

Реферат

Выпускная квалификационная работа состоит из 75 страниц, 9 рисунков, 3 использованных источников.

ГЕОЛОКАЦИЯ, КАРТОГРАФИЯ, ГЕНЕРАЦИЯ МАРШРУТОВ, ОПТИМИЗАЦИЯ МАРШРУТОВ, ВИЗУАЛИЗАЦИЯ КАРТ, ИНТЕРАКТИВНАЯ НАВИГАЦИЯ, ВЕБ-СЕРВИС

Объект разработки: интеллектуальная система генерации маршрутов для транспортных средств и пешеходов.

Цель работы: проектирование и реализация алгоритма генерации маршрутов с фиксированной дистанцией и заданной конечной точкой. Проектирование веб-сервиса и создание макетов бэкенд и фронтенд приложений. Методы проведения работы: в работе использовались методы машинного обучения, а также статистические методы анализа данных.

Результаты работы: был разработан веб-сервис, способный создавать уникальные маршруты, учитывающий пользовательские параметры. Для реализации оберточного слоя над алгоритмом генерации маршрута использовались следующие технологии: Python, Java, Spring Boot, TypeScript, React. Данный набор технологий позволяет построить минимальную рабочую версию продукта с минимальными временными затратами на проверку гипотезы о работоспособности построенного алгоритма. При неуспешном завершении алгоритма генерации маршрута система рекурсивно вызывает метод генерации маршрута с увеличенной погрешностью в целевой дистанции. Система также учитывает такие факторы, как штраф за повторное посещение вершины, удаленность достижимых вершин, штраф за отдаление от целевой вершины.

Область применения: система может быть использована в различных сферах, связанных с генерацией маршрутов для спортивных мероприятий, для генерации туристических маршрутов с обходом достопримечательностей. Также система может использоваться для навигации и построении кратчайших маршрутов.

Прогнозное предположение о развитии объекта разработки: возможны различные направления развития проекта, такие как расширение функционала и добавление новых возможностей, увеличение географического

охвата. Также возможно создание партнерских отношений с компаниями, работающих в сфере построения маршрутов, что может привести к новым возможностям и дополнительным источникам дохода.

ОГЛАВЛЕНИЕ

	Стр.
ВВЕДЕНИЕ	6
ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	7
1.1 Теория графов.....	7
1.1.1 Основные определения	7
1.1.2 Изоморфизм графов.....	11
1.1.3 Деревья	12
1.1.4 Клики	13
1.1.5 Расстояния на графе	14
1.1.6 Раскраска графов.....	15
1.1.7 Эйлеровы и гамильтоновы графы	16
1.1.8 Кратчайшие пути на графе	18
1.2 Вероятностные модели	24
1.2.1 Марковские цепи.....	24
1.2.2 Скрытые марковские модели.....	26
1.2.3 Марковские случайные поля	30
1.3 Хранение геоинформационных данных.....	38
1.3.1 OpenStreetMap	40
1.3.2 Структура данных OSM	42
1.3.3 Отображение карт	44
1.4 Графовые базы данных.....	44
1.4.1 Графовые базы данных для построения маршрутов	46
ГЛАВА 2 РАЗРАБОТКА БИЗНЕС-МОДЕЛИ С LEAN CANVAS	49
2.1 Проблема.....	50
2.2 Существующие альтернативы	52
2.3 Пользователи	53
2.4 Уникальная ценность	54
2.5 Решение	54
2.5.1 Методология проведения исследования	56
2.5.2 Результаты исследования	58
2.6 Каналы продвижения.....	61
2.7 Структура затрат и потоки прибыли	63
2.8 Ключевые метрики.....	67
2.9 Нерыночное конкурентное преимущество	68

ГЛАВА 3	АЛГОРИТМ ГЕНЕРАЦИИ МАРШРУТОВ	70
ГЛАВА 4	ОБЗОР ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ	72
ГЛАВА 5	АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА	73
ЗАКЛЮЧЕНИЕ	74
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	75

ВВЕДЕНИЕ

В современном мире навигационные приложения стали неотъемлемой частью повседневной жизни, облегчая перемещение и ориентацию в городской среде. Они помогают пользователям находить оптимальные маршруты для достижения разнообразных целей: поездок на работу, прогулок по городу, поиска интересных мест и даже спортивных тренировок. Несмотря на широкий спектр возможностей, предоставляемых навигационными приложениями, большинство из них уделяют основное внимание минимизации времени, что не всегда соответствует ожиданиям пользователей.

Сегодняшний активный образ жизни требует от нас более гибкого подхода к навигации. Многие люди стремятся не только эффективно перемещаться по городу, но и контролировать свою физическую активность и поддерживать ее на определенном уровне. Подход, рассматриваемый в данной работе, позволяет строить маршруты не только с целью достижения места назначения, но и построить его с учетом желаемой дистанции, что особенно актуально для занятий спортом, прогулок и здорового образа жизни.

В данной работе представляется новый подход к построению маршрутов на картах, который уделяет внимание заданной дистанции, которую пользователь желает преодолеть. Описанный алгоритм направленного блуждания на графе позволяет строить маршруты для активных прогулок, бега или велосипедных поездок.

Целью данной работы является разработка и описание алгоритма построения маршрутов с учетом заданной дистанции, а также демонстрация его применения в навигационном веб-приложении. В работе предоставляется подробное описание методологии алгоритма и его реализации, а также демонстрируется работа разработанного приложения. Данное приложение может быть востребовано среди пользователей, занимающихся циклическими видами спорта, а также среди людей, которые стремятся контролировать свою физическую активность.

ГЛАВА 1

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Теория графов

Графы – это фундаментальное понятие в дискретной математике, и теория графов играет важную роль в изучении различных аспектов их применения. Графы используются для моделирования и анализа множества реальных явлений, таких как сети, маршрутизация, социальные взаимодействия и многое другое.

В этом разделе мы рассмотрим основные понятия теории графов, которые помогут нам заложить базис для дальнейшего изучения этой темы.

1.1.1 Основные определения

Графы обеспечивают эффективное средство для описания бинарных отношений между различными объектами. Чтобы понять эту концепцию, можно взять пример из социальной сети: люди в этой сети представляют собой множество объектов, обозначаемых как V . В то же время отношения между этими людьми, такие как подписки или дружба, можно рассматривать как набор бинарных связей. Если объект v_i подписан на объект v_j , это означает наличие определенной связи между этими двумя людьми.

Для визуализации графов часто используется графическое представление, в котором объекты изображаются в виде кругов или узлов, а связи между ними – в виде линий или рёбер. Такой способ визуализации помогает лучше понять структуру графа и позволяет легче выявлять взаимосвязи между различными элементами.

В теории графов мы можем классифицировать графы по их структуре и направленности. Граф считается ориентированным, если он содержит непустое множество вершин V и множество отношений или рёбер $E \subset V \times V$, при этом каждое ребро можно трактовать как упорядоченную пару (v_i, v_j) . Это означает, что в ориентированном графе направление связи между вершинами имеет значение. В данном случае, если в графе есть

ребро из вершины v_i в вершину v_j , то может не быть обратного ребра из v_j в v_i . Такой граф часто называют "ориентированным графом" или "диграфом".

Если же граф содержит симметричные отношения между вершинами, он называется неориентированным. В неориентированном графе наличие ребра между вершинами v_i и v_j означает, что связь двусторонняя: если есть ребро (v_i, v_j) , то автоматически имеется и обратное ребро (v_j, v_i) .

Существует также класс графов, называемых смешанными графами. Они обладают одновременно ориентированными и неориентированными рёбрами, что придаёт им более сложную структуру и позволяет описывать более разнообразные ситуации.

На рисунке 1.1 представлен пример различных типов графов, показывающий, как можно визуально различать ориентированные, неориентированные и смешанные графы. Такой подход к изображению графов позволяет легко понять, какой тип отношений присутствует в каждом конкретном случае.

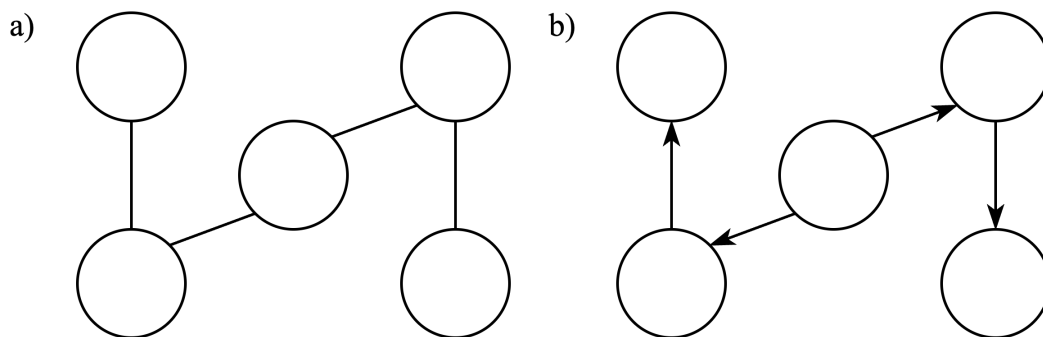


Рисунок 1.1 — Ориентированный граф (a), неориентированный граф (b)

Полный граф — это граф, в котором каждая вершина соединена со всеми другими вершинами. Это означает, что множество рёбер E эквивалентно декартовому произведению множества вершин $V \times V$. В таком графе между любой парой вершин существует прямое отношение или ребро, что создает сеть с максимальной плотностью связей. Полные графы часто используются в различных теоретических моделях и имеют ряд интересных свойств.

Взвешенный граф — это граф, в котором каждому ребру или вершине

присваивается некоторый вес или значение. Этот вес может представлять что угодно: от стоимости или длины пути до количества ресурсов или времени, необходимого для преодоления данного ребра или для работы с данной вершиной. Взвешенные графы позволяют анализировать графы с учетом дополнительных параметров и часто используются для задач, связанных с оптимизацией.

Путь в графе — это последовательность рёбер, которая соединяет ряд вершин, начиная с исходной вершины и заканчивая конечной. Формально, путь определяется как набор рёбер E_1, E_2, \dots, E_n , где конечная вершина одного ребра совпадает с начальной вершиной следующего. Путь считается простым, если никакое ребро не повторяется более одного раза. Элементарный путь — это путь, в котором каждая вершина посещается только один раз, что означает отсутствие повторяющихся вершин.

Цикл — это особый тип пути, который начинается и заканчивается в одной и той же вершине. Если граф является направленным, цикл может называться контуром. Циклы важны для выявления повторяющихся процессов или замкнутых систем в графе.

На изображении 1.2 показан пример пути, который является простым, поскольку каждое ребро используется только один раз, но не элементарным, так как центральная вершина посещается дважды. Такие примеры иллюстрируют различия между простыми и элементарными путями и показывают, как они могут применяться в различных контекстах анализа графов.

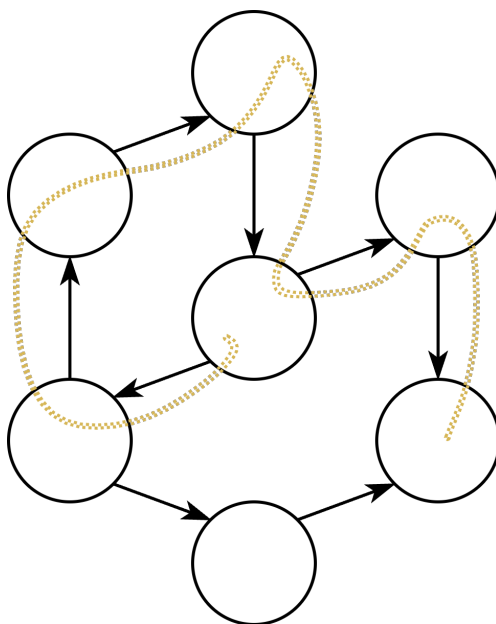


Рисунок 1.2 — Простой, но не элементарный путь в графе

Граф считается связным, если для каждой пары различных вершин в этом графе существует хотя бы один путь, соединяющий их. Это означает, что в связном графе можно переместиться из любой вершины к любой другой, следуя по рёбрам, которые соединяют вершины.

Связность является ключевым свойством графов, которое определяет их структуру и характер. В связном графе все вершины так или иначе соединены друг с другом, даже если путь между ними может проходить через другие промежуточные вершины.

Связные графы часто используются для моделирования систем, в которых элементы имеют некоторую степень взаимодействия или зависимости. Например, в социальной сети связный граф указывает на то, что существует путь коммуникации между любыми двумя участниками сети, даже если они могут быть связаны через нескольких посредников.

Если граф не является связным, то он состоит из нескольких отдельных компонент, каждая из которых представляет собой связный граф. Эти компоненты могут быть изучены отдельно, так как они не имеют связей друг с другом. Исследование связности графов позволяет выявлять изолированные сегменты или группы тесно связанных элементов.

1.1.2 Изоморфизм графов

Изоморфизм в теории графов — это понятие, которое указывает на структурное сходство между графами. Когда говорят, что графы G_1 и G_2 изоморфны, это означает, что между ними можно установить взаимно однозначное соответствие для вершин и рёбер. Иными словами, если имеется биекция между вершинами двух графов, которая сохраняет структуру рёбер, то такие графы считаются изоморфными.

Изоморфизм графов подразумевает, что их визуальная структура может отличаться, но по сути они идентичны. Например, если у одного графа вершины A, B, C соединены рёбрами в определенной последовательности, то в другом графе можно найти соответствие, в котором другие вершины X, Y, Z имеют те же рёбра в той же последовательности. Изоморфизм позволяет понять, когда два графа, несмотря на различие в именах или представлениях, имеют одинаковую структуру.

Подграфы — это части графа, которые включают некоторую подмножество вершин и рёбер из исходного графа. Если два подграфа из разных графов изоморфны друг другу, то их называют изоморфными подграфами. Такое понятие полезно при анализе сложных графов, когда нужно найти общие структуры или повторяющиеся паттерны.

Двойной изоморфизм подграфов — это ситуация, в которой два разных графа содержат подграфы, которые изоморфны друг другу. Это более широкий концепт, который указывает на то, что в двух графах есть общие структуры, но сами графы при этом могут не быть изоморфными.

Таким образом, изоморфизм и его вариации позволяют исследовать и выявлять структурное сходство в графах, что часто используется в таких областях, как компьютерные науки, биоинформатика и теория сетей.

На изображении 1.3 приведен пример двух изоморфных графов.

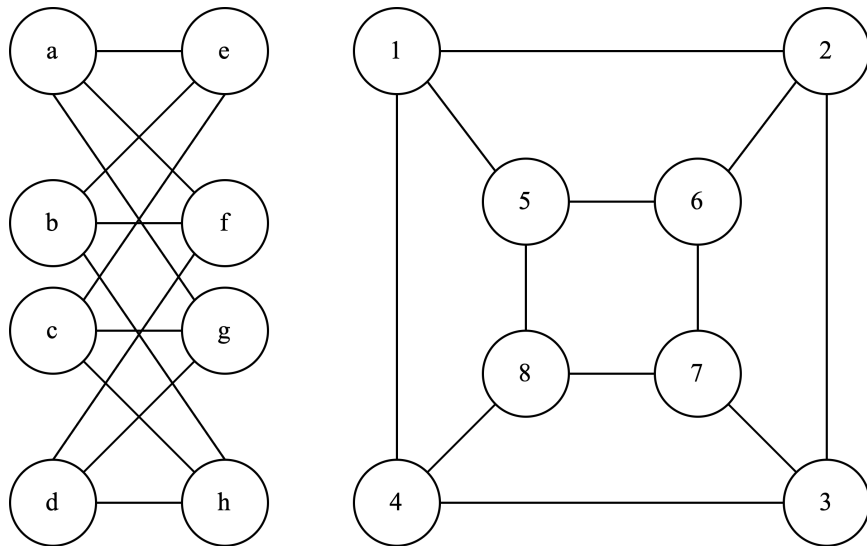


Рисунок 1.3 — Пример изоморфных графов

1.1.3 Деревья

Деревья представляют собой особый вид графов, которые широко применяются в различных областях информационных технологий, таких как алгоритмы, структуры данных. В этой работе деревья будут служить основой для алгоритмов генерации маршрутов.

