第 34 卷第 11 期 通 信 学 报 Vol.34 No. 11
2013 年 11 月 Journal on Communications November 2013
doi:10.3969/j.issn.1000-436x.2013.11.004

DeweyTP：一种面向概率 XML 数据的编码方案

陈子阳[1]，刘佳[1,2]，张刘辉[1]，周军锋[1]
（1.燕山大学 信息科学与工程学院，河北 秦皇岛 066004；2.中国环境管理干部学院，河北 秦皇岛 066004）

摘 要：与普通 XML 文档相比，概率 XML 数据中节点的类型不唯一且节点的出现具有相应的概率。提出一种高效的编码策略 DeweyTP，该编码策略为每个 XML 数据节点分配唯一的能够体现节点类型和路径概率的编码，来支持节点类型检测和路径概率提取，因而提升系统性能。最后通过实验从时间和空间两方面验证了 DeweyTP 编码的高效性。

关键词：概率 XML 文档；DeweyTP 编码；编码方案；Dewey 编码
中图分类号：TP311 **文献标识码：**A **文章编号：**1000-436X(2013)11-0026-07

DeweyTP: a labeling scheme for probabilistic XML data

CHEN Zi-yang[1], LIU Jia[1,2], ZHANG Liu-hui[1], ZHOU Jun-feng[1]
(1. School of Information Science and Engineering, Yanshan Univers ty, Qinhuangdao 066004, China;
2. Environmental Management College of China, Qinhuangdao 066004, China)

Abstract: Compared with ordinary XML documents, nodes in the probabilistic XML documents have two characteristics, the type of nodes was non-unique and the nodes exist with a corresponding probability. As an efficient labeling scheme, DeweyTP was proposed to assign each node a unique label, which contains the type and path probability of nodes, supporting the detection of node type and the extraction of path probability, and thus improves the system performance. Finally, experimentally evaluated DeweyTP encoding scheme were experimentally evaluated in aspects of time and space efficiency.

Key words: probabilistic XML document; DeweyTP encode; labeling scheme; Dewey encode

1 引言

概率 XML 数据管理是近年来研究者关注的热点问题之一[1~6]。一个概率 XML 文档通常可以看作是一个在节点和边上带有标注信息的标签树。该标签树由 2 种节点构成，普通节点和分布节点。普通节点代表实际的数据，分布节点表示其孩子节点出现的概率约束关系。此外，还需为概率树中的每条边附加一个 (0,1]区间内的数字，该数字表示在父亲存在的前提下，其孩子出现的条件概率。图 1 给出了一个概率 XML 树，其中分布节点使用方框或椭圆表示，其他节点均为普通节点，当某条边上没有值时，表示该边对应的概率为 1。

为了提升查询处理的性能，需要为每个节点指定一个唯一的编码。在现有的编码中，Dewey 编码[7~9]是使用最广的一种编码。Dewey 编码不仅支持关键字查询，还支持结构化查询。由于概率 XML 文档中的节点类型有多种而且节点的出现具有概率性，因此单纯使用 Dewey 编码无法支持对概率 XML 数据的查询处理。文献[10]在处理概率 XML 数据时，每个节点的编码由 2 部分组成：①不同节点类型的改进 Dewey 编码；②从根到该节点的路径上节点的条件概率信息。例如，在图 1 中，节点旁边的第 3 行数字是文献[10]为节点指定的改进 Dewey 编码，例如节点 IND_2 的编码为 1/I2/1/M3/1/I1，其中字符“I”和“M”分别表示节点 IND_1 和 Mux_1 是分布节点中的独立节点和互斥节点。对于 IND_2 节点，它的路径概率为 1/0.4/1/0.5/1。文献[10]在使用该方法进行查询时，需要比较整个编码，其时间代价和文档最长路径的长度成正比。另外，

收稿日期：2013-07-20；修回日期：2013-09-20
基金项目：国家自然科学基金资助项目(61272124, 61103139, 61073060)
Foundation Item: The National Natural Science Foundation of China (61272124, 61103139, 61073060)

