**Оглавление**

[А. Специальная часть проекта 5](#_Toc358042700)

[1. Введение 5](#_Toc358042701)

[1.1. Актуальные проблемы кластерного анализа 7](#_Toc358042702)

[1.2. Задачи кластерного анализа 9](#_Toc358042703)

[1.3. Этапы кластерного анализа 9](#_Toc358042704)

[1.4. Цели кластеризации для выработки рекомендаций 10](#_Toc358042705)

[1.5. Построение ценовых (весовых) моделей 11](#_Toc358042706)

[2. Техническое задание 12](#_Toc358042707)

[2.1. Цель и назначение разработки 12](#_Toc358042708)

[2.2. Этапы разработки 12](#_Toc358042709)

[2.3. Основные требования к программному обеспечению 13](#_Toc358042710)

[3. Обзор существующих решений 14](#_Toc358042711)

[3.1 Иерархические методы кластеризации 14](#_Toc358042712)

[3.2. Неиерархические (итеративные) методы кластеризации 18](#_Toc358042713)

[3.3. Сравнительный анализ иерархических и неиерархических методов кластеризации 21](#_Toc358042714)

[3.4. Сеточные алгоритмы 22](#_Toc358042715)

[4. Обзор существующих систем кластеризации 26](#_Toc358042716)

[4.1. Коммерческие инструменты: 26](#_Toc358042717)

[4.2. Выводы 30](#_Toc358042718)

[5. Предлагаемые методы и математическое обеспечение 31](#_Toc358042719)

[5.1. Методы кластеризации 31](#_Toc358042720)

[5.2. Метод вычисления ценовых моделей 38](#_Toc358042721)

[Структура системы 39](#_Toc358042722)

[6.1. Принципиальная схема 39](#_Toc358042723)

[6.2 Программная реализация библиотеки 40](#_Toc358042724)

[6.3. Форматы входных и выходных данных: 44](#_Toc358042725)

[Б. Конструктивно-технологическая часть проекта 45](#_Toc358042726)

[1. Технология программирования 45](#_Toc358042727)

[1.1. Модели на основе инженерного подхода 46](#_Toc358042728)

[1.2. Модели, учитывающие специфику разработки ПО 47](#_Toc358042729)

[1.3. Современные модели 49](#_Toc358042730)

[1.4. Модели быстрой разработки. 50](#_Toc358042731)

[1.5. Адаптированные и комбинированные модели. 53](#_Toc358042732)

[1.6 Выводы 54](#_Toc358042733)

[2. Выбор языка программирования 54](#_Toc358042734)

[2.1. Анализ инструментальных средств 55](#_Toc358042735)

[3. Выводы 58](#_Toc358042736)

[В. Охрана труда 59](#_Toc358042737)

[1.1 Расчет защитного заземления 59](#_Toc358042738)

[1.2 Электробезопасность рабочего места с ПЭВМ 62](#_Toc358042739)

[Г. Экологическая часть 66](#_Toc358042740)

[2.1. Микроклимат 66](#_Toc358042741)

[2.2 Защита от шума 69](#_Toc358042742)

[Д. Решение задачи на ЭВМ 76](#_Toc358042743)

[1. Структура классов 76](#_Toc358042744)

[2. Пример функционирования 78](#_Toc358042745)

[Е. Заключение 79](#_Toc358042746)

[Ж. Список использованной литературы 80](#_Toc358042747)

# А. Специальная часть проекта

## 1. Введение

В последнее время информация, растущая в колоссальных объёмах рождает потребность в обработке больших объёмов данных (Big Data). В этом направлении большое место отведено интеллектуальному анализу данных (Data Mining). Это направление включает в себя методы, отличные от классического анализа, основанные на моделировании, вероятностных, и решающие задачи обобщения, ассоциирования и отыскания закономерностей. В большой степени развитию этой дисциплины способствовало проникновение в сферу анализа данных идей, возникших в теории искусственного интеллекта.

В данной работе я хотел бы рассмотреть частную задачу подобного анализа, а именно, задачу кластерного анализа, известную как задача автоматической группировки объектов. Решение планируется использовать для разработки системы рекомендаций в системах принятия решений.

Главной задачей кластерного анализа является выделение необходимого числа групп объектов схожих между собой внутри группы и максимально отличных от экземпляров других классов. Подобный анализ широко применяется в информационных системах для отыскания закономерностей в данных.

Список областей, в которых применяется кластеризация очень большой: это работа с изображениями, маркетинговые исследования, прогнозирование, текстовый анализ, обнаружение мошеннических действий и многие другие[28]. Сегодня кластеризация является первым шагом анализа данных. Задачи кластеризации формируются в различных научных направлениях: в статистике, оптимизации, машинном обучении, что породило множество различных синонимов понятия кластера – класс, таксон, сгущение. Сегодня уже применяются несколько десятков алгоритмов разбиения и множество их модификаций. Зачастую данные с которыми приходится работать этим алгоритмам обусловлены такими особенностями как высокая размерность и сверхбольшой объём данных (миллионы записей в тысячах полей), так же содержание большого количества числовых (numerical), которые способны упорядочиваться в пространстве, и категорийных (categorical), неупорядочивающихся, атрибутов. Атрибуты получают значения согласно типу шкалы, выбор которой является отдельной задачей.

В своей основе алгоритмы кластеризации сравнивают между собой объекты на основе различных мер сходства или близости, величины, имеющей предел и возрастающей по мере схожести объектов. Эти меры сходства выбираются в зависимости от задачи и шкалы измерений[36]. Для числовых атрибутов очень часто применяют евклидово расстояние: ,

а категорийные атрибуты сравнивают при помощи меры Чекановского-Серенсена и Жаккара .

Необходимость обрабатывать огромные объёмы данных позволила сформулировать ряд требований для алгоритма кластеризации. Рассмотрим эти требования. Минимизация итераций обращения к данным

1. Ограниченность ресурсов, в частности памяти системы
2. Восстанавливаемость алгоритма
3. Работа алгоритма с базой данных в режиме однонаправленного курсора.

Выполнение данных условий, в особенности второго пункта, определяет алгоритм как масштабируемый. Алгоритм называют масштабируемым, если при неизменной емкости оперативной памяти с увеличением числа записей в базе данных время его работы растет линейно.

Трудно соблюсти баланс между высоким качеством кластеризации и масштабируемостью. Поэтому в идеале в арсенале Data Mining должны присутствовать как эффективные алгоритмы кластеризации микромассивов (Microarrays), так и масштабируемые для обработки сверхбольших баз данных (Large Databases).

### 1.1. Актуальные проблемы кластерного анализа

Несмотря на большое число исследований в области кластерного анализа, в этой области существует ряд актуальных проблем. Можно сформулировать основные[28]:

1. Обоснование качества результатов. Проблема заключается в том, что один и тот же объект может быть классифицирован в различные группы вне зависимости от его внутренних свойств, а в связи с различными экспертными данными или различным построением системы. Для избегания этого необходимо разрабатывать и вводить актуальные критерии качества.
2. Анализ большого числа разнотипных данных порождает методологическую проблему выбора метрик. Так же увеличение числа объектов даже однотипных данных может повлечь за собой неразличимость расстояний.
3. Нелинейность взаимосвязей. Классические методы снижения размерности в кластерном анализе направлены на линейной взаимосвязи между переменными. Для поиска более сложных зависимостей необходимо переходить к ядерным методам.
4. Проблема поиска глобального экстремума функции критерия качества. Критерий качества, как правило, является функцией, зависящей от большого числа факторов, нелинейным, обладающим множеством локальных экстремумов. Для нахождения кластеров необходимо решить сложную комбинаторную задачу поиска оптимального варианта классификации. Поэтому алгоритм полного перебора вариантов имеет трудоемкость, экспоненциально зависящую от размерности. Если число групп заранее неизвестно, то переборная задача становится еще сложнее. Таким образом, при увеличении размерности таблиц данных происходит «комбинаторный взрыв». Классические алгоритмы кластерного анализа осуществляют направленный поиск в сравнительно небольшом подмножестве пространства решений, используя различного рода априорные ограничения (на число кластеров или их форму, на порядок включения объектов в группы и т.д.). При этом нахождение строго-оптимального решения не гарантируется. Для поиска оптимального решения применяются более сложные методы, такие как генетические (эволюционные) алгоритмы, нейронные сети и т.д. Существуют экспериментальные исследования, подтверждающие преимущества таких алгоритмов перед классическими алгоритмами. Однако и при использовании эволюционных методов возникают проблемы, связанные со спецификой решаемой задачи кластер-анализа: с трудностью интерпретации используемых операторов рекомбинации и кроссовера.
5. Неустойчивость результатов кластеризации. Зачастую результаты группировки могут сильно меняться в зависимости от выбора начальных условий, порядка объектов, параметров работы алгоритмов. Различными авторами предлагаются способы повышения устойчивости группировочных решений, основанные на применении ансамблей алгоритмов. При этом используются результаты группировки, полученные различными алгоритмами, или одним алгоритмом, но с разными параметрами настройки, по различным подсистемам переменных и т.д. После построения ансамбля проводится нахождение итогового коллективного решения.
6. Недостаточность знаний об объекте. Существует проблема трудноформализуемых областей, в которых становится затруднительным создание модели объекта. В таком случае становится затруднительным применение алгоритмов, основывающихся на представлении класса, как набора распределённых в пространстве переменных.
7. Проблема представления результатов. Помимо хорошей прогнозирующей способности для любого алгоритма анализа данных важно, насколько понятными и интерпретируемыми являются его результаты. Для улучшения интерпретируемости решений можно использовать логические модели. Такого рода модели используются для решения задач распознавания образов и прогнозирования количественных показателей, например, в методах построения решающих деревьев или логических решающих функций.

### 1.2. Задачи кластерного анализа

Кластерный анализ выполняет следующие основные задачи:

* Исследование схем группировки объектов;
* Выработка гипотез на базе исследований данных;
* Подтверждение гипотез и исследований данных;
* Определение присутствия групп внутри данных.

### 1.3. Этапы кластерного анализа

Независимо от предмета изучения применение кластерного анализа предполагает следующие этапы:

1. Формирование выборки для кластеризации;
2. Выделение признакового пространства;
3. Выбор меры сходства (расстояния) между объектами;
4. Применение метода кластерного анализа;
5. Проверка результатов кластеризации.

Существуют два ключевых требования к данным:

* Однородность – необходимость гарантировать единую природу всех кластеризуемых сущностей. То есть все объекты должны описываться схожим набором характеристик;
* Полнота - содержание данных в достаточном по всей их номенклатуре, необходимые для рационального или оптимального решения конкретной задачи.

### 1.4. Цели кластеризации для выработки рекомендаций

* Разбиение выборки на группы схожих объектов для упрощения понимания кластерной структуры, что упрощает обработку данных и принятие решение, применяя к каждому кластеру свой метод анализа.
* Сокращение объёма данных, оставляя по одному или несколько наиболее типичных представителей от каждого класса. В таких задачах важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно.
* Выделение нетипичных объектов, аномалий или выбросов, для определения новизны кластеров или их количества. Наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те в свою очередь дробятся ещё мельче, и т. д. Такие задачи называются задачами таксономии. Результатом таксономии является древообразная иерархическая структура. При этом каждый объект характеризуется перечислением всех кластеров, которым он принадлежит, обычно от крупного к мелкому.

### 1.5. Построение ценовых (весовых) моделей

Методы кластеризации и классификации хорошо приспособлены к прогнозированию того, к какой категории относится какой-либо образец. Однако методы в их математическом виде не оптимальны для выборки прогнозов о числовых данных на основании различных атрибутов. Важной задачей так же становится учитывать распределение вероятностей прогноза и интерпретировать данные[29].

Цена - это хороший способ оценки реальной стоимости вещи, а прогнозирование цены является хорошим примером реализации алгоритма, когда учитываются не только цифры, но и степень их влияния. Например диаметр колес автомобиля не коррелирует напрямую с объёмом двигателя или тактовая частота процессора может быть первостепенной при комплектовании компьютера.

Важной частью числового прогнозирования становится определение того, какие переменные существенны и в каком сочетании. Есть переменные влияющие на цену напрямую, а есть, не учитывающиеся вовсе. Кроме того, в каждом конкретном случае приоритеты характеристик для наборов данных могут быть различными.

Но процесс определения наилучших весов переменных можно автоматизировать и применять для выработки рекомендаций по определённым параметрам на основании созданных кластерах или для их создания.

Обычно приходится работать с наборами данных, построенными кем-то другим, поэтому априорной информации о том, какие переменные существенны, а какие – нет, может и не быть.

Иногда бывает, что определенные данные собирать трудно или дорого, и если оказывается, что ценность их на самом деле невелика, то можно избежать лишних затрат. В других случаях одно лишь знание того, какие переменные существенны – особенно в плане влияния на цену, – может определить направление маркетинговых действий или подсказать, как следует изменить дизайн продуктов, чтобы их можно было продать по наивысшей цене.