Подобно графам, деревья делятся на два класса: ориентированные и неориентированные. Неориентированное дерево — это связный граф, который не содержит простых циклов. Это означает, что если начать обход по дереву из любой вершины, невозможно вернуться к исходной точке, не пройдя по рёбрам назад. Деревья уникальны своей структурой, поскольку они всегда связаны, но не имеют замкнутых путей.

В деревьях есть особые вершины, которые называются листовыми. Листовые вершины — это вершины, которые имеют степень, равную единице, то есть из них выходит только одно ребро. Они представляют собой конечные точки в дереве, и их наличие указывает на то, что это дерево имеет четко определенную структуру без циклов.

Остальные вершины в дереве, из которых выходит более одного ребра, называются внутренними вершинами. Внутренние вершины обеспечивают структуру дерева, соединяя между собой другие вершины и образуя пути,

которые могут вести к листовым вершинам.

В целом, деревья обладают рядом полезных свойств, которые делают их ключевым элементом в алгоритмах, особенно тех, что связаны с обработкой данных, маршрутами и поиском путей. Их использование в алгоритмах генерации маршрутов позволяет оптимизировать процесс поиска и обеспечить целостность структуры данных.

1.1.4 Клики

Полный граф — это особый вид графа, в котором каждый узел (вершина) соединён с каждым другим узлом. Это означает, что для любого набора из двух вершин в полном графе всегда существует ребро, соединяющее их. Полные графы могут быть представлены математически как графы, у которых множество рёбер E равно декартовому произведению множества вершин $V \times V$. Визуально, полный граф с n вершинами будет иметь $n(n - 1)/2$ рёбер.

Полное множество — это подмножество графа, которое само является полным графом. Другими словами, если выбрать определённое количество вершин из графа, и все они взаимно связаны рёбрами, то такое подмножество можно назвать полным множеством. Полные множества позволяют выделять небольшие полностью связанные участки в больших графах.

Клика — это специальный тип полного множества, который является максимальным. Это значит, что клика представляет собой полное множество, но при этом не существует другого полного множества большего размера, которое включало бы клик. Если добавить любую другую вершину в клик, она перестанет быть полным множеством, так как не будет полностью связанной. Клики полезны в графовых алгоритмах, особенно при анализе социальных сетей или других систем, где важна максимальная взаимосвязь между узлами.

Таким образом, полный граф, полное множество и клика представляют собой концепты, позволяющие описывать степень связи и максимальные подмножества в графах. Они используются в различных алгоритмах и

методах анализа графов, в частности при решении задач оптимизации, поиска общих связей и выявления групп узлов с сильной взаимосвязью.

На изображении 1.4 изображен пример клики.

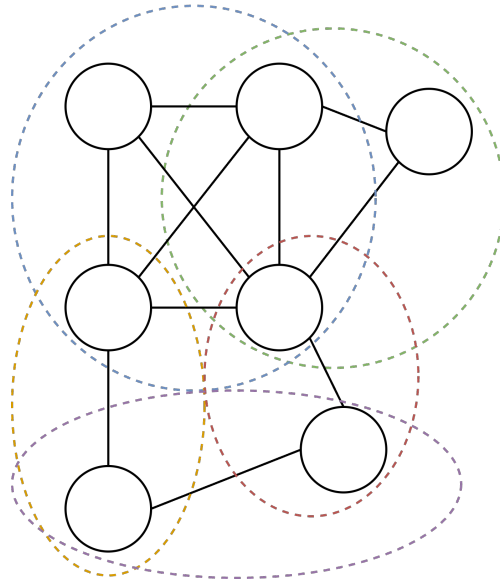


Рисунок 1.4 — Пример графа с 5 кликами

В этом графе есть пять клик. Среди них есть клика, состоящая из четырёх вершин, клика с тремя вершинами, а также три клики, в каждой из которых по две вершины.

1.1.5 Расстояния на графе

В теории графов расстояние между двумя вершинами отражает число рёбер, которые нужно пройти, чтобы соединить эти вершины на кратчайшем пути. Это значение позволяет оценить, насколько близко или далеко находятся вершины друг от друга. В невзвешенных графах расстояние определяется просто количеством рёбер на кратчайшем пути. Однако в взвешенных графах, где каждое ребро имеет свой вес, расстояние часто измеряют как сумму весов рёбер на кратчайшем пути.

Эксцентриситет $\epsilon(v)$ вершины v — это максимальное расстояние между данной вершиной и любой другой вершиной графа. Проще говоря, это показатель того, насколько далеко находится одна вершина от самой уда-

ленной вершины в графе. Эксцентриситет позволяет понять, насколько "центрально" или "отдалена" данная вершина относительно всех других вершин.

Радиус r графа — это минимальный эксцентриситет среди всех вершин, то есть $r = \min_{v \in V} \epsilon(v)$. Радиус показывает, насколько компактно расположены вершины графа. Если радиус мал, это указывает на то, что есть вершина, от которой все другие вершины относительно близко. Радиус имеет практическое значение, поскольку он определяет минимальное расстояние, на котором все вершины могут быть связаны.

Диаметр d графа, напротив, — это максимальный эксцентриситет среди всех вершин, то есть $d = \max_{v \in V} \epsilon(v)$. Диаметр представляет собой максимальное расстояние между любыми двумя вершинами в графе. Это значение важно для понимания "размера" графа в терминах наибольшего расстояния между двумя точками. Диаметр может использоваться для определения сложности или глубины структуры графа.

Итак, радиус и диаметр графа позволяют оценить, насколько "разбросаны" или "сконцентрированы" вершины, и применяются в различных анализах, от моделирования сетей до изучения маршрутизации и топологии графов.

1.1.6 Раскраска графов

Раскраска графа — это метод присвоения определенных меток, или "цветов" элементам графа, следуя определенным правилам. Это понятие широко используется в теории графов и находит множество практических применений в различных областях, от планирования и организации до компьютерных алгоритмов и анализа данных.

В самой базовой форме раскраски графа акцент делается на раскраске вершин таким образом, чтобы ни одна пара соседних вершин не имела одинакового цвета. Это означает, что для каждой вершины в графе ее цвет должен быть отличен от цветов ее соседних вершин. Такая раскраска называется раскраской вершин и представляет собой классическую проблему, часто рассматриваемую в контексте задач минимизации использо-

вания ресурсов или управления конфликтами.

Раскраска вершин позволяет решить проблемы распределения, где важно избегать конфликтов. Например, в задачах составления расписаний раскраска графа может помочь гарантировать, что определенные события или задачи не пересекаются по времени или ресурсам. В академическом контексте это может быть применено к составлению экзаменационных расписаний, чтобы избежать совпадений у студентов, записанных на те же курсы.

Другие формы раскраски графа включают раскраску рёбер, где цель заключается в том, чтобы рёбра с общими вершинами имели разные цвета. Такая раскраска может быть полезна для планирования или организации сетей, где рёбра представляют собой связи или пути, которые не должны пересекаться.

Сложность задачи раскраски графа зависит от структуры графа и количества доступных цветов. Известно, что раскраска графа — это NP-трудная задача, что означает, что не существует известного полиномиального алгоритма, который мог бы решить её для всех возможных графов. Тем не менее, разработаны эффективные эвристики и специальные алгоритмы, которые позволяют решать задачу раскраски для определенных классов графов или в приближении к оптимальному решению.

Таким образом, раскраска графа — это мощный инструмент, который может использоваться для решения широкого спектра проблем, требующих распределения ресурсов, избегания конфликтов и оптимизации.

1.1.7 Эйлеровы и гамильтоновы графы

Эйлеров путь (или эйлерова цепь) — это путь в графе, который проходит по каждому ребру ровно один раз. Эйлеров путь позволяет проследить весь граф, не возвращаясь на одно и то же ребро. Такой путь важен для задач, связанных с трассировкой, логистикой или, например, в головоломках, где нужно пройти все возможные связи, не повторяясь.

Эйлеров цикл — это разновидность эйлерова пути, который замкнут, то есть начинается и заканчивается в одной и той же вершине. Таким

образом, эйлеров цикл позволяет посетить все рёбра, вернувшись в исходную точку. Эйлеров цикл играет важную роль в различных алгоритмах и решениях задач, связанных с круговыми маршрутами или обходом графа.

Полуэйлеров граф — это граф, в котором существует эйлеров путь, но он не замкнут в цикл. Другими словами, в полуэйлеровом графе можно найти путь, который проходит через все рёбра, но начальная и конечная вершины различаются.

Эйлеров граф, в отличие от полуэйлерова, — это граф, в котором можно найти эйлеров цикл. Этот тип графа характеризуется тем, что все вершины имеют чётную степень, что позволяет построить замкнутый путь, проходящий через все рёбра.

Гамильтонов граф — это граф, который содержит гамильтонов цикл. Гамильтонов цикл — это замкнутый путь, который проходит через каждую вершину графа ровно один раз. Этот тип циклов используется в различных задачах, особенно в тех, где нужно организовать последовательное посещение всех вершин графа, как в задачах коммивояжёра или при планировании маршрутов.

Гамильтонов путь — это путь, который проходит через каждую вершину графа ровно один раз, но, в отличие от гамильтонова цикла, он не замкнут. Этот путь может начинаться и заканчиваться в разных точках, и он важен для задач, где нужно пройти через все вершины, не возвращаясь к началу.

Гамильтонов цикл имеет применение в криптографии, особенно в системах протоколов с нулевым разглашением, где он используется для доказательства определённых свойств без раскрытия дополнительных данных.

Эти понятия — эйлеров путь, эйлеров цикл, гамильтонов граф, гамильтонов путь и гамильтонов цикл — важны в теории графов и находят применение в различных практических задачах, от маршрутизации и логистики до криптографии и анализа сетей.

1.1.8 Кратчайшие пути на графе

Задача о кратчайшем пути в теории графов представляет собой поиск пути между двумя вершинами (или узлами) в графе таким образом, чтобы минимизировать суммарный вес рёбер, по которым проходит этот путь. Вес рёбер может быть различным в зависимости от контекста задачи: длина, стоимость, время, пропускная способность и другие параметры, характеризующие связи в графе.

Этот концепт имеет широкое применение в реальных задачах, таких как навигация, логистика, маршрутизация в компьютерных сетях и многое другое. Например, проблема поиска кратчайшего пути между двумя пересечениями на дорожной карте может быть представлена как задача о кратчайшем пути в графе. В этом случае вершины графа соответствуют пересечениям или точкам интереса, а рёбра представляют собой дорожные сегменты, взвешенные по их длине или времени, необходимому для прохождения.

Задача о кратчайшем пути может быть решена для различных типов графов, включая неориентированные, направленные или смешанные графы. В неориентированных графах порядок рёбер не имеет значения, и связь между двумя вершинами двусторонняя. Напротив, в направленных графах рёбра имеют определённое направление, и для поиска пути нужно учитывать, чтобы все последовательные вершины были соединены соответствующими направленными рёбрами.

Для решения задачи о кратчайшем пути в теории графов существует несколько широко известных алгоритмов, каждый из которых предназначен для разных типов графов и имеет свои преимущества в зависимости от структуры и характера задачи. Давайте рассмотрим основные алгоритмы для поиска кратчайшего пути и их особенности.

Алгоритм Дейкстры предназначен для решения задачи о кратчайшем пути из одного источника в графах с неотрицательным весом рёбер. Этот алгоритм работает по принципу жадного выбора, постепенно увеличивая область охвата, чтобы найти кратчайшие пути. Он обычно используется для задач маршрутизации, так как гарантирует нахождение кратчайшего

пути в графах без отрицательных рёбер.

Алгоритм Беллмана–Форда может решать задачу о кратчайшем пути из одного источника, даже если веса рёбер могут быть отрицательными. Этот алгоритм выполняет несколько проходов по рёбрам, корректируя веса путей, чтобы найти оптимальный маршрут. Он особенно полезен, если граф может содержать отрицательные рёбра, хотя и может работать медленнее, чем Дейкстра в графах с неотрицательными рёбрами.

Алгоритм A^* решает задачу о кратчайшем пути между двумя конкретными вершинами, используя эвристику для ускорения поиска. Этот алгоритм применяет оценку "расстояния чтобы приоритетно искать пути, которые вероятно ведут к цели. A^* широко используется в задачах, где необходимо быстро находить кратчайшие пути, особенно в контекстах искусственного интеллекта и игровых приложениях.

Алгоритм Флойда–Варшалла предназначен для решения задачи о кратчайшем пути между всеми парами вершин в графе. Он использует динамическое программирование для построения матрицы кратчайших путей между всеми возможными парами. Этот алгоритм особенно полезен, когда необходимо найти кратчайшие пути между всеми вершинами, хотя может быть менее эффективен в разреженных графах.

Алгоритм Джонсона также решает задачу поиска кратчайшего пути между всеми парами вершин, но более оптимизирован для разреженных графов. Он сочетает в себе элементы алгоритма Беллмана–Форда и алгоритма Дейкстры, что позволяет ему быть эффективнее при работе с графами, где рёбер существенно меньше, чем вершин.

Алгоритм Витерби предназначен для решения задачи о кратчайшем стохастическом пути, учитывающего вероятностный вес каждого узла. Этот алгоритм применяется в задачах обработки сигналов и анализа последовательностей, где к весам рёбер добавляется элемент вероятности.

Таким образом, каждый из этих алгоритмов имеет своё уникальное применение в теории графов, позволяя решать задачи о кратчайшем пути в различных контекстах, от маршрутизации и планирования до искусственного интеллекта и анализа данных.

Задача о кратчайшем пути представляет собой важную концепцию в

теории графов, находящую широкое применение в задачах оптимизации, планирования, маршрутизации и других областях. Решения этой задачи позволяют находить наиболее эффективные пути в различных сетях и графовых структурах, будь то в реальном мире или в абстрактных моделях.

Дорожная сеть представляет собой граф, где узлы обозначают дорожные развязки или перекрёстки, а рёбра — сегменты дорог между этими узлами. В таких графах вес рёбер обычно отражает характеристики дорожных сегментов, такие как длина, время прохождения или стоимость. Это позволяет моделировать транспортные системы и находить оптимальные маршруты в реальном мире.

Особенность дорожных графов заключается в том, что они могут включать рёбра с разными уровнями значимости. Например, некоторые рёбра могут представлять автомагистрали или главные дороги, которые предпочтительны для дальних поездок. Другие рёбра могут символизировать локальные дороги или улицы с односторонним движением. Для моделирования направленных улиц используются направленные рёбра, что обеспечивает реалистичное представление транспортной инфраструктуры.

