DOI:10.13245/j.hust.2012.08.015

基于标签序列的半结构化数据相似度度量

张利军 李战怀 李 宁 李 霞

(西北工业大学计算机学院,陕西 西安 710072)

摘要 针对基于路径的半结构化数据结构相似度度量方法不能很好地处理路径部分相似以及忽略了元素之间兄弟关系的问题,提出一种基于频繁关联标签序列的结构相似度度量方法,该方法将半结构化数据的结构信息视为标签序列的集合,采用数据挖掘技术中频繁模式和关联项集的概念及算法,从半结构化数据中挖掘频繁关联标签序列并以此作为特征计算其结构相似度.实验结果证明:提出的基于频繁关联标签序列的半结构化数据结构相似度度量方法可以解决基于路径方法的不足,计算的结构相似度更准确、更合理.

关键词 数据挖掘;数据管理;半结构化数据;结构相似度;频繁关联标签序列

中图分类号 TP311 **文献标志码** A **文章编号** 1671-4512(2012)08-0077-05

Structural similarity measurement for semi-structured data based on tag sequences

Zhang Lijun Li Zhanhuai Li Ning Li Xia

(School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract As existing path-based structural similarity measurement for semi-structured data approaches cannot handled well the case of matching partially between paths and the sibling relationships between elements are ignored, a new structural similarity measure approach based on association tag sequences was proposed. The structure information in semi-structured data were regarded as set of tag sequences, and adapting the concept and algorithm of frequent pattern and association pattern in data mining field, frequent association tag sequence(FATS) can be mined from semi-structured data collection, and then FATS were used as features of semi-structured data to calculate the structural similarity. Experimental results show that FATS approach is more accurate and reasonable than existing path-based approaches.

Key words data mining; data management; semi-structured data; structural similarity; frequent association tag sequence(FATS)

互联网上存在大量半结构化数据,许多数据管理任务须要计算半结构化数据间的相似度,因此相似度计算成为许多半结构化数据处理技术的基础[1-2].与 Flat Text 数据不同,半结构化数据中包含有表示关键字层次的结构信息,如何利用包含其中的结构信息计算半结构化数据的结构相似度,成为半结构化数据相似度计算中的一个关键问题[3].

半结构化数据结构相似度的计算方法主要有树编辑距离、频繁子树和基于路径的方法等.树编辑距离方法[4-5]将结构信息看作是树,寻找 2 棵树之间互相转换的最小编辑操作序列作为 2 个文档之间距离.然而这种方法复杂度高,且已被证明是 NP 完全问题.频繁子树方法[6-8]从半结构化数据

收稿日期 2011-11-25.

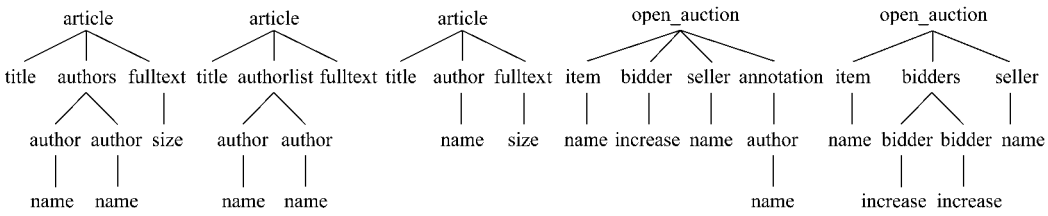
作者简介 张利军(1978-),男,博士研究生,E-mail:zhanglijun@nwpu.edu.cn.

基金项目 国家高技术研究发展计划资助项目(2009AA1Z134);国家自然科学基金资助项目(60803043,60970070);西北工业大学基础研究基金资助项目(JC20110225,JC201261).

集中挖掘频繁子树作为特征,并以此计算其相似度.但频繁子树挖掘方法效率低,计算代价高,尤其是在数据集比较大、结构非常复杂的情况下,难以挖掘出结果.基于路径的方法[9-11]将结构信息看作是由不同路径构成的集合,计算相对简单而被广泛应用.这种方法虽保留了元素间父-子/祖先-后代关系,但忽略了元素之间的兄弟关系,且不能很好地处理路径部分相似的情况,导致相似度结果不够准确、合理.为此,本研究提出一种基于频繁关联标签序列的半结构化数据结构相似度度量方法,以解决基于路径方法的不足.

