XML数据流中面向聚类的指数直方图

高明霞，姚文集，毛国君
（北京工业大学 计算机学院，北京 100124）

摘 要：为了实现XML(extensible markup language)数据流的在线动态聚类，提出一种XML聚类特征指数直方图. 该结构以XML时间聚类特征为基础，遵循指数直方图的维护规律. 采用该结构的聚类算法在真实和模拟数据集上的实验结果说明：这一结构在聚类质量上可以达到甚至超过静态聚类方法；直方图个数固定时，内存开销基本稳定.

关键词：可扩展标记语言(XML)；指数直方图；时间聚类特征
中图分类号：TP 311 **文献标志码**：A **文章编号**：0254 - 0037(2011)08 - 1242 - 07

可扩展标记语言(extensible markup language, XML)[1]是一种用于数据交换和共享的自描述语言，于1998年2月成为W3C的推荐标准. 遵循这些标准的Web应用和服务在实时数据传输及交换过程中将产生大量XML数据. 为了从这些数据中发掘有用的知识，众多研究者集中于XML数据聚类挖掘[2-3]，提出了大量XML文档相似性度量方法. 这些方法大致可以分为2类：基于树编辑距离的方法[4-7]和基于文档特征集的方法[8-12]. 编辑距离的基本思想是将2棵树间的距离定义为利用编辑操作(删除、插入、修剪等)将一棵树转化为另一棵树所需的代价. 不同于编辑距离的间接度量方法，文档特征集的方法是首先提出各种方式用于表示XML文档特征并通过直接计算这些特征间的距离来度量XML文档间的相似性. Nayak[12]提出通过层结构特征的相似性来进行XML文档聚类的方法. 但是，现有的XML数据聚类方法主要处理静态数据集，一般需要多次反复的文档读取和解析，并没有考虑随时间变化的在线聚类研究.

为了在线动态聚类XML数据流，作者在文献[13]中重点讨论了基于滑动窗口的在线聚类算法，为动态实时聚类XML文档提供了有效的方法. 本文提出了一种基于XML层结构表示的XML聚类特征指数直方图结构.

1 XML文档的层结构特征及相似性

数据流区别于传统数据集的特点是数据持续到达且速度快、规模大. 为了满足数据流的这种快速要求，在数据的特征分析方面通常采用近似表达. 异构XML文档是XML数据流的基本单位，这些不同结构和内容的XML文档蕴含着复杂的层次结构和语义信息. 为了能实时在线对XML文档进行聚类特征提取，适应数据流的快速特点，聚类时可以忽略XML文档数据中的一些次要信息，只关注与结构相关的概要特征. 基于这种思想，本文采用了文献[12]的XML文档集聚类特征及相似性计算公式.

1.1 XML文档的层结构特征

根据XML文档中元素出现的顺序，可以将XML文档表示成一棵顺序标记树，其中，每个元素名称用一个确定的整数代替，这个整数表示元素开始标签在文档中出现的顺序. 图1展示了一个XML文档和其对应的顺序标记树. 层结构(LevelStructure)是XML文档对应的顺序标记树或XML文档集对应的顺序标记森林的一个简化，用于展示层次结构及每层包含的无重复数据元素. XML文档集的层结构表示具有性

收稿日期：2009-09-08.
基金项目：国家自然科学基金资助项目(60496322)；北京工业大学博士启动基金资助项目(X0007011200901).
作者简介：高明霞(1973—)，女，河北张北人，讲师.

质 1 所述的可组合性. 图 2(a)是图 1(b)中顺序标记树对应的层结构 LevelStructure(C_2)，图 2(b)是另一个 XML 文档的层结构 LevelStructure(C_1)，图 2(c)是根据性质 1 进行层结构组合后得到的组合层结构 LevelStructure($C_1 \cup C_2$).

