doi:10.3969/j.issn.1005-3026.2017.02.010

模糊XML文档与模糊DTD相似性研究

赵 震[1,2], 马宗民[1]
(1. 东北大学 计算机科学与工程学院, 辽宁 沈阳 110819; 2. 渤海大学 信息科学与技术学院, 辽宁 锦州 121013)

摘 要: 在模糊XML数据管理中,模糊XML文档和模糊DTD的相似性是模糊XML数据整合、模糊XML文档聚类的关键步骤.为了研究模糊XML文档和模糊DTD的相似性,对模糊DTD树进行了规则变换,主要解决元素和属性的析取约束和基数约束问题,即由析取范式转化为合取范式,将元素或属性的重复次数确定化,然后利用树编辑距离算法对模糊XML文档树和转化后的模糊DTD树集合进行相似性对比.通过实验验证了所提方法的性能优势.

关 键 词: 模糊XML文档;文档类型定义(DTD);相似性;结构匹配;数据整合
中图分类号: TP 311 **文献标志码**: A **文章编号**: 1005-3026(2017)02-0200-05

Research on the Similarity of Fuzzy XML Documents and Fuzzy DTD

ZHAO Zhen[1,2], *MA Zong-min*[1]
(1. School of Computer Science & Engineering, Northeastern University, Shenyang 110819, China; 2. College of Information Science and Technology, Bohai University, Jinzhou 121013, China. Corresponding author: ZHAO Zhen, E-mail: zhaozhen@bhu.edu.cn)

Abstract: In fuzzy extensible markup language (XML) data management, the similarity between fuzzy XML document and fuzzy document type definition (DTD) is a key step of fuzzy XML data integration and fuzzy XML documents clustering. In order to study the similarity, the fuzzy DTD tree are transformed by rules, which mainly solves the disjunctive constraint and cardinality constraint problems of the elements and attributes, namely the transformation from disjunctive normal form into conjunctive normal form, thus the number of repetitions of elements or attributes being determined. And then, the tree edit distance algorithm is used to compare the similarity between the fuzzy XML document tree and the transformed fuzzy DTD tree. The advantages of the proposed method are verified by experiments.

Key words: fuzzy XML documents; DTD (document type definition); similarity; structure matching; data integration

近年来,随着互联网的快速发展,XML作为Web信息表示的事实标准越来越受到关注.XML的数据管理包括数据的整合和存储、信息的交换等.由于XML的数据源相互独立,不同应用中XML的数据结构有差异,而管理这些数据的需求又不断增多,这就要求对这些数据进行识别,找出它们之间的相似性后再进行整合等操作.非模糊的XML数据管理问题是以往各国学者研究的热点[1-3],主要研究XML文档和文档类型定义(DTD)之间的相似性[1-2],以及XML文档之间的相似性[3].这些经典方法的提出,对于解决XML数据管理问题十分重要.

在经典XML文档和DTD的相似性研究中,XML文档可以表示为树,而DTD是一组规则表达式,也可以表示为树,但是两者的相似性问题比两棵树的匹配问题复杂得多,目前的解决方案主要有以下几种策略.第一,对于需要进行匹配的XML文档X和文档类型定义D,D中的某些属

收稿日期: 2015-09-24
基金项目: 国家自然科学基金资助项目(61370075); 教育部新世纪优秀人才支持计划项目(NCET-05-0288).
作者简介: 赵 震(1977-),男,辽宁锦州人,东北大学博士研究生; 马宗民(1965-),男,山东金乡人,东北大学教授,博士生导师.

性和元素可能不出现在 X 中，而 X 也可能包含一些 D 中没有的属性或元素；因此，文献[1]通过计算出现在 D 中而不出现在 X 中的元素和出现在 X 中而不出现在 D 中的元素的数目来进行相似性比较. 第二，由于文档类型定义可以描述为扩展的与内容无关的语法，因此使用一个自动机来表示 D，问题可转化为度量自动机和文档树之间的编辑距离[4-5].