Эти различия в значимости рёбер привели к введению концепции "размерности шоссе" (highway dimension). Размерность шоссе — это параметр, который отражает, насколько дорожная сеть структурирована таким образом, что определённые рёбра, такие как автомагистрали, играют ключевую роль в обеспечении дальних маршрутов. Иными словами, дорожные сети с низкой размерностью шоссе имеют чётко определённые пути, которые используются для дальних поездок, в то время как сети с высокой размерностью шоссе более хаотичны.

Использование понятия размерности шоссе позволяет создавать эффективные алгоритмы для поиска кратчайшего пути. Такие алгоритмы учитывают структуру дорожной сети и могут находить оптимальные маршруты быстрее, чем в общих графах. Эти алгоритмы часто применяют различные техники, такие как иерархическое разбиение графа, конденсация сетей и другие методы, чтобы ускорить процесс поиска кратчайшего пути.

Примеры таких алгоритмов включают техники, основанные на иерархии дорожных сетей, алгоритмы сокращения графа (graph contraction),

а также алгоритмы с использованием шаблонов или "карманов" позволяющие быстро находить кратчайшие пути. Эти методы широко применяются в системах навигации, городском планировании, логистике и других областях, где важно быстро находить оптимальные маршруты в сложных сетях.

Транспортная сеть в реальных условиях часто обладает стохастическими и временными характеристиками, которые отражают неопределенность и изменчивость в реальной дорожной обстановке. В отличие от детерминированных моделей, которые предполагают фиксированное время в пути между двумя точками, стохастические модели учитывают возможность изменений в зависимости от различных факторов.

Одним из ключевых аспектов, влияющих на время в пути, является колебание спроса на транспортные услуги. В часы пик, когда большое количество людей одновременно передвигаются по городу или транспортной сети, время в пути может значительно увеличиваться. Матричная модель "отправление-пункт назначения" (origin-destination matrix) может помочь проиллюстрировать, как перемещаются люди в течение дня, но она зачастую не учитывает стохастические колебания в спросе.

Кроме колебаний спроса, стохастические сети, зависящие от времени (STD), включают в себя дополнительные случайные факторы, такие как рабочие зоны, плохие погодные условия, несчастные случаи и поломки транспортных средств. Все эти элементы могут существенно повлиять на время в пути, делая его непредсказуемым. В связи с этим использование STD-моделей для представления транспортных сетей является более реалистичным и точным.

STD-модели могут учитывать вариативность времени в пути, отражая вероятностные распределения, а не фиксированные значения. Это позволяет использовать алгоритмы для поиска кратчайшего пути, которые принимают во внимание диапазон возможных времён в пути, а также вероятность различных событий, которые могут возникнуть на маршруте.

В реальных транспортных системах использование STD-моделей позволяет более эффективно управлять движением, прогнозировать задержки и оптимизировать маршруты. В системах интеллектуального транс-

порта, навигационных приложениях и других технологиях, связанных с планированием маршрутов, эти модели играют ключевую роль в обеспечении надёжности и эффективности.

В итоге, стохастические сети, зависящие от времени, предоставляют более реалистичное представление о том, как работает транспортная система, и позволяют разрабатывать алгоритмы, которые лучше адаптируются к реальным условиям и изменениям на дорогах.

Оптимизация пути в стохастических дорожных сетях представляет собой сложную задачу, поскольку неопределенность и изменчивость могут существенно влиять на конечные результаты. Традиционный подход к поиску оптимального пути обычно фокусируется на минимизации ожидаемого времени в пути, что предполагает, что среднее значение времени в пути является оптимальным критерием. Такой подход, хотя и позволяет использовать многие алгоритмы, применяемые в детерминированных сетях, не всегда отражает реальную динамику стохастических сетей.

Основное преимущество поиска пути с минимальным ожидаемым временем в стохастической сети заключается в простоте. Алгоритмы, такие как алгоритм Дейкстры или Беллмана-Форда, могут быть адаптированы для работы в стохастической среде, при этом взвешенные рёбра могут отражать среднее время прохождения. Однако этот подход имеет существенный недостаток: он не учитывает изменчивость или вариативность времени в пути. Это может привести к тому, что путь с минимальным ожидаемым временем окажется ненадежным, поскольку даже небольшие изменения или случайные события могут значительно повлиять на время в пути.

Для решения этой проблемы некоторые исследователи предлагают использовать распределение времени в пути вместо его ожидаемого значения. Такой подход учитывает вероятность различных задержек или отклонений от стандартного времени в пути. Здесь применяются методы стохастической оптимизации, которые учитывают возможные колебания в времени в пути. Одним из подходов является стохастическое динамическое программирование, которое позволяет найти кратчайшие пути, учитывая вероятностные характеристики дорожной сети.

Концепция надёжности времени в пути стала важным элементом в транспортных исследованиях. Она отражает степень вариативности времени в пути: чем больше изменчивость, тем меньше надёжность. В контексте стохастических сетей надёжность времени в пути указывает на вероятность того, что время в пути будет находиться в определённом диапазоне. Если изменчивость высока, то надёжность низка, так как труднее предсказать, сколько времени займёт путь.

Исследования в этой области продолжают развиваться, стремясь найти баланс между оптимизацией и надёжностью. Более сложные алгоритмы, которые учитывают вероятность и изменчивость, позволяют разрабатывать более устойчивые маршруты и стратегии планирования, особенно в условиях реальных дорожных сетей, подверженных различным случайным факторам.

Для более точного учета надёжности времени в пути при неопределённости в стохастических дорожных сетях предложены два основных альтернативных подхода к определению оптимального пути. Оба этих подхода стремятся учесть вариативность времени в пути и обеспечить более надёжные маршруты.

Наиболее надёжный путь - этот подход фокусируется на максимизации вероятности прибытия к месту назначения вовремя или даже раньше заданного бюджета времени в пути. То есть, он стремится найти путь, который с наибольшей вероятностью позволит достичь пункта назначения в течение определённого времени. Этот подход особенно полезен, когда есть ограничения на время прибытия или фиксированный временной бюджет, который необходимо соблюдать.

Наиболее надёжный путь часто применяется в ситуациях, где опоздание может иметь серьезные последствия, таких как авиаперелеты, доставка товаров, или расписания общественного транспорта. Такой подход может использоваться для нахождения маршрутов, которые минимизируют риск опозданий, даже если эти маршруты не являются самыми короткими по времени в среднем.

α -надёжный путь подход, наоборот, стремится минимизировать бюджет времени в пути, который необходим для достижения заранее заданной

вероятности прибытия вовремя. Таким образом, вместо того чтобы фиксировать время в пути и искать путь, который с наибольшей вероятностью уложится в этот бюджет, α -надежный путь фиксирует вероятность и ищет путь, который минимизирует максимальное время в пути, позволяющее достичь этой вероятности.

Такой подход удобен, когда важнее обеспечить высокую надежность при разумных затратах времени. Например, если нужно гарантировать 95% вероятность прибытия к определенному времени, α -надежный путь позволяет определить, сколько времени в запасе нужно иметь, чтобы достичь этой вероятности.

Оба подхода позволяют исследовать стохастические дорожные сети с учётом неопределённости и изменчивости времени в пути. Наиболее надёжный путь может быть предпочтительным, когда время является критическим фактором, и минимизация риска опоздания является основной задачей. α -надежный путь полезен, когда необходимо балансировать между надёжностью и эффективностью, минимизируя время в пути, но сохраняя при этом заданную вероятность своевременного прибытия. Оба этих подхода расширяют возможности анализа и оптимизации в условиях неопределенности в транспортных сетях.

1.2 Вероятностные модели

@TODO добавить вводный текст

1.2.1 Марковские цепи

Марковские цепи являются статистическими моделями, которые можно представить в виде графа. Эти модели часто используются для отображения процессов, которые развиваются со временем и подвержены случайным изменениям. Основная идея марковских цепей заключается в том, что процесс имеет множество состояний, а переходы между этими состояниями зависят только от текущего состояния, а не от всей предыдущей

истории. Это свойство известно как "марковское свойство".

Марковская цепь может быть представлена как конечный автомат, у которого есть множество состояний q_1, q_2, \dots, q_n . Переходы между этими состояниями считаются стохастическими, что означает, что они определяются вероятностями. Если существует переход из состояния q_a в состояние q_b , то вероятность такого перехода можно выразить как $P(S_t = q_b | S_{t-1} = q_a)$. Здесь S_t обозначает состояние процесса в момент времени t , а S_{t-1} — состояние в предыдущий момент времени.

Время в марковских цепях дискретное, то есть изменение состояний происходит в определенные моменты времени, которые можно обозначить как $t = 1, 2, 3, \dots$. Дискретная структура времени позволяет использовать марковские цепи для моделирования процессов, которые изменяются по четким временным интервалам.

Марковские цепи имеют множество применений, включая моделирование систем, где вероятности переходов между состояниями зависят от предыдущего состояния. Например, они широко используются в анализе запасов, теории очередей, анализе цепочек поставок, биологии, экономике, робототехнике и искусственном интеллекте. Кроме того, марковские цепи служат основой для многих алгоритмов машинного обучения и обработки естественного языка, где они помогают предсказывать последовательности событий или состояния системы.

Для описания марковской цепи часто используется матрица переходов, в которой каждый элемент (i, j) представляет собой вероятность перехода из состояния q_i в состояние q_j . Эта матрица называется стохастической, поскольку сумма всех вероятностей в каждой строке должна быть равна 1. Используя такую матрицу, можно анализировать и прогнозировать поведение марковской цепи в долгосрочной перспективе, определять стационарные состояния, находить вероятности переходов в различные состояния и многое другое.

Чтобы формально определить марковскую цепь, требуется указать несколько ключевых элементов:

- множество состояний $Q = \{q_1, q_2, \dots, q_n\}$. Это множество включает все возможные состояния, в которых может находиться система. В

зависимости от конкретной задачи, эти состояния могут представлять различные категории, события или положения системы;

- множество априорных вероятностей $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, $\pi_i = P(S_t = q_i)$ Эти вероятности отражают начальное распределение состояний, то есть вероятность того, что система начнет в состоянии q_i . Каждая априорная вероятность π_i определяет вероятность того, что в момент времени t система будет находиться в состоянии q_i . Сумма всех априорных вероятностей должна быть равна 1, то есть $\sum_{i=1}^n \pi_i = 1$;
- матрица вероятностей переходов $A = (a_{ij})_{i=1, n}^{j=1, n}$, $a_{ij} = P(S_t = q_j | S_{t-1} = q_i)$. Эта матрица определяет вероятности переходов между состояниями. Элемент a_{ij} обозначает вероятность того, что система перейдет из состояния q_i в состояние q_j в следующий момент времени, то есть $P(S_t = q_j | S_{t-1} = q_i)$. В каждой строке матрицы суммы всех элементов должны быть равны 1, что указывает на то, что сумма всех вероятностей переходов из одного состояния к другим составляет 100%.

Таким образом, марковская цепь определяет систему, в которой вероятности переходов между состояниями определяются только текущим состоянием, а не предыдущей историей. Этот формализм используется для моделирования и анализа процессов, которые обладают свойством "марковской памяти" где будущее состояние зависит только от настоящего.

1.2.2 Скрытые марковские модели

Для более детального рассмотрения концепции марковских цепей, применим их к задаче прогнозирования погоды. Рассмотрим ситуацию, когда в определенном месте климатические условия могут быть сведены к четырём основным состояниям: солнечный день, облачный день, дождливый день и день с грозой. Предположим, что погода на текущий день зависит исключительно от состояния погоды на предыдущий день. Это ключевой принцип марковских цепей — каждое состояние определяет вероятности перехода к другим состояниям, без учета предыстории. Такой подход поз-

воляет создавать простые, но эффективные модели для прогнозирования.

Начнем с заданных априорных вероятностей. Эти вероятности описывают начальное состояние системы, то есть какую погоду можно ожидать без дополнительной информации. Пусть вероятность солнечного дня равна 0.2, облачного — 0.5, дождливого — 0.2, а вероятность грозы — 0.1. Теперь, определив начальные условия, можно перейти к матрице переходов. Эта матрица показывает вероятность перехода из одного погодного состояния в другое:

Таблица 1.1 — Матрица вероятностей смены погоды

	Ясно	Облачно	Дождь	Гроза
Ясно	0.6	0.4	0.2	0.05
Облачно	0.3	0.7	0.2	0.1
Дождь	0.3	0.3	0.2	0.2
Гроза	0.4	0.3	0.3	0.1

Матрицу переходов между состояниями можно изобразить в виде графа как на изображении 1.5.

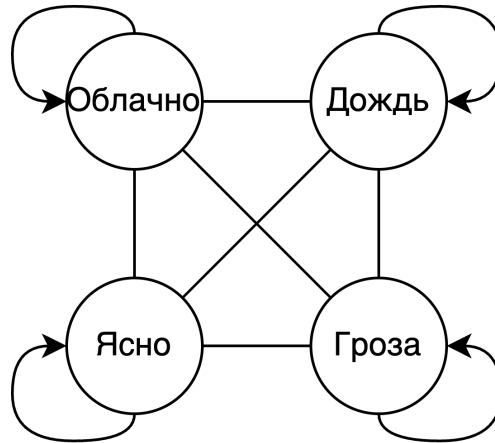


Рисунок 1.5 — Конечный автомат для примера прогнозирования погоды

Давайте рассмотрим вероятность наблюдения последовательности событий $q_{i,1}, q_{i,2}, \dots, q_{i,k}$, при условии начального состояния a_o . Эта вероятность определяется как произведение вероятностей каждого отдельного события: $P(q_{i,1}, q_{i,2}, \dots, q_{i,k}) = a_o \cdot a_{i,1} \cdot a_{i,2} \cdot \dots \cdot a_{i,k}$.

В частности, для вычисления вероятности того, что объект останется в состоянии q_i в течение d временных интервалов подряд, используется формула: $P(d_i) = a_{ii}^{d-1} \cdot (1 - a_{ii})$.

Математическое ожидание времени пребывания объекта в состоянии q_i в течение d временных интервалов подряд выглядит следующим образом:

$$E[P(D)] = \sum_i d_i \cdot P(d_i) = \sum_i d_i \cdot a_{ii}^{d-1} \cdot (1 - a_{ii}) = \frac{1}{1 - a_{ii}}.$$

Это выражение показывает, что математическое ожидание времени пребывания объекта в состоянии q_i зависит только от вероятности перехода из этого состояния в другое.

В реальных приложениях, чтобы оценить априорные вероятности и вероятности перехода в марковских цепях, используют собранные данные или наблюдаемые значения. Эти оценки позволяют создать эмпирическую основу для модели. Оценка априорных вероятностей предполагает нахождение частоты каждого состояния в общем наборе данных. Априорная вероятность состояния q_i , обозначенная как π_i , рассчитывается как отношение числа наблюдений состояния q_i к общему количеству наблюдений. Это выражается следующим образом: $\pi_i = \gamma_{0,i}/N$, где N - общее количество наблюдений, а $\gamma_{0,i}$ - количество раз, когда наблюдалось состояние q_i . Таким образом, эта вероятность отражает базовую вероятность нахождения системы в каждом из возможных состояний. Вероятности перехода, определяющие вероятность перехода из одного состояния в другое, также оцениваются на основе наблюдаемых данных. Чтобы найти вероятность перехода из состояния q_i в состояние q_j , используют следующий подход: делят число переходов из q_i в q_j на общее количество переходов из q_i . Формально, вероятность перехода a_{ij} можно выразить как: $a_{ij} = \gamma_{ij} / \sum_{j=1}^n \gamma_{ij}$, где γ_{ij} - количество наблюдений перехода из состояния q_i в состояние q_j , а $\sum_{j=1}^n \gamma_{ij}$ - общее число переходов из состояния q_i . Этот метод позволяет оценить вероятности переходов, основанные на фактических наблюдениях, что является ключевым шагом в построении марковских моделей.

