

С.И. НИКОЛЕНКО, А.А. ФИШКОВ  
**SCM: НОВАЯ ВЕРОЯТНОСТНАЯ МОДЕЛЬ  
ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ  
ИНТЕРНЕТ-ПОИСКА**

---

*Николенко С.И., Фишков А.А.* **SCM: новая вероятностная модель поведения пользователей интернет-поиска.**

**Аннотация.** Модели поведения пользователей – одно из основных направлений исследований направленных на улучшение интернет-поиска; такие модели обычно основаны на графических вероятностных моделях и обучаются из логов пользовательских действий (click logs). В работе вводится новая модель поведения пользователей – SCM (session click model, клик-модель сессии). Мы показываем, что новая модель проще для вывода, но в практических приложениях даёт результаты лучше, чем существующие модели.

**Ключевые слова:** модели поведения пользователя, графические вероятностные модели, expectation propagation.

*Nikolenko S.I., Fishkov A.A.* **SCM: a new user behaviour model for web search.**

**Abstract.** User behaviour models represent one of the most important directions of studies aimed at enhancing web search; such models are usually based on probabilistic graphical models and are learned from logs of user's actions (click logs). In this work we introduce a new user behaviour model called SCM (session click model). We show that the new model has simpler inference algorithms but in practice gives better results than existing models.

**Keywords:** user behaviour models, probabilistic graphical models, expectation propagation.

---

**1. Введение.** Поисковые системы – важнейшая часть современного интернета. На заре развития сети поиск производился по специальным каталогам, содержащим ссылки на существующие ресурсы, но на сегодняшний день их число настолько велико, что требуются специальные полностью автоматизированные системы для поиска в Интернете; поисковые системы сегодня – неотъемлемая часть информационного общества [27]. Примерами таких систем могут служить Яндекс, Google, Yahoo и Bing.

Взаимодействие с такой системой происходит следующим образом: пользователь вводит запрос, после чего поисковая система представляет ему список документов, которые наилучшим образом соответствуют его запросу. Средняя длина современного поискового запроса составляет всего 2,5 слова (по данным исследования [37]). Такому запросу может соответствовать огромное число

документов из базы поисковой системы, и пользователь при всём желании не сможет просмотреть их все. Важной задачей становится ранжирование документов по степени их соответствия запросу, т.е. их релевантности.

Для решения задачи ранжирования, а также для оценки качества поиска используются экспертные оценки релевантности документа запросу. На основе информации о содержимом документа, тексте запроса и других признаках этой пары ранжирующая функция строится с помощью методов машинного обучения [2–4, 7, 10]. Получение экспертных оценок – процесс трудоёмкий и дорогостоящий. Этот подход не лишён недостатков: эксперты могут расходиться во мнениях, да и сами мнения экспертов могут меняться со временем в результате появления новых страниц или других событий, происходящих в мире или в сети.

Известно, что современные поисковики ведут историю посещений, запросов и переходов для всех пользователей. Интерес представляет использование этой информации для ранжирования и улучшения поиска. Эта информация доступна в больших объёмах, не требует больших затрат на получение и содержит оперативную информацию для огромного числа документов. Впервые эта идея была предложена в работе [15]. Авторами были проведены эксперименты с участием людей и сформулированы основные положения об их поведении при взаимодействии с поисковой системой. Среди последующих работ наиболее сильные результаты были получены в области построения вероятностных моделей поведения пользователя [5, 8, 9, 13, 26]. Существенным недостатком большинства из них является то, что они ограничиваются лишь одним запросом пользователя к системе, тогда как в реальной ситуации пользователь, просматривая результаты, не раз переформулирует свой запрос, прежде чем найдёт требуемый документ.

В настоящей работе мы предлагаем модель поведения пользователя, учитывающую его многократные запросы к поисковой системе. Кроме того, мы проводим исследование использования оценок релевантности, полученных с помощью этой модели для улучшения ранжирования, на анонимизированных данных, опубликованных компанией «Яндекс» [38].

## **2. Определения, обозначения и графические вероятностные модели.**

### **2.1. Графические вероятностные модели**

Многие современные вероятностные модели поведения пользователей представляются в виде *графических вероятностных моделей*. Графические вероятностные модели представляют совместное распределение случайных величин, участвующих в вероятностной модели решаемой задачи, в виде графа, рёбра которого (точнее, их отсутствие) кодируют условные независимости между величинами. Можно выделить следующие основные достоинства этого подхода.

1. Визуальное представление структуры вероятностной модели является наглядным, способствует быстрой разработке, изменению и представлению новых моделей.
2. Важные свойства модели, например условная независимость входящих в неё случайных величин, сводятся к свойствам графа, соответствующего данной модели.
3. Сложные, связанные с использованием модели вычисления, такие, как получение оценок параметров, могут быть сформулированы в виде алгоритмов на соответствующем графе.

Классический граф состоит из множества вершин и множества рёбер. В графической вероятностной модели вершины графа обозначают случайные величины, а рёбра – связи между ними. Графические модели разных типов различаются способом построения и видом графа. Модели, основанные на направленных графах, носят название *байесовских сетей доверия* (Bayesian belief networks) [1, 18, 22, 23, 28, 33]. Модели, основанные на ненаправленных графах, часто называют *марковскими случайными полями* (Markov random fields) [1, 14, 18, 24]. Ещё один тип графических вероятностных моделей представлен в аппарате *алгебраических байесовских сетей* [29–36].

Разные типы моделей отличаются друг от друга своими свойствами и задают немного разные семейства распределений, но основной алгоритм точного байесовского вывода – алгоритм передачи сообщений – в любом типе моделей удобно представлять на *фактор-графе*. Фактор-граф даёт явное представление разложения сложной функции в произведение более простых: рассмотрим набор переменных  $\mathbf{X} = \{X_i\}$  и набор функций (факторов)

$f, f_1, \dots, f_m$ , таких, что  $f(\mathbf{X}) = \frac{1}{Z} \prod_{j=1 \dots m} f_j(Y_j)$ ,  $Y_j \subset \mathbf{X}$ . Двудольный граф  $G$  является фактор-графом для такого разложения функции  $f$ , если:

- вершинами первой доли являются переменные  $X_i$ ;
- вершинами второй доли являются функции  $f_j$ ;
- рёбра соединяют функции с их аргументами;
- $Z$  – нормировочная константа, определяемая как

$$Z = \int \dots \int \prod_{j=1 \dots m} f_j(Y_j).$$

Чаще всего фактор-граф используют для представления плотности совместного распределения случайных величин, однако известны также варианты его использования в теории кодирования [18]. Представление в виде фактор-графа удобно для того, чтобы определить *алгоритм передачи сообщений* (в этом контексте его обычно называют Belief Propagation) [11, 16]. Мы не будем подробно рассматривать базовый алгоритм передачи сообщений и отошлём читателя к книгам, посвящённым этой тематике [1, 18]; однако представляется важным подробно рассмотреть одну конкретную модификацию алгоритма передачи сообщений, позволяющую оперировать сложными факторами, которые понадобятся нам в моделях поведения пользователя.

## 2.2. Алгоритм Expectation Propagation

Один из алгоритмов приближённого вывода, использующий приближение функций в сообщениях, – это Expectation Propagation (EP) [19, 20]. Этот алгоритм в общем случае решает задачу поиска в экспоненциальном семействе распределений такого распределения, которое минимизирует КЛ-дивергенцию (расстояние Кульбака-Лейблера) с фиксированным распределением.

