ГАОУ ВО «Дагестанский государственный университет народного хозяйства»

КобзаренкоДмитрий Николаевич
Мустафаев Арслан Гасанович

Кафедра «Информационные технологии и информационная безопасность»

Учебное пособие

по дисциплине

«Анализ больших данных»

(курс лекций)

Направление подготовки – 38.03.05 Бизнес-информатика, профиль «Электронный бизнес»

Махачкала – 2019

УДК 681.518(075.8)
ББК 32.81.73

Составитель – Кобзаренко Дмитрий Николаевич, доктортехнических наук, профессор кафедры «Информационные технологии и информационная безопасность» ДГУНХ.

Внутренний рецензент – Раджабов Карахан Якубович, кандидат экономических наук, доцент, декан факультета «Информационных технологий и управления» ДГУНХ.

Внешний рецензент – Качаева Гульханум Ибадулаховна, кандидат экономических наук, заведующий кафедрой«Информационная безопасность» Дагестанского государственного технического университета.

Представитель работодателя - Ботвин Тимур Анатольевич, руководитель группы «Кавказ» Яндекс.Такси.

Учебное пособие «Анализ больших данных» разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 38.03.05 Бизнес-информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 11.08.2016 г., № 1002, зарегистрированном в Минюсте России 26.08.2016 г., в соответствии с приказом от 5 апреля 2017г., № 301 Министерства образования и науки РФ.

Учебное пособие дисциплины «Анализ больших данных» размещено на сайте www.dgunh.ru

Кобзаренко Д.Н., Мустафаев А.Г. Учебное пособие дисциплины «Анализ больших данных» для направления подготовки 38.03.05«Бизнес-информатика», профиль «Электронный бизнес». – Махачкала: ДГУНХ, 2019 г. – 107 с.

Рекомендовано к утверждению Учебно-методическим советом ДГУНХ. Председатель Учебно-методического совета ДГУНХ, проректор по учебной работе, доктор экономических наук, профессор Казаватова Н.Ю. 29 мая 2019г.	Одобрено Советом факультета «Информационные технологии и управление» 25 мая 2019г., протокол № 6. Председатель Совета Раджабов К.Я, к.э.н., доцент
Одобрено на заседании кафедры «Информационные технологии и информационная безопасность» 20 мая 2019 г., протокол № 10 Зав.кафедрой, к.ф.-м.н., доцент Галяев В.С.	

Печатается по решению Учебно-методического совета Дагестанского государственного университета народного хозяйства.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Учебное пособие предназначено для студентов 4 курса, обучающихся на дневном отделении факультета «Информационные технологии и управление», направления «Бизнес-информатика».

Объем дисциплины в зачетных единицах составляет 2 зачетные единицы.

Количество академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), составляет 36 часов, в том числе:

на занятия лекционного типа – 12 ч.

на занятия семинарского типа – 24 ч.

Количество академических часов, выделенных на самостоятельную работу обучающихся – 36 ч.

Форма промежуточной аттестации –зачет.

ТЕМА №1: ВВЕДЕНИЕ В БОЛЬШИЕ ДАННЫЕ

Большие данные: определение

Большие данные – комплексный набор методов, подходов и инструментов обработки структурированных и неструктурированных данных колоссальных объемов. Главной целью обработки Big Data является быстрое и эффективное использование всех видов информации в условиях непрерывного изменения и прироста в больших объёмах.

Говоря простым языком, Big data представляет собой безмерный объем информации, который не может быть обработан стандартными инструментами и аппаратными средствами. Основными задачами Big Data являются хранение и обработка информации гигантских объёмов данных.

Большие данные по сравнению с обычными данными требуют иной подход к обработке. При обработке Big data используются собственные инструменты и технологии, которые предназначены для данных со сверхбольшим объёмом информации.

Big Data терминология

Термин Big Data был предложен Клиффордом Линчем (Clifford Lynch) (рисунок 1.1), редактором журнала Nature, который 3 сентября 2008 года выпустил отдельный номер, главной темой которого была «Как могут повлиять на будущее науки, технологии, открывающие возможности работы с большими объёмами данных?» (Оригинал: «Big data: How do your data grow?»). Термин «Большие данные» был предложен по аналогии с терминами «Большая нефть», «Большая руда» и т. д.

Размер больших данных в 2012 году определялся от нескольких десятков терабайт до петабайт (2^{50}). Термин большие данные может быть причислен к данным, связанным с высочайшей изменчивостью источников данных, а также обладающим сложными взаимосвязями и трудностями изменения или удаления отдельных записей. Большие данные характеризуются гигантским объёмом, значительной скоростью поступления данных, а также

многообразием самих данных. Для таких данных требуются новейшие способы обработки, которая в дальнейшем может привести к улучшению методов принятия решений, оптимизации процессов и поиска закономерностей.

Рисунок 1.1 – Клиффорд Линч

История появления термина Big Data

К 2011 году понятие Big Data стало набирать популярность, в основном, в крупных корпорациях таких как Microsoft, IBM, Oracle, EMC, HP и др.

В 2011 году исследовательская компания Gartner отмечает большие данные как тренд номер два в информационно-технологической инфраструктуре после виртуализации. По прогнозам подразумевается, что внедрение технологий Big Data крупно повлияет на информационные технологии в сферах производства, здравоохранении, торговли, государственного управлении, а также в отраслях, в которых регистрируются индивидуальные перемещения ресурсов. С 2013 года большие данные начинают преподавать в университетах в рамках вузовских программ по науке о данных, вычислительным наукам и инженерии.

Причины появления Big Data

Инновационные разработки в области Big Data начинались не в маленьких стартапах, как это часто бывает в IT-индустрии, а в больших компаниях. Так, например, технология распределенной обработки данных MapReduce была разработана компанией Google, а Hadoop, являющийся

свободным программным обеспечением для выполнения распределенных вычислений на кластерах из сотен и тысяч узлов, сразу после создания активно поддержала компания Yahoo. Большинство программных продуктов в области Big Data являются свободными, а их адаптацией и продвижением занимаются те самые стартапы. Традиционные поставщики решений в области хранения и обработки данных, такие как IBM и 3

EMC внимательно относятся к новым разработкам в области Больших Данных и стараются использовать их в своих продуктах совместно с собственными технологиями.

Характеристики Big Data

Существует множество характеристик для Больших данных, но здесь будут рассматриваться самые основные.

Сфера Больших Данных характеризуется следующими признаками:

Volume (объем): накопленная база данных представляет собой гигантский объем информации, для которого обработка и хранение традиционными способами являются трудоёмкими процессами. Такой объем нуждается в новых подходах и в более усовершенствованных инструментах.

- **Velocity** (скорость): данный признак указывает как на увеличивающуюся скорость накопления, так и на скорость обработки данных. В последнее время стали более востребованы технологии обработки данных в реальном времени.
- **Variety** (многообразие): данная характеристика означает возможность одновременной обработки структурированной и неструктурированной информации различных форматов. Главным отличием структурированной информации является возможность классификации. Примером такой информации может служить информация о клиентских транзакциях.
- **Veracity** (достоверность данных): в настоящее время достоверность имеющихся данных является важнейшим критерием для пользователей. Недостоверная информация приводит к затруднению анализа данных.
- **Value** (ценность накопленной информации): Большие Данные должны

быть полезны в усовершенствовании бизнес-процессов, составлении отчетности или оптимизации расходов компаний.

Первые три характеристики определяют так называемый принцип «Трёх V» (рисунок 1.2).

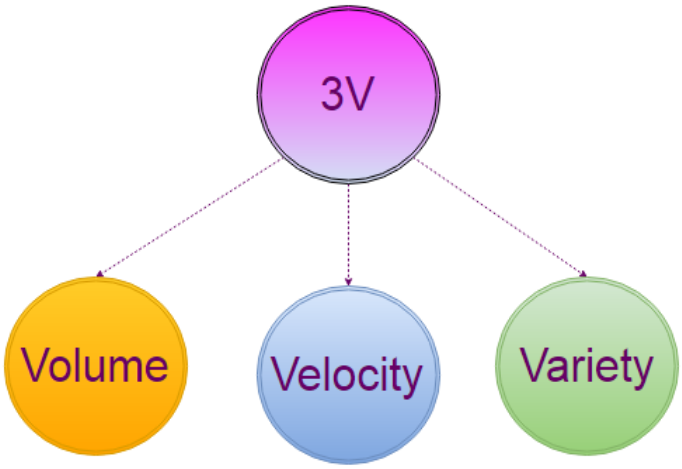


Рисунок 1.2 – Принцип «Трёх V»

По рисунку 1.2 видно, что решающую роль в больших данных играют объем информации, скорость обработки, а также разнообразие появляющихся данных.

Объём относится к наборам данных, размер которых выходит за пределы возможностей программных средств типичной базы данных сбора, хранения, обработки и анализа данных.

Разнообразие определяет способность обработки множества типов, источников и форматов данных от сенсоров, умных устройств, социальных сетей. Также разнообразие характеризуется способностью интегрировать все большее число источников, содержащих различные структурированные, полуструктурированные данные, извлекаемыми из web-страниц, web log файлов, e-mail, документов и др.

Скорость определяет реакцию на текущую информацию за время, ограниченное приложением. Примером является потоковая обработка (например, GPS данных в реальном времени).

Рост объемов информации

По прогнозам, количество данных на планете будет удваиваться каждые два года вплоть до 2020 года. А за период между 2013 и 2020 годом количество информации увеличится десятикратно – с 4,4 трлн гигабайт до 44 трлн. При этом значительная часть произведенных к настоящему моменту данных ни разу не была исследована с помощью специализированных аналитических инструментов. По оценкам International Data Corporation (IDC), к 2020 году только 35% данных будет содержать ценную для анализа информацию.

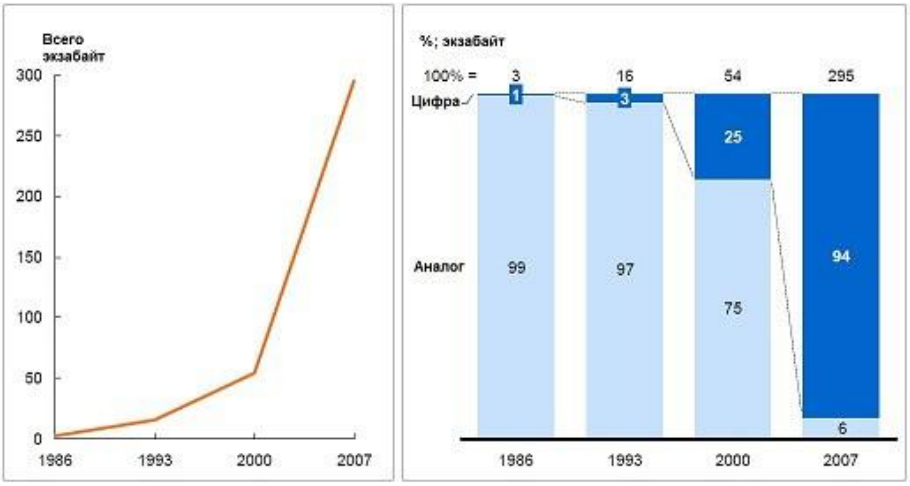


Рисунок 1.3 – Рост объемов данных (слева) на фоне вытеснения аналоговых средств хранения (справа). Источник: Hilbert and López, «The world's technological capacity to store, communicate, and compute information», 2011

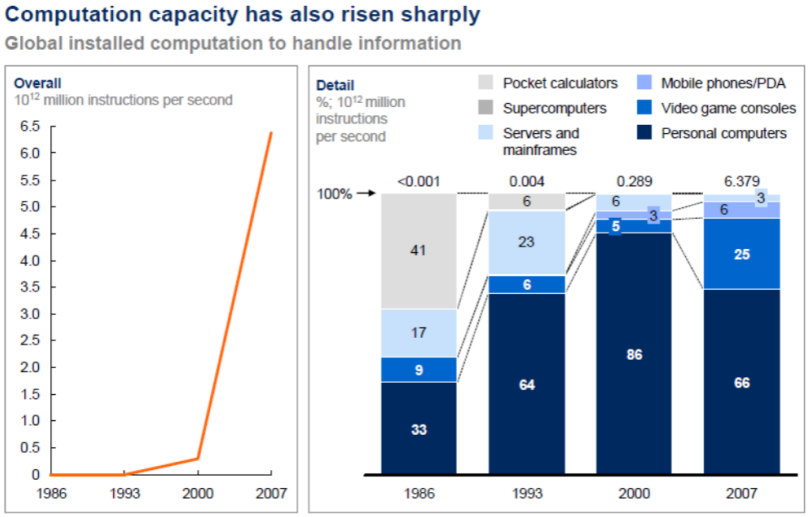


Рисунок 1.4 – Рост вычислительных мощностей

	Video	Image	Audio	Text/ numbers
Banking				
Insurance				
Securities and investment services				
Discrete manufacturing				
Process manufacturing				
Retail				
Wholesale				
Professional services				
Consumer and recreational services				
Health care				
Transportation				
Communications and media[2]				
Utilities				
Construction				
Resource industries				
Government				
Education				

High
Medium
Low

Рисунок 1.5 – Типы данных по отраслям

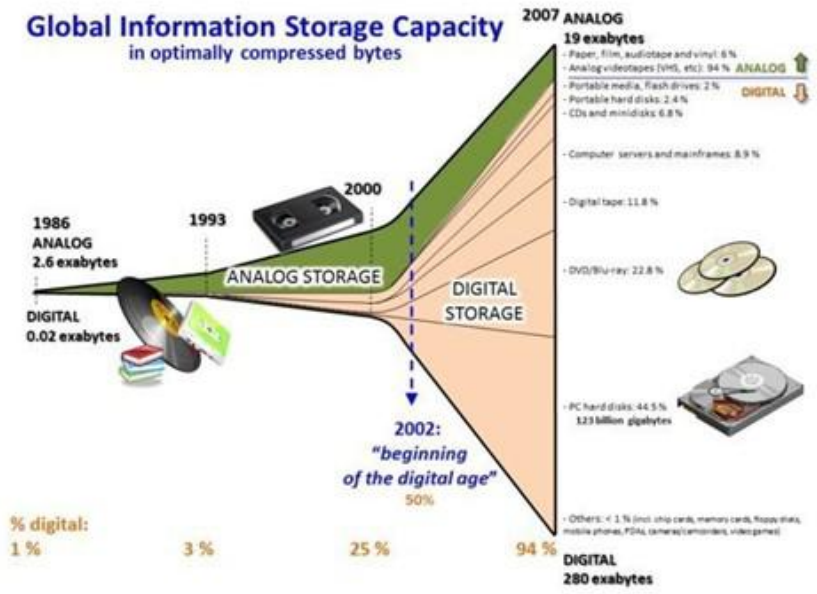


Рисунок 1.6 – Динамика роста запроса Big Data

Источники Big Data

Источниками Big Data являются:

- Социальные сети и их данные
- Данные от измерительных устройств
- Журналы доступа пользователей веб-сайтов

- Сенсорные сети
- Тексты и документы из Интернета
- Научные данные (астрономия, геном человека, исследования атмосферы, биохимия, биология)
- Данные министерства обороны
- Медицинские наблюдения
- Фото- и видео-архивы
- Данные электронной коммерции

Основной причиной появления больших данных являются достижения в области мобильных устройств, такие как цифровое видео, фотографии, аудио, а также современные системы электронной почты и обмена текстовыми сообщениями. Пользователи получают данные в количествах, которые нельзя было представить десять лет назад; при этом появляются новые приложения, такие как Google Translate, предоставляющие функции сервера больших данных – перевод произнесенных или введенных с мобильных устройств фраз. Корпорация IBM в отчете "Тенденции развития технологий в 2013 году говорит в первую очередь о доступе к большим данным с мобильных устройств и характеризует большие данные по объему (volume), разнообразию (variety), скорости (velocity) и достоверности (veracity). Эти данные гораздо менее структурированы, чем записи в реляционных базах данных, но могут быть прокоррелированы с ними.

Архитектуры защиты данных в общем случае должны обеспечивать защиту от потери, неотслеживаемого повреждения, вредоносных программ и злонамеренного изменения данных киберпреступниками или в результате кибервойн. Данные являются активами и все чаще используются правительствами и деловыми кругами для принятия ключевых решений, но если достоверность данных неизвестна, ценность их снижается или даже может быть утрачена, и что еще хуже – могут быть приняты плохие решения.

Четыре основных типа данных

В настоящее время все существующие данные делятся на (рисунок 1.7):

- Структурированные
- Полуструктированные
- Квази структурированные
- Неструктурированные

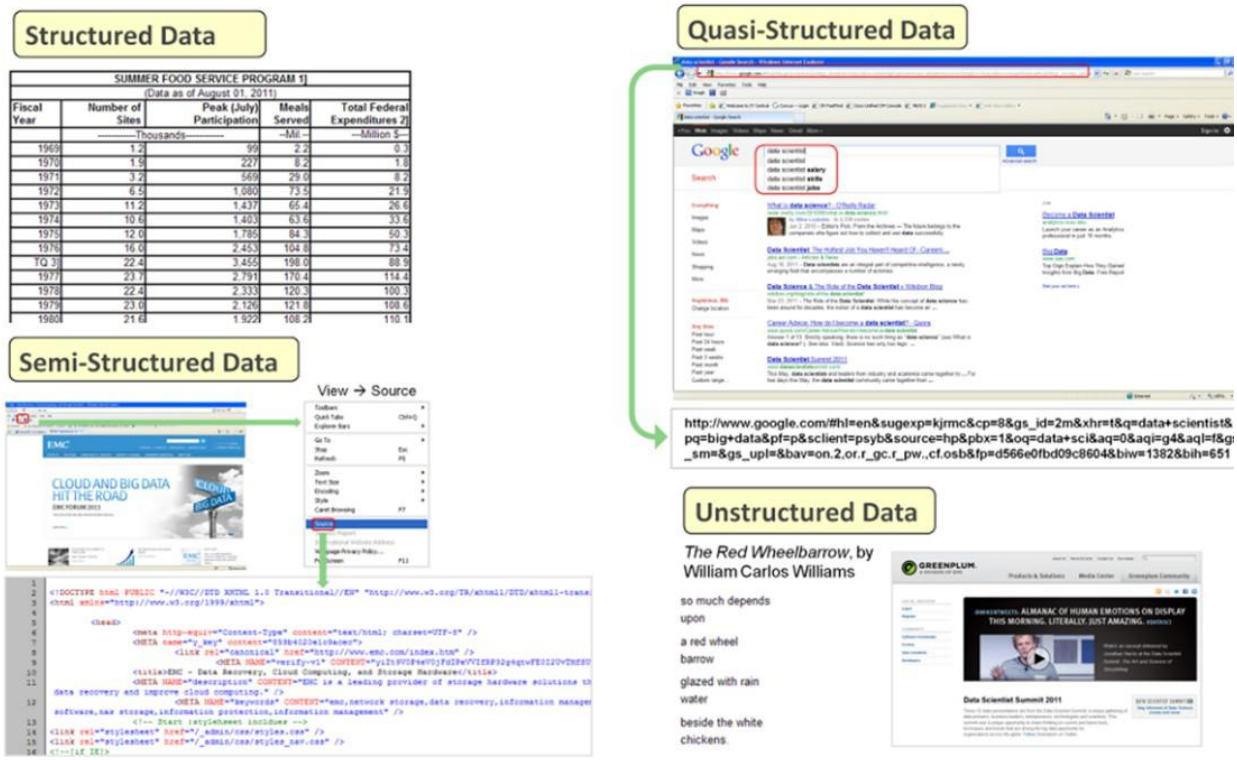


Рисунок 1.7 – Основные типы данных

К структурированным данным относятся данные, определяющие конкретную предметную область. Такие данные упорядочены специальным образом и организованы таким образом, чтобы над такими данными можно было выполнить анализ. Обычно такие данные хранятся в виде таблиц в реляционных базах данных. Почти все алгоритмы машинного обучения и Data Mining работают со структурированными данными.

К полуструктурированным данным относятся данные, которые не соответствуют чёткой структуре таблиц и отношений в реляционных базах данных, однако такие данные содержат специальные теги и иные маркёры, позволяющие отделить семантические элементы. Такие данные принадлежат одному классу, но при этом могут иметь разные атрибуты. Xml документы

являются простейшем примером полуструктурированных данных.

К неструктурированными данным относятся данные, которые не имеют определённой формы, могут включать в себя видео, аудио файлы, свободный текст, информацию, поступающую из социальных сетей. На сегодняшний день 80% информации входит в группу неструктурированной. Такую информацию необходимо комплексно анализировать, для упрощения ее дальнейшей обработки.

Аналитика данных

Анализ данных – это совокупность методов и средств извлечения их организованных данных для принятия решений (рисунок 1.8-1.9).