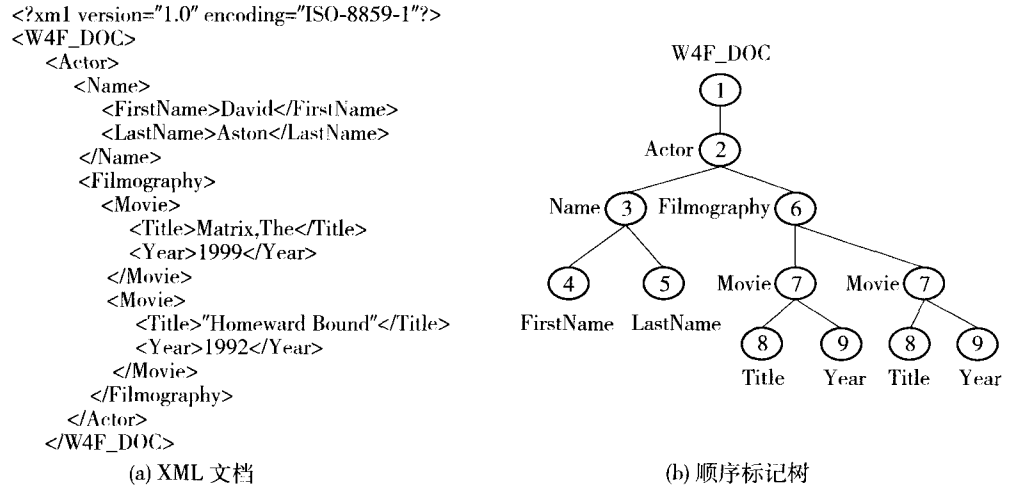

图 1　XML 文档和其对应的顺序标记树

Fig. 1　An XML document and its ordered labeled tree

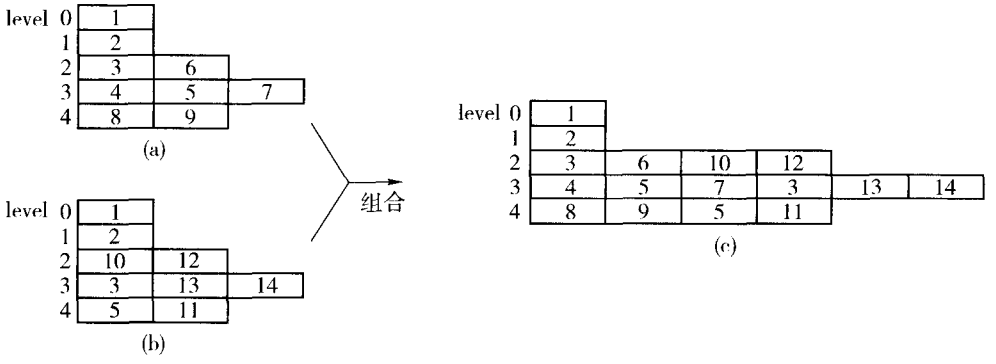

图 2　XML 文档的层结构组合

Fig. 2　Mergering two level structures of XML documents

性质 1　给定 2 组 XML 文档集合 C_1 和 C_2，集合中的文档个数满足 $|C_1| \times |C_2| \geq 1$，LevelStructure($C_1 \cup C_2$)可由 LevelStructure($C_1$)和 LevelStructure($C_2$)以层为单位进行元素集合并操作，同层重复元素只保留一个副本.

1.2　XML 文档集相似性计算

将 XML 文档集表示成对应的层结构后，其相似性可以通过层结构计算. 核心思想是：以层为单位匹配 2 个层结构中涉及的元素，并根据匹配情况度量这些元素的权重. 通常情况下，某个元素的权重反映的是该元素对整个文档的重要程度. 在树结构中，越靠近根的元素越能说明该文档所属的类型，其权重越大. 遵循这一规则，层结构中不同层的元素将被赋予不同的权重，并满足从上层到下层，权重从大到小的变化规律. 具体计算公式可见文献[12].

2　XML 聚类特征指数直方图

XML 查询领域经常将 XML 数据流定义为 XML 文档中节点的有穷序列，即：$X_t = \{x_t^1, x_t^2, \cdots, x_t^i, \cdots\}$.

其中 x_t^i 对应了 XML 文档中的某个节点或标记的取值. 本文关注的是异构 XML 文档间的概要特征表示问题,讨论的 XML 数据流是以文档元流 X_t 为基本单位,按时间顺序先后到达的 XML 文档流.

定义 1 XML 文档流表示为 $S=\{X_1\cdots X_i\cdots\}$,各文档的时标为 $\{T_1\cdots T_i\cdots\}$,且对任意 $i<m$,$T_i<T_m$. 其中 X_i 是一个 XML 文档,表示在 T_i 时标到达的 XML 文档数据.