Распределение принадлежит *экспоненциальному семейству*, если его плотность может быть выражена следующим образом:

$$p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp(\theta^T u(x)),$$

где  $\theta$  – набор *естественных параметров* распределения,  $u(x)$  – вектор-функция соответствующей размерности. Многие стандартные вероятностные распределения принадлежат экспоненциальному семейству: распределение Бернулли, гамма- и бета-распределения, распределение Дирихле, нормальное распределение и другие. Для выборки из набора независимых одинаково распределенных случайных величин с распределением из экспоненциального семейства  $T(X) = \sum_n u(x_n)$  будет достаточной статистикой для параметров  $\theta$ .

Расстояние Кульбака-Лейблера для двух абсолютно непрерывных случайных величин с плотностями  $p, q$  задаётся как

$$KL(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx.$$

Теперь можно записать формально задачу, решаемую алгоритмом EP:

$$\text{Pr}[p] = \arg \min_{q \in F} KL(p||q),$$

где  $F$  – экспоненциальное семейство распределений. Мы будем называть это поиском *KL-проекции* (и обозначать через Pr).

Чтобы получить из этой задачи задачу поиска маргинальных распределений в графической модели, нужно в качестве фиксированного распределения взять распределение, заданное графической моделью, и искать приближение в виде произведения распределений от одной переменной (fully factorized approximation); здесь  $p(x_1, \dots, x_n) = \prod_a f_a(X_a)$  – распределение, заданное в виде произведения факторов, а  $q(x_1, \dots, x_n) = \prod_{i=1}^n q_i(x_i)$  – общий вид аппроксимирующего распределения.

Для аппроксимирующих распределений такого вида справедливо, что

$$\begin{aligned} q = \text{Pr}[p] &\Leftrightarrow \forall i = 1 \dots n \int_{X \setminus x_i} q(x_1, \dots, x_n) dX = \int_{X \setminus x_i} p(x_1, \dots, x_n) dX \\ &\Leftrightarrow \forall i = 1 \dots n q_i(x_i) = \int_{X \setminus x_i} p(x_1, \dots, x_n) dX, \end{aligned}$$

что и означает, что маргинальные распределения совпадают.

Если  $q(x)$  принадлежит экспоненциальному семейству, и  $u(x) = (u_1(x), \dots, u_k(x))$ , то КЛ-проекция упрощается:

$$q = \Pr[p] \Leftrightarrow \forall i = 1 \dots k \int_x u_i(x) q(x) dx = \int_x u_i(x) p(x) dx,$$

то есть достаточно, чтобы совпадали математические ожидания  $u_i$ .

Теперь, считая, что каждое  $q_i$  принадлежит экспоненциальному семейству, и используя эти два утверждения, получим, что

$$\begin{aligned} q = \Pr[p] &\Leftrightarrow \forall i, j \int_x u_{i,j}(x) q(x) dx = \int_x u_{i,j}(x) p(x) dx \\ &\Leftrightarrow \forall i \quad q_i(x_i) = \Pr \left[ \int_{X \setminus x_i} p(x_1, \dots, x_n) dX \right], \end{aligned}$$

где индекс  $i = 1 \dots n$ , а  $j = 1 \dots k_i$  проходит все компоненты соответствующего вектора  $u$  (для каждого  $i$  это могут быть разные числа).

Для решения поставленной задачи используется схема передачи сообщений, очень похожая на алгоритм Belief Propagation. Алгоритм EP так же работает с фактор-графом, передача сообщений осуществляется от факторов к переменным и от переменных к факторам. Сообщение от переменной с индексом  $i$  к фактору с индексом  $a$ :

$$\mu_{i \rightarrow a}(x_i) = \prod_{b \neq a} \mu_{b \rightarrow i}(x_i).$$

Сообщение от фактора с индексом  $a$  к переменной с индексом  $i$ :

$$\mu_{a \rightarrow i}(x_i) = \frac{1}{\mu_{i \rightarrow a}} \Pr \left[ \mu_{i \rightarrow a}(x_i) \int_{X_a \setminus x_i} f_a(X_a) \prod_{j \neq i} \mu_{j \rightarrow a}(x_j) dX_a \right].$$

Сравнивая это выражение с выражением для Belief Propagation, можно сказать, что Expectation Propagation аппроксимирует «истинные» функции сообщений, приравнивая моменты соответствующих распределений (moment matching) [19].

Затем передача сообщений от переменных к функциям и обратно продолжается до сходимости. Expectation Propagation сходится не всегда, однако если установлена сходимость, то полученный результат соответствует локальному минимуму  $KL(p||q)$  в исходной

постановке. Если заранее известны формы маргинальных распределений (например, все переменные дискретные), EP превращается в loopy belief propagation – алгоритм приближённого вывода, часто использующийся на фактор-графах с циклами.

Поскольку алгоритм EP устанавливает конкретный вид передаваемых сообщений, его можно применять к моделям, в которых присутствуют как дискретные, так и непрерывные случайные величины.

### **3. SCM: новая модель поведения пользователя.**

#### **3.1. Гипотезы о поведении пользователя.**

Основной результат данной работы – построение новой вероятностной модели поведения пользователя, учитывающей поведение пользователя на протяжении всей поисковой сессии. Прежде чем строить модель, сформулируем явно те предположения о характере поведения пользователя при работе с поисковой системой, которым должна удовлетворять модель.

1. Несоответствие между релевантностью и пертинентностью (intent bias) – пользователь неточно формулирует запрос к поисковой системе. В частности, он может удовлетвориться нерелевантным документом или проигнорировать релевантный.
2. «Гипотеза просмотра». Прежде чем сделать клик, пользователь просматривает описание ссылки. Модели, использующие это предположение, получили большое распространение и были подкреплены экспериментами [25].
3. «Каскадная гипотеза». Пользователь просматривает выдачу строго последовательно и покидает поиск, как только находит первый устраивающий его документ [6]. Пользователь может сделать несколько кликов на странице выдачи или ни одного.
4. «Свежесть». Пользователь с меньшей вероятностью будет кликать на ранее показанный документ [26].

#### **3.2. Конструкция модели**

Так как новая модель должна учитывать поведение пользователя на протяжении всей поисковой сессии, за основу была взята модель TCM. Однако сама модель TCM, представленная в [26], не

удовлетворяет условиям 1 и 3, к тому же её граф сильно зависит от конкретной сессии и требует реализации нетривиального алгоритма вывода. На уровне запроса в модели ТСМ можно использовать почти любую существующую клик-модель, поэтому, чтобы выполнить свойства 1 и 3, мы будем использовать модель DBN.

Свойство 4 можно поддержать введением дополнительного наблюдения: если документ был показан ранее, это отражено в логах, и его можно использовать как простую бинарную величину. Похожая процедура и так используется для получения индексов  $(i', j')$ . Таким образом, мы получаем модель, структура графа которой не зависит от конкретной сессии, а лишь от количества запросов в сессии.

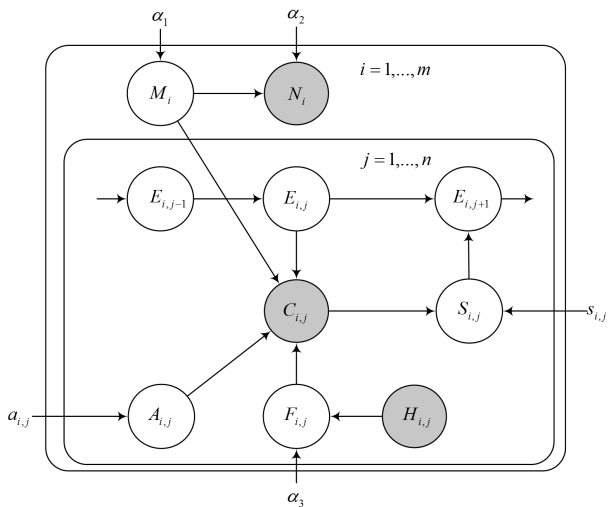


Рис. 1: Модель SCM в виде байесовской сети

На рисунке 1 приведена предлагаемая модель SCM (session click model) в виде байесовской сети. Назначение всех участвующих в модели случайных величин (все величины бинарные) описано в таблице 1. Ниже мы приводим набор условных вероятностей, завершающий формальное определение модели.