1 标签序列基本概念

半结构化文档可建模为一棵标签树,其中每个树节点对应半结构化文档中的一个元素,节点用该元素的标签(tag)标记,如图1用5棵树表示5个半结构化文档.本研究仅关心半结构化数据中的结构信息,忽略其内容信息.对于给定的半结构化数据集C,抽取其中所有元素的标签构成标



(a) 文档1 (b) 文档2 (c) 文档3 (d) 文档4 (e) 文档5

图1 半结构化文档示例

签集(tagset).

定义1 标签序列 标签集中若干标签构成的有序列表.标签序列α表示为$\langle a_1, a_2, \cdots, a_n\rangle$,其中$a_i$为标签集中的标签.标签序列的长度为其中包含的标签个数.

定义2 标签序列的包含关系 对于标签序列$\alpha:\langle a_1, a_2, \cdots, a_m\rangle$和$\beta:\langle b_1, b_2, \cdots, b_n\rangle$,若存在整数$i_1<i_2<\cdots<i_m$,使得$a_1=b_{i_1}, a_2=b_{i_2}, \cdots, a_m=b_{i_m}$,则称$\beta$包含$\alpha$,也称$\alpha$为$\beta$的子序列或$\beta$为$\alpha$的超序列.

标签序列α的支持度为数据集中包含α的文档数与文档总数的比率.若α的支持度大于等于最小支持度阈值δ,则称α为频繁的,若不存在α的频繁超序列,则称α为极大频繁标签序列.

图1中〈article, authorlist, author, name〉和〈author, name〉是2个标签序列,支持度分别为20%(1/5)和80%(4/5).设最小支持度阈值为60%,则〈author, name〉是频繁的,但不是极大的,因为存在其超序列〈article, author, name〉(支持度60%(3/5))是频繁的,而〈article, author, name〉是极大频繁标签序列.

定义3 关联标签序列 是标签序列的集合,且对于其中任意标签序列α,该集合中都不存在另一标签序列β,使得β包含α.

与定义2类似,可定义关联标签序列γ的支持度和频繁关联标签序列(FATS),限于篇幅,不再赘述.

若不存在频繁关联标签序列γ的超集η,使得它们的支持度相等,则称γ是闭频繁关联标签序列.

设最小支持度为40%,则图1中{〈article, fulltext〉, 〈article, title〉}和{〈article, fulltext〉, 〈article, title〉, 〈article, author, name〉}都是关联标签序列且都是频繁的,{〈article, fulltext〉, 〈article, title〉}不是闭频繁关联标签序列,因为{〈article, fulltext〉, 〈article, title〉, 〈article, author, name〉}是它的超集,且二者支持度相同(60%),{〈article, fulltext〉, 〈article, title〉, 〈article, author, name〉}是闭频繁关联标签序列.同样,{〈open_auction, item, name〉, 〈open_auction, bidder, increase〉, 〈open_auction, seller, name〉}也是闭频繁关联标签序列,而{〈open_auction, item, name〉, 〈open_auction, seller, name〉}是频繁的,但不是闭的.

以上定义中,使用序列的概念可以解决路径部分相似的问题,而使用关联的概念捕捉到了元素之间的兄弟关系,在一定程度上弥补了现有基于路径方法的不足.

2 基于关联标签序列的相似度度量算法

对于给定的半结构化文档数据集C,可挖掘闭频繁关联标签序列.半结构化文档的结构可表示为其所包含的闭频繁关联标签序列的集合,从而据此计算2个文档间的相似度.

设闭频繁关联标签序列集合为F,则C中任

意半结构化文档 d_i 的结构可表示为

$$S_i = \{\gamma \mid \gamma \in F \wedge (d_i \text{ 包含 } \gamma)\}. \quad (1)$$

任意 2 个半结构化文档 d_i 和 d_j 间的结构相似度定义为

$$\mathrm{sim}(d_i, d_j) = (S_i \cap S_j)/(S_i \cup S_j). \quad (2)$$

给定半结构化文档数据集 C 及最小支持度阈值 δ，文档间结构相似度的算法描述如下.

