Вероятностная модель для сглаживания целевых метрик качества ранжирования*

Н. А. Волков[1,2]***, М. Е. Жуковский***[1,2]
nikitavolkov@yandex-team.ru; zhukmax@yandex-team.ru
[1]Московский физико-технический институт, Россия, г. Долгопрудный, Институтский пер., 9
[2]Яндекс, Россия, г. Москва, ул. Льва Толстого, 16

Задача информационного поиска — находить релевантные документы по запросу пользователя. Одной из важнейших задач информационного поиска является задача ранжирования результатов поиска. В настоящее время широко распространен подход получения функции релевантности с помощью методов машинного обучения на основе обучающей выборки. Для оценки качества принято использовать метрики качества, например, pFound. Однако большинство из них являются дискретными, что усложняет, например, отбор признаков. Цель данной работы — получить гладкий аналог дискретной метрики. Рассматриваются несколько вариантов определения гладкой метрики, лучший из которых экспериментально определяется по критериям гладкости и похожести на дискретную метрику. Критерий похожести, в отличие от критерия гладкости, численно выразить довольно трудно ввиду большого множества различных случаев поведения метрик, поэтому от численного описания пришлось отказаться. По результатам многочисленных экспериментов была найдена оптимальная модель гладкой метрики.

Ключевые слова: *задача ранжирования; метрики качества; сглаживание метрик; дискретные метрики; метрика pFound*

DOI: 10.21469/22233792.2.4.04

1 Введение

1.1 Задача ранжирования

Пусть D — некоторая коллекция текстовых документов, а Q — множество запросов. Обозначим D_q неупорядоченный набор документов, потенциально релевантных запросу $q \in Q$. Основная задача ранжирования — упорядочить документы внутри D_q по убыванию степени их релевантности запросу, т. е. более релевантные документы должны иметь более высокий ранг.

Если документ d_1 релевантнее документа d_2 по запросу $q \in Q$, то будем писать $(q, d_1) \triangleright (q, d_2)$. Таким образом определяется порядок на парах запрос–документ. Отметим, что этот порядок определен только внутри одного конкретного запроса.

Чтобы упорядочить пары запрос-документ, будем искать *функцию релевантности* $\alpha(q, d)$, которая удовлетворяет условию:

$$(q, d_1) \triangleright (q, d_2) \quad \Leftrightarrow \quad \alpha(q, d_1) > \alpha(q, d_2).$$

Разумеется, для того чтобы найти функцию релевантности, необходимо заранее организовать описанный порядок на парах запрос–документ. Для этой цели специалисты (работники коммерческих поисковых систем), называемые асессорами, для некоторых запросов размечают пары запрос–документ, определяя для них *оценку* релевантности $y(q, d)$.

*Работа выполнена при финансовой поддержке компании Яндекс.

На основе этой асессорской выборки и будем строить функцию α, которая приближает неизвестное $y(q,d)$ на всем множестве пар запрос–документ. Стоит также отметить, что в настоящее время в коммерческих поисковых системах процедура оценивания релевантности ассессорами достаточно хорошо формализована в виде списка многочисленных правил.

По этим данным составляется обучающая выборка $X_{\text{train}} = \{x_i = (q_i, d_i), y_i\}_{i=1}^{n}$, с помощью которой строится композиция деревьев $\alpha(x) = \sum_{i=1}^{N} a_i t_i(x)$, например λ-MART (см., например [1]).

1.2 Метрики качества

Различные ранжирования пар запрос–документ сравниваются при помощи метрик качества ранжирования. Для измерения качества ранжирования традиционно (традиция задана конференцией TREC — Text REtrieval Conference) используются меры точности, например метрики MAP (mean average precision), ERR (expected reciprocal rank) и NDCG (normalized discount cumulative gain).

Приведем описание метрики pFound, которая разработана в Яндексе на базе эмпирических исследований поведения пользователей [2, 3]. Она аппроксимирует вероятность того, что пользователь удовлетворится результатами поиска по запросу. В данном случае предполагается, что $y(q,d) \in \{0,1,2,3,4\}$. На основе экспериментов была получена зависимость от оценки асессоров вероятности $p(q,d)$ того, что пользователь удовлетворится документом d по запросу q. Оценкам 0, 1, 2, 3 и 4 соответствуют вероятности 0, 0,07, 0,14, 0,41 и 0,61. Также эмпирически было получено, что вероятность того, что пользователь прекратит поиск, равна $P_{\text{break}} = 0{,}15$ на каждом документе. Пусть набор документов $\{d_q^{(i)}\}_i$ упорядочен в соответствии со значением $\alpha(q,d)$. Тогда вероятность того, что пользователь при просмотре поисковой выдачи дойдет до i-го документа, равна

$$P_i = \left(1 - p\left(d_q^{(i-1)}\right)\right)\left(1 - P_{\text{break}}\right) P_{i-1}, \;\; P_1 = 1.$$

Окончательно:

$$\text{pFound}(n) = \sum_{i=1}^{n} P_i \; p\left(d_q^{(i)}\right).$$

2 Предпосылки к задаче

Выше был описан метод ранжирования, а также метрики качества. Для хорошего ранжирования используется набор признаков того, что данный документ релевантен запросу. При разработке нового признака возникает естественный вопрос, влияет ли он на качество функции релевантности.

