САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

Миннигареева Лена Рашитовна

Магистерская диссертация

Анализ кликовых моделей для повышения качества ранжирования результатов поисковой выдачи

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа:

Математическое и информационное обеспечение экономической деятельности

Научный руководитель,
доцент
Балыкина Ю.Е.

Санкт-Петербург
2018

Оглавление

Введение

Учитывая стремительный рост популярности веб-поиска в последние годы, все больше внимания среди информационно-поискового сообщества привлекает задача моделирования поведения пользователей поисковых систем. Развитие таких моделей способствует лучшему пониманию поведения пользователя, что в конечном счете помогает сделать процесс поиска необходимой пользователю информации более эффективным. Модели поведения пользователей могут быть также использованы в качестве элемента системы оценивания «ненаблюдаемых элементов», таких как релевантность документов в поисковой выдаче, способствуя улучшению ранжирования результатов поиска. При этом зачастую проведение экспериментов с реальными пользователями труднореализуемо. Так, например, некоторые исследователи имеют ограниченные возможности взаимодействия с пользователями из-за секретности или коммерческих ограничений. Даже компании, имеющие доступ к миллионам пользователей, не могут позволить себе постоянно проводить эксперименты с их привлечением. На этапе разработки им приходится использовать модели, имитирующие поведение пользователя. Поэтому имитационное моделирование пользователей играет существенную роль в понимании последствий внедрения тех или иных алгоритмов для улучшения качества поиска или представления результатов поисковой выдачи.

Поисковым системам необходимо учитывать то, как пользователь воспринимает качество результатов поисковой выдачи. В [1] показано, что пользователи, как правило, предпочитают первый документ на странице результатов поисковой системы последнему, что помогает ранжировать документы наиболее выгодным для пользователей образом. Существуют метрики, основанные на поисковых логах. Обратная связь от пользователей, в том числе в виде кликов, позволяет учитывать предпочтения пользователей. Клики пользователей на результаты веб-поиска являются

очень важной информацией для поисковиков. Клики могут дать представление о том, какие результаты заинтересовали пользователя, а какие – нет. Поисковые системы используют данную информацию для того, чтобы оценивать и улучшать качество поиска. Чтобы понять и описать поведение пользователей, были предложены так называемые кликовые модели. Обученная кликовая модель помогает лучше понять, как именно пользователи кликают на ссылки, представленные в результатах поискового запроса, позволяет предсказать клики и т.д.

В алгоритмах машинного обучения ранжированию все чаще используют кликовые модели для определения релевантности документа. Здесь также очень важно качество обученной кликовой модели, т.к. чем точнее модель предсказывает клики, тем правильнее будет определена релевантность документа, что напрямую влияет на результаты работы алгоритма ранжирования.

На первый взгляд идея ранжирования на основе кликовых логов кажется достаточно простой. Чем выше CTR (click-through rate - коэффициент кликабельности) документа, тем выше нужно располагать его на странице поисковой выдачи. Однако клик на документ или его отсутствие не всегда означает релевантность:

1. Документы, расположенные первыми на странице результатов поисковой выдачи кликаются чаще.

2. Поисковые запросы могут быть достаточно многозначными и возникают ситуации, когда один и тот же документ будет релевантным для одного пользователя и нерелевантным для другого.

3. Необходимо также учитывать, что в поисковой сессии может быть несколько запросов. Так, информация, хранящаяся в документе, может быть уже известна пользователю из прошлых запросов.

4. Пользователь принимает решение о клике на основе информации, описанной в сниппете (snippet – описание документа на странице результатов поисковой выдачи). Даже если документ содержит необходимую пользователю информацию, но она никак не отражена в сниппете, скорее всего клика на документ не произойдет. И наоборот, пользователь может заинтересоваться сниппетом и кликнуть на документ, но нужной информации там не найти.

В связи с этим постоянно разрабатывается множество новых алгоритмов ранжирования. Кликовые модели также широко используются в экспериментах при их внедрении. Чем выше качество обученной модели, тем более точно она предсказывает клики, и соответственно, более точными становятся результаты экспериментов.

Постановка задачи

При использовании кликовых моделей для повышения качества ранжирования результатов поисковой выдачи обычно используют одну кликовую модель, которая предсказывает клики всех пользователей поисковой системы. В данной работе рассматривается предположение о том, что если разделить пользователей на группы и для каждой из групп пользователей обучить свою кликовую модель, качество полученных моделей будет выше и соответственно точность предсказанных кликов также повысится.

Для проверки рассматриваемого предположения необходимо определить признаки пользователей, характеризующие их поведение при информационном поиске. Затем на основе полученных признаков нужно провести кластеризацию пользователей. Далее на каждом из полученных кластеров и на всех пользователях сразу обучаются кликовые модели. Так можно будет проверить, стали ли результаты на группах пользователей лучше. Для каждой из обученных моделей необходимо вычислить метрики качества. Анализ полученных результатов позволит определить, влияет ли кластеризация пользователей на качество предсказываемых кликов.

Обзор литературы

Задача построения функции ранжирования делится на две подзадачи: определение факторов ранжирования и формирование самой функции ранжирования, вычисляющей степень соответствия документа запросу на основе выделенных факторов. [2]

Для нахождения оптимальной функции ранжирования часто применяются методы машинного обучения. [3]

В 2011 году компания Яндекс проводила конкурс по разработке алгоритмов предсказания релевантности «Интернет-математика 2011» [4], в результате которого было разработано множество алгоритмов ранжирования, в том числе с использованием информации о кликах пользователей. [5-8].

