ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
Московский технический университет связи и информатики
Кафедра математической кибернетики и информационных технологий

М.В.Яшина, М.С.Мосева

Учебно-методическое пособие

по дисциплине

ИНТЕЛЛЕКТУАЛЬНЫЕ БАЗЫ ДАННЫХ

(для направления 09.04.01)

Москва 2018

План УМД на 2017/18 уч.г.

Учебно-методическое пособие

по дисциплине

ИНТЕЛЛЕКТУАЛЬНЫЕ БАЗЫ ДАННЫХ

Составители: М.В.Яшина
М.С.Мосева

Издание утверждено советом факультета ИТ. Протокол № 11 от 23.05.1917 г.

Рецензент А.Г.Таташев, д.ф.-м.н., профессор

СОДЕРЖАНИЕ

Введение

Развитие информационного общества и сопровождающая этот процесс вычислительная техника прошли следующие этапы развития.

1. Возникновение ручного счета – с давних времен.
2. Возникновение письменности.
3. Книгопечатание - середина XVI века.
4. Механический этап развития – середина XVII века.
5. Изобретение электромеханических генераторов – конец XIX века.
6. Телекоммуникационные сети, интернет – до XX века.

Следует отметить характерные черты, свойственные этапам развития вычислительной техники.

- 1-е поколение (начало 50-х гг.). Элементная база — электронные лампы. ЭВМ отличались большими габаритами, большим потреблением энергии, малым быстродействием, низкой надежностью, программированием в кодах.

- 2-е поколение (с конца 50-х гг.). Элементная база — полупроводниковые элементы. Улучшились по сравнению с ЭВМ предыдущего поколения все технические характеристики. Для программирования используются алгоритмические языки.

- 3-е поколение (начало 60-х гг.). Элементная база — интегральные схемы, многослойный печатный монтаж. Резкое снижение габаритов ЭВМ, повышение их надежности, увеличение производительности. Доступ с удаленных терминалов.

- 4-е поколение (с середины 70-х гг.). Элементная база — микропроцессоры, большие интегральные схемы. Улучшились технические характеристики. Массовый выпуск персональных компьютеров. Направления развития: мощные многопроцессорные вычислительные системы с высокой производительностью, создание дешевых микроЭВМ.

- 5-е поколение (с середины 80-х гг.). Началась разработка интеллектуальных компьютеров, пока не увенчавшаяся успехом. Внедрение во все сферы компьютерных сетей и их объединение, использование распределенной обработки данных, повсеместное применение компьютерных информационных технологий.

Последняя информационная революция выдвигает на первый план новую отрасль — информационную индустрию, связанную с производством технических средств, методов, технологий для производства новых знаний. Важнейшими составляющими информационной индустрии становятся все виды информационных технологий, особенно телекоммуникации. Современная

информационная технология опирается на достижения в области компьютерной техники и средств связи.

Информационная технология (ИТ) — процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления.

Телекоммуникации — дистанционная передача данных на базе компьютерных сетей и современных технических средств связи.

Поколение ЭВМ	Характеристики			
	I	II	III	IV
Годы применения	1946-1958	1959-1963	1964-1976	1977 - настоящее время
Элементная база	Эл. лампа	Транзистор	ИС	БИС
Количество ЭВМ в мире (шт.)	Десятки	Тысячи	Десятки тысяч	Миллионы
Быстродейст-вие (операций в секунду)	До 10^5	До 10^6	До 10^7	Более 10^7
Характерные типы ЭВМ поколений		Малые, средние, большие, специальные	Большие, средние, мини- и микроЭВМ	СуперЭВМ, ПК, специальные, общие, сети ЭВМ
Типичные модели поколений	UNIVAC, БЭСМ	БЭСМ-6	1BM/360, СМ ЭВМ	ШМ/360, SX-2, 1BM PC/XT/AT, P5/2 сети
Носители информации	Перфокарта, перфолента	Магнитная лента	Диск	Гибкий, жесткий, лазерный диск и др.
Характерное программное обеспечение	Коды, автокоды, ассемблеры	Языки программи-рования	ППП, СУБД, САПРы	Системы параллельного программиро-вания и др.

Современные проблемы общества приводят к исследованиям методов обработки больших данных (BigData), поиск информации в сетевых ресурсах (DataMining), интернет вещей (IoT), новейших технологий производства (3d-printing) и т.д. Хранилище данных, базы данных, как одно из ключевых направлений развития информационных систем, так же эволюционирует в направлении интеллектуализации средств управления и обработки.

В данном учебно-методическом пособии рассмотрены базовые сведения и основные направления развития современных Интеллектуальных Баз Данных.

1. Основные понятия и модели баз данных

Одной из основных конструкций современных информационных систем является концепция баз данных. Согласно этой концепции, основой информационных технологий являются данные, которые должны быть организованы в базы данных в целях адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей.

Одним из важнейших понятий в теории баз данных является понятие информации. Под информацией понимаются любые сведения о каком-либо событии, процессе, объекте.

Данные — это информация, представленная в определенном виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством. Для компьютерных технологий данные — это информация в дискретном, фиксированном виде, удобная для хранения, обработки на ЭВМ, а также для передачи по каналам связи.

База данных (БД) — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, или иначе БД — это совокупность взаимосвязанных данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений в определенной предметной области. БД состоит из множества связанных файлов.

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Автоматизированная информационная система (АИС) — это система, реализующая автоматизированный сбор, обработку, манипулирование данными, функционирующая на основе ЭВМ и других технических средств и включающая соответствующее программное обеспечение (ПО) и персонал. В дальнейшем в этом качестве будет использоваться термин информационная система (ИС), который подразумевает понятие автоматизированная.

Каждая ИС в зависимости от ее назначения имеет дело с той или иной частью реального мира, которую принято называть предметной областью (ПрО) системы. Выявление ПрО — это необходимый начальный этап разработки любой ИС. Именно на этом этапе определяются информационные потребности всей совокупности пользователей будущей системы, которые, в свою очередь, предопределяют содержание ее базы данных.

Банк данных (БнД) является разновидностью ИС. БнД — это система специальным образом организованных данных: баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

СУБД

Можно дать следующую обобщенную характеристику возможностям современных СУБД.

1. СУБД включает язык определения данных, с помощью которого можно определить базу данных, ее структуру, типы данных, а также средства задания ограничений для хранимой информации. В многопользовательском варианте СУБД этот язык позволяет формировать представления как некоторое подмножество базы данных, с поддержкой которых пользователь может создавать свой взгляд на хранимые данные, обеспечивать дополнительный уровень безопасности данных и многое другое.

2. СУБД позволяет вставлять, удалять, обновлять и извлекать информацию из базы данных посредством языка управления данными.

3. Большинство СУБД могут работать на компьютерах с разной архитектурой и под разными операционными системами, причем на работу пользователя при доступе к данным практически тип платформы влияния не оказывает.

4. Многопользовательские СУБД имеют достаточно развитые средства администрирования БД.

5. СУБД предоставляет контролируемый доступ к базе данных с помощью:

- системы обеспечения безопасности, предотвращающей несанкционированный доступ к информации базы данных;
- системы поддержки целостности базы данных, обеспечивающей непротиворечивое состояние хранимых данных;
- системы управления параллельной работой приложений, контролирующей процессы их совместного доступа к базе данных;
- системы восстановления, позволяющей восстановить базу данных до предыдущего непротиворечивого состояния, нарушенного в результате аппаратного или программного обеспечения.

Архитектура СУБД

Одним из важнейших аспектов развития СУБД является идея отделения логической структуры БД и манипуляций данными, необходимыми пользователям, от физического представления, требуемого компьютерным оборудованием.

Одна и та же БД в зависимости от точки зрения может иметь различные уровни описания. По числу уровней описания данных, поддерживаемых СУБД, различают одно-, двух- и трехуровневые системы. В настоящее время чаще всего поддерживается трехуровневая архитектура описания БД (рис. 1.1), с тремя уровнями абстракции, на которых можно рассматривать базу данных. Такая архитектура включает:

- внешний уровень, на котором пользователи воспринимают данные, где отдельные группы пользователей имеют свое представление (ПП) на базу данных;

- внутренний уровень, на котором СУБД и операционная система воспринимают данные;

- концептуальный уровень представления данных, предназначенный для отображения внешнего уровня на внутренний уровень, а также для обеспечения необходимой их независимости друг от друга; он связан с обобщенным представлением пользователей.

Под задачами обработки данных обычно понимается специальный класс решаемых на ЭВМ задач, связанных с видом, хранением, сортировкой, отбором по заданному условию и группировкой записей однородной структуры.

Отдельные программы или комплекс программ, реализующие автоматизацию решения прикладных задач обработки данных, называются приложениями. Приложения, созданные средствами СУБД, относят к приложениям СУБД. Приложения, созданные вне среды СУБД с помощью систем программирования, использующих средства доступа к БД, к примеру, Delphi или VisualStudio, называют внешними приложениями.

Моделью данных называется формализованное описание структуры единиц информации и операций над ними в информационной системе.