然而，实际应用中很多信息都是不确定的，也就是模糊的. 随着模糊信息不断出现在许多实际应用中，这些模糊信息也要相应地用模糊 XML 来表示，这使得对模糊 XML 数据的管理十分必要，它已经成为当前新的研究方向[6]. 研究模糊 XML 与模糊 DTD 的相似性问题对于模糊数据的整合至关重要[7]，但是目前还没有针对模糊 XML 文档与模糊 DTD 匹配问题的相关研究.

本文基于模糊 XML 文档和模糊 DTD 数据模型，将二者转化为树；根据模糊 DTD 的特性，在对模糊 DTD 中元素与属性约束进行转化处理后，提出了计算模糊 XML 文档与模糊 DTD 相似性的匹配算法，对相似性进行计算. 最后用实验验证了该方法的正确性和有效性.

1　模糊 XML 与模糊 DTD 数据模型

1.1　模糊 XML 文档及树形表示

为了表示模糊 XML 文档中的模糊信息，使用基于“隶属度和可能性分布”的模糊 XML 文档的表示模型[8]. 在这个模型中，一个元素可以有相关的隶属度. 元素的隶属度意味着成为其父亲的孩子节点的可能性. 而元素的属性值可以用概率分布来表示，并且这些值可以是析取的，也可以是合取的. 下面给出一个模糊 XML 文档片段，如图 1 所示.

模糊 XML 文档可以用树形结构来表示. 按照 DOM[9]模型，一个模糊 XML 文档也可以表示为一个单根的有序标签树，其中的节点对应文档中的元素和属性. 本文只比较树的结构相似性，所以省略元素和属性的值. 图 1 中文档的树结构如图 2 所示.

1.2　模糊 DTD 及树形表示

模糊 DTD 作为模糊 XML 文档的语法结构，描述了模糊 XML 文档的结构框架. 与非模糊 DTD 不同的是，模糊 DTD 引入了模糊构造子 Dist，Val，Poss，Type. 下面给出图 1 中模糊 XML 文档对应的模糊 DTD，如图 3 所示.

```
< college CName = “NEU” >
  < Val Poss = 0. 8 >
    < department DName = “IST” >
      < student SID = “20130425” >
        < age >
          < Dist Type = “disjunctive” >
            < Val Poss = 0. 8 > 26 < /Val >
            < Val Poss = 0. 9 > 28 < /Val >
            < Val Poss = 0. 8 > 29 < /Val >
          < /Dist >
        < /age >
        < email >
          < Dist Type = “conjunctive” >
            < Val Poss = 0. 6 > John@ yahoo. com < /Val >
            < Val Poss = 0. 8 > John@ qq. com < /Val >
            < Val Poss = 0. 5 > john@ sina. com < /Val >
          < /Dist >
        < /email >
      < /student >
    < /department  >
  < /Val >
< /college >
```