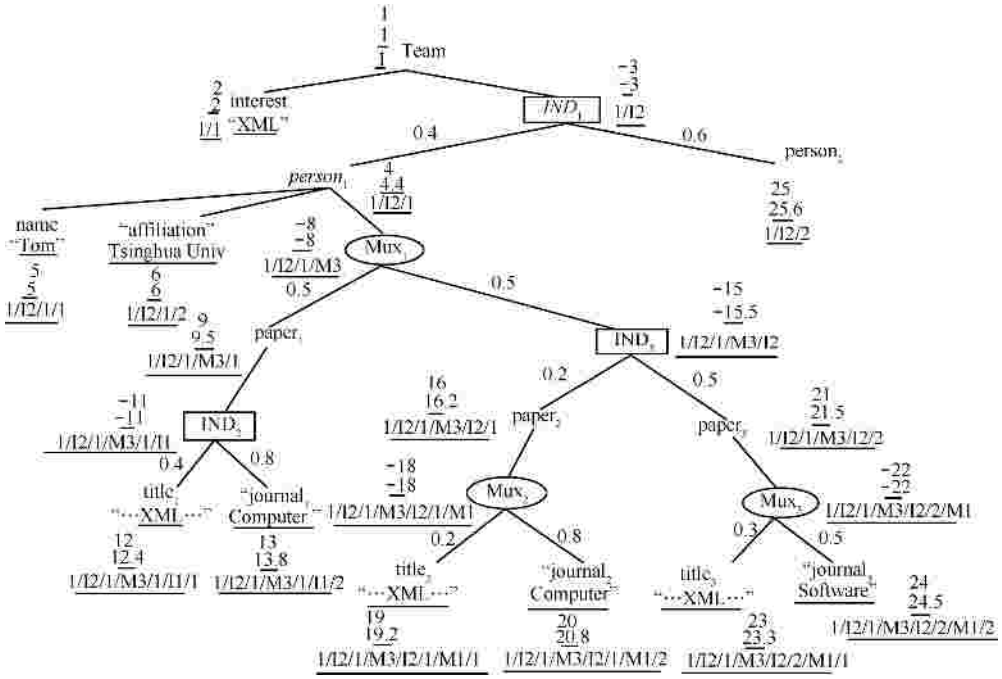


图 1　概率 XML 文档

当把所有节点的 Dewey 编码以及路径概率存储到磁盘上时，存在空间浪费的问题。

本文针对文献[10]中编码策略存在的时间和空间浪费问题，提出一种新的编码策略 DeweyTP。该编码策略在对节点进行编码时，将节点类型及路径概率合并到单一编码中，从而减少编码在查询时的时间消耗和存储到磁盘上的空间消耗。

2　背景

2.1　概率 XML 模型

一个概率 XML 文档表示一组普通 XML 文档的概率分布，其中一个普通文档对应一种可能世界。一个概率 XML 文档可以看作是一个标签树，该标签树由普通节点和分布节点组成。分布节点表示产生每一个普通 XML 文档的概率分布，其他节点为普通的数据节点。本文采用的概率 XML 树模型与文献[2]相同，其中分布节点包括独立节点和互斥节点。

例如，在图 1 中，方框表示独立节点，如 IND_1 节点，椭圆表示互斥节点，如 Mux_1 节点。在图 1 中，独立节点 IND_2 有 2 个孩子节点，$title_1$ 和 $journal_1$。当节点 IND_2 出现时，其孩子出现的条件概率分别为 0.4 和 0.8。因此，当节点 IND_2 出现时，节点 $title_1$ 和节点 $journal_1$ 都不出现的概率为(1- 0.4)×(1- 0.8) = 0.12，节点 $title_1$ 不存在而节点 $journal_1$ 存在的概率为(1- 0.4)×0.8 = 0.48。对于互斥节点 Mux_3，其孩子节点 $title_3$ 和 $journal_3$ 出现的条件概率分别为 0.3 和 0.5。根据互斥语义，当 Mux_3 存在时，至多有一个孩子节点存在。因此当节点 Mux_3 出现时，节点 $title_3$ 存在且节点 $journal_3$ 不存在的概率为 0.3，节点 $title_3$ 和 $journal_3$ 都不存在的概率为 0.2。

2.2　Dewey 编码

一个节点的 Dewey 编码由父亲节点的 Dewey 编码和其自身在兄弟节点间的顺序值构成，并以“/”隔开。在一个 Dewey 编码中，最后一个“/”之前的编码是父节点的编码，“/”之后的数字是其在兄弟节点间的顺序值。另外，根据节点的 Dewey 编码长度可以推出该节点的层次信息。在图 1 中，删除节点改进 Dewey 编码中字符后的编码就等于其 Dewey 编码。例如，节点 IND_2 的改进 Dewey 编码为 1/I2/1/M3/ 1/I1，则其 Dewey 编码为 1/2/1/3/1/1。根据节点 IND_2 的 Dewey 编码可以得出：①其父亲的 Dewey 编码为 1/2/1/3/1；②该节点在兄弟中的顺序值是 1；③该节点处于 XML 文档的第 3 层。

定义 1　（Dewey 的次序）给定 2 个 Dewey 编码 A: $a_1/a_2/\cdots/a_m$ 和 B: $b_1/b_2/\cdots/b_n$，当且仅当满足下面任意一个条件时，$A <_{Dewey} B$ 成立。

1)　$m < n$ 且 $a_1 = b_1, \perp, a_m = b_m$；

2)　$\exists t \le \min(m, n), a_1 = b_1, \perp, a_{t-1} = b_{t-1}, a_t < b_t$。

Dewey 的次序可以看成严格的文档顺序，例

如，$A <_{\text{Dewey}} B$ 表示编码 A 与编码 B 不同而且在先序遍历 XML 文档时，先访问编码 A 代表的节点，后访问编码 B 代表的节点。如果 A 和 B 是 2 个不同的 Dewey 编码，则一定有 $A <_{\text{Dewey}} B$ 或者 $B <_{\text{Dewey}} A$。可以根据下面的性质确定 A 与 B 之间的关系。