2.1 Метод отбора признаков

Пусть U_0 — существующее множество признаков, а U_1 — оно же с добавлением новых признаков, т. е. $U_0 \subset U_1$. Разобьем выборку $X = \{x = (q,d), y\}$, размеченную асессорами, на две непересекающиеся части, которые обозначим X_{train} и X_{test}, и будем называть их обучающей и тестовой выборками соответственно. Обозначим F_{X_0,U_i} модель, обученную по выборке X_{train} на множестве признаков U_i, т. е. по векторам $x = (f(q,d))_{f \in U_i}$. Значением метрики MSE (mean-squared error) по множеству признаков U_i будем называть величину

$$\text{MSE}\left(U_i\right) = \sqrt{\frac{1}{|X_{\text{test}}|} \sum_{x \in X_{\text{test}}} \left(F_{X_0,U_i}(x) - p(x)\right)^2},$$

где $p(q, d)$ — вероятность того, что пользователь удовлетворится документом d по запросу q.

Для вынесения решения о том, что добавление признаков дает прирост в качестве, можно, например, проверить статистическую гипотезу:

$$H_0 : \mathrm{MSE}\,(U_1) - \mathrm{MSE}\,(U_0) > 0.$$

Проверка такой гипотезы происходит с помощью N-кратного разбиения вида $X = X^n_{\text{train}} \cup X^n_{\text{test}}$, $n = 1, \ldots, N$, и подсчета для каждого разбиения числа $\mathrm{MSE}^n(U_i)$ для наборов признаков U_0 и U_1, получая тем самым парную выборку. По полученной выборке строится тест Уилкоксона [4].

Недостатком данного подхода является тот факт, что метрика MSE в силу своего определения предназначена для оценки точности предсказания абсолютных значений функций релевантности $\alpha(q, d)$, а не порядка, который они определяют.

2.2 Дискретность метрик

Выше были описаны недостатки использования метрики MSE. Но почему же тогда она используется при отборе признаков, а не специальная метрика, предназначенная для оценки качества ранжирования, например pFound? Дело в том, что такие метрики дискретны, в силу того что они зависят только от порядка. Поясним эту проблему рассмотрением двух случаев.

Случай 1. Пусть α' и α'' — некоторые функции релевантности, причем их значения на документах d_1 и d_2 по запросу q отличаются не сильно, но так, что порядок этих документов меняется. Формально это можно записать следующими условиями:

$$\alpha'(q, d_1) = \alpha'(q, d_2) + \varepsilon_1\,;$$

$$\alpha''(q, d_1) = \alpha''(q, d_2) - \varepsilon_2\,,$$

где $\varepsilon_1 > 0$ и $\varepsilon_2 > 0$ малы. Поскольку порядок меняется, то меняется и значение метрики.

Случай 2. Пусть теперь α' и α'' — функции релевантности, построенные так, что порядок документов d_1 и d_2 по запросу q не меняется. Однако разница значений функций релевантности для этих документов меняется сильно. Формально это можно записать следующими условиями:

$$\alpha'(q, d_1) = \alpha'(q, d_2) + \beta_1\,;$$
$$\alpha''(q, d_1) = \alpha''(q, d_2) + \beta_2\,,$$

где $\beta_2 > 0$ и $\beta_1 > 0$, но при этом $\beta_1 - \beta_2$ велико. Поскольку порядок не меняется, то не меняется и значение метрики.

Получаем, что в одном случае значения функций релевантности меняются не сильно, но это влечет изменение порядка, а значит, и изменение значения метрики, а в другом случае значения функций релевантности меняются сильно, но при этом такое изменение не влечет изменение порядка, а значит, и не меняется значение метрики. Теперь ясно, почему хорошие метрики, такие как NDCG и pFound, являются дискретными.

3 Постановка задачи и первые попытки решения

В предыдущем разделе были описаны минусы дискретных метрик, которыми являются все основные метрики качества ранжирования. В связи с этим возникает естественное желание получить хорошую гладкую метрику. На самом деле, получить гладкую метрику

может быть не так сложно. Гораздо сложнее добиться того, чтобы она была действительно хорошей в следующем смысле: она должна быть похожа на сглаживаемую дискретную метрику.

В силу всего вышесказанного формулируем задачу следующим образом: *получить гладкую метрику, похожую на хорошую дискретную метрику.* Стоит отметить, что данная формулировка не дает четкого определения слов «гладкая» и «похожая». Они будут пояснены при проведении экспериментов позже.

Существующие способы сглаживания метрик и их недостатки

В статьях [5, 6] рассматривается задача замены дискретных метрик некоторыми гладкими функциями потерь, которые называются суррогатными. К сожалению, такой подход не дает решения данной задачи: ведь он основан только на совпадении решений задач оптимизации дискретной метрики и суррогатной функции потерь. Мы же требуем «похожесть» гладкой метрики в гораздо более широком смысле. В статье [7] решается задача непосредственного сглаживания метрики ERR.

Все известные авторам попытки разработки желаемой метрики основывались на введении некоторой вероятностной модели так, чтобы ранжирование получалось случайным. Тогда в соответствии с этой вероятностной моделью эксперименты можно проводить несколько раз и усреднять результаты, получая тем самым сглаженное значение метрики.

Первые попытки введения вероятностной модели использовали метод перестановки документов в запросе на основе модели Plackett–Luce [8]. Случайные перестановки создавались с помощью некоторого распределения с учетом весов документов — значений функции релевантности на текущей итерации. В этом случае числа, которые выдает модель, определяют не релевантность документа, а вероятность того, что этот документ может быть более релевантным данному запросу. Само ранжирование генерируется с помощью полученного распределения. Таким образом, ожидаемый pFound можно представить как среднее значение по всем полученным перестановкам документов, отвечающих запросу.