## 2. Техническое задание

### 2.1. Цель и назначение разработки

#### 2.1.1 Цель разработки

Целью разработки является создание программной библиотеки для выработки рекомендаций на большом объеме данных. На основании кластерного анализа библиотека должна отыскивать взаимосвязи и выдавать рекомендации на основании пользовательских запросов построения. Разработанная библиотека должна внедряться в программные продукты, подключаться к базе данных и, получая информацию, распределять её.

#### 2.1.2 Назначение разработки

Разрабатываемая библиотека должна являться частью других систем, не неся в себе функции самостоятельного приложение. Назначение библиотеки состоит в том, чтобы автоматизировать процесс группировки данных и поиска в них закономерностей для выработки неочевидных решений.

Библиотека может применяться при анализе групп людей, например в маркетинговых исследованиях.

### 2.2. Этапы разработки

2.2.1. Анализ требований к разработке

2.2.2. Анализ решений конкурентов

2.2.3. Проектирование библиотеки

2.2.4. Разработка и тестирование библиотеки

2.2.5. Написание документации

2.2.6. Тестовое внедрение в проект

### 2.3. Основные требования к программному обеспечению

#### 2.3.1. Общие положения

##### 2.3.1.1.Требования к программным решениям

Программное библиотека должна разрабатываться по модульному принципу. Каждый модуль должен работать с данными независимо, но использовать единый драйвер баз данных. Драйвер баз данных должен иметь методы записи и чтения через методы, формируя внутри себя SQL-запросы. При разработке библиотеки должен использоваться язык С++ со стандартной библиотекой классов. Допускается применение фреймворка «Qt» без использования элементов пользовательского интерфейса.

##### 2.3.1.2. Требование к программному обеспечению

Библиотека должна встраиваться в проекты, разрабатываемые на языке С++ с применением фреймворка «Qt». Библиотека должна быть платформонезависимой и компилируемой MinGW или аналогом.

##### 2.3.1.3. Состав программного комплекса

В библиотеку должны входить модули:

Кластеризации

Построение ценовых моделей

Выработка рекомендаций

Драйвер баз данных

#### 2.3.2. Функциональные требования к СУБД

#### 2.3.3. Общее назначение библиотеки:

1. Формировать набор данных для кластеризации
2. Выделять предполагаемое количество кластеров в конкретном наборе данных
3. Находить центройды и предполагаемую форму кластера
4. Ассоциировать новый экземпляр данных с кластером
5. Определять веса в наборах данных
6. Масштабировать векторы экземпляров
7. Принимать пользовательские запросы
8. Строить ценовую модель на основании пользовательского запроса
9. Ранжировать кластеры и экземпляры внутри кластеров по релевантности на основании пользовательского запроса

## 3. Обзор существующих решений

### 3.1 Иерархические методы кластеризации

Основой иерархической кластеризации является последовательное слияние кластеров для образования больших структур или разделение больших кластеров на меньшие. Такая кластеризация зачастую имеет очень хорошую наглядность, но её используют лишь при небольших объёмах наборов данных.

Иерархические методы кластеризации различны по правилам формирования кластеров. Правила используются для определения схожи ли объекты между собой и могут ли они быть отнесены в один кластер. Методы, соединяющие экземпляры в группы называются агломеративными, а разделяющие – дивизимными[30].

В графической аналогии иерархические методы базируются на построении дендрограмм (от греческого dendron - "дерево"), которые описывают расстояния между отдельными точками и кластерами друг по отношению к другу.

Далее каждый вид будет рассмотрен отдельно через своих попоулярных представителей.

#### 3.1.1 Агломеративные AGNES (Agglomerative Nesting).

Эта группа методов, как было сказано ранее, характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров.

В начале алгоритма все экземпляры объектов представлены как отдельные кластеры. Первый шаг отбирает наиболее похожие объекты, формирую из них кластер. С каждым последующим шагом происходит формирование новых кластеров, а так же объединение кластеров в группы. Так продолжается пока не будет сформирован конечный кластер, содержащий в себе другие кластеры.

##### 3.1.1.1. CURE

Этот алгоритм предназначен для кластеризации очень больших наборов числовых данных, только эффективно работать способен с данными низкой размерности.

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

Алгоритм основан на наборе определяющих точек. Сначала алгоритм формирует дерево кластеров, где каждый объект является кластером единичного размера. Затем кластеры сортируются по расстоянию друг от друга, используя «манхэттенское» или «евклидово» расстояние. После формирования этих данных в «куче» оперативной памяти, происходит слияние ближайших кластеров и пересчёт расстояний. Так происходит до получения нужного количества кластеров.

Во время работы алгоритма кластеры разделяются на две группы: первая группы – кластеры у которых вычисляется минимальное расстояние с новообразованным кластером, вторые – все остальные кластеры. При смене кластеров и новом пересчёте происходит чередование кластеров с которыми происходит сравнивание.

Таким образом, алгоритм является высокоуровневым алгоритмом кластеризации, выделяет кластеры различных форм и имеет линейную зависимость к размеру хранимых данных и временной сложности.

##### 3.1.1.2. ROCK

Алгоритм кластеризации ROCK. Robust Clustering Algorithm - агломеративный иерархический алгоритм, для кластеризации большого количества данных с номинальными (булевыми) атрибутами[34].

Алгоритм по работе схож с K-means, но он учитывает наличие связи у общих соседей. Работа алгоритма построена на составлении матриц схожести. Это матрицы соседства кластеров, количества общих соседей, меры близости объектов. Соседство кластеров определяется отношением конъюнкции и суммы по модулю 2 двух кластеров. Формула определения близости объектов зависит количества объектов в каждом кластере и общих ссылок между двумя кластерами. После формирования матриц начинается итеративная кластеризация, находящая двух наиболее близких кластера, и объединяя их. При этом формируются новые значения в матрице общих ссылок и матрице близости, а данные о двух оригинальных кластерах удаляются. Критериями для завершения итераций являются либо достижения заданного их числа, либо сформировано указанное количество кластеров, все расстояния между кластерами нулевые или сформирован кластер больше заданного размера.

#### 3.1.2. Дивизимные DIANA (Divisive Analysis):

Методы этой группы логически противоположны агломеративным, в начале их работы объекты находятся в одном кластере. В результате работы алгоритмов на каждом их шаге такой кластер будет делиться на меньшие, создавая последовательность расщепляющих групп[33].

##### 3.1.2.1. BIRCH

Алгоритм предложен Тьян Зангом и его коллегами[37].

Достоинства алгоритма лежат в его высокой скорости на фоне большой масштабируемости. Это достигается за счёт двухэтапного процесса кластеризации.

На первом этапе алгоритм формирует предварительный набор кластеров. Второй этап применяет к выделенным кластерам иные методы кластеризации, что делает этот очень нетребовательным к ресурсам.

Тьян Занг приводит следующую аналогию в отношении алгоритма BIRCH: «Если каждый элемент данных представить себе как бусину, лежащую на поверхности стола, то кластеры бусин можно "заменить" теннисными шариками и перейти к более детальному изучению кластеров теннисных шариков. Число бусин может оказаться достаточно велико, однако диаметр теннисных шариков можно подобрать таким образом, чтобы на втором этапе можно было, применив традиционные алгоритмы кластеризации, определить действительную сложную форму кластеров.»

##### 3.1.2.2. MST

Этот алгоритм основан на построении минимального остовного дерева (MST, minimum spanning tree). Граф строится на основании представления объектов как вершин, а расстояний между этими объектами как дуг. На полученный граф применяется метод построения минимального остовного дерева, причём может применяться любой известный метод, но с учётом большого числа дуг в графе (при N документах в коллекции - [http://www.moluch.ru/conf/tech/archive/2/207/images/m4d72222b.gif](http://www.moluch.ru/conf/tech/archive/2/207/images/m4d72222b.gif) дуг). После этого удаляются ребра с наибольшими длинами, образуя лес небольших деревьев, узлы которых порождают кластеры.

Это достаточно гибкий алгоритм, кластеризующий произвольные наборы данных, выделяя кластеры различных форм, и выбирая наиболее оптимальное решение. Скорость работы будет во многом зависеть от выбранного метода получения минимального остовного дерева.

### 3.2. Неиерархические (итеративные) методы кластеризации

С возрастанием объёмов данных иерархические методы теряют все свои привлекательные свойства, тогда эффективными могут оказаться неиерархические методы, представляющие из себя итеративные алгоритмы разделения исходной совокупности. То есть весь набор данных делится на определённое количество кластеров или пока не сработает правило остановки. В этой группе методов, используются два подхода, оба отличных от работы дивизимных иерархических принципов. Методы первого подхода определяют границы кластеров по наиболее плотным участкам многомерного пространства исходных данных. То есть происходит процесс поиска «сгущения точек». Методы второго подхода минимизируют меры различия объектов.

#### 3.2.1. К-средних (k-means)

Это наиболее распространённый неиерархический метод, и возможно самый распространённый метод кластеризации в целом в связи с простотой его реализации[36].

Перед началом работы метода необходимо иметь предположение о вероятном количестве кластеров. K-means хорошо подходит для подтверждения гипотез о количестве кластеров и может использоваться данные от предыдущих вычислений, либо просто от интуитивно выбранного числа. Алгоритм сначала построит заданное количество кластеров, а после этого будет распределять объекты так, чтобы среднее среди всех переменных в кластере будут максимально отличными.

#### 3.2.2. PAM (k-means + k-medoids)

Метод по своей сути является модификацией К-средних с применением К-медианы[30]. Работа алгоритма аналогична K-means, только объекты в нём распределяются не относительно центра кластера, а релятивно его медианы. Алгоритм лучше своего основоположника противостоит выбросам и шумам, поскольку медиана мене подвержена этим явлениям. К сожалению, алгоритм не преобразовался настолько, чтобы применяться к для больших объёмов данных.

#### 3.2.3. Алгоритм CLARA (Clustering LARge Applications)

В результате CLARA предлагает наилучшую кластеризацию. Этот алгоритм достаточно эффективен для больших объёмов данных, но сильно зависит от заранее выбранного начального набора данных, что приводит к хорошим показателям качества кластеризации на тестовой выборке, но может оказаться ошибочным в применении ко всему набору данных.

#### 3.2.4. LargeItem

Алгортим LargeItem был предложен Вангом в 1990 году. Это оптимизационный алгоритм для кластеризации транзакционных данных, основанных на критериальной функции, оптимизации глобального критерия[35]. Этот глобальный критерий использует параметр поддержки (в терминологии здесь много общего с алгоритмами для выявления ассоциативных правил). В вычисление глобального критерия делает алгоритм кластеризации во много раз быстрее, чем при использовании локального критерия при парном сравнении объектов, поэтому "глобализация" оценочной функции – один из путей получения масштабируемых алгоритмов. Алгоритм включает в себя две фазы: фазу распределения и фазу улучшения. Лучшим решением является то, которое лучше минимизирует целевую функцию. Так как найти абсолютно точное решение не представляется возможным с достаточной долей вероятности, то цель этого алгоритма состоит в нахождении приближенного решения, что является достаточным для практического применения. В отличии от К-means, LargeItem позволяет значению k варьироваться, т.е. алгоритм не обязан знать её заранее. Чтобы избежать сканирования всех транзакций при кластеризации, некоторые кассетные функции, такие как |Largei|, |Uki=1=Smalli| и | Uki=1=Largei|, сохраняются после каждого распределения или перемещения транзакции.

Алгоритм использует некоторые стандартные методы индексации, такие как хэш-таблицы и B-дерева в обслуживании и обновлении, доступные для каждого кластера.

#### 3.2.5. CLOPE

В 2002 году группа китайских учёных представила[35] алгоритм CLOPE (Clustering with sLOPE). Особенностью этого алгоритма является высокая производительность работы по сравнению с другими иерархическими методами. Работа этого алгоритма основана на максимизации глобальной функции стоимости, повышающей близость транзакций в кластерах с помощью увеличения параметра кластерной гистограммы.

С помощью параметра, названного авторами CLOPE коэффициентом отталкивания (repulsion), регулируется уровень сходства транзакций внутри кластера, и, как следствие, финальное количество кластеров. Этот коэффициент подбирается пользователем. Чем больше r, тем ниже уровень сходства и тем больше кластеров будет сгенерировано.

Как видно, алгоритм CLOPE является масштабируемым, поскольку способен работать в ограниченном объеме оперативной памяти компьютера. Во время работы в RAM хранится только текущая транзакция и небольшое количество информации по каждому кластеру

### 3.3. Сравнительный анализ иерархических и неиерархических методов кластеризации

Ключевой вопрос встаёт о том, какой группе методов отдать предпочтение при обработке исходных данных. Аналитику приходится учитывать различные факторы и особенности между иерархическими и итерационными методами. Рассматривая эти группы следует выделить основные факторы.

Применение итерационных методов даёт высокую стойкость к шумам и выбросам, но в свою очередь это обусловлено заранее определёнными данными, такими как количество кластеров, число итераций или правила остановки. В таком случае процесс анализа данных будет во многом зависеть от предварительной работы аналитика и построения системы в целом.