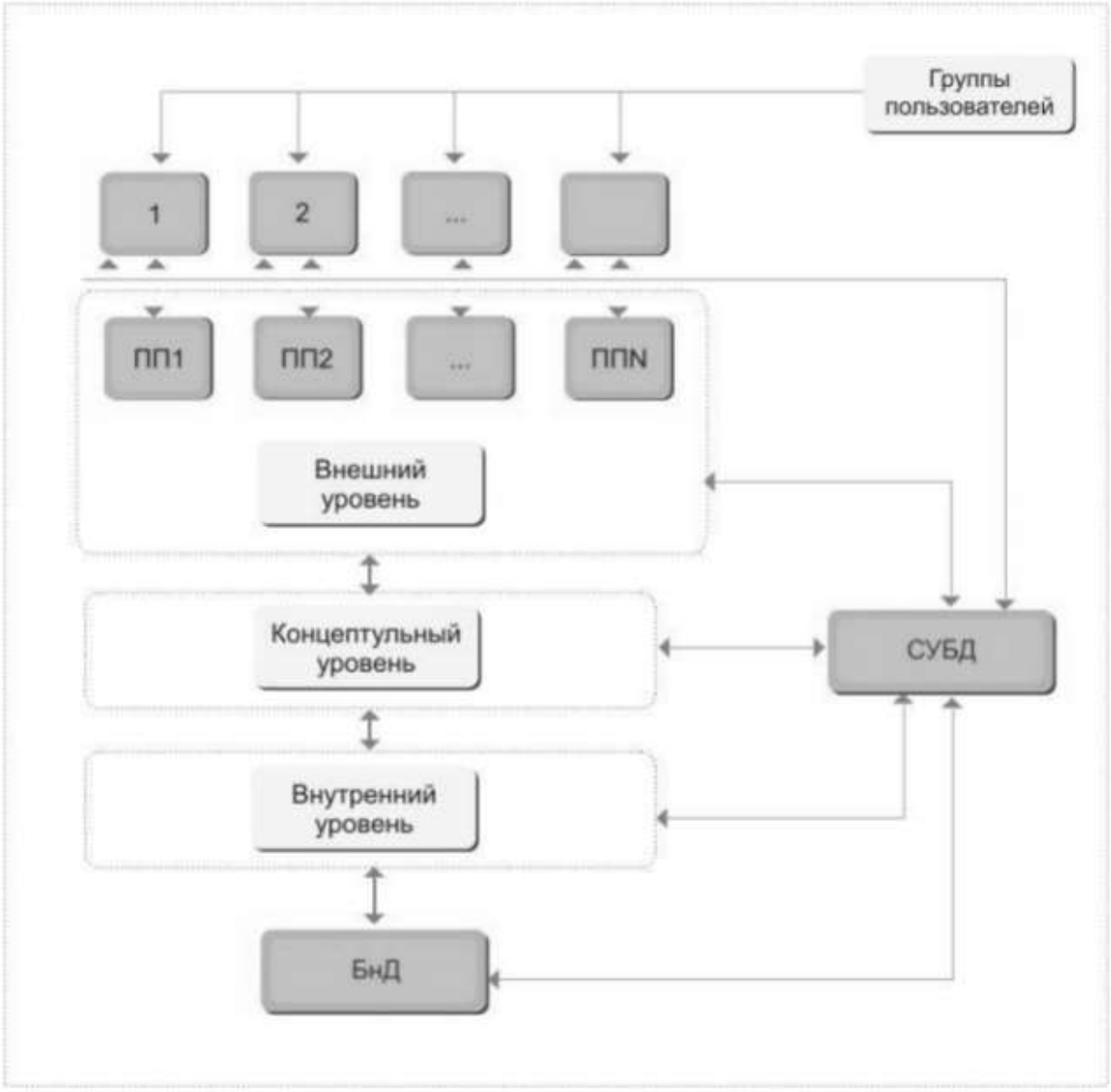

Рисунок 1.1 - Трехуровневая архитектура СУБД

Модель данных — это некоторая абстракция, в которой отражаются самые важные аспекты функционирования выделенной предметной области, а второстепенные — игнорируются. Модель данных включает в себя набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные. В модели данных различают три главные составляющие:

• структурную часть, определяющую правила порождения допустимых для данной СУБД видов структур данных;

• управляющую часть, определяющую возможные операции над такими структурами;

• классы ограничений целостности данных, которые могут быть реализованы средствами этой системы.

Классификация моделей баз данных

Каждая СУБД поддерживает ту или иную модель данных.

По существу модель данных, поддерживаемая механизмами СУБД, полностью определяет множество конкретных баз данных, которые могут быть созданы средствами этой системы, а также способы модификации состоянии БД с целью отображения тех изменений, которые происходят в предметной области.

Общие логические модели данных для баз данных:

- иерархическая модель данных,
- сетевая модель,
- реляционная модель,
- ER-модель,
- расширенная модель сущностных отношений,
- объектная модель.

1.1 Реляционная модель данных

Создателем реляционной модели является сотрудник фирмы IBM доктор Э.Ф. Кодд. Будучи по образованию математиком, Э.Ф. Кодд предложил использовать для обработки данных аппарат теории множеств. В статье "A Relational Model of Data for Large Shared Data Banks", вышедшей в свет в 1970 году, он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как отношение (relation).

Положив теорию отношений в основу реляционной модели, Э.Ф. Кодд обосновал реляционную замкнутость отношений и ряда некоторых специальных операций, которые применяются сразу ко всему множеству строк отношения, а не к отдельной строке. Указанная реляционная замкнутость означает, что результатом выполнения операций над отношениями является также отношение, над которым в свою очередь можно осуществить некоторую операцию. Из этого следует, что в данной модели можно оперировать реляционными выражениями, а не только отдельными операндами в виде простых имен таблиц.

Одним из основных преимуществ реляционной модели является ее однородность. Все данные рассматриваются как хранимые в таблицах и только в таблицах. Каждая строка такой таблицы имеет один и тот же формат.

Основные понятия реляционной модели

В реляционной модели объекты реального мира и взаимосвязи между ними представляются с помощью совокупности связанных между собой *таблиц* (отношений).

Даже в том случае, когда функции *СУБД* используются для выбора информации из одной или нескольких *таблиц* (т.е. выполняется ***запрос***), результат также представляется в табличном виде. Более того, можно выполнить *запрос* с применением результатов другого *запроса*.

Каждая ***таблица*** БД представляется как совокупность ***строк*** и *столбцов*, где *строки* (записи) соответствуют экземпляру объекта, конкретному событию или явлению, а ***столбцы*** (поля) – атрибутам (признакам, характеристикам, параметрам) объекта, события, явления.

В каждой *таблице* БД необходимо наличие ***первичного ключа*** – так именуют поле или набор полей, однозначно идентифицирующий каждый экземпляр объекта или запись. Значение *первичного ключа* в *таблице* БД должно быть уникальным, т.е. в *таблице* не допускается наличие двух и более записей с одинаковыми значениями *первичного ключа*. Он должен быть минимально достаточным, а значит, не содержать полей, удаление которых не отразится на его уникальности.

Реляционная алгебра

Для управления реляционной базой данных Э.Ф. Кодд ввел реляционные языки обработки данных — реляционную алгебру и реляционное исчисление.

Реляционная алгебра — это процедурный язык обработки реляционных таблиц. Это означает, что в реляционной алгебре используется пошаговый подход к созданию реляционных таблиц, содержащих ответы на запросы.

Реляционное исчисление — непроцедурный язык. В реляционном исчислении запрос создается путем определения таблицы запроса за один шаг.

Кодд показал логическую эквивалентность реляционной алгебры и реляционного исчисления. Это означает, что любой запрос, который можно сформулировать при помощи реляционного исчисления, также можно сформулировать, пользуясь реляционной алгеброй, и наоборот.

И реляционная алгебра, и реляционное исчисление в том виде, как они были сформулированы Коддом, являются теоретическими языками.

1.2 Особенности проектирования БД

Нормализация БД

Для устранения рассмотренных выше недостатков и применяется процесс нормализации отношений. Данный процесс — это формальный метод анализа отношений на основе их первичных или потенциальных ключей и существующих функциональных зависимостей. Он включает ряд формальных правил, используемых для проверки всех отношений базы данных. Различают:

1НФ — первую нормальную форму;
2НФ — вторую нормальную форму;
3НФ — третью нормальную форму;
НФБК — нормальную форму Бойса — Кодда;
4НФ — четвертую нормальную форму;
5НФ — пятую нормальную форму.

Каждая нормальная форма налагает определенные ограничения на данные. Эти ограничения вводятся в каждом конкретном отношении, и соблюдение этих ограничений в отношении связано уже с наличием нормальной формы.

1НФ, 2НФ, 3НФ — ограничивают зависимость непервичных атрибутов от ключей.

НФБК — ограничивает зависимость первичных атрибутов.

4НФ — формулирует ограничения на виды многозначных зависимостей.

5НФ — вводит другие типы зависимостей: зависимости соединений.

Процесс перехода от нормальной формы более низкого уровня к нормальной форме более высокого уровня и называется нормализацией отношений (НО).

Для реляционных баз данных необходимо, чтобы все отношения базы данных обязательно находились в 1НФ. Нормальные формы более высокого порядка могут использоваться разработчиками по своему усмотрению. Однако следует стремиться к тому, чтобы довести уровень нормализации базы данных хотя бы до 3НФ, тем самым исключив из базы данных избыточность данных и аномалии обновления.

2. Особенности современных информационных распределенных систем

Основы программирования реляционных данных на языке SQL

Единственным средством общения и *администраторов баз данных*, и проектировщиков, и разработчиков, и пользователей с реляционной базой данных является *структурированный язык* запросов SQL (*Structured Query Language*). SQL есть полнофункциональный язык манипулирования данными в реляционных базах данных. В настоящее время он является общепризнанным, стандартным интерфейсом для реляционных баз данных, таких как Oracle, *Informix, Sybase,* DB/2, MS SQL Server и ряда других (стандарты ANSI и ISO). SQL - непроцедурный язык, который предназначен для обработки множеств, состоящих из строк и колонок таблиц реляционной базы данных. Хотя существуют его расширения, допускающие процедурную обработку. Проектировщики баз данных используют SQL для создания всех физических объектов реляционной базы данных.

Описание основных операторов SQL

SQL состоит из набора команд манипулирования данными в реляционной базе данных, которые позволяют создавать объекты реляционной базы данных, модифицировать данные в таблицах (вставлять, удалять, исправлять), изменять *схемы отношений* базы данных, выполнять вычисления над данными, делать выборки из базы данных, поддерживать безопасность и целостность данных.

Весь набор команд SQL можно разбить на следующие группы:

- команды определения данных (*DDL – Data Defininion Language*);
- команды манипулирования данными (*DML – Data Manipulation Language*);
- команды выборки данных (DQL - Data *Query Language*);
- команды управления транзакциями;
- команды управления данными.

Определение структур базы данных (DDL)

Язык определения данных (Data Definition Language, DDL) позволяет создавать и изменять структуру объектов *базы данных*, например, создавать и удалять *таблицы*. Основными командами языка DDL являются следующие: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX.

Манипулирование данными (DML)

Язык манипулирования данными (Data Manipulation Language, DML) используется для манипулирования информацией внутри объектов *реляционной базы данных* посредством трех основных команд: INSERT, UPDATE, DELETE.

Выборка данных (DQL)

Язык *запросов* DQL наиболее известен пользователям *реляционной базы данных*, несмотря на то, что он включает всего одну команду SELECT. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования *запросов* к *реляционной базе данных*.

