениях пользователь намеренно хочет дать некоторым переменным больший вес по сравнению с другими. Например, при кластеризации баскетболистов мы можем дать больший вес переменной "рост".

В случае, когда каждой переменной f назначен вес w_f, расстояние Минковского вычисляется следующим образом:

$$d(i,j) = \left(w_1|x_{i1} - x_{j1}|^q + \cdots + w_p|x_{ip} - x_{jp}|^q\right)^{1/q}.$$

Бинарные переменные

Переменная называется бинарной, если она может принимать только два значения: 0 и 1. При этом обычно значение 0 означает отсутствие какого-либо признака, а 1 – присутствие этого признака. Например, переменная *курильщик* принимает значение 1, если человек курит, и 0 если не курит. Часто при описании объектов имеются несколько бинарных переменных.

Одним из способов вычисления матрицы различий между двумя объектами i и j, описываемыми одной или несколькими (равнозначными) бинарными переменными, является использование *матрицы сопряженности*:

		Объект j		
		1	0	Сумма
Объект i	1	q	r	$q+r$
	0	s	t	$s+t$
	Сумма	$q+s$	$r+t$	p

размерности 2×2, где

- q – количество переменных, равных 1 для обоих объектов i и j;
- r – количество переменных, равных 1 для объекта i и равных 0 для объекта j;
- s – количество переменных, равных 0 для объекта i и равных 1 для объекта j;
- t – количество переменных, равных 0 для обоих объектов i и j;
- p – общее количество переменных: $p = q + r + s + t$.

Для расчета матрицы различий между объектами, целесообразно различить *симметричные* и *асимметричные* бинарные переменные.

Бинарная переменная **симметрична**, если оба ее возможных значения одинаково важны и имеют одинаковый вес (например, пол человека). Для случая симметричных бинарных переменных широко используемой мерой различий между объектами i и j является *простой коэффициент совпадений* (simple matching coefficient):

$$d(i,j) = \frac{r+s}{q+r+s+t}.$$

Бинарная переменная называется **асимметричной**, если ее возможные состояния не являются равноценными. Например,

переменная “наличие болезни”. Для таких переменных условимся, что более важное состояние (обычно это более *редкое* состояние), соответствует значению 1, а менее важное (менее редкое) – значению 0. Например, 0 – отрицательный результат на ВИЧ, а 1 – положительный.

При заданных двух асимметричных переменных совпадение двух единиц более важно, чем совпадение двух нулей. Для такого случая в качестве меры различий более всего подходит *коэффициент Джаккарда* (Jaccard coefficient), в котором количество совпадений нулей t игнорируется:

$$d(i,j) = \frac{r+s}{q+r+s}.$$

Если среди бинарных переменных присутствуют и симметричные, и асимметричные переменные, то в этом случае используется подход со смешанными типами переменных, описанный ниже.

Номинальные переменные

Номинальная переменная – это обобщение бинарной переменной в том смысле, что она может принимать более двух состояний. Например, переменная “цвет” может принимать, скажем, 4 значения: ’зеленый’, ’красный’, ’синий’, ’желтый’.

Пусть число состояний номинальной переменной равно M. Эти состояния могут обозначаться буквами, символами или целыми числами, например $1, 2, \ldots, M$. Заметим, что эти цифры используются только для обработки данных и не отражают никакого порядка.

Матрица различий может быть вычислена в соответствии с принципом *простых совпадений*:

$$d(i,j) = \frac{p-m}{p},$$

где m – количество переменных, для которых значения обоих объектов i и j совпали, p – общее количество переменных. Данную формулу можно модифицировать путем назначения больших весов переменным, имеющим большее количество состояний (соответственно, меньшую вероятность совпадений).

Номинальные переменные можно закодировать с помощью асимметричных бинарных переменных, по одной на каждое из M возможных состояний номинальной переменной. При данном значении номинальной переменной соответствующая данному состоянию бинарная переменная принимает значение 1, а остальные переменные

равны 0. Для вычисления матрицы различий можно использовать подход, описанный для случая асимметричных бинарных переменных.