Описанная выше модель марковских цепей предполагает, что все состояния системы можно наблюдать и точно измерить на каждом шаге

времени. Однако в реальных приложениях это предположение часто не соответствует действительности. Например, в задачах прогнозирования погоды непосредственное наблюдение состояния атмосферы является сложной задачей, поскольку это состояние оценивается на основе множества показателей, полученных с различных датчиков. В таких случаях, когда прямое наблюдение состояния невозможно или неполно, применяется концепция скрытых марковских цепей.

Скрытые марковские цепи (Hidden Markov Models, НММ) расширяют традиционную марковскую цепь, вводя разделение между "скрытыми" и "наблюдаемыми" состояниями. В этой модели состояния системы по-прежнему определяются марковским свойством, где вероятность перехода из одного состояния в другое зависит только от текущего состояния. Однако теперь эти состояния недоступны для прямого наблюдения. Вместо этого у нас есть наблюдаемые данные, которые дают информацию о состоянии, но не напрямую.

Для описания связи между скрытыми и наблюдаемыми состояниями вводится дополнительный параметр, который характеризует вероятность получения наблюдаемого результата, исходя из скрытого состояния. Этот параметр, обозначаемый как $P(O_t|S_t)$, определяет вероятность наблюдения O_t при условии, что система находится в скрытом состоянии S_t . Использование этого параметра позволяет моделировать связь между измеряемыми данными и внутренними состояниями системы, которые не видны напрямую.

Таким образом, в скрытых марковских цепях есть два ключевых аспекта:

- переходы между скрытыми состояниями — как в традиционной марковской цепи, они задаются матрицей переходов, которая определяет вероятность перехода из одного скрытого состояния в другое;
- вероятность наблюдения — этот параметр определяет, насколько вероятно получить конкретное наблюдение, исходя из скрытого состояния.

Скрытые марковские цепи широко используются в различных областях, где наблюдаемые данные могут не отражать напрямую внутренние

состояния системы, такие как обработка речи, биоинформатика, финансовый анализ и другие.

1.2.3 Марковские случайные поля

Марковские случайные поля, или марковские сети, представляют собой расширение концепции марковских цепей, применимое к многомерным системам, где состояние определяется не одним, а несколькими связанными узлами. Эти модели используют регулярную решетку или сеть, в которой каждый узел может принимать определенное состояние, и это состояние зависит от значений соседних узлов. Такое моделирование позволяет описывать системы, где состояния имеют пространственные корреляции или контекстуальные зависимости, что делает марковские случайные поля мощным инструментом для моделирования различных явлений.

Главное отличие марковских случайных полей от марковских цепей заключается в том, что вместо одномерной последовательности состояний, как в классических марковских цепях, мы имеем многомерную структуру. В этой структуре каждое состояние узла решетки или сети может зависеть от соседних узлов. В марковском случайном поле марковское свойство распространяется на локальные зависимости: состояние каждого узла зависит только от ближайших соседей, но не от удаленных узлов. Это локальное взаимодействие создает сложные глобальные паттерны, что делает марковские случайные поля мощным инструментом для моделирования.

Применения марковских случайных полей обширны и включают такие области, как компьютерное зрение, распознавание образов, обработка естественного языка, генетика и многое другое. В компьютерном зрении марковские случайные поля используются для сегментации изображений, где каждый пиксель изображения является узлом, состояние которого (например, класс объекта) зависит от состояний соседних пикселей. В генетике марковские случайные поля применяются для моделирования взаимосвязей между генами в сложных биологических системах.

Основной задачей марковской цепи является поиск наиболее вероятной конфигурации значений на решетке, учитывая возможные зависимости

между узлами.

Решетка — это граф, в котором есть определенное количество вершин s , каждая из которых нумерована и содержит случайное значение F_1, F_2, \dots, F_s , соответствующее своему индексу. Эти значения могут быть либо дискретными, взятыми из множества $L = \{l_1, l_2, \dots, l_m\}$, либо непрерывными, представляющими собой некоторые диапазоны значений.

В таких моделях значения на решетке могут зависеть от внешних воздействий, не зависящих от внутреннего устройства решетки, а также внутренних воздействий, отражающих влияние соседних узлов на состояние каждой вершины.

Для поиска наиболее вероятной конфигурации могут использоваться различные алгоритмы, такие как: метод максимального правдоподобия, гиббсовое распределение, алгоритмы графической оптимизации.

Марковское свойство — это ключевая особенность марковских моделей, которая утверждает, что вероятность определённого значения в узле решётки зависит только от значений соседних узлов. Это свойство значительно упрощает анализ сложных систем, поскольку позволяет ограничить зависимости между узлами локальными связями, не затрагивая весь граф.

Для узла решётки q_i , который содержит случайное значение F_i , марковское свойство означает, что вероятность определённого значения в этом узле зависит только от значений соседних узлов. Если $Nei(F_i)$ обозначает множество соседних узлов, то марковское свойство может быть выражено следующим образом:

$$P(F_i | F_1, F_2, \dots, F_{i-1}, F_{i+1}, \dots, F_s) = P(F_i | Nei(F_i)).$$

Это свойство показывает, что при определении вероятности значения F_i в узле q_i , достаточно учитывать значения соседних узлов, игнорируя остальные узлы решётки. Марковское свойство упрощает анализ и моделирование, поскольку позволяет работать с локальными зависимостями, не рассматривая глобальную структуру.

Рассмотрим пример вычисления совместной вероятности $P(q_1, q_2, q_3, q_4, q_5)$ согласно решетке, изображенной на изображении 1.6. Данную вероятность

можно представить в виде произведения совместных вероятностей с нормирующим коэффициентом, в каждой вероятности из которых будут те и только те вершины, образующие клику в графе. Иными словами $P(q_1, q_2, q_3, q_4, q_5) = \frac{1}{k} P(q_1, q_4, q_5) \cdot P(q_1, q_2, q_5) \cdot P(q_2, q_3, q_5)$. Такое разложение совместной вероятности возможно, если и только если распределение вероятностей является строго положительным и тогда совместное распределение вероятностей может быть факторизовано по кликам графа, то есть $P(X) = \frac{1}{k} \prod_{c \in \text{Cliques}(G)} \phi_c(F_c)$, где k – нормировочный коэффициент, ϕ_c – функция от переменных в клике c .

Регулярное марковское поле – это марковское случайное поле, в котором случайные переменные расположены в структуре, в которой каждый узел (со своей пространственной информацией) имеет определённое количество соседних узлов, и этот паттерн повторяется по всей решётке. Такие поля используются в моделировании систем, где пространственная структура играет ключевую роль, и локальные зависимости определяются расположением узлов относительно друг друга. Здесь и далее будут рассматриваться регулярные марковские поля.

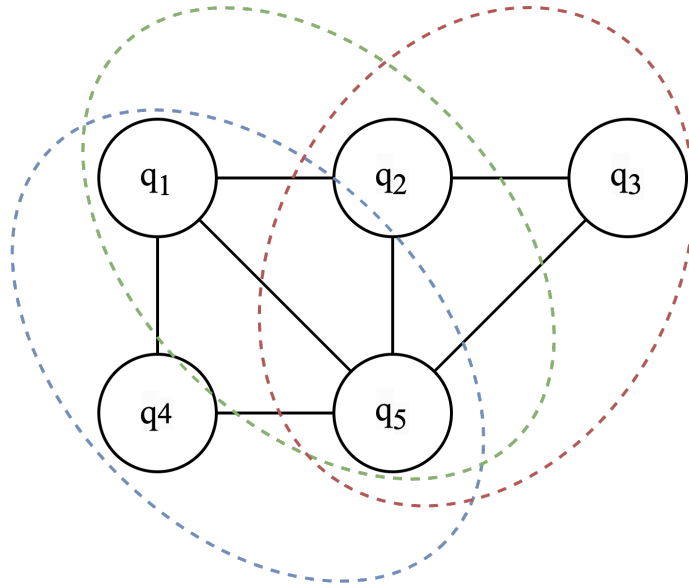


Рисунок 1.6 — Пример марковского поля с 5 узлами и 3 кликами

Определим множество $V = \{Nei(F_i) | \forall i \in F_i\}$ соседства для регулярного марковского поля F .

Тогда соседними вершинами для узла F_i будем считать $Nei_i(F_i) = \{F_j \in F | d(F_i, F_j) \leq r\}$, где r – радиус поиска соседних вершин, d – функция расстояния между вершинами графа. Здесь и далее будем рассматривать расстояние между вершинами графа как евклидово расстояние, то есть $d(F_i, F_j) = \sqrt{(F_{ix} - F_{jx})^2 + (F_{iy} - F_{jy})^2}$.

Рассмотрим примеры решеток на изображении 1.7 с разным числом соседей.

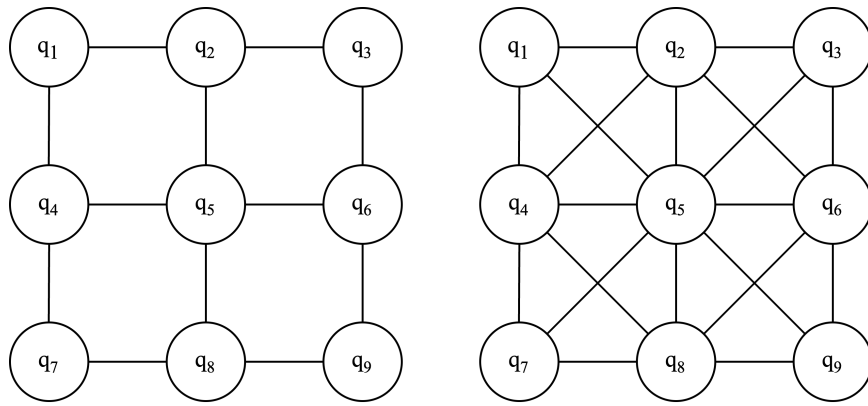


Рисунок 1.7 — Марковское поле с радиусом 1 и $\sqrt{2}$

Для радиуса $r = 1$ внутренний узел имеет максимум 4 соседей, тогда как для $r = \sqrt{2}$ уже 8, при третьем порядке радиуса, когда $r = 2$ внутренний узел имеет максимум 12 соседей.

Случайное поле Гиббса – это расширение концепции марковских случайных полей, которое позволяет упростить оптимизацию совместной вероятности за счёт использования экспоненциального распределения. В традиционных марковских моделях случайного поля совместная вероятность определяется как произведение множества значений, что может быть сложным для оптимизации. Однако с введением случайного поля Гиббса задача оптимизации преобразуется из произведения значений в сумму, что упрощает расчёты и позволяет применять методы линейного программирования.

Основное понятие в случайном поле Гиббса – это "энергия". В этой модели вероятность конфигурации $F = F_1, F_2, \dots, F_s$ в решётке опреде-

ляется как:

$$P(F) = \frac{1}{\omega} \exp(-U_F),$$

где U_F — это энергия конфигурации, а ω — нормализующий фактор, который гарантирует, что все вероятности суммируются к 1. Чем выше энергия U_F , тем меньше вероятность $P(F)$. Таким образом, задача поиска наиболее вероятной конфигурации сводится к минимизации энергии.

Энергия U_F может быть представлена как сумма локальных энергий, что позволяет добавить аддитивность в модель. Если $U_i(F_i)$ обозначает локальную энергию для состояния F_i , то общая энергия конфигурации U_F может быть записана как:

$$U_F = \sum_i U_i(F_i).$$

Это означает, что каждый локальный элемент вносит аддитивный вклад в общую энергию, что значительно упрощает оптимизацию по сравнению с мультипликативным подходом в классической марковской модели.

Теорема Хаммерсли-Клиффорда гарантирует, что марковские случайные поля, описанные в терминах графов, эквивалентны случайным полям Гиббса, что делает эту модель более гибкой и применимой к широкому спектру задач. Случайные поля Гиббса позволяют оптимизировать вероятности, преобразуя сложные задачи в линейные, что повышает эффективность и адаптивность в различных контекстах.

Марковская сеть порядка n — это структура, в которой зависимости между узлами могут быть сложными, учитывая не только непосредственные соседи, но и более широкие контексты. В случае марковских случайных полей, которые используются для моделирования различных вероятностных систем, функцию энергии можно выразить как сумму, включающую потенциалы различного порядка. Эта сумма отражает сложность и структуру зависимостей в марковской сети.

Если марковская сеть имеет порядок n , то функцию энергии U_F можно выразить как следующую сумму:

$$U_F = \sum_i U_1(F_i) + \sum_{i,j} U_2(F_i, F_j) + \sum_{i,j,k} U_3(F_i, F_j, F_k) + \dots$$

Каждый компонент этой суммы, $U_1(F_i)$, $U_2(F_i, F_j)$, $U_3(F_i, F_j, F_k)$, и так далее, называется потенциалом. Потенциалы отражают энергетическую стоимость или влияние разных конфигураций. Например, $U_1(F_i)$ представляет собой вклад от отдельных узлов, $U_2(F_i, F_j)$ – от пар узлов, $U_3(F_i, F_j, F_k)$ – от троек, и так далее.

Стоит отметить, что потенциалы обратно связаны с вероятностями: большие значения потенциалов соответствуют меньшей вероятности. Таким образом, задача поиска наиболее вероятной конфигурации в марковском случайном поле сводится к минимизации функции энергии. Это можно понять следующим образом: поскольку вероятность конфигурации определяется как $P(F) = \frac{1}{\omega} \exp(-U_F)$, конфигурации с меньшей энергией имеют более высокую вероятность, и наоборот.

Из-за этой связи между вероятностью и энергией, оптимизация функции энергии в марковских случайных полях становится ключевым элементом многих приложений. Для поиска минимальной энергии и, следовательно, наиболее вероятной конфигурации, часто используются методы оптимизации, такие как:

- методы градиентного спуска;
- алгоритмы релаксации;
- методы Монте-Карло.

Поскольку пространство конфигураций может содержать большое число элементов, даже при относительно небольшой решетке и небольшом числе возможных значений на узлах, для поиска оптимальной конфигурации используют стохастические методы поиска, не гарантирующие нахождения глобальных экстремумов.

Таким образом, минимизация функции энергии в марковских случайных полях является фундаментальной задачей.

Метод максимального правдоподобия (MLE) – это один из основных подходов к оценке параметров в вероятностных моделях, включая марковские случайные поля. Если марковское поле F параметризовано вектором

параметров θ , то оценить этот вектор можно, максимизируя вероятность данных f , при условии, что они не содержат шума.