<i>Случайная величина</i>	<i>Определение</i>
$N_i$	Продолжает ли пользователь поиск. Наблюдаемая величина
$M_i$	Соответствует ли выдача по запросу $i$ информационным потребностям пользователя
$E_{i,j}$	Просмотрел ли пользователь описание документа $(i, j)$
$C_{i,j}$	Клик на документ $(i, j)$ . Наблюдаемая величина
$A_{i,j}$	Привлѐк ли документ внимание пользователя
$S_{i,j}$	Удовлетворяет ли документ $(i, j)$ информационные потребности пользователя
$H_{i,j}$	Был ли ранее в этой сессии показан документ $(i, j)$
$F_{i,j}$	Считает ли пользователь документ $(i, j)$ «свежим»

Таблица 1: Случайные величины модели SCM

$$\begin{aligned}
P(M_i = 1) &= \alpha_1, \\
P(N_i = 1 | M_i = 1) &= \alpha_2, \\
P(N_i = 1 | M_i = 0) &= 1, \\
P(F_{i,j} = 1 | H_{i,j} = 1) &= \alpha_3, \\
P(F_{i,j} = 1 | H_{i,j} = 0) &= 1, \\
P(A_{i,j} = 1) &= a_{i,j}, \\
E_{i,j-1} = 0 &\Rightarrow E_{i,j} = 0, \\
S_{i,j-1} = 1 &\Rightarrow E_{i,j} = 0, \\
E_{i,j} = 1 &\Rightarrow E_{i,j-1} = 1, S_{i,j-1} = 0, \\
P(S_{i,j} = 1 | C_{i,j} = 1) &= s_{i,j}, \\
P(S_{i,j} = 1 | C_{i,j} = 0) &= 0, \\
C_{i,j} = 1 &\Leftrightarrow M_i = 1, E_{i,j} = 1, S_{i,j} = 1, F_{i,j} = 1.
\end{aligned}$$

Параметр  $a_{i,j}$  определяет привлекательность документа, а точнее, соответствующего ему элемента на странице выдачи. Этот параметр влияет на вероятность клика. Параметр  $s_{i,j}$  можно интерпретировать как пертинентность документа: если запрос соответствовал потребностям пользователя, и пользователь закончил

просмотр выдачи кликом по этому документу, значит, документ оказался ему полезным.

В качестве оценки релевантности документа, как и в DBN, будем использовать

$$r_{i,j} = P(S_{i,j} = 1 | E_{i,j} = 1) = P(S_{i,j} = 1 | C_{i,j} = 1)P(C_{i,j} = 1 | E_{i,j} = 1) = a_{i,j}s_{i,j}.$$

Модель имеет три глобальных параметра поведения пользователя:  $\alpha_1$ ,  $\alpha_2$  и  $\alpha_3$ .

### 3.3. Оценки параметров

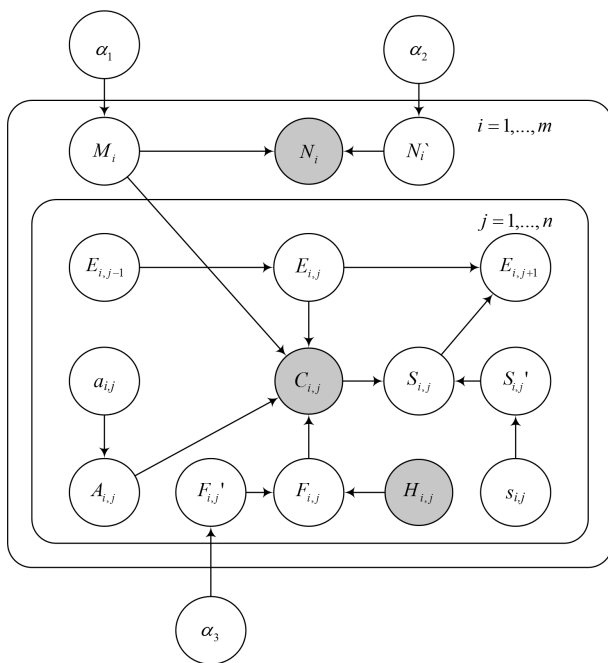


Рис. 2: Модифицированная байесовская сеть модели SCM

Для получения оценок параметров привлекательности ( $a_i, a_j$ ), pertinентности ( $s_i, s_j$ ) и релевантности ( $r_i, r_j$ ) документа, а также глобальных параметров поведения пользователя ( $\alpha_1, \alpha_2, \alpha_3$ ) используется байесовский подход. Каждый из оцениваемых парамет-

ров трактуется как непрерывная случайная величина на отрезке  $[0, 1]$ , оценкой для параметра выступает его апостериорное распределение после наблюдения, при поступлении нового наблюдения апостериорное распределение становится априорным, и процесс повторяется. Основное преимущество данного подхода – это возможность его инкрементальной реализации. Логи поисковых систем растут с огромной скоростью и требуют больших объёмов памяти для хранения. Байесовский подход позволяет получить оценки за один проход по данным и поддерживает пополнение, не требует наличия всех наблюдений в памяти компьютера одновременно.

Так как в графе модели есть циклы, необходимо использовать алгоритмы приближенного вывода. Для этого использовался алгоритм Expectation Propagation (EP) [19, 20]. Алгоритм EP поддерживает как дискретные, так и непрерывные случайные величины, прост в реализации и для обобщения, поскольку работает на фактор-графе (т.е. достаточно реализовать процедуры пересчёта сообщений для некоторых конкретных факторов, и алгоритм будет работать на любых моделях, содержащих факторы из реализованного набора). Специальные алгоритмы, предлагаемые авторами других работ для своих моделей, сложны в реализации и недостаточно хорошо описаны для их точного воспроизведения и сравнения. С помощью Expectation Propagation можно провести сравнение нескольких моделей в равных условиях.

Для применения предлагаемого подхода нужно преобразовать байесовскую сеть в фактор-граф и добавить оцениваемые параметры как отдельные переменные. В дальнейших выкладках будем опускать индексы у переменных моделей (там, где это несущественно), чтобы упростить запись.

Исходная байесовская сеть (рис. 1) задаёт плотность совместного распределения  $p(M, N, E, C, A, S, H, F)$ . Чтобы включить параметры наравне с обычными переменными, мы немного изменим модель. Для каждого использования параметра в условных распределениях введём дополнительную случайную величину по следующему принципу (на примере  $\alpha_1$ ): новая переменная  $N'_i$  будет иметь распределение Бернулли с параметром  $\alpha_1$ . Тогда условное распределение для  $N_i$  будет эквивалентно следующему утверждению:  $N_i = \neg M_i \& N'_i$ . На рисунке 2 приведена получившаяся байесовская сеть.