Однако от этого способа пришлось отказаться, так как в таком случае для подсчета нужно выполнить $N!$ перестановок, если рассматривать только N наиболее релевантных документов, что является достаточно долгой процедурой. Если проводить оценку методом Монте Карло, то полученная оценка оказывалась слишком неточной — при одинаковых экспериментах значения метрики различались слишком сильно, из-за чего случайные факторы проходили тест отбора признаков.

4 Метрика EpFound

В предыдущем разделе были перечислены недостатки введения вероятностной модели на результатах ранжирования. Из-за этих недостатков был выбран другой метод введения вероятностной структуры — вероятностная модель вводится на самих документах. Это в некотором смысле естественно, так как, например, при смене базы D значения признаков могут несколько измениться.

4.1 Базовая модель

Пусть X — некоторое множество пар запрос–документ. Будем считать его случайным с неизвестным распределением P, из которого сгенерировано именно это множество. Это распределение будем называть истинным распределением множества X. Например, P — это распределение множества пар запрос–документ для поисковой машины yandex.ru, а для yandex.com распределение может быть другим.

Разделим множество X на две непересекающиеся части X_{train} и X_{test}, которые будем называть обучающей и тестовой выборками соответственно. Обозначим $\text{pFound}(F(X_{\text{train}}), X_{\text{test}})$ значение pFound на выборке X_{test} по формуле F, обученной по выборке X_{train}. Для того чтобы избавиться от дискретности метрики, можно посчитать ее математическое ожидание $\mathsf{E}_{\mathsf{P}}\text{pFound}(F(X_{\text{train}}), X_{\text{test}})$, где E_{P} — математическое ожидание в предположении, что множества имеют распределение P.

Посчитать это математическое ожидание нам не удастся по двум причинам. Во-первых, распределение может быть чересчур сложным. Однако эту проблему можно решить, сделав оценку математического ожидания с помощью метода Монте Карло

$$\widehat{\mathsf{E}}_{\mathsf{P}}\text{pFound}\left(F(X_{\text{train}}), X_{\text{test}}\right) = \frac{1}{M}\sum_{m=1}^{M}\text{pFound}\left(F(\xi^m_{\text{train}}), \xi^m_{\text{test}}\right),$$

где $(\xi^m_{\text{train}}, \xi^m_{\text{test}})$ сгенерировано из распределения P. Вторая проблема заключается в том, что само истинное распределение P неизвестно. Далее в работе будем пытаться построить оценку P^* для распределения P.

Допустим, мы уже построили некоторую оценку P^* для распределения P. Тогда мы можем оценить неизвестное нам математическое ожидание $\mathsf{E}_{\mathsf{P}}\text{pFound}(F(X_{\text{train}}), X_{\text{test}})$ по распределению P математическим ожиданием $\mathsf{E}_{\mathsf{P}^*}\text{pFound}(F(X_{\text{train}}), X_{\text{test}})$ по распределению P^*, которое, в свою очередь, оценим с помощью метода Монте Карло:

$$\widehat{\mathsf{E}}_{\mathsf{P}^*}\text{pFound}(F(X_{\text{train}}), X_{\text{test}}) = \frac{1}{M}\sum_{m=1}^{M}\text{pFound}(F(\xi^m_{\text{train}}), \xi^m_{\text{test}}),$$

где $(\xi^m_{\text{train}}, \xi^m_{\text{test}})$ сгенерировано из распределения P^*.

Теперь займемся построением оценки P^* распределения P. Для этого примем несколько предположений о распределении P^*, которые описаны ниже.

Обозначим $x = (q, d)$ некоторую пару запрос–документ. Будем обозначать также признаки пары запрос–документ x как f_k. Для каждого признака f_k определим соответствующие ей бинарные признаки g_{kj} по правилу $g_{kj} = I\{f_k(x) > b_{kj}\}$, где $\{b_{k1}, \ldots, b_{kB_k}\}$ — упорядоченный по возрастанию набор точек разбиения, $B_k \geqslant 0$.

Предположение 1. Будем искать оценку P^*, предполагая независимость в совокупности исходных признаков документа.

Предположение 2. Для облегчения процедуры получения оценки P^* примем также заведомо неверный факт о независимости в совокупности бинарных компонент $g_{k1}, \ldots, g_{kn}$ одного небинарного признака f_k. Заметим, что при таком предположении нарушается свойство бинарных признаков, которое будем называть *свойством монотонности*. Это свойство заключается в том, что для любых $a \neq (0, \ldots, 0, 1, \ldots, 1)$ выполнено равенство $\mathsf{P}\left(g_{k1}(x) = a_1, \ldots, g_{kn}(x) = a_n\right) = 0$. Однако в предположении независимости бинарных компонент одного небинарного признака получаем:

$$\mathsf{P}^*\left(g_{k1}(x) = a_1, \ldots, g_{kn}(x) = a_n\right) = \mathsf{P}^*\left(g_{k1}(x) = a_1\right)\cdots\mathsf{P}^*\left(g_{kn}(x) = a_n\right),$$

где правая часть не равна нулю, например, если каждый бинарный признак не является вырожденной случайной величиной.

Создание распределения

Определим окрестность $U_N(x)$ — множество из N ближайших соседей пар запрос–документ x в бинаризованном пространстве по L_1-метрике, которая определяется как

$$\rho(x, x') = \sum_k \sum_j |g_{kj}(x) - g_{kj}(x')| .$$

По умолчанию сама пара запрос–документ x не включается в окрестность $U_N(x)$.