Основные этапы анализа данных:

1. Гипотеза
2. Сбор и систематизация данных
3. Подбор модели
4. Тестирование и интерпретация результатов
5. Использование

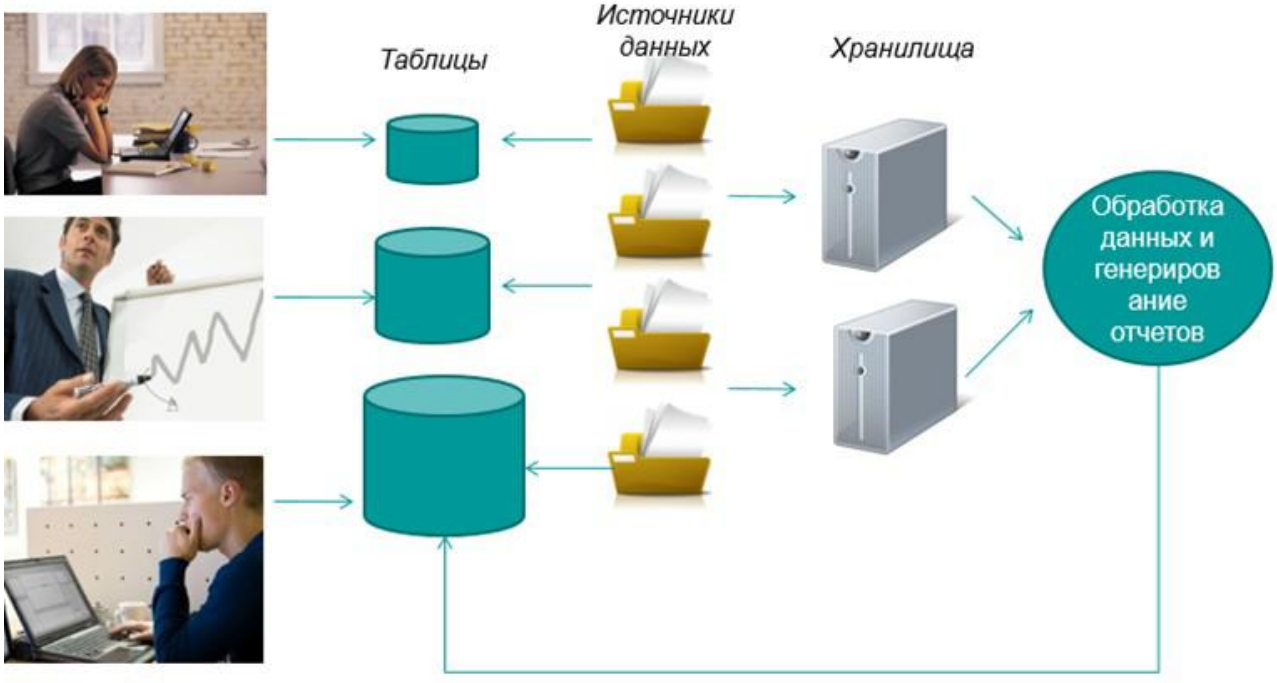


Рисунок 1.8 – Типичная архитектура аналитики

Рисунок 1.9 – Технология сбора и анализа данных

Таблица 1.1 – Аналитика Больших Данных в различных областях

Область	Цель
Здравоохранение	Уменьшение стоимости лечения
Публичные сервисы	Предотвращение пандемии
Наука о жизни	Расшифровка генома человека
IT индустрия	Анализ неструктурированных данных
Online сервисы	Медиа для профессионалов

Задачи, решаемые Big Data:

- Управление документами и доступом
- Выявление мошенничества
- Системы гарантирования доходов
- Анализ текучести клиентов
- Мониторинг интеллектуальных счётчиков
- Мониторинг оборудования
- Оптимизация ценовой политики
- Оптимизация автомобильного движения
- Анализ социальных сетей

- Анализ поведения клиентов
- Оптимизация ИТ инфраструктуры
- Прогноз погоды
- Разведка природных ресурсов
- Управление контентом для предоставления в судебной практике
- Аналитика в медицинских исследованиях
- Управление гарантийными обязательствами
- Анализ рекламных компаний
- Оптимизация web
- Исследования в области естественных наук

Статистика объёма данных

- Каждый день в мире производится **2,5 квинтильона** (10^{18}) байтов данных. При этом **90%** данных созданы за последние два года.
- Каждый час Wal-Mart совершает 1 миллион сделок, пополняя базу данных на **2,5 петабайта** (10^{15}) в 170 раз больше объема данных Библиотеки Конгресса США.
- Объем отправлений, доставляемых американской Почтовой службой за один год, равен **5 петабайтам**, а Google обрабатывает такой же объем данных всего за один час.
- Суммарный объем всей существующей на земле информации составляет несколько больше **одного зеттабайта** (10^{21}).
- Бизнесу во всем мире необходимо собрать, организовать, проанализировать все эти данные для достижения конкурентных преимуществ в конкретной сфере рынка, развития более эффективных деловых операций, увеличения удовлетворения запросов потребителей, выявления новых возможностей.

Big data для операторов связи

Для операторов связи методы больших данных рассматриваются в четырёх различных направлениях. Одни из них используются для улучшения

внутренненой работы программы, другие играют роль дополнительного рыночного продукта для внешних клиентов.

- **высокоточный маркетинг** (precise marketing) — адресное предложение продуктов и услуг тем потребителям, которые наиболее готовы к их приобретению (новые тарифные планы, дополнительные сервисы, платежные терминалы и пр.);
- **управление качеством услуг для клиента** (Customer Experience Management) для повышения его удовлетворенности с целью предотвращения оттока пользователей;
- **оптимизация внутренней работы оператора и планирование развития** (ROI-based Network Optimization and Planning) на основе учета всех объективных факторов и мнений потребителей с целью максимальных гарантий возврата инвестиций в кратчайшие сроки;
- **монетизация информационных** активов (Data Asset Monetization) – продажа в той или иной форме (в том числе в виде долевого участия в проектах) имеющихся у оператора данных своим партнерам, чтобы они могли с их помощью решать свои задачи.

Развернув решение больших данных, мобильный оператор смог начать собирать и анализировать существенно больше информации о поведении и интересах своих клиентов, в том числе об интенсивности использования связи и географическом местоположении. Причем все эти сведения можно было увязывать с данными о работе самой сотовой сети, в том числе о ее загрузке, о возникающих сбоях и пр.

Возможности применения подобных методов видны по полученным результатам. Так, в начале 2013 г. эффективность маркетинговых предложений (для клиентов, которые их приняли) при общей массовой рассылке составляла 0,7%. К концу года за счет простой сегментации абонентов (по возрасту, полу, сроку подписки) эта величина была доведена до 4%, а в течение 2014-го повышена сначала до 11% (учет интенсивности использования услуг и местоположение клиентов) и затем до 24% (учет предпочтительных вариантов

получения предложения – голосовые звонки, SMS, э-почта, социальные сети и пр.). За год удалось сократить число нерезультативных обращений к клиентам на 11 млн., существенно снизив затраты на рекламные кампании.

На основе анализа 85 параметров поведения абонентов была выделена «группа риска», потенциально готовая к уходу от услуг оператора. Внутри нее также была проведена определенная сегментация, и для каждой категории клиентов выработан комплекс мероприятий по повышению уровня их лояльности (скидки, другие тарифные планы, подарки и пр.). Заказчик провел исследование, разделив «группу риска» на две подгруппы: с первой проводились специальные действия по удержанию, с другой ничего не делалось. Анализ такой работы за год показал, что компания смогла существенно сократить отток своих действующих потребителей, удержав более 200 тыс. абонентов; при этом нужно учитывать, что стоимость удержания клиента всегда значительно ниже, чем привлечения нового пользователя.

До использования больших данных расширение географической сети оператора фактически выполнялось только на основе информации о плотности застройки и населения, но внедрив это решение, China Unicom перешел к развитию свой деятельности на базе многофакторного анализа, который учитывал такие показатели, как реальная загруженность трафика и востребованность услуг (например, с учетом места работы людей), «ценность» клиентов (по уровню жизни), требования к качеству связи (расстояние между станциями приема), востребованность разных категорий услуг (от этого зависит использование различной аппаратуры) и пр.

В плане монетизации клиентских данных для внешних партнеров были приведены два примера: во-первых, оптимизация размещения наружной рекламы, причем как в географическом плане (место проживания, работа или транспортные коммуникации нужных клиентов), так и с учетом времени для динамической рекламы (в зависимости от времени суток, дней недели и сезонов года состав публики может меняться), а во-вторых, аналогичные предложения по развитию торговых сетей (с учётом местоположения и ассортимента). Кроме

того, очень выгодным оказывается целевая рассылка мобильной рекламы в реальном времени в соответствии с графиком занятости человека, его интересов и физического пребывания (например, рассылка информации о фильмах-боевиках, которыми клиент интересуется, именно в его свободное время и с учетом близлежащих кинотеатров). Общий отраслевой опыт показывает, что такие адресные методы позволяют повышать доходы от распространения рекламы в разы.

Big data в банках

Массовое внедрение технологий анализа больших данных осложнено тем, что банки зачастую используют разрозненные или просто устаревшие платформы. Тем не менее, уже есть примеры того, как сотрудники, отвечающие за информационную безопасность, предотвращали мошеннические операции. Помимо технологии Big Data эксперты также считают, что бороться с мошенниками позволяет внедрение современных систем идентификации пользователей. Одним из примеров является так называемая непрерывная поведенческая идентификация, анализирующая поведение клиентов на протяжении длительного времени. Это делается при помощи привязки счета к мобильному телефону.

Большие данные способны решать практически все ключевые задачи банков: привлечение клиентов, повышение качества услуг, оценка заемщиков, противодействие мошенничеству и др. Повышая скорость и качество формирования отчетности, увеличивая глубину анализа данных, участвуя в противодействии отмыванию незаконных средств, эти технологии помогают банкам соответствовать требованиям регуляторов.

Основные задачи, для которых банки используют технологии анализа больших данных, – это оперативное получение отчетности, скоринг, недопущение проведения сомнительных операций, мошенничества и отмывания денег, а также персонализация предлагаемых клиентам банковских продуктов.

Технологии больших данных применяются в основном для анализа

клиентской среды. Дмитрий Шепелявый, заместитель генерального директора SAP СНГ, приводит несколько примеров: «Американский банк PNC данные о поведении своих клиентов на сайтах, информацию о покупках и образе жизни конвертирует в политику гибкого начисления процентных ставок, которая в итоге выражается в цифрах роста капитализации. Commonwealth Bank of Australia (CBA) анализирует все транзакции своих вкладчиков, дополняя этот анализ сбором данных о них в социальных сетях. Связав эти потоки данных, банк добился значительного снижения процента неуплаты по кредитам. А в России интересен опыт Уральского банка реконструкции и развития – они стали работать с информацией по клиентской базе для создания кредитных предложений, вкладов и других услуг, которые могут максимально заинтересовать конкретного клиента. Примерно за год применения ИТ-решений розничный кредитный портфель УБРиР вырос примерно на 55%».

Европейские программы, связанные с Big Data

- **GRDI2020:** Глобальная инфраструктура исследовательских данных (GRDI) в 10-летней перспективе. Инфраструктура разрабатывается для использования накапливаемых данных и знаний, производимых научным сообществом
- **ESFRI**: Европейский стратегический форум в области исследовательских инфраструктур. Дорожная карта ESFRI 2010 ориентирована на инвестирование €20b, включая ежегодные €2b операционных затрат для поддержки стратегических политик для Пан-Европейских исследовательских инфраструктур
- **SIENA**: Стандарты и интероперабельность в инициативе реализациии e-инфраструктур. Посвящена открытым, стандартизованным, интероперабельным грид и облачным вычислительным инфраструктурам
- **VENUS-C**: Виртуальные многодисциплинарные среды, использующие облачные инфраструктуры. Совместная программа промышленных партнеров и научного сообщества пользователей облачных инфраструктур. Охватывает 10 тематических областей: биоразнообразие, химию,

строительство и архитектуру, гражданскую оборону, науки о земле, социальные медиа, математику, механику и аэрокосмическую технику, здравоохранение, генетическую биологию, астрофизику.

Контрольные вопросы для повторения и самопроверки темы №1:

1. Что означает термин «Big Data» в информационных технологиях?
2. Что является основной целью обработки Big Data?
3. Кто и в каком году впервые ввел термин «Big Data»?
4. Какие главные характеристики Big Data?
5. Какие данные занимают больше мировой памяти относительно остальных?
6. Какие понятия содержит в себе принцип трех "V"?
7. С какого года Большие данные изучаются как академический предмет в вузовских программах по науке о данных?
8. Что является примером квази-структурированных данных?
9. Как назывался первый суперкомпьютер, оснащенный вопросно-ответной системой искусственного интеллекта?
10. Чем характеризуются "Большие данные"?

Литература по теме №1:

1. Майер-Шенбергер В., Кукьер К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим. Language Arts & Disciplines – 2013. – 599с.
2. Силен Д., Мейсман А., Али М. Основы DataScience и BigData. Python и наука о данных. СПб: Питер. – 2017. – 336с.

Глоссарий по теме №1:

Большие данные – комплексный набор методов, подходов и инструментов обработки структурированных и неструктурированных данных колоссальных объемов.

Принциптрех «V» – Volume, Velocity, Variety.

Volume – объем данных, относится к наборам данных, размер которых выходит

за пределы возможностей программных средств типичной базы данных собирать, хранить, обрабатывать и анализировать данные.

Velocity – скорость, реакция на текущую информацию за время, ограниченное приложением; потоковая обработка.

Variety – разнообразие, способность обработки множества типов, источников и форматов данных.

Структурированные данные – данные, имеющие определенный тип, формат и структуру.

Полуструктурированные данные – данные текстовых файлов с определенными паттеранами для их обработки.

Квазиструктурированные данные – текстовые данные с неустойчивым форматом, которые для обработки инструментами требуют больших временных затрат на преобразование.

Неструктурированные данные – данные, у которых нет строго зафиксированного формата.

ТЕМА №2: ЖИЗНЕННЫЙ ЦИКЛ АНАЛИТИКИ ДАННЫХ

Приведем жизненный пример того, как и при каких условиях необходимо производить анализ данных.

Представим, для аналитика стоит задача оценить продажи ряда магазинов (куда поставляется товар) и каждый магазин ведет свой учет проданных товаров. Реальность такова, что формы учета будут заполнены разными способами, то есть у них будет разная структура и разный формат хранения (некоторая форма таблиц).

Решим эту задачу способом, который первый приходит в голову, то есть предоставим простейший способ решения данной задачи. Пусть у нас есть N таблиц и нам нужно их собрать вместе в одну таблицу. Напишем N скриптов, которые преобразуют эти таблицы, и один сборщик, который собирает их вместе. У такого подхода есть положительные и отрицательные стороны. К минусам можно отнести: необходимо поддерживать N скриптов одновременно (где N в порядках тысяч); при изменении структуры отчетов магазинов во времени (например, в магазине появился новый сотрудник) необходимо искать и переписывать отдельные скрипты; при появлении нового магазина, необходимо писать новый скрипт; при изменении формата отчетности необходимо вносить изменения во все скрипты; сложная отладка и поддержка, так как магазины не уведомляют об изменении структуры и не следуют никаким спецификациям.

Немного усложним задачу и приблизим ее к реалиям современных компаний. Итак, производитель товаров на самом деле работает не напрямую с магазинами, а через некоторых посредников. Посредники посещают магазины и непосредственно своими действиями пытаются стимулировать продажи. Соответственно, они являются материально заинтересованными лицами и информацию, которую они выдают, приходится перепроверять.

На основании вышесказанного давайте переформулируем задачу. Пусть у нас есть N магазинов и K дистрибьюторов. Можем ли мы агрегировать

данные магазинов и сравнить их с результатами дистрибьюторов? При этом у всех данные имеют разную структуру и формат.

Эта задача обладает несколькими основными характеристиками. Входные данные представляют собой огромное количество разнообразных форматов, к которым добавляются отчеты дистрибьюторов. Как правило, задача характеризуется очень низким качеством данных, в том числе дублированием, несогласованностью и ошибками. На основе полученных результатов и сравнения данных отдел по закупкам принимает решения о том, сколько, что, кому и по какой цене отгружать. То есть решение этой задачи непосредственно влияет на финансовые показатели компании, что безусловно важно.

Существуют также другие варианты решения подобной задачи. Например, самописное решение. Компании производителю будет необходимо нанять специалиста не по профилю компании и критичное ПО будет зависеть от данного специалиста. Если он уйдет, то компания будет вынуждена срочно искать замену, которая сможет поддерживать ПО, и качество будет напрямую зависеть от нанятого специалиста.

Также всегда есть возможность закупить ПО у третьей стороны. Отрицательной стороной такого подхода являются цена, качество и время интеграции. Как правило, цена и время интеграции слишком высоки для среднего производителя и в том числе требует существенных временных затрат сотрудников. Выбор поставщика также не тривиален.

Есть вариант применить SaaS решения, но методология еще нова для рынка и многие компании скептически относятся к подобным сервисам.

Процесс анализа на уровне компании в общем случае выглядит следующим образом: данные консолидируются, определенным образом трансформируются (агрегируются) и загружаются в систему для анализа.

Business Intelligence

Термин Business Intelligence был введен аналитиками Gartner как «процесс, ориентированный на бизнес-пользователя и включающий доступ и

исследование информации, ее анализ, выработку интуиции и понимания, которые ведут к улучшенному и неформальному принятию решений».

Иными словами, можно сказать, что BI – это методы и инструменты для перевода необработанной информации в осмысленную, удобную форму. При этом технологии BI обрабатывают большие объемы неструктурированных данных, чтобы найти стратегические возможности для бизнеса.

Business Intelligence имеет отношение к процессу превращения данных в знания, а знаний - в действия бизнеса для получения выгоды; является деятельностью конечного пользователя, которую облегчают различные аналитические и групповые инструменты и приложения, а также инфраструктура хранилища данных.

Главная цель BI состоит в том, чтобы интерпретировать большое количество данных, заостряя внимание лишь на ключевых факторах эффективности, моделируя исход различных вариантов действий, отслеживая результаты принятия решений.

BI-платформа

Для анализа данных на сегодняшний день существует множество решений. BI-платформа – это такая программа, которая предоставляет пользователю удобные инструменты анализа фактически любых данных (будь то файл Excel либо промышленное хранилище данных). Проще говоря, BI-платформа – это продвинутая система отчетности.

Современные системы класса BI располагают следующими основными функциями: возможность подключения к различным источникам данных (от файла Excel до универсального ODBC подключения); возможность построения как простых отчетов (типа график или таблица), так и сложных параметризированных отчетов с комбинированной структурой и ссылочными связями (Drill-Trough, Drill-Up/Drill-Down); возможность прозрачной работы с разными источниками данных (например, Excel и SQL Server) с полноценной обработкой связей между ними; возможность интерактивной работы с данными (формирование отчетов «на лету»); возможность представления реляционных

данных как многомерные; возможность распределения прав доступа, используя как внутренние источники аутентификации, так и внешние (NTLM, LDAP и т. д.); возможность запуска формирования отчетов как вручную, так и автоматически по расписанию; возможность автоматической рассылки сформированных отчетов; возможность построения отчетов в различных форматах (Excel, HTML, PDF и т. д.).

BI состоит из множества компонентов и обладает различными характеристиками: многомерная агрегация и размещение данных в хранилищах; денормализация баз данных; маркировка и стандартизация данных (ETL); отчетность в режиме реального времени с аналитическими оповещениями; статистические выводы и вероятностное моделирование.

BI должна быть разделена на этапы: информационный поиск, аналитическая обработка в реальном времени (OLAP), предупреждения об отклонениях от ожидаемых показателей, бизнес-аналитика, бизнес-отчётность.

Разница между Business Intelligence и аналитикой заключается в том, что последняя включает в себя комплекс методов для анализа уже чистых данных. В отличии от аналитики, основная задача BI – принятие решений для бизнеса.

ETL-процесс

В области аналитики данных часто употребляется аббревиатура ETL (Extract, Transform, Load). Это термин, который переводится с английского дословно как «извлечение, преобразование, загрузка», является одним из основных процессов в управлении хранилищами данных.

С точки зрения процесса ETL, архитектуру хранилища данных можно представить в виде трёх компонентов: источник данных, который содержит структурированные данные в виде таблиц, совокупности таблиц или просто файла (данные в котором разделены символами-разделителями); промежуточная область, содержащая вспомогательные таблицы, создаваемые временно и исключительно для организации процесса выгрузки; получатель данных, который является хранилищем данных или базой данных, в которую должны быть помещены извлечённые данные.

Средства BI

В соответствии с подходом аналитиков выделяют три основных типа инструментальных средств BI: средства предоставления информации, средства интеграции, средства анализа.

Средства предоставления информации подразделяются на средства создания отчетов, которые дают возможность создавать форматированные интерактивные отчеты, и на информационные панели показателей, являющиеся одной из составных частей отчетов, представляющие информацию в виде интуитивно понятного графического изображения, включая диаграммы, круговые шкалы, светофоры и т.п. Также существуют такие инструменты, как генераторы нерегламентированных запросов – данная функция, известная также как создание отчетов в режиме самообслуживания, дает пользователям возможность получать ответы на возникающие вопросы.

Наряду со средствами создания отдельных BI-приложений, BI-платформа должна предоставлять средства программной разработки для интеграции приложений в общий бизнес-процесс или обеспечивать их встраивание в другое приложение. BI-платформа должна давать разработчикам возможность создания BI-приложений без кодирования, на основе применения мастеров для визуального редактирования.

OLAP

Одним из средств анализа данных является технология OLAP (Online Analytical Processing — Оперативная аналитическая обработка данных). Это класс приложений и технологий, предназначенных для сбора, хранения и анализа многомерных данных в целях поддержки принятия управленческих решений. Технология OLAP позволяет аналитикам, менеджерам и управляющим сформировать свое собственное видение данных, используя быстрый, единообразный, оперативный доступ к разнообразным формам представления информации.

Одновременный анализ по нескольким измерениям определяется как многомерный анализ. Каждое измерение включает направления консолидации

данных, состоящие из последовательных уровней обобщения, где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению.

OLAP представляет собой инструмент для анализа больших объемов данных в режиме реального времени и обеспечивает следующие возможности работы с многомерными данными: гибкий просмотр информации, произвольные срезы данных, детализация, свертка или консолидация, вращение, сравнение во времени.

OLAP - Многомерный куб

В ячейках многомерного куба помещаются числовые параметры, предназначенные для анализа, например, объемов продаж. Измерениями OLAP-куба могут служить такие параметры, как время, продукты, регионы, продавцы. Продажи по времени в консолидированном виде могут представляться по годам, при детализации – по кварталам, месяцам и дням.

Пример многомерного куба

Возьмем для примера таблицу Invoices, которая содержит заказы фирмы. Поля в данной таблице будут следующие: дата заказа, страна, город, название заказчика, компания-доставщик, название товара, количество товара, сумма заказа.

Какие агрегатные данные мы можем получить на основе этого представления? Обычно это ответы на следующие вопросы. Какова суммарная стоимость заказов, сделанных клиентами из определенной страны? Какова суммарная стоимость заказов, сделанных клиентами из определенной страны и доставленных определенной компанией? Какова суммарная стоимость заказов, сделанных клиентами из определенной страны в заданном году и доставленных определенной компанией?

Все эти данные можно получить из этой таблицы вполне очевидными SQL-запросами с группировкой. Результатом этого запроса всегда будет столбец чисел и список атрибутов, его описывающих (например, страна) – это одномерный набор данных или, говоря математическим языком, вектор.

Представим себе, что нам надо получить информацию по суммарной стоимости заказов из всех стран и их распределение по компаниям доставщиков – мы получим уже таблицу (матрицу) из чисел, где в заголовках колонок будут перечислены доставщики, в заголовках строк – страны, а в ячейках будет сумма заказов. Это – двумерный массив данных. Такой набор данных называется сводной таблицей (pivot table) или кросс-таблицей.

Если же нам захочется получить те же данные, но еще в разрезе годов, тогда появится еще одно изменение, т.е. набор данных станет трехмерным (условным тензором 3-го порядка или 3-х мерным «кубом»).

Очевидно, что максимальное количество измерений – это количество всех атрибутов (Дата, Страна, Заказчик и т.д.), описывающих наши агрегируемые данные (сумму заказов, количество товаров и т.п.)

Также стоит добавить, что для удобства работы с OLAP существуют языки запросов к многомерным кубам, например, MDX.

Продвинутая визуализация

Инструменты продвинутой визуализации позволяют представлять данные для более эффективного их восприятия посредством использования интерактивных картинок и диаграмм вместо таблиц.

Обычно пользователи в динамическом режиме могут менять графическое представление, использовать масштабирование, объединять данные, изменять цвета.

