РЕФЕРАТ

Магистерская диссертация содержит 41 страниц, 2 таблицы, 4 рисунков. Список использованных источников включает 30 позиций.

ПЕРСОНАЛИЗИРОВАННЫЕ РЕКОМЕНДАЦИИ, ОНЛАЙН КАРТЫ, АЛГОРИТМЫ КОНТЕКСТНЫХ МНОГОРУКИХ БАНДИТОВ, ГЕНЕРАЦИЯ СИНТЕТИЧЕСКИХ ДАННЫХ, НЕЧЕТКИЕ МНОЖЕСТВА

Магистерская диссертация посвящена построению алгоритма по выдаче рекомендаций в сфере онлайн карт. В качестве возможных вариантов рекомендаций рассматриваются способы передвижения по построенному маршруту: такси, автомобиль, общественный транспорт, пешеходный маршрут или передвижение на велосипеде. Рекомендации выдаются после отправки пользователем запроса на построение маршрута по заданным точкам, и во время их выдачи учитываются характеристики пользователя и состояния окружающей среды в момент подачи запроса. Целью магистерской работы являлась разработка такого алгоритма, который сразу бы выдавал пользователю релевантный для него тип маршрута, повышая таким образом конверсию во ведение по выбранному маршруту – бизнес-метрики, которая является одним из самых важных индикаторов при оценке эффективности работы онлайн карт.

Во время изучения существующих алгоритмов были рассмотрены как традиционные методы построения рекомендаций в сфере веб-приложений, так и более новые подходы: набирающие в последнее время популярность алгоритмы многоруких бандитов. Для проведения экспериментов было решено остановиться именно на последних, ввиду их некоторых преимуществ над традиционными алгоритмами, а также большего соответствия специфике предметной области поставленной задачи. В результате численных экспериментов были достигнуты высокие показатели конверсии в навигацию по предложенному типу маршрута и выдвинуты предложения по дальнейшему улучшению рекомендаций.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc73522959)

[ОСНОВНАЯ ЧАСТЬ 9](#_Toc73522960)

[1. Теоретическая часть 10](#_Toc73522961)

[1.1. Постановка задачи 10](#_Toc73522962)

[1.2. Используемые термины 11](#_Toc73522963)

[1.3. Существующие подходы к решению задачи по выдачи персонализированных рекомендаций 12](#_Toc73522964)

[1.4. Алгоритмы многоруких бандитов 14](#_Toc73522965)

[1.5. Алгоритмы контекстных многоруких бандитов 18](#_Toc73522966)

[1.6. Математическая формулировка проблемы и подходы к решению 21](#_Toc73522967)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 26](#_Toc73522968)

[2.1. Схема работы алгоритма многорукого бандита в рамках поставленной задачи 26](#_Toc73522969)

[2.2. Генерация синтетических данных 28](#_Toc73522970)

[2.3. Получение истинных значений наград для каждого наблюдения 33](#_Toc73522971)

[2.4. Оценка качества работы алгоритмов многоруких бандитов 35](#_Toc73522972)

[ЗАКЛЮЧЕНИЕ 37](#_Toc73522973)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 38](#_Toc73522974)

### ВВЕДЕНИЕ

Для большинства современных веб-сервисов важной задачей является удержание клиента. Показано, что затраты на привлечение новых клиентов превышают затраты на удержание старых, поэтому выгоднее выделять большие ресурсы на увеличение благосклонности к продукту у текущих клиентов. [1,30] Зачастую, для удержания клиентов пытаются разработать более удобный и понятный пользовательский интерфейс, внедрить более привлекательный дизайн. Но так как пользователь в первую очередь обращается к продукту, чтобы решить определенную для себя задачу, то никакие улучшения дизайна не помогут, если с помощью веб приложения задачу эту получается решить плохо. Поэтому основным способом повышения лояльности клиентов является разработка простых, быстрых и понятных методов для эффективного решения задач пользователей. Один из таких способов – это внедрение ленты рекомендаций внутри веб приложения. При помощи рекомендаций можно помочь пользователю найти более интересный для него товар или услугу, заинтересовать новыми интересными предложениями и акциями, помочь быстрее решить свою задачу, сузив список товаров до наиболее релевантных. В настоящий момент развитие рекомендательных систем движется в сторону выдачи все более персонализированных рекомендаций, когда не просто выдаются обобщенные предложения, основанные на некоторых частотных характеристиках в поведении пользователя в прошлые моменты времени, а применяются сложные алгоритмы, направленные на выявление уникальных паттернов в поведении, чтобы, основываясь на выученных закономерностях, предложить наиболее релевантный запросам конкретного пользователя контент.[3]

Целью данной магистерской диссертации являлась разработка и применение алгоритма персонализированных рекомендаций по выбору типа маршрута для приложения онлайн карт. Мотивы обращения пользователей к онлайн картам могут быть разные: найти ближайшие магазины, добраться до определенного места, посмотреть текущую загруженность дорог или движение общественного транспорта. Но во всех запросах пользователя к картам так или иначе будет фигурировать понятие маршрута. Таким образом, одной из основных задач онлайн карт является предоставление пользователю наиболее подходящего маршрута, который позволит максимально эффективно и быстро добраться до требуемого места. Стоит отметить, что способов добраться до пункта назначения может быть несколько, задачей является предложить пользователю наиболее подходящие предпочтениям пользователя способы. Естественно предположить, что в приоритете для каждого пользователя онлайн карт будет быстрота и доступность построенных маршрутов. В этом предпочтения пользователей будут сходны, и такие задачи сейчас эффективно решаются при помощи различных алгоритмов, направленных на поиск оптимального пути по заданным точкам. [29] Но способов передвижения по построенному маршруту может быть несколько – можно воспользоваться общественным транспортом, добраться на такси или пешком. И в данной ситуации предпочтения уже могут варьироваться – часть пользователей будет склонна к передвижению на собственном автомобиле, некоторые предпочтут добраться пешком или на общественном транспорте. Логично также предположить, что на предпочтения пользователей в текущий момент времени будут влиять погодные условия и загруженность дорог. Сами пользователи могут быть объединены в группы со схожими предпочтениями, основываясь на общих для них характеристиках (пол, возраст, статус, семейное положение и т.д.) Учитывая выделенные характеристики пользователя и состояние окружающей среды в момент совершения маршрутного запроса, в приложении становится возможным не просто построить маршрут из точки A в точку B, но и предложить наиболее релевантный в текущих условиях способ передвижения вдоль построенного маршрута.

Целью данной магистерской диссертации являлась разработка алгоритма, который позволил бы сразу после построения маршрута предлагать подходящий для пользователя и способ передвижения по нему: на автомобиле, на общественном транспорте, на такси или пешком.