P_1（祖先后代关系）。当 $m < n$ 且 $a_1 = b_1,\cdots,a_m = b_m$ 时，A 是 B 的祖先。

P_2（父子关系）当 $m = n-1$ 且 $a_1 = b_1,\cdots,a_m = b_m$ 时，A 是 B 的父亲。

P_3（文档顺序）。当 $A <_{\text{Dewey}} B$ 时，表示在遍历文档时，A 在 B 之前被访问。

P_4（兄弟关系）。当 $m=n$ 且 $a_1=b_1,\cdots,a_{m-1}=b_{m-1},a_m?b_m$ 时，A 是 B 的兄弟。

3 DeweyTP 编码策略

和文献[10]的编码相比，本文提出的 DeweyTP 编码的最大特点是：单一编码同时支持路径概率和节点关系比较，且无需使用不同字符来区分节点类型。本节首先介绍同时支持节点关系检测和节点类型判断的 DeweyT 编码，然后介绍其扩展 DeweyTP 编码。

3.1 DeweyT 编码策略

使用 DeweyT 编码策略对概率 XML 文档中的节点进行编码时，首先先序遍历概率 XML 树，为每一个节点都分配一个支持节点类型检测的 ID 值，之后结合父亲节点的 DeweyT 编码得到该节点的 DeweyT 编码。例如，给出一个 DeweyT 编码 $a_1/a_2/\cdots/a_m$，可知其父亲的编码为 $a_1/a_2/\cdots/a_{m-1}$，该节点的 ID 值为 a_m。

使用正数表示普通节点（ORD），使用负数表示分布节点，其中负偶数表示互斥节点（MUX）、负奇数表示独立节点（IND），这样通过节点的 ID 值就可以区分不同类型的节点。例如，若节点的 ID 是-11，则表示该节点是独立节点。若节点的 ID 是-16，则表示该节点是互斥节点。下面给出推断节点 ID 的方法。

假设在先序遍历概率 XML 树时，当前被访问节点 v 的编号为 n，则 v 的 ID 值 ID_v 可根据式(1)推出。

$$ID_v=\begin{cases} n & ,\text{Type}=\text{ORD} \\ -n & ,\text{Type}=\text{MUX},n\%2=0 \\ -(n+1) & ,\text{Type}=\text{MUX},n\%2=1 \\ -(n+1) & ,\text{Type}=\text{IND},n\%2=0 \\ -n & ,\text{Type}=\text{IND},n\%2=1 \end{cases} \tag{1}$$

在式（1）中，Type 表示节点 v 的类型，若 Type = MUX，则表示节点 v 为互斥节点。在求出节点 v 的 ID 值后，需要继续遍历概率 XML 文档中的剩余节点。此时，紧接着被遍历节点的节点编号为 $|ID_v|+1$，之后再根据其节点类型求出该节点的 ID。在求出一个节点的 ID 后，根据式（1）可得到该节点的 DeweyT 编码。在图 1 中，在每一个节点旁边的 3 行数字中，第一行数字为该节点的 ID 值。

例如，在图 1 中，在先序遍历概率 XML 文档时，访问独立节点 IND_2 的编号为 10，由于该节点为独立节点，而且 10 是偶数，则 IND_2 的 ID 等于-11，其 DeweyT 编码为 1/-3/4/-8/9/-11。此时，下一个访问的节点是 $title_1$，其编号变成 12，由于 $title_1$ 是普通节点，则其 ID 就为 12。此时节点 $title_1$ 的 DeweyT 编码为 1/-3/4/-8/9/-11/12。根据节点 ID 值的求解方法，可以推出定理 1。

定理 1 假设节点 A 和 B 的 ID 值分别为 ID_A 和 ID_B，且 $A <_{\text{Dewey}} B$ 成立，则 $|ID_A|<|ID_B|$。

证明 假设依次访问节点 A 和节点 B 的编号为 m 和 n，$m>0$，$n>0$，根据 DeweyT 的编码策略，可以得出 $|ID_A|$ 的取值为 $\{m,m+1\}$，$|ID_B|$ 的取值为 $\{n,n+1\}$。根据 $|ID_A|$ 的取值，可以分为 2 种情况，当 $|ID_A|=m$ 时，节点 B 的编号 $n=m+1$，则有 $|ID_B|\geq m+1>m=|ID_A|$，即 $|ID_A|<|ID_B|$。同理可证，当 $|ID_A|=m+1$ 时，也存在 $|ID_A|<|ID_B|$。

由定理 1 可知，使用 DeweyT 编码策略对节点进行编码时，先访问节点的|ID|值小于之后访问节点的|ID|值。

3.2 DeweyTP 编码策略概述

DeweyTP 编码是 DeweyT 编码的扩展，它是由节点的 DeweyT 编码与路径概率结合得到。DeweyTP 编码策略为每一个节点指定 TP 值，之后结合父亲节点的 DeweyTP 编码得到该节点的 DeweyTP 编码。下面给出推断节点 TP 值的方法。