В получившейся сети все условные распределения либо соот-

ответствуют логическим формулам, если считать значения бинарных случайных величин логическими, либо связывают случайную величину из распределения Бернулли с её параметром. Приведём список формул, описывающих новую модель.

$$\begin{aligned}
 p(M_i = 1) &= \alpha_1, \\
 p(N'_i = 1) &= \alpha_2, \\
 N_i &= \neg N'_i | M_i, \\
 p(F'_{i,j} = 1) &= \alpha_3, \\
 F_{i,j} &= \neg H_{i,j} | F'_{i,j}, \\
 p(A_{i,j} = 1) &= a_{i,j}, \\
 E_{i,j} &= E_{i,j-1} \& \neg S_{i,j-1}, \\
 p(S'_{i,j} = 1) &= s_{i,j}, \\
 S_{i,j} &= S'_{i,j} \& C_{i,j}, \\
 C_{i,j} &= M_i \& E_{i,j} \& S_{i,j} \& F_{i,j}.
 \end{aligned}$$

Нетрудно видеть, что совместное распределение, задаваемое этой моделью, может быть представлено с использованием факторов только следующих видов:

- (1)  $p(C|A,B) = f_{AND}(A,B,C)$ , соответствующий связи  $C = A \wedge B$ ;
- (2)  $p(C|A,B) = f_{OR}(A,B,C)$ , соответствующий связи  $C = A \vee B$ ;
- (3)  $p(B|A) = f_{NOT}(A,B)$ , соответствующий связи  $B = \neg A$ ;
- (4)  $p(B|a) = f_{Bernoulli}(a,B)$ , соответствующий  $B \sim Bernoulli(a)$ .

На рисунке не приведены промежуточные случайные величины, вводимые с помощью фактора  $f_{NOT}$ , чтобы не загромождать изображение, так как нам важен лишь сам факт того, что этот набор факторов является достаточным.

Для применения алгоритма Expectation Propagation осталось выбрать тип маргинальных распределений для параметров. В качестве него было выбрано бета-распределение, так как оно принадлежит экспоненциальному семейству и определено для случайной величины из интервала  $[0,1]$ . Плотность бета-распределения

$f(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$  при значениях параметров  $\alpha = \beta = 1$  совпадает со стандартным равномерным распределением.

### 3.4. Вывод формул сообщений для алгоритма EP

В алгоритме Expectation Propagation накладываются ограничения на функции, передаваемые в сообщениях. В нашем случае будет всего два вида сообщений.

- (1) Сообщения от фактора  $f_{\text{Bernoulli}}(a, B)$  к случайной величине  $a$  будут иметь вид функции плотности бета-распределения:

$$\mu_{f \rightarrow a}(a) = \frac{1}{B(\alpha, \beta)} a^{\alpha-1} (1-a)^{\beta-1};$$

таким образом, достаточно передавать только параметры  $\alpha$  и  $\beta$ . Аналогично, сообщения от случайной величины  $a$  к фактору  $f_{\text{Bernoulli}}$  будет иметь такой же вид.

- (2) сообщения от фактора  $f_{\text{Bernoulli}}(a, B)$  к случайной величине  $B$ , а также от всех остальных факторов к другим случайным величинам будут иметь вид функции распределения Бернулли:

$$\mu_{f \rightarrow B}(B) = p^B (1-p)^{1-B},$$

где  $B$  принимает значения 0 или 1, а  $p \in [0, 1]$  – параметр распределения.

В некоторых случаях удобнее будет воспользоваться экспоненциальным представлением распределения Бернулли:

$$\mu_{f \rightarrow B}(B) = \frac{1}{1 + e^{\theta}} e^{\theta B},$$

где  $\theta \in \mathbb{R}$  – параметр распределения, связанный с  $p$ :  $\theta = \ln \frac{p}{1-p}$ . Для передачи такого сообщения достаточно передавать параметр  $\theta$ . Аналогично, сообщения от бинарных случайных величин к факторам будут иметь тот же вид.

Рассмотрим теперь конкретные сообщения, передаваемые в алгоритме Expectation Propagation, более детально.

*Фактор Бернулли.* Данный фактор представляет собой условную плотность распределения Бернулли при условии параметра:  $f_{\text{Bernoulli}}(a, B) = p(B|a) = a^B (1-a)^{1-B}$ . Сначала рассмотрим  $\mu_{f \rightarrow B}$ .

По правилу пересчёта сообщений для алгоритма Expectation Propagation,

$$\mu_{f \rightarrow B}(B) = \frac{1}{\mu_{B \rightarrow f}(B)} \Pr \left[ \mu_{B \rightarrow f}(B) \int f_{\text{Bernoulli}}(a, B) \mu_{a \rightarrow f}(a) da \right].$$

Пусть сообщение  $\mu_{B \rightarrow f}$  имеет вид  $\mu_{B \rightarrow f}(B) = q^B(1-q)^{1-B}$ ; тогда

$$\mu_{f \rightarrow B}(B) = \frac{\Pr \left[ q^B(1-q)^{1-B} \int a^B(1-a)^{1-B} \mu_{a \rightarrow f}(a) da \right]}{q^B(1-q)^{1-B}}.$$

Заметим, что  $\mu_{a \rightarrow f}(a)$  – плотность бета-распределения, а значит

$$\begin{aligned} \mu_{f \rightarrow B}(B) &= \\ &= \frac{1}{q^B(1-q)^{1-B}} \Pr \left[ q^B(1-q)^{1-B} E_{\mu_{a \rightarrow f}} [a]^B (1 - E_{\mu_{a \rightarrow f}} [a])^{1-B} \right] = \\ &= \frac{1}{q^B(1-q)^{1-B}} \Pr \left[ (q E_{\mu_{a \rightarrow f}} [a])^B ((1-q)(1 - E_{\mu_{a \rightarrow f}} [a]))^{1-B} \right]. \end{aligned}$$

Выражение в квадратных скобках является корректной формой распределения Бернулли, поэтому операция проекции не нужна, и мы приходим к следующей формуле:

$$\mu_{f \rightarrow B}(B) = E_{\mu_{a \rightarrow f}} [a]^B (1 - E_{\mu_{a \rightarrow f}} [a])^{1-B},$$

где математические ожидания берутся по распределению, содержащемуся в сообщении  $\mu_{a \rightarrow f}$ . Пусть оно имеет параметры  $\alpha$  и  $\beta$ ; тогда  $E_{\mu_{a \rightarrow f}} [a] = \frac{\alpha}{\alpha + \beta}$ , и итоговая формула для сообщения выглядит как

$$\mu_{f \rightarrow B}(B) = \left( \frac{\alpha}{\alpha + \beta} \right)^B \left( 1 - \frac{\alpha}{\alpha + \beta} \right)^{1-B}$$

Так как вместо функций в сообщениях передаются параметры, достаточно передать  $p = \frac{\alpha}{\alpha + \beta}$ . Если использовать экспоненциальное представление распределения Бернулли, получим

$$\theta = \ln \frac{p}{1-p} = \ln \frac{\frac{\alpha}{\alpha + \beta}}{1 - \frac{\alpha}{\alpha + \beta}} = \ln \frac{\alpha}{\beta}.$$

Таким образом, мы получили замкнутые формулы для пересчёта параметров сообщения  $\mu_{f \rightarrow B}$ .