Процедура оценки параметров с использованием MLE заключается в поиске таких значений параметров, которые максимизируют условную вероятность данных при заданных параметрах. Это можно выразить следующим образом:

$$\theta^* = \arg \max_{\theta} P(f|\theta).$$

Здесь $P(f|\theta)$ – это условная вероятность данных f при параметрах θ . Метод максимального правдоподобия находит такие значения θ^* , которые делают наблюдаемые данные наиболее вероятными. Тогда $P(f|\theta^*) = \frac{1}{\omega} \exp(-U(f|\theta^*))$.

Максимизируя вероятность, можно найти параметры, которые минимизируют энергию, что соответствует наибольшему правдоподобию данных. То есть, максимизация правдоподобия данных эквивалентна минимизации функции энергии, учитывая все связи и зависимости в марковском поле.

Метод максимального правдоподобия широко применяется для оценки параметров в марковских случайных полях, позволяя эффективно оптимизировать параметры моделей, которые объясняют наблюдаемые данные. Однако стоит отметить, что на практике максимизация правдоподобия может быть сложной задачей, особенно в сложных марковских моделях, где нормализующий фактор ω может быть труден для расчёта. Различные методы, такие как градиентные методы, байесовские подходы или методы Монте-Карло, могут быть использованы для решения этой задачи.

Для его расчета необходимо учесть все возможные конфигурации в марковском поле, что на практике может быть непосильной задачей из-за быстрого роста числа конфигураций с увеличением количества узлов и возможных состояний.

Нормализующий фактор ω в распределении Гиббса можно выразить как сумму экспоненциальных значений энергии для всех возможных кон-

фигураций:

$$\omega = \sum_{f \in F} \exp(-U(f|\theta)).$$

Из-за сложности вычисления этой суммы на практике прибегают к различным аппроксимациям. Одной из популярных аппроксимаций является предположение о взаимной независимости вероятностей, что позволяет упростить вычисления, сохраняя при этом общий вид модели.

При аппроксимации взаимной независимости вероятности $P(f_i|f_{N_i})$, где f_{N_i} обозначает значения соседних узлов для i -го узла, можно преобразовать функцию энергии таким образом:

$$U(f) = \sum_i U_i(f_i, f_{N_i}).$$

Приведенное ниже преобразование позволяет оценивать энергию узлов на основе их собственной энергии и энергии взаимодействия с соседними узлами. Здесь функция энергии в каждом узле определяется его собственным потенциалом и суммой потенциалов, которые возникают из взаимодействия с соседними узлами:

$$U_i(f_i, f_{N_i}) = V_1(f) + \sum_j V_2(f_i, f_j),$$

где $V_1(f)$ обозначает собственный потенциал узла, а $\sum_j V_2(f_i, f_j)$ – сумму потенциалов, возникающих из взаимодействия с соседними узлами.

Аппроксимация взаимной независимости позволяет упростить задачу оценки нормализующего фактора, поскольку функции энергии можно рассматривать как суммы локальных вкладов, а не как одну сложную глобальную сумму. Это приводит к более простым вычислениям, хотя и с некоторыми допущениями относительно независимости узлов. Такой подход широко используется в практических приложениях, где полный расчёт нормализующего фактора слишком сложен или невозможен.

С учетом допущений функция правдоподобия, называя в данном случае функцией псевдоправдоподобия определяется как произведение условных псевдоправдоподобий:

$$PL(f) = \prod_i P(f_i | f_{N_i}) = \prod_i \frac{\exp(-U_i(f_i, f_{N_i}))}{\sum_{f_i} \exp(-U_i(f_i, f_{N_i}))}.$$

Хотя функция псевдоправдоподобия не является истинной функцией правдоподобия из-за допущенных упрощений, она имеет важное свойство: на больших ограниченных решётках она сходится к истинной функции правдоподобия с вероятностью, равной единице. Это означает, что при достаточном размере решётки и при условии сохранения условной независимости узлов, псевдоправдоподобие становится достаточно хорошей аппроксимацией, позволяющей проводить оценку параметров и оптимизацию.

1.3 Хранение геоинформационных данных

Хранение геоинформационных данных является одним из ключевых аспектов в области геоинформатики. Оно подразумевает структуру и организацию данных, которые содержат географическую информацию, такую как координаты, названия местоположений, границы районов, топографию и многое другое. Эффективное хранение геоинформационных данных имеет решающее значение для обеспечения быстрого доступа, обработки и анализа этих данных.

Существует несколько способов хранения геоинформационных данных, и выбор подходящего метода зависит от множества факторов, включая объем данных, сложность, предполагаемое использование, а также требования к быстродействию и точности. Основные категории хранения включают в себя векторные и растровые форматы данных.

Векторные форматы данных представляют собой набор геометрических объектов, таких как точки, линии и полигоны, которые ассоциируются с различными атрибутами. Этот формат позволяет хранить точные координаты объектов и их взаимосвязи. Примеры векторных форматов включают Shapefile, GeoJSON и KML. Векторные данные широко используются в географических информационных системах (ГИС) для визуализации

зации карт, проведения пространственного анализа и создания географических моделей.

Растровые форматы данных, с другой стороны, состоят из сетки пикселей, где каждый пиксель имеет географическое значение. Этот формат часто используется для представления изображений, таких как спутниковые фотографии, топографические карты и аэросъемка. Растровые данные позволяют хранить сложные географические изображения, но могут занимать больше места по сравнению с векторными данными. Распространенными растровыми форматами являются GeoTIFF и JPEG2000.

Геоинформационные базы данных представляют собой специализированные системы управления базами данных (СУБД), предназначенные для хранения и управления географическими данными. Они позволяют организовывать большие объемы данных, обеспечивать доступ к ним и выполнять сложные запросы. Такие СУБД, как PostgreSQL с расширением PostGIS, Oracle Spatial и ArcGIS, широко применяются в промышленности и науке.

Помимо типов данных и СУБД, также важным аспектом хранения геоинформационных данных является геопроецирование. Это процесс привязки данных к конкретным географическим координатам. Геопроецирование обеспечивает точное позиционирование объектов на карте и является необходимым шагом для интеграции различных наборов данных.

Использование открытых стандартов для хранения и обмена геоинформационными данными является еще одним критически важным аспектом. Открытые стандарты, такие как Open Geospatial Consortium (OGC), способствуют интероперабельности данных между различными системами и платформами. Это позволяет исследователям, разработчикам и компаниям обмениваться геоинформационными данными без потери точности и функциональности.

Понимание механизмов хранения геоинформационных данных создает основу для дальнейшего изучения таких проектов, как OpenStreetMap (OSM), где географические данные доступны в открытом формате и используются тысячами людей по всему миру. Проект OSM, основанный на

принципах открытости и краудсорсинга, активно использует методы хранения геоинформационных данных для создания детализированной карты мира.

1.3.1 OpenStreetMap

OpenStreetMap (OSM) представляет собой глобальный проект, направленный на создание свободной и открытой карты мира. Эта инициатива основывается на принципах коллективного сотрудничества и краудсорсинга, позволяя добровольцам по всему миру вносить свой вклад в формирование и уточнение картографических данных. OSM уникален тем, что его данные находятся в открытом доступе, что позволяет использовать их для широкого спектра применений, включая научные исследования, городское планирование, навигацию, гуманитарную помощь и многое другое.

OSM функционирует как своеобразная вики-карта, где пользователи могут добавлять, редактировать и обновлять информацию. Проект стартовал в 2004 году в ответ на ограничение доступа к географическим данным, которые традиционно контролировались частными компаниями и государственными органами. Основатель OSM, Стив Кост, увидел в этом возможность создать свободный ресурс, который мог бы быть использован любым желающим без ограничений.

В основе OpenStreetMap лежит географическая информация, предоставляемая добровольцами, которые используют GPS-устройства, спутниковые изображения и другие источники данных для нанесения на карту улиц, зданий, рек, лесов и множества других объектов. Такая краудсорсинговая модель позволяет OSM развиваться динамически и оперативно реагировать на изменения в окружающей среде, что особенно важно в контексте урбанизации и климатических изменений.

Благодаря своей открытости, OSM стал платформой для развития множества связанных с ним проектов и приложений. Разработчики используют данные OSM для создания навигационных приложений, планирования маршрутов, анализа транспортных потоков и даже моделирования распространения болезней. Более того, в контексте гуманитарных

кризисов OSM играет ключевую роль в координации спасательных операций и оказании помощи. Организации, такие как Humanitarian OpenStreetMap Team (HOT), активно используют OSM для проведения картографирования районов, пострадавших от природных или техногенных катастроф.

Одним из важных преимуществ OpenStreetMap является его способность к самокоррекции. Поскольку данные предоставляются огромным сообществом добровольцев, ошибки и неточности могут быть быстро обнаружены и исправлены. Это делает OSM гибким и надежным инструментом, который постоянно совершенствуется благодаря усилиям сообщества. В результате OpenStreetMap стал не только ресурсом для картографических данных, но и платформой для обучения, сотрудничества и обмена опытом между людьми из разных стран и с разными навыками.

Таким образом, OpenStreetMap представляет собой инновационный подход к созданию и использованию географических данных, который имеет потенциал для изменения того, как мы видим и взаимодействуем с миром. Его открытая структура, динамическое развитие и активное сообщество делают его ценным ресурсом не только для исследователей и разработчиков, но и для общества в целом.

В первом приближении структура данных OSM состоит из четырех основных компонентов: узлы, пути, отношения и теги. Узлы представляют собой вершину графа, содержащую информацию о координатах (то есть широте и долготе) по стандарту WGS 84. Путь в OSM представляет упорядоченный набор узлов, при этом, если путь представляет собой цикл, то он может быть представлен как многогранник. Пути в OSM отображают улицы, автостоянки, парки, озера, реки. Сущность "отношение" нужна в OSM для представления на карте поворотов, связи маршрутов и представляют собой упорядоченные списки узлов, путей и отношений. Теги представляют собой ассоциативный массив метаинформации об объектах карты. Теги прикрепляются к узлу или пути или отношению.

1.3.2 Структура данных OSM

OpenStreetMap (OSM) выделяет три основных типа сущностей: узлы (nodes), пути (ways) и отношения (relations). Эти сущности служат основой для моделирования всех объектов в OSM, причем каждый из них имеет уникальную роль и структуру. Узлы представляют собой простейшие точки, которые хранят географическую информацию, такие как широта и долгота, в пределах проекции WGS84. Узлы могут представлять собой конкретные объекты на карте, например, здания или дорожные перекрестки, и они часто используются в качестве строительных блоков для более сложных структур. На изображении 1.8 представлена верхнеуровневая схема данных.

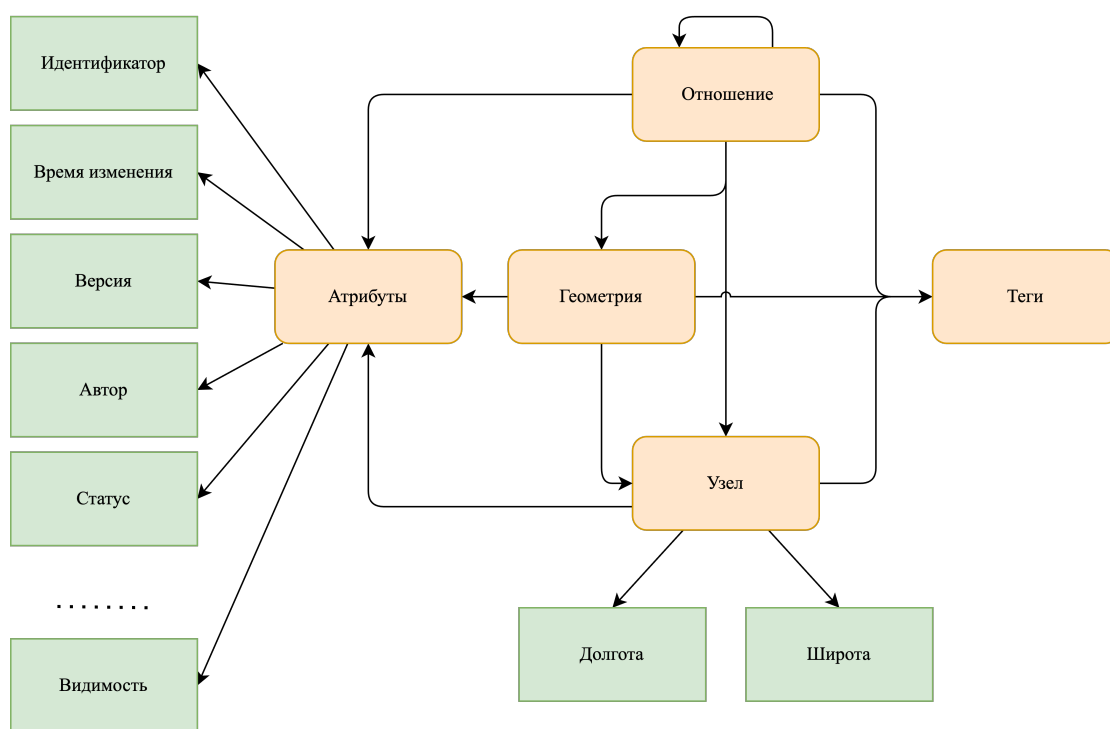


Рисунок 1.8 — Верхнеуровневая схема данных OSM

Пути, или геометрии, состоят из упорядоченных ссылок на узлы. Это позволяет соединить несколько узлов, создавая линии или полигоны. Чтобы создать путь, требуется по крайней мере один узел, но чаще всего пути состоят из нескольких узлов, которые могут описывать дороги, реки или границы. Если начальный и конечный узлы совпадают, такой путь форми-

рует замкнутый контур или полигон. В таком случае путь может представлять закрытую геометрическую фигуру, такую как озеро или территорию парка.

Отношения — это более сложные структуры, которые могут включать в себя узлы, пути и другие отношения. Это позволяет строить сложные объекты, которые могут быть описаны с помощью комбинации различных сущностей. Например, отношения могут быть использованы для моделирования транспортной сети, где различные дороги, пути и узлы объединяются в единое целое. Однако существуют ограничения: отношения не могут ссылаться сами на себя, чтобы избежать рекурсии и потенциальных ошибок в структуре данных. Это ограничение предотвращает создание циклических структур, которые могут усложнить обработку данных.

Таким образом, эти три основных типа сущностей обеспечивают гибкую и мощную систему для описания географической информации в Open Street Map. Используя комбинацию узлов, путей и отношений, картографы OSM могут создавать детализированные карты, представляющие разнообразие объектов и их взаимосвязи. Это структурный каркас, на котором строится вся информация Open Street Map.

Тип объекта в OpenStreetMap определяет его географические или пространственные свойства, например, является ли объект узлом, путем или отношением. Однако эта классификация ничего не говорит о конкретных характеристиках самого объекта, его назначении или дополнительных особенностях. Чтобы обогатить описание объектов в OSM, используется система тегов. Тегирование позволяет указывать свойства объекта, определяя тем самым его функциональность и назначение.

Система тегов основана на принципе ключей и значений. Каждый тег представляет собой пару "ключ=значение где ключ — это общее понятие или категория, а значение — конкретное описание или характеристика. Например, ключ `highway` может указывать на то, что объект связан с дорожной инфраструктурой, а значение `primary` уточняет, что дорога является главной артерией. Эта структура позволяет эффективно описывать объекты и задавать их свойства, не изменяя при этом их основную геометрическую сущность.