假设给出节点 v 的 ID 值 ID_v 和其条件概率 p_v，则 v 的 TP 值可根据式（2）推出。

$$TP_v=\begin{cases} ID_v & ,P_v=1 \\ ID_v+(-P_v) & ,P_v\neq 1,ID_v<0 \\ ID_v+P_v & ,P_v\neq 1,ID_v>0 \end{cases} \tag{2}$$

例如，在图 1 中，节点 IND_2 的 ID 值等于-11，其条件概率为 1，则 IND_2 的 TP 值等于-11。对于节点 $title_1$，其 ID 值等于 12，条件概率为 0.4，则

其 TP 值等于 12.4。对于独立节点 IND_3，其 ID 值等于- 15，条件概率为 0.5，则 IND_3 的 TP 值为- 15.5。由于根节点没有条件概率，因此其 TP 值和 ID 值相等。在图 1 中，每一个节点旁边的第 2 行数字为该节点的 TP 值。根据节点 TP 值的求解方法，可以推出定理 2。

定理 2　假设节点 A 和 B 的 TP 值分别为 TP_A 和 TP_B，且 $A <_{\text{Dewey}} B$ 成立，则$|TP_A| < |TP_B|$。

证明　假设节点 A 和节点 B 根据 DeweyT 编码策略得到的 ID 值分别为 ID_A 和 ID_B，对应的条件概率为 p_A 和 p_B，其中，p_A，$p_B \in (0,1]$。根据式(2)可知，由于节点 A 和 B 的 ID 值和其条件概率不定，因此需要考虑多种情况。这里只考虑一种情况，$ID_A > 0$，$ID_B > 0$，$p_A \in (0,1]$，$p_B = 1$。其他情况的证明和此情况的证明相似，这里不再说明。根据 p_A 的取值范围，可以得出其包含 2 种情况：$p_A = 1$；$p_A \in (0,1)$，而且这 2 种情况是互斥存在的。当条件 $p_A = 1$ 成立时，表示节点 A 的父亲是普通节点，即 $ID_A > 0$，$ID_B > 0$，$p_A = 1$，$p_B = 1$ 成立，则 $TP_A = ID_A$，$TP_B = ID_B$。根据定理 1，可得$|ID_A| < |ID_B|$，即$|TP_A| < |TP_B|$成立。当条件 $p_A \in (0,1)$时，表示节点 A 是独立或互斥节点，即 $ID_A > 0$，$ID_B > 0$，$p_A \in (0,1)$，$p_B = 1$ 成立，则 $TP_A = ID_A + p_A > 0$, $TP_B = ID_B > 0$。又根据定理 1，可得$|ID_A|+1 \le |ID_B|$，而 $p_A \in (0,1)$，则有 $ID_A + p_A < ID_B$ 成立，则 $TP_A < TP_B$，又因为 $TP_A > 0$，$TP_B > 0$，则$|TP_A| < |TP_B|$成立。

由定理 2 可知，使用 DeweyTP 编码策略对节点进行编码时，先访问节点的|TP|值小于之后访问节点的|TP|值。下面给出根据节点的 TP 值，推断节点 ID 值和条件概率的方法。

已知节点 v 的 TP 值 TP_v，则其 ID 值 ID_v 和其条件概率 p_v 可根据式（3）和式（4）推出。

$$ID_v = \begin{cases} TP_v & , \lfloor TP_v \rfloor = TP_v \\ \lfloor TP_v \rfloor + 1 , & TP_v < 0, \lfloor TP_v \rfloor \neq TP_v \\ \lceil TP_v \rceil - 1 , & TP_v > 0, \lfloor TP_v \rfloor \neq TP_v \end{cases} \tag{3}$$

$$p_v = \begin{cases} 1 & , \lfloor TP_v \rfloor = TP_v \\ \lfloor TP_v \rfloor + 1 - TP_v & , TP_v < 0, \lfloor TP_v \rfloor \neq TP_v \\ TP_v + 1 - \lceil TP_v \rceil & , TP_v > 0, \lfloor TP_v \rfloor \neq TP_v \end{cases} \tag{4}$$

在式（3）和式（4）中，当条件$\lfloor TP_v \rfloor = TP_v$成立时，表示 TP_v 是整数，反之，TP_v 是小数。例如，在图 1 中，节点 Mux_2 的 TP 值为- 18，而且- 18 是整数，则 Mux_2 的 ID 等于- 18，其条件概率为 1。对于节点 $title_2$，其 TP 值等于 19.2，由于 19.2 是小数且大于 0，则 $title_2$ 的 ID 值等于 19，其条件概率等于 0.2。对于节点 IND_3，其 TP 值等于- 15.5，由于- 15.5 是小数且小于 0，则 IND_3 的 ID 等于- 15，其条件概率等于 0.5。