输入　半结构化数据集 C，最小支持度阈值 δ.

输出　数据集 C 的结构相似度矩阵 $\boldsymbol{M}$.

步骤 1　将 C 中文档的结构解析为其对应 DOM 树中从根节点到叶子节点的标签序列的集合，得到标签序列数据库 D.

步骤 2　以 δ 为阈值，从 D 中挖掘频繁标签序列集合 T.

步骤 3　根据极大频繁标签序列的定义，剔除 T 中非极大频繁标签序列.

步骤 4　对于 D 中每一个文档的每一个标签序列，若 T 中存在它的一个子标签序列，则将其用该子标签序列代替；若不存在，则删除. 得到新数据库 D'.

步骤 5　以 δ 为阈值，从 D' 中挖掘闭频繁关联标签序列集 F.

步骤 6　对于 D' 中的任意一个文档 d_i，根据式(1)将其表示为闭频繁关联标签序列的集合.

步骤 7　根据式(2)计算结构相似度矩阵 $\boldsymbol{M}$.

步骤 1 为数据预处理步骤. 处理过程中，相同的标签序列在同一个文档中只出现 1 次. 步骤 2 用到序列模式挖掘算法，PrefixSpan[12] 为主流的序列模式挖掘算法，该算法在运行过程中会产生大量重复的投影数据库，影响运行效率. 因此本研究采用文献[13]中的方法，该方法通过记录已产生投影数据库的位置信息以避免相同投影数据库的产生，从而提高算法效率. 步骤 3 进行特征筛选，保留极大频繁标签序列作为特征. 步骤 4 将所有数据用极大频繁标签序列表示. 步骤 5 挖掘闭频繁关联标签序列，本研究采用 CLOSET+[14] 算法. 步骤 6 将文档表示为闭频繁关联标签序列的集合. 步骤 7 计算 C 中任意 2 个文档间的结构相似度，得到相似度矩阵.

3　实验及结果分析

3.1　实验数据集及设置

实验采用 ACM Sigmod Record① 1999，2002，2005 年 XML 版本数据. 这些 XML 文件属于 4 个 DTD：IndexTermsPage. dtd，OrdinaryIssuePage. dtd，ProceedingsPage. dtd，SigmodRecord. dtd. 由于 SigmodRecord. dtd 只包含 1 个文件，因此取其他 3 个 DTD 的数据进行测试. 另外 IndexTermsPage. dtd 有 920 个文件，远多于其他 DTD 的文件，所以本研究随机选择其中 100 个用于测试. 各 DTD 文档数如表 1 所示.

表 1　Sigmod 数据集文档数

DTD 文件	1999 年	2002 年	2005 年	合计
IndexTermsPage	100	0	0	100
OrdinaryIssuePage	51	30	13	94
ProceedingsPage	17	16	0	33

将本文的方法（FATS）与 BOTP[9]，BOXP[9]，CXP[10] 和 subPath[11] 方法对比. CXP 和 FATS 方法使用最小支持度阈值参数，多次实验结果表明 Sigmod 数据集上最小支持度阈值为 30% 时能达到较好的结果，因此本研究在实验中设置最小支持度阈值为 30%.

3.2　相似度结果分析

结构相似度结果如表 2 所示，表中：it，oi，pr 分别表示 IndexTermsPage. dtd，OrdinaryIssuePage. dtd；ProceedingsPage. dtd；it-oi 表示 IndexTermsPage. dtd 与 OrdinaryIssuePage. dtd 中文档结构相似度的算术平均值，其余依此类推.

表 2　Sigmod 数据集不同 DTD 间结构相似度　　%

DTD 文件	it-it	oi-oi	pr-pr	it-oi	it-pr	oi-pr
BOTP	91.201 3	39.284 7	48.005 9	0.000 0	0.000 0	0.000 0
BOXP	72.207 2	21.582 8	36.208 5	0.000 0	0.000 0	0.000 0
subPath	91.845 3	50.842 8	54.344 9	5.666 5	2.819 1	9.409 2
CXP	100.000 0	91.047 4	90.546 8	0.000 0	0.000 0	82.072 5
FATS	74.787 9	69.395 3	46.054 3	6.045 6	10.111 1	33.037 8