Теперь рассмотрим  $\mu_{f \rightarrow a}$ . Согласно правилу,

$$\begin{aligned} \mu_{f \rightarrow a}(a) &= \\ &= \frac{1}{\mu_{a \rightarrow f}(a)} \Pr \left[ \mu_{a \rightarrow f}(a) \sum_{B=0}^{B=1} f_{\text{Bernoulli}}(a, B) \mu_{B \rightarrow f}(B) \right] = \\ &= \frac{1}{\mu_{a \rightarrow f}(a)} \Pr \left[ \mu_{a \rightarrow f}(a) \sum_{B=0}^{B=1} a^B (1-a)^{1-B} p^B (1-p)^{1-B} \right] = \\ &= \frac{1}{\mu_{a \rightarrow f}(a)} \Pr [(ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a)]. \end{aligned}$$

Чтобы найти проекцию, сначала нужно выражение в скобках сделать корректной плотностью распределения (нормализовать), затем приравнять моменты полученного распределения и моменты искомого бета-распределения. Так как сообщение  $\mu_{a \rightarrow f}$  содержит плотность бета-распределения, мы можем найти константу нормализации:

$$Z = \int (ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a) da = p E_{\mu_{a \rightarrow f}}[a] + (1-p)(1 - E_{\mu_{a \rightarrow f}}[a]).$$

Обозначим

$$\begin{aligned} g(a) &= \frac{1}{Z} (ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a), \\ \mu'(a) &= \Pr [(ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a)]. \end{aligned}$$

Мы ищем  $\mu'(a)$  в виде бета-распределения  $\mu'(a) = \frac{1}{B(\alpha, \beta)} a^{\alpha-1} (1-a)^{\beta-1}$ , которое принадлежит экспоненциальному семейству. Тогда для удовлетворения условий проекции необходимо и достаточно, чтобы  $E_g[a] = E_{\mu'}[a]$ ,  $E_g[a^2] = E_{\mu'}[a^2]$ . Так как  $\mu_{a \rightarrow f}$  имеет известные параметры, можно вычислить  $E_g[a]$  и  $E_g[a^2]$ :

$$\begin{aligned} E_g[a] &= \int ag(a) da = \frac{1}{Z} \int a (ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a) da = \\ &= \frac{1}{Z} (p E_{\mu_{a \rightarrow f}}[a^2] + (1-p) E_{\mu_{a \rightarrow f}}[a(1-a)]) = \\ &= \frac{1}{Z} ((2p-1) E_{\mu_{a \rightarrow f}}[a^2] + (1-p) E_{\mu_{a \rightarrow f}}[a]), \end{aligned}$$

$$\begin{aligned}
E_g[a^2] &= \int a^2 g(a) da = \frac{1}{Z} \int a^2 (ap + (1-a)(1-p)) \mu_{a \rightarrow f}(a) da = \\
&= \frac{1}{Z} (p E_{\mu_{a \rightarrow f}}[a^3] + (1-p) E_{\mu_{a \rightarrow f}}[a^2(1-a)]) = \\
&= \frac{1}{Z} ((2p-1) E_{\mu_{a \rightarrow f}}[a^3] + (1-p) E_{\mu_{a \rightarrow f}}[a^2]).
\end{aligned}$$

Вот как выражаются параметры бета-распределения через его моменты:

$$\left\{ \begin{array}{l} E_{\mu'}[a] = \frac{\alpha}{\alpha+\beta} \\ E_{\mu'}[a^2] = \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \alpha = \frac{E_{\mu'}[a]E_{\mu'}[a^2] - E_{\mu'}[a]^2}{E_{\mu'}[a]^2 - E_{\mu'}[a^2]} \\ \beta = \frac{(E_{\mu'}[a]-1)(E_{\mu'}[a] - E_{\mu'}[a^2])}{E_{\mu'}[a]^2 - E_{\mu'}[a^2]} \end{array} \right.$$

Таким образом, достаточно подставить в системы подсчитанные выше  $E_g[a]$  и  $E_g[a^2]$ , и получатся замкнутые формулы для пересчёта  $\mu'(a)$ . Но так как бета-распределение принадлежит экспоненциальному семейству, после деления на  $\mu_{a \rightarrow f}(a)$  мы снова получим бета-распределение (нужно будет просто вычесть из параметров  $\mu'$  параметры  $\mu_{a \rightarrow f}$ ). Можно было бы привести здесь явные формулы для сообщения  $\mu_{f \rightarrow a}$ , подставив во всех местах значения параметров  $\mu_{a \rightarrow f}$ , но формулы получились бы слишком громоздкие; приведённых выше выкладок достаточно для реализации пересчёта параметров  $\mu_{f \rightarrow a}$  на компьютере.

*Фактор OR.* Фактор OR задаётся следующим образом:

$$f_{OR}(A, B, C) = p(C|A, B) = \mathbb{I}(C = A \vee B),$$

где  $\mathbb{I}$  – индикаторная функция. Пусть сообщения от переменных к этому фактору имеют следующий вид:  $\mu_{C \rightarrow f}(C) = r^C(1-r)^{1-C}$ ,  $\mu_{A \rightarrow f}(A) = p^A(1-p)^{1-A}$  и  $\mu_{B \rightarrow f}(B) = p^B(1-p)^{1-B}$ . Тогда

$$\begin{aligned}
\mu_{f \rightarrow C}(C) &= \frac{1}{\mu_{C \rightarrow f}(C)} \times \\
&\times \Pr \left[ \mu_{C \rightarrow f}(C) \sum_{A, B} \mathbb{I}(C = A \vee B) p^A (1-p)^{1-A} p^B (1-p)^{1-B} \right],
\end{aligned}$$

$$\begin{aligned}
\mu(C) &= \sum_{A, B} \mathbb{I}(C = A \vee B) p^A (1-p)^{1-A} p^B (1-p)^{1-B} = \\
&= (p + q - pq)^C (1 - p - q + pq)^{1-C}.
\end{aligned}$$



Мы видим, что операция проекции просто не нужна (в квадратных скобках получается корректное распределение Бернулли), и

$$\mu_{C \rightarrow f}(C) = (p + q - pq)^C (1 - p - q + pq)^{1-C}.$$

В экспоненциальном представлении это будет выглядеть следующим образом:

$$\theta_C = \ln(e^{\theta_A} + e^{\theta_B} + e^{\theta_A + \theta_B}) = \text{OR}(\theta_A + \theta_B),$$

где  $\theta_C$  – параметр сообщения  $\mu_{C \rightarrow f}$ ,  $\theta_A$ ,  $\theta_B$  – параметры сообщений  $\mu_{A \rightarrow f}$  и  $\mu_{B \rightarrow f}$ . Введённая функция OR понадобится нам в дальнейшем.

Теперь переходим к сообщению  $\mu_{f \rightarrow A}$ :

$$\begin{aligned} \mu_{f \rightarrow A}(A) &= \frac{1}{\mu_{A \rightarrow f}(A)} \times \\ &\times \Pr \left[ \mu_{A \rightarrow f}(A) \sum_{C,B} \mathbb{I}(C = A \vee B) r^C (1-r)^{1-C} p^B (1-p)^{1-B} \right], \end{aligned}$$

$$\begin{aligned} \mu(A) &= \sum_{C,B} \mathbb{I}(C = A \vee B) r^C (1-r)^{1-C} p^B (1-p)^{1-B} = \\ &= r^A (1-r-q+2qr)^{1-A}. \end{aligned}$$

Здесь для получения проекции достаточно нормализовать распределение:

$$\mu_{f \rightarrow A}(A) = \left( \frac{r}{1-q-2rq} \right)^A \left( \frac{1-r-q+2qr}{1-q-2rq} \right)^{1-A}.$$

В экспоненциальном представлении получается следующее (обозначения прежние):

$$\theta_C = \ln \left( \frac{1 + e^{-\theta_B}}{1 + e^{-\theta_B - \theta_C}} \right) = \text{Gate}(\theta_B, \theta_C).$$

Введённой функцией Gate мы воспользуемся ниже. Сообщение  $\mu_{f \rightarrow A}$  вычисляется аналогично в силу симметрии.