定义 2　（DeweyTP 的先序次序）　给出 2 个节点 A 和 B，其 DeweyTP 编码分别为 $x_1/x_2/\cdots/x_m$ 和 $y_1/y_2/\cdots/y_n$，当$|x_m| \le |y_n|$时，$A \le_{\text{DeweyTP}} B$ 成立。

DeweyTP 的先序次序具有自反性和传递性。

定理 3　（DeweyTP 先序次序的自反性）给出节点 A 的 DeweyTP 编码 $x_1/x_2/\cdots/x_m$，则 $A \le_{\text{DeweyTP}} A$。

证明　由于 $x_m = x_m$ 成立，则根据定义 2，可知 $A \le_{\text{DeweyTP}} A$ 成立。

定理 4　（DeweyTP 先序次序的传递性）给出 3 个节点的 DeweyTP 编码，A：$x_1/x_2/\cdots/x_m$，B：$y_1/y_2/\cdots/y_n$ 以及 C：$z_1/z_2/\cdots/z_t$，若 $A \le_{\text{DeweyTP}} B$ 且 $B \le_{\text{DeweyTP}} C$，则 $A \le_{\text{DeweyTP}} C$。

证明　由于 $A \le_{\text{DeweyTP}} B$ 成立，则有$|x_m| \le |y_n|$成立。又因为 $B \le_{\text{DeweyTP}} C$ 成立，则有$|y_n| \le |z_t|$成立。综合 2 个不等式，得出$|x_m| \le |y_n| \le |z_t|$成立，即$|x_m| \le |z_t|$，由定义 2 可知 $A \le_{\text{DeweyTP}} C$ 成立。

定义 3　（相等关系）给出 2 个节点 A 和 B，其 DeweyTP 编码分别为 $x_1/x_2/\cdots/x_m$ 和 $y_1/y_2/\cdots/y_n$，当满足条件：$A \le_{\text{DeweyTP}} B$ 且 $B \le_{\text{DeweyTP}} A$ 时，编码 A 和 B 相等，记做 $A =_{\text{DeweyTP}} B$。

定理 5　给出 2 个节点，其 DeweyTP 编码分别为 A：$x_1/x_2/\cdots/x_m$ 和 B：$y_1/y_2/\cdots/y_n$，当满足条件 $x_m = y_n$ 时，$A =_{\text{DeweyTP}} B$。

证明　根据定义 3 可知，只需要证明 $A \le_{\text{DeweyTP}} B$ 且 $B \le_{\text{DeweyTP}} A$ 成立即可。根据定义 2 可知，只有当 $x_m = y_n$ 成立时，才能保证这 2 个条件同时成立。即当 $x_m = y_n$ 时，$A =_{\text{DeweyTP}} B$。

定义 4　（不等集合）出一个由 DeweyTP 编码组成的集合，而且每一个编码对应一个节点。在 DeweyTP 集合中，如果不存在任意 2 个 DeweyTP 编码相等，则该集合是不等集合。

定理 6　根据 DeweyTP 的编码策略，DeweyTP 编码形成的集合是不等集合。

证明　使用反证法证明。假设 DeweyTP 编码形成的集合不是不等集合。则根据定义 4 可知，存在 2 个表示不同节点的 DeweyTP 编码 A：$x_1/x_2/\cdots/x_m$ 和 B：$y_1/y_2/\cdots/y_n$，使得 $A =_{\text{DeweyTP}} B$ 成立。根据定理 5 可知，若 $x_m = y_n$ 成立，则 $A =_{\text{DeweyTP}} B$ 成立。

又因为当 $x_m=y_n$ 成立时，节点 A 和节点 B 是同一个节点，则这与假设矛盾，即使用 DeweyTP 编码策略为节点编码时，每一个节点对应的 DeweyTP 编码都是不同的。

定义 5 （DeweyTP 的次序）给出 2 个 DeweyTP 编码，A：$x_1/x_2/\cdots/x_m$ 和 B：$y_1/y_2/\cdots/y_n$，如果 $A\leq_{\text{DeweyTP}} B$ 且 $A\ ?_{\text{DeweyTP}}\ B$ 成立，则 $A<_{\text{DeweyTP}} B$。

定理 7 给出 2 个 DeweyTP 编码，A：$x_1/x_2/\cdots/x_m$ 和 B：$y_1/y_2/\cdots/y_n$，如果 $|x_m|<|y_n|$，则 $A<_{\text{DeweyTP}} B$。

证明 根据定义 5 可知，要证明 $A<_{\text{DeweyTP}} B$ 成立，只需要证明当 $|x_m|<|y_n|$ 成立时，编码 A 和 B 满足条件 $A\leq_{\text{DeweyTP}} B$ 且 $A\ ?_{\text{DeweyTP}}\ B$ 即可。根据定义 2，当条件 $|x_m|<|y_n|$ 成立时，$A\leq_{\text{DeweyTP}} B$ 成立。根据定理 5 可知，当条件 $|x_m|<|y_n|$ 成立时，$A\ ?_{\text{DeweyTP}}\ B$。即当 $|x_m|<|y_n|$，则 $A<_{\text{DeweyTP}} B$。

