doi:10.3969/j.issn.1005-3026.2016.08.007

概率 XML 关键字检索排序算法

赵 越[1,2], 袁 野[1], 王国仁[1]
(1. 东北大学 计算机科学与工程学院, 辽宁 沈阳 110819; 2. 沈阳大学 信息工程学院, 辽宁 沈阳 110044)

摘 要: 探讨了针对概率 XML 文档集中与内容相关的关键字检索结果的排序问题,针对概率 XML 文档的特征提出了一种新的排序模式. 与仅取决于检索结果概率的检索排序算法不同,本文提出的排序算法充分考虑了节点对文档的区分程度、节点描述文档的程度,以及 XML 文档本身的结构特性,设计了满足以上特征的检索结果排序模型,并针对排序模型提出了新的倒排索引结构. 新的排序算法可以快速完成关键字检索,并将最相关的信息提供给用户. 模拟数据集实验验证了该方法的有效性.

关 键 词: 关键字检索;概率 XML 数据;SLCA;排序
中图分类号: TP 311 **文献标志码**: A **文章编号**: 1005-3026(2016)08-1095-05

A Ranking Algorithm of Keyword Search on Probabilistic XML Data

ZHAO Yue[1,2], *YUAN Ye*[1], *WANG Guo-ren*[1]
(1. School of Computer Science & Engineering, Northeastern University, Shenyang 110819, China; 2. School of Information Engineering, Shenyang University, Shenyang 110044, China. Corresponding author: ZHAO Yue, E-mail: zhaoy0927@163.com)

Abstract: Discusses the problem of efficiently ranking the search results of keyword related only to content on probabilistic XML data. A new ranking model is presented according to the characteristic of probabilistic XML data. Unlike the existing ranking algorithms which only depend on the probabilities of retrieval results, the new ranking algorithm proposed fully considered the degrees of nodes discriminating and describing the documents and the characteristic of probabilistic XML data. A ranking model of retrieval results which satisfied the above features is designed and a new inverted index structure for the ranking model is proposed. The new algorithm can accomplish keyword search quickly, so as to provide the most relevant information to the users. The results of simulation experiment show that the proposed method is effective.

Key words: keyword search; probabilistic XML data; SLCA (smallest lowest common ancestor); ranking

随着 XML 数据在网络数据中的广泛应用,关键字检索受到了更多关注. 关键字检索是一类不需要用户了解查询语言和数据结构的查询模式,因其简单易操作的特性受到了使用者的欢迎. 用户在网络上可以通过关键字检索到感兴趣的信息,但是由于受到一些因素的限制,网络上采集到的信息有时是不精确的. 例如,在通过卫星侦察和高空拍摄来完成信息采集的过程中,受到周围环境和拍摄角度等因素的影响,采集到的信息可信度较低,因此,这一类感知数据都普遍存在不确定性. 为了更好地表示这一类感知数据,文献[1]提出了一种数据模型,用来表示 XML 数据中的不确定数据,即 PrXML[2] 数据模型.

在 PrXML 数据模型上进行关键字检索已经有了一些成果. 文献[3]研究了在概率 XML 数据上关键字检索的 top-k 查询,提出了两种算法

收稿日期: 2014-08-24
基金项目: 国家自然科学基金资助项目(6100024,61332006,U1401256);国家重点基础研究计划项目(2011CB302200-G);中央高校基本科研业务费专项资金资助项目(N130504006).
作者简介: 赵 越(1984-),女,辽宁沈阳人,东北大学博士研究生,沈阳大学讲师;王国仁(1966-),男,湖北崇阳人,东北大学教授,博士生导师.

PrStack 和 EagerTopK,通过自底向上的方式记录各个节点包含关键字的概率分布情况,从而获得 k 个SLCA(smallest lowest common ancestor)检索结果. 文献[4]讨论了如何在概率 XML 数据上检索 ELCA(Exclusive LCA)结果集. 文献[5]分析了概率值对关键字检索结果的影响,提出了通过概率阈值检索 Quasi-SLCA 结果集的定义及剪枝原则. 然而,现有的概率 XML 数据上的关键字检索算法都是针对特定的结果集,查找满足相应条件的检索结果,概率成为判定结果的唯一标准,忽略了 XML 数据本身具有的结构特性. 因此,本文在概率值的基础上研究了 XML 数据本身的特点,讨论了影响关键字检索结果排序的因素,提出了一种基于 SLCA 结果集的概率 XML 关键字排序算法.