Обозначим $p_{kj}(x) = \mathsf{P}(g_{kj}(x) = 1)$. В силу указанных выше предположений набор оценок $\{p^*_{kj}(x)\}_{x,k,j}$ таких чисел задает оценку распределения P^* для бинаризованного множества. Определим их с помощью пар запрос–документ из окрестности $U_N(x)$:

$$p^*_{kj}(x) = \mathsf{P}^*(g_{kj}(x) = 1) = \frac{1}{N} \sum_{x' \in U_N(x)} g_{kj}(x').$$

Эта оценка является оценкой максимального правдоподобия для $p_{kj}(x)$ по выборке документов из окрестности $U_N(x)$.

Генерация множеств

Пусть $X = \{g_{kj}(x)\}_{x,k,j}$ — некоторое бинарное множество и $\mathsf{P}^*(X) = \{p^*_{kj}(x)\}_{x,k,j}$ — оценка распределения множества. Задав распределение, можно сгенерировать из него новые множества. Новое случайное множества $\xi = \{\xi_{kj}(x)\}_{x,k,j}$ генерируется по правилу $\xi_{kj}(x) \sim \mathrm{Bern}\left(p^*_{kj}(x)\right)$. Обозначим эту процедуру $\xi \sim \mathrm{Bern}\left(\mathsf{P}^*(X)\right)$.

Определение EpFound

Как и раньше, будем обозначать X_{train} и X_{test} обучающую и тестовую выборки, и $\mathrm{pFound}\left(F(X_{\text{train}}), X_{\text{test}}\right)$ — значение pFound на X_{test} по формуле F, обученной на X_{train}. Как было сказано выше, будем определять EpFound по правилу:

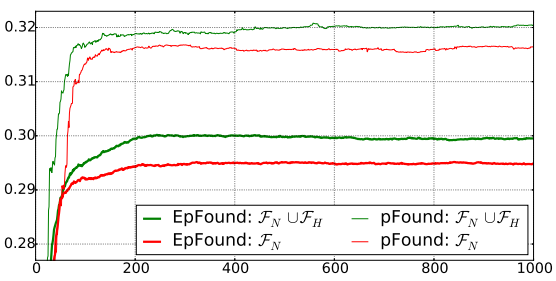
$$\mathrm{EpFound} = \frac{1}{M} \sum_{m=1}^{M} \mathrm{pFound}\left(F(\xi^m_{\text{train}}), \xi^m_{\text{test}}\right),$$

где $\xi^m_{\text{train}} \sim \mathrm{Bern}\left(\mathsf{P}^*(X_{\text{train}})\right), \xi^m_{\text{test}} \sim \mathrm{Bern}\left(P^*(X_{\text{test}})\right)$.

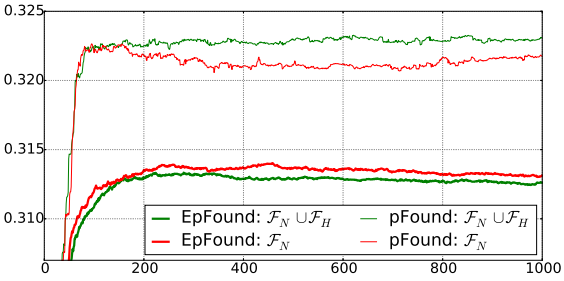
4.2 Проблемы базовой модели

Определенный выше вариант определения EpFound имеет несколько проблем. Самой основной из них является проблема различного поведения pFound и EpFound. Эта проблема может проявляться различными способами. Самый частый случай заключается в различном поведении этих метрик при использовании хостовых признаков и при их отсутствии. Хостовыми называем те признаки, которые полностью определяются хостом страницы и одинаковы для всех страниц данного хоста.

Пусть $\mathcal{F}_N$ и $\mathcal{F}_H$ — множества нехостовых и хостовых признаков соответственно. Одно из проявлений упомянутой проблемы заключается в том, что если pFound, посчитанный при использовании множеств с признаками $\mathcal{F}_N$, принимает значения *меньшие*, чем при использовании таких же множеств, но с признаками $\mathcal{F}_N \cup \mathcal{F}_H$, то EpFound наоборот, при использовании множеств с признаками $\mathcal{F}_N$ может принимать *большие* значения, чем при использовании таких же множеств, но с признаками $\mathcal{F}_N \cup \mathcal{F}_H$. Эта проблема показана на рис. 1, *а*, на котором видно, что относительный порядок для этих двух наборов признаков меняется при замене pFound на EpFound.



(*а*) Набор данных 1



(*б*) Набор данных 2

Рис. 1 График значения метрик pFound и EpFound в зависимости от количества деревьев в композиции; усреднение по 7 итерациям

Другое проявление проблемы хостовых признаков заключается в следующем. В некоторые моменты времени при построении композиции деревьев значение pFound возрастает с ростом количества деревьев в композиции, если используются признаки $\mathcal{F}_N \cup \mathcal{F}_H$, и убывает, если используются только признаки $\mathcal{F}_N$. EpFound же в таких случаях вообще практически не меняет значения, хотя при этом и принимает большие значения при использовании хостовых признаков, чем без них.

Эта проблема показана на рис. 1, *б*. Видно, что значение pFound после добавления 350-го дерева до добавления 400-го дерева в композицию деревьев, которые составляют обученную формулу, немного возрастает при использовании хостовых признаков и убывает без их использования. Однако у EpFound такого эффекта нет.

