基于用户搜索行为的 query-doc 关联挖掘

朱 亮 [1, 2] 陆静雅 [1, 2] 左万利 [1, 2]

摘 要 query 和 doc 之间的关联关系是搜索引擎期望获取的一类有价值的信息. query 和 doc 间准确的关联分析不仅可以帮助搜索结果排序, 也在 query 和 doc 之间的桥接中起到重要作用, 以实现相关 query 和 doc 之间的信息传递, 有利于更深入的 query 理解和 doc 理解, 并在此基础上开展相关应用. 本文提出了一种基于用户搜索行为的 query 和 doc 关联关系挖掘算法, 该方法首先对用户搜索点击日志中的数据进行整理与分析, 构建 query 与 doc 间的二部图, 再通过采用马尔可夫随机游走模型对二部图数据进行建模, 挖掘二部图中的点击数据和 session 数据, 最终挖掘出点击日志中用户没有点击到的 doc 数据, 从而预测出 query 和 doc 间的隐含关联关系, 同时也可以利用该算法得到 query 和 query 潜在的关联关系. 基于以上理论基础, 我们实现了一套完整的日志挖掘系统, 通过大量的实验对比, 该系统在各方面均取得了优异的表现, 其中对检索结果相关性的性能提升可以达到 71.23 %, 这充分表明, 本文所提出的理论和算法能够很好地解决 query 和 doc 之间的隐含关系挖掘问题, 为提高搜索结果的召回率、实现查询推荐和检索结果聚类奠定了良好的前提基础.

关键词 关联关系, 搜索行为, 马尔可夫随机游走, 查询推荐, 检索结果聚类

引用格式 朱亮, 陆静雅, 左万利. 基于用户搜索行为的 query-doc 关联挖掘. 自动化学报, 2014, **40**(8): 1654−1666

DOI 10.3724/SP.J.1004.2014.01654

Query-doc Relation Mining Based on User Search Behavior

ZHU Liang[1, 2] LU Jing-Ya[1, 2] ZUO Wan-Li[1, 2]

Abstract The relationship between queries and docs is a valuable type of information that search engines hope to obtain. An exact correlation analysis between queries and docs is not only helpful for ranking search result, but also important for building a bridge between queries and docs to allow information transfer between related queries and docs, which is beneficial to a deep understanding of queries and to a series of applications. This paper presents a query-doc relation mining algorithm based on user search behavior. Initially, we collect and analyze users′ search log data to build a bipartite graph between queries and docs. Next we model the bipartite data using a Markov random walk model, and then mine the click-through data and session data from the bi-partite graph. Eventually, we can obtain doc data that the user did not click in the click-through data and predict the implied relationship between queries and docs. Besides, we can also take advantage of the algorithm to get the potential relationship between queries and queries. Based on the theoretical foundation described above, we construct a complete log data mining system. Through a large number of experimental contrasts, the system shows outstanding performance on many aspects, such as increasing relevance up to 71.23 %, which indicates that the theory and algorithms proposed in this paper can solve the problem of mining implicit relationships between queries and docs effectively. Our approach provides a good basis for increasing recall of search results, optimizing query recommendation and clustering retrieved results.

Key words Association relation, search behavior, Markov random walk model, query recommendation, clustering of retrieved results

Citation Zhu Liang, Lu Jing-Ya, Zuo Wan-Li. Query-doc relation mining based on user search behavior. *Acta Automatica Sinica*. 2014, **40**(8): 1654−1666

万维网的发展带来了信息爆炸式的增长, 人们的日常生活已离不开搜索引擎这一伟大的时代产物. 目前为止, Google、百度等通用搜索引擎经过十多年的发展, 在功能上已经相当完善, 搜索准确度等方面的性能也在不断提升. 尽管这些商用搜索引擎已经取得了很大成功, 但搜索结果的相关性仍有待提升, 目前大多数用户依旧需要多次调整搜索词才能找到自己真正需要的信息, 搜索结果缺乏个性化. 如何才能扩大相关搜索结果的召回并提升搜索结果与

收稿日期 2013-06-26 录用日期 2014-02-12
Manuscript received June 26, 2013; accepted February 12, 2014
国家自然科学基金 (60973040, 61300148), 中国博士后基金 (2012M510879), 吉林省重点科技攻关项目 (20130206051GX) 资助
Supported by National Natural Science Foundation of China (60973040, 61300148), Science Foundation for China Postdoctor (2012M510879) and Key Scientific and Technological Breakthrough Program of Jilin Province (20130206051GX)
本文责任编委 赵铁军
Recommended by Associate Editor ZHAO Tie-Jun
1. 吉林大学计算机科学与技术学院 长春 130012 2. 吉林大学符号计算与知识工程教育部重点实验室 长春 130012
1. College of Computer Science and Technology, Jilin University, Changchun 130012 2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012

查询间的相关性, 为用户提供更加合理的搜索结果动态排名成了当前亟待解决的问题之一.

在本领域已有一些相关研究, 文献 [1] 给出了一种基于用户历史点击日志的查询推荐方法, 该方法虽然使用了用户日志进行计算相关性并进行查询推荐, 但所采用的方法主要通过计算不同 query 在日志中的共现概率来完成相关推荐. 文献 [2] 通过对 query 的进一步理解来提高检索结果相关性, 但并未考虑到检索结果与用户搜索行为的关系. 文献 [3] 提出了一种基于单标签多关系的关系抽取算法, 文献 [4] 在 [3] 的基础上进行了改进, 提出了一种能够抽取多标签中的多关系的方法. 文献 [5] 同时提出了多实体多标签的关联关系抽取算法, 但都没有考虑到用户交互历史, Anagnostopoulos 等人定义了 max-last 和 max-sum 两个功能函数, 将查询推荐建模为全局功能函数的优化问题, 该问题属于 NP-hard 问题, 作者仅提供了有效的近似算法[6]. 文献 [7] 提出了一种基于 snipet 的点击模型用于查询推荐, 但并不是通过全局的用户交互历史来进行挖掘的 query-doc 之间的关系. Yan 等人通过建模当前查询、点击和下次查询之间的 high-order 关系, 提出情境感知查询推荐方法, 可有效捕获用户的潜在搜索意图, 当查询存在歧义时同样能提供准确的推荐信息, 该文献基于当前查询和点击为用户推荐信息, 不够全面 (如未考虑用户的长期兴趣)[8]. 文献 [9] 在传统的概念语义相似度计算方法基础上, 首次提出了一种自底向上的本体概念出现概率计算方法, 由此改进了基于节点信息量的概念语义相似度度量方法, 但单纯将此种方法应用于搜索引擎的 query 语义一致性判别或 query 聚类中, 效果并不是特别突出, 因为它并没有考虑到搜索引擎中用户的交互历史信息. 文献 [10−11] 分别给出了两种聚类方法, 这两种聚类算法都在传统的聚类基础上进行了创新, 不过将其应用于搜索引擎中时, 都并未考虑到全局用户的点击情况对聚类结果的影响. 文献 [12] 基于证据理论提出了一种基于多准则排序融合的证据组合方法, 可以将该方法应用于搜索引擎的结果排序中, 起到优化传统搜索结果排序准确度的作用, 但该方法同样未过多考虑用户的历史点击记录以及用户交互历史等因素. 文献 [13] 给出了基于情境信息的检索结果排序方法, 文献 [14] 在文献 [13] 的基础上, 考虑情境相关信息的同时, 也考虑到了个人情境与全局情境的不同, 文献 [15] 与文献 [14] 的研究方法大致相同, 都是通过考虑上下文信息来提高检索结果的相关性与个性化, 都并未考虑全局用户的点击情况, 并以此建模分析. 其中, Zhuang 等人提出了 Q-rank 重排算法, 基于查询日志提取查询上下文, 构建 $|Q_{ext}(q)|$ 和 $|Q_{adj}(q)|$, 并在此基础上计算 $\mathrm{RS}(d,q)$, 实现查询结果重排, 有效改善了搜索结果排名, 该方法仅考虑了 q 的邻近查询和基于 q 的扩展查询[16]. 文献 [17] 通过对外部语义关系进行建模来挖掘标签之间的关系, 但同样未考虑到用户交互历史信息, 其应用场景也与本文提出方法不同. 文献 [18] 给出了一种基于机器学习的方法对关联关系进行建模, 以求取关系间的内在联系, 该方法如应用在本文所述的领域范围, 通过考虑搜索引擎用户历史点击记录以及用户交互历史等因素, 同样也可以得到 query 与 doc 间的关联关系, 但该关联关系并没有考虑到全局信息.