Зачастую складывается ситуация, в которой определить начальные данные невозможно, либо это будет отдельной трудоёмкой задачей, превосходящей операции кластеризации. Например, когда объём начальной выборки слишком велик, то можно проводить эксперименты с количеством кластеров или их размерами.

За счет такого "варьирования" результатов достигается достаточно большая гибкость кластеризации.

В свою очередь, иерархические методы опираются не на число кластеров, а строят полную структуру, содержащую вложения. Этот факт накладывает ограничение по объему набора данных. Так же зачастую оказывается трудным выбрать меру близости. В таком свете методы оказываются негибкими, поскольку для оценки даже приближенных результатов, приходится обрабатывать весь объём первоначальной выборки раз за разом.

Преимущество же этой группы методов в сравнении с неиерархическими методами - их наглядность и возможность получить детальное представление о структуре данных.

### 3.4. Сеточные алгоритмы

#### 3.4.1. Алгоритм WaveCluster

Данные алгоритм относится к группе сеточных алгоритмов (Grid-based)

В основе алгоритма лежит метод волновых преобразований. На первом шаге работы, алгоритм обобщает данные, накладывая на пространство данных многомерную решетку. Последующие шаги анализируют уже не конкретные точки, а их обобщённые характеристики в каждой ячейке. После этого алгоритм применяет волновые преобразования к обобщённым данным.

Среди сеточных алгоритмов так же выделяют STING, OptiGrid, GRIDCLUS, GDILC.

Ниже, я хотел бы провести сравнительных анализ некоторых методов кластеризации, о которых ранее упоминалось.

|  |  |  |
| --- | --- | --- |
| Метод | Достоинства | Недостатки |
| CURE | - Кластеризация высокого уровня даже при наличии выбросов  - Выделение кластеров сложной формы и различных размеров  - Обладает линейно зависимыми требованиями к месту хранения данных и временную сложность для данных высокой размерности | - Необходимость в задании пороговых значений и количества кластеров |
| BIRCH | - Двухступенчатая кластеризация, кластеризация больших объемов данных  - Работает на ограниченном объеме памяти  - Является локальным алгоритмом  - Может работать при одном сканировании входного набора данных  - Данные могут быть неодинаково распределены по пространству  - Обрабатывает области с большой плотностью как единый кластер | - Работает только с числовыми данными  - Хорошо выделяет только кластеры выпуклой или сферической формы  - Необходимость в задании пороговых значений |
| MST | - выделяет кластеры произвольной формы (выпуклой и вогнутой форм)  - выбирает из нескольких оптимальных решений самое оптимальное | - чувствителен к выбросам |
| k-средних | - Простота использования  - Скорость работы  - Понятность и прозрачность алгоритма | - Чувствителен к выбросам  - Медленная работа на больших объёмах  - Необходимо задавать количество кластеров  - Невозможность применения алгоритма на данных, где имеются пересекающиеся кластеры  - Не гарантируется достижение глобального ­­минимума  - Работа алгоритма сильно зависит от выбранных начальных центров кластеров, оптимальное значение которых не может быть известно заоанее |
| PAM | - Простота использования  - Скорость работы  - Понятность и прозрачность алгоритма  - Менее чувствителен к выбросам в сравнении с k-means | - необходимо задавать количество кластеров;  - медленная работа на больших базах данных |
| CLOPE | - масштабируемость  - скорость работы  - качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера.  - легко рассчитывается и интерпретируется  - малый объём ресурсов  - автоматически подбирает количество кластеров  - регулируется одним единственным параметром - коэффициентом отталкивания. |  |
| Алгоритм HCM | - Легкость реализации  - Вычислительная простота | - задание количества кластеров  - отсутствие гарантии в нахождении оптимального решения |
| Fuzzy  C-means | - Нечеткость при определении объекта в кластер позволяет классифицировать пограничные объекты | - Вычислительная сложность  - Задание количества кластеров  - Неопределённость с выбросами |
| WaveCluster | - Может обнаруживать кластеры произвольных форм  - Не чувствителен к шумам | - Сложность реализации  - Применим только к данным низкой размерности |

На первый взгляд, кажется, что результаты, полученные в результате кластеризации не имеют статистического обоснования. С другой стороны, при большом разнообразии вариантов понятия кластера, нестатическая интерпретация полученных результатов даёт возможность получить оценку, которая при использовании других методов бывает затруднительна.

## 4. Обзор существующих систем кластеризации

### 4.1. Коммерческие инструменты:

**ClustanGraphics**

На момент написания работы актуальная версия 8 (2005 год релиза). Инструмент основан на K-means, так же применяется иерархическая кластеризация. Программный пакет способен разделить миллион объектов на 20 тысяч кластеров. Мастер в программе позволяет импортировать данные из таблиц Excel, а представление результатов происходит в графическом виде.

**BayesiaLab**, includes Bayesian classification algorithms for data segmentation and uses Bayesian networks to automatically cluster the variables.

Инструмент основан на Баейсовой классификации и использует Байсовые сети для автоматической кластеризации. Продукт поставляется в виде решений для конкретных задач, таких как маркетинг, анализ рисков и медицинское прогнозирование.

**CViz Cluster Visualization** это инструмент, разработанный на Java. Назначение – это кластеризация для визуального анализа.

CViz – это средство визуализации для анализа данных высокой размерности. Основной упор сделан на графическом представлении результатов анализа и предназначена для обзорного анализа. Работа программы проводится на методах линейного дискриминантного анализа.

**IBM Intelligent Miner for Data**

Специализированный прикладной интерфейс SQL API, который состоит из двух уровней с разной степенью детализации и абстракции.

Прикладной интерфейс задач Easy Mining является проблемно-ориентированным и используется для выполнения базовых задач интеллектуального анализа;

Прикладной интерфейс IM Scoring / Modeling SQL/MM API соответствует стандарту ISO/IEC 13249-6:Data Mining и позволяет создавать приложения интеллектуального анализа под конкретные индивидуальные требования пользователя. Этот интерфейс может быть использован через скрипты SQL, или из любого JDBC, CLI, ODBC, или SQLJ приложения. Приложение разработано компанией IBM как Java приложение. Есть возможность не использовать базы данных, а получать данные из таблиц Excel. Программный комплекс включает в себя модули оценки и моделирования, предоставляя набор инструментальных средств разработки программ. Результаты моделирования просматриваются при помощи готового Java продукта IM Visualization. А инструмент Design Studio интегрируется в среду Eclipse. Результаты моделирования данных (ассоциации, последовательности, классификации, кластеризации и регрессии) могут быть просмотрены с помощью готовых Java средств визуализации IM Visualization.

**ELKI: Environment for Developing KDD-Applications Supported by Index-Structures**

ELKI – это программный модульный фреймворк, разработанный на JAVA для исследований и обучения Профессором Хоано-Питеров Крейгелем в Мюнхенском университете Людвига Максимилиана. ELKI использует соединение алгоритмов разделения, типизации и индексации данных, различных методов расстояний. Система использует SQL для хранения даннх, формат SVG для вывода графических результатов, а для математических вводов применяется система LaTeX.

Алгоритмы 4C, COPAC, HiCO, ERiC, CASH были опубликованы в рамках диссертации, а их реализация удостоилась многих наград. Кластерный анализ включает в себя K-means, Максимизационные алгоритмы, односвязную кластеризацию. Включены такие алгоритмы как DBSCAN (Density-Based Spatial Clustering of Applications with Noise); OPTICS (Ordering Points To Identify the Clustering Structure), включая расширения OPTICS-OF, DeLi-Clu, HiSC, HiCO и DiSH; SUBCLU (Density-Connected Subspace Clustering for High-Dimensional Data). Используются априорные данные, динамическое оболочки. Сейчас программа имеет мажорную версии 0.

**CLUTO**

Данный программный пакет предназначен для кластеризации баз данных низкой и высокой размерности. Пакет включает в себя модуль gCLUTO – это кроссплатформеное графческое приложение для анализа характеристик кластеров. Так же в пакет включен модуль wCLUTO, который является web-реализацией

|  |  |  |
| --- | --- | --- |
| Название программной библиотеки | Достоинства | Недостатки |
| ClustanGraphics | Быстрая кластеризация миллиона объектов в 20 тысяч кластеров | Основан на K-means;  Устаревшая версия от 2005 года  Java-ориентирование приложение |
| BayesiaLab | Присутствие большой базы для решения прикладных задач;  Применение Байсовских сетей | Продукт направлен на решение конкретных задач |
| CViz | Реализует методы линейного дискриминантного анализа | Основан на визуальную оценку данных экспертом |
| IBM Intelligent Miner for Data | Простота использования  Быстрое представление данных | Java-ориентированое приложение.  Коммерческий продукт IBM, очень тесно связан с другими продуктами IBM |
| ELKI | Очень мощный продукт, использующий множество актуальных и современных методов  Много возможны форматов данных для работы  Оценён наградами | На момент написания диплома находится в нулевой мажорной версии. Выход версии 1.0 не определён во времени. |
| CLUTO | Включает в себя графический и Веб модули |  |

### 4.2. Выводы

Из вышеизложенного материала очевидно положительно выделяется алгоритм кластеризации CLOPE. Во время изучения материала были выделены группы методов, описанные ранее, следуя которым, алгоритм CLOPE является категорийным итеративным и транзакционным, что не может покрыть весь спектр задач, таких как например сферическая кластеризация с центроидом размерности на целочисленных значениях, совпадающей с объектом размерности. Для решения подобной задачи будет рассмотрен и добавлен в библиотеку алгоритм BIRCH, совмещённый с методом k-meanes.

Среди рассмотренных продуктов практически все ориентированы на Java-интеграцию. Практически все методы используют для своей работы SQL. Многие продукты заранее ориентированы на решение конкретных бизнес-задач. Целью моей работы стоит создание универсального инструмента на пользовательских данных, способного интегрироваться в разрабатываемые продукты, преимущественно на C++. Среди представленных систем ярко выделяется своими возможностями продукт ELKI, но это новый пакет, находящийся на стадии публичного тестирования, а время выхода первой стабильной версии неизвестно.

## 5. Предлагаемые методы и математическое обеспечение

### 5.1. Методы кластеризации

#### CLOPE

Алгоритм CLOPE является алгоритмом кластеризации транзакционных данных. Транзакция в данном контексте – это произвольный набор объектов. То есть в конкретном случае транзакция похожа на кортеж в котором могут находить нетипизированные данные, не имеющие никой связи между собой, и, не выстраиваемые в пространстве. Задача кластеризации подобных данных состоит в получении такого разбиения, чтобы схожие транзакции находились в одном кластере, а отличные – в одном из других. Для этого в алгоритме применяется принцип максимизации глобальной функции стоимости[ссылка], сближающей транзакции в кластерах, увеличивая параметр кластерной гистограммы.

Например, для набора транзакций {(a, b), (a, b, c), (a, c, d), (d, e), (d, e, f)} сравним два разбиения на кластеры:

1. {{ab, abc, acd}, {de, def}};
2. {{ab, abc}, {acd, de, def}}.

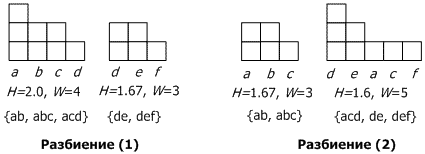


Рис. 5.1.1.1 Разбиение кластеризацией CLOPE

В первом случае кластер представляет четыре независимых уникальных объекта (a, b, c, d), которые входят в кластер количеством (3, 2, 2, 1) соответственно. Таким образом, кластер имеет ширину W=4, то есть включает в себя 4 типа объектов. Площадью S будем считать вхождение в кластер всех объектов, то есть S=3+2+2+1=8. Зная два этих параметра можно рассчитать высоту кластера H=S/W=8/4=2. Если тоже самое проделать с остальными с кластерами, то мы увидим, что при одинаковом количестве элементов в кластере параметры H, W, S могут произвольно меняться. Если посмотреть оба разбиения, то можно понять, что разбиение 1 выгоднее, так как обеспечивает большее число меньшее количество уникальных объектов и большее число наложений. Именно в этом заключается принцип максимизации стоимости.

Пусть D – множество транзакций {t1,…,tn}, где t – набор объектов {i1,…,in}. Найти такое множество кластеров {C1,…,Ck} для множества {t1,…,tn}, такое что

Гистограмма кластера – это графическое изображение его расчетных характеристик.

Алгоритм должен учитывать высоту H при разбиении, потому что транзакция, максимально увеличивающая площадь по отношению к другим существующим кластерам, не увеличивая ширины, наиболее подходит к уже содержащимся в кластере транзакциям. Но оценка высоты не является эффективным критерием в случае разбиений с одинаковой высотой, например H=1. В таком случае вместо оценки высоты необходимо вычислять градиент G(C)=H(C)/W(C)=S(C)/W(C).