4.3 Способы решения проблем

Варианты усреднения

Выше был рассмотрен только один случай усреднения, а именно: мы определяли EpFound по правилу:

$$\text{EpFound} = \frac{1}{M}\sum_{m=1}^{M} \text{pFound}\left(F\left(\xi^m_{\text{train}}\right), \xi^m_{\text{test}}\right),$$

где $\xi^m_{\text{train}} \sim \text{Bern}\left(P^*(X_{\text{train}})\right)$; $\xi^m_{\text{test}} \sim \text{Bern}\left(P^*(X_{\text{test}})\right)$; X_{train} и X_{test} — обучающая и тестовая выборки; $\text{pFound}\left(F(X_{\text{train}}), X_{\text{test}}\right)$ — значение pFound на X_{test} по формуле F, обученной на X_{train}. Тем самым при усреднении генерируются как обучающая, так и тестовая выборки. Этот вариант будем называть первой моделью и обозначать его EpFound_1.

Здесь возникает вопрос: нужно ли генерировать как обучающую, так и тестовую выборки? Может быть, стоит генерировать только одну из них, а в качестве второй брать исходную? Во втором варианте будем генерировать только тестовые выборки, тем самым обучая формулу только один раз:

$$\text{EpFound}_2 = \frac{1}{M}\sum_{m=1}^{M} \text{pFound}\left(F(X_{\text{train}}), \xi^m_{\text{test}}\right),$$

где $\xi^m_{\text{test}} \sim \text{Bern}\left(P^*(X_{\text{test}})\right)$.

В третьем варианте будем генерировать только обучающие выборки, тем самым применяя одну тестовую выборку к набору обученных формул:

$$\mathrm{EpFound}_3 = \frac{1}{M}\sum_{m=1}^{M} \mathrm{pFound}\left(F(\xi^m_{\mathrm{train}}), X_{\mathrm{test}}\right) ,$$

где $\xi^m_{\mathrm{train}} \sim \mathrm{Bern}\left(\mathsf{P}^*(X_{\mathrm{train}})\right)$.

Формально эти формулы можно получить, взяв в качестве оценки распределения P^* вырожденное распределение для обучающей или тестовой выборки.

Окрестности

В базовом варианте мы предполагали, что окрестность $U_N(x)$ — множество из N ближайших соседей пар запрос–документ x в бинаризованном пространстве по L_1-метрике, причем сама пара запрос–документ x не включается в окрестность. Возможен также вариант, при котором пара запрос–документ x включается в окрестность $U_N(x)$ наравне с остальными документами.

Отдельно будем рассматривать вариант, при котором сама пара запрос–документ x не включается в окрестность $U_N(x)$, но в вычислении вероятностей берется с весом $w \in (0, 1)$:

$$p^*_{kj}(x) = w\ g_{kj}(x) + \frac{1-w}{N-1}\sum_{x' \in U_{N-1}(x)} g_{kj}(x').$$

Биномиальная модель

Также рассматривалась модель построения оценки P^*, для которой выполняется свойство монотонности. В этом случае мы генерировали не сами бинарные признаки, а номер последнего бинарного признака для данного исходного признака, которая принимает значение ноль. Распределение этой величины считалось биномиальным, параметр которого оценивался по окрестности аналогичным способом. Однако в данных экспериментах такой метод не привел к успеху. Условия экспериментов аналогичны описанным ниже.

5 Эксперименты

В предыдущих разделах была сформулирована исследуемая задача поиска гладкой метрики, рассказано, почему нас не удовлетворяют существующие способы сглаживания метрик, а также приведены теоретические описания вариантов новой сглаженной метрики. В данном разделе опишем эксперименты, которые были проведены для поиска наилучшего варианта из предложенных.

5.1 Описание данных

Данные для проведения экспериментов были получены с помощью поисковой машины yandex.ru, а также асессоров, которые разметили документы. Данные содержат 100 026 пар запрос–документ, из которых 75 021 пар составляли обучающую выборку, а остальные 25 005 пар — тестовую выборку. Обучающая выборка при этом состояла из 7181 запросов, а тестовая — из 2035 запросов. Разбиение выборок проводилось по запросам случайным образом, т. е. для каждого запроса все пары запрос–документ с этим запросом попадали либо в обучающую выборку, либо в тестовую.

В данных каждая пара запрос–документ описывается меткой релевантности, уникальным идентификатором запроса, а также набором признаков. Все признаки либо бинарные —- принимают значения из $\{0, 1\}$, либо непрерывные, значения которых нормированы на интервал $[0, 1]$. Метки релевантности принимают значения во множестве $\{0, 1, 2, 3, 4\}$.

Эксперименты проводили на трех различных множествах признаков. Первые два были получены случайным выбором 100 нехостовых признаков и 30 хостовых, третий — случайным выбором 100 нехостовых признаков и 14 хостовых.

5.2 Как измерялось качество сглаживания

Напомним сначала основную цель исследования: *получить гладкую метрику, похожую на* pFound. Как было отмечено выше, данная формулировка не дает четкого определения слов «гладкая» и «похожая». Теперь пришло время несколько формализовать данные понятия.

Критерий гладкости. Усредненная метрика имеет гладкий график зависимости ее значения от количества деревьев. Степень гладкости удалось выразить численно. Опишем процедуру подсчета степени гладкости зависимости.

Пусть $a = (a_1, \ldots, a_n)$ — некоторая зависимость. Определим *сглаженное значение* $\widehat{a}_i$ как значение линейной регрессии в точке i, построенной по точкам $(i - r, a_{i-r}), \ldots, (i + r, a_{i+r})$, причем из набора исключены s минимальных и s максимальных значений. Тогда *степенью гладкости* называется величина

$$S(a) = \frac{10^{-7}}{\mathrm{MSE}\,(a, \widehat{a})}.$$

В данных экспериментах использовались $r = 20$ и $s = 5$. Домножение на константу 10^{-7} сделано для того, чтобы значения гладкости получались приятными на глаз, в данных экспериментах это обычно значения от 0 до 100. Гладкость тем выше, чем больше значение степени гладкости для зависимости.