图 1　模糊 XML 文档实例

Fig. 1　Sample of a fuzzy XML document

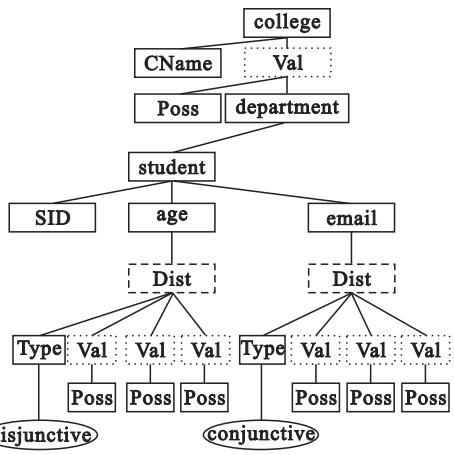

图 2　模糊 XML 文档树实例

Fig. 2　Sample of a fuzzy XML document tree

```
< ! ELEMENT college ( Val + ) >
< ! ATTLIST college CName IDREF #REQUIRED >
< ! ELEMENT Val ( department * ) >
< ! ATTLIST Val Poss CDATA “1. 0” >
< ! ELEMENT department ( student * ) >
< ! ATTLIST department DName IDREF #REQUIRED >
< ! ELEMENT student ( age?, email? ) >
< ! ATTLIST student SID IDREF #REQUIRED >
< ! ELEMENT age ( Dist ) >
< ! ELEMENT Dist ( Val + ) >
< ! ATTLIST Dist type ( disjunctive ) >
< ! ELEMENT email ( Dist ) >
< ! ELEMENT Dist ( Val + ) >
< ! ATTLIST Dist type ( conjunctive ) >
< ! ELEMENT Val ( #PCDATA ) >
< ! ATTLIST Val Poss CDATA “1. 0” >
```

图 3　模糊 DTD 实例

Fig. 3　Sample of a fuzzy DTD

与模糊 XML 文档一样,模糊 DTD 也可以用树形结构来表示. 图 3 中模糊 DTD 的树结构如图 4 所示.

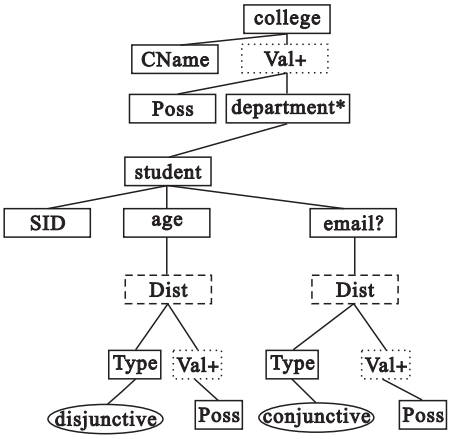

图 4 模糊 DTD 树实例

Fig. 4 Sample of a fuzzy DTD tree

2 模糊 DTD 树的转换规则

由于模糊 DTD 中包含基数约束和析取约束,所以无法将其直接与模糊 XML 文档树进行相似性比较,需要对这些约束条件进行转换处理.

2.1 析取约束的转换

析取约束"|",表示该符号前后元素或属性不能同时出现,"|"即 OR 运算符. 如果模糊 DTD 中包含"|"运算符,需要将其转换为多个不包含"|"的 DTD 集合. 例如:表达式 <! ELEMENT a(b,(c|d))> 可以分解为 <! ELEMENT a(b,c))> 和 <! ELEMENT a(b,d)>两个表达式,分别对应两个 DTD. 这一过程称为析取分解过程. 用规则 1 来表示.

规则 1:处理 D 中析取约束"|",对"|"两边的元素或属性进行选择,形成多个不包含"|"符号表达式的 d,从而构成 DTD 集合 Dset, d 为 Dset 中的 DTD.

特殊地,对于模糊构造子 Val,若该 Val 的父节点 Dist 下 Type 值为 disjunctive,表示 Dist 下的 Val 子树是不能同时出现的,相当于析取约束,即需要根据 Type 下的值判断 Dist 下的 Val 子树的个数. 因为 Val 下子树表示的是属性和它的值,一般地,各个子树结构是相同的. 为了不增加将来相似性比较的复杂度,本文选择只保留一个 Val 子树.

2.2 基数约束的转换规则

模糊 DTD 中元素和属性的基数约束"*","+","?"是用来说明所约束元素或属性的可重复次数. 如果用 e 来表示元素或属性,则 e* 表示 e 可以重复 0 到无限次,e+ 表示 e 可重复 1 次到无限次,e? 表示 e 可重复 0 或 1 次. 对于基数约束组合,可以用下面的原则来转换,以达到简化的目的.