1 概率 XML 数据上的 SLCA 结果集

1.1 概率 XML 数据模型(PrXML)

概率 XML 数据是通过增加概率分布节点来表示数据的不确定性. 数据节点(ordinary)用来表示节点的实际信息,而概率分布节点则表示节点之间的概率分布情况;IND 节点表示其孩子节点之间相互独立,MUX 节点表示孩子节点之间的互斥关系. 因此概率 XML 数据也可以看成是由多个 XML 数据所组成的集合,其中的每一个元素称为一个可能世界实例(possible world).

1.2 SLCA 以及概率 SLCA

SLCA[6-9] 是目前应用较广的关键字查询结果集,当用户输入所有查询的关键字时,系统会在大量的 XML 文档中找出包含所有关键字的文档,并针对每一个文档进行 SLCA 关键字的检索.

定义 1 给定一个 XML 文档 T,和一个包含 t 个关键字的查询 $Q=\{k_1,k_2,\cdots,k_t\}$,若节点的标签或是内容中包含关键字 k_i,则该节点就是对于关键字 k_i 的匹配节点.

定义 2 给定一个 XML 文档 T 和两个节点 v_1 和 v_2,两个节点的 LCA 节点 v 满足下面两个条件:①v 是 v_1 和 v_2 的祖先节点;②在根节点到 v_1 和 v_2 的路径上不存在满足条件 1 的其他节点.

定义 3 给定一个 XML 文档 T 和一个包含 t 个关键字的查询 $Q=\{k_1,k_2,\cdots,k_t\}$,一个节点 v 是 SLCA 节点需要满足下面两个条件:①以 v 为根的子树中包含所有关键字;②以 v 为根的子树中不存在已满足条件 1 的其他节点.

概率 XML 数据可以看成是可能世界实例的集合,概率 SLCA 定义如下.

定义 4 给定一个概率 XML 文档 T 和查询 Q,一个节点 v 是概率 SLCA 节点需要满足:v 在至少一个可能世界实例中是 SLCA 节点.

定义 5 概率 SLCA 节点的概率为节点 v 是 SLCA 节点的所有可能世界实例的概率之和.

概率 SLCA 节点的概率可以表示为

$$P_{\text{slca}}(v)=\sum_{i=1}^{m}\{P(w_i)\mid \text{slca}(v,w_i)=\text{true}\}\ . \quad (1)$$

式中:$\text{slca}(v,w_i)=\text{true}$ 表示节点 v 在可能世界实例 w_i 中是一个 SLCA 节点;$P(w_i)$ 表示可能世界实例 w_i 的存在概率. 式(1)还可以表示为

$$P_{\text{slca}}(v)=P_{\text{exist}}(v)\times P_{\text{slca}}^{T_v}(v)\ . \quad (2)$$

式中:$P_{\text{exist}}(v)$ 表示节点 v 的存在概率;$P_{\text{slca}}^{T_v}(v)$ 表示以节点 v 为根的子树 T_v 中节点 v 是概率 SLCA 节点的概率.

2 关键字结果排序

2.1 关键字排序的影响因素

当用户提交关键字检索请求时,系统会在大量 XML 数据中检索包含关键字的 XML 数据,在每一个 XML 数据中检索包含关键字的最小节点树(以 SLCA 节点为根的子树),然后针对结果集根据其相关度进行排序.

对于一个概率 XML 数据来说,想要检索关键字,最简单也是最直接的方法就是根据概率 XML 数据中的概率分布节点生成所有可能世界实例,在每一个可能世界实例中检索关键字,对于每一个检索结果计算可能世界实例中的概率,得出概率关键字检索结果. 对于一个概率 SLCA 结果来说,概率值是评价结果的重要条件但却不是唯一的标准. 概率值是用来说明结果的存在概率,一般采用概率阈值作为评价标准.