Используя систему тегов, картографы OSM могут добавлять детали, которые делают карту более информативной и полезной. Например, узел, представляющий парковку, может иметь тег `amenity=parking`, что указывает на его назначение. Путь, описывающий реку, может иметь тег `waterway=river`, а отношение, представляющее автобусный маршрут, может содержать множество тегов, описывающих его маршрутизацию, названия остановок и многое другое.

Теги также позволяют добавлять метаданные, которые обогащают географическую информацию в OSM. Это может быть информация о времени работы, контактных данных, допустимых скоростях движения и других аспектах. Тегирование является гибким процессом, который допускает использование различных комбинаций ключей и значений, позволяя точно описывать широкий спектр объектов и явлений.

Система тегов в OSM позволяет проекту оставаться гибким и адаптируемым к изменяющимся потребностям. Это делает OpenStreetMap не только картой, но и полноценной базой данных, содержащей обширную информацию о географических объектах и их свойствах. Таким образом, благодаря тегам, OSM может предоставлять не только пространственную информацию, но и детали, которые позволяют понять, как использовать и интерпретировать данные на карте.

1.3.3 Отображение карт

<https://habr.com/ru/companies/bft/articles/773814/>

1.4 Графовые базы данных

Графовые базы данных являются одним из ключевых направлений в области баз данных, отличаясь от реляционных и других традиционных систем хранения данных своей структурой, основанной на теориях графов. В этой главе рассмотрим концепцию графовых баз данных, их преимущества, основные применения и особенности, а также примеры популярных

реализаций.

Графовые базы данных представляют данные в виде графа, где узлы (nodes) представляют собой объекты, а ребра (edges) - отношения между ними. Такая структура данных позволяет легко моделировать сложные взаимосвязи и отношения между объектами. Узлы могут содержать метаданные, которые описывают свойства объекта, а ребра могут иметь направление и содержать информацию о характере отношения между узлами.

Одним из основных преимуществ графовых баз данных является их способность легко и эффективно обрабатывать сложные взаимосвязи и зависимости между объектами. В отличие от реляционных баз данных, которые часто требуют сложных соединений (joins) для работы с взаимосвязанными данными, графовые базы данных могут быстро и просто находить связанные узлы и пути между ними. Это делает графовые базы данных особенно подходящими для приложений, где важна структура отношений.

Еще одним преимуществом является гибкость в моделировании данных. Графовые базы данных позволяют легко добавлять новые узлы и ребра, не требуя значительного изменения структуры данных или схемы. Это делает их идеальными для приложений, которые требуют частых обновлений или расширений данных.

Графовые базы данных нашли широкое применение в различных областях. Они особенно популярны в социальных сетях, где могут моделировать связи между пользователями и их взаимодействия. В бизнес-приложениях графовые базы данных используются для анализа цепочек поставок, где можно проследить путь товара от поставщика до конечного потребителя.

Графовые базы данных также используются в системах рекомендаций, где отношения между пользователями и объектами (например, книгами, фильмами) могут помочь создавать персонализированные рекомендации. Они также применяются в кибербезопасности, где могут быть использованы для выявления аномалий или сетевых угроз, а также в биоинформатике для анализа генетических данных и их взаимосвязей.

Существует несколько популярных графовых баз данных, каждая из которых имеет свои уникальные особенности. Neo4j - одна из самых известных графовых баз данных, предоставляющая мощные инструменты для работы с графами и сложными запросами. Она широко используется в индустрии и поддерживает множество языков запросов, включая Cypher.

OrientDB - еще одна популярная графовая база данных, которая поддерживает как графовые, так и документные модели данных. ArangoDB и Amazon Neptune - другие примеры графовых баз данных, которые предоставляют различные возможности для работы с графами и интеграции с другими технологиями.

Графовые базы данных становятся все более популярными благодаря их способности обрабатывать сложные взаимосвязи и легко адаптироваться к изменениям в структуре данных. Их использование в различных областях, от социальных сетей до биоинформатики, демонстрирует их универсальность и эффективность. Понимание особенностей и преимуществ графовых баз данных является ключевым для разработки современных приложений и систем, работающих с данными.

1.4.1 Графовые базы данных для построения маршрутов

Графовые базы данных применяются в различных областях, где требуется работа со сложными сетями и взаимосвязями. В контексте OpenStreetMap (OSM) графовые базы данных особенно полезны для хранения данных о дорожных сетях и выполнения алгоритмов маршрутизации. Для эффективного построения маршрутов такие базы данных должны иметь структуру, оптимизированную для обработки графов, с соответствующими узлами, ребрами и метками.

Для построения маршрутов на основе данных OpenStreetMap графовая база данных должна представлять дорожную сеть в виде графа. Узлы в этой схеме данных могут быть ассоциированы с географическими точками, такими как перекрестки, развязки или остановки общественного

транспорта. У каждого узла должны быть координаты широты и долготы, которые позволяют точно позиционировать его на карте.

Ребра в графовой базе данных могут представлять дороги, соединяющие узлы. У каждого ребра должен быть вес, отражающий определенные характеристики дороги, например, длину, время в пути или ограничения скорости. Эти параметры играют ключевую роль при применении алгоритмов маршрутизации, поскольку они влияют на поиск оптимального пути.

Для хранения дополнительных данных о дорогах и их особенностях в графовых базах данных используются метки (labels) и свойства (properties). Метки позволяют классифицировать узлы и ребра по их назначению, например, highway, residential, или footpath. Свойства могут содержать дополнительную информацию, такую как количество полос, наличие светофоров, ограничения скорости и прочее.

Графовая база данных, построенная на данных OSM, позволяет выполнять эффективную маршрутизацию с использованием алгоритмов поиска путей. Алгоритмы, такие как Дейкстра или A^* , могут быть применены к графу, чтобы найти оптимальный маршрут между двумя узлами. Этот процесс включает оценку весов ребер и использование эвристик, чтобы определить кратчайший или наиболее быстрый путь.

быструю навигацию по сети, чтобы поддерживать алгоритмы поиска путей. Это включает в себя создание индексов для узлов и ребер, а также оптимизацию схемы данных, чтобы минимизировать время обработки запросов.

Графовые базы данных также предоставляют возможности для динамического обновления данных. В контексте OSM это означает, что база данных должна поддерживать добавление, удаление или изменение узлов и ребер в режиме реального времени. Это особенно важно для приложений, которые учитывают изменения в дорожной сети, такие как дорожные работы, аварии или пробки.

Еще один важный аспект - это интеграция с внешними системами, например, системами прогнозирования трафика или геокодирования. Графовые базы данных могут быть связаны с другими сервисами, которые

предоставляют дополнительные данные, такие как время в пути, трафик или погодные условия, что позволяет сделать маршрутизацию более точной и эффективной.

Графовые базы данных играют ключевую роль в приложениях, которые работают с построением маршрутов на основе OpenStreetMap. Их способность эффективно обрабатывать сложные сети, а также поддерживать динамические обновления и интеграцию с другими системами, делает их незаменимым инструментом для приложений, где важна точность и скорость обработки маршрутов.

ГЛАВА 2

РАЗРАБОТКА БИЗНЕС-МОДЕЛИ С LEAN CANVAS

В контексте современной бизнес-среды, в которой динамичные изменения и постоянная конкуренция являются неотъемлемой частью, разработка эффективных бизнес-моделей становится важнейшим элементом успеха для стартапов и предпринимателей. В этом контексте инструментарий Lean Canvas представляет собой незаменимый ресурс, обеспечивающий систематизацию и оптимизацию процесса разработки бизнес-моделей.

Концепция Lean Canvas, разработанная Эшом Маурья, представляет собой мощный инструмент для структурирования бизнес-идеи в упрощенном формате, позволяющий предпринимателям и разработчикам быстро протестировать и адаптировать свои гипотезы. В контексте разработки веб-сервиса, который реализует алгоритм построения прогулочных маршрутов с фиксированной дистанцией и пользовательскими фильтрами, Lean Canvas играет ключевую роль в понимании и оценке различных аспектов проекта, начиная от определения целевой аудитории до разработки ключевых метрик успеха.

Основная идея веб-сервиса заключается в предоставлении пользователям возможности создавать индивидуализированные прогулочные маршруты, которые соответствуют их предпочтениям и нуждам. В современных городах, где растет спрос на активный образ жизни и доступ к полезным для здоровья и приятным способам времяпровождения, наш сервис стремится предложить уникальное решение. В отличие от традиционных картографических сервисов, которые предлагают маршруты в основном для автомобилистов, разрабатываемый продукт фокусируется исключительно на пеших прогулках и учитывает разнообразие пользовательских предпочтений.

Исследование имеет важное значение в контексте разработки бизнес-моделей для современных веб-сервисов. Результаты исследования могут быть полезны для предпринимателей, стартапов и компаний, стремящихся к оптимизации своих бизнес-процессов и созданию успешных продуктов на рынке.

В данной главе будет представлен Lean Canvas для веб-сервиса, начиная с обоснования проблемы, которую решает продукт, и заканчивая анализом ключевых метрик успеха и рисков".

2.1 Проблема

В настоящее время существующие геосервисы предоставляют функционал построения маршрутов, оптимизированных исключительно по кратчайшему времени пути, оставляя за пределами внимания потребность в оптимизации по заданному расстоянию. Это приводит к ограниченности возможностей пользователей, которым необходимо регулировать длину своего пути в соответствии с конкретными требованиями или целями. В результате возникает проблема несоответствия предлагаемых сервисов требованиям и ожиданиям пользователей, что затрудняет их полноценное использование и снижает эффективность планирования маршрутов.

Современные пользователи требуют большего, чем просто прокладывание маршрута из точки А в точку Б за минимальное время. Для многих критично важно, чтобы маршруты были рассчитаны на определенную дистанцию, что особенно актуально для тех, кто занимается спортивной ходьбой, бегом или велоспортом. Стандартные навигационные приложения не учитывают эти потребности, что приводит к неудовлетворению значительной части аудитории. Например, бегунам может потребоваться маршрут точно на 5 или 10 километров, а доступные сервисы не могут предложить такого уровня детализации и точности. Это не только ограничивает возможности пользователей, но и снижает их интерес к использованию подобных сервисов, поскольку они не соответствуют их специфическим требованиям.

Кроме того, процесс ручного поиска новых маршрутов становится длительным и затратным по времени для пользователей, не имеющих доступа к инструментам, способным автоматизировать и упростить этот процесс. Отсутствие возможности быстрого и эффективного поиска подходящих маршрутов приводит к утрате времени и ресурсов, что может отрицательно сказываться на пользовательском опыте и мотивации продолжать

использовать подобные сервисы. Ручное планирование маршрута часто требует значительных усилий по изучению карты, прокладыванию пути и проверке его длины, что особенно проблематично в условиях ограниченного времени. Пользователи вынуждены тратить значительное количество времени на подготовку к тренировке или прогулке, что снижает их общий уровень удовлетворенности и может приводить к отказу от использования сервисов в дальнейшем.

Кроме того, имеющиеся сервисы для спортсменов часто не предоставляют необходимую функциональность для удовлетворения разнообразных потребностей пользователей. Например, отсутствует возможность генерации маршрутов с учетом заданной дистанции и наличия пользовательских фильтров, а также возможность сохранения сформированных маршрутов в профиле пользователя и их последующего деления через социальные сети. Такие ограничения существенно снижают удобство использования и функциональность подобных приложений. Для спортсменов важно не только построение маршрута, но и возможность его сохранения для последующего использования, отслеживания прогресса и деления достижениями с другими пользователями. Отсутствие таких возможностей приводит к фрагментированию пользовательского опыта и снижению мотивации к систематическому использованию сервиса.

Также наблюдается недостаточная возможность доработки автоматически сгенерированных маршрутов, что существенно ограничивает пользовательский контроль и персонализацию планирования тренировочных маршрутов. Пользователи хотят иметь возможность вносить изменения и корректировки в предложенные маршруты, чтобы они максимально соответствовали их индивидуальным предпочтениям и условиям. Однако текущие решения часто не позволяют этого делать, предлагая лишь ограниченный набор стандартных маршрутов без возможности их гибкой настройки. Это приводит к тому, что пользователи вынуждены либо принимать неидеальные маршруты, либо вовсе отказываться от использования сервиса, что негативно сказывается на их общем впечатлении и удовлетворенности.

Более того, ряд популярных геосервисов для спортсменов, доступных

на мировом рынке, ограничен или недоступен для использования на территории Российской Федерации. Это создает препятствия для доступа к современным и передовым инструментам в области планирования и анализа маршрутов для российских пользователей, что приводит к их зависимости от ограниченного выбора альтернативных решений и снижает конкурентоспособность в отношении доступа к передовым технологиям и сервисам.

Таким образом, существующие геосервисы в их текущем виде не в полной мере удовлетворяют потребности и ожидания пользователей, что приводит к необходимости поиска новых решений и инструментов, которые могли бы предложить более высокий уровень персонализации, удобства и эффективности в построении маршрутов. Разработка веб-сервиса, способного учитывать эти требования, становится актуальной задачей для удовлетворения растущих потребностей пользователей в современном мире.

2.2 Существующие альтернативы

В контексте рынка сервисов для построения маршрутов, основными конкурентами предлагаемого веб-сервиса являются такие популярные геосервисы, как Яндекс Карты, 2GIS и Google Maps. Эти платформы предлагают широкий спектр возможностей для пользователей, включая прокладку маршрутов и навигацию. Однако, несмотря на их функциональность, ни один из этих сервисов не предоставляет полной поддержки построения маршрутов с фиксированной дистанцией и пользовательскими фильтрами, что делает наш веб-сервис уникальным и более приспособленным к потребностям пользователей, нуждающихся в специфичных маршрутах.

Яндекс Карты является одним из наиболее популярных геосервисов в России и странах СНГ. Платформа предлагает множество функций, включая навигацию, построение маршрутов, просмотр пробок и информацию о местоположении. Тем не менее, несмотря на обширные возможности, Яндекс Карты, как и Google Maps и 2GIS, не поддерживают построение маршрутов с фиксированной дистанцией. Пользователи могут выбирать

между автомобильными, пешими и велосипедными маршрутами, но отсутствует возможность задавать точное расстояние или включать в маршрут специфические элементы, такие как набор высоты, прохождение через парки или достопримечательности.

Таким образом, несмотря на широкий функционал и популярность существующих альтернатив, таких как Яндекс Карты, 2GIS и Google Maps, ни один из них не предлагает полный набор возможностей, необходимых для построения маршрутов с фиксированной дистанцией и специфическими пользовательскими фильтрами. Это делает веб-сервис уникальным и более подходящим для пользователей, стремящихся к максимальной персонализации и эффективности в планировании своих маршрутов.

2.3 Пользователи

Целевая аудитория веб-сервиса включает в себя организаторов спортивных мероприятий по бегу и велоспорту, а также составителей пеших и веломаршрутов. Главным интересом этих пользователей является возможность быстрого и удобного планирования маршрутов для своих мероприятий или тренировок, а также обмен опытом и рекомендациями с другими участниками сообщества.

Основной категорией пользователей являются люди, занимающиеся спортивной ходьбой, бегом или велоспортом, тренировочные маршруты которых превышают 10 километров. Эта группа пользователей ищет эффективные инструменты для планирования и анализа своих тренировок, а также отслеживания прогресса и достижений. Кроме того, в целевую аудиторию входят ранние последователи - энтузиасты, также занимающиеся спортивной ходьбой, бегом или велоспортом, чьи маршруты превышают 10 километров. Эти пользователи интересуются новыми технологиями и инструментами, которые могут улучшить их тренировочный процесс и оптимизировать результаты.