本文的动机是考虑和应用用户点击信息, 提出一种基于用户点击日志的关系挖掘算法, 该方法通过挖掘点击日志中的点击数据、session 数据, 挖掘出点击日志中某个 query 用户没有点击到的相关 doc 数据, 从而预测出 query 和 doc 间隐含的关联关系, 同时也可以利用该算法挖掘出 query 和 query 潜在的关联关系. query 和 doc 之间的关联关系是搜索引擎期望获取的重要信息. query 和 doc 间准确的关联分析不仅可以帮助搜索结果排序, 而且也在 query 和 doc 之间架设了桥梁, 以实现相关 query 和 doc 之间的信息传递, 有利于更深入的 query 理解和 doc 理解, 并在此基础上开展相关应用.

通过上述 query 和 doc 间关联关系挖掘算法, 共可得到如下四类结果, 它们对于搜索引擎结果相关性及相关结果扩大召回具有重要的意义和作用: 1) query-to-doc: 该部分结果可用于搜索, 给出一个 query, 通过关联挖掘结果我们可得非常丰富的与该 query 关联到的但是没有点击的 doc, 关联程度的高低可作为排序时的一个参考因素, 例如该结果也可作为 LTR (Learning to rank) 的特征用到排序模型中; 2) query-to-query: 该部分结果可用于查询推荐, 给出一个 query, 通过关联挖掘结果我们可得到与之相关联的 query, 这部分结果还可用于 query 分类中, 起到语料扩充的作用; 3) doc-to-query: 给定一篇 doc, 我们可得到与之关联的 query 表示, 这样我们可用来为每个 doc 打一个 query tag, 或者将它的所有 query 表示作为表征该 document 的信息, 对其进行 wordsim 等分析; 4) doc-to-doc: 可用于关联反馈, 给定一个 doc, 我们可得到该 doc 关联的 doc, 具体可用到检索结果聚类等应用.

本文的创新之处在于考虑了用户点击信息, 并将点击信息充分融合到二部图的计算模型之中, 用于计算 query-query 之间的相关性. 本文的方法主要通过考虑用户与搜索引擎的交互历史来完成相关计算的. 本文提出的算法的一个重要应用就是查询推荐, 有关查询推荐的相关研究前人已经积累了诸多方法, 其中并没有发现与本文研究内容相近的推

荐方法, 本文的推荐过程是通过计算用户对不同查询对应的相同文档间的关联关系来完成的, 即通过挖掘不同的 query, 但这些 query 都被用户点击了相同的 doc, 此时, 本文的算法认为这些 query 是相关的, 以此来完成相关 query 推荐. 在检索排序方面, 本文给出的算法能够计算得到 query-doc 之间的隐含关联关系, 这一关系可以直接作为排序学习的一个参考因素, 实现更人性化的动态排序.

综上所述, 本文提出的 query-doc 关联关系挖掘算法不仅具有理论支撑, 而且具有重要实用价值. query-doc 间的关联关系可作为 Learning to rank 模型的一个参考因子, 同时, query-doc 的挖掘结果亦可直接作为搜索引擎排序的参考因素, 在传统的基于查询词与文档的相关性排序基础上, 考虑 query-doc 关联关系, 对于传统相关性较差, 排名在后的 doc, 如果在 query-doc 挖掘中具有出色的表现, 可以适当调整 doc 权重与相关性提高结果排名. 因此, query-doc 关联关系挖掘对于搜索引擎结果动态排序以及相关结果扩大召回具有十分重要的意义. 同时, 我们实现了一套完整的挖掘系统, 通过大量的实验对比, 该系统在各方面均取得了优异的表现, 实验部分, 我们对文中提到的算法进行了相关结果对比, 其中, 我们给出了 query-query 结果用于查询推荐, query-query 结果用于查询聚类, query-doc 结果用于检索结果动态排名等的性能测试数据, 结果显示, 对检索结果相关性的性能提升可以达到 71.23 %, 这充分表明, 本文所提出的理论和算法能够很好地解决 query 和 doc 之间的隐含关系挖掘问题, 为提高搜索结果的召回率、实现查询推荐和检索结果聚类奠定了良好的前提基础.

1 马尔可夫随机游走模型

随机游走 (Random walk, RW) 是某事物连续的随机游走这个过程的一个数学形式化描述. 随机游走模型于 1900 年由路易 · 巴舍利耶 (Louis Bachelier) 提出, 他把用于分析赌博的方法用于股票、债券、期货和期权. 在巴舍利耶的博士论文《*The Theory of Speculation*》中, 其具有开拓性的贡献就在于认识到随机游走过程是布朗运动.

1.1 马尔可夫随机游走

马尔可夫随机游走是指假设随机游走是以马尔可夫链或马尔可夫过程的形式出现的, 每一步, 系统根据概率分布, 可以从一个状态变到另一个状态, 也可以保持当前状态, 状态之间的跳转概率只与当前状态有关, 与之前的所有状态无关, 整个系统是无记忆的. 图 1 描述了一个简单的 Markov 随机游走模型, 其中 A 有 20 % 的概率转移到自身, 有 80 % 的概率转移到 B; B 有 40 % 的概率转移到自身, 有 30 % 的概率转移到 A, 有 30 % 的概率转移到 C; C 有 10 % 的概率转移到自身, 有 20 % 的概率转移到 A, 有 25 % 的概率转移到 B, 有 45 % 的概率转移到 D; D 有 35 % 的概率转移到自身, 有 65 % 的概率转移到 B.

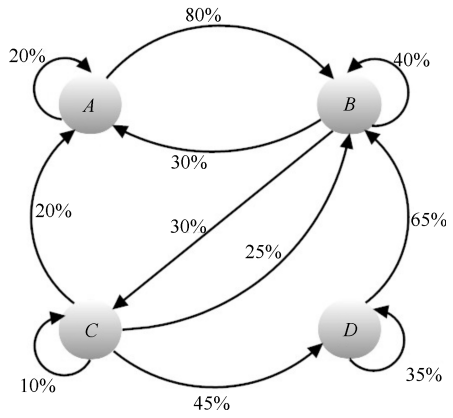

图 1 马尔可夫随机游走

Fig. 1 Markov random walk

在实际应用中, RW 通常作为一个时间随机过程的基本模型被广泛用于计算机科学, 例如, 随机游走模型可用于挖掘事物的内在关联关系、从有限的已标记数据中推测出未标记训练数据的标记、计算 PageRank 值等. Markov 链是一个离散时间随机过程, 这个过程中包括含了 N 个状态, 状态之间的跳转通过转移概率来描述, 并且从一个状态 i 跳转到另一个状态 j 的转移概率只取决于 i, 而和 i 之前的状态无关, 整个系统是无记忆的.

1.2 带自转移的马尔可夫随机游走模型

对于一个节点, $x^{(t+1)} \in \mathbf{R}^{1\times N}$, $x^{(t+1)} = (1-s)x^{(t)}P + sx^{(t)}I = (1-s)x^{(t)}P + sx^{(t)}$, 其中, P 为转移矩阵; I 为单位矩阵; s 为停留概率.

1.2.1 前向计算方法

一步转移概率 $(j \to k)$

$$P_{t+1|t}(k|j) = \begin{cases} (1-s)C_{jk}/\sum_i C_{ji}, & k \neq j \\ s, & k = j \end{cases} \tag{1}$$

其中, C_{jk} 为原始值; s 代表自转移概率. 如图 2 所示.