$e++\rightarrow e+$,

$e**\rightarrow e*$,

$e*+\rightarrow e*$,

$e*?\rightarrow e*$,

$e?+\rightarrow e*$,

$e??\rightarrow e?$.

也就是说,所有的基数约束组合最终都可以转化为 e*,e+,e?. 需要确定的是具有基数约束的元素具体的重复次数. 可以根据该元素或属性在相比较的模糊 XML 文档树中有相同父亲的相同元素或属性的重复次数作为最终的重复次数,即模糊 DTD 中具有基数约束的元素或属性 e 的重复次数等于与之相比较的模糊 XML 树中有相同父亲的元素或属性 e'的重复次数,用 repeat(x. e')来表示. 其中,repeat 表示重复次数,可以用计数器来实现,x. e'表示在模糊 XML 树 x 中有相同父亲的元素 e'. 这使得模糊 DTD 的基数约束能自动适应与之比较的模糊 XML 文档.

综上所述,提出基于这些转换规则的将 DTD 转换为树的算法,算法的描述如下.

```
算法 DTDtoTree(X,d)
输入: X //XML 文档
      d //DTD
输出: DtreeSet
1  If (X 和 d 非空)
2    {For(int i=0;i<d.childcount;i++)
3       { If (d.child[i]约束为 * 或 +,?)
4         {repeat(X.d.child[i])}
5       else {d.child[i]保持不变}
6       DTDtoTree (X.child[i],d.child[i])
7       }
8    }
9  return DtreeSet
```

3 模糊 XML 文档与模糊 DTD 树的相似性比较

在明确模糊 DTD 树的转换规则后,本文提出了模糊 XML 文档与模糊 DTD 树相似性计算方法;本方法与其他方法不同之处在于本文所提出

的方法是专门处理模糊 XML 与模糊 DTD 之间的相似性. 比较模糊 XML 文档与模糊 DTD 树的相似性,首先要将模糊 XML 文档转化为对应的树,模糊 DTD 转化为对应的树集合,然后利用树编辑距离算法计算两者之间的相似性. 算法具体描述如下.

算法 Similrity_Doc_DTD(X,D)

输入：　X //XML 文档

　　D //DTD

输出：Sim(X,D)//相似度

1　Xtree = DoctoTree(X)

2　$D \xrightarrow{\text{规则 1}} DTDset$

3　For($i=1,i<$ | DTDset |,$i++$)

4　　{DtreeSet = DTDtoTree(DTDset[i],X) }

5　Dist[ ] = new[|DtreeSet |]

6　For($i=1,i<$ |DtreeSet |,$i++$)

7　　{Dist[i] = TED_{Doc_DTD}(Xtree, DtreeSet[i]) }

8　Return $Sim(X,D) = Max\left\{\frac{1}{1+Dist[i]}\right\}$

很明显,该算法中分别计算模糊 XML 文档树与模糊 DTD 树集合中的树的编辑距离,编辑距离越小的相似性越大,最后在结果数组中选取最大相似度作为最终相似结果. 树编辑距离算法是计算两棵树之间的相似性的经典算法[10]. 该算法可以很好地实现文中模糊 XML 树与模糊 DTD 树的结构相似性. 算法的描述如下.

算法 TED_{Doc_DTD}(Xtree, Dtree)

输入：Xtree, Dtree

输出：TED (Xtree, Dtree)

1　M = Degree(Xtree)　　//Xtree 的一级子树的度

2　N = Degree(Dtree)　　//Dtree 的一级子树的度

3　Dist[ ][ ] = new[0,…,M][0,…,N]

4　If (Xtree 与 Dtree 的根节点匹配)

5　　{Dist[0][0] =0}　　//匹配节点编辑距离为 0

6　Else

7　　{Dist[0][0] =1 }

8　For (i=1 to M)//$Xtree_i$ 表示 Xtree 的第 i 个孩子

9　　{Dist[i][0] = Dist[i − 1][0] + CostDelTree ($Xtree_i$)}

10　For(j=1 to N)//$Dtree_j$表示 Dtree 的第 j 个孩子

11　　{Dist[0][j] = Dist[0][j −1] + CostInsTree ($Dtree_j$)}

12　For (i=1 to M)

13　　{ For (j=1 to N)