Для этого, необходимо было решить следующие задачи:

1. Анализ существующих алгоритмов по выдаче персонализированных рекомендаций и выбор наиболее подходящего из них для решения поставленной в рамках магистерской диссертации задачи.
2. Имплементация выбранных алгоритмов на языке Python.
3. Генерация синтетических данных для проверки качества работы алгоритмов.
4. Проведение численных экспериментов с целью оценки качества работы выбранных алгоритмов.

Объектом данной работы являются алгоритмы персонализированных рекомендаций в сфере веб-приложений. Предметом магистерской диссертации является персонализированные рекомендации типов маршрута в приложении онлайн карт.

Предполагается, что введение в приложении подобных рекомендаций позволит пользователю быстрее решать свои задачи, повысит долгосрочную удовлетворенность от использования веб-сервиса, а значит, положительно повлияет на метрики возвращаемости и оттока, которые являются одними из наиболее важных метрик при оценке успешности работы любого веб-сервиса. Если говорить именно о приложении онлайн карт, выдача персонализированных рекомендаций, подходящих для пользователя, повысит конверсию в навигацию по выбранному маршруту, что является немаловажным индикатором при анализе эффективности работы онлайн карт. Таким образом, внедрение в приложение онлайн карт подобных рекомендации, имеют высокую практическую значимость.

В разделе 1.1 описана постановка задачи и специфика рассмотренной предметной области, в разделе 1.2 введены понятия основных терминов, используемых в тексте магистерской диссертации, в разделе 1.3 описаны существующие подходы к решению задач по выдаче персонализированных рекомендаций, в разделе 1.3 описана теоретическая постановка задачи в рамках алгоритмов контекстных многоруких бандитов, в разделе 1.4 описаны алгоритмы многоруких бандитов, в разделе 1.5 описан особое семейство многоруких бандитов - алгоритмы, использующие контекстные данные, в разделе 1.6 задана математическая формулировка проблемы и подходы к решению. В разделе 2 описан процесс генерации синтетических данных для обучения алгоритма, а также представлены результаты численных экспериментов. В заключении подведены основные итоги проведенного исследования, а также представлены рекомендацию по возможным способам дальнейшего улучшения работы алгоритма

### ОСНОВНАЯ ЧАСТЬ

# Теоретическая часть

## Постановка задачи

Пользователь использует приложение онлайн карт, чтобы добраться из точки A в точку B. После задания начальной и конечной точки маршрута, пользователю на выбор предлагается 5 вариантов способа передвижения вдоль построенного маршрута: общественный транспорт, автомобиль, пешеходный маршрут, такси, велосипедный маршрут. При этом, в исходных условиях задачи, пользователю сразу открываются детали случайного типа маршрута. Если пользователя не устраивает предложенный ему изначально тип маршрута, он может переключиться на нужный ему в меню выбора.

После того, как пользователь остановился на конкретном типе маршрута, он может перейти в навигацию по данному маршруту. Переход в навигацию по выбранному маршруту представляет из себя конверсию из показа данного типа маршрута в навигацию по нему, или CTR данного типа маршрута (англ. click-through-rate).

В рамках поставленной задачи необходимо предлагать пользователю сразу после запроса на выдачу маршрута именно тот тип способа передвижению по нему, который бы был наиболее релевантным для предпочтений пользователя, а значит, по которому пользователь вероятнее всего перейдет в навигацию в дальнейшем. Таким образом, предполагается максимизировать общую конверсию в навигацию по маршрутам в приложении карт. Повышение общего CTR, в свою очередь, будет являться индикатором того, что пользователь успешно справляется со своими задачами внутри приложения, а значит, что растет долгосрочная лояльность пользователей по отношению к сервису и конкурентоспособность приложения на рынке.

## Используемые термины

В рамках данной задачи будут использоваться следующие термины и понятия:

* Агент – выбирает объекты для рекомендаций, производит обновление оценок значений наград;
* Контекст – набор признаков, используемых для выдачи рекомендаций;
* Действие, вариант – конкретный объект из множества объектов, доступных для рекомендации;

## Существующие подходы к решению задачи по выдачи персонализированных рекомендаций

Выдача персонализированных рекомендаций представляет из себя процесс сбора релевантной информации о посетителях веб-сервиса, анализ поведения пользователей внутри приложения и управление контентом приложения для того, чтобы, основываясь на накопленных знаниях, выдавать подходящие предложения для каждого уникального пользователя.[5] Рекомендательные системы применяется сейчас практически в каждом веб-сервисе, поэтому исследований на данную тему достаточно много. Первые алгоритмы, которые стали применятся к такого рода задачам и которые остаются популярными и по настоящее время, это алгоритмы коллаборативной фильтрации (англ. collaborative filtering) и фильтрации на основе контента (англ. content-based filtering). Алгоритм коллаборативной фильтрации пытается определить схожесть в предпочтениях между пользователями, основываясь на истории их поведения в веб-сервисе. [6] С помощью данного алгоритма можно получить хорошие результаты, если поведение пользователей достаточно схоже между собой, и набор рекомендуемых товаров или услуг является более-менее постоянным во времени. Такие рекомендации используются в сценариях, когда пользователю предлагаются товары или услуги, которые приобретали другие пользователи с аналогичной историей поведения внутри веб-сервиса. Рекомендации на основе контента предлагают пользователю товар, который похож на те товары или услуги, к которым он уже проявлял интерес. [7, 8] Данный вид рекомендации основывается на предположении, что интересы пользователя не меняются с течением времени. [9] Также применяются гибридные методы, которые в той или иной степени стараются совместить первые два алгоритма, и, таким образом, решить некоторые недостатки, связанные с каждым из них. Например, проблемой коллаборативной фильтрации является сложность в выдаче рекомендаций для нового товара/услуги. Однако, это проблему можно решить, если совместить алгоритм с фильтрацией на основе контента, где данного недостатка нет.

Однако, традиционные рекомендательные системы плохо работают в условиях, когда набор предлагаемых товаров/услуг и их популярность постоянно изменяется [10], когда предпочтения пользователя во времени не постоянны, когда регулярно появляются новые пользователи. Такие изменения существенно снижают качество рекомендаций.[11] В данных условиях алгоритм должен одновременно и рекомендовать пользователю наиболее подходящий для него товар/услугу, и одновременно исследовать новые варианты, которые также могли бы подойти пользователю. [12]. Возникает, так называемая проблема исследования/использования (англ. explore/exploit tradeoff), когда необходимо определить долю, которую можно отвести на исследование новых вариантов, чтобы максимизировать выгоду в долгосрочной перспективе.

## Алгоритмы многоруких бандитов

На разрешение проблем, указанных в предыдущем разделе, направлены алгоритмы многоруких бандитов, которые в последнее время начали обретать все большую популярность. Различные вариации данного алгоритма применяются во многих сферах: клинических испытаниях новых лекарств [13], персонализации рекламного [14] и веб-контента страницы [15], рекомендаций курсов при онлайн обучении [16, 17] и во многих других сферах.