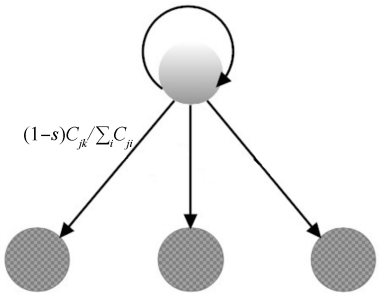

图 2　概率转移图示

Fig. 2　The probability transition diagram

t 步转移概率 $(j \to k)$

$$P_{t|0}(k|j) = [A^t]_{jk} \tag{2}$$

其中, A 表示转移概率矩阵 (行归一化).

1.2.2　后向计算方法

基于贝叶斯规则, 始于 k 止于 j 的概率可以表示如下:

$$P_{0|t}(k|j) \propto P_{t|0}(j|k)P_0(k) \tag{3}$$

$$P_{0|t}(k|j) = [A^t Z^{-1}]_{kj}, \quad Z_{jj} = \sum_i [A^t]_{ij} \tag{4}$$

如果 A 为稀疏矩阵, 可以通过下述简化方式快速计算得到:

$$P_{0|t}(k|j) = [\frac{1}{Z_j} A(...(A(Aq_j)))]_k \tag{5}$$

$$P_{t|0}(k|j) = [(((v_j A)A)...)A]_k \tag{6}$$

1.3　带随机跳转的马尔可夫随机游走模型

对于一个节点, $x^{(t+1)} \in \mathbf{R}^{1\times N}$,

$$x^{(t+1)} = (1-\beta)x^{(t)}P + \beta\frac{1}{N}1 \tag{7}$$

其中, PageRank 算法中就使用这个公式. Page Rank 随机跳转操作可描述如下:

冲浪者可以从一个节点跳到 Web 图上的任一节点. 假设 Web 图中的所有节点数目是 N, 那么随机跳转操作使得冲浪者以 $1/N$ 的概率跳到每一个节点.

在随机游走过程中加入随机跳转操作:

1) 当节点没有出链时, 冲浪者调用随机跳转操作;

2) 当节点包含出链时, 冲浪者将以 $0 < a < 1$ 的概率调用随机跳转操作, 并以 $1\sim a$ 的概率继续进行随机游走, 其中 a 是一个事先选定的固定参数, a 取值一般为 0.1.

1.4　带重新启动的马尔可夫随机游走模型

对于一个节点, $x^{(t+1)} \in \mathbf{R}^{1\times N}$,

$$x^{(t+1)} = (1-\gamma)x^{(t)}P + \gamma e_x \in \{0,1\}^{1\times N} \tag{8}$$

$e_x \in \{0,1\}^{1\times N}$ 是一个向量, 第 x 个元素为 1, 其它元素全为 0. 其中, Personalized PageRank 采用这种形式. e_x 称作 "Restart vector".

$$\begin{aligned}
x^{(1)} &= (1-\gamma)e_x P + \gamma e_x \\
x^{(2)} &= (1-\gamma)x^{(1)}P + \gamma e_x = \\
&\quad (1-\gamma)[(1-\gamma)x^{(1)}P + \gamma e_x] + \gamma e_x = \\
&\quad (1-\gamma)^2 e_x P^2 + \gamma(1-\gamma)e_x P + \gamma e_x \\
x^{(3)} &= (1-\gamma)x^{(2)}P + \gamma e_x = \\
&\quad (1-\gamma)[(1-\gamma)^2 e_x P^2 + \gamma(1-\gamma)e_x P + \\
&\quad \gamma e_x]P + \gamma e_x = \\
&\quad (1-\gamma)^3 e_x P^3 + \gamma(1-\gamma)^2 e_x P^2 + \\
&\quad \gamma(1-\gamma)e_x P + \gamma e_x
\end{aligned} \tag{9}$$

收敛时, 我们得到稳定状态下的概率:

$$x = (1-\gamma)xP + \gamma e_x \Rightarrow x = \gamma e_x[I - (1-\gamma)P]^{-1} \tag{10}$$

由此, 我们得到对于不同的节点, 只要改变 e_x 即可, 即对 $[I-(1-\gamma)P]^{-1}$ 取某一行. $[I-(1-\gamma)P]^{-1}$ 也被称为随机游走的核 (Random walk kernel).

2　基于 click graph 的随机游走模型与 query-doc 关联关系挖掘算法

本文提出了一种基于用户搜索行为的 query 和 doc 关联关系挖掘算法, 该方法首先对用户搜索点击日志中的数据进行整理与分析, 构建二部图, 然后采用马尔可夫随机游走模型进行建模, 挖掘点击日志中的点击数据、session 数据, 最终挖掘出点击日志中用户没有点击到的相关数据, 从而预测出 query 和 doc 间的隐含关联关系, 同时也可以利用该算法得到 query 和 query 潜在的关联关系.

2.1　点击图上的随走游走模型

本文提出的算法首先通过挖掘搜索引擎点击日志中的信息, 构建二部图, 即用户点击情况图, 进而采用马尔可夫随机游走模型进行建模. 可能的点击图如图 3 所示.

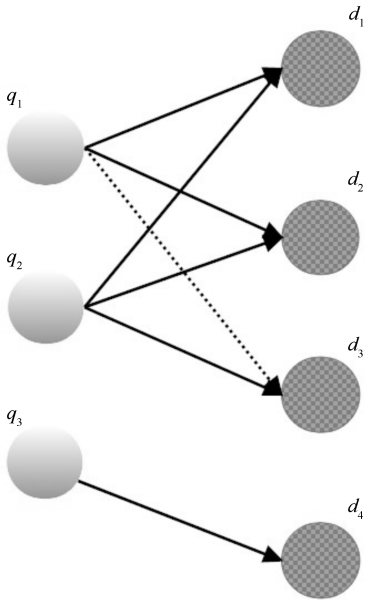

图 3 基于用户搜索历史的点击数据图

Fig. 3 The click-through data graph based on user search history

点击图 (Click graph) 的相关定义如下:

别名: 点击二部图.

节点: 可以是 queries and urls / ads / bid words.

边: 有点击的 query 和 doc 之间连一条边.

边的权值: 绝对点击值/加权点击值/用户满意度, 其中, 绝对点击值等于用户对 doc 的点击次数; 加权点击值等于绝对点击值 *doc 的权重, doc 的权重是通过该 doc 关联的 URL 的权重来计算得到的; 用户满意度作为参考因素, 相当于现今搜索引擎上的搜索结果用户点评功能给出的得分.

挖掘目的: 挖掘 query 和 doc 间隐含的 "边".

相关符号定义如下:

G: 点击矩阵 (n query and m doc).

$$G = [g_{ij}]_{(n+m)\times(n+m)} \tag{11}$$

其中, g_{ij} 为 G 中的第 i 行第 j 列元素.

$P = [p_{ij}]_{(n+m)\times(n+m)}$: 初始概率转移矩阵.

$$p_{ij} = g_{ij} \Big/ \sum_k g_{ik} \in [0,1] \tag{12}$$

$A = [a_{ij}]_{(n+m)\times(n+m)}$: 带自跳转的转移矩阵.

$$a_{ij} = \begin{cases} (1-s)p_{ij}, & i \neq j \\ s, & i = j \end{cases} \tag{13}$$

$$A = (1-s) \times \begin{bmatrix} 0_{n\times n} & Q \\ D & 0_{m\times m} \end{bmatrix} + s \times I \in \mathbf{R}^{(n+m)\times(n+m)} \tag{14}$$

当 $s = 0$ 时,

$$A = P = \begin{bmatrix} 0 & Q \\ D & 0 \end{bmatrix} \in \mathbf{R}^{(n+m)\times(n+m)} \tag{15}$$

其中,

$$\begin{aligned} D &= [D_{dq}]_{m\times n} \in \mathbf{R}^{m\times n} \\ Q &= [Q_{qd}]_{n\times m} \in \mathbf{R}^{n\times m} \end{aligned} \tag{16}$$

G 是一个对称矩阵; Q 是一个 (query, doc) 的行正规化矩阵; D 是一个 (doc, query) 的行正规化矩阵; I 是一个单位矩阵.

