

Rambler pFound - метрика качества поиска

© С. В. Протасов, Д. В. Баранов

Рамблер Интернет Холдинг
s.protasov@rambler-co.ru, d.baranov@rambler-co.ru

Аннотация

В докладе описана метрика Rambler P_{found} , используемая для оценки качества поиска. Данная метрика используется для сравнения качества собственного поиска Rambler с другими поисковыми системами. Метрика $Rambler P_{found}$ не использует оценки ассессоров [1] и рассчитывается на основе поведенческих характеристик пользователей. В докладе проводится сравнительный анализ качества ранжирования Yandex, Google, Mail, Rambler, Bing¹.

1. Введение

Оценка качества поиска является одной из самых важных задач в работе поисковых систем. В общем случае качество поиска можно пытаться оценивать через экспертную оценку, базы маркеров, опросы пользователей, долю рынка и через другие “спорные” способы. Далее мы используем один из наименее “спорных” способов - это технологию сплитов (другие названия - бакеты или A/B тесты) с измерением оттока пользователей и метрики $pFound$

2. Сплиты

При принятии решения о внедрении нового алгоритма поиска требуется сделать вывод о росте качества поиска. Однако смена алгоритма поиска может сопровождаться рекламной кампанией, сменой дизайна и сезонными факторами. Рост доли рынка сразу после смены поискового алгоритма не означает улучшение качества. Чтобы устранить третьи факторы, мы используем технологию сплитов - новый поиск показывается некоторой части аудитории, которая выбирается случайно из общей выборки пользователей. Рост показателей на случайной выборке более надежно доказывает преимущества или недостатки альтернативного ранжирования.

¹Труды 13й Всероссийской научной конференции “Электронные библиотеки: перспективные методы и технологии, электронные коллекции” - RCDL’2011, Воронеж, Россия, 2011.

3. Отток/приток пользователей

“Отток пользователей на сплите” является основной метрикой. Отрицательный отток - это приток пользователей. Отток (Churn Rate) также используется при спорах об обосновании других метрик. Это означает, что вместо поиска корреляций с ассессорскими метриками (например [6]), мы опираемся на корреляцию с оттоком и используем его как “ground truth”.

Метрика оттока, как правило, требует длительного измерения - от недели до месяца, в зависимости от размера сплитов. Например, мы разбили 2 млн пользователей на две случайные выборки размером 1 млн. Тестовой части мы стали показывать новое ранжирование, базовой - оставили старое. Через месяц мы увидели, что у нового ранжирования 1150 тыс. пользователей, а у старого 1100 тыс. То есть присутствует сезонный рост аудитории, который меняет число пользователей как на тестовой, так и на базовой версии. Но кроме сезонного роста, мы имеем рост активности у пользователей нового ранжирования и этот рост существенно выше случайных флуктуаций. Мы делаем вывод, что новое ранжирование создает приток в $(1150-1100)/1100 = 4.5$ процента в месяц, что за год может довольно сильно изменить долю рынка.

Сплиты 1/2 обеспечивают максимальную точность, а использование небольшой части аудитории создает сильную погрешность в оценку оттока. Проблема состоит в том, что мы не всегда можем себе позволить тестировать новые идеи на 1/2 аудитории. Наша задача - в случае, когда измерить отток пользователей тяжело, найти такую метрику, которая бы была легко измеряема и одновременно точно оценивала будущий отток пользователей. Многие метрики, по отдельности, не обладают желаемым свойством точно предсказывать отток. Мы собираемся комбинировать метрики для лучшей точности и расширения границ использования. Мы также опишем ситуации, когда использование метрик не вполне корректно. Далее мы строим модель поиска и описываем метрику $pFound$, которая используется в тех случаях, когда измерение оттока затруднено. После этого мы изучаем возможность оценки $pFound$ через набор поведенческих метрик. В итоге мы получаем

некий “кликер” $pFound$, который не использует оценки ассессоров [1].

Наша модель достаточно проста и не учитывает поведение некоторых групп пользователей. Однако для основной массы запросов и пользователей модель достаточно хорошо моделирует наблюдаемые данные. Главное отличие нашей модели от моделей поиска, обсуждаемых в работах [3–5, 7, 8, 10], состоит в том, что наша модель разбивает качество поиска на две главные составляющие: качество сниппетов и качество ранжирования.