Язык управления данными (DCL – Data Control Language)

Команды управления данными позволяют управлять доступом к информации, находящейся внутри *базы данных*. Как правило, они используются для создания объектов, связанных с доступом к данным, а также служат для контроля над распределением привилегий между пользователями. Команды управления данными следующие: GRANT, REVOKE.

Команды администрирования данных

С помощью команд администрирования данных пользователь осуществляет контроль за выполняемыми действиями и анализирует операции *базы данных*; они также могут оказаться полезными при анализе производительности системы. Не следует путать администрирование данных с администрированием *базы данных*, которое представляет собой общее управление *базой данных* и подразумевает использование команд всех уровней.

Команды управления транзакциями

Существуют следующие команды, позволяющие управлять транзакциями базы данных: COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.

3. Интеллектуальные базы данных

Интеллектуальные базы данных отличаются от обычных баз данных возможностью выборки по запросу необходимой информации, которая может явно не храниться, а выводиться из имеющейся в базе данных. Примерами таких запросов могут быть следующие:

- “Вывести список товаров, цена которых выше среднеотраслевой”,

- “Вывести список товаров-заменителей некоторой продукции”,

- “Вывести список потенциальных покупателей некоторого товара” и т.д.

Для выполнения первого типа запроса необходимо сначала проведение статистического расчета среднеотраслевой цены по всей базе данных, а уже после этого собственно отбор данных. Для выполнения второго типа запроса необходимо вывести значения характерных признаков объекта, а затем поиск по ним аналогичных объектов. Для третьего типа запроса требуется сначала определить список посредников-продавцов, выполняющих продажу данного товара, а затем провести поиск связанных с ними покупателей.

Во всех перечисленных типах запросов требуется осуществить поиск по условию, которое должно быть доопределено в ходе решения задачи. Интеллектуальная система без помощи пользователя по структуре базы данных сама строит путь доступа к файлам данных. Формулирование запроса осуществляется в диалоге с пользователем, последовательность шагов которого выполняется в максимально удобной для пользователя форме. Запрос к базе данных может формулироваться и с помощью естественно-языкового интерфейса.

Для извлечения значимой информации из баз данных используются специальные методы (DataMining), основанные или на применении методов математической статистики, индуктивных методов построения деревьев решений, или нейронных сетей. Формулирование запроса осуществляется в результате применения интеллектуального интерфейса, позволяющего в диалоге гибко определять значимые признаки анализа.

4. Методология DataMining

DataMining – это сочетание широкого математического инструментария (от классического статистического анализа до новых кибернетических методов) и последних достижений в сфере информационных технологий. В технологии DataMining гармонично объединились строго формализованные методы и методы неформального анализа, т.е. количественный и качественный анализ данных.

DataMining (добыча данных, интеллектуальный анализ данных, глубинный анализ данных) — собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой

деятельности. Термин введён Григорием Пятецким-Шапиро в 1989 году.

Основу методов DataMining составляют всевозможные методы классификации, моделирования и прогнозирования. К методам DataMining нередко относят статистические методы (дескриптивный анализ, корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ, компонентный анализ, дискриминантный анализ, анализ временных рядов). Такие методы, однако, предполагают некоторые априорные представления об анализируемых данных, что несколько расходится с целями DataMining (обнаружение ранее неизвестных нетривиальных и практически полезных знаний).

Одно из важнейших назначений методов DataMining состоит в наглядном представлении результатов вычислений, что позволяет использовать инструментарий DataMining людьми, не имеющих специальной математической подготовки. В то же время, применение статистических методов анализа данных требует хорошего владения теорией вероятностей и математической статистикой.

4.1 Методы и алгоритмы DataMining

К методам и алгоритмам DataMining относятся:

- искусственные нейронные сети,
- деревья решений, символьные правила,
- нечеткая логика,
- методы ближайшего соседа и k-ближайшего соседа,
- метод опорных векторов,
- байесовские сети,
- линейная регрессия,
- корреляционно-регрессионный анализ,
- иерархические методы кластерного анализа,
- неиерархические методы кластерного анализа, в том числе алгоритмы k-средних и k-медианы,
- методы поиска ассоциативных правил, в том числе алгоритм Apriori,
- метод ограниченного перебора,
- эволюционное программирование и генетические алгоритмы,
- разнообразные методы визуализации данных и множество других методов.

Большинство аналитических методов, используемых в технологии DataMining – это известные математические алгоритмы и методы. Новым в их применении является возможность их

использования при решении тех или иных конкретных проблем, обусловленная появившимися техническими и программными средствами. Следует отметить, что большинство методов DataMining были разработаны в рамках теории искусственного интеллекта.

Метод представляет собой норму или правило, определенный путь, способ, прием решений задачи теоретического, практического, познавательного, управленческого характера.

4.2 Нечеткая логика

В настоящее время большинство методов DataMining основано на нечеткой логике. Предметом нечёткой логики считается исследование рассуждений в условиях нечёткости, размытости, сходных с рассуждениями в обычном смысле, и их применение в вычислительных системах. Нечеткая логика — это обобщение традиционной аристотелевой логики на случай, когда истинность рассматривается как лингвистическая переменная, принимающая значения типа: "очень истинно", "более-менее истинно", "не очень ложно" и т.п. Указанные лингвистические значения представляются нечеткими множествами.

Лингвистическая переменная отличается от числовой переменной тем, что ее значениями являются не числа, а слова или предложения в естественном или формальном языке. Поскольку слова в общем менее точны, чем числа, понятие *лингвистической переменной* дает возможность приближенно описывать явления, которые настолько сложны, что не поддаются описанию в общепринятых количественных терминах. В частности, *нечеткое множество*, которое представляет собой ограничение, связанное со значениями *лингвистической переменной*, можно рассматривать как совокупную характеристику различных подклассов элементов *универсального множества*. В этом смысле роль нечетких множеств аналогична той роли, которую играют слова и предложения в естественном языке. Например, прилагательное "КРАСИВЫЙ" отражает комплекс характеристик внешности индивидуума. Это прилагательное можно также рассматривать как название нечеткого *множества*, которое является ограничением, обусловленным *нечеткой переменной* "КРАСИВЫЙ". С этой точки зрения термины "ОЧЕНЬ КРАСИВЫЙ", "НЕКРАСИВЫЙ", "ЧЕРЕЗВЫЧАЙНО КРАСИВЫЙ", "ВПОЛНЕ КРАСИВЫЙ" и т.п. — названия нечетких множеств, образованных путем действия модификаторов "ОЧЕНЬ, НЕ, ЧЕРЕЗВЫЧАЙНО, ВПОЛНЕ" и т.п. на *нечеткое множество* "КРАСИВЫЙ". В сущности, эти нечеткие *множества* вместе с нечетким множеством

"КРАСИВЫЙ" играют роль значений лингвистической переменной "ВНЕШНОСТЬ".

Важный аспект понятия *лингвистической переменной* состоит в том, что эта *переменная* более высокого порядка, чем нечеткая *переменная*, в том смысле, что значениями *лингвистической переменной* являются нечеткие переменные. Например, значениями *лингвистической переменной* "ВОЗРАСТ" могут быть: "МОЛОДОЙ, НЕМОЛОДОЙ, СТАРЫЙ, ОЧЕНЬ СТАРЫЙ, НЕ МОЛОДОЙ И НЕ СТАРЫЙ" и т.п. Каждое из этих значений является названием *нечеткой переменной*. Если $\tilde{x}$ — название нечеткой переменной, то ограничение, обусловленное этим названием, можно интерпретировать как смысл *нечеткой переменной* $\tilde{x}$.

Другой важный аспект понятия *лингвистической переменной* состоит в том, что *лингвистической переменной* присущи два правила:

1. Синтаксическое, которое может быть задано в форме грамматики, порождающей название значений переменной;
2. Семантическое, которое определяет алгоритмическую процедуру для вычисления смысла каждого значения.

Определение.

Лингвистическая переменная характеризуется набором свойств $(X, T(X), U, G, M)$, в котором:

X — название переменной;

$T(X)$ обозначает *терм-множество* переменной X, т.е. множество названий лингвистических значений переменной X, причем каждое из таких значений является *нечеткой переменной* $\tilde{x}$ со значениями из универсального *множества* U с базовой переменной u ;

G — синтаксическое правило, порождающее названия $\tilde{x}$ значений переменной X ;

M — семантическое правило, которое ставит в соответствие каждой *нечеткой переменной* $\tilde{x}$ ее смысл $M(\tilde{x})$, т.е. нечеткое *подмножество* $M(\tilde{x})$ *универсального множества* U.

Конкретное название $\tilde{x}$, порожденное синтаксическим правилом G, называется термом. *Терм*, который состоит из одного слова или из нескольких слов, всегда фигурирующих вместе друг с другом, называется атомарным термом. *Терм*, который состоит из более чем одного атомарного терма, называется *составным термом*.

Пример 4.2.1. Рассмотрим *лингвистическую переменную* с именем $X =$ "ТЕМПЕРАТУРА В КОМНАТЕ". Тогда оставшуюся четверку $\langle T, U, G, M \rangle$, можно определить так:

1. Универсальное множество U=[5,35].
2. *Терм-множество* T={"ХОЛОДНО", "КОМФОРТНО", "ЖАРКО"} с такими функциями принадлежностями:

$$\mu''_{\text{холодно}''}(u) = \frac{1}{1 + \left(\frac{u-10}{7}\right)^{12}},$$

$$\mu''_{\text{комфортно}'}(u) = \frac{1}{1 + \left(\frac{u-20}{3}\right)^{6}},$$

$$\mu''_{\text{жарко}''}(u) = \frac{1}{1 + \left(\frac{u-30}{6}\right)^{10}};$$

3. Синтаксическое правило G, порождающее новые термы с использованием *квантификаторов* "и", "или", "не", "очень", "более-менее" и других.
4. M будет являться процедурой, ставящей каждому новому терму в соответствие нечеткое множество из X по правилам: если термы A и B имели функции принадлежности $\mu_A(u)$ и $\mu_B(u)$ соответственно, то новые термы будут иметь следующие *функции принадлежности*, заданные в таблице.