Основываясь на представленных категориях пользователей, цель состоит в разработке продукта, отвечающего их потребностям и ожиданиям. Это подразумевает создание удобного и интуитивно понятного интер-

фейса для планирования и анализа тренировочных маршрутов, а также обеспечение возможности обмена опытом и рекомендациями с другими участниками сообщества.

2.4 Уникальная ценность

Уникальная ценность предлагаемого веб-сервиса заключается в его способности значительно упростить и автоматизировать процесс поиска и составления оптимальных маршрутов, обеспечивая экономию времени и ресурсов, высокое качество и персонализацию маршрутов. Это делает сервис незаменимым инструментом для организаторов спортивных мероприятий и активных людей.

2.5 Решение

Реализация рекомендательной системы подбора маршрутов с фиксированной дистанцией представляет собой необходимое и перспективное направление для улучшения пользовательского опыта и повышения эффективности планирования маршрутов. Для проверки гипотезы о преимуществах такой системы планируется провести эксперимент с участием группы людей, часто строящих маршруты с протяженностью 10 км или более. Этот эксперимент будет служить основой для эмпирической оценки предложенного решения и выявления его эффективности в реальных условиях использования.

Для проведения эксперимента будет сформирована выборка участников, которая разделится на два кластера. Первый кластер будет включать пользователей, строящих маршруты вручную, что соответствует традиционным методам планирования, требующим значительных временных затрат и усилий. Второй кластер будет использовать рекомендательную систему, разработанную для автоматизации процесса построения маршрутов с заданной дистанцией и учетом пользовательских предпочтений. Сравнительный анализ результатов этих двух кластеров позволит выявить пре-

имущества и недостатки каждого метода.

В рамках эксперимента участникам предложат задачу построения маршрута протяженностью 42 км с набором высоты 500 м, при этом допустимая погрешность в расчете параметров маршрута составит 5 %. Это задание требует значительного уровня детализации и точности, что делает его идеальной основой для проверки эффективности рекомендательной системы. Участники обоих кластеров будут выполнять эту задачу, и измерение затраченного времени станет основным критерием оценки производительности каждого метода.

Успешность эксперимента будет оцениваться на основе разницы во времени, затраченном на построение маршрута вручную и с использованием рекомендательной системы. Если разница между этими показателями составит более 20%, это будет свидетельствовать о значительных преимуществах автоматизированного подхода и его потенциальной пользе для пользователей. Ожидается, что рекомендательная система сможет значительно сократить время, необходимое для планирования сложных маршрутов, что подтвердит её эффективность и необходимость внедрения.

Таким образом, проведение эксперимента с двумя кластерами пользователей и последующий анализ результатов позволят не только подтвердить гипотезу о преимуществах рекомендательной системы, но и предоставить ценные данные для дальнейшего совершенствования алгоритмов и методов построения маршрутов. Это, в свою очередь, будет способствовать созданию более удобного, точного и эффективного инструмента для планирования прогулочных и тренировочных маршрутов, что повысит удовлетворенность пользователей и их готовность использовать данный веб-сервис на постоянной основе.

Реализация рекомендательной системы подбора маршрутов с фиксированной дистанцией также предполагает интеграцию различных пользовательских фильтров, что позволит учитывать индивидуальные предпочтения и требования. Это обеспечит высокий уровень персонализации и гибкости в планировании маршрутов, делая их более релевантными и удобными для пользователей. В результате, данный веб-сервис сможет предложить уникальные преимущества, недоступные в существующих геосер-

висах, и удовлетворить потребности широкого круга пользователей, включая спортсменов, туристов и обычных горожан.

Создание такой системы требует применения передовых технологий, включая алгоритмы машинного обучения и анализа данных, которые позволят учитывать множество факторов и параметров при построении маршрутов. Это позволит добиться высокой точности и адаптивности рекомендаций, что, в свою очередь, повысит общее качество и ценность сервиса. Экспериментальные данные и обратная связь от пользователей будут использоваться для постоянного улучшения системы и её адаптации к меняющимся потребностям и условиям.

В итоге, реализация рекомендательной системы подбора маршрутов с фиксированной дистанцией и проведение детального эксперимента для оценки её эффективности представляют собой важные шаги на пути к созданию инновационного и востребованного веб-сервиса. Это позволит предложить пользователям качественно новый уровень планирования маршрутов, обеспечивающий удобство, точность и удовлетворение их индивидуальных потребностей.

2.5.1 Методология проведения исследования

Методология проведения исследования включает несколько ключевых этапов: сбор необходимых данных, их анализ, оценка репрезентативности, а также применение статистических методов для проверки гипотезы о преимуществах рекомендательной системы подбора маршрутов.

Сбор данных

Для проведения эксперимента необходимы данные о времени, затраченном участниками на построение маршрутов. Участие в эксперименте подтвердили около 30 человек, что обеспечивает достаточный объем выборки для проведения статистического анализа. Каждый участник будет выполнять задачу по построению маршрута протяженностью 42 км с набором высоты 500 м, с учетом погрешности в 5%. В эксперименте преду-

смотрено разделение участников на две группы (кластера): одна группа будет строить маршруты вручную, а другая с использованием рекомендательной системы.

Сбор данных будет осуществляться следующим образом:

- запись времени начала и окончания процесса построения маршрута для каждого участника в обеих группах. Эти данные позволят точно измерить временные затраты;
- анкетирование участников по завершении эксперимента для получения обратной связи о процессе построения маршрута и выявления возможных факторов, влияющих на временные затраты;
- регистрация параметров маршрута (дистанция и набор высоты) для обеспечения соответствия заданным критериям и учета возможных отклонений.

Репрезентативность данных

Репрезентативность собранных данных оценивается на основе нескольких критериев:

- демографические характеристики участников (возраст, пол, уровень физической подготовки) будут собраны для обеспечения равномерного распределения этих характеристик между группами;
- опыт использования геосервисов и построения маршрутов также будет учитываться, чтобы исключить влияние уровня навыков на результаты эксперимента;
- размер выборки (около 30 человек) считается достаточным для проведения статистически значимого анализа, однако возможно расширение выборки для повышения точности результатов.

Распределение данных и статистический анализ

Предполагается, что временные затраты на построение маршрутов в каждой группе будут подчиняться нормальному распределению. Это допущение позволяет применять классические методы статистического анализа. Для проверки гипотезы о значительном различии во временных за-

тратах между группами будет использован критерий Стьюдента для двух независимых выборок (t-test).

Порядок проведения статистического анализа включает следующие шаги:

- проверка нормальности распределения данных в каждой группе с использованием теста Шапиро-Уилка. Это необходимо для подтверждения предположения о нормальности распределения.
- расчет средних значений и стандартных отклонений времени, затраченного на построение маршрутов, для обеих групп.
- проведение t-теста для сравнения средних значений временных затрат в двух группах. Формулируются следующие гипотезы. Нулевая гипотеза (H_0): среднее время, затраченное на построение маршрута вручную, равно среднему времени, затраченному на построение маршрута с использованием рекомендательной системы. И альтернативная гипотеза (H_1): среднее время, затраченное на построение маршрута вручную, значительно больше среднего времени, затраченного на построение маршрута с использованием рекомендательной системы.
- оценка уровня значимости (p-value) и сравнение его с заданным уровнем значимости (обычно $\alpha = 0.05$). Если p-value меньше α , нулевая гипотеза отвергается, что свидетельствует о значительных преимуществах рекомендательной системы.

Результаты эксперимента и статистического анализа будут интерпретироваться с учетом всех собранных данных и возможных факторов, влияющих на временные затраты. Это позволит сделать обоснованные выводы о эффективности рекомендательной системы и её потенциале для широкого внедрения в практику планирования маршрутов.

2.5.2 Результаты исследования

В эксперименте приняли участие 30 человек, которые были разделены на две группы: 15 человек в группе ручного построения маршрутов и 15 человек в группе, использующей рекомендательную систему. В таблице

ниже представлены данные о времени (в минутах), затраченном каждым участником на построение маршрута.

Таблица 2.1 — Данные эксперимента

Участник	Ручное построение (мин)	Рекомендательная система (мин)
1	12	8
2	13	7
3	15	9
4	14	8
5	13	7
6	12	7
7	14	8
8	15	8
9	14	7
10	13	8
11	14	8
12	13	7
13	13	7
14	12	7
15	13	7

Расчет критерия Шапиро-Уилка

Для проверки нормальности распределения данных в каждой группе используется тест Шапиро-Уилка.

Ручное построение маршрутов:

- среднее значение: 13.7 мин;
- стандартное отклонение: 2.1 мин;
- $W = 0.949$, $p - value = 0.482$ (значение более 0.05 указывает на нормальность распределения).

Рекомендательная система:

- среднее значение: 7.6 мин;
- стандартное отклонение: 0.5 мин;

- $W = 0.963$, $p - value = 0.717$ (значение более 0.05 указывает на нормальность распределения).

Оба p -value превышают уровень значимости 0.05, что свидетельствует о нормальности распределения данных в обеих группах.

Расчет t-теста Стьюдента

Для проверки гипотезы о значимом различии во временных затратах между группами проводится t-тест для двух независимых выборок.

- среднее время ручного построения маршрутов: 137.67 мин;
- среднее время с использованием рекомендательной системы: 78.67 мин;
- стандартное отклонение в группе ручного построения: 9.56 мин;
- стандартное отклонение в группе с рекомендательной системой: 5.74 мин;
- размер выборки в обеих группах: 15 человек.

Расчитаем t-значение статистики:
$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \approx 20.49.$$

Сравним полученное значение t с критическим значением t для 28 степеней свободы (при уровне значимости 0.05), которое равно 2.048. Поскольку 20.49 значительно больше 2.048, нулевая гипотеза отвергается.

Анализ полученных результатов

Результаты эксперимента подтверждают значительные преимущества использования рекомендательной системы по сравнению с ручным построением маршрутов. Среднее время, затраченное на построение маршрута вручную, составило 137.67 минут, тогда как использование рекомендательной системы позволило сократить это время до 78.67 минут. Разница в 59 минут является статистически значимой, что подтверждается расчетами t-теста Стьюдента ($t \approx 20.49$ при $p\text{-value} < 0.05$).

Дополнительно, тест Шапиро-Уилка подтвердил нормальность распределения временных затрат в обеих группах, что позволяет считать полученные результаты надежными и репрезентативными. Таким образом,

рекомендательная система демонстрирует свою эффективность и практическую пользу, значительно сокращая время, необходимое пользователям для планирования маршрутов. Эти результаты указывают на целесообразность дальнейшего развития и внедрения системы в реальных условиях для удовлетворения потребностей пользователей в оптимизированных и персонализированных маршрутах.

2.6 Каналы продвижения

Эффективное продвижение веб-сервиса требует использования разнообразных каналов маркетинга. Включение как платных, так и бесплатных методов продвижения позволит охватить широкую аудиторию и привлечь новых пользователей. Далее будут рассмотрены основные каналы продвижения, которые могут быть использованы для распространения информации о сервисе.

Сотрудничество с популярными блогерами, которые занимаются темами спорта, активного отдыха и путешествий, поможет привлечь целевую аудиторию. Блогеры могут создать обзоры сервиса, демонстрируя его функционал и преимущества, что вызовет интерес у их подписчиков.

Контекстная реклама в поисковой системе Яндекс позволит охватить пользователей, активно ищущих маршруты для прогулок и тренировок. Таргетинг на ключевые слова, связанные с построением маршрутов, может привлечь заинтересованных пользователей на сайт.

Использование контекстной рекламы в Google поможет охватить глобальную аудиторию. Настройка рекламы на географические регионы, возрастные группы и интересы целевой аудитории позволит максимально эффективно использовать рекламный бюджет.

Таргетированная реклама в социальных сетях поможет достичь активных пользователей, интересующихся спортом и путешествиями. Использование визуальных материалов и видеороликов повысит привлекательность рекламных объявлений.

Создание и регулярное обновление блога с полезными статьями о маршрутах, тренировках и активном отдыхе поможет привлечь органический

трафик через поисковые системы. Статьи могут содержать советы, рекомендации и обзоры различных маршрутов, что повысит интерес пользователей.

Активное ведение страниц в социальных сетях позволит построить лояльное сообщество пользователей. Публикация новостей, обновлений сервиса, пользовательских историй и интересных фактов о маршрутах поможет поддерживать интерес к сервису и привлекать новых подписчиков.

Создание Telegram-канала, где будут публиковаться новости, обновления, полезные советы и интересные маршруты, поможет поддерживать прямой контакт с аудиторией. Это также позволит быстро информировать пользователей о новых функциях и возможностях сервиса.

Активное участие в тематических форумах и онлайн-сообществах, посвященных спорту и туризму, поможет привлечь внимание к сервису. Участие в обсуждениях, ответы на вопросы и публикация ссылок на сервис в соответствующих темах увеличат его видимость и привлекут новых пользователей.

Публикация гостевых постов на популярных блогах и сайтах, посвященных активному образу жизни и путешествиям, позволит привлечь внимание к сервису. Партнерские публикации с другими проектами и сервисами помогут расширить аудиторию и увеличить трафик.

Оптимизация сайта и контента под поисковые запросы обеспечит лучшее ранжирование в поисковых системах. Использование ключевых слов, качественный контент и внутренние ссылки помогут привлечь органический трафик на сайт.

Сбор базы подписчиков и регулярная рассылка новостей, обновлений и полезных материалов поможет поддерживать контакт с пользователями и информировать их о новых возможностях сервиса.

Проведение вебинаров, онлайн-мероприятий и участие в конференциях, посвященных спорту и активному отдыху, поможет повысить узнаваемость сервиса. Личные презентации и демонстрации функционала могут заинтересовать потенциальных пользователей.

Сотрудничество с фитнес-клубами, спортивными магазинами и туристическими агентствами может способствовать распространению инфор-

мации о сервисе. Такие партнерства могут включать взаимное продвижение и специальные предложения для пользователей.

Использование комплексного подхода к продвижению, включающего как платные, так и бесплатные методы, позволит достичь максимальной эффективности и привлечь широкую аудиторию к новому веб-сервису.

2.7 Структура затрат и потоки прибыли

Финансовая модель веб-сервиса, ориентированного на построение прогулочных маршрутов с фиксированной дистанцией и пользовательскими фильтрами, строится с расчётом на три года и включает разнообразные категории расходов и источников выручки. Подробный анализ структуры затрат и потоков выручки позволяет получить полное представление о финансовой устойчивости проекта и определить основные направления для оптимизации затрат.

Структура затрат

Основные категории расходов для запуска и поддержания веб-сервиса включают аренду серверов, расходы на back-офис, рекламу, техническую поддержку и зарплаты сотрудников. Ежемесячная плата за аренду серверов составляет 15000 рублей, что необходимо для обеспечения работы сервиса и его доступности для пользователей. В первый год работы расходы на back-офис составляют 20000 рублей в месяц, а со второго года эта сумма уменьшается до 5000 рублей в месяц, поскольку основные организационные затраты уже покрыты, и средний ежемесячный расход за периоды 5-36 месяцев составляет 8750 рублей. Ежемесячные расходы на рекламу составляют 65000 рублей, включая затраты на различные каналы продвижения, такие как социальные сети, блогеры на YouTube, Яндекс Директ и другие методы привлечения пользователей. Техническая поддержка начинается с пятого месяца и составляет 30000 рублей в месяц; до этого затраты на техническую поддержку равны нулю, так как сервис только запускается и поддержка пользователей на ранних этапах

осуществляется силами команды разработки.