Предиктивное моделирование и Data Mining

Предиктивное моделирование (Predictive Modelling) – это процесс создания (или выбора) модели для предсказания вероятности наступления некоторого события.

Интеллектуальный анализ данных (Data Mining) – компьютерная техника извлечения знаний, которая использует искусственный интеллект для распознавания образов и выделения значимых закономерностей из данных, находящихся в хранилищах или во входных / выходных потоках. Эти методы основываются на статистическом моделировании, нейронных сетях,

генетических алгоритмах и др. Частная методология text mining решает задачи навигации в больших текстовых массивах, поиск взаимосвязей между ключевыми понятиями текстов, структуризация хранилищ документов, поиск информации, выраженный на естественном языке, распределение по рубрикам. Информация, найденная в процессе использования методов Data Mining, должна описывать новые связи между свойствами, предсказывать значения одних признаков на основе других и т.д. Найденные знания должны быть применимы и по отношению к новым данным с некоторой степенью достоверности. Когда извлеченные знания непрозрачны для пользователя, должны существовать методы постобработки, позволяющие привести их к интерпретируемому виду.

Задачи, решаемые методами Data Mining, включают: классификацию – отнесение объектов (наблюдений, событий) к одному из заранее известных классов; регрессию, в том числе задачи прогнозирования; установление зависимости непрерывных выходных от входных переменных; кластеризацию – группировку объектов (наблюдений, событий) на основе данных (свойств), описывающих сущность этих объектов; ассоциацию – выявление закономерностей между связанными событиями; последовательные шаблоны – установление закономерностей между связанными во времени событиями, то есть обнаружение зависимости, согласно которой если произойдет событие X, то спустя заданное время произойдет событие Y; анализ отклонений – выявление наиболее нехарактерных шаблонов.

Инструменты анализа

Данных становится всё больше и больше, поэтому сейчас как никогда важно иметь необходимый инструментарий для анализа данных и принятия решений. Выделим четыре популярных аналитических системы: MS Excel Power Query, MS Power BI, Pyramid Analytics, Компоненты аналитики MS SQL server (MDS, SSIS, SSAS).

Power Query

Power Query позволяет искать и открывать данные из различных

источников доступных онлайн и через корпоративные сети. Он умеет загружать в Excel данные разных типов, форматов и структур, а также из совершенно разных источников: из сети, из файла, из баз данных, из публичных источников данных и корпоративных репозиториев данных (встроена поддержка ETL), из ряда других источников: SharePoint List, OData feed, Active Directory, Facebook, etc.

Power Query обладает как положительными характеристиками, так и отрицательными. Из плюсов можно выделить: работает как с табличными моделями, так и с многомерными, умеет подключать дополнительные источники. Минусы данной программы: сложен в освоении, достаточно медлителен, нет возможности разделения доступа, ограничения на размер файлов/записей.

MS Power BI

Power BI – это инструмент создания интерактивных бизнес отчетов с возможностью совместной работы, визуализации и интерактивной работы.

Power BI обладает возможностью быстрой разработки информативных бизнес отчетов и панелей (в сети) – с возможностью взаимодействия и исследования данных. Также поддерживается автоматическое обновление BI-отчетов и визуализации при изменении данных, поддержка языка запросов, в том числе и Power Query, создание каталога данных с индексами для поиска, язык запросов близкий к естественному (для бизнес-аналитика) и возможность интерактивной работы. Поддерживается работа с мобильными устройствами.

Power BI является новым современным продуктом с дружелюбным интерфейсом и легким в освоении. Из недостатков можно выделить то, что решение «сырое» (некоторые компоненты могут работать нестабильно), оно не работает с OLAP кубами и имеет урезанный функционал в сравнении с конкурентами.

Pyramid Analytics

Pyramid Analytics – облачная платформа бизнес-аналитики, имеющая три ключевых компонента: интеллектуальный анализ данных, интерактивная

работа с данными и визуализацией, представление данных аудитории.

Платформа обладает возможностью совместной аналитики и моделирования данных, а также рядом других полезных возможностей: интеграция с R, работа с Big Data, интерактивная визуализация данных.

Продукт легок в освоении, работает с огромным количеством источников и имеет очень широкую функциональность. Главный и единственный недостаток – высокая цена.

Компоненты аналитики MS SQL server (MDS, SSIS, SSAS)

SQL Сервер позволяет проводить анализ внутри своей экосистемы. У него есть обширный набор компонент, сфокусируемся на трех наиболее известных.

Master Data Services — процессы и инструменты управления мастер-данными компании. Мастер-данные – это данные бизнеса: о клиентах, продуктах, услугах, персонале, технологиях, материалах и т.д.

SQL Server Integration Services — миграцияиинтеграцияданных.

SQL Server Analysis Services - OLAP и data mining внутри SQL сервера.

Business Intelligence vs. Data Science

Business Intelligence отвечает на вопросы: что произошло в прошлом квартале? как много товара мы продали? в чем проблема? в какой ситуации? т.е. говорит в основном о том, что было в прошлом и анализирует исторические данные, полученные на текущий момент.

BI оперирует структурированными данными, традиционными источниками и управляемыми наборами данных.

Data Science отвечает на вопросы: а что, если…? какой оптимальный сценарий для нашего бизнеса? что будет дальше? т.е. ставит цель предугадать, сделать прогноз. Области применения – оптимизация, прогнозирование, статистический анализ.

DS работает с большими структурированными и неструктурированными наборами данных.

Понятие жизненного цикла аналитики данных

Жизненный цикл аналитики данных – это последовательность действий, которую нужно выполнить на наборе входных данных для эффективного достижения цели аналитики с помощью выбранных методов анализа. Жизненный цикл аналитики данных может включать в себя: выявление проблем анализа данных, сбор набора данных, проектирование, анализ данных, визуализацию данных.

Для начала необходимо понять, какова реальная конечная цель проекта, какую пользу он может принести, оценить его критерии успешности или провала, выбрать технологии для сбора, трансформации и анализа данных. Нужно постараться ответить на такие дополнительные вопросы: достаточно ли у вас ресурсов на реализацию проекта? с кем вы будете контактировать по ходу проекта? достаточно ли у вас входных данных? были ли уже в компании попытки анализа? достаточно ли у вас времени на реализацию проекта? с какими проблемами вы можете столкнуться?

В этап выявления проблемы (изучения) входят также такие пункты: определение набора необходимых знаний, которые нужны для ориентации в предметной области; наличие доступных ресурсов (люди, инструменты, данные); структурирование данных с точки зрения аналитики; изучение истории бизнеса.

После изучения следует перейти к подготовке данных. Определите используемые программные средства, то есть выберите ПО, БД, инструменты анализа и визуализации. Получите, очистите и загрузите данные в систему, после чего оцените количество и качество данных.

На этапе проектирования модели нужно определить методы, технологии, рабочие процессы, необходимые для расчета модели и объем данных. Также важно определить корреляцию между переменными: столбцами таблиц и полями данных.

Построение модели заключается в том, чтобы разработать наборы данных для тестирования, обучения и производства. Затем оценить жизнеспособность и надежность данных для использования в модели. И в

конечном итоге выбрать рабочее окружение, то есть аппаратные и программные средства, на которых можно настроить процесс.

В результате предыдущего шага получились некие результаты. Теперь необходимо определить, удалось ли вам достичь результата на основе критериев проекта, а также определить ключевые результаты исследования. В зависимости от целевой аудитории можно разработать диаграммы и графики и сформулировать итоги и рекомендации.

Последний этап – практическая реализация. В процессе этого этапа происходит доставка финальных рекомендаций, отчетов, кода и технических документов, запуск пилотного проекта, реализация модели в производственной среде и интеграция аналитических оценок в панель управления или в операционной системе.

Начиная работать с большими данными, многие сталкиваются с трудностями, которые можно избежать, выполняя простейшие советы, такие как: всегда проводить глубокое изучение предметной области, разбивать задачу на более мелкие, делать аналитику гибкой и масштабируемой, предусмотреть возможность повторения каждого из этапов с возможностью внесения изменений в предыдущий этап, быть готовым к негативному результату, внимательно оценивать затраченное время на каждый этап.

Контрольные вопросы для повторения и самопроверки по теме №2:

1. Что является главным результатом процесса Business Intelligence?
2. Что означает термин «Business Intelligence» в информационных технологиях?
3. Расшифруйте аббревиатуру OLAP.
4. Что относится к средствам предоставления информации в Business Intelligence?
5. Что относится к средствам интеграции в «Business Intelligence»?
6. Какие цели ставит перед собой Data Science?
7. Что такое жизненный цикл аналитики данных?

8. Дайте определение термину «предиктивное моделирование»?
9. Что такое ETL?
10. Какова роль BI-аналитика в проекте?

Литература по теме №2:

1. Майер-Шенбергер В., Кукьер К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим. Language Arts & Disciplines – 2013. – 599с.
2. Силен Д., Мейсман А., Али М. Основы DataScience и BigData. Python и наука о данных. СПб: Питер. – 2017. – 336с.

Глоссарий по теме №2:

BI – это методы и инструменты для перевода необработанной информации в осмысленную, удобную форму

Data Science – раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме.

Хранилище данных – предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчётов и бизнес-анализа.

Интеллектуальный анализ данных – компьютерная техника извлечения знаний, которая использует искусственный интеллект для распознавания образов и выделения значимых закономерностей из данных, находящихся в хранилищах или во входных / выходных потоках.

Data Discovery – подход к созданию аналитических решений.

Жизненный цикл аналитики данных – последовательность действий, которую нужно выполнить на наборе входных данных для эффективного достижения цели аналитики с помощью выбранных методов анализа.

Предиктивное моделирование – процесс создания (или выбора) модели для предсказания вероятности наступления некоторого события.

Мастер-данные – это данные бизнеса: о клиентах, продуктах, услугах, персонале, технологиях, материалах и т.д.

BI-платформа – программа, которая предоставляет пользователю удобные инструменты анализа фактически любых данных.

ТЕМА №3: ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

Hadoop

Hadoop – каркас с открытым исходным кодом, предназначенный для создания и запуска распределённых приложений, обрабатывающих большие объёмы данных.

Hadoop обладает важными отличительными особенностями:

- **Надёжность**: архитектура разработана с учётом возможности частых отказов, содержит несколько копий данных. Это позволяет избежать потерю данных.
- **Доступность:** Hadoop работает на крупных кластерах, собранных из стандартных компьютеров
- **Масштабируемость:** при увеличении объема данных достаточно добавить новые узлы в кластер
- **Простота:** Hadoop позволяет пользователю быстро создавать эффективный параллельный код
- **Эффективность:** параллельная обработка данных на множестве компьютеров
- **Кроссплатформенность:** систему можно развернуть на базе любой операционной системы с виртуальной машиной Java (JVM)

История Hadoop и Map Reduce

Создателем Hadoop является Дуг Каттинг. Проект Hadoop начал свое существование в рамках Apache Nutch – системы веб-поиска с открытым исходным кодом. Hadoop назван в честь плюшевого жёлтого слона сына Дуга. Жёлтый слон также является логотипом системы Hadoop (рисунок 3.1). Подпроекты данной системы также получили названия, связанные с темой животных. Для более мелких компонентов выбор названия делался в пользу

содержательности и простоты. Название само говорит, какие функции выполняет определеный компонент. Так, например, Job Tracker занимается отслеживанием заданий Map Reduce.

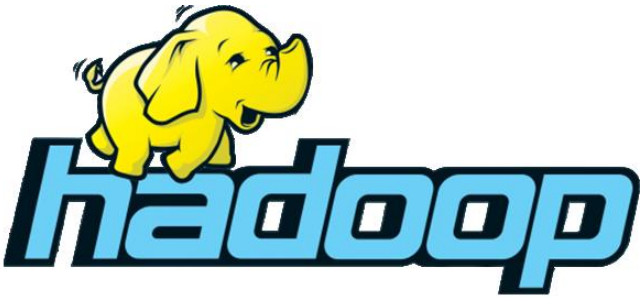


Рисунок 3.1 – Логотип Hadoop

Запуск проекта Nutch произошёл в 2002 году. Поисковая система была запущена в кратчайшие сроки. Вскоре разработчики пришли к выводу, что данную архитектуру не представляется возможным масштабировать на миллиарды web-страниц. Решение проблемы было найдено в 2003 году, когда компания Google опубликовала статью, в которой описывалась архитектура распределённой файловой системы – Google File System(GFS). Такая архитектура была способна решить проблему хранения огромных файлов, которые генерировались в процессе индексирования и обхода. Вдобавок использование данной распределённой системы сэкономило бы время, затрачиваемое на управление узлами хранения данных. Реализация данной системы началась через год, в 2004 году. Система получила название Nutch Distributed Filesystem (NDFS). Ее исходный код является открытым. В этом же году компания Google представила технологию Map Reduce. В 2005 году разработчики Nutch представили работоспособную реализацию Map Reduce. К середине года была выполнена адаптация существующих алгоритмов Nutch для предоставления возможности использования Map Reduce и NDFS. Hadoop был образован в феврале 2006 года, как подпроект Lucene.

Тогда же Дуг Каттинг начал работать в компании Yahoo!. Компания Yahoo! специально выделила группу и ресурсы для развития системы Hadoop. В феврале 2008 года Yahoo продемонстрировала поисковый индекс,

сгенерированный 10000-ядерным кластером Hadoop.

В начале 2008 года Hadoop стал одним из ведущих проектов Apache. Данная технология стала использоваться не только в Yahoo, но в таких компаниях, как Facebook, Last.fm и New York Times.

В апреле 2008 года был обновлен мировой рекорд по скорости сортировки терабайта данных с применением технологии Hadoop. Время сортировки составляло 209 секунд (предыдущий рекорд – 297 секунд). Кластер состоял из 910 узлов.

В ноябре того же года компания Google представила собственную реализацию Map Reduce, способную отсортировать 1 терабайт за 68 секунд.

После этого технология Hadoop стремительно вошла в повседневную деятельность крупных компаний. Данная технология выполняла роль платформы хранения и анализа данных общего назначения. В настоящее время существуют дистрибутивы Hadoop как от крупных общепризнанных фирм, включая EMC, IBM, Microsoft и Oracle, так и от компаний с узкой специализацией – таких, как Cloudera, Hortonworks и MapR.

HadoopDistributedFileSystem

Hadoop поставляется с распределенной файловой системой, которая называется HDFS (Hadoop Distributed File System). Данная файловая система спроектирована для хранения очень больших файлов с потоковой схемой доступа к данным в кластерах обычных машин. Большие файлы в данном случае - это файлы, имеющие размер сотни мегабайт, гигабайт и терабайт.

В основу HDFS заложена концепция однократной записи/многократного чтения как самая эффективная схема обработки данных. Набор данных обычно генерируется или копируется из источника, после чего с ним выполняются различные аналитические операции. В каждой операции задействуется большая часть набора данных (или весь набор), поэтому время чтения всего набора данных важнее задержки чтения первой записи.

Hadoop не требует дорогостоящего оборудования высокой надежности. Система спроектирована для работы на стандартном оборудовании с

достаточно высокой вероятностью отказа отдельных узлов в кластере. Технология HDFS спроектирована таким образом, чтобы в случае отказа система продолжала работу без заметного прерывания.

В HDFS существует концепция блока, размер которого по умолчанию составляет 64 Мбайт. Файлы HDFS разбиваются на блочные фрагменты, которые хранятся независимо друг от друга.

Абстракция блоков в распределенной файловой системе имеет несколько преимуществ:

- Размер файл может превышать размер отдельного диска в сети. Блоки файла имеют возможность использования любых дисков в кластере. Кроме того, когда файл хранится в кластере HDFS, блоки данного файла можно распределить по всем дискам кластера.
- Блоки упрощают подсистему хранения. Блоки имеют фиксированный размер. Это позволяет системе вычислить какое количество блоков хранится на определенном диске. Также блоки предотвращают проблемы, связанные с хранением метаданных.
- Блоки подходят для репликации. За счет блоков происходит улучшение отказоустойчивости и доступности системы. Каждый блок имею несколько копий на отдельных машинах (по умолчанию 3). В случае недоступности какого-либо блока происходит считывание копии данного блока из другого доступного места.

Кластер HDFS имеет 2 вида узлов (рисунок 3.2): Name Node (узел имен) и Data Node (узел данных). Name Node является управляющим узлом, а Data Node – управляемым.

Name Node занимается управлением пространством узлах имен файловой системы, поддержкой метаданных и каталогов в дереве файловой системы. Локальный диск хранит информацию в виде 2 файлов: журнала изменений и образа файловой системы. Name Node содержит информацию о том, на каких именно узлах находятся все блоки файла, при этом информация о местонахождении блоков строится заново при запуске системы, а не хранится

постоянно.

Для обращения к файловой системе клиент связывается с Name Node и Data Node узлами. Интерфейс файловой системы схож с интерфейсом Portable Operating System Interface. Таким образом пользовательский код может не обладать никакой информацией об узлах Name Node и Data Node.

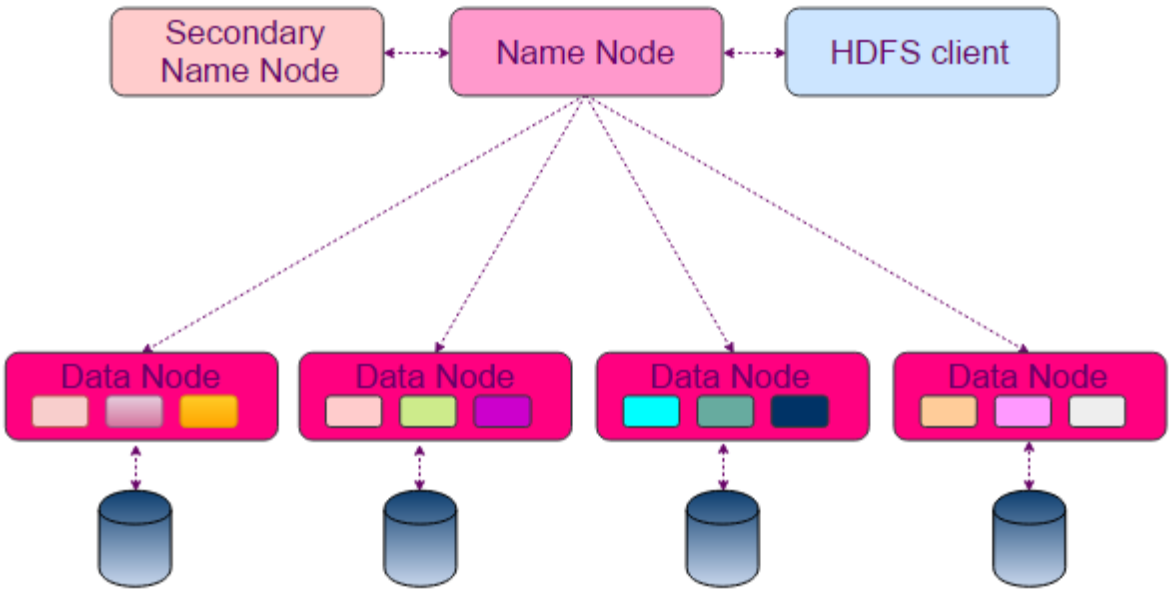


Рисунок 3.2 – Архитектура HDFS

Узлы данных выполняют запись и чтение блоков, когда этого требуют клиенты или узел имен. Кроме того, данные узлы периодически передают список сохраняемых ими блоков непосредственно узлу имен. Узлы, будучи частью системы, напрямую влияют на работоспособность файловой системы. В случае ликвидации машины, на которой работает Name Node, произойдёт потеря всех данных. Именно по этой причине устойчивость узла имен к возможным сбоям важна.

Hadoop предоставляет два способа для осуществления устойчивости:

1. **Архивация файлов**, которые определяют устойчивое состояние метаданных файловой системы. Имеется возможность настроить Hadoop так, чтобы Name Node был способен на запись своего устойчивого

состояния в несколько файловых систем. Операции записи выполняются синхронно и атомарно.

2. **Запуск вторичного узла имен**, не выполняющего функции узла имен. Его основной функцией является периодическое слияние образа пространства имен с журналом изменений во избежание чрезмерного разрастания последнего. Вторичный узел имен обычно работает на отдельной физической машине, так как для того, чтобы выполнить слияние, требуются значительная вычислительная мощность и такое же количество памяти, как на узле имен. Вторичный узел имен имеет копию объединенного образа пространства имен, который может быть использован в случае сбоя основного узла. Однако состояние вторичного узла имен немного отстает от состояния основного узла. При полном отказе основного узла не является возможным предотвратить потерю данных.

Технология Map Reduce

Map Reduce – технология распределённых вычислений. Основными целями Map Reduce являются разделение логики приложения и организация распределённого взаимодействия.

Программист реализует только логику приложения. Распределённая работа в кластере обеспечивается автоматически. Map Reduce работает с данными как с парами **ключ - значение,** например:

- смещение в файле: текст
- идентификатор пользователя: профиль
- пользователь: список друзей
- временная метка: событие в журнале

Работа Map Reduce производится в 2 этапа (рисунок 3.3):

1. **Map:** выполняется предварительная обработка входных данных. Главный узел разделяет полученные данные на части и передаёт их рабочим узлам. При этом порождаются пары ключ -значение. Как ключи, так и значения могут быть составными.

2. **Reduce:** происходит свёртка заранее обработанных данных. Рабочие узлы отправляют ответы главному узлу, а он на основе этих данных формирует решение поставленной задачи.

Операции Map и Reduce могут выполняться распределённым образом.

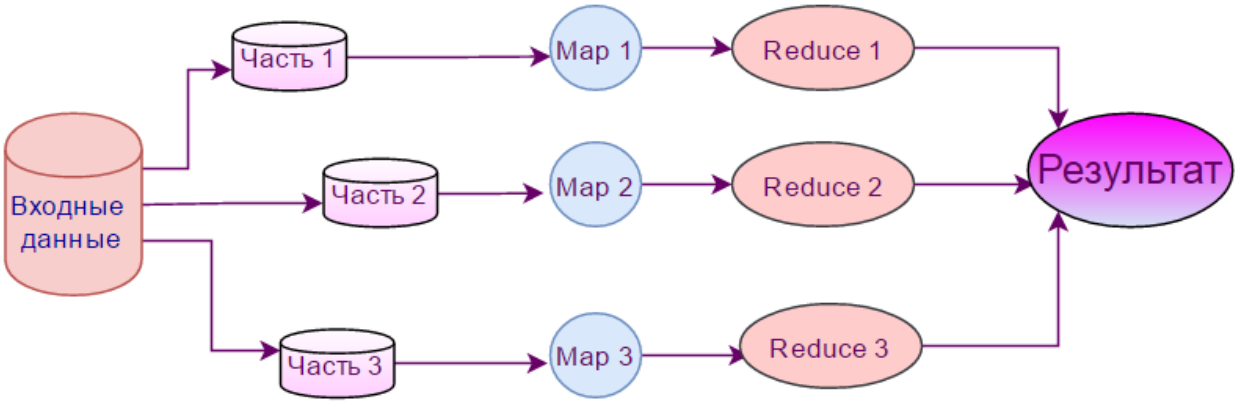


Рисунок 3.3 – Схема работы Map Reduce

Примеры применения Map Reduce

- Масштабный статистический анализ и моделирование
- Анализ и индексация данных
- Построение масштабируемых алгоритмов машинного обучения
- Сбор данных DNS по всему миру для обнаружения сетей распределения контента и проблем с конфигурацией
- Построение карты всей сети Интернет
- Распределенный grep
- Сортировка
- Поиск web-страниц

Достоинства модели Map Reduce

- Автоматическое распараллеливание: функции Map и Reduce могут выполняться параллельно и независимо друг от друга
- Масштабируемость: данные могут располагаться и обрабатываться на разных серверах
- Отказоустойчивость: при отказе сервера функции Map и Reduce

запускаются на другом сервере.