定义 6 给定概率阈值 θ,若某个概率 SLCA 节点 v 的概率值大于 θ,那么该节点 v 就是一个概率阈值 SLCA-θ 节点.

大多数研究都认为,凡是概率值大于给定阈值的结果都将被认定是有意义的检索结果. 而作为一个概率 SLCA 节点,除去该节点的概率值以外,它在由概率 XML 数据所生成的可能世界实例中的重要性还需要通过节点与文档之间的关系以及节点与检索结果之间的关系来决定.

节点与文档之间的关系:从现有的关于确定的 XML 数据中关键字检索的研究和分析中可以知道,一个节点区分 XML 数据的能力和节点直

接、明确描述 XML 数据的能力都是影响节点语义的重要因素.

1）相同的节点可能出现在不同的 XML 数据中，而在不同的 XML 数据中同样的节点因其表示内容的不同而具有了不同的权重. 例如对于一个表示论文的 XML 数据来说，如果一个词分别出现在论文的题目和摘要中，则可以认为在题目中出现的词更符合用户的查询意图. 而对于 XML 数据中的关键字检索问题，根据 XML 数据的结构信息可以知道，由关键字得到的结果子树中的根节点是表示该子树的主要内容. 因此一个 SLCA 节点对 XML 数据的区分能力越强，则子树的权重越大. 节点的权重可以表示为

$$E(v) = \mathrm{e}^{\rho(v)} \times [-\rho(v)\ln\rho(v)]. \quad (3)$$

式中 $\rho(v)$ 表示节点 v 在 XML 文档 T 中所出现的频率. $-\rho(v)\ln\rho(v)$ 是节点的信息熵，用来表示节点的信息量.

2）一个节点的位置信息可以衡量该节点能否直接、明确描述 XML 数据. 为了说明一个节点的位置信息，首先给出一些相关定义.

定义 7　节点间的距离　给定一个 XML 文档 T 和两个节点 v_1 和 v_2. 如果 v_1 和 v_2 之间具有祖先后代关系，则它们之间的距离可以表示为连接两个节点之间最短路径上边的数量. 如果 v_1 和 v_2 之间不具有祖先后代关系，则它们之间的距离可以表示为

$$d(v_1 \to v_2) = d(v \to v_1) + d(v \to v_2). \quad (4)$$

式中 v 代表节点 v_1 和 v_2 之间的 LCA 节点，即 $v = \mathrm{lca}(v_1, v_2)$.

定义 8　节点的层级　给定一个 XML 文档 T 和节点 v_1 和 v_2，其中 v_1 是 v_2 的祖先节点，则节点 v_2 的层级为节点 v_1 的层级与节点 v_1 和 v_2 之间距离的和（根节点的层级为 1）：

$$l_{v_2} = l_{v_1} + d(v_1 \to v_2). \quad (5)$$

节点 v 的层级还可以表示为从根节点到节点 v 的路径上节点的数量.

定义 9　XML 数据的层级　给定一个 XML 文档 T，文档 T 的层级可以表示为在 T 中具有最大层级的节点的层级.

对于节点的位置信息，通常认为，越接近根节点的节点越能够直接描述该 XML 文档，它对文档的贡献也就越大，则其权重可以表示为

$$D(v) = \theta^{l_v} \times \prod_{1 \leqslant i \leqslant l_v} \frac{1}{N(v_i)}. \quad (6)$$

式中：θ 表示一个调节因子（$\theta < 1$），对于调节因子的设置可以参照文献[10]，本文将调节因子设置为 0.8；$N(v_i)$ 为节点 v_i 在从根节点到 v 的路径上出现的次数.

对于一个 XML 数据，一个节点的层级是固定的，但是对于不同的 XML 数据，XML 数据本身所具有的层级是不同的. 为了消除不同 XML 数据对检索结果的影响，本文采用文献[11]的方法对式(6)规范化，得到式(7). l_T 为 XML 数据树的层级.

$$D_{\mathrm{norm}}(v) = \theta^{l_v / l_T} \times \prod_{1 \leqslant i \leqslant l_v} \frac{1}{N(v_i)}. \quad (7)$$

对于概率 XML 数据的特殊结构，节点的层级并不能表示为从根节点到目标节点的路径上节点的数量，因为由概率分布节点的特征以及定义 4 可以知道，概率分布节点不表示任何实际的数据信息，因此概率 XML 数据上的节点层级用数据节点（ordinary）的数量来取代节点的数量.