Квантификатор	***Функция принадлежности*** ($u \in U$)
не t	$1 - \mu_t(u)$
очень t	$(\mu_t(u))^2$
более-менее t	$\sqrt{\mu_t(u)}$
A или B	$\max(\mu_A(x), \mu_B(x))$
A и B	$\min(\mu_A(x), \mu_B(x))$

Графики *функций принадлежности* термов "холодно", "не очень холодно" и т.п. к лингвистической переменной "температура в комнате" показаны на рис. 4.2.1:

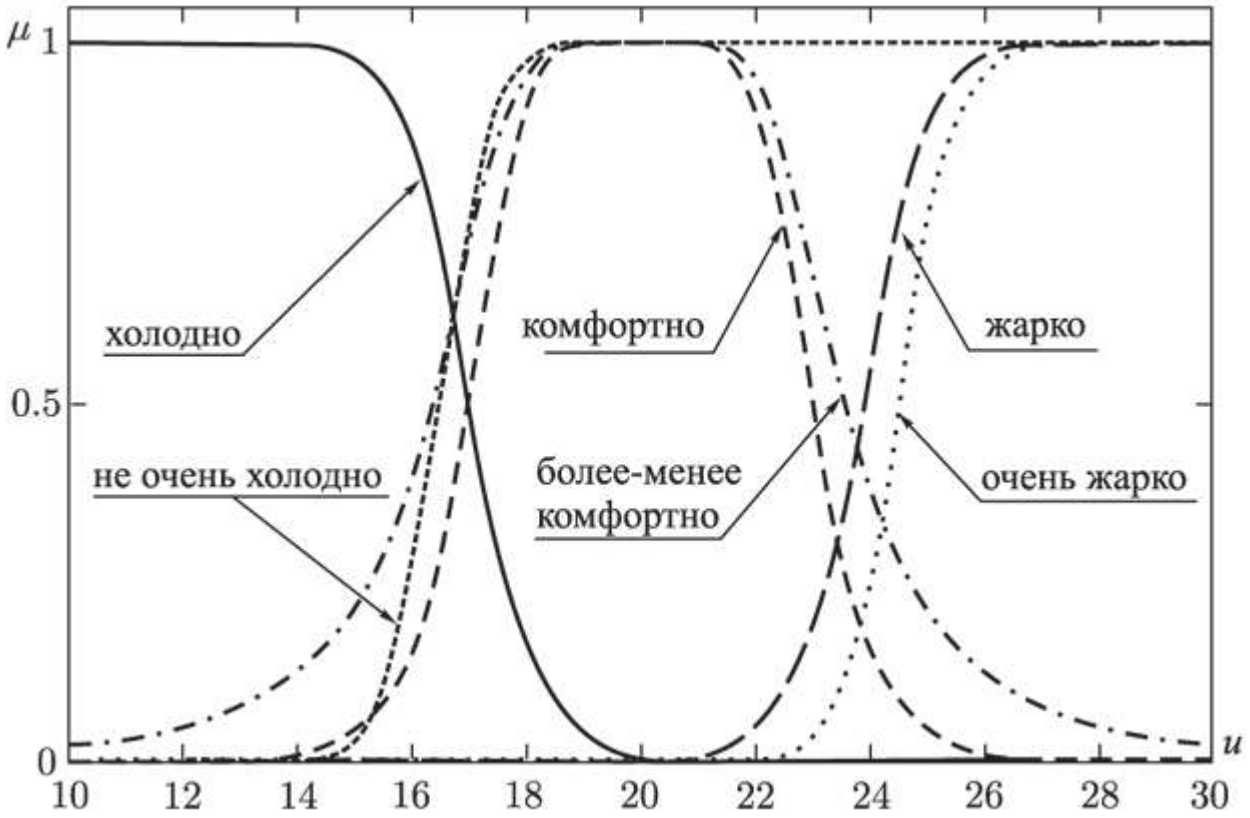

Рисунок 4.2.1 - Графики функций принадлежности термов "холодно", "не очень холодно"

В рассмотренном примере *терм-множество* состояло лишь из небольшого числа термов, так что целесообразно было просто перечислить элементы *терм-множества* $T(X)$ и установить прямое соответствие между каждым элементом и его смыслом. В более общем случае, число элементов в $T(X)$ может быть бесконечным, и, тогда как для порождения элементов *множества* $T(X)$, так и для вычисления их смысла необходимо применять *алгоритм*, а не просто процедуру перечисления.

Будем говорить, что *лингвистическая переменная* X **структурирована**, если ее *терм-множество* $T(X)$ и функцию M, которая ставит в соответствие каждому элементу *терм-множества* его смысл, можно задать алгоритмически.

Пример 4.2.2. В качестве очень простой иллюстрации той роли, которую играют синтаксическое и семантическое правила в случае структурированной *лингвистической переменной*, рассмотрим переменную РОСТ, *терм-множество* которой можно записать в виде:

T(РОСТ) = {ВЫСОКИЙ, ОЧЕНЬ ВЫСОКИЙ, ОЧЕНЬ-ОЧЕНЬ ВЫСОКИЙ, ...}.

M(ВЫСОКИЙ)= $\begin{cases} \left(1+\left(\frac{u-60}{5}\right)^{-2}\right)^{-1}, & \text{если } u \geqslant 60, \\ 0, & \text{в противном случае.} \end{cases}$

M(ОЧЕНЬ ВЫСОКИЙ)=(M(ВЫСОКИЙ))2, и т.д.

Лингвистическую переменную будем называть **булевой**, если ее термы являются булевыми комбинациями переменных вида X_p и hX, где h — лингвистическая неопределенность, X_p — атомарный *терм*.

Пример 4.2.3. Пусть "ВОЗРАСТ" — булева *лингвистическая переменная* с *терм*-множеством вида

T(ВОЗРАСТ) = {МОЛОДОЙ, НЕМОЛОДОЙ, СТАРЫЙ, НЕСТАРЫЙ, ОЧЕНЬ МОЛОДОЙ, НЕ МОЛОДОЙ И НЕ СТАРЫЙ, МОЛОДОЙ ИЛИ НЕ ОЧЕНЬ СТАРЫЙ, ...}.

В этом примере имеются два атомарных терма — МОЛОДОЙ и СТАРЫЙ и одна неопределенность — ОЧЕНЬ.

Если отождествлять союз И с *операцией пересечения* нечетких множеств, ИЛИ — с операцией объединения нечетких множеств, *отрицание* НЕ — с операцией взятия дополнения и модификатор ОЧЕНЬ — с операцией концентрирования, то данная *переменная* будет полностью структурирована.

Лингвистические переменные истинности

В каждодневных разговорах мы часто характеризуем степень истинности утверждения посредством таких выражений, как "очень верно", "совершенно верно", "более или менее верно", "ложно", "абсолютно ложно" и т.д. Сходство между этими выражениями и значениями *лингвистической переменной* наводит на мысль о том, что в ситуациях, когда истинность или ложность

утверждения определены недостаточно четко, может оказаться целесообразным трактовать ИСТИННОСТЬ как *лингвистическую переменную*, для которой ИСТИННО и ЛОЖНО — лишь два атомарных терма в *терм*-множестве этой переменной. Такую переменную будем называть ***лингвистической переменной истинности***, а ее значения — **лингвистическими значениями истинности.**

Трактовка истинности как *лингвистической переменной* приводит к *нечеткой лингвистической логике*, которая совершенно отлична от обычной двузначной или даже многозначной логики. Такая *нечеткая логика* является основой того, что можно было бы назвать приближенными рассуждениями, т.е. видом рассуждений, в которых значения истинности и правила их вывода являются нечеткими, а не точными. Приближенные рассуждения во многом сродни тем, которыми пользуются люди в некорректно определенных или не поддающихся количественному описанию ситуациях. В самом деле, вполне возможно, что многие, если не большинство человеческих рассуждений по своей природе приближенны, а не точны.

В дальнейшем будем пользоваться термином "нечеткое *высказывание*" для обозначения утверждения вида " u есть A ", где u — название предмета, а A — название нечеткого подмножества универсального *множества* U, например, "Джон — молодой", "Х — малый", "яблоко — красное" и т.п. Если интерпретировать A как нечеткий *предикат*, то утверждение " u есть A " можно перефразировать как " u имеет свойство A ".

Будем полагать, что высказыванию типа " u есть A " соответствуют два нечетких подмножества:

1. $M(A)$ — смысл A, т.е. нечеткое подмножество с названием A *универсального множества* U ;
2. Значение истинности утверждения " u есть A ", которое будем обозначать $v(A)$ и определять как возможно нечеткое подмножество *универсального множества* значений истинности V. Будем предполагать, что $V = [0, 1]$.

Значение истинности, являющееся числом в $[0, 1]$, например, $v(A) = 0,8$, будем называть **числовым** значением истинности. Числовые значения истинности играют роль значений базовой переменной для **лингвистической переменной ИСТИННОСТЬ**. Лингвистические значения переменной ИСТИННОСТЬ будем называть **лингвистическими значениями истинности**. Более

точно будем предполагать, что ИСТИННОСТЬ — название булевой лингвистической переменной, для которой атомарным является *терм* ИСТИННЫЙ, а *терм* ЛОЖНЫЙ определяется не как *отрицание* терма ИСТИННЫЙ, а как его зеркальное *отображение* относительно точки $0{,}5$. Далее мы покажем, что такое *определение* значения ЛОЖНЫЙ является следствием его определения как значения истинности высказывания "u есть не A" при предположении, что *значение* истинности высказывания "u есть A" является ИСТИННЫМ.

Предполагается, что смысл первичного терма ИСТИННЫЙ является нечетким подмножеством интервала $V = [0, 1]$ с *функцией принадлежности* типа

$$\mu_{\text{ИСТИННЫЙ}}(u) = \begin{cases} 0, & \text{если } 0 \leqslant u \leqslant a; \\ 2\left(\frac{u-a}{1-a}\right)^2, & \text{если } a \leqslant u \leqslant \frac{1+a}{2}; \\ 1 - 2\left(\frac{u-a}{1-a}\right)^2, & \text{если } \frac{1+a}{2} \leqslant u \leqslant 1, \end{cases}$$

показанной на рис. 4.2.2.