Недостатки модели Map Reduce

- Фиксированный алгоритм обработки данных
- Высокие накладные расходы на распараллеливание

Архитектура Hadoop

На каждом компьютере должны присутствовать два компонента (рисунок 3.4):

- Task Tracker: выполняет маленькую вычислительную задачу на узле
- Data Node: управляют данными, переданными на узел.

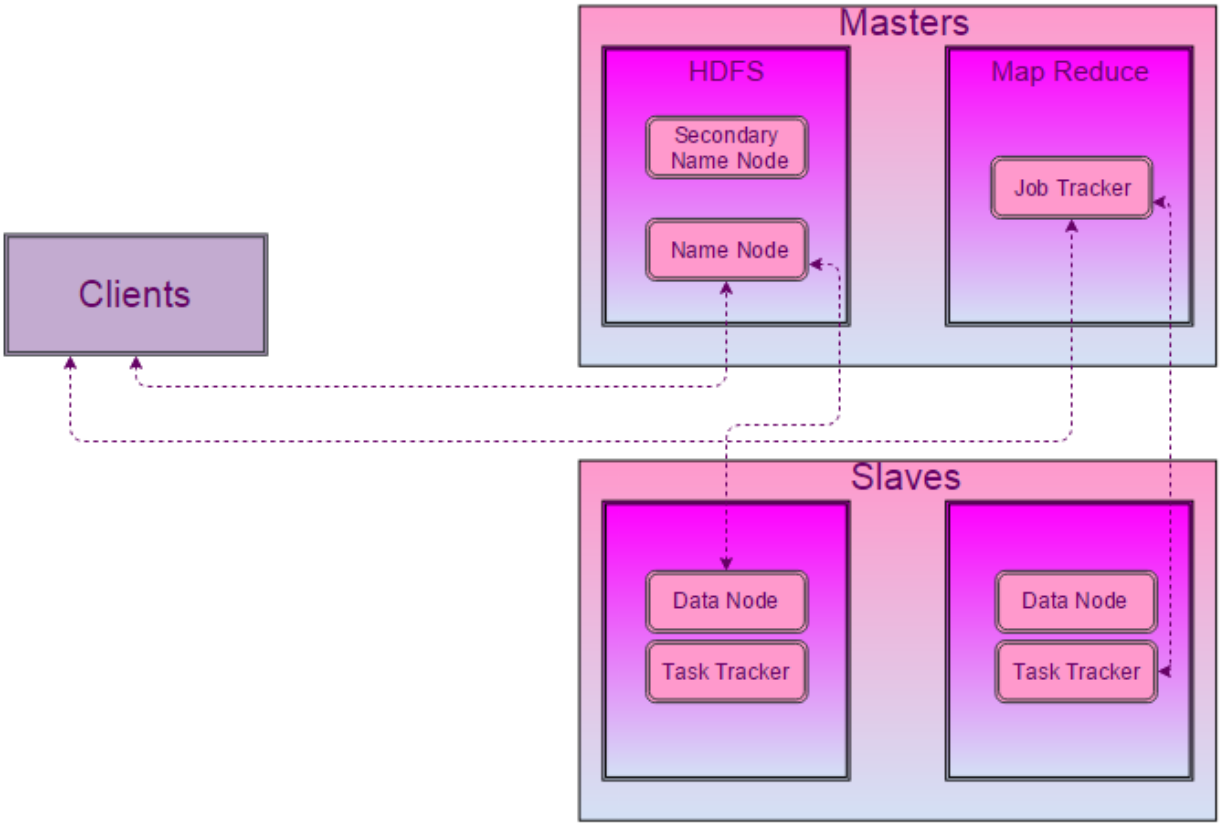


Рисунок 3.4 – Архитектура Hadoop

Все маленькие компьютеры называют Slave. Также существует Master узел. Данный узел кроме Task Tracker и Data Node содержит ещё Job Tracker и Name Node. Task Tracker и Job Tracker являютсякомпонентами Mad Reduce, а Name Node и Data Node – HDFS. Роль и цель Job Tracker, которые исполняется

на Master Node - это разбить большие задачи на маленькие кусочки и отправить каждую маленькую часть вычислений соответствующему Task Tracker на узле, которые в свою очередь, когда смогут закончить вычисления, отправят обратно информацию в Job Tracker, который объединит все результаты вычислений и отправит их обратно приложению.

Hadoop MapReduce 1.0

В Hadoop MapReduce 1.0 входит единственный узел Job Tracker. Данный узел занимается распределением задач по многочисленным узлам Task Tracker, которые выполняют эти задачи (рисунок 3.5).

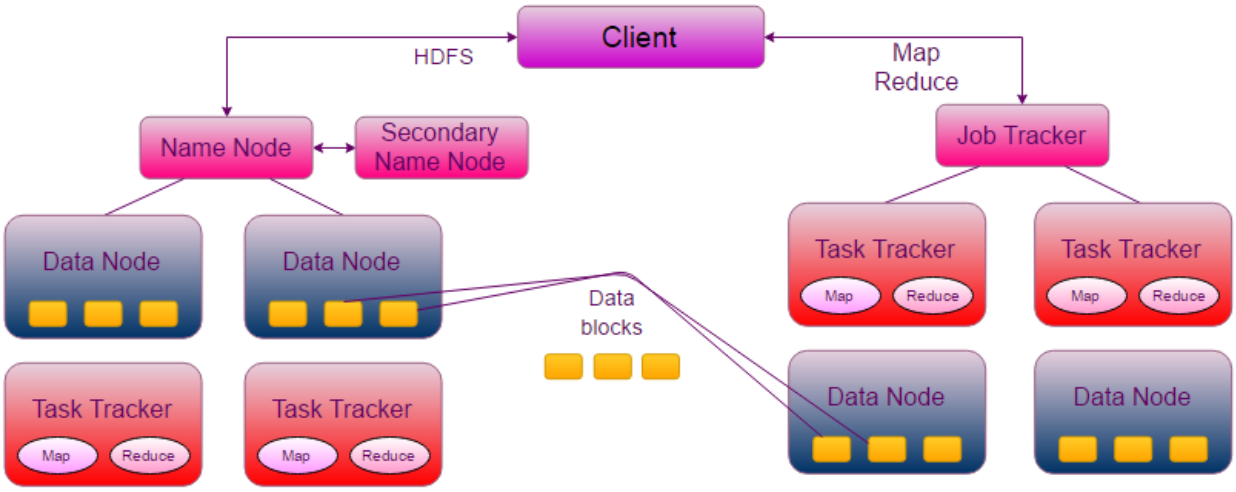


Рисунок 3.5 – Hadoop Map Reduce 1.0

При таком подходе происходит перегрузка CPU и сети. Большую часть времени Job Tracker затрачивает на поддержку жизненного цикла Map Reduce задач. Загрузка сети происходит из-за большого централизованного трафика.

В Hadoop MapReduce 1.0 при сбое JobTracker возникает необходимость перезапуска JobTracker с чтением состояния из специальных журналов. Это приводит к простою кластера. Максимальная масштабируемость находится в пределах 4000 узлов. Map Reduce 1.0 не позволял запускать на Hadoop кластере MPI распределённых алгоритмов из-за сильной связанности фреймворка распределённых вычислений и библиотек, выполняющих распределённый

алгоритм. В первой версии размер файла по умолчанию составляет 64мб.

Hadoop Map Reduce 2.0

В Map Reduce 2.0 ответственность по управлению ресурсами и координации за жизненным циклом выполнения приложений разделена между Application Master и Resource Manager (рисунок 3.6).

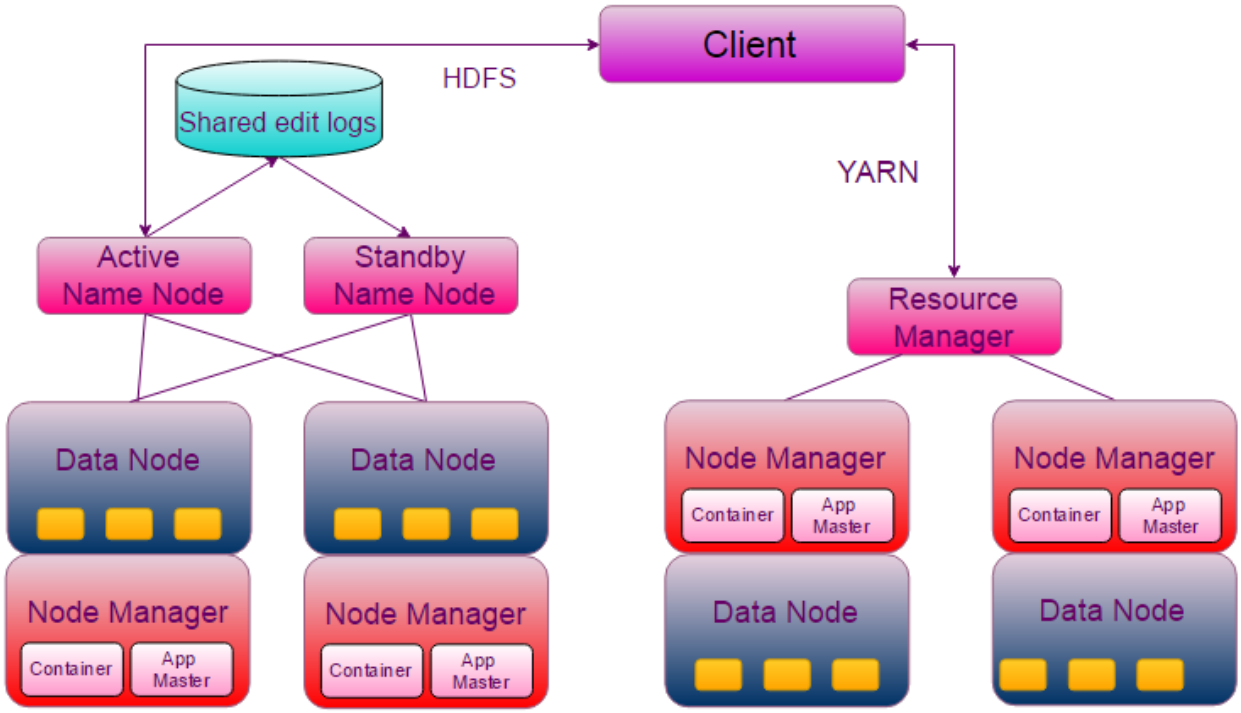


Рисунок 3.6 – Hadoop Map Reduce 2.0

Каждый узел вычисления разбит на произвольное количество контейнеров (Container). Каждый контейнер содержит определённое количество ресурсов. За контейнером следит Node Manager. В Hadoop MapReduce 2.0 сохраняется состояние компонентов Resource Manager и Application Master и обеспечивается система автоматического перезапуска перечисленных компонентов при сбое с подгрузкой последнего успешно сохраненного состояния. Теоретически Map Reduce 2.0 может содержать до 10000 вычислительных узлов. Во второй версии был выделен фреймворк распределенных вычислений YARN и фреймворк, базирующийся на основе YARN, – MR2. YARN не зависит от специфики распределённого алгоритма в

отличие от MR2.

YARN предоставляет компоненты и API, необходимые для разработки распределенных приложений различных типов, также он берет на себя ответственность по распределению ресурсов в ответ на запросы ресурсов от выполняемых приложений и ответственность за отслеживанием статуса выполнения приложений.

Благодаря YARN на Hadoop-кластере возможно запускать не только «map/reduce»-приложения, но и распределенные приложения, созданные с использованием: Open MPI, Spark, Apache HAMA, Apache Giraph и др. Есть возможность реализовать и другие распределенные алгоритмы.

Варианты использования Hadoop

Существует несколько вариантов использования Hadoop:

- Автономный режим (Standalone Mode) – используется для того, чтобы на локальной машине использовать Hadoop как кластер, состоящий из одного узла
- Псевдо-распределенный режим (Pseudo Distributed Mode) – поддержка работы с HDFS, но на одном узле
- Полностью распределенный режим (Fully Distributed Mode) - используется на реальном кластере

Экосистема Hadoop

Экосистема Hadoop постоянно расширяется новыми продуктами и инструментами. Рассмотрим основные компоненты, которые составляют данную экосистему.

HBase: представляет собой NoSQL базу данных. HBase обеспечивает отказоустойчивый способ хранения данных огромных объёмов. Данная база данных работает поверх распределённой файловой системы HDFS. Проект реализован на языке Java. Первая рабочая версия появилась в октябре 2007 года.

Hive: представляет собой систему управления базами данных. Hive является надстройкой для Hadoop. Система Hive позволяет хранить данные в

виде таблиц, выполнять SQL подобные запросы и анализировать данные, которые хранятся в Hadoop. Данный проект был разработан компанией Facebook и передан Apache. Hive поддерживает Hive Query Language (HiveQL). Данный язык запросов основан на SQL. HiveQl имеет функции для работы с XML и JSON форматами, а также поддерживает нескалярные типы данных.

Pig: является высокоуровневым процедурным языком. Pig используется для выполнения запросов к слабоструктурированными данным больших объёмов. Первая реализация Apache появилась в 2008 году.

Mahout: представляет собой набор библиотек для выполнения машинного обучения. Данный проект реализует такие области машинного обучения, как рекомендательные системы, кластеризация и классификация.

HUE: представляет собой web-интерфейс к Hadoop. Hadoop User Experience позволяет выполнять мониторинг задач кластера и облегчить его обслуживание.

Sqoop: представляет собой инструмент для выполнения передачи данных между Hadoop и реляционными базами данных. Благодаря использованию технологии Map Reduce обеспечивается параллельная работа и отказоустойчивость при экспорте и импорте данных.

Avro: представляет собой систему для выполнения сериализации данных со встроенной схемой. Данная система выполняет обмен двоичными данными для сериализации.

ZooKeeper: представляет собой распределённый сервис синхронизации и конфигурирования, является key/value хранилищем.

Flume: представляет собой инструмент для управления потоками данных. Первая версия появилась в 2009 году. Flume предназначен для сбора данных из различных источников и направления этих же данных в централизованный центр. Данный инструмент обладает простой и гибкой архитектурой.

Spark

Apache Spark предназначен для реализации распределённой обработки

слабоструктурированных и неструктурированных данных. Проект позиционируется как инструмент для «молниеносных кластерных вычислений».

Spark состоит из ядра и нескольких расширений:

- **Spark SQL**: поддерживает запросы данных либо при помощи SQL, либо посредством Hive Query Language. Библиотека возникла как порт Apache Hive для работы поверх Spark (вместо Map Reduce), а сейчас уже интегрирована со стеком Spark. Она не только обеспечивает поддержку различных источников данных, но и позволяет переплетать SQL-запросы с трансформациями кода; получается очень мощный инструмент.
- **Spark Streaming**: поддерживает обработку потоковых данных в реальном времени; такими данными могут быть файлы логов рабочего веб-сервера (напр. Apache Flume и HDFS/S3), информация из социальных сетей, например, Twitter, а также различные очереди сообщений, например, Kafka. Spark Streaming получает входные потоки данных и разбивает данные на пакеты. Далее они обрабатываются движком Spark, после чего генерируется конечный поток данных (также в пакетной форме).
- **Spark MLlib:** библиотека для машинного обучения, предоставляющая различные алгоритмы, разработанные для горизонтального масштабирования на кластере в целях классификации, регрессии, кластеризации, совместной фильтрации и т.д. Некоторые из этих алгоритмов работают и с потоковыми данными, например, линейная регрессия с использованием обычного метода наименьших квадратов или кластеризация по методу k-средних.
- **GraphX:** библиотека для манипуляций над графами и выполнения с ними параллельных операций. Библиотека предоставляет универсальный инструмент для ETL, исследовательского анализа и итерационных вычислений на основе графов. Кроме встроенных операций для манипуляций над графами здесь также предоставляется библиотека обычных алгоритмов для работы с графами, например, PageRank.

Контрольные вопросы для повторения и самопроверки темы №3

1. Что такое Apache Hadoop?
2. В чем преимущества решений на базе Hadoop?
3. Что такое MapReduce?
4. Какими достоинствами и недостатками обладает MapReduce?
5. Какому основному принципу следует HDFS?
6. Какой размер блока по умолчанию в HDFS?
7. Какие функции выполняет NameNode в HDFS?
8. Какой узел отвечает за репликацию данных в Hadoop?
9. Какие компоненты содержит Slave узел в Hadoop?
10. Какие компоненты содержит Master узел в Hadoop?
11. Какие компоненты являются частями HDFS?
12. Какое API было добавлено в Hadoop v2.0?
13. Для чего используется автономный режим Hadoop?
14. Какой режим необходим для того, чтобы на локальной машине использовать Hadoop как кластер, состоящий из одного узла?

Литература по теме №3

1. Уайт Т. Hadoop: Подробное руководство. СПб: Питер. – 2013. – 672с.
2. Лэм Ч. Hadoop в действии. Москва: ДМК Пресс. – 2012. – 426с.
3. Майер-Шенбергер В., Кукьер К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим. Language Arts & Disciplines – 2013. – 599с.
4. Силен Д., Мейсман А., Али М. Основы DataScience и BigData. Python и наука о данных. СПб: Питер. – 2017. – 336с.

Глоссарий по теме №3

Hadoop – открытая реализация MapReduce, предназначенная для создания и запуска распределённых приложений, обрабатывающих большие объёмы

данных.

Map Reduce – модель распределенных вычислений, предназначенная для параллельных вычислений над очень большими (до нескольких петабайт) объемами данных.

Apache Spark –проект Apache, который позиционируется как инструмент для «молниеносных кластерных вычислений».

GraphX –библиотека для манипуляций над графами и выполнения с ними параллельных операций.

HDFS (Hadoop Distributed File System) – распределенная файловая система в основе Hadoop.

YARN (Yet Another Resource Negotiator) – модуль, появившийся в Hadoop 2.0, отвечающий за управление ресурсами кластеров и планирование заданий.

Автономный режим Hadoop –использование Hadoop на локальной машине в качестве кластера, состоящего из одного узла.

Псевдо распределенный режим – поддержка работы с HDFS на одном узле.

Полностью распределенный режим – режим для работы на реальном кластере.

Hive – распределенное хранилище данных, управляющее данными, хранимыми в HDFS, и предоставляющее язык запросов на базе SQL для работы с этими данными.

HBase – нереляционная распределенная база данных, исполняемая в HDFS.

Mahout – набор библиотек для выполнения машинного обучения.

Pig – язык управления потоком данных и исполнительная среда для анализа больших объемов данных.

Oozie – сервис для записи и планировки заданий Hadoop.

Flume – распределенный сервис для коллекционирования, сбора и перемещения больших массивов данных.

Sqoop – инструмент для пересылки данных между структурированными хранилищами и HDFS.

Avro – система сериализации для выполненных межъязыковых вызовов и

долгосрочного хранения данных.

ZooKeeper – распределенный координационный сервис, предоставляющий примитивы для построения распределенных приложений.

Ядро Spark – базовый элемент для крупномасштабной параллельной и распределенной обработки данных.

SparkSQL – компонент Spark, поддерживающий запросы данных при помощи SQL или посредством Hive Query Language.

ТЕМА №4: МАСШТАБИРОВАНИЕ И МНОГОУРОВНЕВОЕ ХРАНЕНИЕ ДАННЫХ

NoSQL. Not Only SQL

NoSQL (англ. Not Only SQL, не только SQL) – термин для ряда подходов, преследующих цель решить такие проблемы, как масштабируемость, доступность, устойчивость к разделению и согласованность данных.

В качестве одного из методологических обоснований подхода NoSQL используется эвристический принцип (теорема CAP), утверждающий, что в распределённой системе невозможно одновременно обеспечить согласованность данных, доступность и устойчивость к расщеплению распределённой системы на изолированные части.

Таким образом, при необходимости достижения высокой доступности и устойчивости к разделению предполагается не фокусироваться на средствах обеспечения согласованности данных, обеспечиваемых традиционными SQL-ориентированными СУБД с транзакционными механизмами на принципах ACID.

Масштабируемость

Целью является достижение линейности масштабируемости при большом числе узлов, потому как горизонтальное масштабирование, применяемое на традиционных базах данных, обычно является трудоемким и дорогостоящим удовольствием.

Существуют два вида масштабируемости:

Вертикальное масштабирование: масштабируемость в этом контексте означает возможность заменять в существующей вычислительной системе компоненты более мощными и быстрыми по мере роста требований и развития технологий. Увеличение производительности каждого компонента системы с целью повышения общей производительности. Это самый простой способ масштабирования, так как не требует никаких изменений в прикладных программах, работающих на таких системах.

Горизонтальное масштабирование: масштабируемость в этом контексте означает возможность добавлять к системе новые узлы, серверы, процессоры для увеличения общей производительности. Разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам) и (или) увеличение количества серверов, параллельно выполняющих одну и ту же функцию. Этот способ масштабирования может

требовать внесения изменений в программы, чтобы программы могли в полной мере пользоваться возросшим количеством ресурсов.

Для области Big Data интересно горизонтальное масштабирование, так как оно распределенное.

