# Глава 1 Литературный обзор

Данная глава последовательно даёт обзор источников информации из области современных баз данных, фреймворков (программных каркасов для построения информационных систем) и кластерного анализа данных, в качестве которых рассматриваются литература, электронные документы, компьютерные программы.

Активное развитие сферы информационных технологий значительно отразилось и на посвящённом ей литературном фонде: практически всегда можно найти несколько книг от авторитетных изданий на любую тематику. В то же время видна тенденция перехода формата обучения от печатных источников к видеоурокам и интерактивным обучающим курсам из-за более наглядного представления конечного результата и возможности сразу же приступить к выполнению практических заданий.

Также необходимо отметить быстрое устаревание источников информации на тему информационных технологий: к актуальным можно отнести источники за последние 5-10 лет. Темп развития отрасли также сказывается и на объёме источников информации: значительная часть востребованных знаний представлена в виде коротких статей на веб-ресурсах, ссылаться на которые невозможно по ряду объективных причин.

Источники информации на тему кластерного анализа данных представлены немногочисленными книгами, больше информации можно найти в формате статей. Зачастую вопрос кластеризации данных раскрывается в источниках на смежную тематику.

## 1.1 Обзор современных СУБД

Первоочередной задачей является определение наиболее подходящей системы управления базами данных (СУБД) для хранения знаний.

На данный момент выделяют следующие основные группы СУБД:

- реляционные. Наиболее популярные представители: Oracle, MySQL, MS SQL Server, PostgreSQL, DB2, SQLite;

- столбцовые: C-Store;

- семейство столбцов: Cassandra, Hbase, Hypertable;

- «ключ-значение». Redis, Memcached, Riak, Amazon DynamoDB;

- документоориентированные. MongoDB, Couchbase, CouchDB;

- графовые: Neo4j;

- мультимодельные: OrientDB, ArangoDB [Elmasri].

Вкратце рассмотрим основные особенности каждой из групп.

Один из основных идеологов реляционного подхода к базам данных Эдгар Кодд предложил использовать для обработки данных аппарат теории множеств. Он продемонстрировал, что представление данных является совокупностью двумерных таблиц особого вида, называемых в математике «отношением» (англ. relation). Основными понятиями реляционных БД являются сущность, атрибут, первичный и внешний ключ. На практике сущностями являются таблицы, атрибутами — колонки таблиц, а ключи используются для установления отношений между таблицами [Кириллов].

Реляционные БД поддерживают три типа связи между сущностями: один-к-одному, один-ко-многим, многие-ко-многим. Связи между записями разных сущностей не поддерживаются.

Помимо теоретической работы над реляционным подходом к БД, Эдгар Кодд также создал и практический инструмент для работы с отношениями — реляционную алгебру. Каждая операция данной алгебры использует одну или несколько таблиц в качестве операндов и в итоге создаёт новую таблицу [Кириллов].

Столбцовые СУБД возникли вследствие недостатков производительности реляционных СУБД. В отличие от реляционной СУБД, где вся база данных хранится в одном файле, столбцовые СУБД хранят значения столбцов отдельно друг от друга, в разных файлах. Такой подход ограничивает загружаемый объём данных за каждый запрос (в реляционных СУБД записи загружаются со всеми столбцами), что снижает время совершения запросов и занимаемое дисковое пространство. Столбцовые СУБД используют язык запросов SQL [Фаулер].

Следующую группу направлений развития баз данных специалисты относят к так называемым NoSQL-базам данных. Мы не рассматриваем подробно происхождение и значение этого термина (во многом из-за его неоднозначности), отметим его основное отличие — базы данных этой группы не используют SQL в качестве языка запросов по умолчанию (однако могут поддерживать ради совместимости) и привычную табличную структуру представления данных.