在任意时刻只考虑并处理最近到达的 N 个元组的滑动窗口模型,较好地体现了数据流中元素的重要性随时间而递减的特性,因此滑动窗口模型在数据流计算中占据重要位置[14]. 为了对滑动窗口中的 N 个 XML 文档进行在线实时聚类,本文参考了常建龙等[15]在传统数据流上对滑动窗口内的数据进行的在线聚类研究,提出了一种近似结构(称为 XML 聚类特征指数直方图)用于表示一个实时的微簇(cluster),该结构满足纳伪聚类特征指数直方图的约束和更新条件,可以在线实时维护. 在任意 t 时刻,通过维护滑动窗口中不同数目的 XML 聚类特征指数直方图,就可实现在线聚类最近到达的 N 个 XML 文档数据流的任务.

2.1 XML 聚类特征指数直方图

定义 2 XML 文档流 S 的时间聚类特征(temporal cluster feature, $\mathrm{TCF}^{\mathrm{XML}}$) F^{XML} 为三元组 $(\mathrm{LevelStructure}_{1\to n}, n, t)$. 其中,$\mathrm{LevelStructure}_{1\to n}$ 为 n 个 XML 文档的层结构 $\mathrm{LevelStructure}_i$ 的合并 $\sum_{i=1}^{n}\mathrm{LevelStructure}_i$;$n$ 为时间聚类特征中包含的 XML 文档个数;t 为时间聚类特征中最新到达的 XML 文档的时标 T_n.

XML 文档流的时间聚类特征具有性质 2 中所述的可组合性. XML 聚类特征指数直方图(exponential histogram of cluster feature for XML, EHCF) H^{XML} 是一个满足指数直方图约束的数据结构,随着滑动窗口中 XML 文档的动态插入和删除被动态维护和更新,如定义 3 所述. 为了方便 XML 聚类特征指数直方图的实时在线维护,即插入新 XML 文档和及时删除过期文档,一个 EHCF 中的 XML 文档根据到达的先后次序,被划分为若干组,记作 $G_1, G_2, \cdots$;根据组内文档的个数情况,又将组标记为不同的级别,记作 $G_1^m, G_2^m, \cdots, G_i^j, \cdots$;这些组、组内文档及组的级别要满足和文献[15]中的纳伪指数直方图类似的约束条件:

1) 当 $i<j$ 时,组 G_i 中的所有 XML 文档的时标小于组 G_j 内所有 XML 文档时标;

2) 各个组 G 中包含 XML 文档的数目符合 2 的指数分布,即只能包含 $2^0=1, 2^1=2, 2^2=4$ 等数目的文档且最新组 G_n 只包含 1 个文档;

3) 当 G 中包含文档数为 2^j 时,称组的级别为 j,标记为 G^j,除了最高级别的组以外,必须保证各级别均含 $[1/\varepsilon]$ 或 $[1/\varepsilon]+1$ 个组,其中 ε 为用户指定的误差参数且 $0<\varepsilon<1$;4) 如果 $F^{\mathrm{XML}}(G_1)$ 的 t 项为当前有效时标,可以保证 G_1 之后的组中文档都不过期.

条件 1)可以保证不同组中 XML 文档集形成的时间聚类特征在时间上是有序的,这样才能进行性质 2 的聚类特征合并. 条件 2)是指数直方图的固有条件. 条件 3)设定了一个级别内可容纳的最多时间聚类特征数并将其作为一个用户可变参数,避免了单一设置的局限,形成了按级别递增组合 XML 时间聚类特征的机制. 由于有条件 1)的保证,条件 4)说明删除过期文档时只需要检测 XML 聚类特征指数直方图中标号最小的聚类特征时标.

性质 2 $F^{\mathrm{XML}}(C_1\cup C_2)$ 可由 $F^{\mathrm{XML}}(C_1)$ 和 $F^{\mathrm{XML}}(C_2)$ 进行构建,其中 C_1 和 C_2 为 2 组 XML 文档集.

证明:根据性质 1 中 XML 文档层结构的组合特性可知,XML 文档集的层结构可以由所包含的 XML 文档的层结构按层组合;n 可以由 $F^{\mathrm{XML}}(C_1)$ 和 $F^{\mathrm{XML}}(C_2)$ 中的对应文档数直接累加获得;$F^{\mathrm{XML}}(C_1\cup C_2)$ 中的 t 项等于 $\max\{F^{\mathrm{XML}}(C_1)t, F^{\mathrm{XML}}(C_2)t\}$.