Зарплаты сотрудников распределены следующим образом: аналитик получает 150000 рублей в месяц первые два месяца, проектирование архитектуры осуществляется во второй месяц и составляет 240000 рублей, два разработчика получают по 200000 рублей в месяц каждый, начиная со второго и по четвертый месяц включительно, затраты на тестирование составляют 90000 рублей в месяц в течение третьего и четвертого месяцев. Таким образом, структура затрат на первом году выглядит следующим образом:

Таблица 2.2 — Структура расходов

Месяц	1	2	3	4	5-36
Серверы	15000	15000	15000	15000	15000
Вакс-офис	20000	20000	20000	20000	8750
Реклама	65000	65000	65000	65000	65000
Техподдержка	0	0	0	0	30000
Аналитика	150000	150000	0	0	0
Проектирование	0	240000	0	0	0
Разработка	0	400000	400000	400000	0
Тестирование	0	0	90000	90000	0
Общая сумма	250000	890000	590000	590000	118750

Потоки выручки

Основные источники выручки для веб-сервиса включают подписки пользователей и рекламу. Введение различных уровней подписки, включая базовую бесплатную версию и премиум-версию с расширенными функциями, позволяет пользователям получать доступ к уникальным маршрутам, дополнительным фильтрам и приоритетной технической поддержке. Веб-сервис предлагает подписку стоимостью 50 рублей, которую покупает каждый третий пользователь. Рекламные доходы зависят от количества пользователей и их активности на платформе.

Гипотеза роста числа пользователей по экспоненте $N_{i+1} = N_i^{1.02}$ позво-

ляет прогнозировать увеличение аудитории. Обоснование этой гипотезы основывается на предположении, что каждый пользователь будет рекомендовать сервис своим друзьям и знакомым. Это создаст древовидную структуру рекомендаций, где количество узлов в дереве растет экспоненциально. Такое развитие событий вполне вероятно, учитывая тенденцию к быстрому распространению информации в социальных сетях и среди сообществ людей с активным образом жизни.

Выручка с рекламных показов рассчитывается по формуле:

$$R_{ads} = n \cdot N \cdot v \cdot k,$$

где n — среднее число посещений одним пользователем в месяц, N — количество пользователей, v — средний доход за показ одного рекламного объявления, k — среднее число показов за одно посещение. Эта формула основывается на предположении, что доход от рекламы напрямую пропорционален активности пользователей на платформе, числу их посещений и количеству показов рекламных объявлений во время этих посещений.

Выручка с продажи подписок рассчитывается по формуле:

$$R_{sub} = \frac{50 \cdot N}{3},$$

где 50 рублей — стоимость одной подписки, а каждый третий пользователь покупает подписку. Эта формула предполагает, что доля пользователей, готовых платить за премиум-функции, остается постоянной, и таким образом доход от подписок зависит исключительно от общего количества пользователей.

Использование данных моделей позволяет прогнозировать доходы и анализировать финансовую устойчивость веб-сервиса. Применение экспоненциального роста пользователей и детального учета затрат и выручки помогает принимать обоснованные решения для обеспечения долгосрочного успеха проекта.

На представленном графике 2.1 отображены аккумулятивные доходы и расходы веб-сервиса на протяжении трёх лет. Горизонтальная ось представляет месяцы, начиная с первого месяца запуска проекта и до тридцать

шестого месяца, то есть трёх лет. Вертикальная ось демонстрирует накопленные доходы и расходы в рублях.

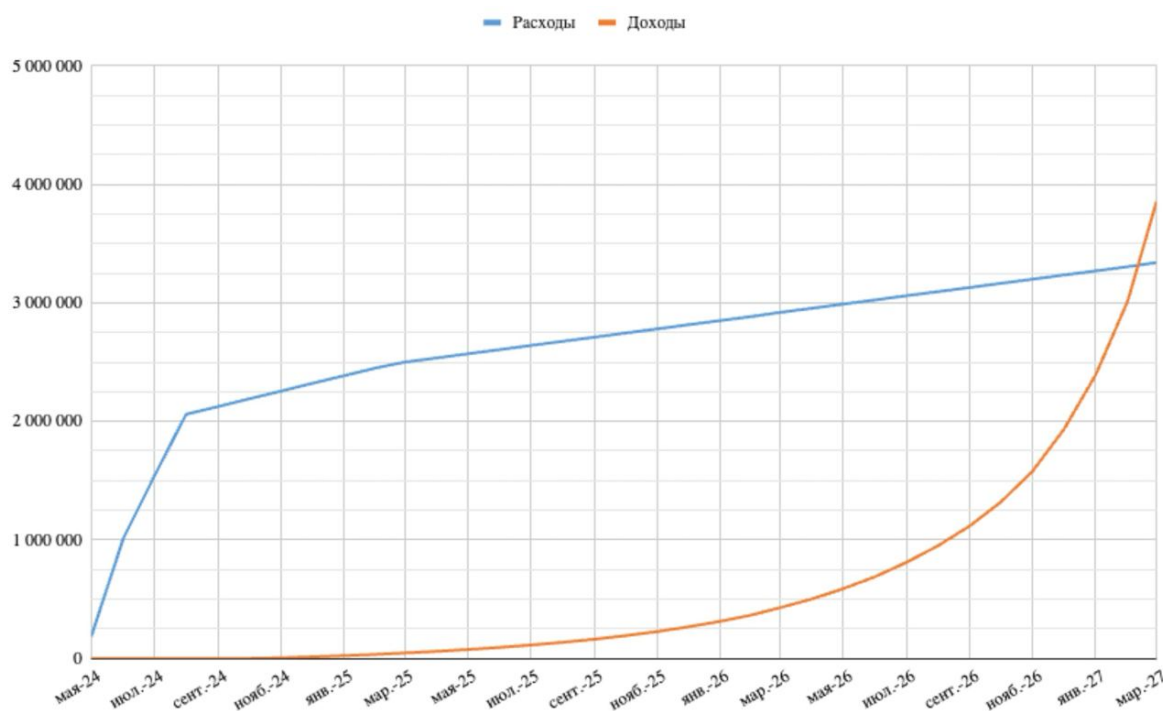


Рисунок 2.1 — График сходимости

Кривая, отображающая аккумулятивный расход, представляет собой кусочно-линейную зависимость, которая учитывает все затраты на аренду серверов, баск-офис, рекламу, техническую поддержку и зарплаты сотрудников. Начальный резкий подъём кривой связан с высокими стартовыми затратами на разработку и запуск проекта, которые затем стабилизируются.

Кривая аккумулятивного дохода также является кусочно-линейной и отражает постепенное увеличение доходов от подписок пользователей и рекламы. Рост доходов предполагает экспоненциальное увеличение количества пользователей благодаря рекомендациям и продвижению сервиса.

Точка пересечения двух кривых указывает на момент окупаемости проекта, когда накопленные доходы равны накопленным расходам. Этот момент важен для оценки финансовой устойчивости и эффективности проекта.

2.8 Ключевые метрики

Для оценки эффективности веб-сервиса и удовлетворенности пользователей используются различные продуктовые метрики. В данной главе рассмотрены основные показатели, такие как NPS (Net Promoter Score), CSI (Customer Satisfaction Index), CSAT (Customer Satisfaction Score) и LTV (Customer Lifetime Value). Эти метрики предоставляют объективную информацию о восприятии сервиса пользователями, их лояльности, удовлетворенности и финансовой ценности.

NPS измеряет лояльность пользователей к сервису и их готовность рекомендовать его другим. Этот показатель важен для понимания уровня приверженности аудитории, поскольку высокая лояльность пользователей способствует органическому росту за счет рекомендаций. NPS рассчитывается как разница между долей промоутеров и детракторов среди пользователей. Применение данной метрики позволяет определить, насколько клиенты готовы рекомендовать сервис своим друзьям и знакомым, что является индикатором общей удовлетворенности и вероятности повторного использования сервиса. Высокий NPS указывает на сильное положительное восприятие продукта и высокую вероятность его дальнейшего роста.

CSI отражает общую удовлетворенность пользователей сервисом. Этот индекс позволяет измерить общее восприятие качества продукта и его соответствие ожиданиям пользователей. CSI рассчитывается как средняя оценка всех отзывов, полученных от пользователей. Применение CSI помогает выявить общее настроение аудитории и понять, насколько продукт удовлетворяет потребности и ожидания пользователей. Высокий CSI указывает на то, что сервис соответствует ожиданиям пользователей и обеспечивает высокий уровень удовлетворенности.

CSAT измеряет удовлетворенность пользователей отдельным компонентом системы. Этот показатель позволяет более детально оценить восприятие конкретных функций или аспектов сервиса. CSAT рассчитывается как средняя оценка отзывов по конкретному компоненту системы. Применение CSAT помогает идентифицировать сильные и слабые стороны продукта, определяя области, которые требуют улучшения. Высокий

CSAT по конкретным компонентам указывает на то, что эти функции успешно выполняют свои задачи и удовлетворяют потребности пользователей.

LTV измеряет чистую прибыль, которую приносит один клиент за весь период использования сервиса. Этот показатель важен для понимания долгосрочной финансовой ценности каждого клиента. LTV рассчитывается на основе средней выручки на одного клиента, средней продолжительности использования сервиса и валовой прибыли. Применение LTV позволяет оценить рентабельность маркетинговых и продуктовых стратегий, а также помогает определить оптимальные пути для увеличения прибыли. Высокий LTV указывает на высокую ценность клиентов и успешность удержания пользователей на длительный срок.

В контексте веб-сервиса, реализующего алгоритм построения прогулочных маршрутов с фиксированной дистанцией и пользовательскими фильтрами, данные метрики имеют широкое применение. NPS позволяет оценить готовность пользователей рекомендовать сервис, что особенно важно для привлечения новой аудитории через сарафанное радио. CSI помогает измерить общее восприятие сервиса, выявляя удовлетворенность пользователей и соответствие их ожиданиям. CSAT дает возможность глубже понять, какие конкретные функции и компоненты сервиса наиболее востребованы и успешны, а какие нуждаются в доработке. LTV предоставляет информацию о финансовой ценности пользователей, помогая принимать обоснованные решения по маркетинговым и продуктовым стратегиям, направленным на увеличение прибыльности и удержание клиентов.

Использование этих метрик в совокупности позволяет получить комплексное представление о восприятии и эффективности веб-сервиса, а также принять меры для его дальнейшего улучшения и роста.

2.9 Нерыночное конкурентное преимущество

Нерыночное конкурентное преимущество предлагаемого веб-сервиса заключается в разработанном алгоритме генерации маршрутов по специ-

фическим фильтрам. Алгоритм интегрирует многочисленные пользовательские фильтры, такие как фиксированная дистанция маршрута, набор высоты, тип местности, наличие достопримечательностей и другие индивидуальные предпочтения, что делает его адаптивным и многофункциональным. Такой уровень персонализации и гибкости позволяет пользователям получать маршруты, максимально соответствующие их специфическим требованиям, что является значительным отличием от существующих геосервисов.

ГЛАВА 3

АЛГОРИТМ ГЕНЕРАЦИИ МАРШРУТОВ

В основе алгоритма построения маршрутов с заданной дистанцией лежит метод направленного блуждания на графе. Этот метод позволяет создавать маршруты, учитывая заданные пользователем параметры, такие как стартовая и конечная точки, а также желаемая дистанция. Начальная погрешность отклонения от желаемой дистанции составляет 10%, и в процессе работы алгоритма эта погрешность может динамически корректироваться. Алгоритм инициализируется данными параметрами и начинает процесс построения маршрута.

На начальном этапе пользователь задает начальную и конечную точки маршрута. Алгоритм также получает желаемую дистанцию маршрута и допустимую погрешность. Эти параметры служат исходными данными для инициализации алгоритма. В качестве основы для алгоритма используется граф, где вершины представляют собой значимые точки (например, перекрестки или достопримечательности), а ребра – возможные пути между этими точками.

Ключевой особенностью алгоритма является динамический выбор следующей вершины маршрута. Этот выбор основывается на нескольких факторах:

- удаленность достижимых вершин: Этот фактор позволяет избежать возврата в предыдущую вершину при наличии альтернативных вершин, обеспечивая более плавный и логичный маршрут;
- штраф за повторное посещение вершин: Чем давнее вершина была посещена, тем меньше назначается штраф. Это позволяет алгоритму минимизировать повторное прохождение уже пройденных участков;
- штраф за отдаление от целевой вершины: Этот фактор учитывает направление движения и стимулирует выбор вершин, которые приближают к цели.

Эти факторы работают в комплексе, позволяя алгоритму выбирать наиболее подходящие вершины для формирования маршрута, соответствующего заданной дистанции.

Алгоритм генерации маршрутов имеет тесную связь с марковскими сетями. Марковские сети используются для моделирования вероятностных процессов, где текущее состояние системы зависит только от предыдущего состояния. В контексте построения маршрутов, каждое состояние представляет собой текущую вершину графа, а переходы между состояниями (вершинами) осуществляются на основе вероятностных правил, учитывающих вышеописанные факторы. Это позволяет создавать маршруты, которые не только соответствуют заданным параметрам, но и оптимизируются по нескольким критериям.

У описанного метода существует недостаток в необходимости ручного выбора шаблона маршрута и угла. Для решения этого недостатка предлагается многократный запуск метода с различными шаблонами маршрута и шагом угла в 10 градусов. В результате создается список маршрутов, многие из которых будут геометрически слабо различимы. Для оптимизации этого процесса проводится кластеризация маршрутов.

Кластеризация позволяет группировать геометрически схожие маршруты, определяя в каждом кластере центроиды – маршруты, максимально удовлетворяющие пользовательским фильтрам. Эти центроиды представляют собой наиболее оптимальные маршруты в каждой группе, обеспечивая пользователю наилучшие варианты с учетом заданных параметров и условий.

Использование описанного алгоритма позволяет создавать маршруты с фиксированной дистанцией и заданными пользователем фильтрами, обеспечивая высокую степень гибкости и адаптивности. Динамическая корректировка погрешности, вероятностный выбор вершин на основе марковских сетей и кластеризация схожих маршрутов делают этот алгоритм эффективным и удобным для пользователей. Таким образом, веб-сервис предоставляет пользователям уникальные возможности для планирования и оптимизации своих маршрутов, значительно улучшая пользовательский опыт и удовлетворяя разнообразные потребности.

ГЛАВА 4

ОБЗОР ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

ГЛАВА 5

АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Andrew Ng, Machine Learning from Stanford University.
<https://www.coursera.org/learn/machine-learning>
2. Воронцов К.В., Введение в машинное обучение от
НИУ ВШЭ & Yandex School of Data Analysis.
<https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie>
3. Samuel, Arthur L. Some Studies in Machine Learning Using
the Game of Checkers // IBM Journal. - 1959. - №3. -
<http://www.cs.virginia.edu/evans/greatworks/samuel1959.pdf>
4. [https://en.wikipedia.org/wiki/Distance_\(graph_theory\)#Related_concepts](https://en.wikipedia.org/wiki/Distance_(graph_theory)#Related_concepts)