В настоящее время в основном исследуются 2 типа моделей поведения пользователей поиска. Первый тип - позиционные модели (position models). В такой модели вероятность клика $P_{cl} = F(rel, num)$ является функцией двух аргументов: релевантности результата rel и его номера в выдаче num . Другими словами, в таких моделях нет связи между различными результатами в одной выдаче. Второй тип - каскадные модели. В таких моделях вероятность клика зависит не только от релевантности результата и его номера в выдаче, но и от предыдущих результатов. Модели второго типа лучше отражают реальное поведение пользователя, так как учитывают взаимное влияние документов в выдаче. Наша модель близка к каскадной модели [7], но имеет дополнительные параметры усталости и качества сниппетов.

Многие работы, например [2, 11], моделируют сессии из нескольких запросов (чего мы не делаем) или вводят слишком много ненаблюдаемых факторов, что создает проблемы с настройкой моделей.

4. Модель поиска

Связь между поведенческими метриками и P_{found} предполагает некоторую модель, о которой будет рассказано далее.

Пользователь просматривает результаты сверху вниз. От 1 до 10. С некоторой вероятностью пользователь отказывается от просмотра выдачи. Например, он отвлекается на рекламу, на поисковый подмес или его цель была увидеть заголовок, сниппет или позицию сайта без клика (см. схему под таблицей 1). С вероятностью P_{look}^1 пользователь просматривает выдачу с целью сделать клик. С вероятностью P_{snip}^1 , (релевантность сниппета) пользователю нравится сниппет и он кликает на него. При этом кликабельность равна

$$(1) \quad CTR^j = P_{look}^j P_{snip}^j$$

Например, предположив что $P_{look}^1 = 0.8$, $P_{snip}^1 = 0.4$ для первого результата $CTR^1 = P_{look}^1 P_{snip}^1 = 0.8 * 0.4 = 0.32$, что совпадает с реально наблюдаемым $CTR_{real} = 0.32$. Таким образом, зная наблюдаемые CTR_{real}^j мы можем подбирать P_{snip}^j . При условии, что сниппет кликнут, с вероятностью $P(rel|cl)$ пользователю нравится результат, и

он удовлетворяется, переходит в состояние $Found$, останавливается и больше нигде не кликает. Вероятность этого события для данной позиции равна

$$(2) \quad P_f^j = P_{look}^j P_{snip}^j P^j(rel|cl)$$

Если пользователь еще не удовлетворен, то он переходит к следующему результату. Однако из-за усталости с некоторой вероятностью $P(break|notcl) = 0.07$ он останавливается неудовлетворенным в состоянии $NotFound$. Если пользователь сделал клик на текущий результат, то усталость несколько больше $P(break|cl) = 0.1$ (мы можем подбирать эти P для лучшего сходства с наблюдаемыми данными):

$$(3) \quad P_{look}^{j+1} = P_{look}^j ((1 - P_{snip}^j)(1 - P(break|notcl)) + P_{snip}^j (1 - P^j(rel|cl))(1 - P(break|cl)))$$

Например, взяв $P_{snip}^1 = 0.42$, $P^1(rel|cl) = 0.5$

$$P_{look}^2 = 0.8 * ((1 - 0.42) * (1 - 0.07) + 0.42 * (1 - 0.5) * (1 - 0.1)) \approx 0.58$$

Если пользователь дошел до 10 результата, то он переходит в состояние $NotFound$.

Мы можем просуммировать P_f и получить важную метрику P_{found} , которая, к сожалению, не наблюдается явно.

$$(4) \quad Rambler P_{found} \stackrel{\text{def}}{=} \sum_{j=1}^{10} P_f^j = \sum_{j=1}^{10} P_{look}^j P_{snip}^j P^j(rel|cl)$$