2.2 点击图上的前向与后向计算模型

前向随机游走:

$$P_{t|0}(d|q) = [A^t]_{qd}, \quad A^t = A \times A \cdots \tag{17}$$

后向随机游走:

$$P_{0|t}(d|q) = [A^t Z^{-1}]_{dq}, \quad Z = \mathrm{diag}(A^t 1) \tag{18}$$

2.3 query-doc 关联关系挖掘算法

由第 2.2 节可知, 通过 A 矩阵连乘可以计算得到子矩阵 Q 与 D, Q 即代表了 query 与 doc 间的关联关系, D 代表了 doc 与 query 间的关联关系.

算法 1. 基于用户搜索行为的 query-doc 关联挖掘

输入. *data*

输出. M

```
for all input data do
    confidence := calculate_confidence();
    cutting_data(confidence);
end for
s := calculate_retention_probability(data);
graph := generate_click_graph(data);
Matrix m := generate_matrix(graph);
Matrix M := m; INT idx=2;
while m is not convergent do
    if s == 0 then
        M := M * m;
    else
        M := sum_{k=0}^{idx} C_2^k s^k (1-s)^{(idx-k)} m_{idx}, s >= 0
        idx := idx + 1;
    end if
end while
```

17 $M :=$ add_confidence(M);

18 $M :=$ post_cutting_data(M)

由算法 1 可以计算得到四个子矩阵, 它们分别对应四类不同的结果, 其中, 右上角的子矩阵 (query → doc) 和左下角的子矩阵 (doc → query) 代表了不同类的信息, 可以用来做链接预测 (Link prediction); 左上角的子矩阵 (query → query) 和右下角的子矩阵 (doc → doc) 代表了同类信息, 可以用来做结果聚类 (Clustering). 具体来讲, 通过该算法可以得到四类信息, 分别存在如下的重要应用: query-to-query: 该部分结果可用于查询推荐, 给出一个 query, 通过关联挖掘结果我们可得到与之相关联的 query, 这部分结果还可用于 query 分类中, 起到语料扩充的作用; doc-to-query: 给定一篇 doc, 我们可得到与之关联的 query 表示, 这样我们可用来为每个 doc 打一个 query tag. 或者将它的所有 query 表示作为表征该 doc 的信息, 对其进行 wordsim 等分析; doc-to-doc: 可用于关联反馈, 给定一个 doc, 我们可得到该 doc 关联的 doc, 具体可用到检索结果聚类等应用.

本文提出的 query-doc 关联关系挖掘算法的大致流程如算法 1 所示, 该算法的具体细节内容可见以下各小节分述.

在算法 1 中, 输入 *data* 代表搜索引擎的用户点击日志数据集, s 代表马尔可夫随机游走模型中的自转移概率, 输出的 M 为结果矩阵, 即第 2.1 节中给出的基于点击图的点击矩阵通过算法迭代计算直至收敛时的结果, 它的右上角子矩阵 Q 代表的是 query-doc 间的关联关系, 左下角的子矩阵 D 代表的是 doc-query 间的关联关系.

2.3.1 迭代计算的步长

通过实验发现, 在实际的运算中并不需要迭代计算直至收敛, 其主要原因是计算代价太大, 并且也不需要那么精确的结果.

实际应用中的迭代次数通常是: query-doc 迭代 3 次; query-query 迭代 4 次; 这样既可以保证召回又可以控制风险, 同时计算成本较低, 还可以在同等外界因素的情况下获得较高收益.

2.3.2 算法优化

通过第三部分的实验测试可以发现, 在 query-doc 实验中表明 $s = 0$ 效果较好, 并且计算简单, $s > 0$ 时, 对于点击应用扩大召回应用中相关性指标提升不大.

前向随机游走计算时可使用小矩阵来进行: $\overline{Q} = Q \times D \times Q$, 当 $s = 0$ 时, 可化解为两个小矩阵相乘, 随机游走总共走三步, 其中 $\overline{Q} \in \mathbf{R}^{n\times m}$ 的第一行的值加和依然是 1.

后向随机游走计算时也可采用小矩阵来表示: $\overline{D} = D \times Q \times D$, 当 $s = 0$ 时, 同样可化解为两个小矩阵相乘, 随机游走总共走三步, 其中, $\overline{D} \in \mathbf{R}^{m\times n}$ 的每一列的值加和不是 1, 需要对其进行归一化.

对于自转移概率 $s > 0$ 的情况, 在第三部分实验中可以发现其实并没有 $s = 0$ 时效果好, 而且由于存在自跳转概率, 计算上更加复杂. 不过通过我们这里给出的优化方法, 可以转化为 $s = 0$ 的问题, 这时就可以利用 $s = 0$ 的结果进行计算了, 降低了运算复杂性的同时也可使得 s 调参更加简便. 这里, 我们给出了通过多项式展开进行推导优化的方法. 具体推导过程可参见附录部分.

由附录部分给出的计算可以得到, 算法经过计算优化后的一般公式 (当自转移概率 $s > 0$ 时):

$$A^{\tau} = \sum_{k=0}^{\tau} C_{\tau}^{k} s^{k} (1-s)^{\tau-k} A_k, \quad s \geq 0 \tag{19}$$

其中, A_k: $s = 0$ 时的 k 个 A 矩阵连乘, 这样便可先计算出 $s = 0$ 时的结果, 然后再利用 s 的系数, 对各个结果求和. 同时, 经过优化后的计算过程, 也不需要每调一次 s, 都去重新在大矩阵 A 上做乘法. 在考虑算法的实际实现情况时, 可以把重点放在优化小矩阵的乘法和加法上.

2.3.3 置信度

由于在搜索引擎的用户点击日志中存在冷门点击问题, 一个 query 只点击了一个 URL, 结果虽然为 100 %, 但置信度不高. 为了解决冷门点击问题, 我们不仅希望得到概率值, 还希望有置信度信息. 在本文提出的算法中, 我们采用了两种方式来计算得到置信度, 一是通过结果概率值乘以原始矩阵得到置信度 (Confidence), 这种方法是出于结果值没有置信区间的保证, 不是很可靠, 这时, 通过乘以点击次数就可以得到比较可靠且有说服力的结果值, 因为原始矩阵 (如绝对点击次数) 本身代表了置信度; 另一种方案是利用网络流中的最大流最小割思想, 这一想法存在一定的物理意义, 即路径上最小权重边上的权重值代表该路径的置信度; 多条路径则置信度累加.

2.3.4 裁剪与剪枝策略

裁剪目的:

1) 避免转义;

2) 减少中间数据的存储和运算时间.

剪枝策略:

1) 裁剪 doc: 关联过多 query 的 doc (网站首页, e.g. www.sina.com.cn);

2) 裁剪 query: 关联过多 doc 的 query (泛需求 query); 搜索频次、点击频数、查询天数、一天内查

询次数小于一定值的 query;

3) 裁剪 query-doc pair: 运算过程中概率值过小的 query-doc 对.

3 实验与结果分析

3.1 实验准备

3.1.1 数据集

本文采用的数据集原始数据来源于百度公司网页搜索用户点击日志, 该数据收集了 2012 年 7 月至 9 月间的用户搜索记录, 总记录数超过 2 000 亿, 部分数据来自于搜狗搜索用户日志. 裁剪前的 query 数约 253.1 亿, doc 数约为 729.2 亿, 点击的 query-doc pair 数约为 1 971.2 亿. 本文通过对数据进行置信度评估与适量裁剪后, 使用文中提出的 query-doc 关联挖掘算法对这些数据进行建模, 挖掘 query 与 doc 间存在的隐含关联关系.

3.1.2 数据预处理

数据集本身包含的是搜索引擎真实的用户点击日志数据, 在使用本文提出的 query-doc 关联关系挖掘算法之前, 需要进行一些预处理工作, 这样才能使算法达到更加理想的效果. 预处理工作的目的主要是为了避免转义, 同时减少中间数据存储和运算时间.