Отличительной особенностью баз данных «семейств столбцов» (англ. column-family, тж. wide-column) является распределение данных как на основе строк, так и на основе столбцов. При этом происходит объединение столбцов в группы или в семейства с целью показать, какие столбцы лучше хранить вместе. Базы данных данного типа позволяют каждой строке иметь различную структуру столбцов без каких-либо ограничений со стороны схемы, размер строк также может быть разным. В качестве параллели с базами данных «семейства столбцов» можно привести двумерный массив данных. Преимуществом данного вида БД является возможность легко добавить новые столбцы к существующим строкам; таблица может быть разреженной (множество null-значений) без каких-либо накладных расходов. Структурно БД «семейства столбцов» занимают положение между реляционными СУБД и БД «ключ-значение» [Редмонд]

Хранилища данных «ключ-значение» состоят из простых пар «ключа» и ассоциируемого с ним «значений», которое обычно является массивом данных. Подобные базы данных обеспечивают структуру, которая позволяет хранить и читать значения на основе «ключа». «Ключ» обычно является строкой и во многих отношениях схож с первичным ключом в реляционной БД. Отдельные записи в «значении» не отслеживаются и не различаются, поэтому при необходимости их изменения необходимо обновление всего «значения». Для базы данных «значение» представляет собой произвольный набор байтов, любая обработка которого отводится на использующую БД систему. Единственные операции, которые позволяют осуществлять БД «ключ-значение», - это put (для записи «значения»), get (для чтения «значения») и delete (для удаления пары «ключ-значение»). Операция обновления данных не поддерживается. [Hoffner]

Документоориентированная СУБД хранит данные в виде структурированных документов, обычно в формате XML или JSON. При этом определение «документоориентированная СУБД» не подразумевает какую-либо специфику насчёт модели хранения: документоориентированные СУБД могут выполнять ACID-транзакции или другие функции традиционных реляционных СУБД, хотя популярные документоориентированные обеспечивают относительно скромную транзакционную поддержку.

Документоориентированные базы данных, позволяя описывать данные без использования схемы, возможно, являются золотой серединой между жёсткой схемой реляционных баз данных и свободных от схемы хранилищ «ключ-значение». Сочетание с практикой веб-разработки вылилось в появление JSON-баз данных (MongoDB в частности), которые стали выбором по умолчанию для многих веб-разработчиков [Harrison G. Next Gen.].

Графовая база данных состоит из набора вершин (узлов, сущностей) и граней (связей, отношений, рёбер). Узлы воспринимаются как объекты со свойствами, между которыми моделируются отношения с помощью граней, которые также могут иметь свойства. Отношения имеют направления, на их основе происходит организация узлов, что позволяет единожды записать данные и затем по-разному их интерпретировать [Jordan] [Фаулер NoSQL].

Графовая структура позволяет представить данные в более естественном виде без искажений, как это может произойти в реляционных базах данных, а также применить различные типы графовых алгоритмов к этим данным. Одна из ключевых особенностей графовых БД — возможность обхода графа по его узлам и граням, перемещения от одного узла к другому, следуя направленным отношениям. Эта возможность называется «index free adjacency» (примерно переводится как смежность без индекса), смысл которой заключается в поиске прилежащих узлов без использования поиска по индексу, что значительным образом сказывается на производительности [Bruggen].

## 1.2 Текущее состояние области кластерного анализа данных

Термин «кластерный анализ» предложен К. Трионом в 1939 г. (англ. cluster — гроздь, скопление, пучок). Синонимами могут выступать следующие выражения: автоматическая классификация, таксономия, распознавание без обучения, распознавание образов без учителя, самообучение и др. [Мандель Кластерный анализ].

Кластер-анализ — это способ группировки многомерных объектов, основанный на представлении результатов отдельных наблюдений точками подходящего геометрического пространства с последующим выделением групп как «сгустков» этих точек [Мандель Кластерный анализ].

Рассматривая область кластерного анализа данных, нужно отметить многообразие алгоритмов. На данный момент не существует общепринятой классификации алгоритмов кластеризации. Отдельные исследователи предлагают различные модели классификации, однако среди каждой из них можно выделить несколько основных направлений.

Кластерный анализ используется в различных областях (экономике, биологии, информатике, психологии и т. д.), что обусловило параллельное развитие алгоритмов кластеризации и разное их наименование при одинаковых принципах работы. Тем не менее цель всех алгоритмов кластеризации — выделение устойчивых групп элементов с какими-либо схожими характеристиками [Суслов].

Формальная постановка задачи: дано множество документов T, каждый из которых представлен набором атрибутов. Необходимо построить множество кластеров C и отображение F множества T на множество C, то есть F : T → C. Отображение F является моделью данных, служащей решением задачи [Барсегян].

Множество T определяется следующим образом:

где tj – текущий рассматриваемый документ.