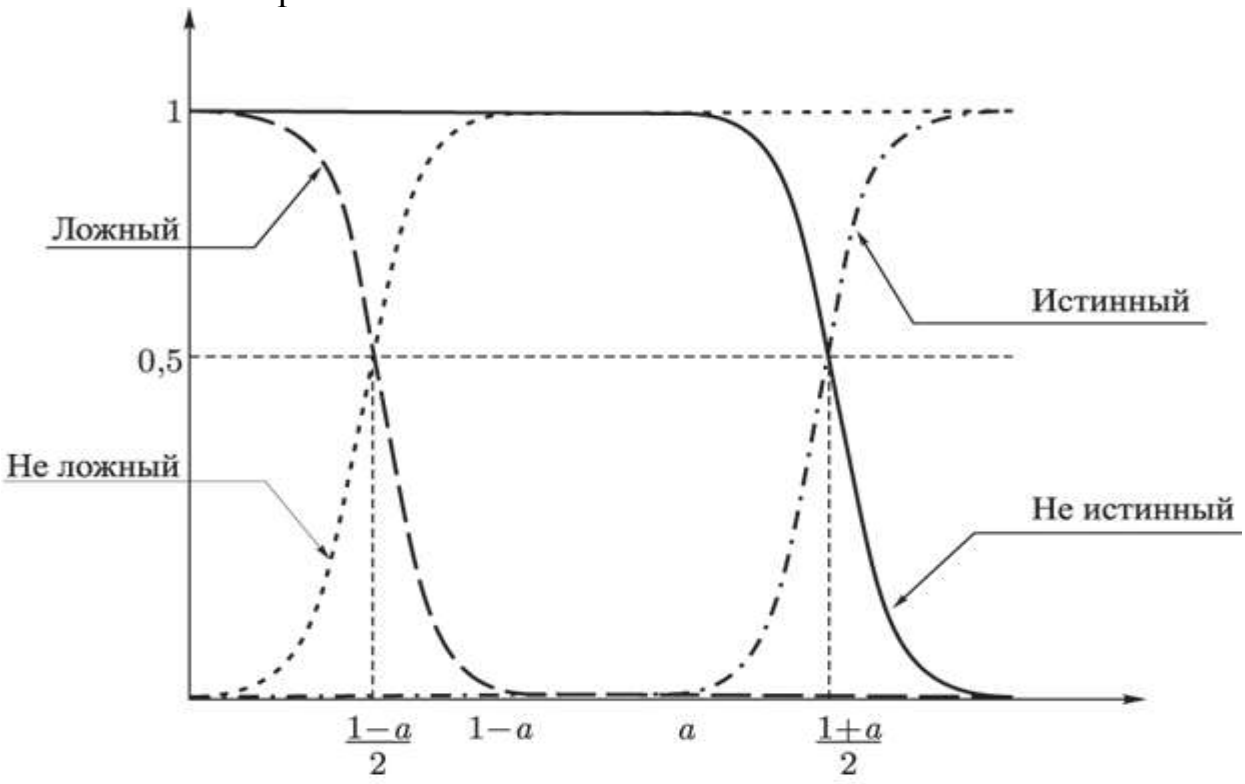

Рисунок 4.2.2 - Функция принадлежности терма ИСТИННЫЙ

Здесь точка $u = \frac{1+a}{2}$ является точкой перехода. Соответственно, для терма ЛОЖНЫЙ имеем

$$\mu_{\text{ЛОЖНЫЙ}}(u) = \mu_{\text{ИСТИННЫЙ}}(1 - u).$$

Логические связки в нечеткой лингвистической логике

Чтобы заложить основу для *нечеткой лингвистической логики*, необходимо расширить содержание таких логических операций, как *отрицание, дизъюнкция, конъюнкция* и *импликация* применительно к высказываниям, которые имеют не числовые, а лингвистические значения истинности.

При рассмотрении этой проблемы полезно иметь в виду, что если A — нечеткое *подмножество универсального множества* U и $u \in U$, то два следующих утверждения эквивалентны:

1. Степень принадлежности элемента u нечеткому множеству A есть $\mu_A(u)$.
2. Значение истинности нечеткого предиката " u есть A " также равно $\mu_A(u)$.

Таким образом, вопрос "Что является значением истинности высказывания " u есть A " И " u есть B ", если заданы лингвистические значения истинности высказываний " u есть A " и " u есть B "? аналогичен вопросу "Какова степень принадлежности элемента u множеству $A \cap B$, если заданы степени принадлежности элемента u множествам A и B?".

В частности, если $v(A)$ — точка в $V = [0,1]$, представляющая *значение* истинности высказывания " u есть A " (или просто A), где u — элемент *универсального множества* U, то *значение* истинности высказывания " u есть не A " (или A) определяется выражением

$$v(\neg A) = 1 - v(A).$$

Предположим теперь, что $v(A)$ — не точка в $[0,1]$, а нечеткое *подмножество* интервала $[0,1]$, представленное в виде

$$v(A) = f(x), \qquad f : [0,1] \to [0,1].$$

Тогда получим $v(\neg A) = f(1-x)$.

В частности, если *значение* истинности A есть ИСТИННО, т.е. $v(A) =$ ИСТИННО, то *значение* истинности ЛОЖНО является значением истинности для высказывания $\neg A$.

Замечание. Следует отметить, что если ИСТИННЫЙ $= f(x)$, то *функция* $1 - f(x)$ будет интерпретироваться термом НЕ ИСТИННЫЙ, а *функция* $f(1-x)$ — термом ЛОЖНЫЙ, что в принципе не одно и то же (рис. 4.2.3).

То же самое относится к лингвистическим неопределенностям. Например, если ИСТИННЫЙ $= f(x)$, то *значение* терма ОЧЕНЬ ИСТИННЫЙ равно $f^2(x)$ (рис. 4.2.3).

С другой стороны, если *значение* истинности высказывания A есть $f(x)$, то *функция* $f(x^2)$ будет выражать *значение* истинности высказывания "очень A ".

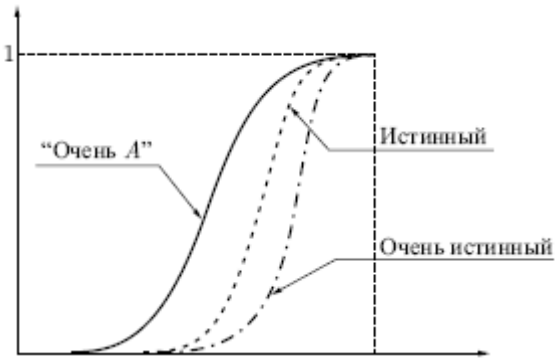

Рисунок 4.2.3 - График функции

Перейдем к бинарным связкам. Пусть $v(A)$ и $v(B)$ — лингвистические значения истинности высказываний A и B соответственно. В случае, когда $v(A)$ и $v(B)$ — точечные оценки, имеем:

$$v(A) \wedge v(B) = v(A \text{ И } B), v(A) \vee v(B) = v(A \text{ ИЛИ } B),$$

где *операции* $\wedge$ и $\vee$ сводятся к операциям нечеткой логики.

Если $v(A)$ и $v(B)$ — лингвистические значения истинности, заданные функциями

$$v(A) = f(x), \quad v(B) = g(x), \qquad f, g : [0,1] \to [0,1],$$

то, согласно *принципу обобщения*, *конъюнкция* и *дизъюнкция* будут вычисляться по следующим формулам:

$$v(A) \wedge v(B) \quad \Leftrightarrow \quad \sup_{z=x \wedge y} \left(\mu_A(x) \wedge \mu_B(y)\right),$$

$$v(A) \vee v(B) \quad \Leftrightarrow \quad \sup_{z=x \vee y} \left(\mu_A(x) \wedge \mu_B(y)\right).$$

Замечание. Важно четко понимать разницу между связкой И (ИЛИ) в терме, например, ИСТИННЫЙ И (ИЛИ) НЕ ИСТИННЫЙ и символом $\wedge$ ($\vee$) в высказывании ИСТИННЫЙ $\wedge$ ($\vee$) НЕ ИСТИННЫЙ. В первом случае нас интересует смысл терма

ИСТИННЫЙ И (ИЛИ) НЕ ИСТИННЫЙ, и связка И (ИЛИ) определяется отношением

M (ИСТИННЫЙ И (ИЛИ) НЕ ИСТИННЫЙ)=

= M (ИСТИННЫЙ) $\cap$ ($\cup$) M (НЕ ИСТИННЫЙ),

где $M(A)$ — смысл терма A. Напротив, в случае терма ИСТИННЫЙ $\wedge$ ($\vee$) НЕ ИСТИННЫЙ нас в основном интересует *значение* истинности высказывания ИСТИННЫЙ $[\wedge\ (\vee)]$ НЕ ИСТИННЫЙ, которое получается из равенства

v (А И (ИЛИ) В) = $v(A) \wedge\ (\vee) v(B)$.

Значения истинности НЕИЗВЕСТНО и НЕ ОПРЕДЕЛЕНО

Среди возможных значений истинности *лингвистической переменной* ИСТИННОСТЬ два значения привлекают особое внимание, а именно *пустое множество* $\varnothing$ и единичный *интервал* $\mathfrak{F} = [0,1]$, которые соответствуют наименьшему и наибольшему элементам (по отношению включения) решетки нечетких подмножеств интервала $[0,1]$. Важность именно этих значений истинности обусловлена тем, что их можно интерпретировать как значения истинности НЕ ОПРЕДЕЛЕНО и НЕИЗВЕСТНО соответственно. Важно четко понимать разницу между 0 и $\varnothing$. Когда мы говорим, что степень принадлежности точки u множеству A есть $\varnothing$, мы имеем в виду, что *функция принадлежности* $\mu_A : U \to [0,1]$ не определена в точке u. Предположим, например, что U — множество действительных чисел, а μ_A — *функция*, определенная на множестве целых чисел, причем $\mu_A(u) = 1$, если u четное, и $\mu_A(u) = 0$, если u нечетное.

Тогда степень принадлежности числа $u = 1,5$ множеству A есть $\varnothing$, а не 0.

С другой стороны, если бы μ_A была определена на множестве действительных чисел и $\mu_A(u) = 1$ тогда и только тогда, если u — четное число, то степень принадлежности числа 1,5 множеству A была бы равна 0.

Понятие значения истинности НЕИЗВЕСТНО в сочетании с принципом обобщения помогает уяснить некоторые понятия и соотношения обычных двухзначных и трехзначных логик. Эти логики можно рассматривать как вырожденные случаи *нечеткой логики*, в которой значением истинности НЕИЗВЕСТНО является весь единичный *интервал*, а не множество $\{0, 1\}$.