*Фактор NOT.* Фактор NOT определён следующим образом:

$$f_{NOT}(A,B) = p(B|A) = \mathbb{I}(B = \neg A).$$

Пусть сообщения от переменных к этому фактору имеют следующий вид:

$$\begin{aligned}\mu_{A \rightarrow f}(A) &= p^A(1-p)^{1-A}, \\ \mu_{B \rightarrow f}(B) &= q^B(1-q)^{1-B}.\end{aligned}$$

Тогда

$$\begin{aligned}\mu_{f \rightarrow B}(B) &= \frac{1}{\mu_{B \rightarrow f}(B)} \Pr \left[ \mu_{B \rightarrow f}(B) \sum_A \mathbb{I}(B = \neg A) p^A(1-p)^{1-A} \right] \\ &= (1-p)^B p^{1-B}.\end{aligned}$$

В экспоненциальном представлении получится просто

$$\theta_B = -\theta_A.$$

Так как  $B = \neg A \Leftrightarrow A = \neg B$ , то аналогично получаем и другое сообщение:  $\mu_{f \rightarrow A}(A) = (1-q)^A q^{1-A}$ , и  $\theta_A = -\theta_B$ .

*Фактор AND.* Фактор AND задаётся следующим образом:

$$f_{AND}(A,B,C) = p(C|A,B) = \mathbb{I}(C = A \wedge B).$$

Сообщения от переменных имеют тот же вид, что и в предыдущих пунктах. Тогда, воспользовавшись тем, что  $C = A \wedge B \Leftrightarrow \neg C = \neg A \vee \neg B$ , можем заключить, что

$$\begin{aligned}\mu_{f \rightarrow C} : \theta_C &= -\text{OR}(-\theta_A, -\theta_B), \\ \mu_{f \rightarrow A} : \theta_A &= -\text{Gate}(-\theta_C, -\theta_B), \\ \mu_{f \rightarrow B} : \theta_B &= -\text{Gate}(-\theta_C, -\theta_A).\end{aligned}$$

Формулы для стандартной параметризации распределения Бернулли не приводятся, так как они довольно громоздки и ничего не добавляют ни для понимания, ни для практической реализации.

#### 4. Результаты экспериментов.

Исследования в данной области невозможны без доступа к реальным данным – историям поиска крупных действующих поисковых систем. Обычно эти данные являются закрытыми и используются исследовательскими отделами самих компаний-владельцев

поисковиков. С октября по декабрь 2011 года компания Яндекс проводила конкурс «Интернет–Математика» [38], в рамках которого участникам были предоставлены пользовательские поисковые логи. Все данные были полностью анонимизированы: документы и запросы представлялись лишь числовыми идентификаторами, запросы сгруппированы только по сессиям и не привязаны ни к каким пользователям.

В рамках работы были реализованы модели DBN, TCM и SCM с использованием алгоритма Expectation Propagation для байесовского оценивания параметров. Далее полученные с их помощью оценки релевантности документов использовались для ранжирования документов несколькими способами: в первом случае документы сортировались по оценке их релевантности для каждой модели, а затем оценки релевантности каждой из моделей использовались совместно с другими признаками документа и запроса для построения ранжирующей функции с помощью методов машинного обучения.

**4.1. Входные данные.** Предоставленные данные [38] представляют из себя пользовательские поисковые логи, содержащие запросные сессии, запросы, списки отранжированных документов, показанных пользователям, и пользовательские клики на них. Помимо этого, данные содержат экспертные оценки релевантности для некоторых документов. Все данные полностью анонимизированы, все запросы, сессии, документы представлены только числовыми идентификаторами. Запросы сгруппированы только по сессиям и не привязаны ни к каким идентификаторам пользователя.

Данные содержат:

- уникальных запросов: 30,717,251;
- уникальных документов: 117,093,258;
- сессий: 43,977,859;
- всего записей (строк): 340,796,067;
- оцененных троек запрос-регион-документ: 71,930;
- оцененных пар запрос-регион: 8,410.

Логи представляют собой набор строк, где каждая строка представляет одно из возможных пользовательских действий: запрос или клик.

Запрос:

```
SessionID TimePassed TypeOfAction QueryID RegionID ListOfURLs
```

Клик:

```
SessionID TimePassed TypeOfAction URLID
```

Здесь `SessionID` – уникальный идентификатор пользовательской сессии, `TimePassed` – время, прошедшее с начала текущей сессии в условных временных единицах (число миллисекунд в одной временной единице не разглашается), `TypeOfAction` – тип пользовательского действия (это может быть либо запрос Q, либо клик C), `QueryID` – уникальный идентификатор запроса, `RegionID` – уникальный идентификатор страны, из которой сделан запрос (этот идентификатор может принимать четыре значения), `URLID` – уникальный идентификатор документа, `ListOfURLs` – список документов, отранжированный слева направо так, как они были показаны пользователям на странице выдачи Яндекса (сверху вниз).

Пример:

```
109 0 Q 103 2 671 2183 3840 180 4566 514 4148 3153 1439 4991
109 103 C 2183
109 955 Q 100 2 197 197 11 179 5859 1624 1587 1624 5859 2186
109 960 C 197
```

Каждая строка в файле, содержащем оценки, имеет следующий формат:

```
QueryID RegionID
URLID RelevanceLabel
```

Здесь `QueryID` – уникальный идентификатор запроса, `RegionID` – уникальный идентификатор страны, из которой сделан запрос, `URLID` – уникальный идентификатор документа, `RelevanceLabel` – бинарная оценка релевантности (0 или 1). Пример:

```
1209161 2 5839294 1
1209161 2 1912415 1
1209161 2 1621201 1
1209161 2 1111 0
```

**4.2. Реализация.** Для реализации моделей была выбрана библиотека Infer.NET [21], разрабатываемая компанией Microsoft. В этой библиотеке реализованы современные алгоритмы приближённого вывода в графических моделях (в том числе и Expectation Propagation); она удобно интегрируется с платформой Microsoft .NET Framework и языком C#: операции с графической моделью выполняются посредством языковых структур самого C#.

Для моделей DBN, TCM и SCM были реализованы консольные приложения на языке C#, принимающие на вход текстовый файл в формате лога (описанного в предыдущем разделе). Выходом программ являются текстовые файлы, содержащие оценки релевантности для троек (запрос, регион, документ), встретившихся во входном файле. Все реализации следуют байесовскому подходу для оценки параметров, их общий алгоритм представлен далее.

- Вход: текстовый файл необходимого формата с поисковыми сессиями.
- Выход: оценки параметров релевантности для встретившихся документов.
- Используемые структуры: контейнер В (хеш-таблица).
- Для всех сессий во входном файле:
  - получить из контейнера В апостериорные распределения параметров для всех троек (запрос, регион, документ) в текущей сессии. Если какие-то из них ещё не содержатся в нём, инициализировать их равномерными распределениями;
  - получить наблюдаемые значения кликов для текущей сессии;
  - на основе наблюдаемых значений и априорных распределений получить апостериорные распределения параметров релевантности для документов текущей сессии с помощью алгоритма Expectation Propagation;
  - обновить апостериорные распределения в контейнере новыми значениями.
- Вернуть содержимое контейнера В – апостериорные оценки для всех троек (запрос, регион, документ).

**4.3. Результаты.** Мерой качества ранжирования в рамках конкурса [38] являлся AUC (Area Under Curve) [17]. Мы также использовали эту меру качества, так как она имеет ясную интерпретацию и позволяет сравнить себя с другими участниками конкурса. В ходе первого численного эксперимента модели CCM, DBN, TCM и SCM сравнивались по среднему значению AUC при ранжировании документов на основе лишь оценок релевантности документов; результаты приведены в таблице 2. Тестовое множество состояло из 370 запросов, принадлежащих к одному региону, для которых были предоставлены экспертные оценки (однако число оценённых документов для каждого запроса было различным).