定义 10　概率 XML 数据中节点的层级　给定一个概率 XML 文档 T 和节点 v，节点 v 的层级为从根节点到节点 v 的路径上数据节点的数量.

节点与检索结果之间的关系：从关键字匹配节点与 SLCA 节点之间的关系可以知道，关键字匹配节点之间的关系越紧密，以 SLCA 节点为根的子树就越能够满足查询者的查询意图. 这里用节点之间的距离来表示节点之间的相关度，距离越短则节点之间的相关度越高.

3）关键字匹配节点之间的距离可以用来衡量节点之间的相关度. 作为一个 SLCA 节点 v，以该节点为根的子树 T_v 的结构是最终作为检索结果输出给用户的. 因此关键字匹配节点之间的相关度会影响到关键字检索结果的权重. 其权重为

$$R(T_v) = \sum_{i=1}^{k} \sum_{j=1}^{n} d(v \to k_{ij}). \quad (8)$$

式中 k 表示用户输入的关键字数量，n 表示在以 v 为根的子树中关键字 k_i 的匹配节点数量.

定义 11　概率 XML 数据中节点之间的距离　给定一个概率 XML 文档 T 和节点 v_1 和 v_2，节点之间的距离可以表示为以下 3 种情况：

1）两个节点的 LCA 节点是互斥概率分布节点（MUX），则两个节点之间的距离为 0.

对于互斥概率分布节点来说，其孩子节点不会同时出现在一个可能世界实例中，也就是说两个节点是不能共存的，因此节点之间的距离为 0.

2）两个节点的 LCA 节点是独立概率分布节点（IND），则两个节点之间的距离为节点之间最短路径上数据节点的数量.

3）两个节点的 LCA 节点是数据节点，则两

个节点之间的距离为节点之间最短路径上数据节点的数量减 1.

对于一个 SLCA 子树,一个节点在子树中的层级是固定的,但是对于不同的 SLCA 子树,子树本身所具有的层级则是不同的. 为了消除不同子树层级对检索结果的影响,本文采用文献[11]的方法对公式(8)进行规范化,得到

$$R_{\text{norm}}(T_v) = \frac{R(T_v)}{m \times l_{T_v}}. \tag{9}$$

式中:m 表示在以 v 为根的子树 T_v 中关键字匹配节点的数量;l_{T_v} 为 T_v 的层级.

2.2 关键字编码

Dewey 编码又称为前缀编码,即每一个节点的编码都是以其父节点的编码作为本身节点编码的前缀. 因此,如果两个节点之间具有祖孙后代关系或是父子关系,则通过节点之间的编码就很容易得到判断. 但是传统的 Dewey 编码并没有考虑到概率分布节点和概率值的问题. 为了使 Dewey 编码适用于概率 XML 数据中的节点,通过增加节点类型的字母来区分节点的类型:IND 类型的节点可以增加字母 I,而 MUX 节点则可以增加字母 M.

节点之间的概率可以通过与原有 Dewey 编码的数字相加来区分,因为根据 Dewey 编码的方式,所有的数值都是大于或等于 0 的正整数,而概率值通常都是大于 0,小于 1,因此二者相加并不会影响原有编码对节点关系的判断.

2.3 计算模型

基于 2.1 节中所描述的影响关键字语义权重的几个因素,将 SLCA 节点在子树 T_v 中的权重定义为

$$\omega(T_v) = E(v) \cdot D_{\text{norm}}(v) \cdot R_{\text{norm}}(T_v). \tag{10}$$

基于定义 5 可知,SLCA 节点在概率 XML 数据中的权重可以表示为

$$\text{Rank}(v) = p(w_i) \cdot \omega(T_v) = p_{\text{exist}}(v) \cdot p_{T_v}(w_i) \cdot \omega(T_v). \tag{11}$$

式中 $p_{T_v}(w_i)$ 表示在以 v 为根的子树中节点 v 是 SLCA 节点的可能世界实例的概率,其概率可以根据文献[3]的方法计算.