Лингвистические вероятности

В классическом теоретико-вероятностном подходе событие A определяется как элемент σ-поля A подмножеств пространства элементарных событий Ω. Так, если P — нормированная мера над измеримым пространством (Ω, A), то вероятность $P(A)$ события A определяется как мера множества A и является числом из интервала [0, 1]. Существует много реальных проблем, в которых нарушается одно или больше предположений, неявно присутствующих в приведенном выше определении. Во-первых, часто бывает плохо определено само событие A, как, например, в вопросе «Какова вероятность того, что завтра будет **теплый день?»** В этом случае событие **теплый день** — нечеткое событие в том смысле, что не существует резкой грани между его появлением и непоявлением. Такое событие можно охарактеризовать как нечеткое подмножество A пространства элементарных событий Ω с измеримой функцией принадлежности μ_A.

Во-вторых, даже если A — вполне определенное обычное (не нечеткое) событие, его вероятность $P(A)$ может быть определена плохо. Например, на вопрос «Какова вероятность того, что через месяц средняя цена на акции фирмы «Доу-Джонс» будет выше?» было бы, по-видимому, неразумно однозначно отвечать числом, например 0.7. В этом случае неопределенный ответ типа «вполне вероятно» более соответствовал бы нашему нечеткому пониманию динамики цен на акции и, следовательно, более реалистично, хотя и менее точно, характеризовал бы рассматриваемую вероятность.

Ограничения, обусловленные предположением о том, что A — вполне определенное событие, можно устранить по крайней мере частично, если допустить, что A может быть нечетким событием. Другой и, возможно, более важный шаг, который можно предпринять с целью сделать теорию вероятностей применимой к плохо определенным ситуациям, состоит в допущении того, что вероятность P может быть лингвистической переменной в смысле определения. Ниже мы изложим в общих чертах, каким образом это можно сделать, и

исследуем некоторые элементарные следствия, вытекающие из того, что вероятность P — лингвистическая переменная. Чтобы упростить изложение, будем рассматривать переменную X с конечным универсальным множеством $U = u_1 + u_2 + \ldots + u_n$.

Кроме того, будем предполагать, что ограничение, обусловленное X, совпадает с U. Иными словами, любая точка в U может быть выбрана в качестве значения переменной X.

Каждому элементу $u_i, i = 1, \ldots, n,$ мы поставим в соответствие лингвистическую вероятность P_1, которая является булевой лингвистической переменной в смысле определения 5.9; $p_i, 0 \le p_i \le 1$, — базовая переменная для P_1. Для определенности предположим, что универсальное множество V, соответствующее P_1, представляет собой либо единичный интервал [0, 1], либо конечное множество

$$V = 0 + 0.1 + \ldots + 0.9 + 1 . \qquad (4.2.1)$$

Будем употреблять P в качестве общего названия переменных P_1; типичное терм-множество для P имеет такой вид:

T(p) = **правдоподобно + неправдоподобно + неправдоподобно + + очень правдоподобно + более или менее правдоподобно + очень не правдоподобно + ... + вероятно + невероятно + очень вероятно + ни очень вероятно + ни очень невероятно, ни очень невероятно + ... + близко к 0 + близко к 0.1 + ... + близко к 1 + ... + очень близко к 0 + очень близко к 0.1 + ...,** (4.2.2)

где термы **правдоподобно, вероятно** и **близко к** играют роль первичных термов.

Будем считать, что функция принадлежности нечеткого множества **правдоподобно** имеет тот же вид, что и функция принадлежности нечеткого множества **истинно**, а функции принадлежности нечетких множеств **не правдоподобно и неправдоподобно** определим следующим образом:

$$\mu_{\text{не правдоподобно}}(p) = 1 - \mu_{\text{правдоподобно}}(p), \qquad (4.2.3)$$

$$\mu_{\text{неправдоподобно}}(p) = \mu_{\text{правдоподобно}}(1-p), \qquad (4.2.4)$$

где p — общее название переменных p_i.

Пример 4.2.4. Графический пример смысла, приписываемого термам **правдоподобно, не правдоподобно, очень правдоподобно** и **неправдоподобно**, приведен на рис. 4.2.4.

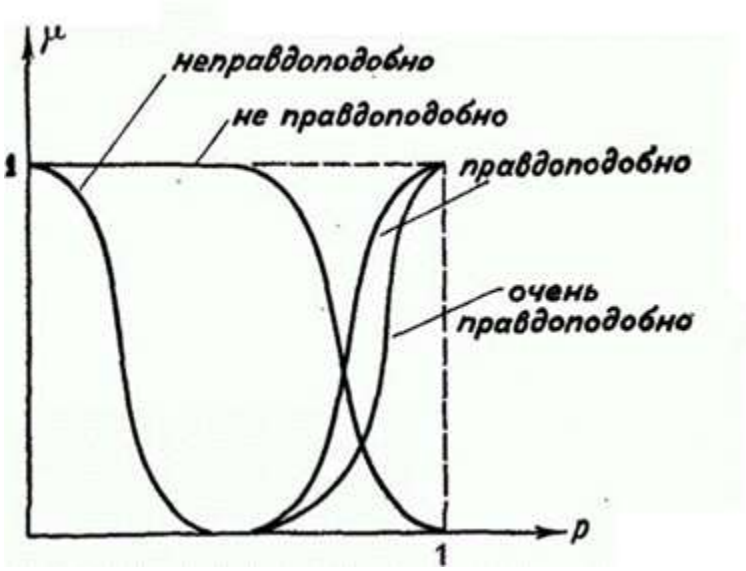

Рисунок 4.2.4 - Функции совместимости значений правдоподобно, неправдоподобно, неправдоподобно и очень правдоподобно

Численное выражение первичного терма **правдоподобно** имеет вид

правдоподобно = 0.5/0.6 + 0.7/0.7 + 0.9/0.8 + 1/0.9 + 1/1, (4.2.5)

откуда

неправдоподобно = 1/(0 + 0.1 + 0.2 + 0.3 + 0.4 + 0.5) + 0.5/0.6 + 0.3/0.7 + 0.1/0.8, (4.2.6)

неправдоподобно = 1/0 + 1/0.1 + 0.9/0.2 + 0.7/0.3 + 0.5/0.4, (4.2.7)

очень правдоподобно = 0.25/0.6 + 0.49/0.7 + 0.81/0.8 + + 1/0.9 + 1/1. (4.2.8)

Будем предполагать, что терм **вероятно** более или менее синонимичен терму **правдоподобно**. Терм **близко к** α, где α — число из интервала [0, 1], будем записывать сокращенно

как a или $\ll a \gg$), считая, что a — «наилучший пример» нечеткого множества «a». Имея это в виду, можно записать

$$\textbf{правдоподобно} \triangleq \textbf{близко к}\ 1 \triangleq \ll 1 \gg, \tag{4.2.9}$$

$$\textbf{маловероятно} \triangleq \textbf{близко к}\ 0 \triangleq 0 \ll 0 \gg, \tag{4.2.10}$$

$$\textbf{близко к}\ 0.8 \triangleq \ll 0.8 \gg = 0.6/0.7 + 1/0.8 + 0.6/0.9, \tag{4.2.11}$$

отсюда следует, что

$$\textbf{очень близко к}\ 0.8 = \textbf{очень} \ll 0.8 \gg = (\ll 0.8 \gg)^2 =$$
$$= 0.36/0.7 + 1/0.8 + 0.36/0.9$$

Терм в $T(\mathrm{P})$ будем обозначать через P_j или P_{ji} в случае, когда двойной индекс необходим. Так, если $P_4 \triangleq$ **очень правдоподобно**, то P_{43} обозначает, что терм **очень правдоподобно** назначен в качестве значения лингвистической переменной P_3.

Введем n-арную лингвистическую переменную $(P_1, \ldots, P_n)$, которая представляет собой список значений лингвистических вероятностей, соответствующий X. Саму переменную X будем называть при этом лингвистической случайной переменной. По аналогии с распределениями значений истинности совокупность списков значений лингвистических вероятностей будем называть распределением лингвистических вероятностей.

Назначение переменной P_i значения P_j можно выразить равенством

$$\mathrm{P}_i = P_j, \tag{4.2.12}$$

где P_i используется как общее название нечетких переменных, составляющих P_i. Например, можно писать

$$\mathrm{P}_3 = \mathrm{P}_4 = \textit{очень правдоподобно}, \tag{4.2.13}$$

где **очень правдоподобно** можно отождествить с P_{43} (т.е. со значением P_4, назначенным переменной P_3).

Важное свойство лингвистических вероятностей $P_1, \ldots, P_n$ состоит в том, что они являются - взаимодействующими. Взаимодействие между P_i есть следствие ограничения ($+^{\triangleq}$ арифметическая сумма)

$$p_1 + p_2 + \ldots + p_n = 1, \tag{4.2.14}$$

в котором p_i — базовые переменные (т.е. числовые вероятности), связанные с P_i .

Более конкретно, пусть $R(p_1 + \ldots + p_n = 1)$ обозначает нечетко n-арное отношение в $[0,1] \times \ldots \times [0,1]$, представляющее (4.2.14). Пусть, кроме того, $R(P_i)$ обозначает ограничение на значения переменной P_i. Тогда ограничение, обусловленное n-арной нечеткой переменной $(P_1, \ldots, P_n)$, можно записать в виде

$$R(P_1, \ldots, P_n) = R(P_1) \times \ldots \times R(P_n) \cap R(p_1 + \ldots + p_n = 1). \tag{4.2.15}$$

Откуда следует, что без ограничения (4.2.14) нечеткие переменные $P_1, \ldots, P_n$ были бы невзаимодействующими.

Пример 4.2.5. Допустим, что

$$P_1 = \textit{правдоподобно} = 0.5/0.8 + 0.8/0.9 + 1/1, \tag{4.2.16}$$

$$P_2 = \textit{неправдоподобно} = 1/0 + 0.8/0.1 + 0.5/0.2. \tag{4.2.17}$$

Тогда

$$\begin{aligned} R(P_1) \times R(P_2) &= \textit{правдоподобно} \times \textit{неправдоподобно} = \\ &= (0.5/0.8 + 0.8/0.9 + 1/1) \times \\ &\times (1/0 + 0.8/0.1 + 0.5/0.2) = \\ &= 0.5/(0.8,\ 0) + 0.8/(0.9,\ 0) + 1/(1,\ 0) + \\ &+ 0.5/(0.8,\ 0.1) + 0.8/(0.9,\ 0.1) + 0.8/(1,\ 0.1) + \\ &+ 0.5/(0.8,\ 0.2) + 0.5/(0.9,\ 0.2) + 0.5/(1,\ 0.2). \end{aligned} \tag{4.2.18}$$

Что касается отношения $R(p_1 + \ldots + p_n = 1)$, то его можно выразить в виде

$$R(p_1 + p_2 = 1) = \sum_k 1/(k,\ 1-k),\ k = 0,\ 0.1,\ \ldots,\ 0.9,\ 1, \qquad (4.2.19)$$

и, образуя пересечение (4.2.18) и (4.2.19), получаем

$$R(P_1,\ P_2) = 1/(1,0) + 0.8/(0.9,0.1) + 0.5/(0.8,\ 0.2), \qquad (4.2.20)$$

т.е. выражение для ограничения, обусловленного составной переменной $(P_1,\ P_2)$. Ясно, что $R(P_1,\ P_2)$ состоит из тех членов выражения для $R(P_1) \times R(P_2)$, которые удовлетворяют ограничению (4.2.14).