Каждый из документов имеет набор атрибутов:

Каждая переменная xh может принимать значения из некоторого множества:

В кластерном анализе используются следующие метрики, или расстояния:

1. евклидово расстояние;
2. взвешенное евклидово расстояние
3. расстояние городских кварталов (также манхэттенское расстояние, расстояние Хэмминга)
4. расстояние Чебышёва
5. расстояние Махаланобиса
6. пиковое расстояние

Иерархические алгоритмы: аггломеративные, дивизимные (BIRCH, CURE, ROCK, Chameleon, Echidna)

Вероятностные (K-means, K-medoids, K-modes, PAM, CLARANS, CLARA, FCM)

Плотностные (DBSCAN, OPTICS, DBCLASD, DENCLUE)

Сеточные (Wave-Cluster, STING, CLIQUE, OptiGrid)

Моделируемые (EM, COBWEB, CLASSIT, SOMs)

Иерархические алгоритмы разделяются на две группы: аггломеративные и дивизимные. Аггломеративные алгоритмы более популярны и многочисленны, нежели дивизимные, представляющие скорее исторический интерес к этапам развития кластерного анализа.

Иерархическим алгоритмам кластеризации присуще плохое масштабирование (O(n2) и хуже). В общем случае сложность аггломеративной кластеризации равна O(n3), что делает её слишком медленной для больших наборов данных. Сложность дивизимной кластеризации со всесторонним поиском равна O(2n), что хуже. Однако в некоторых исключительных случаях сложность аггломеративных алгоритмов равна O(n2): SLINK для одинарной связи и CLINK для полносвязной кластеризации [Sasirekha].

Результаты иерархического анализа обычно представляются в виде дендрограммы.

Рассматривая кластерный анализ текстовых документов, нужно выделить возможные типы данных: численный, категориальный (например, пол, нация, раса, почтовый индекс), текстовый, временной

## 1.3 Обзор существующих программных комплексов для осуществления кластерного анализа

Основная цель данной работы – предоставить анализ существующих программных комплексов, в функциональность которых входит какой-либо вид кластерного анализа, выделить их особенности, преимущества и недостатки. Программные комплексы отбираются на основе их популярности и доступности для рядового пользователя.

**RStudio** – интегрированная среда разработки с открытым исходным кодом для языка программирования R, предназначенная для статистической обработки данных и работы с графикой. Язык программирования R по умолчанию включает в себя минималистичный графический интерфейс, RStudio добавляет следующие возможности: как консоль для выполнения команд, продвинутый редактор кода, средства отладки, браузер объектов, интеграция с документацией, средства для управления графиками [Verzani].

Язык программирования R использует менеджер пакетов CRAN, поэтому пользователю предоставляется на выбор около сотни пакетов, предназначенных для работы с кластерным анализом. Среди них представлены методы иерархического (AGNES, CLARA - пакеты cluster, genie, hybridHclust, isopam, protoclust и др.), вероятностного (KMeans, KCentroid, пакеты kmeans, cluster, apcluster, clustMixType и др.), модельно-ориентированного (EM-алгоритм, алгоритмы на основе Байесовской оценки решений) и другие.

Преимущества RStudio: open-source, многообразие пакетов, кроссплатформенность, результат напрямую зависит от навыков пользователя.

Недостатки RStudio: сложность освоения – нужно знать язык программирования R и изучать документацию пакетов.

**RapidMiner** – программный комплекс, который представляет собой платформу для работы с данными не посредством программирования, а управления блоками, обозначающими ту или иную операцию. Блоки называются «операторами», имеют вход и выход, называемые «портами». На вход принимаются данные, на выход поступают обработанные данные [Крахалёв].

RapidMiner поддерживает следующие виды операторов, осуществляющих кластерный анализ: K-means, X-means, K-medoids, DBSCAN, EM-алгоритм, аггломеративная иерархическая кластеризация, случайная кластеризация.

Преимущества RapidMiner: open-source, кроссплатформенность, удобный и наглядный конструктор операторов.

Недостатки RapidMiner: не поддерживается русский язык.

**ClusterEnG** – веб-ресурс, предоставляющий возможность осуществить кластерный анализ с графическим отображением результатов. ClusterEnG поддерживает следующие методы кластеризации: K-means, K-medoids, Affinity Propagation, Mixture model, спектральная и иерархическая кластеризация [Manjunath].