Для уменьшения числа скрытых параметров мы можем задать, что вероятность клика P_{snip} зависит от релевантности P_{rel} и качества сниппетов $P(snip|rel)$ и в случае, если результат релевантен, задать вероятность клика равной $P(snip|rel) = 0.7$, а если результат нерелевантен, то задать $P(snip|notrel) = 0.3$. В итоге мы можем рассчитывать P_{snip} через P_{rel} и у нас нет необходимости задавать P_{snip} :

$$(5) \quad P_{snip}^j = P(snip|rel)P_{rel}^j + P(snip|notrel)(1 - P_{rel}^j)$$

Например

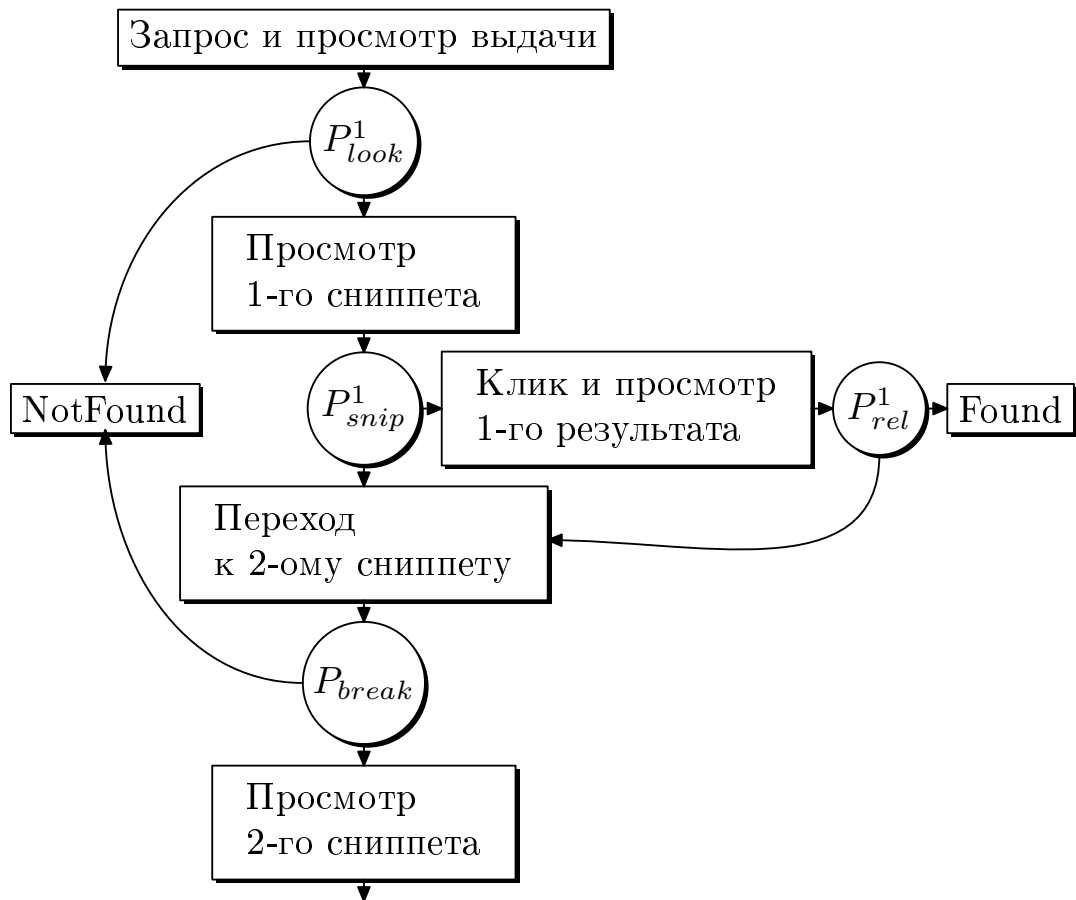
$$P_{snip}^1 = 0.7 * 0.3 + 0.3 * (1 - 0.3) = 0.42$$

j	P_{look}^j формула 3 %	P_{snip}^j формула 5 %	P_{rel}^j задаем %	$P^j(rel click)$ формула 6 %	CTR^j формула 1 %	CTR_{real}^j наблюдаем %	P_f^j формула 2 %	P_{found}^j формула 4 %
1	80.0	42.0	30	50.0	33.6	32	16.8	16.8
2	58.3	36.0	15	29.2	21.0	17	6.1	22.9
3	48.1	34.8	12	24.1	16.7	13	4.0	27.0
4	40.6	34.0	10	20.6	13.8	12	2.8	29.8
5	34.8	33.6	9	18.8	11.7	10	2.2	32.0
6	30.0	33.2	8	16.9	10.0	8	1.7	33.7
7	26.1	32.8	7	14.9	8.6	7.73	1.3	34.9
8	22.9	32.8	7	14.9	7.5	6.91	1.1	36.1
9	20.0	32.8	7	14.9	6.6	6.67	1.0	37.0
10	17.6	32.8	7	14.9	5.8	6.78	0.9	37.9