Обобщая, выведем формулу глобального критерия

(5.1.1.2)

Где |Ci| - количество объектов в i-м кластере

k – количество кластеров.

r – коэффициент отталкивания (repulsion), положительно вещественное число превосходящее 1.

При помощи коэффициента отталкивания определяется степень сходства транзакций в кластере, причём зависимость обратно пропорциональна, то есть с повышением значения коэффициента, снижается показатель схожести транзакций при сравнении, и тем самым увеличивается количество кластеров.

Таким образом, постановка задачи кластеризации для алгоритма CLOPE выглядит так:

Работа алгоритма проходит методом итеративного перебора записей базы данных, а глобальность критерия оптимизации, основанном на расчете параметров кластеров, позволяет обрабатывать данные значительно быстрее, чем в сравнении транзакций между собой.

Алгоритм работает в два этапа. На первом этапе происходит первый проход по базе с целью инициализации данных и построения первичного разбиения. Второй этап осуществляет итеративные обходы базы, оптимизируя функцию стоимости до прекращения изменений в кластерной структуре.

Вполне очевидно, что CLOPE – масштабируемый алгоритм, способный работать на ограниченных ресурсах. Во время его работы в памяти находится только текущая транзакция и кластерные характеристики (CF – cluster features). При вхождении 10 000 объектов в 1 000 кластер требуется всего около 40 Мб памяти для хранения данной информации.

Вычислительная сложность алгоритма возрастает линейно с увеличением таблицы и ростом числа кластеров, она равна O(N\*K\*A), где

N – общее число транзакций;

K – максимальное число кластеров;

A – средняя длина транзакции.

Таким образом алгоритм относится к эффективным на большим объёмах данных.

Данный алгоритм можно успешно применять не только для транзакционных данных, но и для любых категорийных. Основным требованием является нормализация данных. Это может быть бинарная матрица или отображение уникальных объектов {u1, u2, …, uq} и множеством целых чисел {1, 2, …, q}. Таким образом к виду транзакции можно свести любой набор данных, поэтому CLOPE находит своё место в обработке категорийных данных.

#### 5.1.2. K-meanes

Теперь рассмотрим подробно самый популярный и простой в реализации алгоритм k-means. Этот алгоритм был открыть в различных дисциплинах Ллойдом (1957), Форджи (1965), Фридманом и Рубином (1967), а так же МакКуином (1967).

K-means применим к объектам в d-мерном векторном пространстве, представимых как набор D = {xi | i=1, …, N}, где xi ∈ Rd – i-й объект. Суть алгоритма состоит в объединении D так, чтобы каждая xi попала только в один k раздел. В результате образуется кластерный составной вектор m длиною N, где mi – номер кластера xi. Параметр k – входное значение для работы алгоритма и является конечным числом кластеров. Этот параметр является экспертным вводом, основанным на наблюдениях, анализе предварительных результатов или просто интуитивном предпочтении аналитика. Неважно каким будет значение k для понимание разделения набора данных, но оптимальное значение достигается путём ряда итеративных экспериментов.

В работе алгоритма каждый кластер представлен точкой в Rd и представляется как множество C={cj | j=1, …, k}. Эти состояния называют центроидами кластера. Для кластеризации применяются меры близости, в частности евклидово расстояние, уже рассмотренное ранее. Работа алгоритма заключается в минимизации функции стоимости, которая представлена как квадрат расстояния между каждой точкой xi и ближайшим представителем кластера cj.

[http://intellect-tver.ru/wp-content/uploads/%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D0%B0-2.1.jpg](http://intellect-tver.ru/?attachment_id=266)

(5.1.2.1)

Приведённое уравнение часто называют целевой функцией k-means.

Алгоритм строится в 2 шага, которые итеративно чередуются между собой:

1. Переписывание номера кластера для всех объектов в D
2. Обновление данных кластера по содержащимся в нём объектам

Сначала инициализируются представители кластера, но основании выборки k из Rd случайным образом, устанавливая их как решения подмножества, либо нарушая среднее значение данных k-раз. Затем выполняется итерации до сходимости за 2 шага:

1. Каждому объекту присваивается самый близкий представитель, произвольно нарушая связи данных, что приводит к их разделению;
2. Центроид кластера перемещается на место среднего арифметического всех объектов в кластере.

Алгоритм сходится при невозможности осуществлять перемещения. Таким образом целевая функция уменьшается с каждым шагом, а сближение гарантировано за конечное число итераций.

К сожалению значение целевой функции не информативно с точки зрения подбора количества кластеров, потому что функция стоимости принимает минимальное значение в том случае, когда число кластеров равно числу объектов.

Реализация алгоритма:

1. Выберем случайным образом k точек из D как представителей класса C в соответствии с формулой (1)
2. Вычислим центр для каждого кластера
3. Перераспределим объекты по кластерам

Каждая итерация в таком случае будет иметь сложность O(N\*k), а число итераций, необходимых неизвестно, но растёт с числом объектов N. Решение проблемы скорости в данном случае может быть предложено путём распределения вычислений на потомки, в этом случае необходимо разбить множество данных на P частей равному по значению количеству потоков, а каждый поток будет работать со своим набором данных.

Актуальной проблемой разбиения является проблема «пустых кластеров». Этот недостаток усиливается с увеличением числа k, когда в момент работы образуется центройд кластера cj, такой, что все точки xi в D оказываются ближе к центроиду другого кластера. В таком случае точки перераспределятся, а исходному кластеру будут назначены нулевые значения, образуя из него пустое множество. В таком случае необходимо предусмотреть переинициализацию центроида пустого кластера.

Несмотря на недостатки, алгоритм k-means является наиболее распространённым инструментом кластеризации на практике. Это простой и масштабируемый метод, который обладает достаточной эффективностью и способен работать как дополнение к более сложным методам, что я рассмотрю далее.

#### 5.1.3. BIRCH

Алгоритм BIRCH представляет из себя двухэтапный процесс кластеризации, который хорошо кластеризует большие наборы числовых данных. Ранее был рассмотрен алгоритм k-means, который предполагается использовать как второй этап в данном методе. Хорошая связь этих методов состоит в том, что они оба работают с числовыми данными и строят кластеры сферических форм. Для работы алгоритма BIRCH необходимо только указание пороговых значений, а количество кластеров, необходимое для k-means будет определено во время первого этапа.

Рассмотрим работу алгоритма:

Построение начального CF-дерева (CF Tree, кластерное дерево). Кластерное дерево – это взвешенно сбалансированное двухпараметрическое дерево, где B- коэффициент разветвления, а T – пороговая величина. Каждый узел, не являющийся листом дерева, имеет не более B вхождений узлов, представленных в форме [CFi, Childi], где i=1, …, B; а Childi – указатель на дочерний i-й узел. Лист дерева имеет указатель на два соседних узла, а радиус кластера, состоящего из элементов этого узла не должен превосходить пороговое значение.

Кластер представляется как тройка (N, LSS, SS), где N – число элементов входных данных кластера, LS – сумма элементов входных данных, SS – сумма квадратов элементов входных данных.

Сжатие данных (необязательный этап) осуществляется перестроением кластерного дерева с увеличением пороговой величины T.

Глобальная кластеризация происходит применением выбранного алгоритма кластеризации на листьевых компонентах кластера. В выбранном случае здесь вступает в работу алгоритм K-means, с полученным на предыдущих этапах числом кластеров.

Улучшение кластеров (необязательный этап) использует центры тяжести, полученные в результате глобальной кластеризации, перераспределяя данные между «близкими» кластерами. Данный этап гарантирует что одинаковые данные попадут в один кластер.

### 5.2. Метод вычисления ценовых моделей

В случае автоматизации выбора весов предлагается использовать метод подбора значений на основании использования метода перекрёстного контроля. Идея данного метода состоит в том, чтобы разделять данные на обучающий и тестовый наборы. Обучающий набор должен быть оснащён правильными результатами в том событии, для которого строится прогнозирование. После этого данные передаются в алгоритм, который пытается их спрогнозировать. Зачастую процедура проводится несколько раз с различным разбиением. Тестовый набор составляет 5% от выборки, а остальные 95% - обучающий набор. В качестве оценки используется сумма квадратов разностей. Суммирование квадратов разностей – хороший метод, так как с увеличением разности влияние на сумму увеличивается нелинейно. Таким образом, порождая большие отклонения, метод работает эффективнее остальных, образующих умеренно близкие результаты. Подбор весов трудоёмкая и длительная операция, но она производится один раз для каждой новой обучающей выборки.

Теоретически, можно подбирать множество различных сочетаний весовых коэффициентов вручную, но подобный подход невероятно трудоёмок. Для оптимизации процесса необходимо задать область определения переменных, диапазон и целевую функцию. Областью определения – диапазон всех весов по каждому измерению. Преимущество оптимизации масштабов по осям, это наглядность важности тех или иных переменных, уровень их влияния. Иногда одно лишь знание таких данных позволяет пересмотреть все ранее полученные результаты и гипотезы.

## Структура системы

### 6.1. Принципиальная схема



### 6.2 Программная реализация библиотеки

Модуль ввода данных

Связь данных пользователя и данных работы библиотеки задаётся не произвольно. Надсистема библиотеки не может просто так подключить свою базу данных для обработки информации. В первую очередь это связано с необходимостью строго типизировать библиотеку и уменьшать абстракцию, следовательно, сжимать область применения программного продукта. Во-вторых, работая с абстрактными данными библиотека не получает прямого доступа к пользовательским данным, а так же не несёт в себе опасности кражи данных. Надсистема, использующая библиотеку перед началом работы должна передать конфигурации для подключения к заранее созданной базе данных. В этой базе будут размечены необходимые таблицы. Если таблицы уже существуют, то драйвер базы данных библиотеки продолжит заполнение таблиц информацией. После подключения к базе данных надсистема получает возможность передавать библиотеке данные в итеративном режиме по одному событию. Событием будем называть целочисленное значение, характеризованное двумя уникальными метками. Первая метка – идентификатор процесса, которому принадлежит событие, вторая – кэш имени события. Библиотека не обязана работать с истинными значениями процесса и события, поэтому надсистеме рекомендуется оперировать двумя хэш-таблицами вида «процесс | идентификатор» и «событие | кэш имени события». Ввод информации в систему образован в виде передачи функции добавления процесса с двумя строковыми и одним целочисленным параметрами. Первые два строковых параметра – это идентификатор процесса и кэш имени события соответственно, а третье, целочисленное, инкремент счётчика событий. При получении параметров, библиотека отыскивает в базе данных ячейку, соответствующую паре «идентификатор процесса – кэш имени события» и увеличивает её значение на значение инкремента. Если указанных процесса или события не существует, то происходит их создание. В базе данных также находятся две несвязанных таблицы – хэш таблица идентификатор «процесса | строка» и кэш имени «события | столбец». Удаление записей из базы данных библиотекой не предусмотрено, эта функция ложится на администратора базы данных и сторонние редакторы. В случае обновления исходных данных, рекомендуется сделать резервную копию существующей базы данных библиотеки, очистить текущее содержимое таблиц и агрегировать процессы заново. Это поможет не образовывать коллизий, а скорость выполнения данной операции O(N), где N – число процессов.

Модуль построения ценовых моделей.

Данный модуль предназначен для приёма вектора весов от надсистемы в случае, если экспертом были определены параметры заранее, либо вызов метода, содержащего алгоритм автоматического взвешивания. При вызове метода передачи весов библиотека ожидает получить вектор значений двойной точности с определённым количеством элементов в определённом порядке. Каждый элемент будет ассоциирован с процессом в порядке расположения кэша имени процесса в таблице. Для синхронизации данных предусмотрен метод получение списка событий. Данный метод возвращает вектор строк, где каждый элемент – кэш имени процесса, а положение в векторе соответствует номеру столбца таблицы.

Для автоматического взвешивания необходимо вызвать соответствующей метод, который сформирует вектор коэффициентов и вернёт его надсистеме.

Модуль кластеризации.

В библиотеке предусмотрены два вида кластеризации CLOPE как представитель итерационного метода для категорийных атрибутов и BIRCH с использованием k-means на второй фазе, как представитель иерархического метода для отыскания сферических кластеров с центроидами для объектов с числовыми атрибутами, использующий меру евклидово расстояние как меру схожести.

Для работы алгоритма CLOPE необходим параметр, названный коэффициентом отталкивания (repulsion). Поэтому при вызове метода данного алгоритма ожидается передача параметра двойной точности значением больше 1.

В своей работе алгоритм BIRCH так же не требует априорных данных, но метод кластеризации с применением данного алгоритма ожидает получения порогового значения, числа двойной точности, определяющего максимальный размер кластера.

Данные результатов работы алгоритмов хранятся в таблицах, представляющих из себя следующую структуру данных:

Модуль рекомендации на основании запросов.

Библиотека решает два экспертных вопроса – это отыскание группы объектов, схожих с данным, и прогнозирование неизвестных параметров новых объектов.