Критерий похожести. Усредненная метрика ведет себя «похоже» на *pFound*. К сожалению, данный критерий формализовать достаточно сложно. Поясним немного эту проблему.

Любая формализация данного критерия должна предполагать сравнение изменения трендов значений pFound и EpFound в зависимости от количества деревьев. Например, если pFound начинает расти, то и EpFound должен расти, и наоборот. Однако в случае, представленном на рис. 1, мы вовсе не требуем такую зависимость. Дело в том, что данное поведение можно еще интерпретировать как увеличение разности значения pFound при использовании хостовых признаков и без них. Такой же эффект мы хотим получить от EpFound. Это означает, что нас даже устроит случай, при котором абсолютные значения EpFound уменьшаются, но их разность возрастает.

Подобных частных случаев похожего поведения можно придумать достаточно много, поэтому какие-либо попытки формализации данного критерия скорее всего будут заточены под такие частные случаи.

5.3 Выбор лучшей модели

По критерию похожести лучше всего оказывается *простая модель с третьим типом усреднения*, в которой генерируется только обучающая выборка. Однако она лишь немного уступает упомянутой выше биномиальной модели по критерию гладкости и имеет значения гладкости от 20 до 25, в то время как биномиальная модель имеет значения гладкости от 35 до 40. Поскольку большей гладкости можно добиться увеличением количества формул при усреднении, данный недостаток можно считать незначительным. В свою очередь биномиальная модель оказывается намного хуже по критерию похожести. Таким образом, можно заключить, что эта простая модель с третьим типом усреднения является наилучшей. Теперь же нужно для нее подобрать оптимальное значение веса.

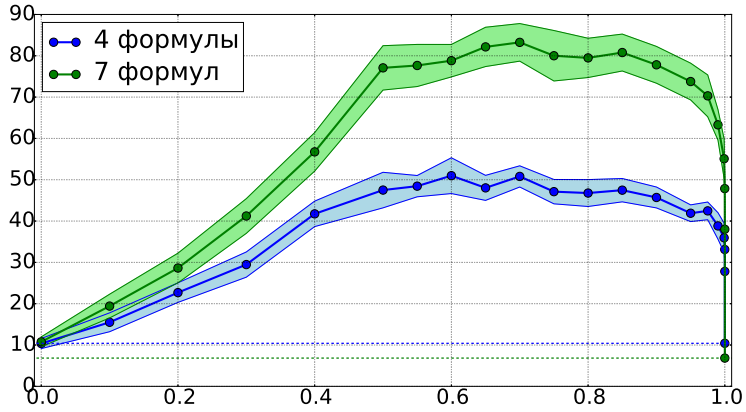


Рис. 2 График доверительных интервалов уровня доверия 0,95 для среднего значения степени гладкости

Было замечено (результаты экспериментов не приводятся, поскольку они заняли бы много места), что критерии гладкости и похожести часто дают схожие результаты, т. е. если одна модель хорошая с точки зрения критерия похожести, то она будет хорошей и по критерию гладкости, и наоборот. Таким образом, для нахождения его веса можно найти оптимальное значение по критерию гладкости, а затем проверить по критерию похожести.

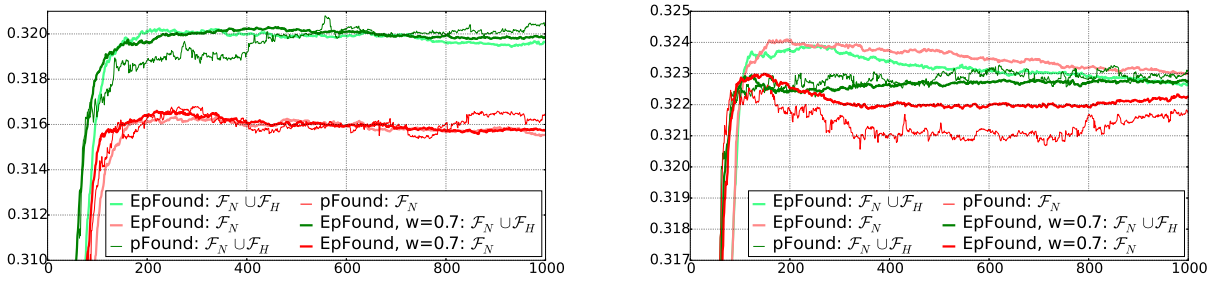
Для нахождения оптимального значения веса по критерию гладкости была проделана следующая процедура для некоторых значений весов. Для значения веса w было посчитано 10 значений степени гладкости EpFound на множестве 2 без использования хостовых признаков и 10 значений с их использованием. Полученные значения являются выборкой из некоторого неизвестного распределения. Из предположения, что эти значения имеют нормальное распределение, по ним был построен доверительный интервал уровня доверия 0,95 для параметра сдвига. График полученных доверительных интервалов приведен на рис. 2.

Доверительные интервалы были построены только для множества 2. Дело в том, что такая процедура является достаточно ресурсоемкой задачей. Для построения этого графика потребовалось обучить 2942 моделей, каждая из которых представляет из себя композицию из 1000 деревьев.

Из графика видно, что степень гладкости монотонно возрастает при возрастании веса от 0 до 0,5. Для значении веса от 0,5 до 0,9 значение степени гладкости меняется несильно. При приближении веса к 1 происходит резкое уменьшение значений степени гладкости. Вес $w = 1$ соответствует обычному pFound. Из графика видно, что оптимальное значение веса находится в окрестности 0,7, поэтому будем считать оптимальным вес $w = 0{,}7$.