根据定理 7 可以得出，如果编码 A 和 B 是不等集合中的 2 个不等的 DeweyTP 编码，则肯定有 $A<_{\text{DeweyTP}} B$ 或 $B<_{\text{DeweyTP}} A$。可以根据下面的性质确定 A 和 B 的关系。

P_1（祖先后代关系）。当 $m<n$ 而且 $x_m=y_m$ 时，A 是 B 的祖先。

P_2（父子关系）。当 $m=n-1$ 而且 $x_m=y_m$ 时，A 是 B 的父亲。

P_3（文档顺序）。当 $A<_{\text{DeweyTP}} B$ 时，表示在遍历文档时，A 在 B 之前被访问。

P_4（兄弟关系）。当 $m=n$ 而且 $x_{m-1}=y_{n-1}$ 而且 $x_m?y_n$ 时，A 是 B 的兄弟。

4 实验

4.1 实验环境

实验使用的机器配置为 Intel Core 2 Duo E7500 2.93 GHz CPU，2 GB 内存，操作系统为 Windows XP。为了方便描述，本文称文献[10]中的编码策略为 DeweyTypePro。在实验中，所有的算法都用 Visual C++实现。运行时间是每个算法循环 20 次后的平均时间。

实验使用的 XML 数据集为 XMark，其大小分别为 10 MB、20 MB、40 MB。本文使用文献[6]提到的方法对普通的 XML 文档进行转换，生成对应的概率 XML 文档。具体方法是，首先先序遍历传统的 XML 文档，每访问一个节点 v 时，都随机产生一些分布节点，如独立节点或者互斥节点，并把这些分布节点作为 v 的孩子。之后，随机选择 v 在原始 XML 文档中的孩子节点，作为新生成的分布节点的孩子节点，并为这些孩子节点随机指定其条件概率。对于互斥节点，其孩子节点的条件概率之和不能超过 1。对于每一个数据集，分布节点在总节点中占有的比例大约为 20%~30%。表 1 给出在实验中使用的概率 XML 文档的相关信息。

表 1 概率 XML 文档的节点信息

大小	独立节点个数	互斥节点个数	普通节点个数
10 MB	25 192	25 197	167 865
20 MB	50 419	50 373	336 244
40 MB	90 173	90 070	601 550

实验比较的标准是在存储 2 种编码时需要的磁盘空间以及求解节点间最低公共祖先（LCA）关键字分布的时间消耗。

在概率 XML 文档中，附属在同一个叶子节点下所有文本值的 DeweyTP 编码都相等。为了防止在磁盘中多次存储相同 DeweyTP 编码，需要对 DeweyTP 编码进行压缩存储。在存储文本的 DeweyTP 编码时，使用文本的 ID 值替代其 DeweyTP 编码，其中该 ID 值是根据 DeweyT 编码策略产生的。另外，为了还原文本的 DeweyTP 编码，还需要将文本的 ID 和其 DeweyTP 编码的对应关系存储到磁盘。由于一个 Dewey 编码唯一标识一个节点，因此无法对文献[10]中的 DeweyTypePro 编码进行压缩存储。表 2 给出在实验中存储 2 种编码的文件信息。

表 2 存储不同编码的文件信息

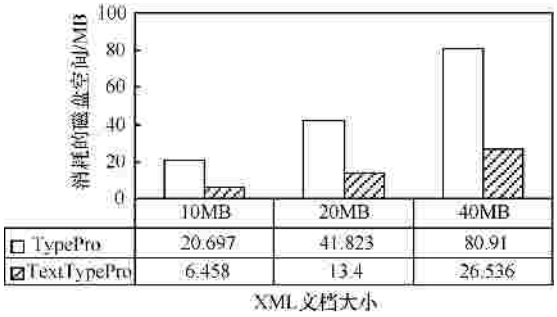
编码策略	文件	文件存储的内容
DeweyTP	TypePro	压缩后的 DeweyTP 编码
	TextType-Pro	文本的 ID 和其 DeweyTP 编码的对应关系
DeweyTypePro	Type	改进的 Dewey 编码
	Pro	路径概率

关键字分布的求解是在概率 XML 上进行关键字查询的重要操作[10]，高效的编码策略是提高查询效率的主要手段之一。根据节点的关键字分布，可以直接得到该节点成为 SLCA[11,12]或 ELCA[13]的概率，进而避免产生概率 XML 文档的可能世界。在求解节点的关键字分布时，需要用到节点的类型和路径概率。对于不同类型的节点，其关键字分布的求解方法也不同[10]。实验分别使用 2 种编码来计算