В [7], например, предложен алгоритм улучшения качества ранжирования на основе предсказания релевантности по паре запрос-документ. Для этого используется 24 фактора, 5 из них получаются путем определения релевантности с помощью кликовой модели.

В [8] предложен метод предсказывания релевантности документа на основе моделей DBN (Dynamic Bayesian Network) и UBM (User Browsing Model).

В [5] используются различные вероятности клика на документ (вероятность клика, вероятность, что на документ кликнут последним, вероятность клика на документ рангом выше и т.д.), которые определяются также с помощью кликовых моделей.

В [9] рассмотрено влияние кликов пользователя как на переранжирование с учетом логов предыдущих запросов, так и на сами алгоритмы машинного обучения ранжированию, предложен метод улучшения ранжирования на основе логов. Из логов выделяется множество факторов ранжирования, таких как доля кликов на данный документ

относительно всех показанных документов, среднее время чтения документа, клики на предыдущие документы страницы результатов поисковой выдачи и т.д.

При разработке нового алгоритма ранжирования и экспериментов с целью анализа качества разработанного алгоритма для имитации поведения пользователя также используются кликовые модели [10].

Одной из первых работ, посвященных кликовым моделям, является [11], в ней вводится каскадная модель и ряд других кликовых моделей, основанных на гипотезе о смещении [1], которая, в свою очередь, основывается на анализе логов и наблюдении за тем, куда направлен взгляд пользователя при веб-поиске. В [12] и [13] представлены кликовые модели, используемые для оценки ранжирования результатов поисковой выдачи.

На основе [11] были разработаны следующие модели: UBM (User Browsing Model) [14], DBN (Dynamic Bayesian Network) [15] и DCM (Dependent Click Model) [16]. Большинство существующих кликовых моделей основываются на них.

В [17] описаны модели, предсказывающие поведение пользователя по движению мыши. Недавние исследования в этой области были использованы для улучшения кликовых моделей [18].

В последнее время веб-поиск становится все более персонализированным и это также отражается в кликовых моделях [19].

Оценка качества информационного поиска онлайн

Рассмотрим три типа критериев, по которым определяется качество информационного поиска и основные метрики.

1. Клики. Метриками кликов являются их количество, ранг кликнутых документов и отсутствие клика. Считается, что чем больше кликов, тем лучше, с рангом и отсутствием клика наоборот.

2. Время. Метрики: время, которое мы читаем результаты (чем больше, тем лучше), время, которое пользователь провел на странице с результатами поисковой выдачи до первого клика (чем меньше, тем лучше) и время, потраченное до последнего клика (чем больше, тем лучше).

3. Запросы. Запросы мы измеряем количеством их переформулировок и количеством запросов без единого клика. Чем меньше значение этих метрик, тем лучше.

A/B testing

Имеется две модификации поисковой системы. В эксперименте участвует небольшой процент пользователей. Половина из них в течение какого-то времени (например, неделю) получает результаты из одной модификации, другая половина из другой. По истечении срока эксперимента на основе выбранных метрик определяется, какая из модификаций лучше.

Этот метод хорош тем, что позволяет измерить любую модификацию поисковой системы, будь то изменение цвета сниппета или изменение алгоритма ранжирования. Также плюсом является то, что можно измерять что угодно, т.к. для каждой из систем доступны полные результаты.

Однако разбиение пользователей может оказаться неравномерным и, например, в одну половину попадут пользователи, которые кликают много, а

в другую – мало, тогда для получения хороших результатов необходимо проводить достаточно много экспериментов.

Interleaving

Имеется запрос от пользователя и две системы, которые могут на него ответить. Результаты этих систем смешиваются и показываются пользователю. На основе выбранных метрик выбирается лучшая система.

Interleaving не требует разбиения пользователей, достаточно чувствителен и занимает меньше времени, чем A/B тестирование, но может учитывать только клики.

Онлайн метрики хороши тем, что учитывают поведение реальных пользователей и не требуют больших финансовых затрат, однако полученные результаты смещенные, их сложно интерпретировать, эксперименты занимают много времени и могут оттолкнуть пользователя. Также обычно компании не имеют возможности проводить одновременно много экспериментов.

Оценка качества информационного поиска оффлайн

Precision and recall [20]. Точность и полнота являются наиболее часто используемыми и базовыми мерами эффективности информационного поиска.

Точность (P) – количество релевантных документов среди всех документов на странице результатов поисковой выдачи. Чем выше стоят релевантные документы, тем выше точность.

Полнота (R) – количество выведенных релевантных документов среди всех релевантных документов. Чем больше релевантных документов выводится, тем выше полнота.

Чем выше точность, тем меньше полнота и наоборот. Для некоторых пользователей важна высокая точность, релевантность первых результатов поиска. Для других важна полнота и они готовы потерпеть низкую точность ради полной информации.

Существует метрика, связывающая между собой понятия полноты и точности: F-мера.

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R},$$

$$\beta^2 = \frac{1-\alpha}{\alpha}$$

Значение параметра $\beta < 1$ используются, когда нам важнее точность, $\beta > 1$ – когда важнее полнота. Чаще используется F_1-мера – F-мера со значениями $\alpha = 0{,}5, \beta = 1$:

$$F_1 = \frac{2PR}{P+R}$$

Минусом этих метрик является то, что ранги документов здесь не учитываются.

Полнота и точность для каждого ранга:

$$P@k = \frac{\text{количество релевантных документов ранга } k}{k}$$

$$R@k = \frac{\text{количество релевантных документов ранга } k}{\text{количество всех релевантных документов}}$$

Reciprocal rank (RR)

$$RR = \frac{1}{\text{ранг первого релевантного документа}}$$

Average precision (AP)

$$AP = \frac{\sum_{d=rel} P@k_d}{\text{количество релевантных документов}}$$

Проходим по всем релевантным документам и считаем точность на том ранге, на котором они появляются.