Посмотрим теперь, насколько хорошо модель с весом $w = 0{,}7$ удовлетворяет критерию похожести.

На множестве 1 значения и поведения метрик EpFound и pFound практически совпадают, в частности схожа динамика перед добавлением 400-го дерева. Значение степени гладкости равно 77,4. Некоторое отличие значений pFound от EpFound наблюдается только в случае использования хостовых признаков от добавления 100-го дерева до добавления 400-го дерева. Однако поскольку до 100-го дерева и после 400-го дерева значения



(*а*) Набор данных 1 (*б*) Набор данных 2

Рис. 3 График значения метрик pFound и EpFound в зависимости от количества деревьев в композиции; усреднение по 7 итерациям, модель усреднения 3, вес 0,7

обоих метрик практически совпадают, можно сделать предположение о некотором недостатке самого pFound, который устраняется с помощью метрики EpFound.

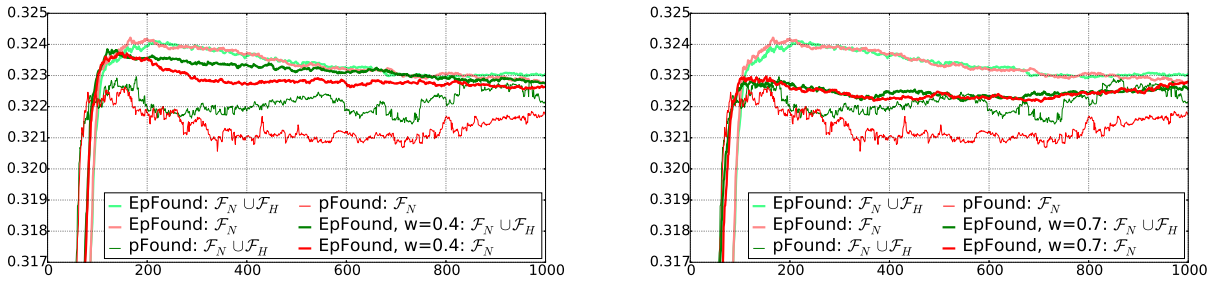
На множестве 2 поведение метрики EpFound достаточно хорошо повторяет поведение метрики pFound. При использовании модели с оптимальным весом значения метрики EpFound возрастают при достаточно большом количестве деревьев при использовании только нехостовых признаков, что полностью соответствует поведению метрики pFound (рис. 3). Значение степени гладкости равно 75,9.

Несмотря на то что метрика EpFound с оптимальным весом оказалась лучше других моделей по всем критериям на множествах 1 и 2, на множестве 3 она уступает моделям с другими весами. Дело в том, что ее значения при использовании хостовых признаков и при их отсутствии практически совпадают. Лучшим же по критерию похожести оказывается модель с весом $w = 0{,}4$. Значения степеней гладкости для весов 0,4 и 0,7 равны соответственно 60,0 и 77,4. Такая модель на множествах 1 и 2 оказывается хуже модели с весом $w = 0{,}7$ по критерию похожести.

6 Выводы и планы на будущее

На основании всех проведенных экспериментов можно сделать следующие выводы.

1. Была поставлена задача получить гладкую метрику, похожую на pFound. Для ее решения были предложены несколько методов, которые различаются способом генерации выборки, типом усреднения, а также видом рассматриваемых окрестностей точек. По совокупности критериев чаще всего лучшей оказывается простая модель EpFound с третьим типом усреднения, в которой генерируется только обучающая выборка. В связи с этим ее можно считать метрикой, которая лучше всего решает поставленную задачу.
2. Вес точки при подсчете вероятности является гиперпараметром модели. Для разных данных по различным критериям может быть свое значение оптимального веса. Наилучшие результаты получаются при значении веса от 0,4 до 0,9.
3. Качество оптимальной метрики как по критерию похожести, так и по критерию гладкости, сильно зависит от используемого множества.
4. Биномиальная модель, в которой сгенерированные значения бинарных признаков удовлетворяют свойству монотонности, имеет преимущество по критерию гладкости, но сильно проигрывает по критерию похожести. В связи с этим биномиальная модель не является оптимальной.



(*а*) Вес 0.4 (*б*) Вес 0.7

Рис. 4 График значения метрик pFound и EpFound в зависимости от количества деревьев в композиции, усреднение по 7 итерациям, модель усреднения 3, набор данных 3

5. Вторая модель усреднения, в которой генерируется только тестовая выборка, чаще оказывается хуже остальных, в особенности по критерию похожести. Поэтому данную модель использовать не рекомендуется.
6. Увеличение числа соседей с $N = 10$ до 30 не приводит к какому-либо улучшению качества. Для получения хороших результатов стоит использовать $N = 10$ соседей.

Разумеется, работа оставляет открытыми некоторые естественные вопросы. Перечислим те из них, ответы на которые мы собираемся получить, продолжив исследования.

1. Стоит проверить гипотезу о том, может ли метрика EpFound лучше измерять качество ранжирования, чем метрика pFound. Имеется в виду, например, ситуация, изображенная на рис. 4 для модели с весом 0,7 при использовании хостовых признаков. Возможно, различие в поведении pFound и EpFound объясняется плохим поведением самого pFound до добавления 400-го дерева в композицию.
2. Кроме того, мы планируем найти зависимость оптимального веса по критерию гладкости от количества формул для усреднения. Исходя из графика доверительных интервалов можно рассматривать гипотезу о том, что оптимальное значение веса меняется в зависимости от количества формул в усреднении.
3. Наконец, планируется по заданному множеству данных найти критерий выбора веса. Как следует из результатов экспериментов, оптимальное значение веса может быть разным для различных моделей.