где j - номер позиции в выдаче,
 P_{look}^j - видимость j-ой позиции,
 P_{snip}^j - кликабельность с учетом сниппета,
 P_{rel}^j - релевантность j-ой позиции,
 $P^j(rel|click)$ - релевантность j-ой позиции при клике,
 CTR^j - кликабельность j-ой позиции с учетом видимости и релевантности,
 CTR_{real}^j - реальная кликабельность,
 P_f^j - вероятность найти ответ в данной позиции,
 P_{found}^j - накопленная вероятность найти ответ с первой до j-ой позиции.

Параметры модели:
 $P_{look}^1 = P_{lookf} = 80\%$
(видимость первого результата),
 $P(break|cl) = 10\%$
(усталость при клике),
 $P(break|notcl) = 7\%$
(усталость без клика),
 $P(snip|rel) = 70\%$
(кликабельность релевантного результата),
 $P(snip|notrel) = 30\%$
(кликабельность нерелевантного результата).

Таблица 1: Модель поиска



Наша модель подразумевает, что $P(snip)$ не может быть меньше $P(snip|notrel) = 0.3$, так как в самом худшем случае сниппет содержит релевантные ключевые слова из запроса.

Кроме этого по формуле Байеса

$$(6) \quad P^j(rel|cl) = \frac{P(snip|rel)P_{rel}^j}{P_{snip}^j} = \frac{P(snip|rel)P_{rel}^j}{P(snip|rel)P_{rel}^j + P(snip|notrel)(1 - P_{rel}^j)}$$

Например, задав $P_{rel}^1 = 0.3$

$$P^1(rel|cl) = \frac{0.7 * 0.3}{0.42} = 0.50$$

В Таблице 1 представлены значения P_{look} , P_{snip} , P_{rel} , $P(rel|cl)$, P_{found} , которые достаточно близко совпадают с реально наблюдаемыми из таблицы 2 и одновременно удовлетворяют формулам нашей модели поиска.

Значения параметров подбирались вручную, однако этот процесс поддается автоматизации.

Например, значения P_{rel}^j можно настроить так, чтобы максимально согласовать CTR^j и CTR_{real}^j в таблице 1.

Из формул 1, 6, 5 можно вывести P_{rel}^j :

$$(7) \quad P_{rel}^j = \frac{CTR_{real}^j / P_{look}^j - P(snip|notrel)}{P(snip|rel) - P(snip|notrel)}$$

Кроме этого, так как мы наблюдаем явно P_{c1t1} (доля запросов, где пользователь сделал ровно 1 клик по самой первой ссылке),

$$P_{c1t1} = P_{look}^j P_{snip}^j (P^j(rel|cl) + (1 - P^j(rel|cl)) * P(break|cl)) \approx P_f^1$$

а из формулы 6

$$(8) \quad P_{c1t1} \approx P_f^1 = P_{look}^1 P_{snip}^1 P^1(rel|cl) = P_{look}^1 P_{snip}^1 \frac{P(snip|rel)P_{rel}^1}{P_{snip}^1} = P_{look}^1 P(snip|rel)P_{rel}^1$$

то это дает нам возможность оценить, как меняется качество сниппетов $P(snip|rel)$ в предположении фиксированного P_{look}^1 и релевантности P_{rel}^1

$$(9) \quad P(snip|rel) \approx \frac{P_{c1t1}}{P_{look}^1 P_{rel}^1}$$

Значения P_{look}^1 , $P(snip|notrel)$, $P(snip|rel)$, $P(break|notcl)$, $P(break|cl)$ настраивались так,

	наблюдаем	модель P_{found}
dl	1.46	1.35
ds1c	2.14	1.90
P_{c1t1}	0.17	0.20
P_{c0}	0.336	0.287
P_{c1}	0.355	0.373
аср	3.26	3.25
p1cl	2.75	2.28

dl - длина сессии,

ds1c - длина кликов,

P_{c0} - вероятность отсутствия клика на поисковую выдачу,

P_{c1} - вероятность ровно 1-го клика,

аср - средняя позиция кликов,

p1cl - средняя позиция

первого клика,

P_{c1t1} - вероятность ровно 1-го клика

по первому результату.