Минус этих метрик: пользовательское поведение не учитывается.

Discounted cumulative gain (DCG). Одна из самых часто используемых метрик. Используется в Яндекс, Google и многих других компаниях. Эта метрика учитывает больше двух рангов

$$R_k \in \{0,1,2,3,4\}$$

Полезность от результатов:

$$CG = \sum_{k=1}^{N} (2^{R_k} - 1)$$

$$DCG = \sum_{k=1}^{N} \frac{2^{R_k} - 1}{\log(k+1)}$$

Также часто используют $NDCG$:

$$NDCG = \frac{DCG}{DCG_{ideal}},$$

где DCG_{ideal} - значение DCG для идеальной страницы результатов поисковой выдачи.

Rank-baised precision (RBP). Пользователь смотрит на каждый следующий документ с вероятностью θ. Вероятность посмотреть на документ с рангом k равна θ^{k-1}.

Польза документов с рангом k:

$$U@k = \theta^{k-1} R_k$$

$$RBP = (1-\theta) \sum_{k=1}^{N} \theta^{k-1} R_k$$

Expected reciprocal rank (ERR). Если пользователь встретил релевантный документ, он останавливается, иначе он смотрит следующий документ с вероятностью θ. Вероятность посмотреть на документ с рангом k равна $\prod_{i=1}^{k-1}(1-R_i)\theta$.

$$ERR = \sum_{k=1}^{N} \frac{1}{k} \theta^{k-1} R_k \prod_{i=1}^{k-1} (1-R_i)$$

Базовые кликовые модели

RCM (Random Click Model) [21]

Случайная кликовая модель – самая простая. Она имеет только один параметр и определяется следующим образом:

$$P(C_u = 1) = \rho.$$

Здесь C_u – случайное событие – клик на документ u. Это означает, что все документы имеют одну и ту же вероятность быть кликнутыми и эта вероятность равна ρ. Вероятность клика ρ можно посчитать как отношение количества всех кликов к количеству показанных документов.

RCTR (Rank-based CTR model) [22]

Данная модель основывается на метрике CTR (click-through rate), которая характеризует кликабельность документа. Так, например, CTR первого документа на странице результатов поисковой выдачи приблизительно равен 0,45, тогда как у десятого документа CTR принимает значение 0,05. Данная модель, также как и RCM имеет один параметр – вероятность клика на документ. Но здесь эта вероятность зависит от позиции документа:

$$P(C_r = 1) = \rho_r$$

Здесь C_r – случайное событие – клик на документ с рангом r.

DCTR (Document-based CTR model) [11]

Модель отличается от RCTR тем, что вероятность клика на документ зависит от пары запрос-документ:

$$P(C_u = 1) = \rho_{uq}$$

Данную модель (как и другие, более сложные модели) нужно переобучать чаще, чем RCM или RCTR, особенно учитывая тот факт, что некоторые документы или пары запрос-документ могли отсутствовать в тех логах, на которых мы обучали кликовую модель.

PBM (Position-based model) [11]

Многие кликовые модели основываются на следующей гипотезе:

$$C_u = 1 \Leftrightarrow E_u = 1, A_u = 1.$$

Здесь E_u – случайное событие – прочитать сниппет (небольшой отрывок текста, использующийся в качестве описания ссылки в результатах поиска) документа u, A_u – случайное событие – заинтересоваться сниппетом документа u. Это означает, что пользователь кликает на документ тогда и только тогда, когда он прочел его сниппет и был им заинтересован. Случайные события E_u и A_u считаются независимыми.

В [1] показано, что вероятность того, что пользователь прочтет сниппет, зависит в основном от его ранга (позиции) на странице результатов поиска и обычно уменьшается с увеличением ранга. Для того, чтобы включить это в модель, каждому рангу задается своя вероятность прочтения сниппета. Позиционную кликовую модель формально можно записать следующим образом:

$$P(C_u = 1) = P(E_u = 1) \cdot P(A_u = 1)$$
$$P(A_u = 1) = \alpha_u$$
$$P(E_u = 1) = \gamma_u$$

CM (Cascade Model) [11]

Каскадная модель предполагает, что пользователь просматривает документы на странице результатов поиска сверху вниз до тех пор, пока не найдет нужный. В соответствии с этим, первый документ просматривается

всегда, тогда как остальные документы рассматриваются только в том случае, если предыдущий документ был просмотрен, но не был кликнут. Каскадную кликовую модель можно описать следующим образом:

$$C_u = 1 \Leftrightarrow E_u = 1 \; and \; A_u = 1$$

$$P(A_u = 1) = \alpha_u$$

$$P(E_1 = 1) = 1$$

$$P(E_u = 1 | E_{u-1} = 0) = 0$$

$$P(E_u = 1 | C_{u-1} = 1) = 0$$

$$P(E_u = 1 | E_{u-1} = 1, C_{u-1} = 0) = 1$$

Данная кликовая модель имеет простую оценку, т.к. случайное событие – чтение сниппета – наблюдаемая величина: модель подразумевает, что все документы, вплоть до первого кликнутого, были просмотрены пользователем.

Основное различие между позиционной и каскадной моделями в том, что в позиционной модели вероятность клика на документ не зависит от кликов на предшествующие ему документы, но с ее помощью можно описывать сессии с количеством кликов более одного, что невозможно в каскадной модели.

Случайная кликовая модель простая и быстрая, но она не учитывает то, читал ли пользователь сниппет и заинтересовался ли он им. Это учитывается в позиционной кликовой модели, но в ней факт чтения сниппета документа не зависит от факта чтения сниппета документов, расположенных выше. Это, в свою очередь, учитывается в каскадной кликовой модели, но она не описывает сессии с количеством кликов более одного.