Каждый из алгоритмов решает проблему исследования/использования, но делает это, используя различные инструменты и подходы с введением различных предпосылок. Можно выделить две основные группы алгоритмов в данном направлении: стохастические и не стохастические (англ. adversarial bandits). Для стохастических бандитов распределения наград для каждого из вариантов фиксировано и не меняется с течением времени, не зависит от предыдущих действий. В реальных задачах такие предпосылки зачастую не отражают действительности, поэтому развилась отдельная группа алгоритмов, где четких требований к наградам и механизму их получения нет.

Для стохастических бандитов условно можно выделить две группы: частотные алгоритмы (англ. frequentist approaches) и байесовский подходы (англ. bayesian approaches). В рамках частотных алгоритмов можно выделить эпсилон-жадный алгоритм и UCB алгоритм. Для стохастических бандитов награда на каждом раунде получается путем сэмплирования из распределения, которое зависит только от выбранного варианта.

Эпсилон-жадный алгоритм [18] работает таким образом, что в каждый момент времени он с вероятностью эпсилон будет исследовать случайным образом новые варианты, а с вероятность (1 –) выбирать вариант с максимальной оценкой среднего выигрыша. Если устремить количество раундов к бесконечности, то каждый вариант будет испробован бесконечное число раз, и оценка выигрыша для каждого варианта сойдется к истинному значению среднего с вероятностью, равной 1. Еще одно свойство данного алгоритма заключается в том, что при правильном подборе значения эпсилон среднее значение потерь на каждом шаге будет сходится к нулю с вероятностью 1. [19] При установке значения эпсилон в постоянную константу можно будет ожидать, что среднее значения потерь будет линейно сходится к нулю. При установке значения эпсилон пропорционально единице, деленной на номер раунда, можно будет ожидать логарифмической скорости схождения потерь. Минусы данного алгоритма заключаются в том, что во время исследования новый вариант выбирается случайным образом, вне зависимости от того насколько велика уверенность в его оценке средней выгоды. Поэтому может произойти такая ситуация, когда алгоритм на очередном раунде будет выбирать вариант, по которому существует уже достаточно много накопленной информации, вместо того, чтобы выбирать то действие, которое пока что редко проверялось, а значит, в перспективе может иметь большую награду, чем уже хорошо исследованные к текущему моменту варианты.

UCB алгоритм использует более продвинутую стратегию выбора варианта для исследований на каждом шаге.[20, 21] На каждом шаге алгоритм вычисляет оценку среднего выигрыша для каждого варианта, а также доверительный интервал так, что с высокой вероятностью (1 - ), где > 0, истинное среднее будет лежать ниже границы этого доверительного интервала. При правильном подборе границ доверительного интервала сходимость среднего значения потерь на каждом шаге к нулю будет логарифмической.[22] Этот алгоритм работает таким образом, что выбирает те варианты, о которых пока имеется меньше всего информации. Таким образом, мы настроены оптимистично по поводу вариантов с большой неопределенностью, то есть для нас с большим потенциалом. UCB алгоритм рассчитывает этот потенциал через установление верхней границы, ниже которой должно находится истинное значение ожидаемой награды с некоторой вероятностью. Эта верхняя граница представляет из себя функцию от числа раз, которые мы выбрали этот вариант. Таким образом, чем чаще мы выбираем вариант, тем уже становится его верхняя граница, тем меньше мы будем этот вариант выбирать в будущем (разумеется, только если значение его верхней границы будет меньше аналогичных значений у других вариантов).

Далее рассмотрим алгоритмы, которые используют теорему Баейса. Одним из таких алгоритмов является Томпсоновское сэмплирование. [23] Следуя этому алгоритму, первоначальная неопределенность в истинных распределениях наград выражается через априорное распределение, затем, каждый раз при получении новой информации (при выборе варианта) мы обновляем нашу оценку параметров распределения по теореме Баейса. Отличительной особенностью Томпсоновского сэмплирования является то, что оценка ожидаемой награды при каждом выборе определенного действия получается не из текущего значения математического ожидания для распределения, а путем сэмплирования случайной величины из данного распределения. Чем чаще мы будем выбирать определенное действие, тем ближе к истинному значению среднего будут оценки параметров для его распределения, что позволит нам определить лидера, и начать активно использовать его. Однако, именно сэмплирование позволяет нам каждый раз с некоторой вероятностью получать значения, достаточно отдаленные от среднего значения для данного распределения, это значит, что в ходе работы алгоритма даже после установления лидера, некоторые действия все равно будут время от времени проверяться. Этот факт снижает вероятность зацикливания на определенном действии, даже если наша оценка после нескольких раундов на нем максимальна, когда на самом деле существуют другие варианты с более высокой ожидаемой наградой (Эта проблема, например, характерна для эпсилон-жадных алгоритмов).

Другая группа алгоритмов моделирует работу не в стохастических условиях, когда получаемые награды не являются независимыми случайно распределенными величинами. Примером таких алгоритмов является Exp3 алгоритм. Для этого алгоритма, как и для всех не стохастических алгоритмов, значения наград для каждого раунда заранее фиксировано. Более того, противник (англ. adversary) может менять значения наград в зависимости от того, какую стратегию выберет играющий агент. Например, если стратегия игрока такова, что он все время выбирает один и тот же вариант, то нападающий может зафиксировать награды таким образом, что в каждом раунде награды для этого действия будут равны нулю, а для остальных – единице. Суть алгоритма Exp3 заключается в том, что для каждого действия фиксируются и обновляются на каждом раунде веса, в соответствии с которыми агент выбирает, какой из вариантов выбрать на каждом раунде. Задача алгоритма является быстрое реагирование на возможные изменения в динамике получаемых наград для каждого действия: если для ранее не очень прибыльного действия вдруг начинают поступать высокие награды, то необходимо быстро на это отреагировать.