14　　　{Dist[i][j] = Min{

15　　　　Dist[$i-1$][$j-1$] + TED($Xtree_i$,$Dtree_j$),

16　　　　Dist[$i-1$][j] + Cost($Xtree_i$,“delete”),

17　　　　Dist[i][$j-1$] + Cost($Dtree_j$,“insert”) }}

18　　}

19　Return Dist[M][N]

本文方法的特点在于处理对象的模糊特征,即 XML 文档 和 DTD 都是模糊的. 相对于经典的 XML 与 DTD 的匹配问题,这一特点无疑增加了处理的复杂度. 此外,本方法合理扩展了经典 XML 与 DTD 的匹配方法,利用基于树编辑距离的算法达到了匹配的目的. 由于本方法是基于目前处理树匹配问题最有效的方法进行的扩展,因此具有先天的性能优势.

4　实　　验

本文选用真实数据集对算法进行测试,以进一步评估本文方法的性能. 同时通过实验对本文提出的模糊 XML 文档和模糊 DTD 相似性的计算方法与非模糊 XML 文档和 DTD 的相似性计算方法进行了性能对比. 下面简单介绍主要的评估指标以及它们的定义.

A 是正确匹配并被识别的模糊 XML 文档与模糊 DTD 数量;B 是错误匹配并被识别的模糊 XML 文档与模糊 DTD 数量;C 是正确匹配但是没有被识别出的模糊 XML 文档与模糊 DTD 数量.

P 是精确度,即正确匹配的程度：

$$P = \frac{A}{A+B}.$$

R 是召回率,即匹配的完整性：

$$R = \frac{A}{A+C}.$$

两者的调和平均值用 F-measure 来表示：

$$F\text{-measure} = \frac{2PR}{P+R}.$$

精确度和召回率越高,匹配方法越有效.

为保证数据的真实性和可信度,本文所用的数据集为 DBLP 和 SigmodRecord. 需要在这些数据集中加入模糊算子,使其成为模糊数据集;另外,需要将这些数据集分解为 0.1 MB 到 2 MB 的不同大小的数据,以便对比不同大小的数据的算法响应时间.

图 5 显示了在 DBLP 和 SigmodRecord 数据集上使用文中提出的方法所得到的精确度、召回率和 F-measure 值的对比情况. 可以看出,SigmodRecord 数据集上的有效性要好于 DBLP

数据集,这与 DBLP 数据集结构较复杂有关.

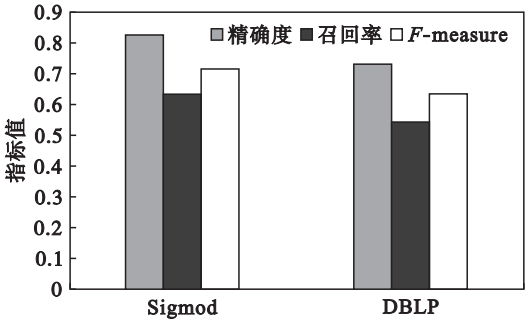

图 5 DBLP 和 SigmodRecord 数据集匹配的有效性对比

Fig. 5 Matching effectiveness comparison between DBLP and SigmodRecord datasets

图 6 显示了在 DBLP 和 SigmodRecord 数据集上运行本文方法所得到的响应时间对比. 可以看出,SigmodRecord 数据集上的响应时间远少于 DBLP 数据集,这与 DBLP 数据集较大有关.