预处理的主要工作就是对数据进行剪枝, 裁剪策略主要包括以下三个主要方面: 裁剪 doc, 关联过多 query 的 doc, 比如网站首页 www.sina.com.cn; 裁剪 query, 关联过多 doc 的 query (泛需求 query); 搜索频次、点击频数和查询天数, 一天内查询次数小于一定值的 query; 裁剪 query-doc pair, 主要指的是运算过程中概率值过小的 query-doc pair 对. 实验中详细的数据量参见下表.

表 1 实验数据裁剪前后数目对比

Table 1 The number of experimental data comparison before and after cutting

对比说明	query	doc	query-doc
裁剪前	25.31	72.92	197.12
裁剪后	4.551	10.37	21.82
QQ 裁剪	0.25	0.631	0.8
QD 裁剪	0.31	0.943	2.74

表 1 给出了本文实验数据的具体数目, 其中裁剪前的数据是未经过处理的原始数据, 裁剪后的数据是通过上述的裁剪策略裁剪后的. QQ 裁剪显示的数据是用于第 3.2.2 节对 query-query 进行统计分析所使用的数据, QD 裁剪显示的数据是用于第 3.2.2 节对 query-doc 进行统计分析所使用的数据.

3.1.3 实验环境

本文的实验环境是由 1 200 台 Dell 服务器组成的 Hadoop 集群分布式运行环境, 其中单台服务器配备 4 颗 8 核 Intel Xeon 7 500 处理器, 内存 32 GB. 集群总计存储空间 24 PB. Hadoop 集群设有 600 map 槽位数和 600 reduce 槽位数.

说明: 由于计算资源有限, 不能每天跑 3 个月或更长时间的点击数据, 需要把最近 1 天或几天的点击数据增量式的加入进来, 以反映用户最新的点击行为.

3.2 实验结果

3.2.1 不同迭代次数的影响

大量的矩阵乘法运算会占用过多资源, 因此, 实验中, 我们针对不同的迭代次数得出的结果进行了对比, 以寻找在真实数据背后隐藏的秘密. 因为本文提出的算法属于预估类的, 其实也并不需要太精确的结果, 只要得出的结果是我们可以接受的就可以, 通过实验发现, 在实际的运算中并不需要迭代计算直至收敛, 因为在迭代到一定步长时, 结果的差异化越来越小, 在精度允许的范围内完全可以忽略不计.

表 2 不同迭代次数时的整体结果差异化对比

Table 2 The overall results comparison of different iterations

迭代次数	query-query	query-doc
1	0.510013	0.120115
2	0.343429	0.001037
3	0.116822	0.000813
4	0.010449	0.000001
5	0.009314	0.000121
6	0.007113	0.000731
7	0.008511	0.000413
8	0.007111	0.000133
9	0.004423	0.000002

表 2 给出了针对不同的迭代次数时, 结果的差异化数据, 给出的数据是当前迭代次数与前一次迭代时计算得出的结果值的平均差值, 我们在此, 通过计算排名前 10 000 的 query-query 或 query-doc 对数目来计算每一次迭代时的差值, 再求取其平均值得出表 1 中的数据. 通过数据可以明显看到, 算法经过 3～4 步即可得到在精度允许范围内的可取值, 充分印证了第 2.3.1 节中给出的结论: 在实际的运算中并不需要迭代计算直至收敛, 实际应用中的迭代次数通常是: query-doc 迭代 3 次; query-query 迭代 4 次; 这样即可以保证召回率又可以控制风险, 同时

计算成本较低, 还可以在同等外界因素的情况下获得较高收益.

3.2.2　实验整体效果

本文提出的基于用户搜索历史点击记录的 query-doc 关联关系挖掘算法, 采用马尔可夫随机游走模型对数据进行建模, 通过迭代计算会得出 query-doc 间的隐含关联关系, 同时, 除此之外, 还会得到 query-query, doc-query 和 doc-doc 三类关系, 比较重要的是 query-query 间的关联关系, 它对查询推荐以及查询改写等重要领域都会产生深远影响.

表 3 展示了基于该算法计算得到的 query-query 间的关联关系, 本次实验时采用的数据集是经过优化与裁剪后的, 对数据进行了一定的去噪处理, 最终包含有 query 个数 2.5 亿, doc 个数 6.3 亿, 点击 pair 数 8 亿. 在上一节给出的计算平台上共计运算时长 3.5 小时.

表 3　query-query 关联关系挖掘结果

Table 3　The query-query relation mining result

Rank	关联 query	权重
1	微博	0.191827
2	sina	0.132450
3	新浪网首页	0.054581
4	新浪网	0.051091
5	sina 微博	0.031500
6	weibo	0.029929
7	新浪邮箱	0.021058
8	xinlang	0.017145
9	新闻	0.015981
10	新浪体育	0.015113
11	www.sina.com	0.014001
12	新浪财经	0.009878
13	体育	0.008697
14	sina 邮箱	0.008490
15	www.sina.com.cn	0.008295
16	sina 体育	0.005700
17	新浪新闻	0.005185
18	新闻网	0.004652
19	新浪博客	0.004477
20	新浪首页	0.004401
21	体育新闻	0.003759
22	sian	0.002605
23	新浪邮箱登陆	0.002345

表 4 展示了基于本文给出的算法计算得到的 query-doc 间的隐含关联关系, 本次实验采用的数据集同样是经过优化与裁剪后的, 其中包含有 query 数目 3.1 亿, doc 数目 9.4 亿, 点击 pair 数目 27.4 亿. 在上一节给出的计算平台上共计运算时长 4.5 小时.

表 4　query-doc 关联关系挖掘结果

Table 4　The query-doc relation mining result

Rank	关联 URL	权重
1	http://www.sina.com.cn/	0.468695
2	http://t.sina.com.cn/	0.311185
3	http://sports.sina.com.cn/	0.063119
4	http://news.sina.com.cn/	0.052557
5	http://mail.sina.com.cn/	0.039044
6	http://finance.sina.com.cn/	0.015114
7	http://blog.sina.com.cn/	0.006403
8	http://finance.sina.com.cn/stock/	0.004468
9	http://news.qq.com/	0.003445
10	http://t.qq.com/	0.003215
11	http://sports.sohu.com/	0.002247
12	http://mail.sina.com.cn/cnmail/	0.002086
13	http://sports.qq.com/	0.002017
14	http://www.chinanews.com/	0.001750
15	http://baike.baidu.com/view/1567099.htm	0.001648
16	http://baike.baidu.com/view/2410.htm	0.001450
17	http://book.sina.com.cn/	0.001140
18	http://news.sohu.com/	0.001098
19	http://video.sina.com.cn/	0.000995
20	http://news.ifeng.com/	0.000906
21	http://i.blog.sina.com.cn/	0.000836
22	http://mil.news.sina.com.cn/	0.000787
23	http://ent.sina.com.cn/	0.000690

从表 3 中可以看出, 查询词 “新浪” 与 “微博” 最为相关, 这也很符合日常我们的搜索行为, 大多数人在查询 “新浪” 之后, 都会去点击 “新浪微博” 的官方网站, 从而通过本文算法计算得到 “新浪” 与 “微博” 之间存在密切的联系.

本文给出的 query-doc 准确度与 query-query 准确度都是通过给出在不同 pairs 数目的情况下, 强关联. 泛关联以及不关联的比例来反映的, 图 4 与图 5 分别给出了两者的强关联, 泛关系与不关系的具体曲线图. 其中, 实验中定义超过一定阈值的 pair 为强关联, 同理, 也可统计出泛关联与不关联的相关信息.

图 4 展示了 query-doc 关联挖掘实验的对比图, 实验中, 我们从结果中随机抽取若干对数据, 进行对比, 其中, 强关联: 泛关联: 不关联的比例稳定在 $31:61:8$. 由此可见, 本文的算法能够准确地挖掘出泛关联关系. 同理, 图 5 展示了 query-query 关联挖掘实验的对比, 其中, 强关联: 泛关联: 不关联的比

例平均稳定在 24: 170: 6.