## Алгоритмы контекстных многоруких бандитов

Однако, для выдачи персонализированных рекомендаций все из перечисленных в предыдущем разделе алгоритмов будут мало эффективны, так как ни один из них не учитывает контекст, в рамках которых эти рекомендации выдаются. Персонализацию предлагаемого контента можно достичь, учитывая совместно характеристики пользователя и рекомендуемых товаров/услуг. Именно поэтому в последнее время возрос интерес к алгоритмам контекстных бандитов, которые позволяют учитывать не только динамику в получаемых ответах от пользователя, но также и различные характеристики пользователей и окружающей среды. [24] Данный алгоритм нашел широкое применение для решения различных практических задаx, связанных с персонализацией рекомендаций. В статье [25] авторы рассматривают применение контекстных бандитов для рекомендаций в сфере спортивных приложений, направленных на приобщение пользователей к здоровому образу жизни. В данной статье авторы сначала указывают на существующие проблемы в сфере медицинских рекомендаций, с которыми традиционные методы плохо справляются: постоянно меняющиеся предпочтения пользователей (человек может решить сбросить вес, тогда ему понадобится рекомендации, связанные с повышенной активностью и умеренностью в питании, что может не соответствовать тем предложениям, которые он получал ранее), наличие тенденции у традиционных алгоритмов к рекомендации тех товаров и услуг, по которым собрано достаточно исторических данных, что может привести к однобоким рекомендациям, понижению уникальности в рекомендациях для каждого пользователя. Поведение конкретного пользователя по отношению к собственному здоровью находится под постоянным влиянием различной контекстной информации, зависит от постоянно меняющегося медицинского статуса, накопленного опыта по управлению собственным самочувствием, социального контекста. Вся эта информация формирует индивидуальную траекторию развития постоянно меняющихся интересов пользователя в этой сфере, несет в себе важную информацию. Поэтому, авторы предлагают использовать глубокое обучение, чтобы иметь возможность извлечь из истории поведения пользователя эти уникальные паттерны, уловить различные взаимодействия признаков. Они применяют техники выделения признаков из данных, основывая на глубоком обучении, получая таким образом векторы эмбеддингов пользователей. Аналогично, они получают векторы эмбеддингов рекомендуемых объектов. Все выделенные признаки они подают на вход алгоритму контекстных многоруких бандитов. Авторы выдвигают предпосылку, что решение -ого пользователя по поводу объекта в момент времени ,стохастически генерируется по определенной вероятности, которое зависит от контекста. Авторы моделируют эту вероятность, используя логистическую функцию:

где , вектор контекста, состоящий из – вектора эмбеддинга -ого пользователя, – вектора эмбеддинга - объекта, – вектора неизвестных коэффицентов, оценка которого постоянно уточняется в рамках работы алгоритма, используя Томпсоновское сэмлирование. Целью медицинской рекомендательной системы является максимизация величины за все время работы, по всем пользователям и рекомендуемым объектам.

В статье [26] авторы используют признаки пользователя и веб-страницы, чтобы отобразить на этой веб-страницы такую рекламу, по которой пользователь кликнет с наибольшей вероятностью. В статье [27] авторы применяют алгоритм контекстного бандита для персонализированной выдачи списка новостей, которые будут наиболее интересны пользователю. В частности, авторы выдвигают новый контекстный алгоритм – LinUCB (англ. Linear Upper Confidence Bound), который позволяет быстрее проводить вычисления, при этом обладает сравнительной, с лучшими алгоритмами из этого семейства, асимптотикой фунции потерь.

Основываясь на ряде преимуществ контекстных алгоритмов над традиционными при выдаче персонализированных рекомендаций, в данной магистерской работе было принято решения использовать именно контекстные алгоритмы для решения поставленной задачи.

Существуют различные подходы к решению задачи контекстных многоруких бандитов в зависимости от способа учета контекстной информации для выбора наиболее предпочтительного действия в каждом раунде. Глобально можно выделить подходы, которые используют линейные функции для моделирования зависимости значения наград от входящих признаков, и нелинейные алгоритмы, которые не предполагают существования какой-либо линейной зависимости между признаками и наградами.

В случае линейных алгоритмов, пространство признаков является подмножеством . При выборе действия получаем награду, ожидаемое значение которой является линейной комбинацией вектора и вектора неизвестных коэффициентов . Линейная структура задачи позволяет при выборе действия получать информацию о параметре выбранного действия, а также неявно о значениях других вариантов. Поэтому, в данном алгоритме традиционный подход в оценке средней награды заменяется на оценку признаков вектора .

## Математическая формулировка проблемы и подходы к решению

Приведем общую математическую постановку задачи для алгоритмов линейных контекстных многоруких бандитов.

Имеем набор раундов t = 1, 2, … работы алгоритма. Для каждого раунда алгоритм A выполняет следующие действия:

1. Алгоритм получает на вход текущего пользователя ut  Ut и текущий набор возможных действий l Lt. Каждое из действий характеризуется вектором признаков xl ∈ [0, 1]d , и ожидаемая награда для каждого действия линейна по этому вектору признаков:

для некоторого фиксированного набора неизвестных коэффициентов ∈ [0, 1]d ;

1. Алгоритм выбирает действие, основываясь на предыдущих полученных наградах по каждому из вариантов, и получает вознаграждение rt,lt ;
2. Далее, алгоритм обновляет свои оценки, основываясь на (xt,lt , lt, rt,lt );

Еще раз отметим, что отличительной особенностью данной постановки задачи является наличие вектора контекста, который включает в себя все релевантные признаки, способные повлиять на итоговое значение награды для каждого из действий. В традиционных алгоритмах многоруких бандитов данная информация не учитывается.

В данной магистерской диссертации рассматриваются и применяются два алгоритма линейных контекстных многоруких бандитов.

Первый, LinUCB [27], был представлен группой исследователей в 2010 году, и в качестве стратегии решения задачи заимствует подход традиционного алгоритма UCB, описанного в разделе 1.3. Конкретнее, в представленном подходе строится верхняя граница доверительного интервала для коэффициентов вектора контекста (в отличие от обычного алгоритма UCB, где доверительный интервал строится для значения ожидаемой награды). Для вычисления оценки весов в модели используется Ridge-регрессия (с параметром регуляризации = 1). по историческим данным векторов контекста и полученных наград для каждого варианта отдельно. Для каждого варианта используется свой вектор оценки коэффициентов. В модели присутствует всего лишь один параметр – , значение которого можно подобрать через Grid-search (Параметр отвечает за степень, с которой мы будем исследовать новые варианты). Сложность вычислений данного алгоритма линейно возрастает с увеличением количества вариантов. Верхняя оценка значения совокупных потерь - , где – набор доступных действий, – размерность признакового пространства, – количество раундов работы алгоритма, что сопоставимо со значениями других признанных алгоритмов в данной области.[28]

Ниже представлен псевдокод для алгоритма LinUCB:

**Дано:**

* Параметр

**Псевдокод алгоритма LinUCB:**

1. в течение = 1, 2, 3, …, выполнять
2. получаем признаки для всех действий
3. для всех
4. Условие: если раньше не встречалось, тогда

7. завершение условия

10. конец цикла
11. выбрать действие
12. получить значение награды по выбранному действию