Порядковые переменные

Дискретная порядковая переменная так же, как и номинальная переменная, может принимать несколько различных состояний. Отличие от номинальной переменной состоит в том, что M значений порядковой переменной естественным образом упорядочены. Например, образование может быть начальным, средним и высшим. Офицерские звания также упорядочены: лейтенант, ст. лейтенант, капитан, майор, подполковник и т.п. Порядковые переменные также могут быть получены в результате дискретизации непрерывных измерений путем разбиения всего множества возможных значений на интервалы. Например, порядковая переменная "возраст" может принимать значения 0..20,21..40,41..60,61..80,>80.

Пусть порядковая переменная f имеет M_f возможных состояний, закодированных значениями $1, \ldots, M_f$. Считаем, что порядок этих значений соответствует естественному порядку состояний переменной. Пусть $x_{if} \in \{1, \ldots, M_f\}$ – значение порядковой переменной f для объекта i.

Нормализуем переменную, масштабировав ее значения на отрезок $[0, 1]$:

$$z_{if} = \frac{x_{if} - 1}{M_f - 1}.$$

После этого для вычисления матрицы различий можно использовать методы, рассмотренные для веществennозначных переменных.

Комбинация переменных разных типов

Во многих реальных баз данных объекты описываются комбинацией переменных, имеющих разные типы. Пусть p – общее число переменных. В этом случае матрицу различий можно вычислить следующим образом:

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},$$

где индикатор $\delta_{ij}^{(f)} = 0$, если либо (1) x_{if} или x_{jf} – пропущено, либо (2) $x_{if} = x_{jf}$ и переменная f – асимметричная бинарная; во всех остальных

случаях $\delta_{ij}^{(f)} = 1$. Величина $d_{ij}^{(f)}$ вычисляется в зависимости от типа переменной f:

- Если переменная f – бинарная или номинальная, то $d_{ij}^{(f)} = 0$, если $x_{if} = x_{jf}$ и $d_{ij}^{(f)} = 1$ в обратном случае.
- Если переменная f – вещественнозначная, то

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}},$$

  где h пробегает по всем объектам, у которых известно значение переменной f.
- Если f – порядковая переменная, то вычисляем $z_{if} = \frac{x_{if}-1}{M_f-1}$, и рассматриваем z_{if} как вещественнозначную переменную.

Таким образом, матрица различий может быть вычислена и для случая, когда имеются несколько переменных различных типов.

5.3 Алгоритм k-средних

Пусть имеется база данных из n объектов, и пусть задано k – количество кластеров, которые требуется сформировать. Алгоритм разбивает объекты на k групп ($k \leq n$), где каждая группа представляет один кластер. Кластеры формируются таким образом, чтобы минимизировать некоторый критерий, часто называемый *функцией схожести*, так что объекты одного кластера «похожи» в то время как объекты разных кластеров «непохожи».

В алгоритме k-средних функция схожести формируется на основе *центров тяжести* кластеров. Данный алгоритм действует следующим образом. Во-первых, случайным образом выбираются k объектов, каждый из которых изначально представляет центр своего кластера. Каждый из оставшихся объектов сопоставляется тому кластеру, центр которого наиболее близок в смысле расстояния. В результате каждый объект принадлежит некоторому кластеру. После этого, для каждого полученного кластера заново вычисляется центр тяжести, и так далее. Данный процесс повторяется до тех пор, пока выбранный критерий не перестанет уменьшаться. При этом обычно используется критерий суммы квадратов ошибок:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - m_i)^2,$$

где E – сумма квадратов ошибок для всех объектов базы данных, p – точка в пространстве объектов, m_i – центр кластера C_i (и p, и m_i – многомерные вектора).

Алгоритм 5.1 Алгоритм k-средних.

Ввод: Количество кластеров k; Множество S, содержащее n объектов.
Вывод: Множество из k кластеров.