Решение первого вопроса представлено методом поиском кластера. Данный метод переопределён для двух видов запроса:

* Найти кластер, в который попал процесс. Тогда параметром ожидается строка, содержащая идентификатор искомого процесса. В результате работы метода будет возвращён номер кластера.
* Найти кластер, в который наиболее вероятно попадёт новый процесс с данными, обладающими полнотой, то есть метод ожидает принять новый вектор, заполненный значениями

На второй вопрос поможет ответить метод прогнозирования события.

Вспомогательный модуль

Вспомогательный модуль позволяет работать с таблицами вне зависимости от модулей вычислений.

Класс получения конкретного значения события позволяет получить данные из таблицы «идентификатор процесса | кэш имени события». Метод получения конкретного процесса принимает параметром идентификатор процесса и возвращает вектор целочисленных данных в порядке размещения кэшей имени процессов в таблице.

Метод получения списка всех процессов выполняется без параметра и возвращает вектор идентификаторов процессов в порядке их расположения в таблице.

Метод получения списка всех событий выполняется аналогично, но возвращает вектор кэшей имён событий в порядке их размещения в таблице.

### 6.3. Форматы входных и выходных данных:

#### 6.3.1 Модуль ввода начальных данных.

Формат единицы процесса – две бинарные строки формата юникод. Параметры входа [уникальный идентификатор процесса] | [уникальная хэш-сумма имени процесса]

#### 6.3.2. Модуль формирования весовой модели:

Выходные данные:

Строка с хэш-суммами имён процессов, разделённых одной запятой и пробелом. [хэш-сумма №1, ] | [хэш-сумма №2, ] … [хэш-сумма №n].

Входные данные:

Строка с весами в числовом виде, разделёнными запятой. Каждое число должно находиться на позиции, соответствующей ему хэш-суммы имени процесса. При избыточном количестве чисел система отбросит лишние числа справа. При недостаточном их количестве система дополнит входные данные справа единицами.

Если надсистема, использующая библиотеку, будет использовать автоматическую систему поиска весов, то в соответсвующий метод необходимо передать параметры искомого атрибута.

#### 6.3.3. Модуль кластеризации

Входные данные:

Для кластеризации методом CLOPE в соответствующей метод необходимо передать коэффициент отталкивания в виде числового значения двойной точности.

# Б. Конструктивно-технологическая часть проекта

## 1. Технология программирования

С ростом масштабов задач, менеджеры не справляются с управлением выделенными ресурсами. Увеличение ресурсов может усугублять проблему, не решая поставленных задач. Человеческие ресурсы в разработке программного обеспечения играют первостепенную роль, а проблема роста сложности управления хорошо описывается высказыванием, что «девять женщин не родят за месяц ребёнка». Эту действительность раскрыл[39] в 1975 году Фредерик Брукс в своей книге «Мифический человеко-месяц». В своей более поздней работе Брукс выделил две составляющие сложности разработки ПО: трудности, присущие специфике создания ПО, и акцидентальные – в данном контексте такие, которые связаны с ограничениями уровня развития науки и техники. Вторые более или менее успешно преодолеваются на пути научно-технического прогресса, зато первые будут сопровождать разработку ПО всегда.

Управление процессом становится действием, напрямую зависимым от восприятия состояния и поведения объекта. В том, что касается создания ПО, это является весьма сложной задачей, поскольку процесс разработки – сугубо интеллектуальная, во многом творческая деятельность, для которой конвейерные либо другие им подобные методы неприменимы. Поэтому и были предприняты активные попытки представить модель процесса создания ПО, которая в максимальной степени смогла бы учесть присущие ему особенности и сделать его управляемым.

### Модели на основе инженерного подхода

#### 1.1.1. Каскадная модель.

В 1970 У.У. Ройс опубликовал статью[40] в которой концептуально описал то, что сейчас называется «Каскадной моделью». В своей работе Ройс представляет процесс разработки как поток, проходящий свои фазы последовательно. При такой модели разработчик переходит от стадии к стадии последовательно, полностью завершая текущую ступень.

Данная методика зачастую критикуется[41] за недостаточную гибкость. При такой работе формализация встаёт на первое место перед сроками, а так же ведёт к увеличению стоимости разработки. Впрочем, формализация позволяет снизить риски в разработке.

#### 1.1.2. V-образная модель.

В 1980 году немецкая аэрокосмическая компания IABG и Американский национальный совет по системной инженерии независимо друг от друга разработали концепцию V-образной модели[32]. Современная версия «V-Model XT» была утверждена в 2005 году и является стандартом немецкий оборонных, правительственных проектов, и прочих национальных производителей программного обеспечения.

V-Model по своему принципу есть ни что иное, как вариация каскадной модели, в которой разработка спускается по левому ребру сверху вниз, а тестирование поднимается вверх по правому. Внутри задачи зависимы горизонтально, что показывает, как задачи разработки влияют на систему тестирования.

Данная модель так же имеет свои недостатки. Например, тестирование происходит поздно, что, при внесении изменений, влияет на сроки разработки, а следовательно на стоимость. Модель не анализирует риски и не гибка для динамического изменения требований. В защиту метода нужно упомянуть высокий уровень планирования разработки, верификация для всех этапов разработки, а так же входных и выходных данных.

### Модели, учитывающие специфику разработки ПО

Поскольку первые модели были заимствованы из традиционной инженерной области, они не учитывали в полной мере специфику производства ПО. Однако последующие модели были уже гораздо больше ориентированы на особенности этого вида деятельности, имеющего много принципиальных отличий от конструирования предметов материального мира.

#### Инкрементная модель. Инкрементная модель строится на инкрементной стратегии конструирования. Сначала определяются системные и пользовательские требования, оставшуюся часть реализуют в виде последовательности версий. Первая версия реализует зачастую базовый функционал, а следующая уже содержит дополнительные возможности. Природа инкрементного подхода итеративна и на каждой итерации обладает готовым продуктом.

#### Спиральная модель.

Спиральную модель в 1986 году предложил Барри Боэм. Модель стала революционной в области разработки программного обеспечения. Модель сочетает в себе проектирование, прототиприрование и анализ. Большая часть рисков образуется из организационных и процессных взаимодействий. На каждом ветке находится фрагмент или версия продукта, здесь определяются цели проекта, его качество, характеристики и планируется следующий виток. Это позволяет развивать и конкретизировать детали проекта, реализуя обоснованный вариант. Виток разбивается 4 секторами:

* Оценка и разрешение рисков
* Определение целей
* Разработка и тестирование
* Планирование

Каждый виток может гибко подстраиваться и использовать различные модели разработки. В итоге модель в своём глобальном представлении является сочетанием прототипирования и водопадной модели. Итеративность процессов позволяет начинать новый виток, когда ещё не завершён предыдущий, перенося какие-то этапы с одно витка на другой. Для определения сроков перехода между витками используется вводятся временные ограничения на каждый этап. Переход осуществляется по плану вне зависимости от степени готовности работ.

Спиральная модель ориентирована на большие, дорогостоящие и сложные проекты. В условиях, когда бизнес цели таких проектов могут измениться, но требуется разработка стабильной архитектуры, удовлетворяющей высоким требованиям по нагрузке и устойчивости, имеет смысл применение Spiral Architecture Driven Development. Данная методология, включающая в себя лучшие идеи спиральной модели и некоторых других, позволяет существенно снизить архитектурные риски, что является немаловажным фактором успеха при разработке крупных систем.

### Современные модели

К середине 1990-х годов индустрия ПО стала достаточно развитой, сложные проекты успешно реализовывались с помощью приобретающей популярность объектно-ориентированной методологии, а команды разработчиков стали применять подходы, основанные на использовании наиболее значимых преимуществ предыдущих моделей.

#### Объектно-ориентированная модель.

#### Объектно-ориентированная модель разработки была создана на основе объектно-ориентированной парадигмы программирования. Изначально эта методология представляла из себя копирование принципов объектно-ориентированного программирования, но позже распространилась на весь жизненный цикл, и включила в себя действия и нотацию языка. Объектно-ориентированная разработка не опирается ни на какой язык программирования и может даже применяться не к объектно-ориентированному программированию. Данная модель, благодаря своим формальным конструкциям, позволяет понять многие различные аспекты и упростить дальнейшую реализацию, тестирование и развитие версий.

#### Очень популярна OMT (Object Modeling Technique) методология, поддерживающие первые две стадии жизненного цикла. Эта методология получила большое распространение благодаря своей системе графических обозначений.

#### Итеративная модель.

В 1995 году Филипп Кратчен объединил[32] преимущества спиральной инкрементной и объектно-ориентированных методологий, предложив итеративную модель. Эта модель включает в себя 4 фазы жизненного цикла разработки ПО:

• Начало

• Исследование

• Построение

• Внедрение

На каждой фазе процесс итеративно изменяется: сначала разрабатываются прототипы, уточняются требования, решаются наиболее трудные аспекты; завершающие итерации расширяют функциональность и приводят к созданию продукта. Так же модель включает в себя две группы процессов:

• Рабочие – управление требованиями, анализ, проектирование, реализация, тестирование, развертывание;

• Вспомогательные – управление конфигурацией и изменениями, проектом и процессом.

Число и назначение процессов определяются непосредственно задачей и могут иметь свои циклы, соответствующие основным фазам. Управление рисками здесь схоже со спиральной моделью – приоритеты и трудозатраты определяются в начале каждой итерации. Эта модель хорошо подходит для большинства программных разработок, а особенно для версионных продуктов.

### Модели быстрой разработки.

**Экстремальное программирование**

Экстремальное программирование – методология, представленная Кентом Беком, Уордом Каннингемом, Мартинов Фаулером и их коллегами в 1996 году. Эта методология является одной из наиболее гибких в разработке ПО и состоит из 12 приёмов в 4 группах

* Короткий цикл обратной связи (Fine scale feedback)
  + Разработка через тестирование (Test driven development) – включает в себя модульное тестирование (unit testing) и функциональное тестирование. Так же приоритетным является подход TDD (Test Driven Develoopment) – сначала пишется тест, а потом для его прохождения разрабатывается логика.
  + Игра в планирование (Planning game) – создание приблизительного плана в котором заказчик берёт на себя принятие бизнес-решений, а разработчики отвечают за решения технические.
  + Заказчик всегда рядом (Whole team, Onsite customer)
  + Парное программирование (Pair programming) – предполагает работу двух программистов за одним компьютером: один пишет, другой смотрит на общую картину разработки.
* Непрерывный, а не пакетный процесс
  + Непрерывная интеграция (Continuous Integration) – интеграция выполняется не как в других методах, в конце разработки, а несколько раз в день, после тестирования модулей.
  + Рефакторинг (Design Improvement, Refactor) – улучшение кода без затрагивания функциональности, что не позволяет коду «деградировать».
  + Частые небольшие релизы (Small Releases) – этот приём позволяет решить две проблемы: первая – это начать получать прибыль от продукта раньше, а вторая, но не менее важная, - так же раньше получать информацию о соответствии продукта от заказчика.
* Понимание, разделяемое всеми
  + Простота (Simple design) – проектирование в экстремальном программировании выполняется этапами, а не сразу целиком в начале проекта.
  + Метафора системы (System metaphor) – это представление системы для понимания, как она работает, какие модули компоненты в какой форме находятся.
  + Коллективное владение кодом (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership) – каждый член команды ответственен за весь исходный код, он может вносить правки в любой участок программы. Избежать ошибок в разработке позволяет модульное тестирование, а применение такого подхода ускоряет процесс разработки.
  + Стандарт кодирования (Coding standard or Coding conventions) – команда формирует общие указания, которые соблюдаются потом каждым её членом. Это позволяет производить качественный рефакторинг, избежать споров, увеличить эффективность и унифицировать разработку в целом.
* Социальная защищенность программиста (Programmer welfare):
  + 40-часовая рабочая неделя (Sustainable pace, Forty hour week)

**SCRUM (Agile)**

Scrum – это быстрый, адаптивный эмпирический метод разработки. Это логическое противостояние водопадной модели, а методология строится на повторяющихся циклах, что позволяет методу самоорганизовываться, быть гибким и предсказуемым. Метод по своей работе основан на встречах: ежедневных и циклических 30-ти дневных. Каждый день решаются вопросы: что было сделано? Что будет делаться? Что мешает?

Scrum основан на принципах индивидуализма и взаимодействия строгих методов, и процессов. Главным в методе ставится работающее ПО, а не сложная документация, а взаимодействие с заказчиком выходит на первый план перед контрактными договорённостями. Главным представлением Srum является то, что задача важнее плана.

Построение SCRUM-команды - это 5-9 человек среди которых Scrum Master – менеджер и локальный лидер. На встречах могут присутствовать люди, но они будут лишены права голоса. Говорить могут только члены команды. Команда строится не только из программистов, но так же и тестировщики, дизайнеры и другие заинтересованные люди.

В конце каждого 30-ти дневного Sprint`а разработчик демонстрирует заказчику рабочий прототип, требования к которому были сформулированы в начале Sprint`а в бэклоге – таблице соотношения задачи – времени.

Успех данного подхода во многом определяется личными качествами Scrum Master`а.