Далее было проведено сравнение результатов работы модели SCM с другими моделями в режиме реальной работы: были подсчитаны дополнительные признаки для пар (запрос, документ) на основе лишь той информации, которая содержалась в предоставленных данных. Примеры признаков: средняя позиция документа в выдаче по запросу, среднее время до первого клика для запроса, отношение числа показов документа к числу кликов и тому подобное, всего 60 признаков. Затем параметры релевантности, полученные с помощью моделей поведения пользователя, объединялись с этими признаками для построения ранжирующей функции на основе метода градиентного бустинга с квадратичной функцией потерь [12, 14]. Результаты приведены в таблице 3.

Таблица 2: Ранжирование на основе оценок релевантности

<i>Модель</i>	<i>AUC</i>
CCM	0,5466
DBN	0,5654
TCM	0,5987
SCM	0,6034

## 5. Заключение.

В этой работе мы представили новую вероятностную модель поведения пользователя системы интернет-поиска – модель SCM (session click model). Мы подробно рассмотрели вывод в модели SCM, основанный на алгоритме Expectation Propagation, и привели результаты экспериментов, демонстрирующие на достаточно

Таблица 3: Ранжирование со статическими признаками

<i>Модель</i>	<i>AUC</i>
Ранжирование Яндекса	0,6119
CCM	0,6239
DBN	0,6252
TCM	0,6271
SCM	0,6296
Все модели и признаки вместе	0,6315

репрезентативном наборе данных, что SCM улучшает ранее известные модели. В дальнейшем мы планируем продолжать работу над разработкой моделей поведения пользователей, а также попробовать другие подходы к задаче ранжирования результатов поиска.

### Литература

1. *Bishop C. M.* Pattern Recognition and Machine Learning. Springer, 2006.
2. *Burges C. J., Shaked T., Renshaw E., Lazier A., Deeds M., Hamilton N., Hullender G.* Learning to rank using gradient descent // Proceedings of the 22<sup>nd</sup> International Conference on Machine Learning. 2005. P. 89–96.
3. *Burges C. J. C.* From RankNet to LambdaRank to LambdaMART : An Overview: Tech. rep.: Microsoft Research, 2010.
4. *Burges C. J. C., Svore K. M., Bennett P. N., Pastusiak A., Wu Q.* Learning to Rank Using an Ensemble of Lambda-Gradient Models // Journal of Machine Learning Research. 2011. Vol. 14. P. 25–35.
5. *Chapelle O., Zhang Y.* A dynamic Bayesian network click model for web search ranking // Proceedings of the 18<sup>th</sup> International Conference on World Wide Web. 2009. P. 1–10.
6. *Craswell N., Zoeter O., Taylor M., Ramsey B.* An experimental comparison of click position-bias models // Proceedings of the 1<sup>st</sup> ACM International Conference on Web Search and Data Mining. 2008. P. 87–94.

7. *Donmez P., Svore K. M., Burges C. J. C.* On the local optimality of LambdaRank // Proceedings of the 32<sup>nd</sup> Annual ACM SIGIR Conference. ACM, 2009. P. 460–467.
8. *Dupret G., Liao C.* A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine // Proceedings of the 3<sup>rd</sup> ACM International Conference on Web Search and Data Mining. 2010. P. 181–190.
9. *Dupret G., Piwowarski B.* A user browsing model to predict search engine click data from past observations // Proceedings of the 31<sup>st</sup> Annual ACM SIGIR Conference. 2008. P. 331–338.
10. *Freund Y., Iyer R., Schapire R. E., Singer Y.* An efficient boosting algorithm for combining preferences // Journal of Machine Learning Research. 2005. Vol. 4. P. 933–969.
11. *Frey B. J.* Extending Factor Graphs so as to Unify Directed and Undirected Graphical Models // Proceedings of the 19<sup>th</sup> Conference on Uncertainty in Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003. P. 257–264.
12. *Friedman J.* Greedy function approximation: a gradient boosting machine // Annals of Statistics. 2001. Vol. 29. P. 1180.
13. *Guo F., Liu C., Kannan A., Minka T., Taylor M., Wan Y., Faloutsos C.* Click chain model in web search // Proceedings of the 18<sup>th</sup> International Conference on World Wide Web. 2009. P. 11–20.
14. *Hastie T., Tibshirani R., Friedman J.* Elements of Statistical Learning. Springer, New York, 2008.
15. *Joachims T., Granka L. A., Pang B., Hembrooke H., Gay G.* Accurately Interpreting Clickthrough Data as Implicit Feedback // Proceedings of the 28<sup>th</sup> Annual ACM SIGIR Conference. 2005. P. 154–161.
16. *Kschischang F. R., Frey B. J., Loeliger H.-A.* Factor Graphs and the Sum-Product Algorithm // IEEE Transactions on Information Theory. 2001. Vol. 47, N. 2. P. 498–519.
17. *Ling C. X., Huang J., Zhang H.* AUC: a Statistically Consistent and more Discriminating Measure than Accuracy // Proceedings of the International Joint Conference on Artificial Intelligence 2003. 2003. P. 519–526.



18. *MacKay D. J.* Information Theory, Inference and Learning Algorithms. Cambridge University Press, 2003.
19. *Minka T.* Expectation Propagation for approximate Bayesian inference // Proceedings of the 17<sup>th</sup> Conference on Uncertainty in Artificial Intelligence / Ed. by J. S. Breese, D. Koller. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2001. P. 362–369.
20. *Minka T.* A family of algorithms for approximate Bayesian inference: Ph.D. thesis / Massachusetts Institute of Technology. 2001.
21. *Minka T., Winn J., Guiver J., Knowles D.* Infer.NET 2.4. 2010. Microsoft Research Cambridge. <http://research.microsoft.com/infernet>.
22. *Pearl J.* Probabilistic reasoning using graphs // Uncertainty in Knowledge-Based Systems / Ed. by B. Bouchon, R. R. Yager. Springer-Verlag, 1987. P. 201–202.
23. *Pearl J.* Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. NY etc.: Morgan Kaufmann, 1994.
24. *Prince S.* Computer Vision: Models, Learning, and Inference. Cambridge University Press, 2012.
25. *Richardson M., Dominowska E., Ragno R.* Predicting clicks: estimating the click-through rate for new ads // Proceedings of the 16<sup>th</sup> International Conference on World Wide Web. 2009. P. 521–530.
26. *Zhang Y., Chen W., Wang D., Yang Q.* User-click modeling for understanding and predicting search-behavior // Proceedings of the 17<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, 2011. P. 1388–1396.
27. *Заболотский В. П., Юсупов Р. М.* Основные проблемы устойчивости перехода к информационному обществу // Труды СПИИРАН. 2002. Т. 1, № 1. С. 13–26.
28. *Николенко С. И., Тулупьев А. Л.* Самообучающиеся системы. М.: МЦНМО, 2009. 288 с.
29. *Сироткин А. В.* Байесовские сети доверия: дерево сочленений и его вероятностная семантика // Труды СПИИРАН. 2006. Т. 1, № 3. С. 228–239.
30. *Сироткин А. В.* Вычислительная сложность алгоритмов локального апостериорного вывода в алгебраических байесовских сетях // Труды СПИИРАН. 2011. Т. 18. С. 188–214.