15. конец цикла

Особенностью LinUCB алгоритма является то, что значения весов в модели обновляется только для действия, которое было выбрано в текущем раунде. То есть, моделируется ситуация, когда мы получаем знания о выигрыше только для действия, которое мы выбираем. Но в контексте текущей задачи такое ограничение может привести к потери некоторой информации о предпочтениях пользователя. На практике, мы имеем фиксированное количество вариантов типов маршрута, более того, в каждом моменте t пользователь имеет доступ ко всем вариантам, отсюда можно утверждать, что при выборе определенного типа, пользователь также неявно выражает свое предпочтение по отношению к другим типам доступных маршрутов. То есть, можно предположить, что, делая выбор в пользу одного типа маршрута, пользователь выражает свое отношение и к другим типам, который в момент выбора были также представлены на экране в меню выбора. Это наводит на мысль о том, что игнорируя данную информацию, мы можем потерять некоторое ценное знание по отношению пользователя к другим типам маршрута, которые не были выбраны. Самое простое в такой ситуации – это предположить, что все варианты, которые пользователь не выбрал, не отвечают его предпочтениям, то есть в терминах нашей модели они бы имели значение награды = 0. Однако, все же мы не можем быть уверены в этом наверняка – возможно, пользователь просто не заметил на экране этот вариант, а если бы он его заметил, то возможно выбрал именно его. Или пользователь просто не искал другие варианты, по привычке выбирая определенный тип, но если бы он посмотрел и другие варианты, возможно, он бы поменял свое мнение и выбрал бы что-то другое. В общем, появляется мысль, что мы должны построить определенную оценку ожидаемых наград для таких не выбранных вариантов. Группа исследователей в 2014 году предложила вариант алгоритма, LinPRUCB[4], который как раз решает данную задачу, путем сведения ситуации с доступностью информации только по выбранному варианту к ситуации, аналогичной ситуации полной информацией по всем вариантам (англ. full information setting). Так как в традиционном понимании получить ситуацию полной информации нет возможности, они пытаются сымитировать ее, вводя, так называемые, псевдо-награды, которые оцениваются для тех случаев, когда вариант не был выбран. Эти награды также выражаются через сумму скалярного произведения вектора весов на вектор признаков и слагаемого, которое выражает неопределенность по данному варианту. И в это слагаемое неопределенности авторы и предлагают внедрить информацию по тем ситуациям, когда действие не было выбрано. Для этого в модели теперь для каждого варианта поддерживается два набора обучающих выборок – одна для случаев, когда этот вариант был выбран, а вторая – когда не был выбран, оценка весов выводится с учетом этой дополнительной информации, и при непосредственном выборе действия эта информация также учитывается.

Ниже представлен псевдокод для алгоритма LinPRUCB:

**Дано:**

* Параметр
* Инициализация: для каждого варианта

**Псевдокод алгоритма LinUCB:**

1. в течение = 1, 2, 3, …, выполнять
2. получить признаки
3. выбрать действие
4. получить значение награды по выбранному действию
5. для всех
6. условие: если тогда

9. иначе,


13. конец условия

16. конец цикла
17. конец цикла

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## Схема работы алгоритма многорукого бандита в рамках поставленной задачи

В задаче по выдаче персонализированных рекомендаций в приложении онлайн карт определим основные понятия и характер их соответствия терминам, используемым в теоретической части для описания работы алгоритмов.

В качестве среды, где происходит работа алгоритма, выступает приложение онлайн карт. Агентом, который выбирает действия в каждом раунде и производит обновление оценок, выступает сам алгоритм по выбору типа маршрута. Набором доступных действий для рекомендаций являются пять доступных типов маршрутов: пешеходный маршрут, авто маршрут, такси, маршрут общественного транспорта, велосипедный маршрут. В качестве контекстных признаков, который алгоритм будет использовать для выдачи рекомендаций, выступают социально-демографические признаки пользователя: пол, возрастная группа, семейное положение, наличие собственного автомобиля, и различные характеристики окружающей среды, в момент совершения маршрутного запроса: балл пробок, погодные условия, количество осадок и сила ветра. Данные признаки были выбраны, так как они прямо влияют на предпочтения к способу передвижения по построенному маршруту. Действием со стороны пользователя является переход или не переход в навигацию по предложенному маршруту. В качестве награды мы получаем 1, если пользователь перешел в навигацию по предложенному маршруту, и 0 – если не перешел. Строго говоря, во время работы алгоритма, награда является числом из множества действительных чисел в промежутке от нуля до одного. Но во время оценки работы алгоритма происходит расчет точности, где за единицу мы принимаем случай, когда рекомендации и фактический выбор совпали, то есть мы получаем награду, и ноль – когда пользователь не выбрал предложенный тип маршрута.

В таблице 2.1 ниже кратко представлены описанные выше соответствия.

Таблица 2.1 Значение теоретических терминов в рамках поставленной задачи

|  |  |
| --- | --- |
| Обозначение | Соответствующее понятие в рамках задачи |
| Среда | Приложение онлайн карт |
| Агент | Алгоритм выбора типа маршрута |
| Набор доступных вариантов | Пешеходный маршрут, ОТ маршрут, такси, авто маршрут, велосипедный маршрут |
| Контекст | Социально-демографические признаки пользователя, характеристики типа маршрута, характеристики окружающей среды |
| Действие | Клик в навигацию по предложенному типу маршрута |
| Награда | 1 – если был клик, 0 - иначе |

## Генерация синтетических данных

Для проверки эффективности работы представленных алгоритмов, потребовалось сгенерировать такие синтетические данные, которые бы достаточно адекватно отражали реальность и содержали в себе некоторые закономерности, которые использованные алгоритмы смогли бы выучить.

Генерация синтетических данных происходила в несколько этапов. Первоначально были отдельно сгенерированы случайные пользователи и случайные вектора состояния окружающей среды. Для генерации пользователей сначала были отобраны несколько признаков (пол, возраст, семейное положение, уровень дохода, наличие собственного автомобиля) и через вероятности заданы распределения для каждого признака отдельно. В момент генерации пользователя происходило сэмплирование значения из заданного распределения, и в итоге получался набор признаков для одного пользователя. Для генерации векторов состояния окружающей среды (пробки, погода, количество осадков, сила ветра) была проведена аналогичная генерация по статистическим данным. По статистическим данным были заданы вероятности появления каждого признака в это время года (для климатических признаков) и дня недели (для балла пробок). Далее случайным образом был сгенерирован месяц года и день недели, и потом, основываясь на заданных вероятностях для этого времени, получено значение признака путем сэмплирования значения из соответствующего вероятностного распределения.

После этапа генерации отдельных признаков необходимо было перейти к генерации предпочтений пользователей к определенным типам маршрута. Формирование предпочтений происходило в две стадии: сначала сгенерированные пользователи были объединены в сегменты по схожести признаков (например, сегмент с пользователями среднего возраста, высокого достатка, с собственным автомобилем), и каждому получившемуся сегменту были заданы вероятностные распределения значения награды по каждому из типов маршрута. Моделирование предпочтений происходило при помощи нормального распределения, где для разных сегментов для одного и того же типа маршрута было разное математическое ожидание. Значение награды ограничивалось промежутком от нуля до одного. Чем выше значение, тем более предпочтительнее данный маршрут для этого сегмента. Эти предпочтения отражали постоянные предпочтения пользователя, которые зависят только от того сегмента, в который он попал.

Некоторые распределения признаков пользователей можно видеть на рисунках ниже.