### Адаптированные и комбинированные модели.

Модели разработки, определяющие жизненный цикл, эволюционируя, не заменяя старые, а организовываясь под собственную сферу использования. Не все методики могут применяться в любом случае, иногда они могут быть даже тормозящим или тупиковым процессом. Иногда для идеального решения задачи может подойти один и только один метод. Руководство разработки должны рассматривать все варианты задолго до начала проектирования продукта. Иногда модели приходится адаптировать модели под свои задачи, применять комбинации или даже создавать свой путь.

Быстрая разработка сменила консервативные процессы в историческом плане, но на практике старые методы адаптировались, изменились, и переняв эффективные приёмы, нашли применение в современных процессах

### 1.6 Выводы

После анализа большинства популярных методов разработки, и учитывая возможности разработки, наиболее очевидным является метод экстремального программирования. Данный подход позволяет быстро оценить результат разработки и вносить сложные изменения. Принцип разработки через тестирование является сильной стороной данного метода для разработки одним человеком. Сложности возникают в принципах коллективной работы, которые приходится нарушать, консолидируя всю работу в одном разработчике.

## Выбор языка программирования

Множество языков программирования ставят перед разработчиком проблему выбора, основанную на множестве различных факторов, таких как удобства, производительность и даже личные предпочтения.

Нужно понимать, что не существует одного языка, который станет наилучшем выбором для любой ситуации. В своём выборе можно отдавать предпочтения производительности или безопасности, количеству строк кода или удобному представлению структуры программы, но нужно понимать, что всегда придётся идти на компромисс. Правильный выбор позволит создать компактное, простое решение с лёгким документированием. Рассматривая качественную оценку языка можно выделить основные факторы:

* Целевая платформа
* Гибкость языка
* Время исполнения проекта
* Производительность
* Поддержка и сообщество

При выборе языка для персонального проекта можно применять личные предпочтения, где скорость реализации и количества кода будут для разработчика минимальными и являться приоритетной чертой выбора языка.

### 2.1. Анализ инструментальных средств

Рассмотрим популярные языки и программные среды с точки зрения приспособленности под различные классы задач.

**BASIC**

BASIC (англ. Beginner’s All-purpose Symbolic Instruction Code — универсальный код символических инструкций для начинающих; англ. BASIC — основной, базовый). Язык был разработан в 1963 году преподавателями Дартмутского Колледжа Джоном Кемени и Томасом Куртцом.

BASIC был спроектирован для обучения студентов без математического программирования и предназначен для разработчиков заинтересованых в простоте реализации, закрывая глаза на структурность, скорости работы и многие другие спекты. Со временем появилось множество диалектов и реализаций даного языка.

Восемь требования, представляшихся при разработке языка:

* простота в использовании для начинающих;
* общность назначения (отсутствие специализации);
* возможность расширения функциональности средствами, доступными программистам;
* интерактивность;
* четкие и понятные сообщения об ошибках;
* высокая скорость работы на небольших программах;
* отсутствие необходимости понимания работы аппаратного обеспечения для написания программ;
* эффективное посредничество между пользователем и операционной системой.

В 70-е годы Microsoft популяризовали этот язык, как бызовый для программирования на своих системах, а в 1991 появился Visual Basic – современная реализация, ставшая наболее полпулярной на плотформе Windows. Сегодня BASIC – это целое семейство языков с развитым деревом реализаций, но в своей основе это всё тот же язык, предназаченый для решения небольших прикладных задач.

**Pascal**

Pascal – один из наиболее популярных языков, так же как и BASIC направленного на обучение. Язык выделяется строгой типизацией и средствами процедурного программирования, став первопроходцем в этом направлении.

Паскаль (англ. Pascal) — язык программирования общего назначения. Один из наиболее известных языков программирования, используется для обучения программированию в старших классах и на первых курсах ВУЗов, является базой для ряда других языков.

Подробное описание всех недостатков привёл начале 1980-х Брайан Керниган в статье «Почему Паскаль не является моим любимым языком программирования».

Наиболее известной реализацией Паскаля, обеспечившей широкое распространение и развитие языка, является Turbo Pascal фирмы Borland, выросшая затем в объектный Паскаль для DOS (начиная с версии 5.5) и Windows и далее в Delphi, в которой были внедрены значительные расширения языка.

**C и C++**

В основе языка C - требования системного программиста: полный и эффективный доступ ко всем ресурсам компьютера, средства программирования высокого уровня, переносимость программ между различными платформами и операционными системами. С++, сохраняя совместимость с C, вносит возможности объектно-ориентированного программирования, выражая идею класса (объекта) как определяемого пользователем типа. Благодаря перечисленным качествам, C/C++ занял позицию универсального языка для любых задач. Но его применение может стать неэффективным там, где требуется получить готовый к употреблению результат в кратчайшие сроки, либо там, где невыгодным становится сам процедурный подход.

**Python**

В основе языка Python лежит принцип соединения нескольких парадигм программирования, таких как структурное, функциональное, императивное, объектно- и аспектно-ориентированниое. В языке применяется динамическая типизация, интроспекция, многопоточные вычисления, автоуправление памятью.

Python — один из наиболее динамичных языков совремнного времени. Новые версии выходят часто, поэтому отсутствуют стандарт ANSI, ISO или другие официальные стандарты.

**Perl**

Perl (Practical Extraction and Report Language, англ. - Практический Язык для Извлечения Данных и Составления Отчётов) разработан лингвистом Ларри Уоллом. Язык богат возможностями для работы с текстом и регулярными выражениями, встроенными в синтаксис.

Perl унаследовн от Си и является процедурным, реализуя переменные, присваивания, управляющие структуры и функции.

Общая структура Perl в общих чертах ведёт своё начало от языка Си. Perl — процедурный по своей природе, имеет переменные, выражения присваивания, блоки кода, отделяемые фигурными скобками, управляющие структуры и функции. Регулярные выражения хорошо работают для «парсинга текста»

Perl заимствует массивы из Lisp, регулярные выражения из AWK и sed, из AWK также позаимствованы хеши («ассоциативные массивы»). Регулярные выражения облегчают выполнение многих задач по парсингу, обработке текста и манипуляций с данными. В языке реализоана мощная функция автоматической типизации данных, но невозможные операции приводят к фатальным ошибкам.

**Java**

Объектно-ориентированный язык программирования, от компанией Sun Microsystems 23 мая 1995. Приложения транслируются в байт-код и работают внутри виртуальной машины. Это снимает зависимость кода от среды исполнения, но накладывает ограничение на наличие виртуальной машины, а любые неполномочные операции завершают выполнение виртуальной машины.

По данным сайта shootout.alioth.debian.org, Java в отдельных случаях в несколько раз медленнее C/C++, что в среднем в полтора-два раза больше, а потребление памяти Java-машиной было в 10-30 раз больше.

1. Выводы

В заключении заметим, что с профессиональной точки зрения не так важно на каком языке и в какой среде работает программист, сколько как он выполняет свою работу. Меняется аппаратура и операционные системы. Возникают новые задачи из самых различных предметных областей. Уходят в прошлое и появляются новые языки. Но остаются люди - те, кто пишет и те, для кого пишут новые программы и чьи требования к качеству остаются теми же вне зависимости от этих изменений.

# В. Охрана труда

## 1.1 Расчет защитного заземления

Защитное заземление – это преднамеренное электрическое соединение с землей или с ее эквивалентом металлических нетоковедущих частей, которые могут оказаться под напряжением (ГОСТ 12.1.030-81[16] ССБТ. Электробезопасность. Защитное заземление, зануление).

Назначение защитного заземления – исключение опасности поражения людей электрическим током при появлении напряжения на конструктивных частях электрооборудования, то есть при замыкании на корпус. Принцип действия защитного заземления – снижение до безопасных значений напряжений прикосновения и шага, обусловленных замыканием на корпус. Это достигается снижением потенциала заземленного оборудования, а также выравниванием потенциалов за счет поднимания потенциала основы, на которой стоит человек, к потенциалу, близкому по значению к потенциалу заземленного оборудования.

Рассчитаем систему защитного заземления, выполненную из вертикальных труб, соединенных ленточной шиной.

Характеристики заземляющего устройства:

* диаметр трубы *d* = 0,05 м;
* длина трубы ** = 2,4 м;
* величина заглубления *h* = 0,8 м;
* расстояние между трубами *a* = 4,5 м;
* ширина полосы *b* = 0,06 м;
* климатический коэффициент чернозема ;
* удельное сопротивление чернозема *ρ* = 200 Ом∙м.

Расчет заземления осуществляется в такой последовательности:

* определяют расчетное удельное сопротивление грунта;
* рассчитывают сопротивление растеканию тока одного вертикального заземлителя;
* определяют необходимое количество заземлителей и ориентировочное их расположение по периметру помещения, расстояние между ними;
* рассчитывают сопротивление растеканию тока соединительной шины;
* рассчитывают общее сопротивление заземляющего устройства с учетом соединительной шины.

Расчетное удельное сопротивление грунта, Ом⋅м, определяют по формуле

 Ом⋅м,

где *ρ* *–* удельное сопротивление глины по измерениям;

*φ* *–* климатический коэффициент, который зависит от характера грунта и его влажности во время измерений.

Расстояние от поверхности земли к середине заземлителя определим по следующей формуле:

 м,

где *h* – величина заглубления;

 – длина трубы.

Сопротивление растеканию тока, Ом, одного вертикального стержневого (трубчатого) заземлителя при углублении составит:

 Ом,

Ориентировочное количество вертикальных заземлителей, шт.:

 шт,

где *Rн* *–* наибольшее допустимое сопротивление заземляющего устройства (в соответствии с правилами устройства электроустановок *R*н = 4 Ом).

Путем расположения полученного количества заземлителей на плане определяем ориентировочно расстояние между ними и коэффициент использования вертикальных заземлителей η в зависимости от количества стержней и отношения расстояния между ними к их длине.

Рассчитаем необходимое количество заземлителей с учетом коэффициента использования.

Определим величину соотношения расстояния между трубами к их длине

Рассчитаем необходимое количество заземлителей с учетом коэффициента использования.

Определим величину соотношения расстояния между трубами к их длине



По соотношению  и величине  определим коэффициент использования заземлителей  – для заземлителей, размещённых в ряд.

Необходимое количество заземлителей с учетом коэффициента использования 

 шт.

Сопротивление растеканию тока соединительной шины при углублении с учетом коэффициента его использования =0,42 Ом:

,

где *L* – длина шины;

b – ширина шины;

h – глубина закапывания шины.

Длина шины *L* может быть определена по следующей формуле:

 м,

где а – расстояние между заземлителями.

Тогда получим

 Ом.

Общее сопротивление сложного заземляющего устройства, Ом:

;

 Ом.

Поскольку расчётное сопротивление Ом меньше нормативного  Ом, то, следовательно, сопротивление рассчитано верно и рассчитанная система заземления обеспечивает условие безопасной работы с оборудованием.

## 1.2 Электробезопасность рабочего места с ПЭВМ

Электрические устройства, к которым можно отнести оборудование ПЭВМ, представляют для человека потенциальную опасность. Воздействие тока может привести к электрическим травмам, то есть повредить организм электрическим током или электрической дугой которая возникает в аварийных режимах (ГОСТ 12.1.009-76[14]). Очень важное значение для исключения электрического травматизма имеет правильная организация обслуживания действующих электрических установок, установленная «Правилами технической эксплуатации электроустановок потребителей» (ПТЭ) и «Правилами устройства электроустановок» (ПУЭ). Помещения, где находятся рабочие места операторов, относятся к категории без повышенной опасности, данное оборудование относится к классу до 1000 В. Оператор работает с оборудованием на 220 В. Чаще всего происходят случаи прикосновения рукой или другими конечностями корпусов компьютеров и дисплеев. Для уменьшения риска электрического травматизма необходимо применять наиболее эффективный способ защиты. Такими способомами являются защитное заземление и зануление. Принципом действия защитного заземления является уменьшение во много раз тока, который протекает через человека в случае его утечки. Человек работающий оператором должен быть обучен всем правилам эксплуатации электрооборудования и оказанию первой медицинской помощи при поражении электрическим током.

В результате неправильной эксплуатации наружных электрических сетей, несвоевременного и низкого качества их ремонта, отсутствия контроля за состоянием трасс провода линий ЛЭП провисают или обрываются. Если человек дотрагивается до оборванного или провисшего провода велика вероятность получить травму.

Смертельно опасно не только дотрагиваться, но и подходить близко (8-10 м) к находящемуся на земле поврежденному проводу воздушной линии, в связи с тем, что в зоне растекания тока находится «шаговое» напряжение. Покидать зону распространения тока необходимо на сомкнутых вместе ногах или прыгая на одной из ног. Действие электрического тока в пределах 8-10 м от находящегося на земле провода сильно не ощущается.