Таблица 2: Расхождения между наблюдаемыми и модельными характеристиками

чтобы максимально согласовать аср, P_{c0} , P_{c1} , P_{c1t1} , p1cl, dl, ds1c в таблице 2.

Мы могли бы добиться большей близости, исключив навигационные запросы, но текущей точности нам вполне достаточно, чтобы найти связь между P_{found} и другими метриками.

Мы могли бы добавить больше скрытых параметров, и уменьшить расхождение в таблице 2, но основная цель - добиться близости большого количества наблюдаемых параметров через использование небольшого количества скрытых параметров.

В работах [7] для решения аналогичной задачи нахождения скрытых параметров используется минимизация кросс-энтропии, в работе [5] минимизируют расстояние Кульбака-Лейбнера. Общий обзор методов можно найти в труде [9]

В серии компьютерных экспериментов мы можем регулировать параметры релевантности P_{rel} и через логлинейную интерполяцию² обнаружить степенные зависимости.

Например при изменении P_{rel} на 10%, мы видим³, что P_{found} улучшается на 12%, а P_{c0} только на 2.3%.

То есть при улучшении (уменьшении) P_{c0} на 2.3% можно ожидать роста P_{found} на 12%.

Используя несколько точек, мы можем обнаружить, что между P_{c0} и P_{found} есть степенная зависимость со степенью около 5.

Мы можем сделать аналогичные оценки и для других поведенческих метрик.

²Для двух переменных x и y мы берем логарифмы $a = \log(x)$, $b = \log(y)$ и ищем зависимость вида $a = c0 + c1 * b$ подбирая $c0$ и $c1$ линейной регрессией на 5-20 точек

³мы могли бы получить данный результат аналитически через формулы модели, но удобнее просто сделать 10 млн. испытаний

$$\begin{aligned}
1 - P_{found} &= P_{notfound} \approx k_{acp}(acp)^2, \\
P_{notfound} &\approx k_{c0}(P_{c0})^5, \\
P_{notfound} &\approx k_{p1c}(P_{p1c})^4, \\
P_{notfound} &\approx k_{br}(P_{br})^2, \\
P_{found} &\approx k_{c1t1}(P_{c1t1})^{1.0} \\
P_{found} &\approx k_{c1}(P_{c1})^{1.6} \\
P_{found} &\approx k_{rel}(P_{rel})^{1.4}
\end{aligned}$$

Мы видим, что метрика P_{c1t1} менее чувствительна к изменениям релевантности, чем например P_{c0} .

Данные зависимости можно использовать для оценки P_{found} через поведенческие метрики.

5. Поведенческие метрики

В пределах одной сессии (одного запроса) существует некоторый набор краткосрочных метрик, которые могут использоваться вместо оттока пользователей, например:

P_{br} - вероятность возврата в поисковую выдачу после клика на результат⁴.

P_{otk} - вероятность просмотра результата менее чем 20 секунд и возвращения в поисковую выдачу.
аср - средняя позиция кликов.

P_{c0} - вероятность полного отсутствия кликов на поисковую выдачу.

Каждая метрика неявным образом связана с качеством поиска. Однако мы можем придумать ситуации, когда каждая метрика в отдельности не совсем корректно работает. Например, если у всех результатов очень хорошие сниппеты, но страницы выдают "404", то P_{c0} будет очень хорошим, а качество очень плохим.

А если сниппеты не содержат ключевых слов, а результаты очень хорошие, то некорректная работа будет у метрики P_{otk} .

Мы вводим глобальную метрику P_{found} , которая характеризует вероятность того, что пользователь нашел то, что искал. P_{found} можно оценивать через метрики P_{c0} , P_{br} , аср и другие.

Комбинируя метрики можно получить более удобную метрику, для которой придумать ситуации некорректной работы становится тяжелее.