3 实验结果及分析

现阶段还没有真实的概率 XML 数据,因此在实验时选取传统的 XML 数据,通过在 XML 树中随机添加概率分布节点和概率值的方式来得到概率 XML 数据[1]. 在读取到某个节点的时候,该方法会随机为该节点添加概率分布节点作为该节点的孩子节点,并将其原有的孩子节点作为概率分布节点的孩子节点添加到概率分布节点的下面. 在添加概率分布节点的时候需要考虑概率分布节点的性质. 根据 IND 的性质,该节点的每一个孩子节点的概率为大于 0 小于 1 的数值,而 MUX 节点的孩子节点的概率总和控制为小于或等于 1 的数值.

本文采用 XMARK 和 DBLP 两个数据集,选取的关键字数量均为 1 ~ 5. 通过上述的方法添加概率分布节点,由 XMARK 和 DBLP 两个数据集生成的概率 XML 数据集和关键字如表 1 所示. DBLP 数据集是一个相对层级较浅,但规模较大的数据集;而 XMARK 数据集则是一个在层级、结构和规模等方面都趋于相对平衡的数据集.

表 1 查询关键字概率 XML 数据集

Table 1 Probabilistic XML database in keyword search

文档	数据集	规模/MB	数据节点	IND	MUX
DOC1	XMARK	10	170 369	15 740	16 332
DOC2	XMARK	20	364 285	40 357	37 229
DOC3	XMARK	40	690 381	75 280	60 771
DOC4	XMARK	80	1 501 443	160 339	1614553
DOC5	DBLP	20	358 470	69 553	71 820
DOC6	DBLP	40	731 001	239 770	227 349
DOC7	DBLP	80	1 479 230	443 281	403 378
DOC8	DBLP	160	3 258 800	790 569	770 329

对于关键字检索的算法,查全率和查准率是检测算法性能的主要根据.

查全率 = (检索出的相关子树数量/相关子树总数) × 100% ;

查准率 = (检索出的相关子树数量/检索出的子树总数) × 100% .

本文通过两个步骤来判断结果的准确性. 首先针对 XMARK 和 DBLP 数据集的 XML 数据,根据传统的 XML 数据关键字检索算法(PrSLCA)来求得 SLCA 检索结果. 将两种方法的检索结果相结合,得出测试数据集的检索结果标准. 其次找到 50 个不同专业不同年级的学生,在给定他们查询关键字和检索结果后,由他们来投票选出最符合他们查询意图的检索结果及结果的排序. 由测试结果可知,本文采用排序原则算法(Ranking-SLCA)的查全率可以达到 100% . 这是因为本文算法采用的是与文献[3]相同的方法,在不考虑概率阈值的情况下,是可以查询到所有检索结果. 查准率可以达到 80% ~ 90% ,这是因为本文算法主要是针对结果的打分来进行排序.

不同个体针对相同的查询关键字会反映出不同的查询意图. 由查准率可知，本文算法可以极大满足用户的查询意图.

为了测试数据大小对查询效率的影响，采用表 1 中的 DOC1 ~ DOC4 作为测试对象，以 2 个关键字为例，测试关键字检索的时间. 然后，选取表 1 中的 DOC5 ~ DOC8 作为测试的数据对象. 结果如图 1 所示. 从图 1 中可以看出，如果关键字的数量相同，那么检索时间随着文档的增大而增加，这是因为文档越大，关键字匹配节点就越多，检索时间也会随之增加.