定义 3 XML 聚类特征指数直方图是按照指数直方图约束进行维护的 F^{XML} 集合 $H^{\mathrm{XML}}=\{F_0^{\mathrm{XML}}(G_0), \cdots, F_i^{\mathrm{XML}}(G_i), \cdots\}$,其中,$F_i^{\mathrm{XML}}(G_i)$ 由窗口中的一组时标为 $T_{i_1}\cdots T_{i_n}$ 的 XML 文档 $G_i=\{X_{i_1}\cdots X_{i_n}\}$ 组合形成,且当 $j<m$ 时,时标满足 $T_{i_j}<T_{i_m}$.

2.2 XML 聚类特征指数直方图的维护

滑动窗口中同时维护着多个 H^{XML}，当有新 XML 文档到达时，需要根据 1.2 节中 XML 文档集的相似性计算方式计算新文档和每个现有 H^{XML} 的相似性，并据此决定新文档要归属于哪个微簇.

假设新 XML 文档 X_p 属于 H_i^{XML}，则需要如下步骤对 H^{XML} 进行增量式维护. 首先根据定义 2 生成一个新的 0 级 $F^{XML}(G^0)$，其中 G^0 为仅包含 X_p 的数据集. 然后将 $F^{XML}(G^0)$ 加入到 H_i^{XML} 中. 若存在 $[1/\varepsilon]+2$ 个 0 级组，则将最老的 2 个 0 级 F^{XML} 根据性质 2 合并生成一个新的 1 级 F^{XML}，并从级别 1 开始继续这种合并过程，直至某级别中 F^{XML} 的个数满足约束条件 3) 时为止. 最后需要根据窗口下限，检查 H_i^{XML} 中最老的 $F^{XML}(G_1)$ 时标是否过期，如果过期则将其删除. 图 3 给出了 1 个说明性例子来演示 H^{XML} 动态处理以时标 $T_1 \cdots T_{10}$ 顺序到达的 $X_1 \cdots X_{10}$ 这 10 个 XML 文档的插入、合并以及删除过程. 假设 ε 设置为 0.5，根据约束条件 3)，每一级别的组个数最大保持 3 个. 在时标为 T_1 时，X_1 单独组成 1 个 0 级组，并根据定义 2 创建了 1 个新的 0 级 $F^{XML}(G^0)$；接下来的 2 个文档和第 1 个类似，形成了 2 个新的 0 级 $F^{XML}(G^0)$；在时标为 T_4 时，由于 0 级 $F^{XML}(G^0)$ 的个数已经大于 3，因而最老的 2 个 0 级 $F^{XML}(G^0)$ 根据性质 2 合并形成 1 个 1 级 $F^{XML}(G^1)$；在时标为 T_{10} 时，不但 0 级 $F^{XML}(G^0)$ 进行了合并操作，它合并后增加的 1 个 1 级 $F^{XML}(G^1)$ 又引发了一级 $F^{XML}(G^1)$ 的合并操作，生成了 1 个 2 级的 $F^{XML}(G^2)$. 如果此时这个指数直方图最老的聚类特征的时标 $F^{XML}(G_1) = T_4$ 已经超出了窗口下限，则将整个 $F^{XML}(G_1)$ 删除.