UBM (User Browsing Model) [14]

Эта модель является расширением позиционной кликовой модели (PBM) с добавлением некоторых элементов каскадной кликовой модели

(CM). Ее идея заключается в том, что вероятность прочесть сниппет не только от ранга документа r, но и от ранга предыдущих кликнутых документов r'.

$$P(E_r = 1 | C_1 = c_1, \dots, C_{r-1} = c_{r-1}) = \gamma_{rr'},$$

$$r' = \max\{k \in \{0, \dots, r-1\} : c_k = 1\}, \qquad c_0 = 1.$$

DCM (Dependent Click Model) [16]

Эта модель является расширением каскадной модели, предназначенной для сессии с несколькими кликами. Предполагается, что после клика пользователь продолжает рассматривать другие документы, т.е.:

$$P(E_r = 1 | C_{r-1} = 1) = \lambda_r,$$

где λ_r – параметр, зависящий от ранга документа.

Введем параметр S_r, обозначающий удовлетворенность пользователя после клика.

$$P(S_r = 1 | C_r = 0) = 0$$
$$P(S_r = 1 | C_r = 1) = 1 - \lambda_r$$
$$P(E_r = 1 | S_{r-1} = 1) = 0$$
$$P(E_r = 1 | E_{r-1} = 1, S_{r-1} = 0) = 1.$$

CCM (Click Chain Model) [23]

Эта модель является расширением DCM. В данной модели вводится релевантность документа для данного запроса - $\alpha_{uq} \in$ [0,1], а также параметры τ_1, τ_2, τ_3, характеризующие поведение пользователя на странице результатов поисковой выдачи. Пользователь просматривает сниппет, далее с вероятностью α_{uq} он кликает на неё, после чего с вероятностью τ_1 он завершает работу, а с вероятностью $1 - \tau_1$ продолжает просматривать сниппеты. После клика на документ вероятность просмотра остальных

документов определяется релевантностью просмотренного документа и значениями τ_2, τ_3.

$$C_r = 1 \Leftrightarrow E_r = 1 \; and \; A_r = 1$$

$$P(A_r = 1) = \alpha_{uq}$$

$$P(E_1 = 1) = 1$$

$$P(E_r = 1 | E_{r-1} = 0) = 0$$

$$P(E_r = 1 | E_{r-1} = 1, C_{r-1} = 0) = \tau_1$$

$$P(E_r = 1 | C_{r-1} = 1) = \tau_2 \left(1 - \alpha_{u_{r-1}q}\right) + \tau_3 \alpha_{u_{r-1}q}$$

SDBN (Simplified Dynamic Bayesian Network). [24]

Модель является упрощенным вариантом модели DBN, в которой предполагается, что пользователь читает сниппеты каждого документа. После клика на тот, который его заинтересовал, он прекращает поиск, если ему понравился прочтенный документ. Иначе с некоторой вероятностью он продолжит просматривать документы дальше. В SDBN же вероятность продолжить чтение сниппетов в случае неудовлетворенности пользователя равна 1.

$$C_r = 1 \Leftrightarrow E_r = 1 \; and \; A_r = 1$$

$$P(A_r = 1) = \alpha_{uq}$$

$$P(E_1 = 1) = 1$$

$$P(E_r = 1 | E_{r-1} = 0) = 0$$

$$P(S_r = 1 \,| C_r = 1) = \sigma_{u_r q}$$

$$P(E_r = 1 \,| S_{r-1} = 1) = 0$$

$$P(E_r = 1 \mid E_{r-1} = 1, S_{r-1} = 0) = \gamma = 1$$

Оценка параметров кликовой модели

Для того чтобы использовать кликовую модель, в первую очередь, ее необходимо обучить, то есть оценить ее параметры. Основными методами оценки параметров кликовой модели являются метод максимального правдоподобия и EM-алгоритм. Первый используется в тех случаях, когда модель работает только с наблюдаемыми случайными величинами (каскадная модель). EM-алгоритм применяется, если модель содержит скрытые переменные (позиционная кликовая модель).

Метод максимального правдоподобия

Пусть дана выборка:

$$X = (x_1, x_2, \dots, x_n) \sim p_\theta(x),$$

ее функция правдоподобия:

$$\mathcal{L}(\theta) = \prod_{i=1}^{m} p_\theta(x_i).$$

Необходимо найти значение θ, при котором функция правдоподобия будет принимать максимальное значение:

$$\theta_{ML} \stackrel{\text{def}}{=} arg\max_\theta \mathcal{L}(\theta).$$

θ_{ML}- параметры нашей модели.

EM-алгоритм

Идея алгоритма EM (expectation-maximization) [14] заключается в следующем: вводится вспомогательный вектор скрытых переменных G, обладающий свойствами:

1. Он может быть вычислен при известных значениях вектора параметров θ.

2. При известных значениях скрытых переменных поиск максимума правдоподобия становится проще.

Алгоритм состоит из двух повторяющихся шагов [25]:

1. На E-шаге (expectation) вычисляется ожидаемое значение вектора скрытых переменных G по текущему приближению вектора параметров θ.

2. На M-шаге (maximization) максимизируется правдоподобие и по текущим значениям векторов G и θ находится следующее приближение вектора θ.

Шаги E и M повторяются до тех пор, пока значения G и θ не стабилизируются.

Область применения EM-алгоритма достаточно обширна: кластеризация, дискриминантный анализ, обработка сигналов и изображений, восстановление пропусков в данных. В данной работе EM-алгоритм используется для разделения смеси распределений.