В общем случае мы строим модель для $P_{notfound} = 1 - P_{found}$

$$(10) \quad P_{notfound} \approx k_{c0acpbr}(P_{c0}^5 acp^2 P_{br}^2)^{1/3}$$

предполагая, что P_{br} пропорционален доле отказных кликов P_{otk} , мы можем упростить модель до

$$(11) \quad P_{notfound} \approx k_{c0acpotk}(P_{c0}^5 acp^2 P_{otk}^2)^{1/3}$$

Похожую метрику мы и будем использовать далее, дав ей особое название $casrc0$:

⁴по сути $P_{br} = (1 - P(break|cl)) * (1 - P_{rel})$ по всем позициям в модели поиска

$$(12) \quad casrc0 = (P_{otk} * acp * P_{c0})^{1/3}$$

При тестировании на сплите 1/16 аудитории одного из алгоритмов ранжирования мы установили, что улучшение $casrc0$ на 3 процента уменьшает отток (то есть создает приток) на 3 процента в месяц по числу сессий и на 2 процента в месяц по числу пользователей. Данный сплит происходил по кукам, хотя дополнительные проверки можно сделать через сплиты по IP, e-mail или используя другие идентификаторы пользователей:

$$(13) \quad 3\%casrc0 \approx \frac{3\%}{month} ChurnRate_{sess}$$

$$(14) \quad 3\%casrc0 \approx \frac{2\%}{month} ChurnRate_{users}$$

Однако следует отметить, что отключив рекламу над поисковой выдачей, мы можем ухудшить среднюю позицию кликов (аср)⁵, ухудшить комбинированную метрику ($casrc0$), но получить приток пользователей, то есть наша модель обоснована только для измерения качества ранжирования при фиксированном дизайне и объеме рекламы. Мы можем усложнять модель - вводить разные группы пользователей (например группу нелояльных к рекламе, группу воспринимающих рекламу как часть поисковой выдачи, группу открывающих сразу несколько вкладок, группу просматривающих выдачу снизу вверх), группы запросов (например группу навигационных запросов, группу запросов, не требующих клика), но для для нашей текущей цели нам вполне достаточно простой модели.

В некоторых случаях нам понадобится усложнить модель. Например, мы обнаружили ситуацию, что, изменив ранжирование, мы увидели отток пользователей, а метрика $casrc0$ не изменилась. Это будет хороший повод использовать новую модель или комбинировать метрики другим способом.

В каких случаях нам будет достаточно более простой модели? В тех случаях, когда, используя более простую модель, мы можем получить метрику, так же хорошо предсказывающую отток пользователей.

Как мы можем упростить текущую модель⁶? Например, мы можем ввести один параметр, влияющий на усталость, один параметр, задающий качество сниппетов. Падение релевантности с ростом позиции также можно описать формулой с

⁵мы предполагаем, что аср рассчитывается по позициями результатов поиска и реклама не имеет своей позиции

⁶под более простой моделью мы понимаем модель с меньшим числом задаваемых параметров