1: Выбрать случайным образом k объектов в качестве исходных центров кластеров.
2: **repeat**
3: **for all** $s \in S$ **do**
4: Найти номер кластера i, центр которого наиболее близок по расстоянию к s.
5: Назначить объект s в кластер с номером i.
6: **end for**
7: **for** $i = 1$ to k **do**
8: Рассчитать центр кластера с номером i на основе входящих в него объектов.
9: **end for**
10: **until** Никаких изменений не произошло

Итак, алгоритм k-средних пытается определить k разбиений так, чтобы минимизировать критерий суммы квадратов ошибок. Он работает достаточно хорошо в тех случаях, когда кластеры представляют из себя ограниченные выпуклые скопления, четко отделенные друг от друга. Данный метод достаточно масштабируем и эффективен, так как его вычислительная сложность равна $O(nkt)$, где n – количество объектов, k – количество кластеров, t – количество итераций. Обычно $t \ll n$ и $k \ll n$. Метод часто останавливается в точке локального минимума.

Ограничения и недостатки алгоритма k-средних следующие:

- Требуется, чтобы для каждого кластера было возможно определить центр. Однако в некоторых приложениях это невозможно, когда например используются категориальные атрибуты.

- Количество кластеров должно быть введено пользователем. Часто это бывает неудобно, так как k необходимо как-то выбрать.

- Алгоритм k-средних невозможно использовать в том случае, когда кластеры имеют невыпуклую или сильно вытянутую форму, или размеры кластеров сильно отличаются друг от друга.

- Метод слишком чувствителен к «выбросам», так как даже небольшое количество таких объектов может сильно повлиять на вычисление центров кластеров.

Для преодоления последней из перечисленных проблем алгоритма k-средних используется алгоритм k-медоидов.

5.4 Алгоритм k-медоидов

В данном алгоритме вместо центров тяжести для назначения кластеров используются *медоиды* – наиболее центрированные объекты кластеров.

Алгоритм 5.2 Алгоритм k-медоидов.

Ввод: Количество кластеров k; Множество S, содержащее n объектов.
Вывод: Множество из k кластеров.

1: Выбрать случайным образом множество $M = \{s_1, \ldots, s_k\} \subset S$ в качестве исходных медоидов кластеров.
2: Назначить все объекты из S в тот кластер, медоид которого наиболее близок по расстоянию к этому объекту.
3: Присвоим E значение критерия суммы квадратов ошибок при таком разбиении.
4: **repeat**
5: **for all** $s \in S \setminus M$ **do**
6: **for all** $m \in M$ **do**
7: Рассмотрим ситуацию, когда медоид m заменяется на объект s. Новое множество медоидов равно $M' = (M \setminus \{m\}) \cup \{s\}$.
8: Вычислим E' – значение критерия суммы квадратов ошибок при разбиении точек на кластеры в соответствии с новым множеством медоидов M'.
9: **if** E'<E **then**
10: Присваиваем M:=M'
11: Переходим к шагу 5.
12: **end if**
13: **end for**
14: **end for**
15: **until** Никаких изменений не произошло

5.5 Упражнения

5.1 Приведите способы вычисления различий между объектами, описываемыми следующими типами переменных:

(a) Асимметричные бинарные переменные;

(b) Номинальные переменные;

(c) Веществennозначные переменные.

5.2 Пусть имеются следующие наблюдения для переменной *Возраст*:

$$18, 22, 25, 42, 28, 43, 33, 35, 56, 28.$$

Стандартизируйте эту переменную следующим образом:

(a) Вычислите среднее абсолютное смещение переменной *Возраст*.

(b) Вычислите z-оценку для первых четырех наблюдений.

5.3 Для двух объектов, заданных кортежами (22,1,42,10) и (20,0,36,8):

(a) вычислите Евклидово расстояние между двумя объектами;

(b) вычислите Манхеттеновское расстояние между двумя объектами;

(c) вычислите расстояние Минковского между двумя объектами, используя $q = 3$.