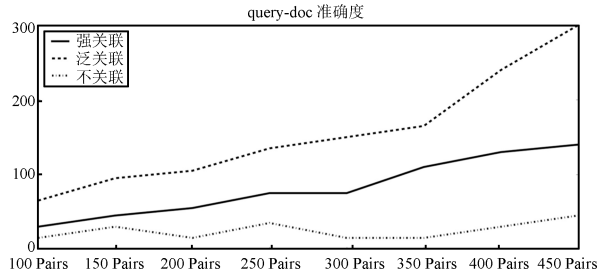

Fig. 4 query-doc 关联挖掘准确度对比

Fig. 4 The accuracy comparison of query-doc relation mining

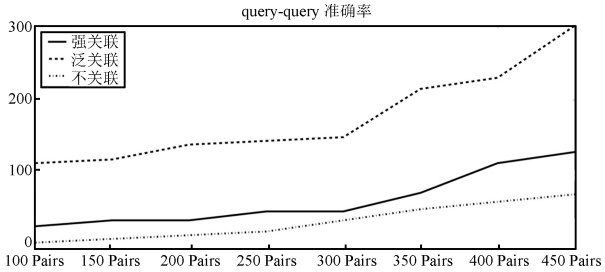

Fig. 5 query-query 关联挖掘准确度对比

Fig. 5 The accuracy comparison of query-query relation mining

由实验数据统计得出本算法的召回率也相当可观, 在 query-doc 关联挖掘实验中, 每个 query 关联的 doc 数, 挖掘前与挖掘后的比例是 1.2 : 32; 在 query-query 关联挖掘实验中, 每个 query 关联的 query 数, 挖掘前与挖掘后的比例是 10 : 121. 其中, 这里的挖掘前指的是直接关联.

3.2.3 对比实验

3.2.3.1 三种不同的 query-doc 建模方法比较

本文共实验了三种方法计算 query-doc 之间的关联程度, 分别是上述提出的基于马尔可夫随机游走模型的 RW 算法, SimRank 和 SimRank++. 图 6 是分别采用上述三种算法进行的实验结果对比图, 从中可以明显看出, RW 算法的表现最佳, 在不同召回的情况下均可获得较好的结果.

表 5 给出了当 $s = 0$ 与 $s \neq 0$ 两种情况下算法迭代计算得到的准确度比例与召回率比例对比情况, 其中 s 代表自转移概率, N 代表随机抽取的结果数目 (pair 数). 通过本对比实验可以得出以下重要结论: 在 query-doc 实验中表明 $s = 0$ 时效果最好, 并且计算简单. $s > 0$ 时对于点击应用扩大召回应用中相关性指标提升不大.

综上所述, 本文提出的基于马尔可夫随机游走模型的 query-doc 关联关系挖掘算法在各方面的性能都能够达到指标, 在很大程度上明显优于 SimRank 和 SimRank++ 算法, 而且该算法同时还能够得出 query-query, doc-query 以及 doc-doc 之间的关联关系, 这些结果分别在查询推荐、查询改写、文档的 query 表示以及检索结果聚类等重要应用中具有重要的作用. 实验中同时考虑了用户点击次数与加权点击值, 使得 query-doc 的关联关系更加精确, 整体效果较好. 由此可见, 本文所提出的理论和算法能够很好地解决 query 和 doc 之间的隐含关系挖掘问题, 为搜索引擎的结果扩大召回、查询推荐以及检索结果聚类等重要应用的实现奠定了基础.

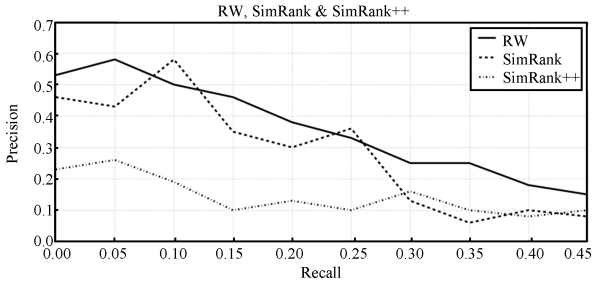

Fig. 6 SimRank 和 SimRank++ 算法精度对比

Fig. 6 The accuracy comparison between SimRank and SimRank++ algorithm

表 5 $s = 0$ 与 $s \neq 0$ 时准确度与召回率比例对比

Table 5 The accuracy and recall rate comparison between $s = 0$ and $s \neq 0$

	强: 泛: 不	挖掘前: 挖掘后	备注
query-doc	31 : 61 : 8	1.2 : 32	$s = 0 \,\&\, N = 1\,000$
query-doc	33 : 60 : 9	1.2 : 34	$s > 0 \,\&\, N = 1\,000$
query-query	24 : 170 : 6	10 : 121	$s = 0 \,\&\, N = 1\,000$
query-query	27 : 168 : 7	11 : 120	$s > 0 \,\&\, N = 1\,000$
query-doc	31 : 65 : 9	1.2 : 32	$s = 0 \,\&\, N = 2\,000$
query-doc	31 : 62 : 8	1.2 : 33	$s > 0 \,\&\, N = 2\,000$
query-query	24 : 171 : 6	10 : 119	$s = 0 \,\&\, N = 2\,000$
query-query	24 : 170 : 17	11 : 121	$s > 0 \,\&\, N = 2\,000$
query-doc	31 : 60 : 8	1.2 : 32	$s = 0 \,\&\, N = 3\,000$
query-doc	31 : 61 : 8	1.2 : 33	$s > 0 \,\&\, N = 3\,000$
query-query	24 : 179 : 7	10 : 122	$s = 0 \,\&\, N = 3\,000$
query-query	24 : 178 : 7	11 : 123	$s > 0 \,\&\, N = 3\,000$

3.2.3.2 query-query 结果用于查询推荐

本文提出的 query-doc 关联挖掘算法可以同时挖掘出 query-query 间的潜在的关联关系, 这种关联关系可用于 query 推荐. 在查询推荐方面, 我们将该方法与以下两种方法做出对比: 1) SimSearch, 该方法在文献 [1] 中提到过, 在传统的查询推荐算法中, 可以获得较高的评价, 性能较好; 2) CompSearch,

该方法在文献 [1] 中的相关研究部分提及, 通过计算全部用户历史日志中的 query 共现信息进行查询推荐.

在本次对比实验中, 我们采用了 1 万人工标注好的标准的 query-query pair 进行测试, 同时采用 360 搜索引擎 3 天的非标注 query 进行查询推荐. 我们在此分别评测了两种数据, 一种是针对标准的 query-query pair 进行的, 统计不同的查询推荐算法是否成功包括相应的标准数据集中的 query; 第二是, 通过对 3 天的 query 进行在线实时评估, 对查询推荐进行大规模测试, 统计用户点击数据, 浏览数据等, 作为判断哪种方法更优的一个选择时的参考因素.

表 6　不同查询推荐算法的结果对比

Table 6　The comparison of different query recommendation algorithm results

算法	召回率	加权 UV 评分
SimSearch	82.03 %	0.531396
CompSearch	89.21 %	0.562331
QQSearch	97.75 %	0.710018

表 6 给出的是三种不同的查询推荐算法在实际数据集上测试的效果, 其中第三个 QQSearch 算法是本文提出的 query-doc 关联挖掘算法在查询推荐上的应用. 召回率指的是在 1 万标准 query-query pair 中, 算法实际召回的个数与总数之间的百分比; 加权 UV 评分, 指的是将不同方法应用于 3 天的 query 的查询推荐上, 统计用户点击次数、浏览次数, 再归一化成 0～1 之间得到的结果. 由表 6 数据可以明显看出, 本文提出的 query-doc 关联挖掘算法应用于查询推荐中, 可以明显改善传统的查询推荐效果, 在质与量上都对传统的查询推荐算法予以补充.

3.2.3.3 query-query 结果用于查询聚类

本文提出的 query-doc 关联挖掘算法, 计算得到的 query-query 数据可以用于需要查询聚类等的重要应用, 在此, 我们给出了该算法用于查询聚类与传统的 K-means 方法用于查询聚类时的性能与效果对比.

我们针对 12 亿 query 进行聚类, 约 8 亿 query 自成一类, 其余 4 亿 query 所属类有至少 2 条以上 query.