Замечание. Следует отметить, что ограничение $R(P_1,\ P_2)$ вида (4.2.20) является нормальным ограничением. Это справедливо также и в более общем случае, когда P_1 имеют вид

$$P_i = \ll q_i \gg,\ i = 1,\ \ldots,\ n, \qquad (4.2.21)$$

и $q_1 + \ldots + q_n = 1.$ Заметим, что в примере 2.2 мы имеем

$$P_1 = \ll 1 \gg, \qquad (4.2.22)$$

$$P_2 = \ll 0 \gg, \qquad (4.2.23)$$

$$1 + 0 = 1. \qquad (4.2.24)$$

Во многих приложениях теории вероятностей, например, при вычислении средних значений, дисперсий и т.п., часто встречаются линейные комбинации вида ($+^{\triangleq}$ арифметическая сумма)

$$z = a_1 p_1 + \ldots + a_n p_n, \qquad (4.2.25)$$

где a_i — действительные числа, а p_i — значения вероятностей из интервала [0, 1]. Если p_i — числа из интервала [0, 1], то

вычисление значения комбинации z при заданных a_i и p_i не представляет труда. Однако оно становится нетривиальным, когда рассматриваемые вероятности являются лингвистическими по своей природе, т.е. когда

$$Z = a_1 P_1 + \ldots + a_n P_n, \qquad (4.2.26)$$

где P_i — такие лингвистические значения вероятностей, как **правдоподобно, неправдоподобно, очень правдоподобно, близко** к α и т.п. Соответственно Z — недействительное число, как в (4.2.25), а нечеткое подмножество действительной оси $W \triangleq (-\infty, \infty)$, причем функция принадлежности подмножества Z зависит от функций принадлежности P_i.

В предположении, что нечеткие переменные $P_1, \ldots, P_n$ - невзаимодействующие (не считая ограничения (4.2.14)), ограничение, обусловленное набором $(P_1, \ldots, P_n)$, принимает вид (см. (4.2.15))

$$R(P_1, \ldots, P_n) = R(P_1) \times \ldots \times R(P_n) \cap R(p_1 + \ldots + p_n = 1). \qquad (4.2.27)$$

Пусть $\mu(p_1, \ldots, p_n)$ - функция принадлежности ограничения $R(P_1, \ldots, P_n)$ и пусть $\mu_i(p_i)$ - функция принадлежности ограничения $R(P_i), i = 1, \ldots, n$. Тогда, применяя принцип обобщения к (4.2.25),

Можно выразить Z в виде нечеткого множества

($+ \triangleq$ арифметическая сумма)

$$Z = \int_W \mu(p_1, \ldots, p_n) / (a_1 p_1 + \ldots + a_n p_n), \qquad (4.2.28)$$

которое с учетом (4.2.27) можно записать как

$$Z = \int_W \mu_1(p_1) \wedge \ldots \wedge \mu_n(p_n) / (a_1 p_1 + \ldots + a_n p_n), \qquad (4.2.29)$$

понимая при этом, что p_i в (4.2.29) удовлетворяют ограничению

$$p_1 + \ldots + p_n = 1. \qquad (4.2.30)$$

Таким образом мы можем представить линейную комбинацию значений **лингвистических вероятностей** нечетким подмножеством действительной оси.

Выражение для Z можно записать другим более удобным для вычислений способом. Так, пусть $\mu(z)$ обозначает функцию принадлежности множества Z, причем $z \in W$. Тогда из (4.2.29) следует, что

$$\mu(z) = \bigvee_{p_1, \ldots, p_n} \mu_1(p_1) \wedge \ldots \wedge \mu_n(p_n) \qquad (4.2.31)$$

при ограничениях

$$z = a_1 p_1 + \ldots + a_n p_n, \qquad (4.2.32)$$

$$p_1 + \ldots + p_n = 1. \qquad (4.2.33)$$

В этом случае вычисление Z сводится к решению задачи нелинейного программирования с линейными ограничениями. Более точно эту задачу можно сформулировать следующим образом: максимизировать z при следующих ограничениях ($+^{\triangle}$ арифметическая сумма):

$$\begin{aligned} &\mu_1(p_1) \geq z, \\ &\ldots\ldots\ldots\ldots \\ &\mu_n(p_n) \geq z, \\ &z = a_1 p_1 + \ldots + a_n p_n, \\ &p_1 + \ldots + p_n = 1. \end{aligned} \qquad (4.2.34)$$

Пример 4.2.6. Проиллюстрируем изложенное следующим очень простым примером. Предположим, что

$$R_1 = \textit{правдоподобно}, \tag{4.2.35}$$

$$R_2 = \textit{неправдоподобно}, \tag{4.2.36}$$

$$\textit{правдоподобно} = \int_0^1 \mu_{\textit{правдоподобно}}(p)/p \tag{4.2.37}$$

где

$$\textit{неправдоподобно} = \neg\textit{правдоподобно}. \tag{4.2.38}$$

тогда [см. (4.2.4)]

$$\mu_{\textit{неправдоподобно}}(p) = \mu_{\textit{правдоподобно}}(1-p), \quad 0 \le p \le 1. \tag{4.2.39}$$

Предположим, что мы хотим вычислить математическое ожидание ($+^{\wedge}$ арифметическая сумма) вида

$$Z = a_1\ \textit{правдоподобно} + a_2\ \textit{неправдоподобно}. \tag{4.2.40}$$

используя (4.2.22), получаем

$$\mu(z) = \bigvee_{p_1 p_2} \mu_{\textit{правдоподобно}}(p_1) \wedge \mu_{\textit{неправдоподобно}}(p_2) \tag{4.2.41}$$

при ограничениях

$$\begin{aligned} z &= a_1 p_1 + a_2 p_2, \\ p_1 &+ p_2 = 1. \end{aligned} \tag{4.2.42}$$

Теперь если $p_1 + p_2 = 1$, имеем

$$\mu_{\textit{правдоподобно}}(p_1) = \mu_{\textit{неправдоподобно}}(p_2), \tag{2.2.43}$$

и, следовательно,

$$\mu(z) = \mu_{правдоподобно}(p_1),$$
$$z = a_1 p_1 + a_2(1 - p_1), \tag{4.2.44}$$

или, в более явной форме,

$$\mu(z) = \mu_{правдоподобно}\left(\frac{z - a_2}{a_1 - a_3}\right). \tag{4.2.45}$$

Из этого результата следует, что нечеткость нашего знания вероятности p_1 приводит к соответствующей нечеткости математического ожидания (рис. 4.2.5) $z = a_1 p_1 + a_2 p_2$.

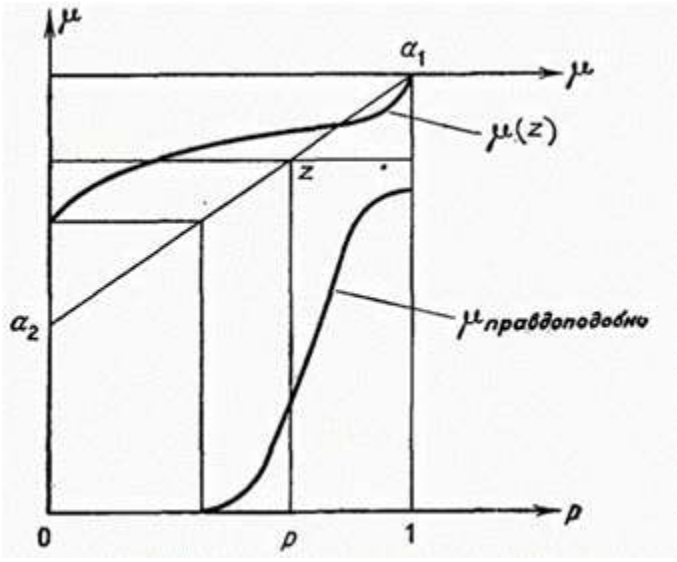

Рисунок 4.2.5 - Вычисление лингвистического значения переменной $a_1 / p_1 + a_2 / p_2$

Если предположить, что универсальное множество значений вероятности есть, то есть выражение для Z можно получить непосредственно. В качестве иллюстрации предположим, что

$$B_1 = \ll 0.3 \gg = 0.8/0.2 + 1/0.3 + 0.6/0.4, \tag{4.2.46}$$

$$B_2 = \ll 0.7 \gg = 0.8/0.6 + 1/0.7 + 0.6/0.8 \tag{4.2.47}$$

($\oplus \triangleq$ арифметическая сумма)

$$Z = a_1 R_1 \oplus a_2 R_2, \tag{4.2.48}$$

где символ $\oplus$ используется во избежание путаницы со знаком объединения.

Получаем,

$$\begin{aligned} Z &= a_1(0.8/0.2+1/0.3+0.6/0.4) \oplus a_2(0.8/0.6+1/0.7+0.6/0.8) = \\ &= (0.8/0.2a_1+1/0.3a_1+0.6/0.4a_1) \oplus (0.8/0.6a_2+1/0.7a_2+0.6/0.8a_2). \end{aligned} \tag{4.2.49}$$

Раскрывая скобки в правой части, следует иметь в виду ограничение $p_1 + p_2 = 1$, которое означает, что член вида

$$\mu_1 / p_1 a_1 \oplus \mu_2 / p_2 a_2 \tag{4.2.50}$$

сводится к

$$\mu_1 / p_1 a_1 \oplus \mu_2 / p_2 a_2 = \begin{cases} \mu_1 \wedge \mu_2 / (p_1 a_1 \oplus p_2 a_2), & \text{если } p_1 + p_2 = 1, \\ 0 & \text{в противном случае.} \end{cases} \tag{4.2.51}$$

Таким образом, мы получаем

$$Z = 1/(0.3a_1 \oplus 0.7a_2) + 0.6/(0.2a_1 \oplus 0.8a_2) + 0.6/(0.4a_1 \oplus 0.6a_2), \tag{4.2.52}$$

т.е. выражение для Z как нечеткого подмножества действительной оси $W = (-\infty, \infty)$.