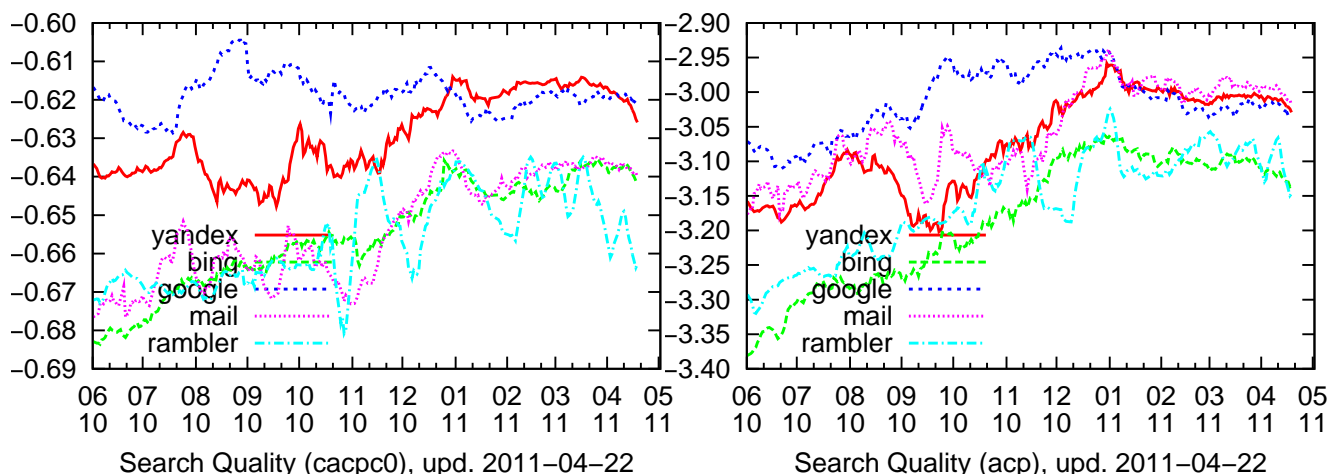


Рис. 1: $P_{found} \approx \text{casrc0}$, выше - лучше

Рис. 2: Средняя позиция кликов, выше - лучше

одним параметром. Расхождения в таблице 2 увеличатся, но у нас немного другая цель - найти комбинацию наблюдаемых метрик, хорошо предсказывающую отток пользователей.

6. Сравнение поисковиков

Не имея возможности измерить отток для других поисковиков, мы делаем оценки через метрику casrc0 (рисунок 1), которая очень сильно коррелирует с P_{found} и по сути отличается от неё лишь масштабированием.

Справочно (рисунки 2, 3, 4) мы приводим графики и для составляющих acp , P_{c0} , P_{otk} .

Каждый день с декабря 2010 по май 2011 года каждый сторонний поисковик получал от 2 до 6 тыс. случайно выбранных запросов. Данные сплиты происходили по уникальным запросам пользователей и имели размер около 1/1024 по трафику. Ответы поисковиков отрезались от фото-видео подмесов, и показывались в адаптированном виде пользователям Рамблера. Например, если сниппет имел дополнительную информацию о числе сообщений в форуме, то мы обрезали эту информацию, оставляя лишь основной текст сниппета.

Графики метрик строятся как скользящая средняя за 6 недель для всех поисковиков, кроме Рамблера. Погрешность при 100 тыс запросах составляет около 0.5 процентов для P_{c0} и 0.25 процента для acp и P_{otk} .

Для уменьшения погрешности оценок мы использовали дополнительные ограничения - не более 10 запросов на уникальный запрос и не более 10 уникальных запросов на пользователя в день⁷.

Для Рамблера график является скользящей средней за 1 неделю и рассчитывается на осно-

⁷ в том смысле, что десятый еще учитываем, а 11 уже не учитываем

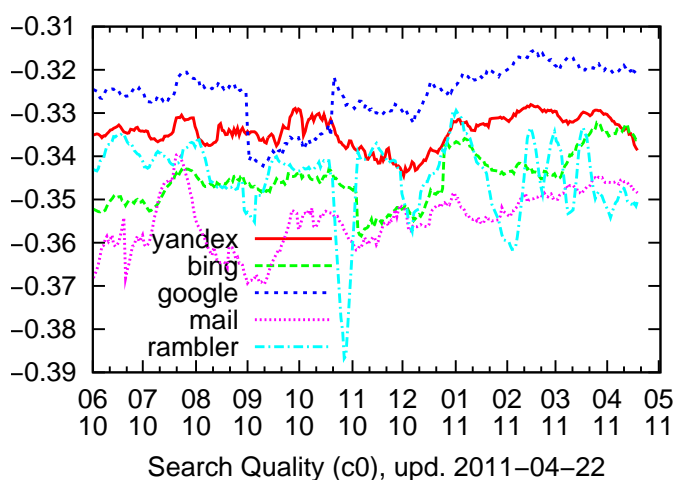


Рис. 3: Доля без кликов на выдачу, выше - лучше

ве более 20 млн. запросов⁸ с погрешностью менее 0.10 процента.

В работах [12, 13] приводится метод смешивания результатов на одной странице, что позволяет еще больше увеличить точность поведенческих метрик, однако в нашем случае для конкретного запроса все 10 результатов были от одной системы.