Рисунок 2.1 Распределение пользователей по возрастным группам

В соответствии с рис. 2.1 можно видеть, что получившееся распределение, в целом, отражает реальную ситуацию в стране: многочисленная группа людей среднего возраста, самая наименьшая группа людей в старшем возрасте. Количество молодежи тоже невелико.

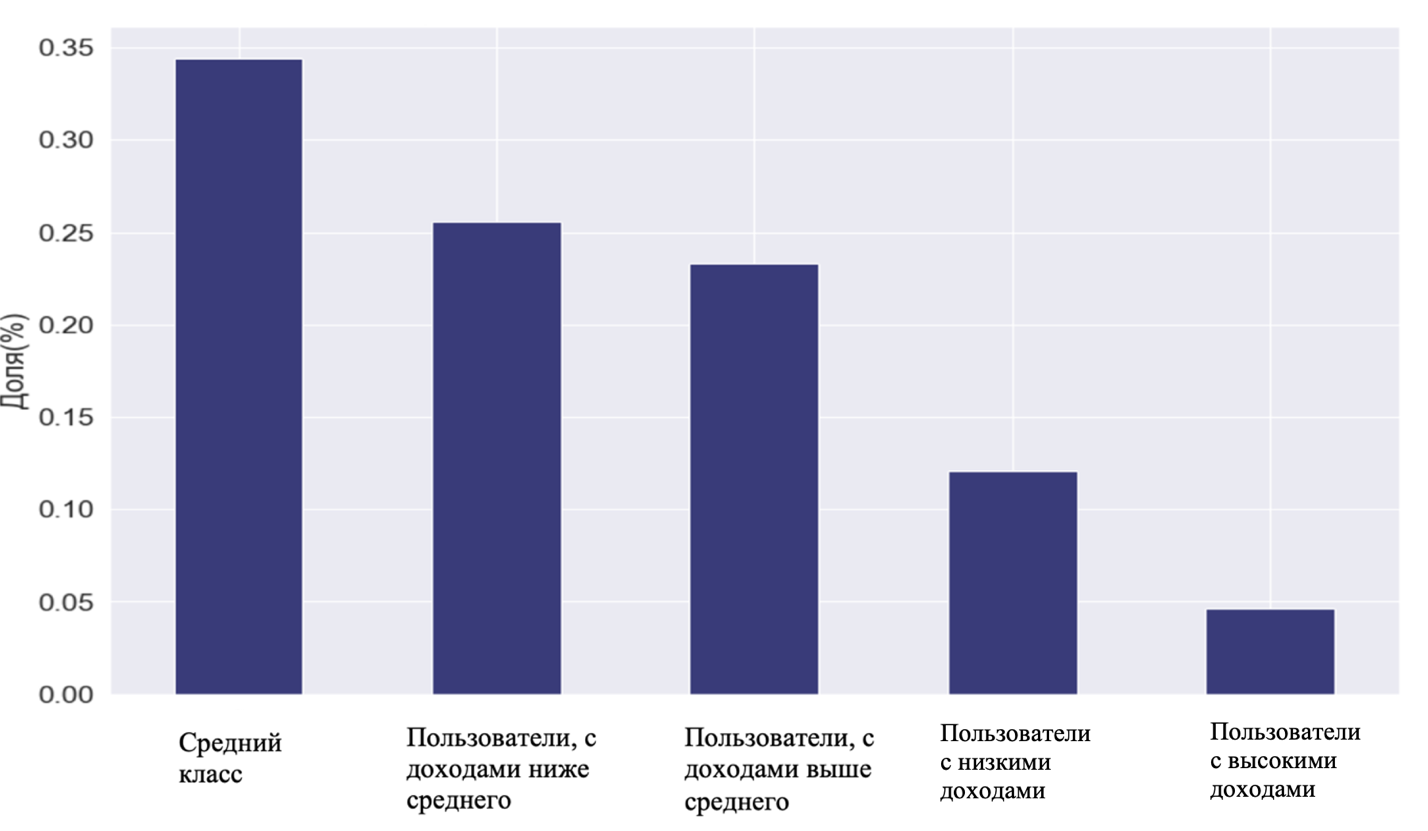


Рисунок 2.2 Распределение пользователей по уровню достатка

В соответствии с рис. 2.2 можем видеть, что распределение пользователей по уровню достатка тоже получилось достаточно логичным: самая многочисленная группа среднего класса, затем идут люди с доходами чуть ниже среднего, далее с доходами чуть выше, далее идут люди, живущие за чертой бедности, и затем самая малочисленная группа – очень богатые пользователи.

Далее были заданы правила влияния состояния окружающей среды в момент запроса на построение маршрута на постоянные предпочтения пользователей. Вводилась следующая предпосылка о том, что определенные характеристики состояния окружающей среды могут на время поменять постоянные предпочтения пользователя. Например, во время сильной пробки будет быстрее добраться на велосипеде или пешком, даже не смотря на то, что обычно пользователь предпочитает добираться общественным транспортом. Во время плохой погоды (жара, мороз, сильный ветер) лучше добираться на транспорте, даже если постоянно до этого пользователь предпочитал добираться пешком. Для того, чтобы учесть влияние окружающей среды, сначала признаки, получившиеся после генерации векторов состояния окружающей среды, были сгруппированы в нечеткие множества.