Наиболее часто в связи с прикосновением к оборванным или провисшим проводам могут пострадать ребенок, в основном мальчики. Нельзя залезать на крыши домов, где неподалеку находятся электрический провод, играть рядом с воздушной линией, на опоры наружных электросетей, запускать там бумажного змея, открывать двери трансформаторных подстанций и т.п.

При нахождении оборванных или провисших проводов высоковольтной линии необходимо оградить место повреждения, сообщить людям о данной опасности и немедленно уведомить об аварии электромонтера или в районные электрические сети. Телефоны районных электрических сетей обязаны быть записаны на дверях шкафа трансформаторной подстанции, которая установлена на одной территории с поселком или садово-огородническим товариществом.

Есть вероятность поражения электрическим током в ситуации, если наружные электросети исправны, но расстояние от человека до провода минимально. Например, под проводами проходят работы длинномерными предметами или инструментами; вблизи ответвлений и вводов в здание неправильно установлены теле- и радиоантенны; с деревьев в саду снимают плоды с применением длинных металлических предметов; производится обрезка крон деревьев, растущих под проводами, и т.д. В любом из перечисленных случаев человек находится на земле (проводнике электрического тока), а его близкое положение к голым токоведущим частям или к участкам провода с поврежденной изоляцией может привести к несчастному случаю.

Таким образов, опасность поражения током может зависеть от следующих факторов: напряжения сети, качества заземления, «схемы включения» человека в электрическую сеть. Сети с глухозаземленной нейтралью применяются в местах, где нет возможности обеспечить правильную изоляцию или быстро найти и устранить повреждение изоляции. К данным сетям можно отнести и распределительные сети в сельской местности, где люди отстраивают коттеджи, осваивают садово-огородные участки

Техническими способами электрозащиты являются:

1) зануление;

2) использование безопасного напряжения (12-42 В);

3) использование потребителей с двойной изоляцией.

Главная мера защиты от повреждений током в случае прикосновения к корпусам электрооборудования, находящегося под напряжением из-за повреждения изоляции, - это зануление, т.е. специальное электрическое соединение с нулевым защитным проводником металлических нетоковедущих частей, которые могут оказаться под напряжением. Данное соединение превращает всякое замыкание токоведущих частей на землю или на корпус в однофазное короткое замыкание.

Допустим, человек стоит на деревянном полу в сухой резиновой обуви, которая не проводит электрический ток. По закону Ома сила тока прямопропорциональна напряжению и обратнопропорциональна сопротивлению участка цепи: I = V/R, где V - напряжение 220 В; R - сопротивление, равное сумме последовательных сопротивлений:

* тела человека - 1000 Ом;
* обуви человека - 45000 Ом;
* пола - 100000 Ом;
* сопротивления зануления, составляющего от суммы приведенных сопротивлений менее 1% (не учитывается).

Всего 146000 Ом.

Таким образом, сила тока, который проходит через человека, равна 220/146000 = 0,0015 А.

Данный ток безопасен для человека. А в случае, если плитка не подсоединена к системе зануления (позиция В), человек находится на сырых токопроводящих полах, сопротивление которых условно можно принять равным 0, человек в токопроводящей обуви, сопротивление которой также условно принимаем равным 0. В этом случае при нарушении изоляции фазного провода напряжение попадает на корпус плитки, проходит сквозь человека, токопроводящую обувь, токопроводящие полы и на землю. Проходящий через человека ток будет иметь смертельно опасное значение: 220 / 1000 = 0,22 А.

Поражение человека электрическим током часто случается при единовременном прикосновении его к корпусу электрического прибора, на котором из-за неисправности находится фазовое напряжение, и к естественному заземлителю, которым в бытовых условиях оказывается труба отопления, водопровода, канализации и др. В данном случае ток проходит от корпуса неисправного прибора через тело человека на естественный заземлитель.

# Г. Экологическая часть

## 2.1. Микроклимат

В помещениях нужно создавать поступление свежего воздуха, количество которого определяется технико-экономическим расчетом и выбором схемы вентиляции и кондиционирования. Минимальный расход воздуха определяется из расчета 20-60 куб.м /ч, но не менее двукратного воздухообмена в час. Вентиляция - организованный воздухообмен, заключающийся в исключении и удалении из рабочего помещения загрязненного воздуха и подаче вместо него свежего наружного и очищенного воздуха. В зависимости от назначения, вентиляция бывает:

• приточная;

• вытяжная.

В зависимости от способа перемещения воздуха вентиляция бывает:

• естественная;

• искусственная.

Параметры воздуха, поступающего в приемные отверстия и проемы местных отсосов технологических и других установок, расположеных в рабочей зоне помещения, следует принимать в соответствии с ГОСТ 12.1.005-88[27]. При размерах помещения 8 на 5 метров и высоте 3 метра, его объем 120 куб. м. Таким образом, вентиляция должна обеспечивать расход воздуха в 240 куб. м./час. В летнее время следует рассмотреть установку кондиционера с целью ограничения превышения температуры в помещении для устойчивой работы оборудования. Необходимо уделить должное внимание количеству пыли в воздухе, так как это непосредственно влияет на надежность и ресурс эксплуатации ПЭВМ.

Освещенность рабочего места – один из важных факторов релизации нормальных условий труда. Правильно продуманное и сделанное освещение может обеспечить хороший уровень работоспособности и производительности, создает положительное психологическое воздействие на оператора, способствует повышению эффективности труда. Для ВЦ о важности вопросов производственного освещения говорит и тот факт, что условия деятельности операторов в системе «оператор - машина» связаны с явным преобладанием зрительной информации - до90% общего объема. Освещение нормируется согласно СНиП 11-4-79[20], согласно которому, освещенность должна быть не менее 300 лк.

По конструктивному выполнению искусственное освещение может быть общим или комбинированным. При общей освещенности каждое рабочее место получает рабочее освещение от общей установки. Комбинированное освещение наряду с общим включает в себя локальное освещение рабочего места. Следуя требованиям отсутствия бликов и равномерности освещения, лучше выбирать общее искусственное освещение помещения.

Для искусственного освещения помещения необходимо применить в основном люминесцентные лампы, которые обладают высокой световой отдачей и потоком, продолжительным сроком службы, малой яркостью светящейся поверхности, близким к естественному спектральному составу света. Наиболее используемы лампы ЛБ (белый свет) и ЛТБ (тепло-белый свет) мощностью 20, 40 или 80 Вт. Система общего искусственного освещения должна быть выполнена потолочными встроенными или подвесными лампами, находящиеся параллельно светопроемам и равномерно по потолку. В связи с тем, что многие люди видят мерцание люминесцентных ламп, работающих от сети 50 Гц, некоторые специалисты хотят полностью исключить их или заменить на соответствующие более высокочастотные.

Во избежании отражений и бликов, снижающих четкость восприятия, не стоит размещать рабочее место сразу под источником света. Следуя вышеперечисленным условиям, выбираются светильники дневного света УСП-35 открытого типа.

Метеоусловия производственной среды (согласно ГОСТу 12.1.005-88[27]) - это сочетания температуры, относительной влажности, скорости движения и запыленности воздуха. Данные параметры оказывают существенное влияние на функциональную деятельность человека и на его производительность, его самочувствие и здоровье и на надежность работы средств вычислительной техники.

Параметры микроклимата в помещении нормируются согласно СН 512-78.

В помещении необходимо поддерживать содержание:

1. Кислорода - 21-22 об. %; озона - не более 0.1 мг/куб.м;
2. Легких ионов - 1500-3000 положительных и 3000-5000 отрицательных в 1 куб. см. воздуха.

Для отделки интерьера помещения невозможно использовать строительные материалы, содержащие органическое сырье: ДСП, декоративного бумажного пластика, поливинилхлоридных пленок, моющихся обоев и др. Для обеспечения надлежащего качественного (в т.ч. аэроионного и непыльного) состава воздуха необходимы:

• систематические проветривания;

• влажная ежедневная уборка;

• ежемесячное протирание спиртом клавиатуры и экрана;

• наличие приточно-вытяжной вентиляции;

• установка увлажнителей;

• установка автономных кондиционеров в оконных рамах, число которых определяется согласно расчету воздухообмена по количеству теплоизбытков от машин, людей и солнечной радиации.

Для исключения ухудшения микроклимата (и освещение) влияния солнечной радиации на окнах должны быть предусмотрены шторы, занавески или жалюзи.

## 2.2 Защита от шума

С физиологической точки зрения шум понимают, как звук, мешающий разговорной речи и негативно влияющий на слух человека. Основными физическими величинами, характеризующими шум и звук в любой точке помещения, с точки зрения воздействия на человека, можно считать:

• частота;

• интенсивность;

• звуковое давление.

В соответствии с ГОСТ 12.1 003-83[3], защита от шума, создаваемого на рабочих местах осуществляется следующими методами:

• уменьшением шума;

• применение средств коллективной защиты (ГОСТ 12.1.0280[4]);

• применение средств индивидуальной защиты;

• рациональная планировка рабочего места;

• акустическая обработка помещения.

Для исключения шума необходимо применять следующие меры:

• увеличение звукоизоляции;

• уплотнение по дверей, перекрывающих проходы в помещении;

• уменьшение шума источников с помощью применения прокладок из эластичных материалов.

В качестве звукопоглощающих конструкций можно использовать:

• маты из стекловолокна;

• перфорированные плиты.

Для оценки звукопоглощающей способности используют понятие звуконепроницаемости, численно равной отношению звуковой энергии, прошедшей через ограждение к падающей на него. Нормирование уровня шума для персонала, осуществляющего эксплуатацию ЭВМ, производится согласно ГОСТ12.1 003-83[3] по которому допустимый уровень шума составляет около 50 Дб.

Эргономика и эстетика рабочего помещения являются составными частями культуры производства, т.е. комплекса мер по организации труда, направленных на улучшение рабочей обстановки. В основе повышения культуры производства лежат требования научной организации трудовой деятельности. Культура производства достигается правильной организацией трудовых отношений и отношений между работающими, благоустройством рабочих мест и эстетическим преобразованием окружающей среды.

Эргономика - наука, которая изучает функциональные возможности человека в трудовых отношениях с точки зрения физиологии и психологии в целях создания средств, орудий и условий труда, а также технических процессов, наиболее соответствующих высокой производительности труда человека. Огромную роль играет состояние рабочего места и помещения, которое должно соответствовать требованиям удобства выполнения работ и экономии энергии, и времени рабочего, рационального использовании производственных площадей и удобства обслуживания устройств ЭВМ.

Во время работы часто возникают ситуации, в связи с которыми оператор ЭВМ обязан за минимальный срок времени принять правильное решение. Для успешного труда в таких условиях необходимы рационально организованная окружающая среда, ограничивающая оператора от воздействия посторонних раздражителей, которыми могут быть: окраска стен, неудобное расположение различных устройств рядом, клавиш управления и т.д. Поэтому всеми средствами нужно снижать утомление и напряжение оператора ЭВМ, создавая обстановку производственного уюта.

Производственная среда, которая является предметным окружением оператора, обязана сочетать в себе рациональное архитектурное и планировочное решение, оптимальные санитарно - гигиенические нормы, которыми являются микроклимат, освещение, вентиляция, научно обоснованную цветовую окраску и создание высоко моральной системы интерьеров в рабочем помещении.

Оператор работает с ПЭВМ в диалоговом режиме и главным источником информации для него является монитор компьютера.

Ряд исследований, проведенных учеными различных стран, показали связь между работой на компьютере и такими состояниями организма, как уставшие глаза, другие болезненные ощущения в зрении человека, боли в спине, пояснице и шее, запястный сидром или болезненное поражение нервов запястья и другие нарушения в нервно-мышечном аппарате, стенокардия, стрессы и другие неблагоприятные изменения функционального состояния нервной системы оператора.

Какую бы тревогу не вызывали некоторые статистические данные, следует помнить, что многих недомоганий, связанных с работой за компьютером, можно избежать. Зная наиболее распространенные причины компьютерных заболеваний можно избежать их, полностью изменив устройство рабочего места, стандартный ритм работы и интерьер окружающей среды.

На сегодняшний день специалисты в области эргономики труда уже осознали, что нет возможности находиться в таком положении, чтобы можно было бы пребывать и работать в течение всего рабочего дня. Для многих людей удобное рабочее место – это место, которое можно приспособить не менее чем для 2-3 позиций тела. При этом положение стула, дисплея и клавиатуры должны каждый раз соответствовать характеру выполняемой работы, антропологическим данным и привычкам оператора и исключать неудобные позы и длительные напряжения организма. Например, многие считают, что для работы на компьютере больше всего подходит вертикальное положение со слегка наклоненным вперед сидением.