5.4 Пусть имеется служба, в которую обращаются люди, которые хотят познакомиться с другими людьми для последующей переписки. О себе они сообщают различные сведения, которые используются службой для выбора наиболее подходящих партнеров по переписке. Пусть такими сведениями будут: имя, пол, черта-1, черта-2, черта-3, черта-4. Таблица данных имеет вид:

Имя	*Пол*	*Черта-1*	*Черта-2*	*Черта-3*	*Черта-4*
Артем	М	Нет	Да	Да	Нет
Марина	Ж	Нет	Да	Да	Нет
Илья	М	Да	Нет	нет	Да

Считаем, что *пол* – симметричная переменная, а переменные *черта-N* – асимметричные, для которых состоянию “Да” соответствует значение 1, а состоянию “Нет” – значение 0. Предположим, что различие между объектами вычисляется только на основе асимметричных переменных.

(a) Изобразите *метрицу сопряженности* для всех возможных пар из списка “Артем, Марина, Илья”

(b) Для всех пар вычислите *простой коэффициент совпадений*.

(c) Для всех пар вычислите *коэффициент Джаккарда*.

(d) Выберите наилучшую на ваш взгляд пару для переписки. Какая из пар наименее совместима?

(e) Предположим, мы включили в анализ переменную “Пол”. Какая пара теперь является наиболее совместимой при использовании коэффициента Джаккарда?

5.5 Даны следующие восемь точек на плоскости, которые необходимо разбить на 3 кластера по методу k-средних:

$$(2, 10), (2, 5), (8, 4), (5, 8), (7, 5), (6, 4), (1, 2), (4, 9).$$

Используется Евклидово расстояние на плоскости. Пусть в качестве начальных центров кластеров выбраны точки $(2, 10), (5, 8), (1, 2)$. Найдите

(a) центры трех кластеров после выполнения первой итерации в методе k-средних;

(b) Итоговые три кластера.

Глава 6

Введение в теорию нечетких множеств

6.1 Нечеткие множества

Часто в жизни человек оперирурует нечеткими понятиями, например «много» и «мало», «молодой» и «старый», «богатый» и «бедный», «опытный» и «малоопытный», «чистый», «грязный», «очень грязный», и т.д. Очень часто такие понятия использовать намного удобнее, чем четко определять возраст, доход, стаж работы.

Например, рассмотрим гипотетическое правило выдачи кредита: выдавать кредит тем, кто работает больше 2 лет и имеет доход не менее 15000 рублей в месяц. Это значит, что человек получит кредит только тогда, когда зарабатывает 15000 рублей в месяц, но не получит, если имеет доход к примеру 14995 рублей. Такое четкое разделение обычно бывает неудобным.

То же самое касается, например, скорости автомобиля. Если четко определить, что автомобиль едет быстро, если он едет со скоростью 60 км/ч и выше, то получится, что автомобиль, движущийся со скоростью 59 км/ч едет не быстро. Интуитивно ясно, что такое определение быстроты выглядит не совсем естественно.

Предположим, вы договариваетесь встретиться с приятелем на следующий день в 10 часов утра. Скорее всего, ваш друг придет в любое время в окрестности 10 часов утра, скажем, с 9:50 до 10:10. Мы можем нарисовать график *функции принадлежности*, осью абсцисс которого является время, а осью ординат – степень принадлежности времени к окрестности 10:00. Данная функция имеет пик в точке 10:00 и приближается к нулю по мере удаления от этой точки. Таким образом

Рис. 6.1: S- и π-функции

определяется *нечеткое множество* точек, лежащих в окрестности 10:00.

Теория нечетких множеств была впервые рассмотрена в 1965 году математиком Заде (Zadeh), как средство представления неточностей в повседневной жизни. Эта теория дает приближенные и в то же время эффективные средства для описания характеристик произвольной системы, которая слишком сложна или плохо поддается точному математическому анализу. Нечеткие множества широко используются в задачах классификации в многочисленных областях, включая медицину и финансы.