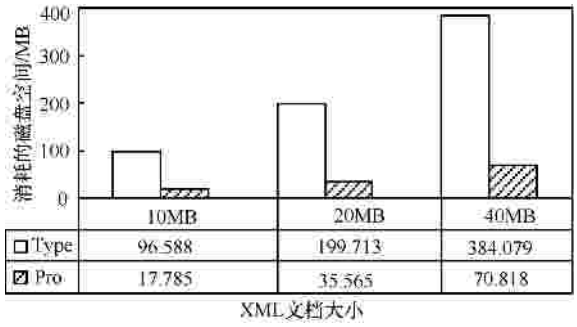
节点间最低公共祖先的关键字分布，并通过比较其消耗的时间来评价新编码的查询性能。

4.2　实验结果及分析

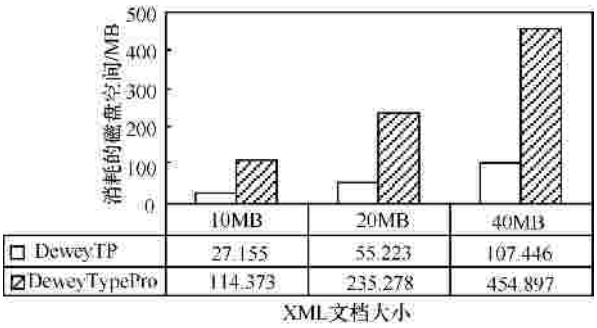
图 2 给出了存储 DeweyTP 编码和 DeweyTypePro 编码时需要的磁盘空间。实验中，根据 2 种编码策略，分别对 4 种不同大小的 XML 文档进行编码，并将编码存储在磁盘中。



(a) 存储 DeweyTP 编码的空间消耗



(b) 存储 DeweyTypePro 编码的空间消耗



(c) 存储 DeweyTP 编码和 DeweyTypePro 编码的空间消耗

图 2　不同编码存储空间的比较

图 2(a)给出了存储不同概率 XML 文档中节点的 DeweyTP 编码时，其对应文件分别消耗的磁盘空间，其中，*X* 轴表示普通 XML 文档的大小，*Y* 轴表示文件 TypePro 和 TextTypePro 消耗的磁盘空间。从图 2(a)可以看出，随着 XML 文档大小的成倍增加，文件 TypePro 和 TextTypePro 消耗的磁盘空间也成倍增加。对于每一个概率 XML 文档，由于文件 TypePro 中内容包含文件 TextTypePro 中的内容，因此文件 TypePro 消耗的磁盘空间总是大于文件 TextTypePro 消耗的磁盘空间。

图 2(b)给出了存储不同概率 XML 文档中节点的 DeweyTypePro 编码时，其对应文件分别消耗的磁盘空间，其中，*X* 轴表示普通 XML 文档的大小，*Y* 轴表示文件 Type 和 Pro 消耗的磁盘空间。从图 2(b)可以看出，随着 XML 文档大小的成倍增加，文件 Type 和 Pro 消耗的磁盘空间也是成倍增加。对于每一个概率 XML 文档，文件 DeweyType 占用的磁盘空间总是大于文件 PathProb 占用的存储空间。这是因为叶子节点以及附属在叶子节点下文本的路径概率是一样的，因此在文件 Pro 中仅仅存储节点的路径概率，而在文件 Type 存储的是节点和文本值的改进 Dewey 编码，所以文件 DeweyType 占用的磁盘空间总是大于文件 PathProb 占用的存储空间。

图 2(c)给出了存储不同概率 XML 文档中节点的 DeweyTP 和 DeweyTypePro 编码时，分别消耗的磁盘空间，其中，*X* 轴表示普通 XML 文档的大小，*Y* 轴表示 2 种编码分别消耗的磁盘空间。存储编码 DeweyTP 和 DeweyTypePro 消耗的磁盘空间分别是图 2(a)和图 2(b)中 2 个文件占用的磁盘空间之和。从图 2(c)可以看出，随着 XML 文档大小的成倍增加，2 种编码消耗的磁盘空间也是成倍增加的。对于每一个概率 XML 文件，存储 DeweyTypePro 编码所消耗的磁盘空间要比存储 DeweyTP 编码所消耗的磁盘空间多 4 倍。这是由于 DeweyTP 编码对节点进行编码时考虑到了节点的类型和路径概率，并对文本对应的 DeweyTP 编码进行了压缩存储，因此与 DeweyTypePro 编码相比，DeweyTP 编码占用的磁盘空间更少。

在求解节点的关键字分布时，需要根据 DeweyTP 和 DeweyTypePro 编码策略，获得所有元素的 2 种编码集合。在每一个集合中，编码是按照文档顺序排列的。实验是从所有元素的编码集合中选出 2 个元素的编码集合，之后，分别在 2 个编码集合中取出一个编码，并求其最低公共祖先的关键字分布。

图 3 给出了根据 DeweyTP 编码和 DeweyTypePro 编码计算节点关键字分布时的时间消耗。其中，*X* 轴表示编码集合中编码的个数，*Y* 轴表示计算所有节点对的最低公共祖先的关键字分布的时间消耗。从图 3 可以看出，随着编码个数的成倍增加，