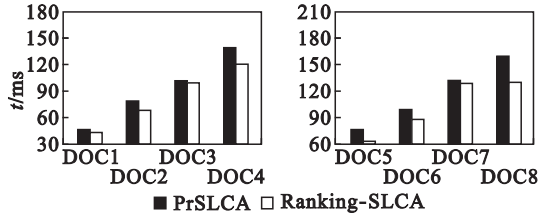

图 1　数据规模对查询效率的影响

Fig. 1　Impact of data size on query efficiency

图 2a 探讨了关键字数量对检索时间的影响. 利用文档 DOC1，分别以 1 ~ 5 个关键字为例. 从图 2a 中可以看出，当关键字数量增加的时候，检索时间也会随之增大. 这是因为关键字数量的增加会使 XML 文档中关键字的匹配节点数量成倍地增加，随之增加的计算过程还包括计算 SLCA 概率和节点权重的过程. 图 2b 所示为概率阈值对关键字排序算法的影响. 分别将概率阈值设为 0.4，0.5，0.6，0.7 和 0.8. 从图中可以看出，当概率阈值增大时，检索时间增加，这是因为在检索过程中可以根据节点的存在概率和以该节点为根的子树中节点的 SLCA 概率来预先估计节点的 SLCA 概率. 根据表 1 的数据可知，概率分布节点占数据节点量的 20% 以下，因此，概率阈值越大，在关键字检索的过程中排除节点的机会也就越大. 从而缩短关键字检索的时间.

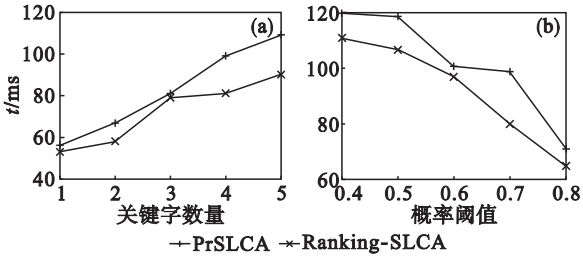

图 2　关键字数量和概率阈值对查询效率的影响

Fig. 2　Impact of the number of keywords and probability threshold on query efficiency

4　结　　语

本文将 SLCA 节点集的语义用于概率 XML 数据的关键字检索问题当中，提出了概率 SLCA 节点的概念和概率 SLCA 排序算法. 通过探讨关键字检索节点与 XML 文档之间的关系和关键字检索节点与检索结果结构之间的关系，在考虑到概率阈值的基础上，计算检索结果的权重，找出满足更符合用户检索要求的检索结果.

参考文献：

[1] Nierman A, Jagadish H V. ProTDB: probabilistic data in XML[C]//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, 2003: 646 – 657.

[2] Kimelfeld B, Kosharovsky Y, Sagiv Y. Query efficiency in probabilistic XML models [C]//Proceedings of ACM SIGMOD. Vancouver, 2008: 701 – 714.

[3] Li J X, Liu C F, Zhou R, et al. Top-*k* keyword search over probabilistic XML data [C]//Proceedings of International Conference on Data Engineering. Hannover, 2011: 673 – 684.

[4] Li J X, Liu C F, Zhou R, et al. Quasi-SLCA based keyword query processing over probabilistic XML data [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(4): 957 – 969.

[5] Zhou R, Liu C F, Li J X, et al. ELCA evaluation for keyword search on probabilistic XML data [C]// Proceedings of International World Wide Web Conference. Rio de Janeiro, 2013: 171 – 193.

[6] Guo L, Shao F, Botev C, et al. XRANK: ranked keyword search over XML documents [C]//Proceedings of ACM SIGMOD. San Diego, 2003: 16 – 27.

[7] Li Y, Yu C, Jagadish H V. Schema-free XQuery [C]// Proceedings of the 30th International Conference on Very Large Data Bases. Toronto, 2004: 72 – 83.

[8] Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML databases [C]//Proceedings of ACM SIGMOD. Baltimore, 2005: 537 – 548.

[9] Sun C, Chan C Y, Goenka A K. Multiway SLCA-based keyword search in XML data [C]//Proceedings of International World Wide Web Conference. Banff Alberta, 2007: 1043 – 1052.

[10] Gao N, Deng Z H, Jiang J J, et al. Combining strategies for XML retrieval [C]//Proceedings of INEX Conference. Berlin, 2011: 319 – 331.

[11] 张利军，李战怀，陈群，等. 基于关键字语义信息的 XML 文档分类[J]. 吉林大学学报(工学版)，2012，42(6)：1510 – 1514.
(Zhang Li-jun, Li Zhan-huai, Chen Qun, et al. Classifying XML documents based on term semantics [J]. *Journal of Jilin University* (*Engineering and Technology Edition*), 2012, 42(6): 1510 – 1514.)