Репликация

Репликация — это копирование данных на другие узлы при обновлении, позволяющее создать полный дубликат базы данных так, что вместо одного сервера их будет несколько, повысить отказоустойчивость, снизить нагрузку на каждый отдельный сервер и как добиться большей масштабируемости, так и повысить доступность и сохранность данных. Включает в себя два вида:

- master-slave
- peer-to-peer

Репликация Master-Slave (рисунок 4.1):

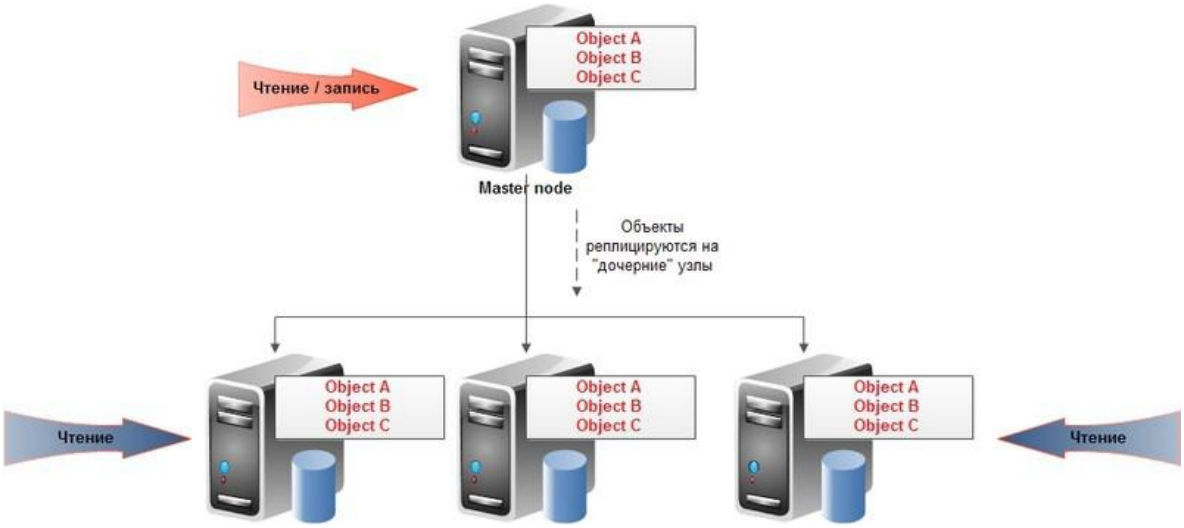


Рисунок 4.1 – Репликация Master-Slave

Master – это основной сервер БД, куда поступают все данные и где происходят изменения в данных (добавление, обновление, удаление).

Slave – это вспомогательный сервер БД, копирующий все данные с мастера, предназначен только для чтения и может быть в количестве больше одного.

Потому как операций чтения (SELECT) данных часто намного больше, чем операций изменения данных (INSERT/UPDATE), то репликация позволяет таким образом разгрузить основной сервер за счет переноса операций чтения на слейв, потому что в его использовании два или больше одинаковых серверов вместо одного.

Репликация Peer-to-peer (рисунок 4.2):

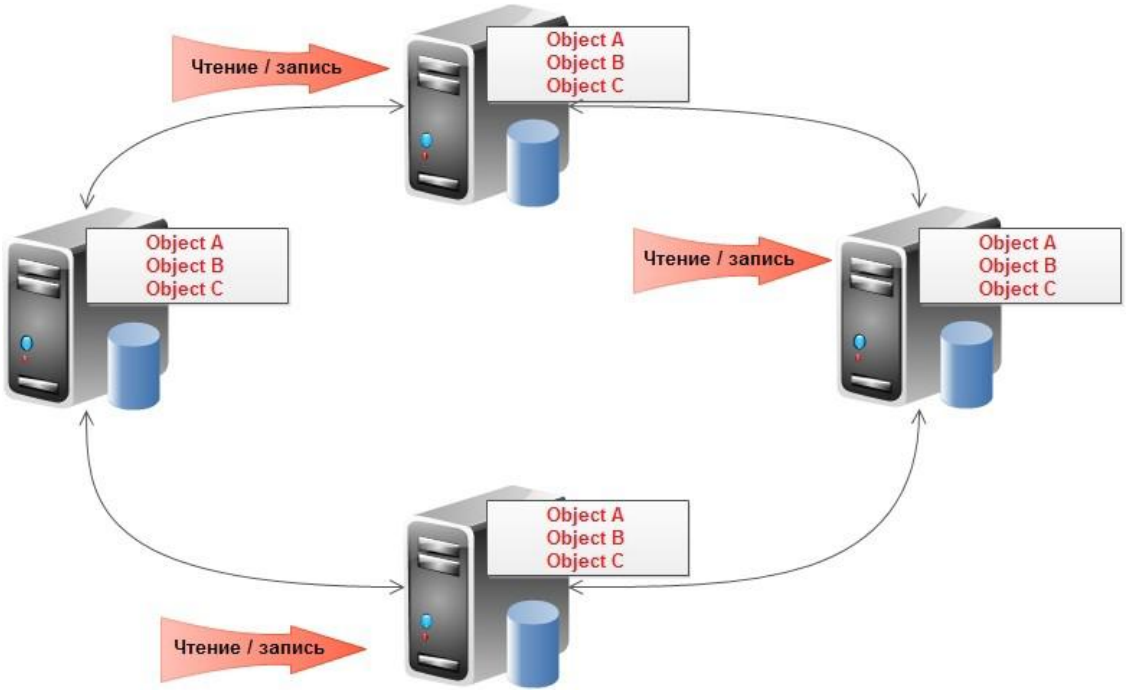


Рисунок 4.2 – Репликация Peer-to-peer

Одноранговая сеть – это сеть, которая основывается на равноправии всех участников сети. В данном виде сети отсутствует главный сервер, и каждый узел является как клиентом, так и выполняет функции сервера.

Сама по себе репликация не самый удобный механизм масштабирования, так как происходит рассинхронизация данных и возможны задержки в копирования с мастера на слейв. Но вместе с тем это отличное средство для обеспечения отказоустойчивости, потому как всегда можно переключиться на слейв в случае отказа мастера. Для надежности обычно используется совместно с шардингом.

Шардинг (рисунок 4.3):

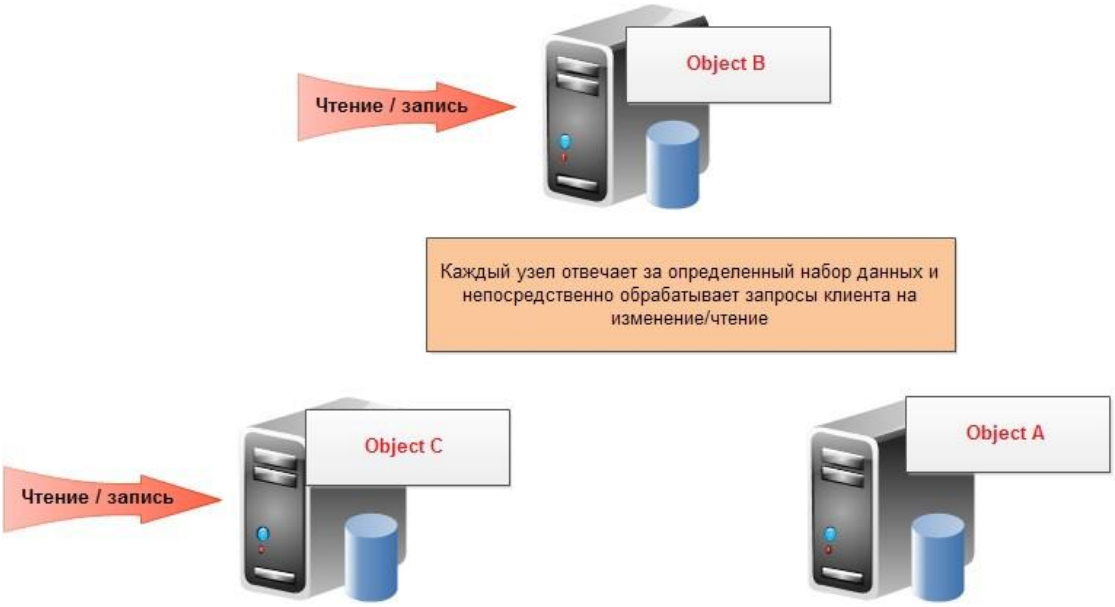


Рисунок 4.3 – Шардинг

Шардинг –это другая техника масштабирования работы с данными, суть которой заключается в разделении базы данных на отдельные части так, чтобы каждую из них можно было вынести на отдельный сервер. Этот процесс зависит от структуры базы данных и выполняется прямо в приложении, в отличие от репликации. Существуют два типа шардинга:

Вертикальный шардинг – это выделение таблицы или группы таблиц на отдельный сервер.

Горизонтальный шардинг – это разделение одной таблицы на разные сервера. Это необходимо использовать для огромных таблиц, которые не умещаются на одном сервере. Разделение таблицы на куски делается по такому принципу:

1. На нескольких серверах создается одна и та же таблица (только структура, без данных).

2. В приложении выбирается условие, по которому будет определяться нужное соединение (например, четные на один сервер, а нечетные – на другой).

3. Перед каждым обращением к таблице происходит выбор нужного соединения (допустим, для нечетных пользователей мы будем работать с первым сервером, а для четных – со вторым).

CAP – теорема

Теорема CAP (известная также как теорема Брюера), – эвристическое утверждение о том, что в любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств (рисунок 4.4):

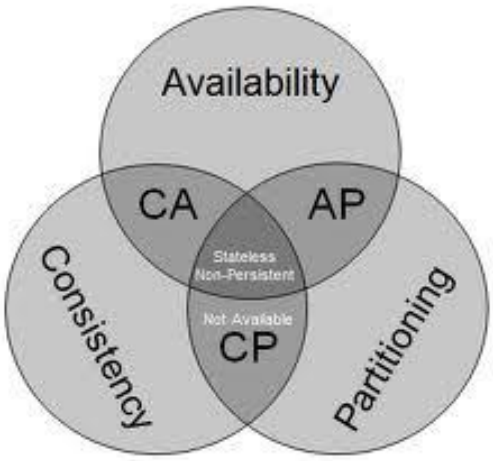


Рисунок 4.4 – Теорема CAP

1. Согласованность данных (англ. **C**onsistency) – во всех вычислительных узлах в один момент времени данные не противоречат друг другу. Как только мы успешно записали данные в наше распределенное хранилище, любой клиент при запросе получит эти последние данные.

2. Доступность (англ. **A**vailability) – любой запрос к распределённой системе завершается корректным откликом. В любой момент клиент может получить данные из нашего хранилища или получить ответ об их отсутствии, если их никто еще не сохранял.

3. Устойчивость к разделению (англ. **P**artition tolerance) – расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций. Потеря сообщений между компонентами системы (возможно даже потеря всех сообщений) не влияет на работоспособность системы. Здесь очень важный момент состоит в том, что если какие-то компоненты выходят из строя, то это тоже подпадает под этот случай, так как можно считать, что данные компоненты просто теряют связь со

всей остальной системой.

Следствие из теоремы CAP

CA: Система, во всех узлах которой данные согласованы и обеспечена доступность, жертвует устойчивостью к распаду на секции. Такие системы возможны на основе технологического программного обеспечения, поддерживающего транзакционность в смысле ACID. Примерами таких систем могут быть решения на основе кластерных систем управления базами данных или распределённая служба каталогов LDAP.

CP: Распределённая система, в каждый момент обеспечивающая целостный результат и способная функционировать в условиях распада в ущерб доступности, может не выдавать отклик. Устойчивость к распаду на секции требует обеспечения дублирования изменений во всех узлах системы, в этой связи отмечается практическая целесообразность использования в таких системах распределённых пессимистических блокировок для сохранения целостности.

AP: Распределённая система, отказывающаяся от целостности результата. Хотя системы такого рода известны задолго до формулировки принципа CAP (например, распределённые веб-кэши или DNS), рост популярности систем с этим набором свойств связывается именно с распространением теоремы CAP. Так, большинство NoSQL-систем принципиально не гарантируют целостности данных и ссылаются на теорему CAP как на мотив такого ограничения. Задачей при построении AP-систем становится обеспечение некоторого практически целесообразного уровня целостности данных, в этом смысле про AP-системы говорят как о «целостных в конечном итоге» (англ. eventually consistent) или как о «слабо целостных» (англ. weak consistent).

Перейдем к определению ACID, что расшифровывается как:

Atomicity – транзакции атомарны, то есть либо все изменения транзакции фиксируются (commit), либо все откатываются (rollback).

Consistency – транзакции не нарушают согласованность данных, то есть

они переводят базу данных из одного корректного состояния в другое. Тут можно упомянуть допустимые значения полей, внешние ключи и более сложные ограничения целостности.

Isolation – работающие одновременно транзакции не влияют друг на друга, то есть многопоточная обработка транзакций производится таким образом, чтобы результат их параллельного исполнения соответствовал результату их последовательного исполнения.

Durability – если транзакция была успешно завершена, никакое внешнее событие не должно привести к потере совершенных ей изменений.

Основы NoSQL

В основе идеи NoSQL лежит следующее:

Основные возможности:

- Нереляционная модель данных;
- Открытый исходный код;
- Хорошая горизонтальная масштабируемость;
- Не требуют жестко заданной схемы данных.

Другие характеристики:

- Свободны от одного или нескольких свойств ACID;
- Поддержка репликации:
- Легкое API (если SQL, то только облегченный вариант).

Не полностью поддерживают возможности традиционных реляционных БД:

- нет join операций, т.к. они требуют жестко заданной структуры и связей (Consistency)

История:

Впервые термин NoSQL стал использоваться в конце 90-х, но реальный смысл в том виде, как он используется сейчас, приобрел только в середине 2009. В июне 2009 в Сан-Франциско Йоханом Оскарссоном была организована встреча, на которой планировалось обсудить новые веяния на ИТ рынке хранения и обработки данных. Главным стимулом для встречи стали новые

opensource-продукты наподобие BigTable и Dynamo. Для яркой вывески для встречи требовалось найти емкий и лаконичный термин, который отлично укладывался бы в твиттеровский хэштег. Один из таких терминов предложил Эрик Эванс из RackSpace – «NoSQL». Термин планировался лишь на одну встречу и не имел под собой глубокой смысловой нагрузки, но так получилось, что он распространился по мировой сети наподобие вирусной рекламы и стал де-факто названием целого направления в ИТ-индустрии.

Стоит еще раз подчеркнуть, что термин “NoSQL” имеет абсолютно стихийное происхождение и не имеет общепризнанного определения или научного утверждения за спиной. Это название скорее характеризует вектор развития информационных технологий в сторону от реляционных баз данных.

Механизмы доступа к NoSQL:

RESTful интерфейсы – это интерфейс, похожий на основной протокол интернета – HTTP. В рамках данного подхода предполагается, что каждый объект, с которым можно манипулировать, имеет свой уникальный адрес. Обращаясь по этому адресу, можно запрашивать, создавать, редактировать или удалять указанный объект. При этом на сервере не сохраняется никакого состояния, то есть каждый запрос обрабатывается независимо от других запросов.

Языки запросов отличные от SQL

- GQL – SQL-подобный язык для Google BigTable;
- SPARQL – язык запросов семантического веба;
- Gremlin – язык обхода графов;
- Sones Graph Query Language – языкзапросовк Sones Graph.

API запросов

- Google BigTable DataStore API
- Neo4j Traversal API

Категориями хранилищ в зависимости от размера и сложности NoSQL являются:

Хранилище для Ключ/Значение – Key/value store:

В этом случае база данных представляется как коллекция элементов ключ/значение, в каждой паре которых ключ обязательно является уникальным. Отсутствуют требования для нормализации, структуры данных сложные. Данные распределяются по разным узлам кластера на основе ключа.

Основные операции:

- insert(key,value);
- delete(key);
- update(key,value);
- lookup(key);
- execute(key, operation, parameters).

Имеют серьезные ограничения в языке запросов

Дополнительные операции:

- Вариации перечисленных выше операций – обратный lookup;
- Итерации.

Хранилище колоночного типа – Column store

Эта база представляет собой большую таблицу с тремя измерениями: колонки, строки и временные метки. Набор колонок не фиксирован и может различаться от строчки к строчке.

Документо-ориентированные хранилища – Document store

Такие базы немного напоминают Key-Value базы, но в данном случае база данных знает, что из себя представляют значения. Обычно значением является некоторый документ или объект, к структуре которого можно делать запросы.

Основное понятие документа – слова, фразы, предложения, параграфы и т.д. Имеет гибкую структуру, в которой подкомпоненты структуры могут быть вложенными и различной формы. Поддерживаются метаданные – название, автор, время, встроенные теги. Форматы документов – PDF, XML, JSON, текст, сканированные рисунки.

Основныеоперации:

- insert(key, document);

- fetch(key);
- update(key);
- delete(key).

Графо-ориентированныехранилища – Graph store

Подобные базы ориентированы на поддержку сложных взаимосвязей между объектами. Это направленные графы, в которых узлы и грани содержат некие свойства. Структура данных в таких базах представляет собой набор узлов, связанных между собой ссылками. При этом и узлы, и ссылки могут обладать некоторым количеством атрибутов.

MongoDB

Одним из инструментов для NoSQL является MongoDB, разрабатываемый компанией 10gen на языке C++. Общение с клиентом происходит через специализированный бинарный протокол, а для работы с документами используется BSON – посути тоже самое, что и JSON, но в бинарном виде. Главная причины выбора BSON вместо JSON - скорость парсинга данных. Бинарные данные расшифровываются намного быстрее. Имеет большой набор вариантов репликации и распределения данных. Для использования требуются специальные драйверы, которые созданы для большинства популярных языков программирования.

Документы разделяются на коллекции по типу, что напоминает табличную структуру реляционных баз данных. Но в отличие от RDBMS коллекции разделяют данные только по смыслу. У каждого документа есть идентифицирующее поле _id и никакого поля, позволяющего следить за ревизиями, нет.

Репликация вида master-slave: вся работа осуществляется с главным сервером, а изменения отправляются на подчинённый сервер. Если мастер отключается, то подчинённый не может автоматически занять его место. Репликационные множества (Replica sets): до 7 компьютеров, при потере мастера, автоматически выбирается новый.

Шардинг поддерживается, начиная с версии 1.6. Отсутствует жёсткое

ограничение на число серверов и каждая отдельная группа (shard) может быть представлена не одним сервером, а репликационным множеством.

Сравнение NoSQL с традиционной базой данных:

Достоинства NoSQL:

- В сравнении с реляционными базами данных лучшая производительность при индексировании больших объёмов данных и большим количеством запросов на чтение;
- Легче масштабируются в сравнении с SQL-решениями;
- Децентрализированы (данные распределены на нескольких нодах, отсутствует проблема single-point-of-failure);
- Легко менять "схему" данных: не нужно выполнять никаких операций обновления для добавления новых полей;
- Нет проблем с хранением неструктурированных данных;
- Единое место хранения всей информации об объекте: меньше операций вида "join";
- Простой интерфейс общения с БД: ключ → значение, нет SQL.

Достоинства традиционных БД:

- Теоретический базис;
- Гибкость, организованность и надежность;
- Риски известны.

Недостатки NoSQL:

- Не поддерживает (операции Joins, Group by; ACID-принципы; SQL; интеграцию с приложениями, которые построены на SQL).
- Отсутствие транзакционной логики и контроля целостности в большинстве реализаций (необходимо реализовывать её в логике приложения; специализированная логика может оказаться эффективнее общих алгоритмов реляционных БД).
- Для обработки данных необходимо использование дополнительного языка программирования (плюс для программистов, которые этот язык уже

знают, кроме того, иногда для этих целей можно использовать разные языки).

Недостатки традиционных БД:

- Сложность создания запросов на SQL, которые выходят за рамки простого SELECT;
- Посредственная масштабируемость на сверхвысокую нагрузку в виде большого количества запросов на чтение и запись;
- Плохая адаптируемость к сетям распределённых вычислений;
- Плохая поддержка объема данных больше 10-100 Тб;
- Ограничение на жесткую структуру и схему данных;
- Сложность при Join для данных на нескольких узлах.

Они плохо подходят под вызовы Big Data. В то время как NoSQL используется для:

- Аналитической обработки большого объёма данных;
- На сайтах с очень высокой нагрузкой, поскольку легко масштабируются горизонтально;
- Совместно с традиционными решениями на реляционных базах данных.

Когда:

- В приложениях, которые работают с огромными объемами полу-структурированных данных (анализ логов приложения/системы; анализ социальных сетей);
- Большинство приложений в организациях используют небольшой объем данных и не имеют необходимости в высокой скорости выполнения запросов на обновление и выборку.

В этом случае традиционные РСУБД – правильный выбор.

Вывод

Нереляционные БД удобны при хранении плохо структурированной информации, масштабируются намного лучше RDBMS, и таким образом неизбежны при обработке очень крупных объёмов данных, эффективны для аналитической обработки данных и часто не имеют транзакционных

механизмов.

Контрольные вопросы для повторения и самопроверки по теме №4

1. Что является отличительной особенностью NoSQL?
2. В каком случае стоит применять NoSQL хранилища?
3. Что, согласно теореме CAP, возможно обеспечить в любой реализации распределённых вычислений?
4. Какое свойство означает, что транзакции не нарушают согласованность данных, то есть они переводят базу данных из одного корректного состояния в другое?
5. Какой способ хранения данных используется в MongoDB?
6. Что относится к плюсам репликации?
7. Что относится к преимуществам нереляционных БД?
8. На какие три группы подразделяют пользователей в MongoDB?
9. Что такое шардинг?
10. Какие три свойства фигурируют в определении теоремы CAP?

Литература по теме №4

1. Фаулер М., Прамодкумар Дж. Садаладж. NoSQL: новая методология разработки нереляционных баз данных. – М.: «Вильямс», 2013. – 192с.
2. Смородин В.В., Волкова Е.В., Алиев А.А. От хранения данных к управлению информацией. – СПб.: Изд-во Питер, 2010. – 528с.

Глоссарий по теме №4

NoSQL (англ. Not Only SQL, не только SQL) – термин для ряда подходов, преследующих цель решить такие проблемы, как масштабируемость, доступность, устойчивость к разделению и согласованность данных.

Вертикальное масштабирование –масштабируемость в этом контексте означает возможность заменять в существующей вычислительной системе компоненты более мощными и быстрыми по мере роста требований и развития

технологий.

Горизонтальное масштабирование – масштабируемость в этом контексте означает возможность добавлять к системе новые узлы, серверы, процессоры для увеличения общей производительности.

Репликация – это копирование данных на другие узлы при обновлении, позволяющее создать полный дубликат базы данных так, что вместо одного сервера их будет несколько, повысить отказоустойчивость, снизить нагрузку на каждый отдельный сервер и как добиться большей масштабируемости, так и повысить доступность и сохранность данных.

Одноранговая сеть – это сеть, которая основывается на равноправии всех участников сети.

Шардинг – это другая техника масштабирования работы с данными, суть которой заключается в разделении базы данных на отдельные части так, чтобы каждую из них можно было вынести на отдельный сервер.

Вертикальный шардинг – это выделение таблицы или группы таблиц на отдельный сервер.

Горизонтальный шардинг – это разделение одной таблицы на разные серверы.

Master при репликации Master-Slave – это основной сервер БД, куда поступают все данные и где происходят изменения в данных (добавление, обновление, удаление).

Slave при репликации Master-Slave – это вспомогательный сервер БД, копирующий все данные с мастера, предназначен только для чтения и может быть в количестве больше одного.