E-шаг. Пусть $p(x, \theta_j)$ плотность вероятности того, что объект x получен из j-й компоненты смеси. Тогда по формуле условной вероятности:

$$p(x, \theta_j) = p(x)p(\theta_j|x) = \omega_j p_j(x).$$

Пусть $g_{ij} \equiv p(\theta_j|x_i)$ – неизвестная вероятность того, что объект x_i получен из j-й компоненты смеси. Будем рассматривать эти величины как скрытые переменные. Обозначим $G = (g_{ij})$. Предполагается, что каждый из объектов может быть сгенерирован только одной компонентой. По формуле полной вероятности из этого следует условие нормировки для g_{ij}:

$$\sum_{j=1}^{k} g_{ij} = 1 \text{ для всех } i = 1, \dots, l.$$

Имея параметры компонент ω_j, θ_j, легко вычислить g_{ij} по формуле Байеса:

$$g_{ij} = \frac{\omega_j p_j(x_i)}{\sum_{s=1}^{k} \omega_s p_s(x_i)} \text{ для всех } i, j.$$

М-шаг. Этот шаг сводится к вычислению весов компонент ω_j через среднее арифметическое и оцениванию θ_j путем решения k независимых оптимизационных задач:

$$\omega_j = \frac{1}{m}\sum_{i=1}^{m} g_{ij}, \qquad j = 1, \dots, k.$$

$$\sum_{i=1}^{m} g_{ij} ln\varphi(x_i; \theta_j) \to \max_{\theta_j}, \qquad j = 1, \dots, k.$$

Оценка качества модели

Для того чтобы быть уверенными в том, что вводя более сложную модель, мы получаем лучшие результаты, необходима оценка качества модели. Одними из традиционных методов оценки качества кликовых моделей являются log-likelihood и perplexity. Они позволяют понять, насколько хорошо наша модель описывает поведение пользователя.

Log-likelihood

Логарифм правдоподобия считает вероятность того, что модель выдаст определенную последовательность кликов, т.е. оценивает, насколько хорошо модель считает условные вероятности. Для каждой тестовой поисковой сессии мы считаем, насколько модель соответствует действительности:

$$\mathcal{L}(s) = P_M\big(C_1 = {c_1}^{(s)}, \dots, C_n = {c_n}^{(s)}\big),$$

где ${c_u}^{(s)}$ – наблюдаемая величина, M – наша модель, s – поисковая сессия. Если предположить независимость поисковых сессий, можем посчитать логарифм правдоподобия:

$$\mathcal{LL}(M) = \sum_{s \in S} \log P_M(C_1 = {c_1}^{(s)}, \dots, C_n = {c_n}^{(s)})$$

$$\mathcal{LL}(M) = \sum_{s \in S} \sum_{u=1}^{n} \log P_M(C_u = {c_u}^{(s)} | C_{<u} = {c_{<u}}^{(s)})$$

Логарифм правдоподобия принимает неположительные значения и его значение для идеальной модели равно нулю.

Perplexity

В [11] было предложено использование перекрестной энтропии из теории информации [26] в качестве метрики кликовых моделей. Этот

показатель оказалось не так просто интерпретировать и широкого применения он не нашел. Вместо этого в [12] предложили использовать perplexity (перплексию). Этот метод оценивает качество подсчета полной вероятности.

$$p_u(M) = 2^{-\frac{1}{|S|}\sum_{s \in S}(\log_2 P_M(C_u{}^{(s)} = c_u{}^{(s)}))}.$$

Перплексия идеальной кликовой модели принимает значение 1, случайной кликовой модели с вероятностью 0.5 – значение 2. Следовательно, для реальных кликовых моделей перплексия принимает значения от 1 до 2.

Conditional perplexity

Перплексия оценивает, насколько хорошо модель предсказывает вероятность клика на документ. Логарифм правдоподобия оценивает, насколько хорошо модель предсказывает вероятность последовательности кликов. Условная перплексия же оценивает, насколько хорошо модель предсказывает вероятность клика на документ, учитывая клики на документы, расположенные выше на странице результатов поисковой выдачи и клики на документы из предыдущих запросов сессии:

$$\tilde{p}_r(M) = 2^{-\frac{1}{|S|}\sum_{s \in S}(c_r{}^{(s)} \log_2 \tilde{q}_r{}^{(s)} + (1 - c_r{}^{(s)}) \log_2 (1 - \tilde{q}_r{}^{(s)}))},$$

$$\tilde{q}_r{}^{(s)} = P_M\left(C_r = 1 \middle| C_1 = c_1{}^{(s)}, \dots, C_{r-1} = c_{r-1}{}^{(s)}\right).$$

Так же как и перплексия, данная метрика принимает значения от 1 до 2.

Кластеризация пользователей

Кластеризация – это разделение элементов множества на группы в зависимости от их схожести.

Кластеризация состоит из следующих этапов:

1. Выделение признаков. В первую очередь нужно определить, по каким признакам объекты будут делиться на группы. Для этого сначала необходимо выбрать все свойства, характеризующие рассматриваемые объекты. Далее из этих свойств выбираются наиболее важные, это позволяет ускорить процесс кластеризации и в некоторых случаях позволяет визуально оценить полученные результаты.

Объект для кластеризации представляется в виде вектора, координатами которого являются значения выбранных признаков.

2. Определение метрики.

3. Разбиение объектов на группы в соответствии с выбранными признаками.

4. Представление результатов.

Метод k-средних

Алгоритм k-means или k-средних – это простой алгоритм машинного обучения без учителя. Он группирует набор данных в заданное пользователем число кластеров (k).