31. *Сироткин А. В., Тулупьев А. Л.* Локальный априорный вывод в алгебраических байесовских сетях: комплекс основных алгоритмов // Труды СПИИРАН. 2007. № 5. С. 100–111.
32. *Тулупьев А. Л., Николенко С. И., Никитин Д. А., Сироткин А. В.* Синтез апостериорных оценок истинности суждений в интегрированных базах знаний: детерминированный вариант // Известия высших учебных заведений: Приборостроение. 2006. № 11. С. 35–39.
33. *Тулупьев А. Л., Николенко С. И., Сироткин А. В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 608 с.
34. *Тулупьев А. Л., Николенко С. И., Сироткин А. В.* Синтез апостериорных оценок при поступлении свидетельств с неопределенностью в интегрированную систему знаний о неточных вероятностях // Известия высших учебных заведений: Приборостроение. 2006. № 11. С. 39–44.
35. *Тулупьев А. Л., Сироткин А. В., Николенко С. И.* Синтез согласованных оценок истинности утверждений в интеллектуальных информационных системах // Известия высших учебных заведений: Приборостроение. 2006. № 7. С. 20–26.
36. *Тулупьев А. Л., Сироткин А. В., Николенко С. И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петербургского ун-та, 2009. 400 с.
37. *Яндекс.* Поиск в интернете: что и как ищут пользователи. Информационный бюллетень «Яндекс», [http://download.yandex.ru/company/yandex\\_search\\_autumn\\_2008\\_ru.pdf](http://download.yandex.ru/company/yandex_search_autumn_2008_ru.pdf). 2008.
38. *Яндекс.* Конкурс «Интернет–математика». <http://imat-relpred.yandex.ru/>. 2011.

**Николенко Сергей Игоревич** — к.ф.-м.н.; научный сотрудник лаборатории математической логики ПОМИ РАН, доцент кафедры математических и информационных технологий, старший научный сотрудник проблемной лаборатории вычислительной биологии СПбАУ НОЦНТ РАН; [snikolenko@gmail.com](mailto:snikolenko@gmail.com); СПбАУ НОЦНТ РАН, ул. Хлопина, д. 8, корп. 3, г. Санкт-Петербург, 194021, РФ; р.т. +7(812)297-2145, факс +7(812)448-6998.

**Sergey I. Nikolenko** — Ph.D.; Researcher at Steklov Mathematical Institute, Adjunct Prof. at the Chair of Mathematical and Information Technologies department, Senior Researcher at the Laboratory of Algorithmic Biology at St. Petersburg Academic University; [snikolenko@gmail.com](mailto:snikolenko@gmail.com); St. Petersburg Academic University,

ul. Khlopina, d. 8, korp. 3, St. Petersburg, 194021, Russia; office phone +7(812)297-2145, fax +7(812)448-6998.

**Фишков Александр Александрович** — студент Санкт-Петербургского Государственного Политехнического Университета; jetsnguns@gmail.com; СПбГПУ, Политехническая, 29, 195251, Санкт-Петербург, РФ; р.т. +7(812)297-2095, факс +7(812)552-6080. Научный руководитель – С.И. Николенко.

**Alexander A. Fishkov** — M.Sc. student at St. Petersburg Polytechnical University; jetsnguns@gmail.com; St. Petersburg Polytechnical University, ul. Polytechnicheskaya, d. 29, St. Petersburg, 195251, Russia; office phone +7(812)297-2095, fax +7(812)552-6080. Advisor – S.I. Nikolenko.

**Поддержка исследований.** Работа была поддержана грантами РФФИ 11-01-00760-а и 11-01-12135-офи-м-2011, грантом Президента РФ для молодых кандидатов наук МК-6628.2012.1, грантом Президента РФ для ведущих научных школ НШ-3229.2012.1 и Федеральной целевой программой «Научные и научно-педагогические кадры инновационной России» за 2009–2013 гг.

Рекомендовано кафедрой МИИТ СПбАУ НОЦНТ РАН, зав. каф. Омельченко А.В., д.ф.-м.н.; лаб. ТИМПИ, зав. лаб. Тулупьев А.Л., д.ф.-м.н., доц.

Статья поступила в редакцию 02.02.2012.

## РЕФЕРАТ

*Николенко С.И., Фишков А.А.* **SCM: новая вероятностная модель поведения пользователей интернет-поиска.**

Для решения задачи ранжирования, а также для оценки качества поиска используются экспертные оценки релевантности документа запросу. На основе информации о содержимом документа, тексте запроса и других признаках этой пары ранжирующая функция строится с помощью методов машинного обучения, например бустинга. Модели поведения пользователей – одно из основных направлений исследований в области улучшения интернет-поиска, призванных породить такие признаки; такие модели обычно основаны на графических вероятностных моделях и обучаются из логов пользовательских действий (click logs).

В настоящей работе мы предлагаем модель поведения пользователя, учитывающую многократные запросы пользователя к поисковой системе. Модель получила название SCM (session click model) и является вариантом модели TCM (task-centric click model), в котором вывод в модели существенно упрощается и ускоряется. Вывод в модели производится при помощи одного из алгоритмов приближённого вывода, использующего приближение функций в сообщениях, а именно алгоритма Expectation Propagation. Этот алгоритм в общем случае решает задачу поиска в экспоненциальном семействе распределений такого распределения, которое минимизирует КЛ-дивергенцию (расстояние Кульбака-Лейблера) с фиксированным распределением. В работе подробно показан вывод конкретных сообщений в алгоритме передачи сообщений по методу Expectation Propagation.

В рамках практической части работы были реализованы модели DBN, TCM и SCM с использованием алгоритма Expectation Propagation для байесовского оценивания параметров. Далее полученные с их помощью оценки релевантности документов использовались для ранжирования документов несколькими способами: в первом случае документы сортировались по оценке их релевантности для каждой модели, а затем оценки релевантности каждой из моделей использовались совместно с другими признаками документа и запроса для построения ранжирующей функции с помощью методов машинного обучения. Оценки релевантности, полученные с помощью этих моделей, тестировались на анонимизированных данных, опубликованных компанией «Яндекс». Результаты экспериментов демонстрируют на достаточно репрезентативном наборе данных, что SCM не только ускоряет вывод, но и улучшает ранее известные модели.

## SUMMARY

*Nikolenko S.I., Fishkov A.A.* **SCM: a new user behaviour model for web search.**

In order to solve the ranking problem and to estimate search quality, search engines employ expert estimates of a document's relevance to a query. Based on the information about the document's contents, query text, and other features of this pair, the ranking function is constructed with such machine learning technique as boosting. User behaviour models constitute a major direction of study in web search research; they aim to provide informative features for the ranking function; such models are usually based on probabilistic graphical models and are learned from click logs.

In the present work, we propose a new user behaviour model that takes into account multiple user queries to the search engine. The model is called SCM (session click model), and it is a modification of the TCM model (task-centric click model) in which the inference is significantly simplified and therefore speeded up. Inference in this model is done with an approximate Bayesian inference algorithm that approximates functions in messages, namely the Expectation Propagation algorithm. This algorithm in the general case solves the problem of finding a distribution from the exponential family that minimizes KL-divergence (Kullback–Leibler distance) with a given distribution. In this work, we show detailed derivations of specific messages in the message passing algorithm based on the Expectation Propagation approach.

For the experimental part of this work, we have implemented DBN, TCM, and SCM models with Expectation Propagation for Bayesian parameter estimation. Further, document relevance estimates found with these models were used to rank documents in two settings: first, documents were sorted by their relevance estimate provided by each model, and then relevance estimates from these models were used, in a more practice-oriented setting, together with other document and query features to construct an actual ranking function with corresponding machine learning techniques. Relevance estimates found with these models were tested on an anonymized dataset published by the *Yandex* search engine. Experimental results demonstrate, on a sufficiently representative dataset, that SCM not only speeds up inference but actually improves the previously known models.