ТЕМА №5: ВИЗУАЛИЗАЦИЯ ДАННЫХ И РЕЗУЛЬТАТОВ АНАЛИЗА

Человек – сложная биологическая система, тем не менее, восприятие им информации сильно ограничено, поэтому нужны простые её представления. Принципиальное значение для этого, для ее интерпретации, имеет визуализация. Визуализация - наглядное представление результатов анализа. До сих пор ученые ведут исследования в области совершенствования современных методов представления данных в виде изображений, диаграмм и анимаций.

Визуализация очень важна в современном мире, особенно с большими объемами данных. Все потому, что во всей получаемой информации существует множество связей и зависимостей, просто так их очень сложно заметить. Кроме того, она дает ответы на многие вопросы гораздо быстрее. К примеру, по колонке цифр не все так хорошо ясно, как по графику.

На сегодняшний день рост информации достигает невероятных скоростей. Чтобы справиться с возрастающей сложностью и разнообразием данных необходимо использовать визуализацию. Смотреть на цветные картинки всем нравится больше, чем на скучные таблицы с цифрами. Вместе с тем, субъективное восприятие информации, доверие к ней на много выше, когда она подается визуально.

Рассмотрим области использования визуализации.

Статистика и отчеты. Данные за некий период времени показываются вместе. Например, статической картинкой в приложении к отчету или настраиваемым графиком в сервисе статистики, с возможностью изменения параметров его отображения.

Справочная информация. Дополнение к основному тексту, наглядно иллюстрирующее его упоминаемыми данными. Например, дать общее представление о динамике одного из показателей, либо отобразить какой-то процесс и его этапы; может быть - показать структуру некоего явления.

Интерактивные сервисы. Продукты и проекты, в которых инфографика является частью функциональности. Так, в качестве средства

навигации по сервисам может являться диаграмма процесса. Почти все, что связано с работой с картами в специализированных системах вроде диспетчерских и большей части компьютерных игр.

Иллюстрации. Красивое отображение данных для создания самостоятельных иллюстраций.

Чертежи и схемы. Специализированные документы, показывающие структуру и процесс работы сложных инженерных и природных систем

Эксперименты и искусство. Визуализация данных в виде сложных и громоздких изображений, которые сложно “прочитать” бегло - объем данных и взаимосвязей между ними таков, что нужно разбираться с картинкой по частям; либо просто абстрактные изображения, автоматически сгенерированные. В последнее время направление все более популярно и периодически выходит за рамки компьютерной графики - например, в виде графиков-скульптур.

Типы визуализации

Выделим условно три типа визуализации.

- Научная визуализация. При моделировании различных объектов или процессов появляются большие объемы данных.
- Информационная визуализация. Описание/представление некой абстрактной информации, полученной при сборе и обработке многокатегориальных данных, для анализа которых необходимо применять различные количественные и качественные меры оценки.
- Визуализация работы программного обеспечения.

Задачи визуализации

Визуализация BigData имеет определенные задачи:

- визуализация потоков данных;
- визуальный интеллектуальный анализ данных;
- визуальный поиск и рекомендации;
- описание ситуаций на основе больших данных с использованием визуализации;
- масштабируемые методы параллельной визуализации;

- современные аппаратные средства и архитектуры для анализа и визуализации данных;
- человеко-компьютерный интерфейс и визуализация больших данных;
- приложения визуализации больших данных.

Требования к визуализации

Можно сформулировать требования к такого рода визуализации:

- оценка пригодности (адекватности в визуализации) видов отображения,
- естественность (привычность для пользователей),
- устойчивость к масштабированию,
- возможность вывода сверхбольших объемов данных,
- возможности для представления сложных структур, а также объектов особого интереса, особых точек, аттракторов, сингулярностей.

Как решить проблемы визуализации BigData?

Необходимо:

- использовать достаточно простые, но четко интерпритируемые мотоды представления.
- формализовывать описания объектов программного обеспечения
- производить верификацию и валидацию визуализации.

Традиционные виды визуализации

Рассмотрим традиционные виды визуализации.

- Графики и диаграммы
- Инфографика и схемы
- Презентация и анализ данных
- Интерактивный сторителлинг
- Бизнес аналитика и дашборды
- Научная и медицинская визуализация
- Карты и картограммы

Графики и диаграммы

Наверное, самый привычный вид визуализации данных (рисунок 5.1).

Используется как для презентации данных, так и для анализа. Встретить их можно и на работе, и в журнале, и в научном отчете. Обычно знания о существующих типах диаграмм и графиков мы получаем из школы или из стандартного набора в Excel. Однако мир графиков и диаграмм не ограничивается точечным графиком, столбиковой и круговой диаграммой. Существуют порядка 15 общеизвестных типов диаграмм, а всего их более 60, при этом их количество увеличивается с каждым днём – люди придумывают новые типы для визуализации сложных и необычных данных.

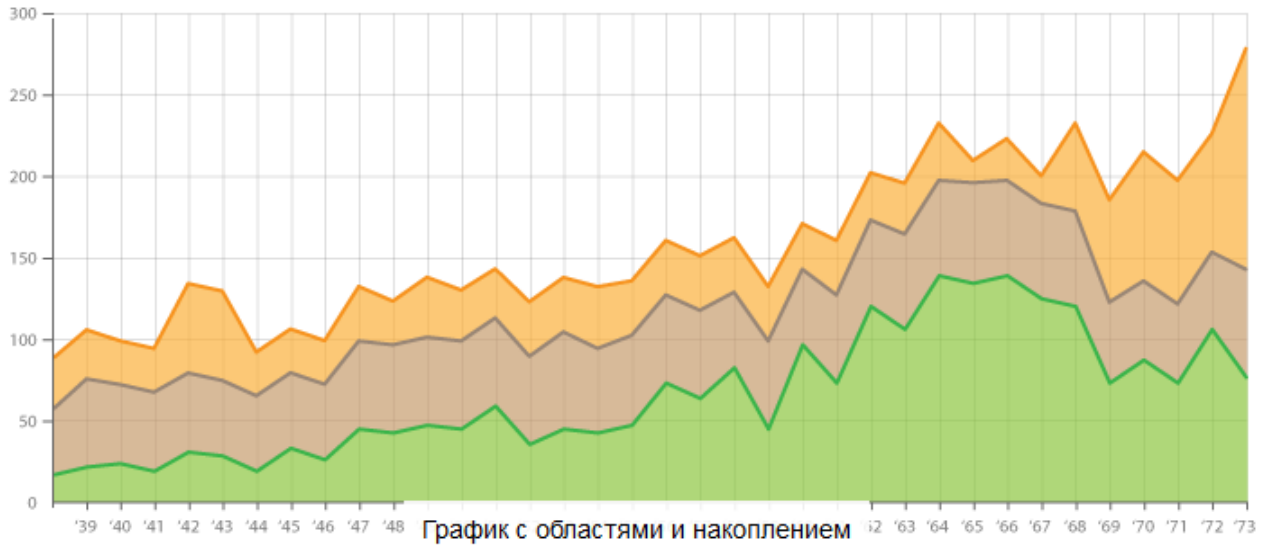


Рисунок 5.1 – Пример визуализации в виде графика

Инфографика

Инфографика стала очень популярна в последние годы, хотя существуют уже давно. Инфографика относится к журналистике данных, где графики и схемы объясняют какие-либо факты по выбранной теме. Обычно инфографика статична и представляет собой длинную «простыню» с картинками и текстом. Отличительной особенностью инфографики является то, что в ней приводятся уже готовые выводы, то есть читателя проводят за руку по выбранной теме и при этом приправляют это все цифрами и картинками. Часто используется рисованный или анимационный стиль. Часто используется не к месту или «для красоты», хотя конечно же есть замечательные и интересные

примеры.

Презентация и анализ данных

Один самых привычных способов использования визуализации данных - презентация информации в виде диаграмм или инфографики. И если с этим все понятно, то использование визуализации для анализа информации в основном используется только бизнес-аналитиками и учеными. В чем заключается отличие?

При анализе данных с помощью визуализации используют так называемое быстрое прототипирование – то есть создание большого количества различных визуальных представлений одних и тех же данных. Делается это для возможности нахождения скрытых, на первый взгляд, взаимосвязей и зависимостей, а также первичной оценки набора данных для возможности применения в дальнейшем более сложных инструментов анализа. Этот подход называется Exploratory data analysis (EDA), что на русский можно перевести как разведывательный анализ данных. Основное отличие от презентации данных - визуализация здесь может быть «черновой», но выполняется быстро и одним человеком или небольшой рабочей группой.

Интерактивный сторителлинг

Сторителлинг – это преподнесение какой-либо полезной информации в форме интересного рассказа. Интерактивный сторителлинг – рассказ, с которым слушатель может взаимодействовать. Пользователь может управлять отображением информации и находить те зависимости, которые не нашёл автор. В этом смысле он близок к разведывательному анализу данных, но отличается тем, что данные заранее обработаны и представлены в удобном для анализа виде, а также имеются подсказки или заранее прописанные сценарии использования.

Поэтому, чаще всего интерактивный сторителлинг называют интерактивной инфографикой, но для того чтобы ей стать недостаточно просто к статичной инфографике добавить всплывающие окошки.

Дашборды и бизнес аналитика

Визуализация активно используется в бизнесе. Принцип «говорите с данными» помогает компаниям зарабатывать больше, а клиентам получать лучший сервис. Для разового анализа обычно используется Excel или R. Однако это неудобно если необходимо следит за какими-то показателями на постоянной основе. Для отслеживания используют дашборды – дисплеи, на которых выведены все необходимые показатели в одном месте в виде графиков, диаграмм и таблиц. Проектирование эффективных дашбордов – сложная и неординарная задача. Зачастую их перегружают ненужной информацией или стараются использовать все возможные типы шаблонных графиков. Часто для того, чтобы спроектировать хороший дашборд, необходимо создание новых типов визуализации информации. Тематика активно развивается за счет все большего применения аналитики в бизнесе. Также дашборды применяются и для личного использования (фитнес трекеры, анализ личных расходов и т. п.)

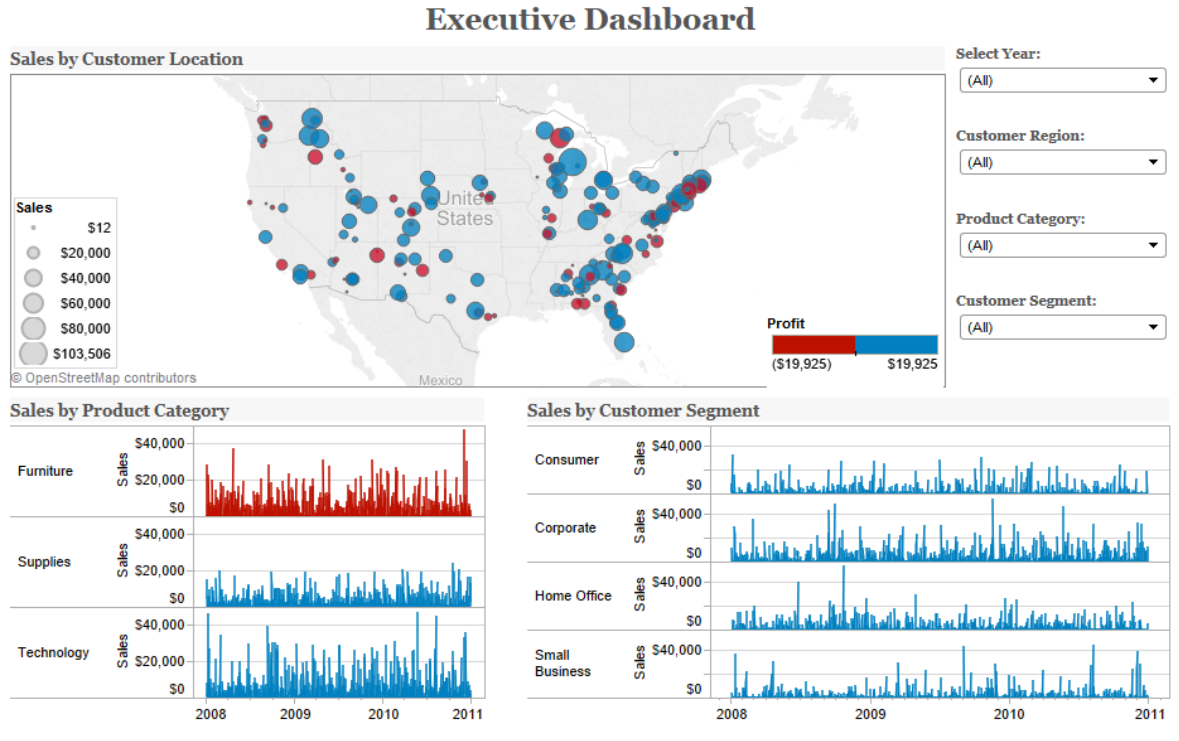


Рисунок 5.2 – Пример дашборда

Визуализация в медицине и науке

Специфический вид визуализации Его целью обычно является

выделение закономерностей или аномалий. От обычной визуализации данных отличается тем, что часто бывает трёхмерной и требует специальной подготовки для интерпретации.

Карты и картограммы

Карты – одни из древнейших способов визуализации, отображающих окружающую реальность. Картограмма – карта с нанесенной на неё информацией в виде цвета или других способов. Картограммы могут быть использованы для отображения любой информации – от плотности населения, до частоты использования мобильных телефонов в каждом районе страны.

Облако тегов

Каждому элементу в облаке тегов присваивается определенный весовой коэффициент, который коррелирует с размером шрифта. В случае анализа текста величина весового коэффициента напрямую зависит от частоты употребления (цитирования) определенного слова или словосочетания.

Позволяет читателю в сжатые сроки получить представление о ключевых моментах сколько угодно большого текста или набора текстов.

Кластерграмма

Метод визуализации, использующийся при кластерном анализе (рисунок 5.3). Показывает, как отдельные элементы множества данных соотносятся с кластерами по мере изменения их количества. Выбор оптимального количества кластеров – важная составляющая кластерного анализа.

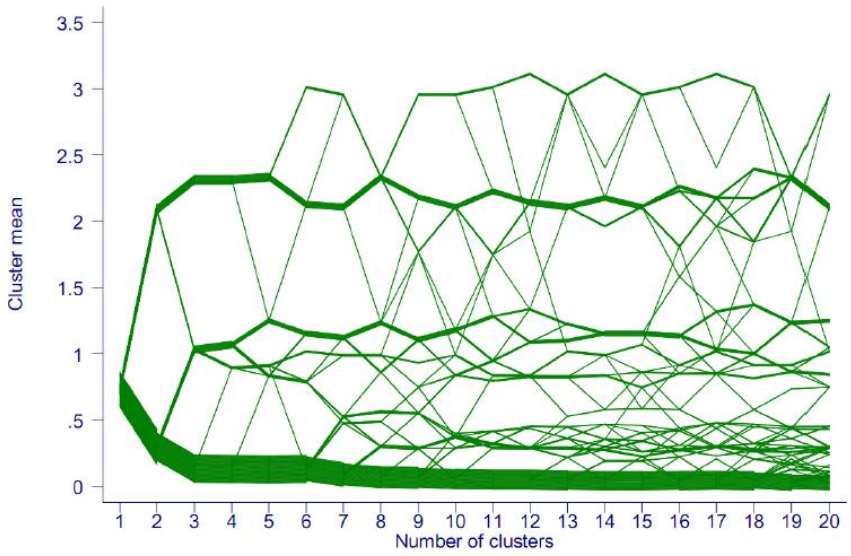


Рисунок 5.3 – Пример кластерграммы

Исторический поток

Помогает следить за эволюцией документа, над созданием которого работает одновременно большое количество авторов. В частности, это типичная ситуация для сервисов wiki в том числе. По горизонтальной оси откладывается время. По вертикальной - вклад каждого из соавторов, т.е. объем введенного текста. Каждому уникальному автору присваивается определенный цвет на диаграмме

Пространственный поток

Используется для отслеживания пространственного изменения информации.

Язык R

Это интуитивно понятный язык сценариев. Позволяет импортировать и использовать большое количество пакетов для анализа и визуализации. Для подобного основанного на модели анализа можно также использовать другие научные инструменты визуализации, такие как, например, MATLAB, но R содержит различные пакеты, хорошо выполняющие многомерный анализ наборов данных, не являющихся научными по своей природе (визуализация бизнес-аналитики).

Существует язык, основанный на R, который предназначен для статистических исследований и работы с графикой. Это opensource проект R

Foundation. Широкий спектр различных функций (временные ряды, прогнозирование, классификация, кластеризация и др.). Важное отличие в том, что доступны простые средства построения самых сложных графиков и диаграмм. А также возможность расширения благодаря технологии разработки дополнительных пакетов участниками проекта

Отличия языка R

Инструменты, подобные MATLAB, предоставляют среду интерактивного научного и инженерного анализа для исследования моделей и данных инженерам и ученым; R предоставляет те же средства бизнес-аналитикам и аналитикам больших данных всех типов. Возможность интерактивного исследования больших данных при помощи таких инструментов, как R и BigQuery, отличает анализ больших данных от пакетного и глубинного анализа данных, которые часто выполняются с помощью MapReduce. В любом случае целью является формирование новых моделей и поддержка принятия решений с использованием больших данных.

Основные возможности R

Основные возможности языка R:

- Можно вводить команды одну за другой в командной строке (>) или запустить последовательность команд из файла-источника
- Язык R содержит огромное количество различных типов данных, включая векторы (числовые, строковые, логические), матрицы, блоки данных и списки
- Гибкость языка R обеспечивается посредством встроенных и пользовательских функций. Во время работы с R все пользовательские данные хранятся в памяти программы
- Базовые функции доступны по умолчанию. Другие функции содержатся в особых статистических пакетах и могут быть загружены во время работы
- R содержит встроенные базы данных, которые можно использовать для обучения

Типы данных

Типы данных языка R: вектор, матрица, список, таблица данных. Статистические функции: среднее, медиана, дисперсия, стандартное отклонение, межквантильный размах.

Из стандартных операторов можно выделить суммирование, разность, умножение, деление, возведение в степень, остаток от деления, целочисленное деление, логические операции.

Язык легко встраивается в OracleDB благодаря 100% совместимости с R-интефейсом и клиентскими приложениями. Данные сохраняются и статические вычисления уже выполняются в базе данных.

Amazon S3

Рассмотрим Amazon S3 (где S3 расшифровывается, как Simple Storage Service). Это веб-служба, предлагаемая AmazonWebServices, она предоставляет возможность хранения и получения любого объема данных в любое время из любой точки земного шара. Это очень надежное средство хранения объектов, которое легко масштабируется и отлично подходит для разработчиков. Оно очень удобно и оснащено простым интерфейсом. Платить придется только за используемый объем. Также есть бесплатная версия на 1 год.

Достоинства Amazon S3

- Доступность – AWS будет работать всегда
- Масштабирование – не нужно решать сложные задачи распределения файлов между серверами
- Нагрузка – внезапные всплески популярности не приведут к отказу железа
- Надежность – предоставляется практически 100% гарантия целостности и доступности.
- Общий объем хранимых данных и количество объектов неограничены
- Размер отдельных объектов Amazon S3 может быть от 1 байта до 5 терабайт
- Самый крупный объект, который можно загрузить через один запрос – 5 гигабайт.

- Для объектов крупнее 100 мегабайт клиентам рекомендуется использовать функцию многокомпонентной загрузки.

Многокомпонентная загрузка

При многокомпонентной загрузке файл загружается по частям последовательно или параллельно через специальный API. Есть возможность шифрования данных при загрузке. Преимущества многокомпонентной загрузки: увеличение пропускной способности; быстрое восстановление загрузки при отказе машины (или сети); возможность приостановить и возобновить загрузку; возможность загрузки объекта в момент его создания.

Загрузка проходит в 3 этапа

- Инициирование загрузки
- Загрузка всех частей файла
- Завершение загрузки

После этого Amazon S3 строит объект (файл) из загруженных частей.

Шифрование данных используется:

- На стороне сервера (передача через SSL, Amazon S3 шифрует данные при записи, расшифровывает при чтении)
- На стороне клиента (клиент сам шифрует данные, отправляет их в S3, управляя процессом шифрования, Amazon предоставляет соответствующий API)

Особенности хранения в S3

- Файлы хранятся в отдельных бакетах, в которых можно создавать директории и поддиректории (бакет - некоторый контейнер, содержащий директории и файлы; часть в файловой системе S3)
- Бакеты хранятся в разных регионах (Region; регионы – ЦОД в различных регионах мира (США, Европа, Азия), есть возможность выбора региона)
- Уникальные ключи
    - При хранении данных объектам назначается уникальный ключ, который может использоваться впоследствии для доступа к данным.
    - Ключ назначает пользователь.

    - Для запроса файла необходимо знать бакет, где он находится, и его ключ.
    - Ключи могут быть любой строкой кодировки UTF-8 не более 1024 байт, могут быть созданы таким образом, чтобы имитировать иерархические атрибуты
    - Ключи уникальны внутри одного бакета
- Логирование (S3 может логировать запросы и складывать отчёты в отдельный бакет)
- Версионирование (S3 предполагает версионность файлов, всегда можно восстановить файл предыдущей версии, т.е. откатиться до нужного состояния).

В S3 используется свободная масштабируемость. S3 автоматически масштабируется в зависимости от потребностей клиентов. Amazon предлагает не беспокоиться о возможностях масштабируемости системы и предоставляет гарантию на то, что резкий всплеск данных в хранилище не создаст никаких проблем.

Дедупликация данных

Дедупликация – технология, при помощи которой обнаруживаются и исключаются избыточные данные в дисковом хранилище. Есть два способа дедупликации:

- Онлайн дедупликация (может осуществляться непосредственно в момент записи данных на диски (данные кэшируются))
- Оффлайн дедупликация (может быть реализована «постпроцессом», в фоновом режиме после записи данных на диск)

Хорошо подходят для дедупликации резервные копии, виртуальные машины, любые данные с большим количеством повторяющихся блоков. Плохо подходят для дедупликации картинки, музыка, видео, сжатые данные.

Сравним два способа дедупликации – онлайн и офлайн.

Плюсы «оффлайн» дедупликации (минусы «онлайн» дедупликации):

- можем использовать более эффективные и точные алгоритмы обнаружения

дубликатов данных;

- не нужно идти на компромиссы, чтобы не перегрузить работой процессор и не снизить производительность работы системы хранения с дедупликацией;
- можем анализировать и обрабатывать значительно большие объемы данных;
- можем делать дедупликацию тогда и там, когда и где удобно.

Минусы «оффлайн» дедупликации (плюсы «онлайн» дедупликации):

- если записывать сильно дуплицированные данные (например, 90% - дупликаты), придется сначала выделить на запись место, заполнить его дупликатами, и лишь потом, в ходе процесса дедупликации, 90% этого пространства освободится;
- повышается стоимость сервиса за счет использования трафика на 90% дуплицированных данных, запросов на обращение к этим же данным (для сравнения), запросов на удаление.