В задачах классификации и кластеризации возникают случаи, когда классы перекрываются, что приводит к неточности. В обычной технике классификации предполагается, что объект всегда принадлежит только одному классу. Физически это условие бывает невыполнимо вследствие нечеткости или случайности. Для найденных паттернов мы должны предусмотреть возможность задания степени принадлежности более чем одному классу.

Пусть R – некоторое континуальное множество точек. *Нечеткое множество* A определяется *функцией принадлежности* $\mu_A : R \to [0, 1]$. При этом значение $\mu_A(r)$ задает степень принадлежности элемента $r \in R$ множеству A.

Если $\mu_A(r) = 1$, то элемент r строго принадлежит множеству A, если $\mu_A(r) = 0$, то элемент r строго не принадлежит множеству A.

Часто для задания функций принадлежности бывает удобно использовать стандартизованные S и π функции:

$$S(r; \alpha, \gamma) = \begin{cases} 0, & \text{если } r \le \alpha \\ 2\left(\frac{r-\alpha}{\gamma-\alpha}\right)^2, & \text{если } \alpha \le r \le (\alpha+\gamma)/2 \\ 1 - 2\left(\frac{r-\alpha}{\gamma-\alpha}\right)^2, & \text{если } (\alpha+\gamma)/2 \le r \le \gamma \\ 1, & \text{если } r \ge \gamma, \end{cases}$$

$$\pi(r; c, \lambda) = \begin{cases} S(r; c-\lambda, c), & \text{если } r \le c \\ 1 - S(r; c, c+\lambda), & \text{если } r \ge c. \end{cases}$$

Пример. Рассмотрим нечеткие множества, соответствующие понятиям «молодой» и «пожилой». Здесь R – это множество возрастов: $R = \mathbb{R}^+$. Тогда мы можем, например, рассмотреть следующие функции

Рис. 6.2: Лингвистические значения “молодой”,“пожилой”

принадлежности:

$$\mu_{\text{молодой}} = 1 - S(20, 40),$$
$$\mu_{\text{пожилой}} = S(50, 70).$$

6.2 Операции над нечеткими множествами

Перечислим основные отношения и операции, определенные для нечетких множеств (здесь A, B, C – произвольные нечеткие множества):

- $A = B \Leftrightarrow \mu_A(r) = \mu_B(r)$ для всех $r \in R$;
- $A = \overline{B} \Leftrightarrow \mu_A(r) = 1 - \mu_B(r)$ для всех $r \in R$;
- $A \subseteq B \Leftrightarrow \mu_A(r) \leq \mu_B(r)$ для всех $r \in R$;
- $C = A \cup B \Leftrightarrow \mu_C(r) = \max(\mu_A(r), \mu_B(r))$ для всех $r \in R$;
- $C = A \cap B \Leftrightarrow \mu_C(r) = \min(\mu_A(r), \mu_B(r))$ для всех $r \in R$;

Часто в языке используются модификаторы «не», «очень», «более или менее». На примере лингвистического значения «молодой» они могут быть определены следующим образом:

$$\mu_{\text{не молодой}} = 1 - \mu_{\text{молодой}},$$
$$\mu_{\text{очень молодой}} = \mu^2_{\text{молодой}},$$
$$\mu_{\text{не очень молодой}} = 1 - \mu^2_{\text{молодой}},$$
$$\mu_{\text{более или менее молодой}} = \mu^{1/2}_{\text{молодой}}.$$

Нечеткие множества широко используются во многих практических областях. Обширными областями их применения являются: задачи управления техническими системами, где параметры систем известны только приближенно; теория оптимизации с нечеткими ограничениями; медицинская и техническая диагностика, и т.д.

В царстве технологий интеллектуального анализа данных роль нечетких множеств без сомнений растет. Многие системы анализа данных уже реализованы с их использованием.