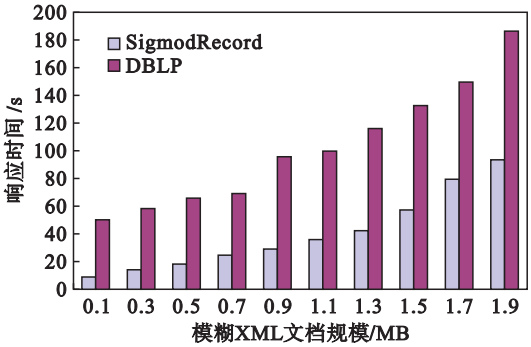

图 6 DBLP 和 SigmodRecord 数据集响应时间对比

Fig. 6 Response time comparison between DBLP and SigmodRecord datasets

图 7 是本文提出的模糊 XML 与模糊 DTD 树相似计算方法和非模糊 XML 与非模糊 DTD 相似性计算方法的响应时间对比. 由图 7 可以看出,本文提出的方法响应时间长于非模糊方法,这是因为模糊数据集加入了模糊算子,增加了处理复杂度. 另外,随着文档不断增大,两种方法响应时间差距有缩小趋势.

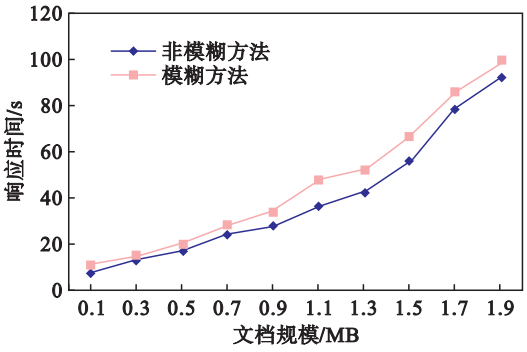

图 7 模糊与非模糊方法响应时间对比

Fig. 7 Response time comparison between fuzzy and non-fuzzy methods

5 结 语

本文基于模糊 XML 与模糊 DTD 模型,根据模糊 DTD 析取约束和基数约束的特性,首先提出了模糊 DTD 模型的析取约束和基数约束转换规则,然后研究了模糊 XML 与模糊 DTD 的相似性,给出了相似性比较的匹配算法. 用具体实例说明了比较过程,并通过实验验证了所提出方法的正确性和有效性.

参考文献:

[1] Bertino E, Guerrini G, Mesiti M. Measuring the structural similarity among XML documents and DTDs[J]. *Journal of Intelligent Information Systems*, 2008, 30(1): 55 – 92.

[2] Amavi J, Bouchou B, Savary A. On correcting XML documents with respect to a schema[J]. *The Computer Journal*, 2014, 57(5): 639 – 674.

[3] Tekli J, Chbeir R. A novel XML document structure comparison framework based-on sub-tree commonalities and label semantics[J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012, 11(1): 14 – 40.

[4] Ng P K L, Ng V T Y. Structural similarity between XML documents and DTDs[M]. Berlin: Springer, 2003: 412 – 421.

[5] Canfield E R, Xing G. Approximate XML document matching[C]//Proceedings of the 2005 ACM Symposium on Applied Computing. New York: ACM, 2005: 787 – 788.

[6] Zhao X, Bi X, Wang G, et al. Uncertain XML documents classification using extreme learning machine[M]. Berlin: Springer, 2015: 51 – 60.

[7] Liu J, Zhang X X. Data integration in fuzzy XML documents[J]. *Information Sciences*, 2014, 280(1): 82 – 97.

[8] Ma Z M, Yan L. Fuzzy XML data modeling with the UML and relational data models[J]. *Data & Knowledge Engineering*, 2007, 63(3): 972 – 996.

[9] World Wide Web Consortium. The document object model[EB/OL]. [2015 – 08 – 12]. http://www.w3.org/DOM.

[10] Nierman A, Jagadish H V. Evaluating structural similarity in XML documents[C]//Proceedings of the ACM SIGMOD International Workshop on the Web and Databases. Madison, 2002: 61 – 66.