由表 2 可看出：同类结构数据(it-it，oi-oi，pr-pr)采用不同方法均能得到较高的结构相似度，尤其是 subPath，CXP 和 FATS 方法；而对于异类结构数据(it-oi，it-pr，oi-pr)，BOTP，BOXP 和 CXP 方法不能很好地区分. BOTP 和 BOXP 方法计算的结构相似度皆为 0，CXP 计算的 it-oi，it-pr 为 0，即认为两者之间完全不相似，这与事实不符. 另外，CXP 方法认为 oi-pr 相似度高于

80%,对于这样一组异构数据而言,相似度评估过高,不太合理. 与之相对,subPath 和 FATS 方法对同类结构和异类结构数据结构相似度的计算结果更为合理.

为了进一步测试各方法区分半结构化数据间微小差别的能力,对 Sigmod 数据集稍作修改,使得同一 DTD 不同年份的数据在结构上有所不同,不同 DTD 文档的结构尽量不同,结果如表 3 所示. 表中 oi02 表示 OrdinaryIssuePage. dtd 2002年数据文档,以下讨论中,Sim_{BOTP}(oi02,oi05)表示 BOTP 方法计算 oi02 和 oi05 文档间结构相似度平均值,其余依此类推. 由表 3 可知:

表 3 Sigmod 数据不同 DTD 不同年份文档间结构相似度 %

			BOTP	BOXP	subPath	CXP	FATS
同构数据	it99	it99	91.2013	72.207 2	91.845 3	100.000 0	74.787 9
	oi02	oi02	88.214 8	51.492 8	98.245 6	85.456 3	65.655 2
	oi05	oi05	100.000 0	51.157 5	99.383 8	100.000 0	100.000 0
	oi99	oi99	98.462 8	53.293 2	99.369 6	100.000 0	92.444 4
	pr02	pr02	97.892 4	86.504 6	99.389 3	100.000 0	88.333 3
	pr99	pr99	100.000 0	64.246 5	99.519 6	100.000 0	82.843 1
异构数据	it99	oi02	0.000 0	0.000 0	5.291 9	0.000 0	10.977 8
		oi05	0.000 0	0.000 0	6.443 6	0.000 0	8.666 7
		oi99	0.000 0	0.000 0	5.688 7	0.000 0	2.476 2
		pr02	0.000 0	0.000 0	2.749 0	0.000 0	15.166 7
		pr99	0.000 0	0.000 0	2.885 0	0.000 0	5.352 9
	oi02	oi05	0.000 0	0.000 0	68.604 5	84.621 4	54.222 2
		oi99	0.000 0	0.000 0	8.124 7	88.284 3	52.333 3
		pr02	0.000 0	0.000 0	9.486 3	82.426 4	18.472 2
		pr99	0.000 0	0.000 0	0.953 7	84.621 4	47.477 1
	oi05	oi99	0.000 0	0.000 0	8.842 3	86.602 5	72.222 2
		pr02	0.000 0	0.000 0	9.719 2	81.649 7	37.500 0
		pr99	0.000 0	0.000 0	1.019 1	100.000 0	48.235 3
	oi99	pr02	0.000 0	0.000 0	1.422 3	70.710 7	8.805 6
		pr99	0.000 0	0.000 0	23.921 7	86.602 5	38.235 3
	pr02	pr99	0.000 0	0.000 0	11.885 0	81.649 7	9.007 4

$$Sim_{BOTP}(oi02,oi05) = Sim_{BOTP}(oi02,oi99) = Sim_{BOTP}(oi05,oi99) = Sim_{BOTP}(pr02,pr99) = 0.0\%;$$

$$Sim_{BOXP}(oi02,oi05) = Sim_{BOXP}(oi02,oi99) = Sim_{BOXP}(oi05,oi99) = Sim_{BOXP}(pr02,pr99) = 0.0\%,$$

说明即使同一 DTD 的数据, BOTP 和 BOXP 方法仍认为完全不相似,不合理.

$Sim_{subPath}(oi02,oi99) < Sim_{subPath}(oi02,pr02)$;

$Sim_{subPath}(oi05,oi99) < Sim_{subPath}(oi05,pr02)$,

说明 subPath 方法认为相同 DTD 文档间相似度低于不同 DTD 文档间的相似度,不合理.