Литература

[1] *Burges C. J. C.* From RankNet to LambdaRank to LambdaMART: An overview. Microsoft Research Technical Report, 2010. `http://research.microsoft.com/pubs/132652/msr-tr-2010-82.pdf`.

[2] *Воронцов К. В.* Методы обучения ранжированию. Лекции ШАД. `http://www.machinelearning.ru/wiki/images/8/89/Voron-ML-Ranking-slides.pdf`.

[3] *Гулин А., Карпович П., Расковалов Д., Сегалович И.* Оптимизация алгоритмов ранжирования методами машинного обучения // РОМИП-2009. `http://romip.ru/romip2009/15_yandex.pdf`.

[4] *Wilcoxon F.* Individual comparisons by ranking methods // Biometrics Bull., 1945. Vol. 1. No. 6. P. 80–83. `http://sci2s.ugr.es/keel/pdf/algorithm/articulo/wilcoxon1945.pdf`.

[5] *Buffoni D., Calauzenes C., Gallinari P., Usunier N.* Learning scoring functions with order-preserving losses and standardized supervision // NIPS, 2011. http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Buffoni_447.pdf.

[6] *Calauzenes C., Usunier N., Gallinari P.* On the (non-)existence of convex, calibrated surrogate losses for ranking // NIPS, 2012. http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2012_0118.pdf.

[7] *Shia Y., Karatzogloub A., Baltrunasb L., Larsona M., Hanjalic A.* xCLiMF: Optimizing expected reciprocal rank for data with multiple levels of relevance // Conference on Recommender Systems, 2013.

[8] *Guiver J., Snelson E.* Bayesian inference for Plackett–Luce ranking model // 26th Conference (International) on Machine Learning Proceedings, 2009. http://research.microsoft.com/pubs/81134/plackett.pdf.

Поступила в редакцию 03.09.2016

On a probabilistic model for smoothing discrete ranking quality metrics*

***N. A. Volkov*[1,2] *and M. E. Zhukovskii*[1,2]**

nikitavolkov@yandex-team.ru; zhukmax@yandex-team.ru

[1]Moscow Institute of Physics and Technology
9 Institutskiy Per., Dolgoprudny, Moscow Region, Russia

[2]Yandex, 16 Leo Tolstoy Str., Moscow, Russia

The information retrieval aim is to find relevant documents by given user's query. One of the main information retrieval task is the ranking problem of searching results. Currently, the method of getting relevant function with machine learning methods based on train sample is widely distributed. The quality of ranking is assessing with quality metrics, for example, pFound. However, most of them are discrete, it is difficult for feature selection. The aim of this work is to get a smoothing analogue of the discrete metric. Some different definitions of smoothing metric have been considered, the best of which has been found by the smoothing criteria and the criteria of a similarity to discrete metric by means of experiments. In comparence with smoothing criteria, it is difficult to obtain numerical description of the similarity criteria because there are a lot of different cases of metric behavior. So, the authors decided not to use numerical description. As a result of the large number of experiments, an optimal model of a smoothing metric has been got.

Keywords: *ranking problem; quality metrics; smoothing metrics; discrete metrics; metric pFound*

DOI: 10.21469/22233792.2.4.04

References

[1] Burges, C. J. C. 2010. From RankNet to LambdaRank to LambdaMART: An overview. Microsoft Research Technical Report. Available at: http://research.microsoft.com/pubs/132652/msr-tr-2010-82.pdf (accessed December 29, 2016).

*The research was supported by Yandex.

[2] Vorontsov, K. V. Metody obucheniya ranzhirovaniyu [Learning to rank]. Yandex School of Data Analysis lectures. Available at: http://www.machinelearning.ru/wiki/images/8/89/Voron-ML-Ranking-slides.pdf (accessed December 29, 2016).

[3] Gulin, A., P. Karpovich, D. Raskovalov, and I. Segalovich. 2009. Optimizatsiya algoritmov ranzhirovaniya metodami mashinnogo obucheniya [Optimizing the ranking algorithms by machine learning methods]. *ROMIP-2009*. Available at: http://romip.ru/romip2009/15_yandex.pdf (accessed December 29, 2016).

[4] Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics Bull.* 1(6):80–83. Available at: http://sci2s.ugr.es/keel/pdf/algorithm/articulo/wilcoxon1945.pdf (accessed December 29, 2016).

[5] Buffoni, D., C. Calauzenes, P. Gallinari, and N. Usunier. 2011. Learning scoring functions with order-preserving losses and standardized supervision. *NIPS*. Available at: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Buffoni_447.pdf (accessed December 29, 2016).

[6] Calauzenes, C., N. Usunier, and P. Gallinari. 2012. On the (non-)existence of convex, calibrated surrogate losses for ranking. *NIPS*. Available at: http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2012_0118.pdf (accessed December 29, 2016).

[7] Shia, Y., A. Karatzogloub, L. Baltrunasb, M. Larsona, and A. Hanjalic. 2013. xCLiMF: Optimizing expected reciprocal rank for data with multiple levels of relevance. *Conference on Recommender Systems*.

[8] Guiver, J., and E. Snelson. 2009. Bayesian inference for Plackett–Luce ranking model. *26th Conference (International) on Machine Learning Proceedings*. Available at: http://research.microsoft.com/pubs/81134/plackett.pdf (accessed December 29, 2016).

Received September 3, 2016