Данный алгоритм состоит из следующих шагов:

1. Случайно выбираются k точек, они будут являться центрами кластеров.

2. Каждый объект относится к кластеру, центр масс которого является ближайшим.

3. Центры масс кластеров пересчитываются согласно текущему членству.

4. Необходимо вернуться к шагу 2, если не выполняется критерий остановки (изменение среднеквадратической ошибки или состава кластеров минимально).

Метод локтя

Метод k-средних кластеризует данные в заданные k кластеров, даже если это количество не является верным. Поэтому при использовании этого алгоритма нужен способ определения нужного количества кластеров.

Одним из методов определения количества кластеров является elbow method или метод локтя. Идея этого метода заключается в том, чтобы запустить метод k- средних для набора данных со значениями k в некотором диапазоне, например от 1 до 10. Для каждого значения k вычисляется сумма квадратов ошибок и строится график зависимости суммы квадратов ошибок от количества кластеров. Если представить, что на графике изображена рука, то точка, в котором график больше всего похож на локоть будет иметь наилучшее значение k. То есть мы хотим получить небольшое количество кластеров, чтобы при этом сумма квадратов ошибок также была небольшой.

Используемые данные

Информация о поведении пользователя при информационном поиске, его запросах, страницах, которые он посещал, хранится в поисковых логах. Для решения поставленной задачи использованы данные, предоставленные Яндексом для Personalized Web Search Challenge [27]. Логи были выбраны в связи с тем, что в них содержится вся необходимая для исследования информация, в том числе информация о пользователе. В связи с приватностью данных вся информация представлена в виде уникальных идентификаторов сессий, запросов и т.п.

Были рассмотрены 100000 поисковых сессий, это около 14000 уникальных пользователей.

Логи имеют следующий формат:

Данные о сессии:
SessionID TypeOfRecord Day USERID

Запросы:
SessionID TimePassed TypeOfRecord SERPID QueryID ListOfTerms ListOfURLsAndDomains

Клики:
SessionID TimePassed TypeOfRecord SERPID URLID

Здесь SessionID – уникальный идентификатор поисковой сессии. TypeOfRecord – тип записи в логах (M – поисковая сессия, Q – запрос пользователя, Q – клик на документ), Day – количество дней, прошедших с начала сбора логов, USERID – уникальный идентификатор пользователя, TimePassed – количество единиц времени, прошедших с начала поисковой сессии, SERPID – уникальный идентификатор страницы результатов поисковой выдачи, QueryID – уникальный идентификатор поискового запроса, ListOfTerms – список термов, из которых состоит запрос,

ListOfURLsAndDomains – упорядоченный (сверху вниз) список документов и соответствующих адресов, представленных на текущей странице результатов поисковой выдачи

Пример логов:

744899 M 23 123123123

744899 0 Q 0 192902 4857, 3847, 2939 632428,2384 309585,28374 319567,38724 6547,28744 20264,2332 3094446,34535 90,21 841,231 8344,2342 119571,45767

744899 1403 C 0 632428

Практические результаты

Из данных кликовых логов по каждому пользователю были найдены следующие его признаки: среднее количество кликов в сессию, среднее время между запросами в пределах сессии и среднее время между кликами в пределах сессии. Рассматривались и другие признаки пользователей, например, среднее количество кликов в сессию, но были исключены из рассмотрения, т.к. сильно коррелировали с кликами и никакой новой информации для кластеризации не несли.

Пример полученных данных:

UserId	Clicks	Time queries	Time clicks
0	2	0	594
1	2.03	1090.8	2318.63
2	1.67	187.2	182.78
3	1	0	12.33
4	7	366.94	927.24

Далее был построен граф на пользователях. Вершинами графа являются пользователи. Вершины соединяются ребром, если пользователи оба кликнули на один и тот же документ. Пользователи, чьи вершины оказались изолированными исключены из рассмотрения.

На графе были посчитаны следующие признаки: центральность по посредничеству, центральность по близости, центральность собственного вектора.

Пример полученных данных:

UserId	Betweenness centrality	Closeness centrality	Eigenvector centrality
0	0	0	0
1	112.662	0.001	0.001
2	0	0.001	0
3	0	0.001	0.011
4	0	0.001	3.315

С помощью метода k-средних с оптимальным количеством кластеров, равным 3, пользователи были разделены на три группы: 3674, 9833 и 889 пользователей. На графике изображена зависимость внутрикластерной суммы квадратов расстояний от количества кластеров, на которые разделяются пользователи:

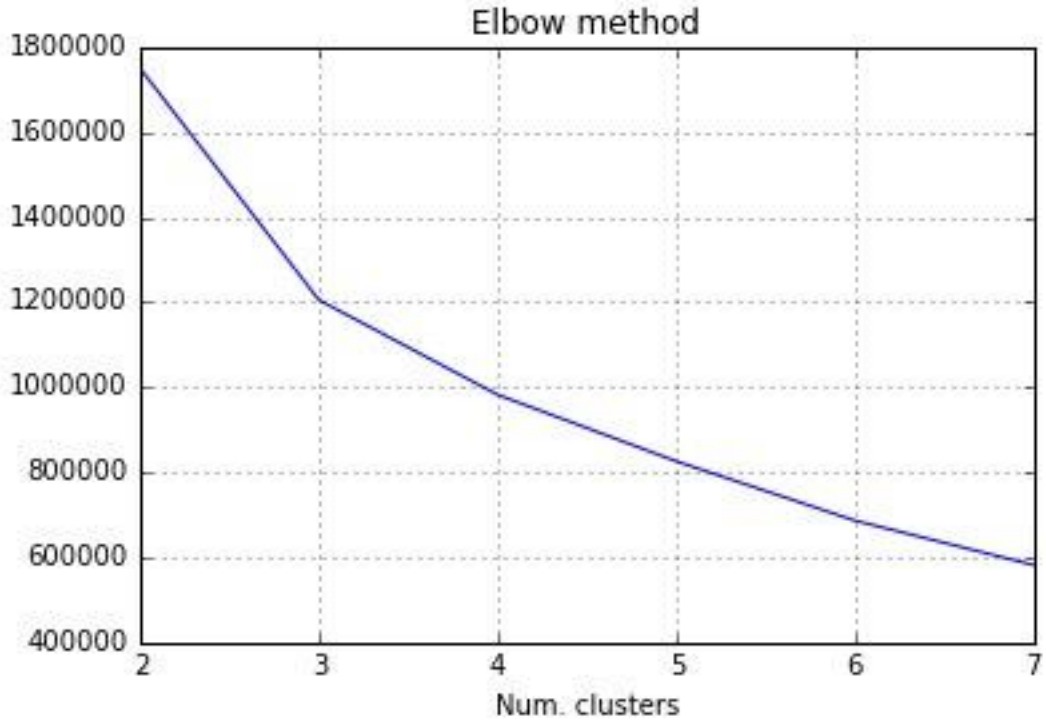

Рис. 1. Метод локтя

Пользователи в группах отличаются поисковой активностью и областью интересов. Пользователи из третьей группы, например, меньше

кликают, но тратят больше времени между кликами и запросами. В таблице приведены средние значения рассматриваемых признаков, полученных из кликовых логов для каждой группы:

Признак	Группа 1	Группа 2	Группа 3
Clicks	2.47	2.42	2.37
Time queries	307.73	307.94	357.20
Time clicks	684.14	638.35	728.91

На каждой из полученных групп пользователей и на всех пользователях сразу были обучены модели RCM, RCTR, DCTR, CM, DCM, SDBN. Для каждой обученной модели посчитаны метрики: логарифм правдоподобия, перплексия и условная перплексия.

Получены следующие значения логарифма правдоподобия:

Модель	Группа 1	Группа 2	Группа 3	Все пользователи
RCM	-0.2848	-0.2972	-0.2998	-0.3085
RCTR	-0.1977	-0.1885	-0.2055	-0.2517
DCTR	-0.3608	-0.2428	-0.2890	-0.6053
CM	-∞	-∞	-∞	-∞
DCM	-0.2843	-0.2466	-0.2473	-0.4353
SDBN	-0.2814	-0.2377	-0.2436	-0.4355

В таблице видно, что значение логарифма правдоподобия всех моделей, обученных на группах пользователей, значительно выше, чем у моделей, обученных на всех пользователях сразу. Так как CM не может обрабатывать сессии с количеством кликов больше одного, для нее в таблице указано значение $-\infty$. Наименьшее значение логарифма правдоподобия в случае использования одной модели для всех пользователей у DCTR, однако

при обучении DCTR на группах пользователей логарифм правдоподобия стал значительно больше. Это связано с тем, что при кластеризации пользователей учитывались документы, на которые они кликают, и так как модель основывается на паре запрос - документ, она лучше предсказывает клики в группах. Для всех групп пользователей наибольшее значение логарифма правдоподобия у модели RCTR, следовательно для рассматриваемых пользователей эта модель лучше других предсказывает последовательность кликов.

Полученные значения перплексии:

Модель	Группа 1	Группа 2	Группа 3	Все пользователи
RCM	1.4361	1.5134	1.4900	1.4176
RCTR	1.2454	1.2389	1.2547	1.3074
DCTR	1.4427	1.2850	1.3444	1.8326
CM	1.2276	1.1805	1.2225	1.3266
DCM	1.2583	1.2087	1.2431	1.3341
SDBN	1.2597	1.1978	1.2385	1.3532

Во всех моделях, кроме случайной (RCM), значение перплексии стало меньше. Особенно это заметно в случае с DCTR, что также связано с учетом документов, на которые кликают пользователи при кластеризации. Наилучшее значение перплексии на всех трех группах пользователей показывает каскадная модель (CM). Но т.к. данная модель не рассматривает сессии с количеством кликов больше одного, а у рассматриваемых пользователей среднее количество кликов в сессию больше двух, лучше воспользоваться другой моделью. Если исключить каскадную модель из рассмотрения, наименьшее значение перплексии для первой группы пользователей у модели RCTR, для второй группы у DCM, а для третьей

группы у SDBN. Если же обучать одну модель для всех пользователей, лучше всего воспользоваться RCTR.

Полученные значения условной перплексии:

Модель	Группа 1	Группа 2	Группа 3	Все пользователи
RCM	1.4361	1.5134	1.4900	1.4176
RCTR	1.2454	1.2389	1.2547	1.3074
DCTR	1.4427	1.2850	1.3444	1.8326
CM	∞	∞	∞	∞
DCM	1.3426	1.2944	1.2927	1.5558
SDBN	1.3386	1.2823	1.2879	1.5561

Здесь, так же как и в случае с перплексией, во всех моделях кроме случайной значения на кластерах стали меньше, что особенно заметно для DCTR. Для каскадной модели в таблице стоит ∞, так как каскадная модель не рассматривает сессии с количеством кликов больше одного. Наименьшее значение условной перплексии для всех рассматриваемых групп пользователей у модели RCTR, и вообще по данной метрике кликовые модели расположены в том же порядке, что и по логарифму правдоподобия. Условная перплексия не сильно отличается от логарифма правдоподобия и не дает принципиально новой информации о качестве рассматриваемых моделей.