Таким образом, дешевле использовать «онлайн» дедупликацию, но это крайне медленно. Удобнее и эффективнее использовать «оффлайн» дедупликацию, но это дороже.

Предлагается использовать оффлайн дедупликацию с временным хранилищем:

- Данные с клиента записываются во временное хранилище (не в S3, а в другое (например, на основе HDFS))
- Запускается “постпроцессная” дедупликация в хранилище
- Данные дедуплицируются
- Дедуплицированные данные загружаются в S3
- Временное хранилище очищается

Таким образом, достигается оптимальное соотношение между двумя подходами: минимизация стоимости, увеличение производительности (можно дедуплицировать, когда необходимо), увеличение скорости работы системы.

Рассмотрим традиционный алгоритм дедупликации:

- Входные данные разбиваются на блоки по определенному алгоритму и на блоки определенного размера.
- Для каждого блока вычисляется хеш-функции (используются различные алгоритмы вычисления).
- Хеш сравнивается с существующими (хэш хранится в таблицах).
- Если хеш существует, то заменяем данные на ссылку.

Каждая служба облачной дедупликации (CDS) должна иметь:

- Веб интерфейс
- Модуль приема данных
- Алгоритм дедупликации данных для записи и чтения
- Алгоритм вычисления хэш-функции
- Место хранения

У службы должны быть следующие характеристики:

- Масштабируемость
- Гибкость
- Доступность
- Балансировка нагрузки

Каждый модуль должен хорошо сочетаться с облачной архитектурой.

Контрольные вопросы для повторения и самопроверки по теме №5

1. Для чего нужна визуализация?
2. Как называется один из самых популярных языков сценариев?
3. Какие достоинства у Amazon S3?
4. Какие традиционные виды визуализации?
5. Какие отличия и основные возможности у языка R?
6. В чем особенности хранения в Amazon S3?
7. Что такое дедупликация данных?
8. В чем основные задачи визуализации?
9. Какие требования предъявляются к визуализации?

10. Какие типы визуализации можно выделить?

Литература по теме №5

1. Яу Н. Искусство визуализации в бизнесе. Как представить сложную информацию простыми образами. – Wiley Publishing, Inc. – 2013.
2. Храмов Д.А. Сбор данных в Интернете на языке R. – М.: ДМК-пресс – 2017. – 282с.

Глоссарий по теме №5

Прототипирование – то есть создание большого количества различных визуальных представлений одних и тех же данных.

Картограмма – карта с нанесенной на неё информацией в виде цвета или других способов.

Бакет – контейнер, содержащий директории и файлы; часть в файловой системе Amazon S3.

Дедупликация – технология, при помощи которой обнаруживаются и исключаются избыточные данные в дисковом хранилище.

Онлайн дедупликация – дедупликация непосредственно в момент записи данных на диски с кэшированием данных.

Оффлайн дедупликация – дедупликация в фоновом режиме после записи данных на диск.

Fuse Based File System – файловая система пользовательского уровня, которая используется для представления файлов, содержащихся внутри Dedup Engine в виде тома файловой системы.

Dedup Storage Engine – сервис на стороне сервера, который хранит и извлекает блоки дедуплицированных данных.

SDFS – распределенная и расширенная файловая система, которая обеспечивает встроенную дедупликацию.

ТЕМА №6: КЛАССИФИКАЦИЯ ЗАДАЧ АНАЛИЗА ДАННЫХ

Определение DataMining

Data Mining (добыча данных, интеллектуальный анализ данных (ИАД), 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.

Предмет интереса для анализа

- Нетривиальные знания
- Неявные зависимости
- Ранее неизвестные знания
- Практически полезные знания
- Доступные для интерпретации

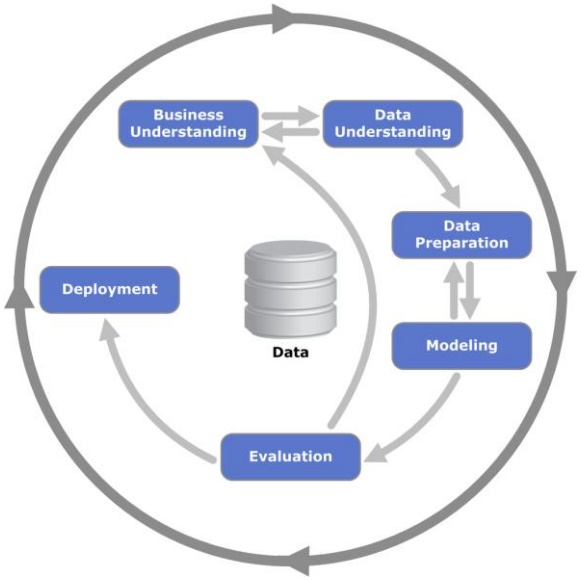


Рисунок 6.1 – Ключевые шаги DataMining

Отличия от традиционного анализа

- Огромные объемы данных (требуются масштабированные алгоритмы для террабайтных БД)

- Данные высокой размерности (до десятков тысяч измерений)
- Высокая сложность данных
- Данные временных рядов, временные данные, данные последовательностей событий
- Структурные данные, графики, социальные отношения, данные со множественными ссылками
- Гетерогенные источники данных, БД
- Пространственные, пространственно-временные, мультимедиа, текстовые и Web-данные

Задачи ИАД

- Описательные: наглядное описание имеющихся скрытых закономерностей
- Предсказательные: предсказание для тех случаев, для которых данных ещё нет

Виды ИАД

- Выявление ассоциативных связей
    - Определение закономерностей в событиях или процессах
    - Ассоциативная связь различных фактов одного события
    - Результат:
    - Лучшее понимание природы процессов
    - Прогнозирование новых событий
- Кластеризация объектов
    - Разделение исследуемого множества объектов на группы (кластеры) по принципу сходства.
    - В процессе кластеризации методами ИАД определяются схожие характеристики объектов, на их основе объекты объединяются.
- Классификация
    - Отнесение объектов к одному из известных классов на основе их характеристик
- Задачи регрессии

- Задача определения значения одного из параметров анализируемого объекта на основе других характеристик
- Все характеристики - количественные

Типы Data Minig

Сфера применения Data Mining ничем не ограничена, Data Mining нужен везде, где имеются какие-либо данные.

- Анализ рыночных корзин
- Управление взаимоотношениями с клиентами
- Анализ текстовой информации (Text Mining)
- Анализ информации, порождаемой в сети Интернет (Web Mining)
- Анализ социальных медиа (Social Mining)

Анализ рыночной корзины (market basket analysis) – это поиск наиболее типичных, шаблонных покупок в супермаркетах (поиск ассоциативных правил).

Анализ рыночной корзины производится путем анализа баз данных с целью определения комбинаций товаров, которые связанны между собой. В каждой такой паре один товар будет ключевым, а товар, покупаемый вместе с ним – сопутствующим.

Подобный анализ позволит выявить частоту покупки парных товаров, а также вероятность с которой сопутствующий товар покупается вместе с ключевым.

Text Mining

Анализ структурированной информации (например, в БД) уже довольно непростая задача. Но не все виды данных можно успешно структурировать. Например, текстовые документы практически невозможно структурировать без потери семантики и смысловых отношений.

Методы анализа текстов лежат на стыке нескольких областей:

- Data Mining
- обработкаестественныхязыков (NLP - Natural Language Processing)
- поиск и извлечение информации

Text Mining – это нетривиальный процесс обнаружения действительно новых, потенциально полезных и понятных шаблонов в неструктурированных текстовых данных. Ключевая особенность - "неструктурированные текстовые данные".

Ключевые задачи интеллектуального анализа текстов:

- Категоризация текстов, классификация
- Извлечение информации и информационный поиск

Web Mining

Web Mining – это использование методов интеллектуального анализа данных (Data mining, ИАД) для автоматического обнаружения веб-документов и услуг, извлечения информации из веб-ресурсов и сервисов.

Веб-пространство – огромная, широко распространенная информационная сеть, которая включает:

- Информационные сервисы
- Новости, реклама, потребительская информация
- Финансовые рынки, образовнаие, правительство
- Электронная коммерция, развлекательные ресурсы, социальные медиа
- Гипер-текстовая информация (Hyperlinks)
- База данных статистики использования ресурсов

Веб-информация – это:

- Web Content – текст, изображения, таблицы и т.д.
- Web Structure – гиперссылки, тэги, и т.д.
- Web Usage – логи веб-серверов, прокси

Выделяют различные категории Web Mining в зависимости от предмета интереса:

- Web Content Mining – извлечение знаний непосредственно из контента
- Web Usage Mining – анализ использования веб-ресурсов
- Web Structure Mining – выявление моделей, лежащих в основе ссылочной структуры сети

Web Content Mining

Задача: автоматизация поиска информационных ресурсов в Интернете, включая в себя добычу содержимого из веб-данных

Веб-данные:

- Текст
- Изображения
- Аудио-данные
- Видео-данные
- Мета-данные и гиперссылки

Web usage analysis

Каждая страница содержит какую-либо информацию. Гиперссылка – своеобразная “связь” между двумя страницами. Изучая поток кликов по ссылкам, можно понять, как пользователи перемещаются по веб-страницам, шаблоны их поведения

Web Usage Mining (анализ кликов)

Зачем анализировать использование веб-сайта, понимать паттерны пользователей?

- Улучшение клиенто-ориентированности веб-сайта
- Помощь в предотвращении дезориентации
- Лучшие практики по размещению важной информации на странице (именно там, где ее ищет пользователь)
- Предварительная загрузка и кеширование необходимых веб-страниц

Вопросы, на которые можно получить ответы:

- Есть ли различия в шаблонах действий различных пользователей, от чего они зависят?
- Как меняется поведение пользователей с течением времени?
- Как распределяется сетевая нагрузка в течение дня?

Web structure mining

Гиперссылки – ключевая часть технологии Интернет. Через ссылки выражается взаимосвязь между отдельными веб-страницами. Внутри модели

связи страниц в Интернете также могут лежать значимые знания и законы.

Web Structure Mining – это процесс выявления структуры ссылок и модели, лежащей в основе ссылочной структуры в Интернете.

Чтоможет Web Structure Mining?

- Модель, основанная на топологии гиперссылки может быть использована для классификации Веб-страницы (отнесения ее к какой-либо категории)
- Получение информации о сходстве и отношениях между веб-сайтами
- Определение наиболее авторитетных страниц по отношению к заданным критериям поиска.
- Данный подход широко используется в веб-поиске и ранжировании страниц. (поисковые системы)

Social media mining

Social Media – массовая коммуникация посредством сети Интернет. Изначальная идея – приложения в сети Интернет, основанные на идеях Web 2.0, позволяющие создавать и обмениваться пользовательским контентом (user-generated content). Большое количество такого контента сейчас создается на таких ресурсах, как Facebook, Twitter, Google+.

Системный анализ, поиск скрытых ценных знаний из социальных медиа – Social Data Mining.

Особенности:

- Пользовательской информации очень много
- Но и “шумов” в ней тоже много
- Распределена по разным источникам
- Слабоструктурированная, а большинство не структурирована вовсе
- Динамичная, обновляется постоянно

Почему важно?

- Каждый день количество информации, размещенной пользователями растет, очень малое количество обрабатывается
- Платформа для маркетинга и рекламы
- Моделирование поведенческих законов

- Предсказательный анализ
- Рекомендательный контент

RapidMiner

RapidMiner – инструмент, созданный для Data Mining. Основная идея – аналитик не должен программировать при выполнении своей работы.

- Свободная лицензия, Open Source
- Разработан на Java, кроссплатформенность
- Доступен на Github

Достоинства:

- Достаточно хороший набор операторов, решающих большой спектр задач получения информации из разнообразных источников
- Базы данных
- Файлы
- URL
- Большая коллекция операторов обработки информации
- Фильтрация
- Аггрегация
- Сортировка

Недостатки:

- Несмотря на обилие моделей анализа в стандартном пакете, иногда их бывает недостаточно для выполнения сложных задач
- Нет возможности добавлять собственные модели и операции

Идея – визуальное программирование. Процесс – совокупность операторов, соединенных между собой в заданном порядке для выполнения требуемой задачи анализа/обработки данных. Оператор – логическая единица процесса. Оператор производит действия над данными. У него есть вход-выход («порты»), на вход приходят данные, на выход идут обработанные оператором данные. Несколько операторов в рамках процесса позволяют составлять цепочки обработки данных. К примеру, начать с считывания транзакций из БД,

найти самые большие, сконвертировать в доллары и выдать результат. Цепочки операторов возможно параллелить.

Можно с уверенностью говорить, что RapidMiner – полноценный инструмент для ETL. Спектр решаемых задач:

- Сбор информации с источников (файлы, БД, URL и др.)
- Простая статистическая обработка данных (фильтрация, агрегация, и т.д.)
- Аналитическое моделирование (классификация, нейронные сети и т.д.)
- Валидация моделей
- Визуализация результатов (хоть и идостаточно примитивная)

Контрольные вопросы для повторения и самопроверки темы №6

1. Чем анализ больших данных отличается от традиционного анализа?
2. Какие основные типы Data Mining?
3. Какие категории Web Mining можно выделить?
4. В чем основная задача Web Content Mining?
5. В чем основные задачи интеллектуального анализа текстов?

Литература по теме №6

1. Шитиков В.К., Мастицкий С.Э. Классификация, регрессия, алгоритмы Data Mining с использованием R. – Электронная книга, адрес доступа: https://github.com/ranalytics/data-mining – 2017.
2. Барсегян А. А., Куприянов М.С., Степаненко В.В., Холод И.И. Технологии анализа данных. DataMining, VisualMining, TextMining, OLAP.

Глоссарийпо теме №6

Data Mining – 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.

Анализ рыночной корзины (market basket analysis) - поиск наиболее типичных, шаблонных покупок в супермаркетах.

Text Mining – процесс обнаружения новых, потенциально полезных и понятных шаблонов в неструктурированных текстовых данных.

Web Mining – использование методов интеллектуального анализа для автоматического обнаружения веб-документов и услуг, извлечения информации из веб-ресурсов и сервисов.

ТЕМА №7: СТАТИСТИЧЕСКИЕ МЕТОДЫ АНАЛИЗА ДАННЫХ

Чтобы дать определение статистическим методам анализа данных, необходимо разделить его на главные составляющие и понять, что из себя представляют «Статистика», «Методы» и «Анализ данных».

Статистика (лат. status – состояние дел):

- это отрасль знаний, описывающая вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных;
- это изучение количественной стороны массовых общественных явлений в числовой форме;
- это некая совокупность цифровых данных, представляемых в отчётности предприятий, организаций, отраслей экономики, публикуемых в сборниках, справочниках, периодических изданиях;
- это отрасль общественных наук, специальная научная дисциплина, изучаемая в учебных заведениях.

Статистическими данными являются значения, объединенные по некоторому признаку, свойственному определенным объектам изучения, которыми, как правило, являются массовые социально-экономические явления, процессы и прикладная статистика, представляющие количественную оценку в конкретных условиях места и времени.

Методы – это общая совокупность действий, направленных на решение задачи.

Анализ данных – это процесс, посредством которого извлекается необходимая для обработки информация, в последствии предоставляющая решение поставленной перед ним задачи/проблемы.

Отсюда следует главное определение:

Статистические методы анализа данных – это совокупность неких шагов (действий), занимающихся фильтрацией и обработкой предоставленных статистических данных некоего объекта изучения, с целью получения решения поставленной задачи.

Методы статистического анализа строятся на основе следующих шагов:

- Сбор статистических данных, являющихся характеристикой отдельных единиц совокупности общих данных;
- Статистическое исследование на основе полученных сведений, выявление общих закономерностей, свойственных каждой единице;
- Создание методов анализа и обработки получаемых данных.

Самими методами статистического анализа являются статистические гипотезы и машинное обучение.

Статистические гипотезы

Основной целью статистического анализа является выявление свойств изучаемой *генеральной совокупности*, которая представляет собой полный набор объектов, связанных с поставленной перед изучением проблемой. И в случае, если изучаемая генеральная совокупность представляет собой большое или бесконечное множество элементов, тогда необходимо провести отбор из генеральной совокупности в виде некоего подмножества из N элементов, называемых *выборкой* с необходимым объемом данных, где выявленные в выборке свойства обобщаются на всю генеральную совокупность и называются *статистическим выводом*.

Статистический вывод – это утверждение о том, что представляют собой законы, лежащие в основе изучаемой генеральной совокупности; имеющий два раздела, являющихся по своей сути частными случаями задачи принятия решений:

1. Оценивание параметров законов – вычисление по выборке точечных и интервальных оценок.

2. Проверка статистических гипотез – установление справедливости данного утверждение по отношению к параметрам генеральной совокупности.

Перейдём к определению самих статистических гипотез. Пусть в статистическом эксперименте доступна наблюдению случайная величина $\boldsymbol{X}$,для которой выполняются следующие условия:

- Распределение $\boldsymbol{P}$ известно полностью или частично;

- Любое утверждение, касающееся $\boldsymbol{P}$ называется статистической гипотезой.

На практике, как правило, рассматривают простую гипотезу $\boldsymbol{H_0}$(нулевая гипотеза), и противоречащую ей гипотезу $\boldsymbol{H_1}$ (конкурирующая или альтернативная). Для проверки гипотезы используют критерии, позволяющие принять или опровергнуть гипотезу. Так называемые статистические критерии, о которых говориться далее.

Потому как при принятии решения выявлялись статистические ошибки, их поделили на два типа:

- Ошибка первого рода – обнаружение различий или связей, на самом деле не существующих. Вероятностью возникновения данной ошибки называется уровень значимости, и обозначается буквой α, откуда название ошибки α-error.
- Ошибка второго рода – не обнаружение различий или связей, на самом деле существующих. Названия для вероятности данной ошибки не существует, но обозначением является греческая буква β (β-error). Но вводится обозначение мощность критерия (1-β), и чем она выше, тем меньше вероятность α ошибки.

Для того, чтобы провести проверку статистических гипотез, выделили необходимую последовательность, состоящую из следующих этапов:

1. Формулировка основной гипотезы $\boldsymbol{H_0}$ и конкурирующей гипотезы $\boldsymbol{H_1}$, задание значения $\boldsymbol{\alpha}$.

2. Задание некоего критерия $\boldsymbol{\varphi}$, такого что, его величина зависит от исходной выборки; по его значению можно делать выводы об истинности гипотезы $\boldsymbol{H_0}$; сама статистика $\boldsymbol{\varphi}$должна подчиняться какому-то известному закону распределения, так как сама $\boldsymbol{\varphi}$является случайной в силу случайности $\boldsymbol{X}$.

3. Построение критической области. Из области значений $\boldsymbol{\varphi}$выделяется подмножество $\boldsymbol{C}$(критическая область) таких значений, по которым можно судить о существенных расхождениях с предположением. Ее размер выбирается с учетом выполнения равенства. Выделяются три вида критических областей:

а) Двусторонняя критическая область, определяется интервалами $(-\infty, x_{\alpha/2}) \cup (x_{1-\alpha/2}, +\infty)$, где $x_{\alpha/2}$ и $x_{1-\alpha/2}$находят из условий: $P(\varphi < x_{\alpha/2}) = \alpha/2$, $P(\varphi < x_{1-\alpha/2}) = 1 - \alpha/2$.

б) Левосторонняя критическая область определяется интервалом $(-\infty, x_{\alpha})$, где x_{α} находят из условия: $P(\varphi < x_{\alpha}) = \alpha$.

в) Правосторонняя критическая область определяется интервалом $(x_{1-\alpha}, +\infty)$, где $x_{1-\alpha}$ находятся из условия: $P(\varphi < x_{1-\alpha}) = 1 - \alpha$.

4. Вывод об истинности гипотезы. Наблюдаемые значения выборки подставляются в статистику $\boldsymbol{\phi}$и по попаданию (или непопаданию) в критическую область выносится решение об отвержении (или принятии) выдвинутой гипотезы $\boldsymbol{H_0}$.

Статистические критерии

Задачей статистики является выбор критериев, при помощи которых будут доказываться гипотезы. И есть несколько типов статистических критериев:

- Критерий значимости – это гипотезы о численных значениях известного закона распределения, где $\boldsymbol{H_0 : \alpha = \alpha_0}$ – нулевая гипотеза, а $\boldsymbol{H_1 : \alpha > \alpha_0\ (\alpha < \alpha_0)}$– конкурирующая или альтернативная гипотеза.
- Критерий согласия – проверка подчинения предполагаемому закону распределения (в том числе, критерий Пирсона, Колмогорова и т.д.)
- Непараметрические критерии – критерии, оперирующие частотой и рангом, не включающие параметров распределения в расчеты.
- Параметрические критерии, которые берут в расчет параметры вероятностного распределения признака (средний и дисперсия). Такие как t-критерий Стьюдента, критерий Фишера и т.д.

Более подробно разберем параметрический критерий под названием *t*-критерий Стьюдента, который является общим названием для статистических тестов, в которых присутствует распределение Стьюдента. Он может быть использован для проверки равенства между средними значениями двух выборок, где нулевая гипотеза предполагает, что средние равны.

Одним из требований данного критерия является то, что исходные

данные имеют нормальное распределение, другим – в случае, если применяются двухвыборочные критерии для независимых друг от друга выборок, необходимым является соблюдение условий равенства дисперсии.

Машинное обучение

В этом методе, как очевидно, обучается машина. Но не при помощи исследователя или специалиста, а по данным, которые являются обучающей выборкой. И задачей исследователя становится умение строить алгоритмы, обучающиеся по данным. Обучение проводится по прецедентам (ситуаций или объектов), когда дано конечное множество данных о прецедентах, с собранными (измеренными) по каждому из них данными (описанием). Все имеющиеся описания прецедентов собираются воедино и называются это выборкой.

Обучение проходит как с учителем, так и без. В случае, когда *обучение проходит с учителем* (supervised learning), каждый прецедент – это заданные пара «объект, ответ», и для нахождения функциональной зависимости ответов от описаний необходимо составить алгоритм, который на входе принимает описание, а на выходе дает ответ. В зависимости от ответов, выполняются разные задачи. При задаче классификации множество допустимых ответов конечно, их называют метками классов (множество объектов, имеющих данное значение метки). При задаче регрессии ответом является действительное число или числовой вектор. В случае, когда *обучение проходит без учителя* (unsupervised learning), ответы не задаются и приходится искать зависимости между заданными объектами. В этом случае выполняются задачи кластеризации, когда необходимо сгруппировать объекты в кластеры, пользуясь данными о сходстве парных объектов. И задача поиска ассоциативных правил, где исходные данные – это признаковые описания, и требуется найти наборы признаков и их значения, встречающиеся неслучайно часто в описаниях объектов.

Обозначение в машинном обучении:

- $\boldsymbol{X}$– множество объектов;

- Y– множество ответов;
- $y: X \to Y$ – неизвестная зависимость.