5. Особенности проектирования ИБД

Представление моделей анализа данных в базах данных

Прогресс в исследованиях интеллектуального анализа данных позволил эффективно выполнять несколько операций интеллектуального анализа данных на больших базах данных. Хотя это, безусловно, важный вклад, мы не должны упускать из виду конечную цель интеллектуального анализа данных - это позволить разработчикам приложений баз данных создавать модели интеллектуального анализа данных (например, классификатор дерева решений, модель регрессии, сегментацию)

из своих баз данных, использовать эти модели для выполнения различных предсказуемых и аналитических задач и делиться этими моделями с другими приложениями. Такая интеграция является предварительным условием для успешного выполнения интеллектуального анализа данных в мире базы данных.

Признавая вышеупомянутый факт, очевидно, что ключевым аспектом интеграции с системами баз данных, которые необходимо изучить, является то, как обрабатывать модели интеллектуального анализа данных как объекты первого класса в базах данных. К сожалению, в этом отношении добыча данных по-прежнему остается островом анализа, который плохо интегрирован с системами баз данных. Напомним, что модель интеллектуального анализа данных (например, классификатор) получается посредством применения алгоритма интеллектуального анализа данных в наборе данных обучения. Хотя модель интеллектуального анализа данных может быть получена с использованием приложения SQL, реализующего алгоритм обучения, система управления базами данных полностью не знает семантики моделей интеллектуального анализа, поскольку модели интеллектуального анализа явно не представлены в базе данных. Но, если такое явное представление не включено, возможности системы управления базой данных не могут быть использованы для совместного использования, повторного использования и управления моделями интеллектуального анализа. В частности, даже если были созданы несколько моделей интеллектуального анализа, нет возможности для пользователя или приложения для поиска набора доступных моделей на основе его свойств, указать, что определенную модель следует применять для прогнозирования столбца неизвестных данных. Установить, а затем запросить результат предсказания, например, для сравнения результатов предсказаний с двух моделей.

Чтобы эффективно представлять модели интеллектуального анализа данных в реляционных базах данных, нам необходимо зафиксировать создание моделей интеллектуального анализа данных с использованием произвольных алгоритмов добычи, просмотр таких моделей (рассмотрение их структуры или содержимого) и применение выбранной модели к набору данных для решения аналитических задач, таких как прогнозирование. Кроме того, для набора данных, который является результатом прогнозирования, достаточные метаданные должны быть доступны с полученным набором данных, чтобы инструменты анализа могли интерпретировать свойства прогнозирования, например, его точность.

Реляционные системы баз данных понимают и поддерживают только отношения как объекты первого класса, поэтому, если мы хотим представить модель интеллектуального анализа данных в базах данных, ее следует рассматривать как структуру, похожую на таблицу. Однако, на первый взгляд, модель больше похожа на график со сложной интерпретацией его структуры, например, классификатором дерева решений. Таким образом, попытка представить модель интеллектуального анализа в виде таблицы (или набора строк) выглядит неестественной. К счастью, это не должно быть камнем преткновения. Ключевыми шагами жизненного цикла модели интеллектуального анализа являются создание и заполнение модели с помощью алгоритма на источнике данных обучения и возможность использования модели интеллектуального анализа для прогнозирования значений для наборов данных. Если мы сможем выполнить эти шаги с использованием конструкций SQL, то это обеспечит, возможность использования методов интеллектуального анализа данных без изменения используемой парадигмы программирования.

Темы индивидуальных заданий по проектированию интеллектуальной базы для следующих проблем

1. Бинарное изображение как результат сегментации формы. Фигура, как модель формы, открытое множество, связное множество, область, односвязная и многосвязная области, замкнутая область.
2. Непрерывные и дискретные модели формы. Граничное и скелетное описание фигуры, их достоинства и недостатки. Дискретные модели границы и скелета фигуры. Эквивалентность непрерывной и дискретной моделей формы.
3. Непрерывные границы дискретной фигуры. Связность в дискретном пространстве, дискретная фигура.
4. Поиск и прослеживание границы. Постановка задачи прослеживания границ дискретной сцены.
5. Аппроксимация граничного коридора разделяющей фигурой минимального периметра. Алгоритм вытягивания нити. Аппроксимация границы составными кривыми Безье.
6. Обобщения диаграммы Вороного и триангуляции Делоне: в манхэттенской и в шахматной метриках.
7. Обобщение диаграммы Вороного и триангуляции Делоне: для сайтов-окружностей в метрике Лагерра, для разделённых сайтов-сегментов.
8. Диаграмма Вороного многоугольной фигуры. Связь диаграммы Вороного многоугольника и его скелета, алгоритм построения

скелета многоугольника по его диаграмме Вороного. Обобщённая триангуляция Делоне для коллекции сайтов.

9. Преобразования формы изображений в компьютерной графике и в распознавании образов. Жирная кривая, циркулярная фигура, осевой граф, функция ширины, силуэт.
10. Определение кругов в эго-подграфах графа социальной сети (задача, данные, их загрузка, редакторское расстояние), приложения анализа социальных сетей.
11. Анализ социальных сетей, определение кругов пользователей: динамические графы, приложения анализа социальных сетей, погружение графов в признаковое пространство, сходство вершин, важность вершин, прогнозирование появления рёбер в динамическом графе.
12. Оценка среднего, оценка вероятности, оценка плотности. Весовые схемы: проблема оценки среднего, выбросы, разные целевые функционалы, оценка минимального контраста, среднее по Колмогорову.
13. SMAPE-минимизация, двухэтапные алгоритмы и их настройка, пересчёт вероятности и прямая оценка, введение весовых схем, устойчивость весовых схем, ансамблирование, непараметрическое восстановление плотности, весовые схемы при оценке плотности.
14. Линейная классификация и регрессия: персептронный алгоритм, режимы обучения, концепция поощрение-наказание, концепция минимизации функционала, линейная регрессия, SGD, хэширование признаков.
15. Регуляризация, обобщения регрессии, прогноз раскупаемости, прогноз методом kNN, прогноз линейным оператором, линейный алгоритм над SVD, признаковое прогнозирование спроса, профили товаров, сезонность, LibSVM, LibLinear.
16. Анализ текстов: классификация и регрессия - этапы работы с текстом, токенизация, стоп-слова, векторное представление документа, n-граммы, стемминг, алгоритм Портера, TF*IDF, оценки качества (точность, полнота, F-мера).
17. Классификация спама, Local and Global Consistency, этапные алгоритмы, устойчивые признаки, иерархическая классификация текстов, основные методы (Роше, kNN, SVM), приведение к шаблону, обнаружение оскорблений, распределение по топикам (задача со многими классами), блендинг алгоритмов, фонетические алгоритмы. Представление программы Vowpal Wabbit.
18. Случайные леса: универсальные методы анализа данных, бэггинг и бустинг, построение одного дерева, OOB(out of bag)-

проверка, параметры случайного леса (random forest: mtry, nodesize, samplesize) и их настройка, рейтинг признаков (importance). Программирование случайного леса.

19. Области устойчивости функционалов. Искусство генерации признаков: географические и временные признаки. Концепция чёрного ящика на примере GBM. Настройка параметров GBM, суммирование. Нестандартные функционалы и настройка на них. Калибровка ответов алгоритмов. Сведение задачи рекомендации к регрессии. Критерии расщепления.
20. Метод k ближайших соседей, настройка комбинаций алгоритмов: Сглаживание функционалов качества при использовании весовых схем. Ограничение методов типа kNN (тренд, некорректность метрики).

Список литературы

1) Fayyad U., Piatetsky-Shapiro G., Smyth P. From data mining to knowledge discovery in databases // AI magazine. – 1996. – Т. 17. – № 3. – С. 37.
2) Cook D. J., Holder L. B. (ed.). Mining graph data. – John Wiley & Sons, 2006.
3) Ясницкий Л.Н. Интеллектуальные системы: учебник. - М.: Лаборатория знаний. – 2016.
4) Барсегян А.А., Куприянов М.С., Степаненко В.В. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. - СПб.: БХВ-Петербург, 2007. - 384 с.
5) Чубукова И.А. Data Mining: учебное пособие. - М.: Интернет-университет информационных технологий БИНОМ: Лаборатория знаний, 2006. - 382 с.
6) Han J., Pei J., Kamber M. Data mining: concepts and techniques. – Elsevier, 2011.
7) Сигов А.С., Нечаев В.В., Кошкарёв М.И. Архитектура предметно-ориентированной базы знаний интеллектуальной системы // International Journal of Open Information Technologies. – 2014. – Т. 2. – № 12.
8) M. Seeger. Gaussian processes for machine learning. International Journal of Neural Systems, 14: 2004, 2004.
9) Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. - М.: Радио и связь. – 2002. – Т. 286. – С. 4.
10) Дюк В., Самойленко А. Data Mining: учебный курс. - СПб.: Питер. – 2001. – Т. 368. – С. 16.
11) http://sernam.ru
12) http://www.intuit.ru/studies/courses
13) http://xreferat.com/33/489-1-ponyatie-lingvisticheskoiy-peremennoiy-yazyk-programmirovaniya-prolog.html

Подписано в печать 27.07.2017г. Формат 60x90 1/16.
Объём 2,7 усл.п.л. Тираж 50 экз. Изд. № 78.

ВЫГОДНО. УДОБНО. НАДЕЖНО.

ИНТЕРНЕТ

WI-FI
СТАБИЛЬНАЯ СКОРОСТЬ
НАДЕЖНОЕ СОЕДИНЕНИЕ

ТЕЛЕВИДЕНИЕ

ИНТЕРЕСНЫЕ ТЕЛЕКАНАЛЫ СО ВСЕГО МИРА НА РАЗНЫХ ЯЗЫКАХ
HDTV

WWW.AKADO.RU

ОАО «КОМКОР», 117535, РОССИЯ, МОСКВА, ВАРШАВСКОЕ ШОССЕ, 133
ЛИЦЕНЗИИ № 123058, 123059, 123056, 123057, 153190, 153191, 153189, 123060