Заключение

Сделано предположение, что использование нескольких кликовых моделей, обученных на кластерах пользователей, вместо одной, обученной на всех пользователях сразу, повышает точность предсказания кликов. Для подтверждения данного предположения:

1. Из кликовых логов для каждого пользователя найдены среднее количество кликов в сессию, среднее время между запросами и среднее время между кликами в пределах сессии.
2. Построен граф пользователей, в котором пользователи соединяются ребром, если они кликают на один и тот же документ.
3. Из графа пользователей найдены центральность по близости, по посредничеству и центральность собственного вектора.
4. На основе найденных признаков проведена кластеризация пользователей.
5. На полученных кластерах и на всех пользователях сразу обучено шесть кликовых моделей: RCM, RCTR, DCTR, CM, DCM, SDBN.
6. Вычислены метрики качества кликовых моделей (логарифм правдоподобия, перплексия, условная перплексия) для каждого случая.

Результаты показали, что на группах пользователей значение рассматриваемых метрик лучше, чем на всех пользователях сразу. Таким образом, точность предсказания кликов стала выше, предположение оказалось верным.

Список литературы

1. Joachims T. et al. Accurately interpreting clickthrough data as implicit feedback //ACM SIGIR Forum. – Acm, 2017. – Т. 51. – №. 1. – С. 4-11.
2. Гулин А. и др. Оптимизация алгоритмов ранжирования методами машинного обучения //Тр. Росс. сем. по оценке методов информационного поиска. СПб.: НУ ЦСИ. – 2009. – С. 163-168.
3. Liu T. Y. et al. Learning to rank for information retrieval //Foundations and Trends in Information Retrieval. – 2009. – Т. 3. – №. 3. – С. 225-331.
4. «Интернет – математика 2011». https://academy.yandex.ru/events/data_analysis/relpred2011/
5. Гуда С., Рябов Д. Отчет по конкурсу Relevance Prediction Challenge.
6. Figurnov M., Kirillov A. Linear combination of random forests for the Relevance Prediction Challenge //Proc. of Int. Conf. on Web Service and Data Mining workshop on Web Search Click Data. New York: ACM. – 2012. – С. 71-75.
7. Агеев М. С. Ранжирование документов по запросу на основе лога действий пользователей поисковой системы //вычислительные методы и программирование. – 2012. – Т. 13. – №. 4. – С. 559-571.
8. Hu B., Liu N. N., Chen W. Learning from click model and latent factor model for relevance prediction challenge //Proceedings of the Workshop on Web Search Click Data, WSDM. – 2012. – Т. 2012.
9. Agichtein E., Brill E., Dumais S. Improving web search ranking by incorporating user behavior information //Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2006. – С. 19-26.
10. Камалов М. В., Мартынов Р. С. Сравнительный анализ алгоритмов онлайн-обучения ранжированию поисковой выдачи //Процессы управления и устойчивость. – 2017. – Т. 4. – №. 1. – С. 382-388.

11. Craswell N. et al. An experimental comparison of click position-bias models //Proceedings of the 2008 international conference on web search and data mining. – ACM, 2008. – С. 87-94.
12. Dupret G., Murdock V., Piwowarski B. Web search engine evaluation using clickthrough data and a user model //WWW2007 workshop Query Log Analysis: Social and Technological Challenges. – 2007. – Т. 2.
13. Moffat A., Zobel J. Rank-biased precision for measurement of retrieval effectiveness //ACM Transactions on Information Systems (TOIS). – 2008. – Т. 27. – №. 1. – С. 2.
14. Dupret G. E., Piwowarski B. A user browsing model to predict search engine click data from past observations //Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2008. – С. 331-338.
15. Chapelle O., Zhang Y. A dynamic bayesian network click model for web search ranking //Proceedings of the 18th international conference on World wide web. – ACM, 2009. – С. 1-10.
16. Guo F., Liu C., Wang Y. M. Efficient multiple-click models in web search //Proceedings of the second acm international conference on web search and data mining. – ACM, 2009. – С. 124-131.
17. Diaz F. et al. Robust models of mouse movement on dynamic web search results pages //Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. – ACM, 2013. – С. 1451-1460.
18. Huang J. et al. Improving searcher models using mouse cursor activity //Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2012. – С. 195-204.
19. Xing Q. et al. Incorporating user preferences into click models //Proceedings of the 22nd ACM international conference on Conference

on Information & Knowledge Management. – ACM, 2013. – С. 1301-1310.

20. Manning C. D., Raghavan P., Schutze H. Introduction to Information Retrieval‖ Cambridge University Press, 2008 //Ch. – Т. 20. – С. 405-416.
21. Chuklin A., Markov I., Rijke M. Click models for web search //Synthesis Lectures on Information Concepts, Retrieval, and Services. – 2015. – Т. 7. – №. 3. – С. 1-115.
22. Joachims T. et al. Accurately interpreting clickthrough data as implicit feedback //ACM SIGIR Forum. – Acm, 2017. – Т. 51. – №. 1. – С. 4-11.
23. Guo F. et al. Click chain model in web search //Proceedings of the 18th international conference on World wide web. – ACM, 2009. – С. 11-20.
24. Chapelle O., Zhang Y. A dynamic bayesian network click model for web search ranking //Proceedings of the 18th international conference on World wide web. – ACM, 2009. – С. 1-10.
25. Воронцов К. В. Лекции по статистическим (байесовским) алгоритмам классификации //Bayes. pdf. – 2008.
26. Larson R. R. Introduction to information retrieval //Journal of the American Society for Information Science and Technology. – 2010. – Т. 61. – №. 4. – С. 852-853.
27. Personalized Web Search Challenge. https://www.kaggle.com/c/yandex-personalized-web-search-challenge/data