В рамках решения задачи, дано конечно множество точек:

- $\{x_1,\ldots, x_n\} \subset X$ – обучающая выборка;
- $y_1,\ldots,y_n, y_i = y(x_i)$ – известные ответы.

Задачей является нахождение такой функции α, которая аппроксимировала бы целевую зависимость: $\alpha: X \to Y$, решающую функцию, приближающую y на всем множестве X (не только на обучающей выборке).

Типы ответов:

- $Y = \{-1, +1\}$– классификация на 2 класса;
- $Y = \{1, \ldots, M\}$ – на M непересекающихся классов;
- $Y = \{0, 1\}^M$– на M классов, могут пересекаться;
- $Y = R$ – задача регрессии (прогнозирование вещественных значений).

$f_i: X \to D_j$ – множество признаков объекта.

Типы признаков:

- бинарный $\{0, 1\}$
- номинальный$Dj < \infty$
- количественный $Dj = R$

Модель обучения:

Неизвестная функция аппроксимации (целевая): $\alpha: X \to Y$

Для моделирования неизвестной функции вводится параметрическое семейство функций:

$A = \{g(x, \theta) | \ \theta \in \Theta\}$, где $g: X \times \Theta \to Y, \Theta$ – множество допустимых значений параметра **.**

Таким образом, задача обучения сводится к:

Выбору вида функции $g(\ldots)$,который зависит от выбранной задачи:

- Линейные модели ($g(x)$ – линейная композиция от Θ) – регрессия.
- Для классификации: $g(x, \Theta) = sign\Sigma Q_j f_j(x), Y = \{-1, +1\}$

Определению значений параметров Θпо обучающей выборке.

Что нужно, чтобы определить целевую функцию $\alpha: X \to Y$?

Метод обучения – это отображение вида: $\mu: (X \times Y)^n \to A$

- По произвольной выборке $X^n = (x^i, x^i)^n_{i=1}$
- Строится некий алгоритм $\alpha \in A$

Как оценить качество обучения модели?

$L(\alpha, x)$ – функция потерь – величина ошибки алгоритма $\alpha \in A$ на объекте X.

Функция потерь для задачи классификации:

- $L(\alpha, x) = [\alpha(x) != y(x)]$ – индикатор ошибки.

Функция потерь для задачи регрессии:

- $L(\alpha, x) = |\alpha(x) - y(x)|$ – абсолютное значение ошибки.
- $L(\alpha, x) = (\alpha(x) - y(x))^2$ – квадратичная ошибка.

В качестве идеи каждого алгоритма зачастую лежат вполне определенные гипотезы, представления о природе данных:

- Метрические классификаторы – идея об измеримости пространства, метрическом взаимоотношении между объектами;
- Логические классификаторы – идея о логических закономерностях происхождения данных;
- Линейные классификаторы – идея о наличии линейных разделяющих плоскостей между классами в пространстве объектов;
- И т.д.

Метрический классификатор

Метрический классификатор – это некий алгоритм, который вычисляет оценку сходства предоставленных объектов. Они основываются на гипотезе компактности, предполагающей, что все похожие друг на друга объекты чаще всего принадлежат одному классу.

Самым простым метрическим классификатором является Методы ближайших соседей, где объект, над которым проводится классификация, относят к классу, состоящему из схожих с ним объектов. Классифицируемый

объект $\boldsymbol{x}$ принадлежит тому классу $\boldsymbol{y}$, которому принадлежит его ближайший объект выборки. К преимуществам относится простота реализации и интерпретируемость решений, а к недостаткам – неустойчивость к погрешностям, отсутствие параметром, которые можно настроить, низкое качество классификации и то, что приходится хранить всю выборку полностью.

Одной из разновидностей данного метода является метод $\boldsymbol{k}$ ближайших соседей, в котором берется $\boldsymbol{k}$ число ближайших соседей, и объект относится к тому классу, к которому относится большинство его соседей. Преимуществом является меньшая чувствительность к шуму, а недостатком, что близости к двум классам могут быть равны.

Также, существует Метод взвешенных ближайших соседей, когда по убывающей с ростом ранга соседям приписывается вес, для того, чтобы не возникало неоднозначности при количестве классов более двух.

Линейный классификатор

Идеей подхода является линейность пространства объектов. *Линейный классификатор* – это алгоритм классификации, который основан на построении разделяющей линейной поверхности. В случае двух классов данная поверхность является гиперплоскостью, делящей пространство признаков на полупространства. Для того, чтобы понять степень погруженности (насколько далеко от границы находится объект, принадлежащий тому или иному классу), существует понятия *отступа*. И чем значение отступа меньше, тем выше вероятность ошибки, потому как это значит, что объект ближе к границе.

Для применения методов оптимизации используют замену пороговой функции потерь, где заменяют функционал эмпирического риска на его непрерывную аппроксимацию, и тогда минимизируется не функционал функции потерь, а его верхняя оценка:

$$\boldsymbol{Q(w) = \sum_{i=1}^{n} L(g(w, x_i), y_i) = \sum_{i=1}^{n} L_i(w) \rightarrow \min_{w}}$$

Для формирования весов объектов функции используется градиентный спуск, использующий движение вдоль градиента (направленного на

возрастание вектора):

$$\boldsymbol{w^{t+1} = w^t - h\sum_{i=1}^{n} \nabla L_i(w^t)}$$

Разновидностью градиентного спуская является Стохастический градиентный спуск (SG), который на вход принимает выборку – $\boldsymbol{X^n}$, темп обучения – $\boldsymbol{h}$ и темп забывания – $\boldsymbol{\lambda}$, на выходе предоставляя вектор весов – $\boldsymbol{w}$. Первым делом, происходит инициализация веса: $\boldsymbol{w_j}$, градиентов: $\boldsymbol{G_i = \nabla L_i(w)}$ и оценки функционала: $\boldsymbol{\bar{Q} := \frac{1}{n}\sum_{i=1}^{n} L_i(w)}$. После чего зациклить следующие шаги: выбор объекта $\boldsymbol{x_i}$ из $\boldsymbol{X^n}$ случайным образом; вычисление потери $\boldsymbol{\varepsilon_i = L_i(w)}$; вычисление градиента $\boldsymbol{G_i = \nabla L_i(w)}$; градиентный шаг $\boldsymbol{w := w - h\sum_{i=1}^{n} G_i}$; и оценка функционала $\boldsymbol{\bar{Q} := (1-\lambda)\bar{Q} + \lambda n \varepsilon_i}$, до тех пор, пока значение $\boldsymbol{\bar{Q}}$ $\frac{\text{и}}{\text{или}}$ веса $\boldsymbol{w}$ не сойдутся. Преимуществом данного алгоритма является простая реализация, возможность работать со сверхбольшими выборками даже при неполной их обработке. А к недостаткам – возможная расходимость или медленная сходимость, застревание в минимумах и проблема переобучения.

Характеристиками бинарного классификатора будут являться:

1. Точность – показатель количества позитивных предсказанных позитивных объектов.

$$Presicion = \frac{TP}{TP + FP}$$

2. Полнота – показатель количества предсказанных позитивных классов от общего количества реальных позитивных объектов.

$$Recall = \frac{TP}{TP + FN}$$

3. F-мера – характеристика, позволяющая дать оценку одновременно точности и полноте.

$$F_{measure} = \frac{1}{\alpha\frac{1}{Pr} + (1-\alpha)\frac{1}{Recall}}$$

ROC – кривая

ROC-кривая («receiver operating characteristic») – графическая

характеристика бинарного классификатора, показывающая зависимость TPR (доля верных положительных классификаций, или полнота) от FPR (доля ложных положительных классификаций). Данная кривая монотонно не убывает, и чем выше лежит кривая, тем лучше качество классификации.

Площадь под ROC-кривой называется AUC (Area Under Curve) и является агрегированной характеристикой качества классификации, которая не зависит от соотношения цен ошибок. Показатель предназначен для сравнительного анализа нескольких AUC, но содержа информации о FPR и TPR модели.

Кластерный анализ

Кластеризация – это обучение без учителя.

Кластеризация – это задача разделить множество объектов на различные группы, которые называются кластерами.

Целью кластеризации является:

- Упрощение дальнейшей обработки данных – разбиение пространства объектов на группы схожих между собой объектов для работы с каждой группой по отдельности (классификации, регрессии, прогнозирования);
- Сокращение объема хранимых данных – оставление по представителю каждого из кластеров (сжатие данных);
- Выделение нетипичных объектов – тех, что не удовлетворяют потребностей кластеров (одноклассовая классификация);
- Построение иерархии множества объектов – таксономия.

Дается X – пространство объектов, $X^n = \{x_i\}_{i=1}^n$ – обучающая выборка и $p: X \times X \to [0, \infty)$ – функция расстояния между объектами. И необходимо найти множество кластеров Y – и алгоритм кластеризации – $\alpha: X \to Y$, такой, что каждый кластер состоит из близких объектов и объекты разных кластеров существенно различаются.

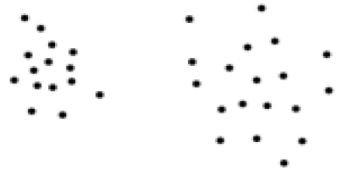
Некорректностью задачи кластеризации является её неоднозначность, потому как:

- Четкой постановки задачи не существует;

- Большое количество критериев качества кластеризации;
- Большое количество эвристических методов деления на кластеры;
- Число кластеров заранее неизвестно;
- Результат зависит от субъективной метрики ***p***.

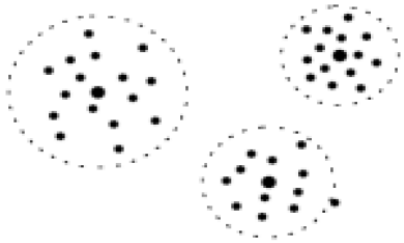
Типы кластерных структур:

1. С расстояниями внутри кластера меньше, чем между ними.

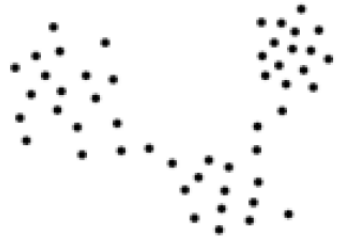


2. Ленточные кластеры.

3. Кластеры с центром.



4. Соединенные кластеры.



5. Наложенные на фон разреженных объектов.

6. Перекрывающие кластеры.

Каждый метод кластеризации имеет свои ограничения и выделяет кластеры лишь некоторых типов. Понятие «Тип кластерной структуры» зависит от метода и не имеет формального определения. Самыми известными методами кластеризации являются Метод К-средних, Метод С-средних, Сеть Кохонена и другие. Которые теперь разберем поподробнее.

Алгоритм K-means

1. Инициализировать центры каждого кластера случайным образом – $\boldsymbol{\mu}$.
2. Вычислить расстояния $\boldsymbol{\rho\ (x_i, \mu_k)}$ для каждого центра $\boldsymbol{\mu_k}$.
3. Для каждого центра из множества объектов $\boldsymbol{X_k}$ отобрать подмножество с минимальным расстоянием до центра. Таким образом, сформировать кластеры.
4. Вычислить новые центры кластеров, как среднее значение всех точек кластера: $\boldsymbol{\mu_k = \frac{1}{s_k}\sum X_k}$, где $\boldsymbol{s}$ – количество точек в кластере.
5. Если значения центров изменились (сдвинулись), то повторять пункт 2, в ином случае конец работы.

Функция оценки качества кластеризатора:

$$\boldsymbol{Q = c \cdot \frac{d_i}{d_0}}$$

где $\boldsymbol{c}$ – количество кластеров, $\boldsymbol{d_i}$– среднее внутрикластерное расстояние, $\boldsymbol{d_i}$ – среднее межкластерное расстояние. Задача обучения – минимизация

функционала качества.

Недостатками являются:

- Отсутствие гарантии достижения глобального минимума, а только одного из локальных минимумов;
- Результат зависит от выбора исходных центров кластеров, и их оптимальный выбор неизвестен;
- Число кластеров надо заранее знать;
- Кластерный анализ не всегда дает удовлетворительные результаты;
- Необходим эвристический подбор параметров алгоритма, в частности начальных значений центра.

Алгоритм C-means

Для заданного множества входных векторов X_kи N выделяемых кластеров c_j предполагается, что любой X_k принадлежит любому c_j с принадлежностью $\mu_{jk}:[0,1]$, где j– номер кластера, а k – номер входного вектора.

1. Инициализация: необходимое количество кластеров N, $2 < N < K$; фиксированный параметр q (обычно = 1,5); матрица принадлежности на нулевой итерации $U^{(0)} = (\mu_{jk})^{(0)}$ объектов, с учетом заданных начальных центров кластеров c_j.
2. Регулирование позиций центров кластеров: На *t*-м итерационном шаге при неизвестной матрице вычисляется $c_j^{(t)}$.
3. Корректировка значений принадлежности μ_{jk}: учитывая известные $c_j^{(t)}$ вычисляются $\mu_{jk}^{(t)}$.
4. Остановка алгоритма. Если матрица U не меняется, остановка, иначе пункт 2.

Проблема, которая ставится в k-means – неопределенность принадлежности к кластерам разным в равной степени или непринадлежности ни к одному из них – решается в данном алгоритме, определяющем вероятность принадлежности к тому или иному алгоритму. Здесь кластеры представляют собой нечеткие множества, границы между которыми также нечеткие.

Вычисляется вероятность принадлежности объекта к кластеру, и данная степень определяется расстоянием объекта до кластерных центров. *Целью алгоритма является минимизация всех взвешенных расстояний.*

Поиск ассоциативных правил

Обозначения:

X – пространство объектов;

$F = \{f_1,\ldots,f_k\}$, $f_j: X \to \{0, 1\}$ – бинарные признаки;

$X^n = \{x_1, \ldots, x_n\} \subset X$ – обучающая выборка;

Каждому подмножеству $\varphi \subseteq F$ соответствует конъюнкция $\varphi(x) = \wedge_{f\in\varphi} f(x)$, $x \in X$.

Если $\varphi(x) = 1$, то «признаки из φсовместно встречаются у x». Частота встречаемости φ в выборке X^n.

$$v(\varphi) = \frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)$$

Если $v(\varphi) >= \delta$, то «набор φ частный». Параметр минимальная поддержка, MinSupp.

Ассоциативное правило $\varphi \to y$ – это пара непересекающихся наборов, $\varphi, y \subseteq F$ таких, что:

1. Наборы φиyсовместно часто встречаются, $v(\varphi \cup y) >= \delta$;

2. Если встречается φ, то часто встречается также и y, $v(y|\varphi) \equiv \frac{v(\varphi\cup y)}{v(\varphi)} \geq \varkappa$

$v(y|\varphi)$ – значимость правила.

Параметр δ – минимальная поддержка, MinSupp.

Параметр $\varkappa$ – минимальная значимость, MinConf.

Свойство анти-монотонности

Это нахождение элементов, встречающихся наиболее часто. Типичным подходом является обычный перебор всех возможных элементов. И предназначен для снижения размера всего пространства поиска. То есть, с ростом размера набора элементов поддержка уменьшается, либо остается точно

такой же.

Для любых $\psi, \varphi \in F$ из $\varphi \supset \psi$ следует $v(\varphi) >= v(\psi)$

Алгоритмом, позволяющим эффективно находить ассоциативные правила, является алгоритм Apriori:

На входе получающий X^n – обучающую выборку; минимальную поддержку – δ, минимальную значимость – $\varkappa$. На выходе выдающий $R = \{(\varphi, y)\}$ – список ассоциативных правил.

1. Множеств всех частых исходных признаков: $G1 := \{f \in F | v(f) >= \delta\}$;
2. Для всех $j = 2,\ldots,n$
3. Множество всех частых наборов мощности j:

$$G_j := \{\varphi \cup \{f\} | \varphi \in G_{j-1}, f \in G_1 \backslash \varphi, v(\varphi \cup \{f\}) \geq \delta\};$$

4. Если $Gj = \varnothing$ то
5. Выход из цикла по j;
6. $R := \varnothing$
7. Для всех $\psi \in G_j, j = 2,\ldots,n$
8. AssocRules ($R, \psi, \varnothing$); // R – список ассоциативных правил, (φ,y) – ассоциативное правило.
9. ПРОЦЕДУРА AssocRules(R, ψ, y);
10. Для всех $f \in \varphi : id_f > \max_{g \in y} id_g$ // чтобы избежать повторов y, id_f – порядковый номер признака f в $F=\{f_1,\ldots,f_n\}$
11. $\varphi^I := \varphi \setminus \{f\}$; $y^I := y \cup \{f\}$;
12. Если $v(y^I, \varphi^I) >= \varkappa$ то
13. Добавить ассоциативное правило (y^I, φ^I) в список R;
14. Если $|\varphi^I| > 1$ то
15. AssocRules (R, y^I, φ^I);

Контрольные вопросы для повторения и самопроверки темы №7

1. Что изучает статистика?
2. К каким алгоритмам классификации относится метод ближайших соседей?

3. Что является целью кластеризации?
4. С помощью какого алгоритма можно найти ассоциативное правило?
5. Что подразумевается под определением "статистический вывод"?
6. Чем отличаются ошибки первого и второго рода?
7. Что является результатом решения задачи регрессии?
8. Что такое α-error?
9. Какая основная цель статистического анализа?
10. Что такое генеральная совокупность?
11. Как оценить качество обучения модели?

Литература по теме №7

1. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. М.: ДМК Пресс, 2015.
2. Домингос П. Верховный алгоритм: как машинное обучение изменит наш мир. М.: Манн, Иванов и Фербер, 2016.

Глоссарий по теме №7

Методы – это общая совокупность действий, направленных на решение задачи.

Анализ данных – это процесс, посредством которого извлекается необходимая для обработки информация, впоследствии предоставляющая решение поставленной перед ним задачи/проблемы.

Статистические методы анализа данных – это совокупность неких шагов (действий), занимающихся фильтрацией и обработкой предоставленных статистических данных некоего объекта изучения, с целью получения решения поставленной задачи.

Генеральная совокупность – полный набор объектов, связанных с поставленной перед изучением проблемой.

Статистический вывод – это утверждение о том, что представляют собой законы, лежащие в основе изучаемой генеральной совокупности.

Оценивание параметров законов – вычисление по выборке точечных и

интервальных оценок.

Проверка статистических гипотез – установление справедливости данного утверждение по отношению к параметрам генеральной совокупности.

Ошибка первого рода – обнаружение различий или связей, на самом деле не существующих.

Ошибка второго рода – не обнаружение различий или связей, на самом деле существующих.

Метрический классификатор – алгоритм, который вычисляет оценку сходства предоставленных объектов.

Линейный классификатор – это алгоритм классификации, который основан на построении разделяющей линейной поверхности.

ROC-кривая («receiver operating characteristic») – графическая характеристика бинарного классификатора, показывающая зависимость TPR (доля верных положительных классификаций, или полнота) от FPR (доля ложных положительных классификаций).

Кластеризация – задача разделить множество объектов на различные группы, которые называются кластерами.

Критерий согласия – проверка подчинения предполагаемому закону распределения.

Статистика – это отрасль знаний, описывающая вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных; изучение количественной стороны массовых общественных явлений в числовой форме; совокупность цифровых данных, представляемых в отчётности предприятий, организаций, отраслей экономики, публикуемых в сборниках, справочниках, периодических изданиях; отрасль общественных наук, специальная научная дисциплина, изучаемая в учебных заведениях.

ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ п/п	Автор	Название основной и дополнительной учебной литературы, необходимой для освоения дисциплины	Выходные данные	Количество экземпляров в библиотеке ДГУНХ
I Основная учебная литература				
1.	Нестеров С.А.	Интеллектуальный анализ данных средствами MS SQL Server 2008	2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 338с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электронным изданиям от 1.10.2018г.
2	Кухаренко Б.Г.	Интеллектуальные системы и технологии: учебное пособие	Москва: Альтаир: МГАВТ, 2015. – 115с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электроннымизданиямот 1.10.2018г.
3	Шипунов А.Б., Балдин Е.М., Волкова П.А.	Наглядная статистика. Используем R!	ISBN: 978-5-97060-094-8	Бесплатная электронная версия в сети «Интернет».
4	Мастицкий С.Э., Шитиков В.К.	Статистический анализ и визуализация данных с помощью R	ISBN: 978-5-97060-301-7	Бесплатная электронная версия в сети «Интернет».
5	Шитиков В.К., Мастицкий С.Э.	Классификация, регрессия и другие алгоритмы Data Mining с использованием R.	-	Бесплатная электронная версия в сети «Интернет».
6	Савина Е.В.	Практикум по информатике (MS Windows, word, excel) для направления бакалавриата: Прикладная информатика (в экономике), Математика и компьютерные науки, Информационная безопасность, Бизнес-информатика	Махачкала: ДГУНХ. 2016	48 экз.
II Дополнительная учебная литература **А) Дополнительная учебная литература**				
1.	Соловьев Н., ЧернопрудоваЕ., Лесовой Д.	Основы теории принятия решений для программистов: учебное пособие	Оренбург: ОГУ, 2012. – 187с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электронным изданиям от 1.10.2018г.

2.	Чубукова И.А.	Data Mining	2-е изд., испр. – Москва: Интернет-Университет Информационных Технологий, 2008. – 383с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электронным изданиям от 1.10.2018г.
3.	Белов В.С.	Информационно-аналитические системы: основы проектирования и применения: учебно-практическое пособие	2-е изд., перераб. и доп. – Москва: Евразийский открытый институт, 2010. – 111с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электронным изданиям от 1.10.2018г.
4.	Кравченко Ю.А., Кулиев Э.В., Заруба Д.В.	Тенденции развития компьютерных технологий: учебное пособие	Таганрог: Издательство Южного федерального университета, 2017. – 107с.	15000 в соответствии с договором № 149-09/2018 на оказание услуг по предоставлению доступа к электронным изданиям от 1.10.2018г.

Б) Официальные издания: сборники законодательных актов, нормативно-правовых документов и кодексов РФ

1. ГОСТ Р ИСО/МЭК 15288-2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем. 2006 г.
2. ГОСТ Р ИСО/МЭК 17799-2005. Информационная технология.
3. ГОСТ Р ИСО 11442-2014. Техническая документация на продукцию. Управление документацией. 2015 г.
4. Федеральный закон от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и о защите информации" (с изменениями и дополнениями).
5. ГОСТ 34.320-96. Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы. 2001 г.

В) Периодические издания

1. «Windows IT Pro/RE» - профессиональное издание на русском языке, целиком и полностью посвященное вопросам работы с продуктами семейства Windows и технологиям компании Microsoft.
2. «Информационные технологии» - рецензируемый научный журнал.
3. «Вестник компьютерных и информационных технологий» - рецензируемый научный журнал.
4. «Программная инженерия» - рецензируемый научный журнал.