本文提出的算法与 K-means 进行了简单的对比, 对比方法如下: 1) 从 K-means 的结果中抽样 40 万个类, 共 250 多万 query; 2) 对这些 query, 看是否在这边聚类结果的相同类中.

结果如下, 1) 在两边都属于相同类别中的 query 50 多万; 2) 对于在 K-means 的类别中不在这边聚类的相同类里, 大致有几种情况: 由于这边聚类粒度较细, 这边倾向于更细的划分, 比如口袋妖怪黑白什么精灵能学冲浪和口袋妖怪黑白冲浪怎么学这两个 query:

第一个跟以下 query 聚成一块:

口袋妖怪黑白什么精灵能学冲浪　2.0
口袋妖怪黑什么精灵可以学冲浪　2.0
口袋妖怪金版什么精灵能学冲浪　2.0

第二个跟以下 query 聚成一块:

口袋妖怪黑白冲浪怎么学　2.0
口袋妖怪黑白冲浪哪里学　2.0
口袋妖怪黑白冲浪在哪学　2.0

3) 由于这边的聚类对文本的相似度依赖更重, 原本语义一样的被划分到不同的类里去了. 这类例子, 一般还是在同一个大类中相邻;

4) K-means 算法聚类明显没有本文的方法好的, 比如, 比如: 中国少年先锋队队歌怎样指挥和中国少年先锋队队歌指挥.

前一个 query 在:

中国少年先锋队队歌怎样指挥　2.0
中国少年先锋队队歌怎么指挥　4.0

后一个 query:

中国少年先锋队队歌和指挥　2.0
中国少年先锋队队歌指挥　203.0
中国少年先锋队队歌指挥手势　4.0
中国少年先锋队队歌指挥教学　2.0
中国少年先锋队队歌指挥教程　2.0
中国少年先锋队队歌指挥视频　5.0
中国少年先锋队队歌的指挥　3.0

看起来, 本文提出的方法应用于查询聚类更好些.

另外, 包含这 250 万个对比 query 的类别中一共有 540 万 query, 去除其中孤立点 140 万, 约 400 万 query, 相对而言, 说明聚类结果的量上对 K-means 的聚类结果有所补充.

3.2.3.4 query-doc 结果用于检索结果动态排名

本文提出的 query-doc 关联挖掘得到的 query-doc 间的关联关系的一个重要应用就是将其用于检索结果的排名中, 传统的检索结果排序算法只考虑查询词与文档间的相似度等因素, 很少或几乎不会考虑用户的与搜索引擎的交互历史, 因此在排序上, 始终都是唯一的结果. 我们以 360 搜索引擎做为测试, 假定当用户进行翻页操作, 即代表对当前页的检索结果相关性不满意, 这样, 我们统计用户在两种排序方式下, 对 query 的翻页操作数据, 进行对比, 同时统计用户点击了第一页检索结果的次数. 我们采用了 3 天内 2 亿次查询数据进行评测, 得到结果如表 7 所示.

表 7 query-doc 数据用于动态排序与传统排序性能对比
Table 7 The performance comparison between dynamic ranking and traditional ranking by using query-doc data

测试方式	传统方法	本文方法
翻页操作次数	2 013 814	1 213 089
点击首页次数	132 000 000	189 000 000

由表 7 可以看出, 在采用本文的 query-doc 关联挖掘算法数据作为排序时的一个参考因素后, 首页点击次数明显提升, 同时, 翻页操作次数明显减少, 这充分表明, 本文提出的方法在检索结果相关性的性能提升方面占有举足轻重的地位, 在这里, 我们定义由表 7 中两种方式计算得到的性能提升值分别为 $VAL1$ 和 $VAL2$, 总的性能提交为两者平均值, 即 $(VAL1+VAL2)/2$.

上述实验数据通过对 2 亿 query 进行评测, 表 7 为其结果, 同时我们采用了不同的 query 数量进行测试, 综合得到最终的总体性能提升到 71.23 %. 这充分印证了本文提出的算法的有效性.

3.3 实验相关说明

因本文采用的数据量确实很大, 实验中也遇到了很棘手的问题, 现将有关情况说明如下:

其中主要就是针对大矩阵的存储与计算. 对于这么大的数据量, 本文目前的研究主要通过如下三点来完成:

1) 对原始数据资源进行裁剪, 减少对结果产生负面影响的部分数据, 来减少输入数据规模. 因篇幅所限, 本文在第 2.3.4 节中简单介绍了采用的剪枝策略;

2) 本文在计算中, 采用的是小矩阵的运算, Q 与 D, 而并非直接采用矩阵 A 进行;

3) 优化矩阵运算的开销, 本文算法并不需要迭代计算直至收敛才停止, 只需要进行 3 ∼ 4 次就可以得到较为理想的结果, 其中, 实验显示, s 不为 0 时的结果也并没有明显优于 s 为 0 的结果, 而且 s 不为 0 时, 明显增加计算量, 故本文实验中采用 $s=0$ 的情形进行展开. 同时, 为了更具有一般性, 本文在最后附录部分已经给出了目前采用的 s 不为 0 时的优化计算方法, 该方法不必每次都进行大矩阵运算, 而且可以部分直接引用 $s=0$ 的结果.

通过以上三点优化存储与计算后, 数据量其实也不小, 在算法初启与稳定后的占用内存资源同样会很大. 算法的高效执行还需要依赖于矩阵的计算优化, 而这一部分与本文研究的出发点不同, 本文重点是提出 query-doc 关联挖掘的算法, 并不是将重点放在矩阵运算与存储的优化上, 故在本文中, 只是提到了本文在计算时, 考虑到的优化因素. 对于较小规模的数据量, 本文的算法同样有效, 如果计算资源允许, 可以提高数据量进行实验. 对于结合其它方法进行相关 query-doc 关联挖掘, 并考虑是否可以引入传统机器学习方法与本文方法相结合, 并进一步优化小矩阵乘法运算将是进一步研究的重点内容, 在本文中因篇幅所限, 就不再过多展开讨论.

4 结束语

传统的搜索引擎基本都基于用户搜索关键词与文档之间的相关性来对搜索结果进行排序, 尽管现今多数商用搜索引擎在排序时充分考虑了其它很多方面的因素, 部分实现了动态排序, 但用户对搜索结果相关性与准确度的要求也在不断提高, 通常情况下用户还是需要多次修正查询词才能从海量的信息中找到所求, 因此, 如何提高搜索引擎结果的相关性以及相关结果的扩大召回显得尤为重要.

为了解决上述问题, 本文提出了一种基于用户搜索行为的 query 和 doc 关联关系挖掘算法, 该方法充分考虑了用户的搜索历史行为, 对全网用户的点击记录进行建模, 实现对用户点击过的 URL 动态调权, 从而使搜索结果更加人性化, 大幅度提高了搜索结果的准确度. 该算法首先对用户搜索点击日志中的数据进行整理与分析, 构建二部图, 再通过采用马尔科夫随机游走模型进行建模, 挖掘点击日志中的点击数据, session 数据, 最终挖掘出点击日志中用户没有点击到的相关数据 (同一 query, 但用户并未进行点击的相关 doc 数据), 从而预测出 query 和 doc 间的隐含关联关系, 同时也可以利用该算法得到 query 和 query 潜在的关联关系. 基于以上理论基础, 我们实现了一套完整的挖掘系统, 通过大量的实验对比, 该系统在各方面均取得了很好的表现, 其中对检索结果相关性的性能提升可以达到 71.23 %, 这充分表明, 本文所提出的理论和算法能够很好地解决 query 和 doc 之间的隐含关系挖掘问题, 为搜索引擎的结果扩大召回率, 实现查询推荐以及检索结果聚类等重要应用的实现奠定了良好的前提基础.

今后的研究工作主要包括以当前采用的 Random walk 算法为基础进行算法改进与优化, 设计通用算法框架, 并增加增量更新机制. 同时, 由于本文采用的输入数据并没有进行预处理, 可能带来覆盖率严重不足的问题, 因此, 后续的工作重点还包括对 query 进行归一化, 并优化 query-doc 权重计算方法, 使用更可靠、更稳定的权重. 通过找字面上相近的 query, 也可以提高覆盖率, 不过查找这样的 query 同样需要强大的自然语言处理技术作为支撑, 这样才能够从语义级别真正解决覆盖率不足的问题. 由此可见, 本文后续的研究工作任重而道远.