7. Выводы

Мы выяснили, что без учета мультимедийных подмесов⁹, качество ранжирования разных поисковиков нестабильно и отличается незначительно. Разница между лучшим и худшим находится в пределах 10 процентов по метрике casrc0 , что может приводить к оттоку пользователей на уровне около 6 процентов в месяц. Данное качество оце-

⁸ после фильтрации 200 млн. запросов, содержащих в основном запросы роботов

⁹ не стоит также забывать про гео-ранжирование и различные расширенные сниппеты

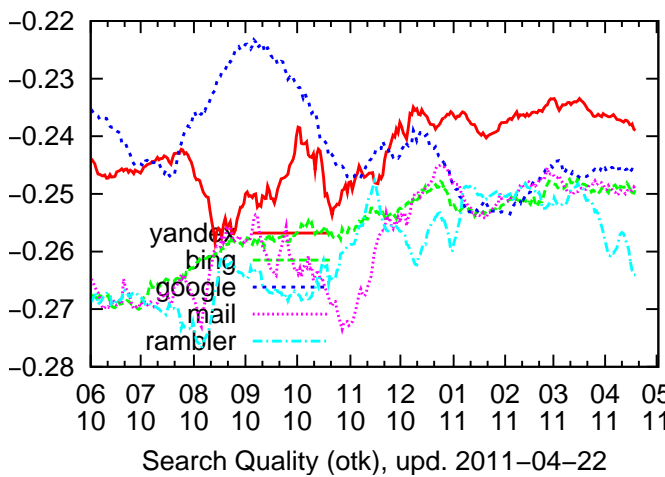


Рис. 4: Доля отказных кликов, выше - лучше

нивается через удовлетворенность 5 млн. пользователей Рамблера¹⁰, что не является самой правильной выборкой для сравнения поисковиков, но вполне подходит для обсуждения вопроса “какой поисковик лучше всего подошел бы для пользователей Рамблера?”¹¹.

Согласно нашей модели поиска вероятность P_{found} того, что пользователь найдет ответ в случайном запросе, составляет около 40 процентов.

В наших планах стоит дальнейшее развитие моделей, метрик и сравнение XML версий поисковиков¹².

Список литературы

- [1] Гулин Андрей, Карпович Павел, Расковалов Денис и Сегалович Илья. Оптимизация алгоритмов ранжирования методами машинного обучения. // Яндекс на РОМИП 2009
- [2] И. Е. Куралёнок, М. А. Скачков, О. В. Басков. Экономия времени как мера качества поисковой системы. // RCDL 2010
- [3] E. Agichtein, E. Brill, S. Dumais and R. Ragno. Learning user interaction models for predicting web search result preferences. // SIGIR 2006
- [4] Ben Carterette, Rosie Jones. Evaluating Search Engines by Modeling the Relationship Between Relevance and Clicks. // Advances in Neural Information Processing Systems, 2007
- [5] O. Chapelle. A Dynamic Bayesian Network Click Model for Web Search Ranking. // WWW 2009
- [6] O. Chapelle. Expected Reciprocal Rank for Graded Relevance // Conference on Information and Knowledge Management (CIKM) 2009

¹⁰ по данным TNS Russia Web Index

¹¹ В июне 2011 года Рамблер поставил XML версию Яндекса

¹² выдача XML Яндекса и Google не совсем совпадает с версией, которая показывается обычным пользователям

- [7] Nick Craswell, Onno Zoeter, Michael Taylor and Bill Ramsey. An experimental comparison of click position-bias models. // WSDM 2008
- [8] Doug Downey, Susan Dumais and Eric Horvitz. Models of Searching and Browsing: Languages, Studies, and Applications. // IJCAI 2007
- [9] Gelman A. , Carlin J. B. , Stern H. S. Bayesian data analysis // CRC Press
- [10] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais and Thomas White. Evaluating implicit measures to improve web search. // ACM Transactions on Information Systems 2005
- [11] T. Lau and E. Horvitz. Patterns of Search: Analyzing and Modeling Web Query Refinement. // UM 1999
- [12] Filip Radlinski, Madhu Kurup, Thorsten Joachims. How Does Clickthrough Data Reflect Retrieval Quality. // 17th ACM conference on Information and knowledge management, 2008
- [13] Y. Yue, Y. Gao, O. Chapelle, Y. Zhang and T. Joachims. Learning More Powerful Test Statistics for Click-Based Retrieval Evaluation. // SIGIR 2010

Rambler pFound - Search Quality Metric
 © S. V. Protasov, D. V. Baranov
 Rambler Media

This paper describes a new search quality metric Rambler P_{found} . Unlike Yandex pFound, this metric is not an editorial metric. Instead, it is based on user clicks and was used for comparison between Rambler and other commercial search engines. We show some plots for Yandex, Google, Mail, Rambler and Bing.