计算最低公共祖先关键字分布的时间消耗也成倍增加。求解节点间最低公共祖先的关键字分布时，DeweyTP 编码要比 DeweyTypePro 编码高效。这是因为使用 DeweyTypePro 编码策略对节点进行编码时，没有把节点的类型与路径概率融合到单一编码中，而导致在求解节点的关键字分布时，需要多次探测散列表，获得节点的路径概率。另外，由于一个 Dewey 编码唯一标识一个节点，在使用 DeweyTypePro 编码进行查询时，需要比较节点的整个编码。因此，与 DeweyTP 编码相比，在求解节点间最低公共祖先的关键字分布时，使用 DeweyTypePro 编码耗时更多。

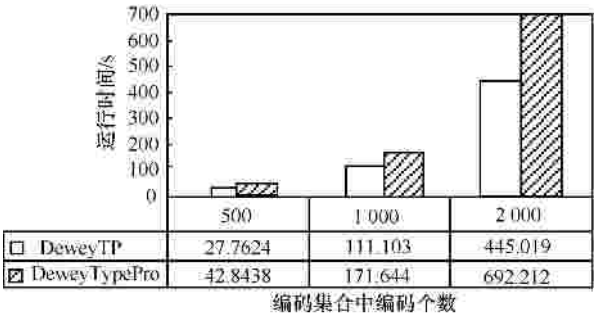


图3 不同编码计算关键字分布时的时间比较

5 结束语

针对已有编码处理概率 XML 数据的时间和空间浪费问题，本文提出了一种新编码策略 Dewey TP，在对节点进行编码时，同时考虑到节点的类型和路径概率，从而减少编码在查询时的时间消耗和存储到磁盘的空间消耗。最后的实验结果表明，本文提出的编码策略比已有的编码策略更加高效。

参考文献：

[1] SENELLART P, ABITEBOUL S. On the complexity of managin probabilistic XML data[A]. The 28th ACM SIGACT-SIGMOD- SIGART Symposium on Principles of Database Systems[C]. 2007. 283-292.

[2] NIERMAN A, JAGADISH H V. ProTDB: probabilistic data in XML[A]. The 28th International Conference on Very Large Data Bases[C]. 2002. 646-657.

[3] HUNG E, GETOOR L, SUBRAHMANIAN V S. Pxml: a probabilistic semistructured data model and algebra[A]. The 19th International Conference on Data Engineering[C]. 2003. 467-478.

[4] KEULEN M V, KEIJZER A D, ALINK W. A probabilistic XML approach to data integration[A]. The 21st International Conference on Data Engineering[C]. 2005. 459-470.

[5] ABITEBOUL S, KIMELFELD B, SAGIV Y, *et al*. On the expressiveness of probabilistic XML models[J]. The International Journal on Very Large Data Bases, 2009, 18(5):1041-1064.

[6] KIMELFELD B, KOSHAROVSKY Y, SAGIV Y. Query efficiency n probabilistic XML models[A]. The ACM SIGMOD International Conference on Management of Data[C]. 2008. 701-714

[7] ABITEBOUL S, ALSTRUP S, KAPLAN H, *et al*. Compact labeling scheme for ancestor queries[J]. SIAM J Comput 2006, 35(6): 1295-1309.

[8] COHEN E, KAPLAN H, MILO T. Labeling dynamic XML trees[A]. The Symposium on Principles of Database Systems[C]. 2002. 271-281.

[9] TATARINOV I, VIGLAS S, BEYER K S, *et al*. Storing and querying ordered XML using a relational database system[A]. The ACM SIGMOD International Conference on Management of data[C]. 2002. 204-215.

[10] LI J X, LIU C F, ZHOU R, *et al*. Top-*k* keyword search over probabilistic XML data[A]. The 27th International Conference on Data Engineering[C]. 2011.673-684.

[11] XU Y, PAPAKONSTANTINOU Y. Efficient keyword search for smallest LCAs in XML databases[A]. The ACM SIGMOD International Conference on Management of Data[C]. 2005. 537-538.

[12] SUN C, CHAN C Y, GOENKA A K. Multiway slca-based keyword search in xml data[A]. The 16th International Conference on World Wide Web[C]. 2007.1043-1052.

[13] ZHOU J F, BAO Z F, WANG W, *et al*. Fast SLCA and ELCA computation for XML keyword queries based on set intersection[A]. The 28th International Conference on Data Engineering[C]. 2012. 905-916.

作者简介：

陈子阳（1973-），男，黑龙江五常人，燕山大学教授、博士生导师，主要研究方向为数据库理论与系统等。

刘佳（1978-），女，黑龙江鹤岗人，燕山大学博士生，主要研究方向为 XML 关键字查询等。

张刘辉（1987-），男，河南周口人，燕山大学硕士生，主要研究方向为 XML 关键字查询等。

周军锋（1983-），男，陕西西安人，燕山大学副教授，主要研究方向为 XML 数据库、XML 关键字查询和字符串相似匹配等。