附录

$$A=(1-s)\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s\times I$$

$$A^2=(1-s)^2\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+2(1-s)s\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s^2\times I$$

$$A^3=(1-s)^3\times\begin{bmatrix}0&QDQ\\DQD&0\end{bmatrix}+s(1-s)^2\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+2(1-s)^2s\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+2(1-s)s^2\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+(1-s)s^2\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s^3\times I=$$

$$(1-s)^3\times\begin{bmatrix}0&QDQ\\DQD&0\end{bmatrix}+3s(1-s)^2\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+3(1-s)s^2\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s^3\times I$$

$$A^4=(1-s)^4\times\begin{bmatrix}QDQD&0\\0&DQDQ\end{bmatrix}+s(1-s)^3\times\begin{bmatrix}0&QDQ\\DQD&0\end{bmatrix}+3s(1-s)^3\times\begin{bmatrix}0&QDQ\\DQD&0\end{bmatrix}+3s^2(1-s)^2\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+3s^2(1-s)^2\times\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}+3s^3(1-s)\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s^3(1-s)\times\begin{bmatrix}0&Q\\D&0\end{bmatrix}+s^4\times I=$$

$$(1-s)^4\times A_4+4s(1-s)^3\times A_3+6s^2(1-s)^2\times A_2+4s^3(1-s)\times A_1+s^4\times I=\sum_{k=0}^{4}C_4^k s^k(1-s)^{4-k}A_k \tag{A1}$$

其中,

$$\vdots$$

$$A_5=\begin{bmatrix}0&QDQDQ\\DQDQD&0\end{bmatrix}\in\mathbf{R}^{(n+m)\times(n+m)}$$

$$A_4=\begin{bmatrix}QDQD&0\\0&DQDQ\end{bmatrix}\in\mathbf{R}^{(n+m)\times(n+m)}$$

$$A_3=\begin{bmatrix}0&QDQ\\DQD&0\end{bmatrix}\in\mathbf{R}^{(n+m)\times(n+m)}$$

$$A_2=\begin{bmatrix}QD&0\\0&DQ\end{bmatrix}\in\mathbf{R}^{(n+m)\times(n+m)}$$

$$A_1=\begin{bmatrix}0&Q\\D&0\end{bmatrix}\in\mathbf{R}^{(n+m)\times(n+m)}$$

$$A_0=I\in\mathbf{R}^{(n+m)\times(n+m)} \tag{A2}$$

References

1 Bhatia S, Majumdar D, Mitra P. Query suggestions in the absence of query logs. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. Beijing, China: ACM, 2011. 795−804

2 Li X. Understanding the semantic structure of noun phrase queries. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics Association for Computational Linguistics. Uppsala, Sweden: ACL, 2010. 1337−1345

3 Mintz M, Bills S, Snow R, Jurafsky D. Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2 Association for Computational Linguistics. Suntec, Singapore: ACL, 2009. 1003−1011

4 Peters S, Jacob Y, Denoyer L, Gallinari P. Iterative multi-label multi-relational classification algorithm for complex social networks. *Social Network Analysis and Mining*, 2012, **2**(1): 17−29

5 Surdeanu M, Tibshirani J, Nallapati R, Manning C D, Center A I. Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL). Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. 455−465

6 Anagnostopoulos A, Becchetti L, Castillo C, Gionis A. An optimization framework for query recommendation. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. New York, USA: ACM, 2010. 161−170

7 Liu Y, Miao J, Zhang M, Ma S, Ru L. How do users describe their information need: query recommendation based on snippet click model. *Expert Systems with Applications*, 2011, **38**(11): 13847−13856

8 Yan X H, Guo J F, Cheng X Q. Context-aware query recommendation by learning high-order relation in query logs. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. Glasgow, UK: ACM, 2011. 2073−2076

9 Li Wen-Qing, Sun Xin, Zhang Chang-You, Feng Ye. A semantic similarity measure between ontological concepts. *Acta Automatica Sinica*, 2012, **38**(2): 229−235

(李文清, 孙新, 张常有, 冯烨. 一种本体概念的语义相似度计算方法. 自动化学报, 2012, **38**(2): 229−235)

10 Zhou Lin, Ping Xi-Jian, Xu Sen, Zhang Tao. Cluster ensemble based on spectral clustering. *Acta Automatica Sinica*, 2012, **38**(8): 1335−1342
(周林, 平西建, 徐森, 张涛. 基于谱聚类的聚类集成算法. 自动化学报, 2012, **38**(8): 1335−1342)

11 Wang Li, Wu Cheng-Dong, Chen Dong-Yue, Li Meng-Xin, Chen Li. Exploring linear homeomorphic clusters on nonlinear manifold. *Acta Automatica Sinica*, 2012, **38**(8): 1308−1320
(王力, 吴成东, 陈东岳, 李孟歆, 陈莉. 非线性流形上的线性结构聚类挖掘. 自动化学报, 2012, **38**(8): 1308−1320)

12 Yang Yi, Han De-Qiang, Han Chong-Zhao. Evidence combination based on multi-criteria rank-level fusion. *Acta Automatica Sinica*, 2012, **38**(5): 823−831
(杨艺, 韩德强, 韩崇昭. 基于多准则排序融合的证据组合方法. 自动化学报, 2012, **38**(5): 823−831)

13 Xiang B, Jiang D, Pei J, Sun X, Chen E H, Li H. Context-aware ranking in web search. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Geneva, Switzerland Cochairs: ACM, 2010. 451−458

14 Chang L J, Xu Y J, Qin L. Context-sensitive document ranking. *Journal of Computer Science and Technology*, 2010, **25**(3): 444−457

15 Chen L J, Papakonstantinou Y. Context-sensitive ranking for document retrieval. In: Proceedings of the 2011 International Conference on Management of Data. Athens, Greece: ACM, 2011. 757−768

16 Zhuang Z M, Cucerzan S. Exploiting semantic query context to improve search ranking. In: Proceedings of the 2008 IEEE International Conference on Semantic Computing. Santa Clara, California, USA: IEEE, 2008. 50−57

17 Nguyen T V T, Moschitti A. End-to-end relation extraction using distant supervision from external semantic repositories. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: ACL, 2011: 277−282

18 Riedel S, Yao L, Mccallum A. Modeling relations and their mentions without labeled text. *Machine Learning and Knowledge Discovery in Databases*, 2010, **6323**(3): 148−163

朱 亮 吉林大学计算机科学与技术学院硕士研究生. 2011 年获吉林大学计算机科学与技术学院理学学士学位. 主要研究方向为网络搜索引擎, 信息检索与排序学习理论.
E-mail: zhuliang11@mails.jlu.edu.cn
(**ZHU Liang** Master student at the College of Computer Science and Technology, Jilin University. He received his bachelar degree from Jilin University in 2011. His research interest covers web search engines, information retrieval and learning to rank.)

陆静雅 吉林大学计算机科学与技术学院硕士研究生. 2012 年获吉林大学计算机科学与技术学院理学学士学位. 主要研究方向为网络搜索引擎, 信息检索与机器学习.
E-mail: lujy12@mails.jlu.edu.cn
(**LU Jing-Ya** Master student at the College of Computer Science and Technology, Jilin University. She received her bachelar degree from Jilin University in 2012. Her research interest covers web search engines, information retrieval and machine learning.)

左万利 吉林大学计算机科学与技术学院教授, 2005 年获吉林大学计算机软件与理论专业工学博士学位. 主要研究方向为数据库, 数据挖掘, 机器学习, 信息检索, 搜索引擎. 本文通信作者.
E-mail: wanli@jlu.edu.cn
(**ZUO Wan-Li** Professor at the College of Computer Science and Technology, Jilin University. He received Ph. D. degree in computer software and theory discipline from Jilin University in 2005. His research interest covers database, data mining, machine learning, information retrieval and search engines. Corresponding author of this paper.)