Для того чтобы осуществить кластерный анализ с помощью ClusterEnG, необходимо зайти на сайт проекта, загрузить файл с данными (максимальный объём – 1Гб), выбрать нужный алгоритм и указать параметры кластеризации (число кластеров или эпсилон), если того требует алгоритм.

Преимущества ClusterEnG: находится в свободном доступе, можно произвести кластерный анализ онлайн.

Недостатки ClusterEnG: данные передаются по сети – при невысокой скорости соединения и большом объёме данных будут сложности, не поддерживается русский язык, небольшие возможности настройки работы алгоритмов.

**Weka** – программный комплекс, созданный для решения задач машинного обучения (machine learning). Weka использует пакетный менеджер, позволяющий подключать алгоритмы для обработки данных различных категорий (классификация, предобработка, регрессия, кластеризация и другие) на различных языках программирования (R, Python, Clojure) при поддержке популярных технологий (JDBC, Cassandra, Hadoop, Spark).

Осуществление кластерного анализа в Weka происходит путём выбора файла с данными, выбора алгоритма кластеризации и его настройки.

Преимущества Weka: open-source, кроссплатформенность, многообразие пакетов и поддерживаемых технологий.

Недостатки Weka: не поддерживается русский язык.

**NCSS** – программный комплекс, включающий в свою функциональность около 500 инструментов для работы с графиками и диаграммами и интеллектуального анализа данных. Работа с NCSS очень похожа на работу с редактором электронных таблиц (например, Excel): по умолчанию данные представляются в виде электронной таблицы, затем пользователь выбирает в меню метод обработки данных, настраивает его параметры в появившемся окне, и программа демонстрирует результат в виде отчёта.

NCSS поддерживает следующие виды и методы кластеризации: нечёткая, регрессионная, иерархическая аггломеративная кластеризация, K-means, K-medoids.

Преимущества NCSS: огромное количество инструментов для обработки данных, простота освоения.

Недостатки NCSS: платная лицензия (free trial – 30 дней), не поддерживается русский язык.

**SPSS Statistics** – программный комплекс от компании IBM, схожий по своим возможностям с комплексом NCSS. Данные также представляются в виде электронной таблицы, на выбор предоставляется около сотни инструментов для работы с данными. Кластерный анализ осуществляется аналогично. SPSS поддерживает иерархическую и двухэтапную кластеризацию и метод K-means.

Преимущества SPSS: простота освоения, поддерживается русский язык.

Недостатки SPSS: запутанная документация, платная лицензия (free trial – 14 дней).

**Deductor** – разработанный в России программный комплекс, предлагающий продвинутую аналитику без программирования, схожий по концепции с RapidMiner (однако продукты различаются в способе импорта данных). Рабочий процесс в Deductor заключается в импорте файла с данными и последующем выборе и настройки метода обработки данных, результат сразу можно представить в различных способах представления.

Deductor поддерживает методы K-means и G-means, CLOPE, EM-алгоритм, самоорганизующиеся карты Кохонена.

Преимущества Deductor: простота освоения, поддерживается русский язык, поддержка редких методов кластеризации.

Недостатки Deductor: бесплатная версия ограничена в возможностях.

**Statistica** - программный комплекс от компании Dell, ориентированный на решение задач бизнес-аналитики: анализ, управление, добыча данных. Statistica может быть интегрирована с языками программирования R и Python.

Аналогично SPSS, Statistica поддерживает иерархическую и двухэтапную кластеризацию и метод K-means.

Преимущества STATISTICA: множество поддерживаемых форматов импорта, большое разнообразие инструментов для анализа и визуализации.

Недостатки STATISTICA: не поддерживается русский язык, платная лицензия (free trial – 30 дней).

**Orange** - программный комплекс, предоставляющий возможность обработки данных с помощью функциональных блоков без применения программирования, что делает его схожим с комплексом RapidMiner, кластерный анализ осуществляется аналогично. Orange поддерживает подключение скриптов на языке Python.

Преимущества Orange: простота освоения, open-source, кроссплатформенность, удобный и простой конструктор операторов.

Недостатки Orange: не поддерживается русский язык.

Подводя итоги, можно отметить такие программные комплексы, как RapidMiner, Weka и Orange, по большинству параметров не уступающие дорогостоящим бизнес-решениям в лице NCSS и SPSS. Более продвинутым пользователям, обладающим минимальным опытом программирования, можно посоветовать изучить язык программирования R и использовать RStudio для более полного контроля процесса кластерного анализа.