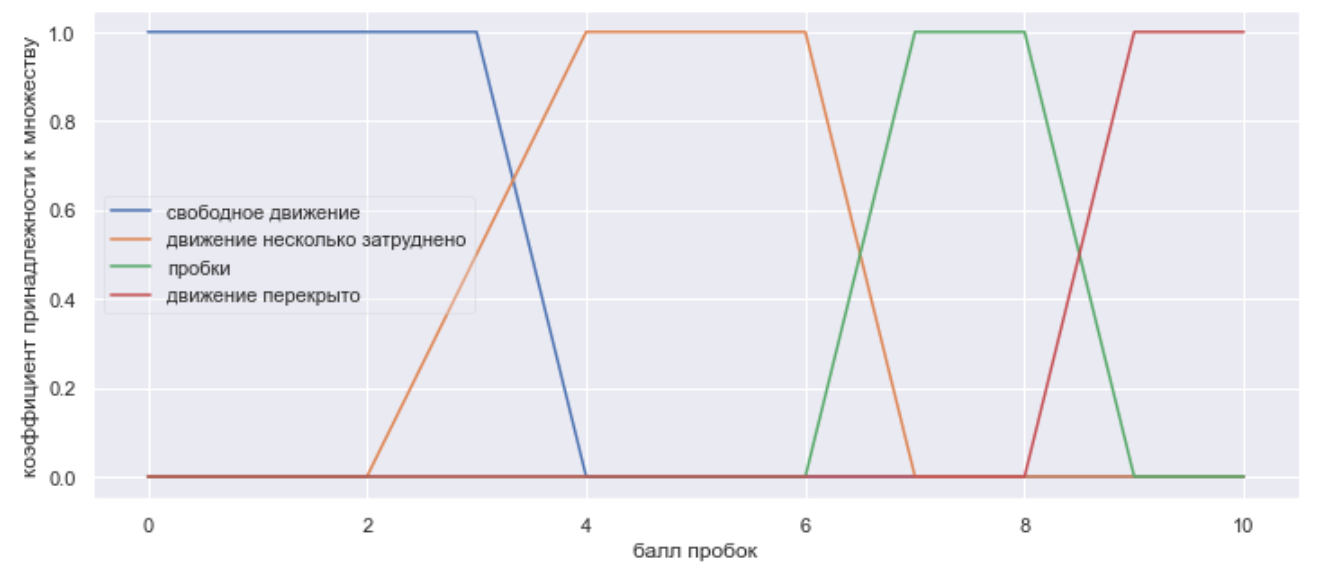


Рисунок 2.3 Пример отображение признака "Балл пробок" на нечеткое множество

В соответствии с рис. 2.3 можно видеть пример формирования нечеткого множества на основе признака “Балл пробок”. Можно видеть, что балл пробок относится к определенному подмножеству нечеткого множества, и одно значение может принадлежать сразу нескольким подмножествам. Например, значение пробок “3” может одновременно ощущаться как свободное движение и одновременно как несколько затрудненное. Удобство работы с нечеткими множествами как раз и заключается в том, что появляется возможность в более привычном виде охарактеризовать тот или иной признак, а также отразить субъективность текущего состояния этого признака, размыв границы его значений.

Далее, на получившиеся нечеткие множества были наложены правила соотнесение между принадлежностью признака к определенному множеству и предпочтительным вариантом типа маршрута. Например, было задано такое правило: “Если на улице тепло, нет ветра и осадков, но при этом затор на дорогах, то предпочтительным типом маршрута является пешеходный или велосипед”. Всего, на основании имеющихся нечетких множеств, было задано порядка 4 тысяч различных правил. После получения постоянного предпочтения пользователя к каждому из типов маршрута, если состояние окружающий среды подходило под существующее нечеткое правило, то постоянные предпочтения корректировались в соответствии с этим нечетким правилом. Корректировка происходила следующим образом. Награда, получившаяся после формирования постоянного предпочтения, сдвигалась в большую или меньшую сторону в зависимости от того, предпочтительным ли по соответствующему правилу является данный тип маршрута. Сдвиг происходил на величину, равную стандартному отклонению по всем получившимся постоянным предпочтениям, и чем в более возрастной сегмент попадал пользователь, тем меньше был этот сдвиг (моделируется ситуация, что чем старше люди становятся, тем менее охотно они меняют свои привычки).

Кратко, процесс формирования итоговых предпочтений пользователя к типам маршрута можно описать схемой ниже (рис. 4):

Корректирует постоянные предпочтения пользователя в момент совершения маршрутного запроса

Рисунок 2.4 Процесс формирования итоговых предпочтений пользователя к определенным типам маршрута

## Получение истинных значений наград для каждого наблюдения

В прошлом разделе по итогу был получен набор синтетических данных, каждое наблюдение (строка) которого состояло из вектора признаков пользователя, вектора признаков состояния окружающей среды и вектора с итоговыми предпочтениями пользователя по каждому из пяти типов маршрутов, выраженных как значения от нуля до единицы (по сути, получаемая величина награды может интерпретироваться как вероятность того, что пользователь кликнет на данный тип маршрута при условиях, заданных соответствующими признаками) Для того, чтобы во время запуска алгоритма многоруких бандитов была возможность посчитать точность работы и значение потерь, необходимо было получить истинные значения награды путем оценки весов по каждому из признаков для всех типов маршрута. После получения весов на части синтетических данных, на оставшейся части можно было бы сгенерировать истинные значения наград для каждого типа маршрута путем скалярного произведения весов и векторов признаков и добавления некоторого шума. Получившийся набор данных с истинными значениями наград и оценками весов можно было в дальнейшем использовать для запуска на нем алгоритмов многоруких бандитов.

Для оценки весов на части данных были использованных две модели: линейная регрессия с двумя видами регуляризации ridge-регуляризацией и lasso-регуляризацией, а также стохастический градиентный спуск. Для каждого типа маршрута оценка весов происходила отдельно. Был проведен оптимальный поиск параметров на тренировочных данных с помощью метода grid search, и в итоге, для оценки весов и последующей генерации истинных наград по этим весам была выбрана модель, дающая наилучшее качество (качество оценивалось по среднеквадратичной ошибке).

В итоге получился набор данных, включающий в себя наблюдение и истинное значение награды по каждому из типов маршрута для этого наблюдения. На этом наборе данных были запущенны, рассмотренные в предыдущих разделах, алгоритмы контекстных многоруких бандитов.

## Оценка качества работы алгоритмов многоруких бандитов

На полученном в предыдущем разделе наборе данных были запущены несколько алгоритмов многоруких бандитов: случайная модель, в качестве baseline-модели, которая на каждом шаге случайным образом выбирает тип маршрута для рекомендации, UCB для возможности сравнить работу алгоритма многорукого бандита без учета контекстных данных и LinUCB c LinPRUCB, как основные алгоритмы, которые рассматриваются в данной магистерской диссертации.

Также, в качестве эксперимента, отдельно были созданы две выборки данных, в одну из которых были добавлены полиномиальные признаки (вплоть до второй степени) и аналогичные признаки, но включающие только взаимодействие между признаками (без степеней для каждого признака).

Для оценки работы алгоритмов использовалась метрика точности. Она считалась следующим образом: на каждом раунде работы алгоритм выдавал ожидаемое значение наград по каждому из вариантов. Вариант максимальным значением считался предпочтительным, и выдавался в качестве рекомендации. Затем происходило обращение к истинному значению наград для этого наблюдения, и аналогичным образом наиболее предпочтительный вариант выбирался как вариант, с максимальным значением наград. Далее, по итогу работы алгоритма на всем объеме данных, подсчитывалась стандартная метрика точности:

где – истинное значение наиболее предпочтительного варианта на шаге ,

– значение наиболее предпочтительного варианта на шаге , выданного алгоритмом, – количество раундов работы алгоритма.

Результаты запуска алгоритмов на каждой из выборок можно посмотреть в таблице 2.2 ниже:

Таблица 2.2 Результаты работы алгоритмов

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм** | **С добавлением степеней признаков (до 2 степени включительно) + взаимодействий между признаками** | **С добавлением только взаимодействий между признаками** | **Без добавления степеней признаков** |
| Random | 0.19 | 0.19 | 0.20 |
| UCB | 0.33 | 0.23 | 0.44 |
| LinUCB | 0.42 | 0.25 | **0.53** |
| LinPRUCB | 0.42 | 0.25 | **0.53** |

В соответствии с таблицей 2.4.1 можно видеть, что наибольшее качество достигается для алгоритмов, которые используют контекстные данные, LinUCB и LinPRUCB. Причем, эту закономерность можно видеть для всех выборок. Также можно заметить, на выборках с полиномиальными признаками алгоритмы проявили себя хуже, чем на выборке без них. Это можно объяснить, что полиномы усложняют модель и замедляют обучение. Также, возможно, использование полиномов требует более тонкой настройки параметров самих алгоритмов многоруких бандитов.