В задаче поиска ассоциативных правил они используются для того, чтобы получаемые правила были сформулированы как можно понятнее

для человека. Например, человеку гораздо понятнее правило «Если облачно и давление очень низкое, то будет сильный дождь», чем «Если концентрация облаков>x и давление<y, то количество осадков>z », где x, y, z – какие-то конкретные числа.

Кроме того, при использовании нечетких множеств и лингвистических переменных отпадает необходимость в искусственной дискретизации количественных атрибутов. Например, нет необходимости делить возраст на интервалы, такие как $0-20, 20-35, 35-50, 50-70, >70$.

В кластерном анализе подход с использованием нечеткости используется в тех случаях, когда невозможно отнести любой объект какому-то классу. В таком случае говорят лишь о степени принадлежности объекта данному классу. Рассмотрим простой алгоритм нечеткой кластеризации, который называется *алгоритм нечетких k-средних*.

6.3 Алгоритм нечетких k-средних

Данный алгоритм разбивает множество n объектов $\{x_i\}_{i=\overline{1,n}}$ на k кластеров. Степень принадлежности объекта i классу j определяется величиной $\mu_{ij} > 0$, при этом должно выполняться соотношение

$$\sum_{j=1}^{k} \mu_{ij} = 1 \quad \forall i. \tag{6.1}$$

Разбиение на классы выбирается таким образом, чтобы минимизировать целевую функцию:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} \mu_{ij}^{c} \|x_i - m_j\|^2,$$

где c – некоторый параметр, m_j – центр тяжести кластера j:

$$m_j = \frac{\sum_{i=1}^{n} \mu_{ij}^{c} x_i}{\sum_{i=1}^{n} \mu_{ij}^{c}}, \tag{6.2}$$

Можно показать (см. упражнения), что минимум функции J при заданных значениях m_i по переменным μ_{ij} при условии (6.1) достигается, когда

$$\mu_{ij} = \left(\sum_{s=1}^{k} \left(\frac{\|x_i - m_j\|}{\|x_i - m_s\|} \right)^{\frac{2}{c-1}} \right)^{-1}. \tag{6.3}$$

Схема алгоритма следующая:

1. Случайным образом выбираем $m_1, \ldots, m_k$.
   Устанавливаем $\mu_{ij} = 1/k$.

2. Для всех i от 1 до n, для всех j от 1 до k вычисляем величины μ_{ij} по формуле (6.3). Если при этом максимальное изменение μ_{ij} не превысило величины ε, то прекращаем работу алгоритма. Иначе переходим к следующему пункту.

3. Определяем m_i в соответствии с (6.2). Переходим к шагу 2.

6.4 Упражнения

6.1 Приведите примеры. где используются нечеткие множества. Поясните, в чем преимущества использования нечеткости.

6.2 Убедитесь в справедливости правил де Моргана для нечетких множеств:

$$\overline{A \cap B} = \overline{A} \cup \overline{B} \quad \text{и} \quad \overline{A \cup B} = \overline{A} \cap \overline{B}.$$

6.3 Приведите пример нечеткого множества A, такого что $A \cap \overline{A} \neq \emptyset$.

6.4 Приведите пример нечеткого множества A, такого что $A \cup \overline{A} \neq \Omega$.

6.5 Докажите формулу (6.3).

Литература

[1] Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Классификация и снижение размерности. – М.: Финансы и статистика. – 1989.

[2] Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург. – 2004.

[3] Кофман А. Введение в теорию нечетких множеств. – М.: Радио и связь. – 1982.

[4] Han J., Kamber M., Data mining: Concepts and Techniques. – Morgan Kaufmann Publishers. – 2001.

[5] Konar A., Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. – CRC Press LLC. – Boca Raton, Florida/ – 2000.

[6] Mitra S., Acharya T., Data Mining. Multimedia, Soft Computing, and Bioinformatics. – John Wiley & Sons, Inc. – Hoboken, New Jersey. – 2003.