$Sim_{CXP}(oi05,pr99) = 100\%$,即 CXP 方法认为不同结构下的 oi05 和 pr99 完全相似,同样不合理. 而且 $Sim_{CXP}(oi02,oi05) = Sim_{CXP}(oi02,pr99)$,同一 DTD 文档间的相似度等于不同 DTD 文档间相似度,也不合理,而 FATS 方法无以上不合理之处,说明其结果更为准确.

$Sim_{subPath}(oi99,pr99) > Sim_{subPath}(pr02,pr99)$;

$Sim_{CXP}(oi99,pr99) > Sim_{CXP}(pr02,pr99)$;

$Sim_{FATS}(oi99,pr99) > Sim_{FATS}(pr02,pr99)$.

通过对实际数据的分析发现,oi99 和 pr99 的结构比 pr02 和 pr99 的结构更为相似,几种方法得出类似的结果,与实际情况相符.

3.3 聚类结果分析

为了进一步说明本文方法的有效性,将几种方法用于对 Sigmod 数据集聚类,并评价其聚类结果. BEP 指标可以综合考量准确率和召回率 2 个指标,因此用 BEP 作为评价指标,比较几种方法的结果.

实验得到的 BEP 值分别为:BOTP(50.51), BOXP(50.51), subPath(49.96), CXP(66.92), FATS(67.23). 由此可知:几种不同的相似度度量方法用于聚类时,CXP 和 FATS 方法明显好于其他 3 种,而 FATS 方法又略好于 CXP 方法.

以上实验表明:本方法可有效解决现有基于

路径方法存在的问题，计算结果更合理、更准确.

参　考　文　献

[1] Leung H, Chung F, Chan S C. On the use of hierarchical information in sequential mining-based XML document similarity computation[J]. Knowledge and Information Systems, 2005, 7(4): 476-498.

[2] Tekli J, Chbeir R, Yetongnon K. An overview on XML similarity: background, current trends and future directions[J]. Computer Science Review, 2009, 3(3): 151-173.

[3] Algergawy A, Mesiti M, Nayak R, et al. XML data clustering: an overview[J]. ACM Computing Surveys, 2011, 43(4): 1-41.

[4] Zhang K, Statman R, Shasha D. On the editing distance between unordered labeled trees[J]. Information Processing Letters, 1992, 42(3): 133-139.

[5] Nierman A, Jagadish H V. Evaluating structural similarity in XML documents[C]// Proceedings of the 5th International Workshop on the Web and Databases(WebDB). Wisconsin: ACM, 2002: 61-66.

[6] Termier A, Rousset M C, Sebag M. TreeFinder: a first step towards XML data mining[C]// Proceedings of IEEE International Conference on Data Mining (ICDM). Maebashi: IEEE Computer Society, 2002: 450-457.

[7] Zaki M J. Efficiently mining frequent trees in a forest: algorithms and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(8): 1021-1035.

[8] Yang J, Wang S. Extended VSM for XML document classification using frequent subtrees[C]// Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval. Berlin: Springer-Verlag, 2010: 441-448.

[9] Joshi S, Agrawal N, Krishnapuram R, et al. A bag of paths model for measuring structural similarity in Web documents[C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington: ACM, 2003: 577-582.

[10] Leung H, Chung F, Chan S C, et al. XML document clustering using common Xpath[C]// Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration. Tokyo: IEEE Computer Society, 2005: 91-96.

[11] Rafiei D, Moise D, Sun D. Finding syntactic similarities between XML documents[C]// Proceedings of the 17th International Conference on Database and Expert Systems Applications. Krakow: Springer, 2006: 512-516.

[12] Pei J, Han J, Mortazavi-Asl B, et al. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth[C]// Proceedings of the 17th International Conference on Data Engineering (ICDE). Washington: IEEE Computer Society, 2001: 215-224.

[13] 张利军，李战怀，王淼. 基于位置信息的序列模式挖掘算法[J]. 计算机应用研究, 2009, 26(2): 529-531.

[14] Wang J, Han J, Pei J. CLOSET+: searching for the best strategies for mining frequent closed itemsets[C]// Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining (SIGKDD). Washington: ACM, 2003: 236-245.