Также можно заметить, что LinUCB и LinPRUCB имеют аналогичные результаты работы. Это можно объяснить тем, что для превосходства LinPRUCB над LinUCB требуется более тщательная настройка параметров: в LinPRUCB их три, в то время как в LinUCB только один параметр. Таким образом, можно заключить, что в условиях приоритета времени и быстроты работы, вариант с LinUCB является предпочтительным, так как работает быстрее, при этом без долгой настройки выдает приемлемое качество.

### ЗАКЛЮЧЕНИЕ

В данной магистерской диссертации ставилась задача разработки персонализированных рекомендаций для приложения онлайн-карт. Были рассмотрены основные существующие подходы, проанализированы преимущество и недостатки каждого. В ходе сравнительного анализа алгоритмов для формирования персонализированных рекомендаций было принято решение использовать алгоритмы контекстных многоруких бандитов, так как они решают многие проблемы более традиционных методов, таких коллаборативная фильтрация или фильтрация на основе контента.

Было выбрано два конкретных алгоритма из семейства контекстных алгоритмов многоруких бандитов: LinUCB и LinPRUCB. Выбранные алгоритмы были реализованы на языке Python, при помощи различных библиотек для анализа и визуализации данных.

Для проверки работы алгоритмов потребовалось сгенерировать набор синтетических данных, которые включали в себя сгенерированные признаки пользователей о окружающей среды в момент маршрутного запроса. Для генерации данных использовались вероятностные подходы и подходы на основе составления нечетких множеств.

На полученных синтетических данных были проведены различные численные эксперименты. Было выявлено, что наибольшую точность выдают алгоритмы, использующие контекстные данные. Для сравнения, алгоритм UCB, который работает по схожему с LinUCB подходу, но не учитывает контекст, выдавал качество на 25% хуже.

В целом, контекстные алгоритмы работали эффективнее случайной модели на 165%, что может говорить о значительных возможных выгодах, которые можно получить при внедрении их в реальное веб-приложении.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Gummesson E. 1999. Total Relationship Marketing: Rethinking marketing Management from 4Ps to 30Rs. Butterworth Heinemann: Oxford (p.9).
2. A. Slivkins. Introduction to multi-armed bandits. Foundations and Trends in Machine Learning, 12(1-2):1–286, 2019. ISSN 1935-8237. [17, 358]
3. P. Brusilovsky, A. Kobsa, and W. Nejdl, editors. The Adaptive Web — Methods and Strategies of Web Personalization, volume 4321 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007.]
4. Ku-Chun Chou, Hsuan-Tien Lin, Chao-Kai Chiang, and Chi-Jen Lu. 2014. Pseudo-reward Algorithms for Contextual Bandits with Linear Payoff Functions.. In ACML.
5. B. Mobasher and S. S. Anand, editors. Intelligent Techniques for Web Personalization, volume 3169 of Lecture Notes in Artificial Intelligence. Springer-Verlag, 2005.
6. Goldberg D., Nichols D., Oki B. M., Terry D. Using collaborative filtering to weave an information Tapestry // Special issue on information filtering, 1992 Vol. 35, Issue 12 P. 61-70.]
7. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions // IEEE Transactions on Knowledge and Data Engineering, 2005 Vol. 17, Issue 6 P 734-749.
8. Celma O. Music Recommendation and Discovery. Springer-Verlag Berlin Heidelberg, 2010 194 p.
9. Pazzani M., Billsus D. Learning and revising user profiles: The identification of interesting web sites // Machine Learning - Special issue on multistrategy learning, 1997 Vol. 27, Isue 3 P. 313–331.
10. S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste. Naïve filterbots for robust cold-start recommendations. In Proc. of the 12th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining, pages 699–705, 2006
11. W. Chu and S.-T. Park. Personalized recommendation on dynamic content using predictive bilinear models. In Proc. of the 18th International Conf. on World Wide Web, pages 691–700, 2009.
12. C. C. Aggarwal. Recommender Systems: The Textbook. Springer, 2016 p.418
13. S. S. Villar, J. Bowden, and J. Wason, “Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges,” Statistical science: a review journal of the Institute of Mathematical Statistics, vol. 30, no. 2, p. 199, 2015.
14. Schwartz, E. M., E. T. Bradlow, and P. S. Fader. 2017. “Customer acquisition via display advertising using multi-armed bandit experiments”. Marketing Science. 36(4): 500–522
15. AGARWAL, A., BIRD, S., COZOWICZ, M., HOANG, L., LANGFORD, J., LEE, S., LI, J., MELAMED, D., OSHRI, G., RIBAS, O., SEN, S., AND SLIVKINS, A. A multiworld testing decision service. arXiv preprint arXiv:1606.03966 (2016).
16. Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In EDM, pages 424–429, 2016.
17. J. Xu, T. Xing, and M. van der Schaar. Personalized course sequence recommendations. IEEE Trans. Signal Process., 64 (20):5340–5352, 2016.
18. John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In Advances in Neural Information Processing Systems 20, 2008.
19. H. Robbins. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5):527–535, 1952.
20. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47(2–3):235–256, 2002.
21. T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics, 6(1):4–22, 1985.
22. T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics, 6(1):4–22, 1985.
23. W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3-4):285–294, 1933.
24. John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In Advances in neural information processing systems, pages 817–824, 2008.
25. Tewari Ambuj, Murphy Susan A. From ads to interventions: Contextual bandits in mobile health. In: Rehg Jim, Murphy Susan A, Kumar Santosh., editors. Mobile Health: Sensors, Analytic Methods, and Applications. Springer; 2017
26. Naoki Abe and Atsuyoshi Nakamura. Learning to optimally schedule internet banner advertisements. In Proceedings of the Sixteenth International Conference on Machine Learning, pages 12–21. Morgan Kaufmann Publishers Inc., 1999
27. Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web (WWW), 2010.
28. P. Auer. Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research, 3:397–422, 2002
29. Басараб М.А., Домрачева А.Б., Купляков В.М. Алгоритмы решения задачи быстрого поиска пути на географических картах // Инженерный журнал: наука и инновации, 2013. № 11. - URL: http://engjournal.ru/ catalog/it/hidden/1054.html
30. Третьяк О.А. Ценность клиента в течение его жизненного цикла: развитие одной из ключевых идей маркетинга взаимоотношений // Российский журнал менеджмента №9 (3). C. 55-68 (С57)