Положение тела оператора обычно соответствует направлению взгляда. Мониторы, расположенные слишком низко или под неправильным углом, являются основными причинами появления сутулости и заболеваний спины. Уровень глаз должен приходиться на центр экрана и на 2/3 высоты экрана. Линия взгляда должна быть перпендикулярна центру монитора, и оптимальный ее наклон в вертикальной плоскости должен находиться в пределах 5 град., допустимое 10 град. Оптимальный обзор в горизонтальной плоскости от центральной оси экрана должен быть в пределах 15 град., допустимый 30 град. При рассматривании информации, находящейся в крайних положениях дисплея, угол рассматривания, ограниченный линией взора к поверхностью экрана, должен быть не менее 45 град. Чем больше угол обзора, тем легче воспринимать информацию с экрана и меньше будут уставать глаза. Для людей носящих очки, угол между направлением прямого взгляда и взгляда на экран может быть больше. Расстояние от монитора до глаз должно лишь немного превышать привычное расстояние между книгой и глазами, т.е. оптимально 60-70 см, допустимо не менее 50 см.

Например, для режима 25 строк по 80 символов на экране дисплея персонального компьютера при S=3 мм минимальное расстояние L должно быть около 51.6 см.

У кресла должны быть подлокотники и подъемно-поворотное устройство для регулирования высоты сидения и спинки, а также угла наклона спинки. Желательно, чтобы поверхность спинки кресла повторяла форму спины. Высота поверхности сидения должна регулироваться в пределах 30-60 см., угол наклона спинки - в пределах 90-100 град. Ширина и глубина сидения должна быть в пределах 40 см. Высота опорной поверхности спинки - не менее 30 см., а ее ширина - не менее 38 см.

Материал покрытия рабочего места должен давать возможность легкой очистки от загрязнения. Поверхность сидения и спинки должна быть мягкой, с нескользящим, не электризующимся и воздухонепроницаемым покрытием.

Стул либо кресло необходимо устанавливать на такой высоте, чтобы не ощущалось давление на копчик, возникающее при низком расположении стула или при слишком высоком расположении. Хоть и большинство операторов ЭВМ предпочитает сидеть за рабочим местом немного откинув спинку кресла назад, специалисты по эргономике уверяют, что угол между ногами и позвоночником должен равняться ровно 90 град.

Работающий за терминалом оператор должен стараться сидеть прямо, опираясь в области нижней части лопаток на спинку стула, не сгибаясь, с несильным наклоном головы вперед (до 8 град.). Предплечья должны опираться на поверхность кресла, убирая этим статическое напряжение плечевого пояса и конечностей рук. Руки необходимо располагать таким образом, чтобы они находились на расстоянии нескольких десятков сантиметров от тела. Стул и клавиатура устанавливаются так, чтобы не приходилось тянуться на некоторое расстояние от края. При изменении положения туловища, например с вертикального на наклонное, обязательно следует изменить и положение клавиатуры на столе. При этом удобно пользоваться регулируемой подставкой клавиатуры, но можно поставить клавиатуру и на колени.

Кроме того, многие разновидности профессиональных заболеваний операторов компьютеров можно предотвратить, используя так называемую переламываемую клавиатуру, при применении которой кисти во время работы повернуты друг к другу. Несколько исследований, прошедших в Германии, доказали, что благодаря такому расположению рук сильно уменьшается нагрузка, приходящаяся на верхнюю часть тела.

Длина рабочей поверхности стола (слева направо) должна быть не менее 60 см, ширина стола должна обеспечивать место перед клавиатурой не менее 30 см для расположения документов, текста программы, устройств и др. Поверхность стола, на которой располагаются клавиатура и документы, должна иметь наклон 10-13 град.; в некоторых случаях допускается и горизонтальная поверхность рабочего стола. Высота края стола, обращенного к оператору за видеотерминалом, кресла или стула над полом и ширина пространства для ног под столом должны приниматься в зависимости от ростом рабочего.

Ширина пространства для ног под рабочим столом должна быть не менее 60 см., глубина - не менее 50 см. Удобная высота стола очень важна в том случае, когда на нем располагается клавиатура. Если стол слишком высокий и его нельзя уменьшить, а у клавиатуры отсутствует или довольно низкая подставка, следует повыше поднять сидение стула, а под ноги подставить скамеечку или что-то другое что может увеличить высоту. Если стол слишком низкий, нужно что-нибудь подложить под его ножки.

Согласно санитарным требованиям и правилам для операторов вычислительных центров при вводе информации, редактировании данных, чтении с экрана непрерывная продолжительность работы с видеотерминалом не должна превышать 3-4 часов. Для снижения усталости и напряженности труда необходимо по возможности равномерно распределять нагрузку и рационально изменять характер деятельности.

Через каждый час работы необходимо чтобы перерыв был на 5-10 минут, а через 2 часа - около 15 минут. Несколько раз в час необходимо выполнять серию легких упражнений на растягивание, которые могут способствовать уменьшению напряжение, которое накапливается в мышцах при достаточно длительной работе на компьютере.

Не следует делать более 10-11 тысяч нажатий на клавиши в час или 30 тысяч за 4 часа работы, это достаточно много.

В целях профилактики и устранения усталости и перенапряжения желательно после окончания рабочего дня и во время перерывов проводить сеансы психофизиологической разгрузки организма и снятия переутомления.

# Д. Решение задачи на ЭВМ

## 1. Структура классов

## 2. Пример функционирования

Для примера использования нашей библиотеки была использована тестовая кластеризация. Для начальных данных были взяты публичные наборы атрибутов грибов. Набор составлял более 1000 векторов с пропущенными атрибутами. На каждый объект приходилось порядка 25 атрибутов. Задача заключалась в кластеризации данных, определяющей пригоден ли гриб к питанию. В результате приемлемые результаты были получены уже на первой итерации метода CLOPE. Для остановки разделения потребовалось три итерации с параметром repulsion = “2.6”.

Алгоритм BIRCH + k-means хорошо показал себя на тестовых данных об автомобилях. В тестовый набор были включены 1400 наборов данных по 90 атрибутов без пропуска. В результате кластеризации автомобили были разбиты на 5 кластеров в которых можно было проследить соотношение стоимости автомобиля его характеристикам.

Библиотека так же хорошо работает на данных о страховой стоимости, или для выделения групп в маркетинговых исследованиях. Но все же самым главным в реализации кластеризации является правильное понимание области данных экспертом, формирующим исходную выборку. Правильно предоставленные данные могут давать незначительные отклонения при ошибочно выбранных начальных параметров алгоритмов. Представленные алгоритмы хорошо выделяют выбросы и аномалии. Так в случае с кластеризацией грибов, многие объекты имели пропущенный атрибут, но это всё равно не повлияло на результат работы в следствии устройства структуры алгоритма, работающей непосредственно с параметрами кластера, а не самими объектами.

# Е. Заключение

В результате дипломного проектирования были рассмотрены методы кластерного анализа, популярные продукты для кластеризации данных и их применение. На основании чего было установлено, что множество систем реализованы под конкретные задачи. Во время разработки библиотеки были рассмотрены методы, применяемые в современных системах, из которых выбрались 3 основных метода из разных групп. Реализация методов была успешно проведена в с применением фреймворка Qt на языке С++. Для меня встала задача не только кластеризации, но и разработки модуля для проведения экспертных оценок на основании пользовательских запросов. Данная задача была решена при помощи введения системы ценовых моделей как в «ручном», так и в «автоматическом» режимах. После применения ценовых моделей формировалась итоговая таблица взвешенных атрибутов непосредственно к которым применялся один из выбранных алгоритмов кластеризации. После успешной кластеризации формируется таблица атрибутов кластеров на основании которой пользователь может делать запросы, в которых предоставляя вектор с полной информацией, мог узнать к какому кластеру наиболее вероятно может быть отнесен данный объект. Либо при введении кластера с неполной информацией, происходит соотнесение его с наиболее вероятным кластером, и на основании содержащихся там элементов происходит прогнозирование неизвестного атрибута на основании среднего значения.

В итоге была реализована библиотека, которую можно применять в больших системах для проведения анализа данных и поиска неочевидных связей.

# Ж. Список использованной литературы

1. ССБТ. Ультразвук. Общие требования безопасности. ГОСТ 12.1.001-89

2. Санитарные нормы. Гигиенические нормы инфразвука на рабочих местах. СН № 2274-80

3. ССБТ. Шум. Общие требования безопасности. ГОСТ 12.1.003-83 (1999)

4. ССБТ. Средства ми методы защиты от шума. Классификация ГОСТ 12.1.029-80 (2001)

5. СНиП Защита от шума. СНиП 11-12-77

6. ССБТ. Шум. Допустимые уровни в жилых и общественных зданиях. ГОСТ 12.1.036-81 (2001)

7. ССБТ. Методы измерения шума на рабочих местах ГОСТ 12.1.050-86 (2001)

8. ССБТ. Средства защиты рук от вибрации. ГОСТ 12.4.002-97

9. ССБТ. Вибрация Средства измерения и контроля вибрации на рабочих местах. Технические требования. ГОСТ 12.4.012-83 (1986)

10. Трудовой кодекс Российской Федерации от 30.12.2001 г. 197-ФЗ

11. ССБТ. Общие требования к системе управления охраной труда в организации. ГОСТ Р 12.0.006-2002

12. ССБТ. Опасные и вредные производственные факторы. Термины и определения. ГОСТ 12.0.002-80\*

13. ССБТ. Опасные и вредные производственные факторы. Классификация. ГОСТ 12.0.003-74 (99)

14. ССБТ. Электробезопасность. Термины и определения. ГОСТ 12.1.009-76 (1999)

15. ССБТ. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. ГОСТ 12.1.038-82

16. ССБТ. Электробезопасность. Защитное заземление. Зануление. ГОСТ 12.1.030-81 (2001)

17. ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты. ГОСТ 12.1.019-96

18. ПУЭ (правила устройства электроустановок), утверждены приказом Минэнерго России от 9.04.2003 г. № 150

19. Санитарные правила и нормы. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий. СанПиН 2.2.1/2.1.1.1278-03

20. Санитарные правила и нормы. Искусственное освещение. СНиП 11-4-79

21. ССБТ. Лампы электрические. Требования безопасности. ГОСТ 12.2.0ССБТ. Ультразвук. Общие требования безопасности ГОСТ 12.1.001-89 (1999)

22. Правила технической эксплуатации электроустановок потребителей, утверждены Министерством энергетики 13 января 2003 г., №6

23. Санитарные правила и нормы. Гигиенические требования к микроклимату производственных помещений СанПиН 2.2.4.548-96

24. Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений. СанПиН 2.2.4.1294-03

25. ССБТ. Воздух рабочей зоны. Требования к методикам измерения концентрации вредных веществ. ГОСТ 12.1.016-79 (2001)

26. ССБТ. Вредные вещества. Классификация и общие требования безопасности. ГОСТ 12.1.007-82 (1999)

27. ССБТ. Общие санитарно-гигиенические требования к воздуху санитарной зоны. ГОСТ 12.1.005-88 (2001)

28. И. А. Чубукова Data Mining - Интернет-университет информационных технологий, Бином. Лаборатория знаний ISBN 978-5-94774-819-2; 2008 г.

29. Тоби Сегаран (пер. А. Слинкин) Программируем коллективный разум (Programming Collective Intelligence) Символ-Плюс ISBN 978-5-93286-119-6, 5-93286-119-3, 0-596-52932-5; 2008 г.

30. Дюк В., Самойленко А. Data Mining: учебный курс (+CD). - СПб.: Изд. Питер, 2001. - 368 с.

31. Гудков А. А. Известия Пензенского государственного педагогического университета им. В.Г. Белинского 2007

32. Питер Морвиль, Луис Розенфельд (пер. С. Маккавеев, Е. Смогайлов) Информационная архитектура в Интернете (Information Architecture for the World Wide Web) ISBN 978-5-93286-164-6, 978-0-596-52734-1; 2010 г.

33. Х. Марманис, Д. Бабенко (пер. М. Низовец) Алгоритмы интеллектуального Интернета. Передовые методики сбора, анализа и обработки данных (Algorithms of the Intelligent Web) Символ-Плюс, ISBN 978-5-93286-186-8, 978-1-933988-66-5; 2011 г.

34. Sudipto Guha, Rajeev Rastogi, Kyuseok Shim ROCK A Robust Clustering Algorithm for Categorical Attributes 0-7965-0071-4/99 1999 IEEE

35. Н. Паклин. «Кластеризация категорийных данных: масштабируемый алгоритм CLOPE». Ссылка: http://www.basegroup.ru/library/analysis/clusterization/clope/

36. Н. Паклин «Алгоритмы кластеризации на службе Data Mining». Ссылка: http://www.basegroup.ru/clusterization/datamining.htm

37. Tian Zhang, Raghu Ramakrishnan, Miron Livny «BIRCH: An Efficient Data Clustering Method for Very Large Databases». Электронное издание.

38. Daniel Fasulo «An Analysis Of Recent Work on Clustering Algorithms». Электронное издание

39. Фредерик Брукс Мифический человеко-месяц, или Как создаются программные системы Символ-Плюс ISBN 5-93286-005-7, 0-201-83595-9; 2010 г.

40. Dr. Winston W. Rovce ANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

41. Критика известных экспертов PMI концепции "водопада" в PMBOK 3 http://www.microsoftproject.ru/articles.phtml?aid=158#agile