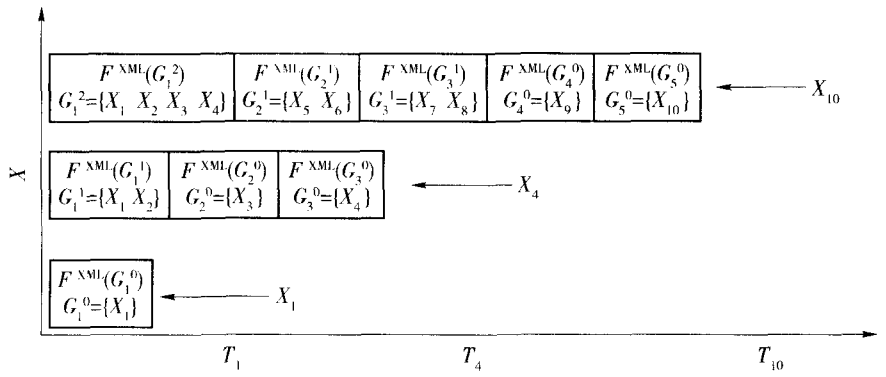

图 3　XML 聚类特征指数直方图的维护

Fig. 3　Merging processes of EHCF

如果数据流中新文档和滑动窗口中现有微簇的相似性都不满足最小相似性阈值 ω，就要生成新的 H^{XML}，代表一个新微簇. 在线维护这些 H^{XML} 数据结构需要内存空间，当 H^{XML} 数目大于某个阈值 N_C 而引起内存空间不足时，需要通过某种策略减少窗口中 H^{XML} 的数目. 此时最简单而直接的方法是抛弃某个 H^{XML}. 选择标准可以是包含 XML 文档数最少的微簇，这说明 XML 流中此类文档属于孤立点；也可以是最低级别的 XML 时间聚类特征中包含时标最老的 H^{XML}，这说明此 H^{XML} 最近都没有更新，接近于窗口中的过期簇；此外，也可以仿照文献[15]中的策略，对 2 个或多个 H^{XML} 进行合并操作，只要满足合并后生成的新 H^{XML} 既可以满足新文档的到达，又可以适时删除过期的 F^{XML}. 本文的实验采用最后一策略.

3 实验

3.1 实验设置

所有实验在一台 PentiumIV 2.4 GHz 的 PC 上进行，操作系统为 Windows XP. 聚类分析中，对聚类质量的评估指标使用了类内相似性与类间相似性. 定义 4 和定义 5 结合滑动窗口和层结构特征对这 2 个指标进行了重新定义.

定义 4　一个聚类 C_i 的类内相似性为类内每对文档层次相似性的均值，当 n 是聚类 C_i 中的文档数

时,计算公式为

$$\mathrm{IntraSim}(C_i) = \frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathrm{LevelSim}_{i,j}}{0.5 \times n \times (n-1)} \tag{1}$$

定义 5 假设时刻 T_C 在窗口大小为 N 的窗口中实时维护着 k 个 XML 聚类特征指数直方图 $C = \{H_1, H_2, \cdots, H_k\}$,其类内相似性(类间相似性)为所有聚类的类内相似性(类间相似性)的平均值. 当 n_i 是 C_i 中的文档数,N 是所有聚类中的文档总数,k 是聚类的数目,$\mathrm{LevelSim}_{i,j}$ 是 2 个类的类间相似性时,类内相似性计算公式为

$$\mathrm{IntraSim} = \frac{\sum_{i=1}^{k}\mathrm{IntraSim}(C_i) \times n_i}{N} \tag{2}$$

类间相似性计算公式为

$$\mathrm{InterSim} = \frac{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathrm{LevelSim}_{i,j}}{0.5 \times k \times (k-1)} \tag{3}$$

3.2 聚类质量

为了说明 XML 聚类特征指数直方图结构在聚类 XML 数据流时的有效性,选用文献[12]中的静态聚类算法 XCLS 作为对比算法,并采用相同的真实数据集 XMLFile 用于实验. 这个数据集由 460 个 XML 文档组成. 这些文档来自于 23 个自然领域,其中电影 74,大学 22,汽车 208,文献 16,公司 38,食宿信息 24,旅游 10,订单 10,拍卖数据 4,约定 2,文档页 15,书店 2,游戏 20,社团 12,医疗 2,营养 1. 这些文档的标签范围为 10 ~ 100,层数为 2 ~ 15.

为了使聚类结果更准确并排除 XML 文档顺序的影响,作者在不同文档顺序的情况下,使用真实数据集 XMLFile 在滑动窗口模式(标注为 CluXMLWin)和 XCLS 算法下各运行了 5 次,并计算了平均的类内相似性和类间相似性作为结果值. 图 4 为窗口大小为 100 时的类内相似性聚类结果和类间相似性聚类结果,在每个时间点滑动窗口内获取的聚类效果都达到甚至超过了静态聚类算法——基于层相似性的 XML 文档聚类(XML documents clustering with level similarity, XCLS). 从本质上说,2 种算法采用了相同的聚类相似性计算方法,聚类质量基本相当. 但是,滑动窗口模式下会抛弃过期的文档,这可能导致聚类同一微簇时,指数直方图维护的对应文档个数相对较少,层结构的影响就略小.

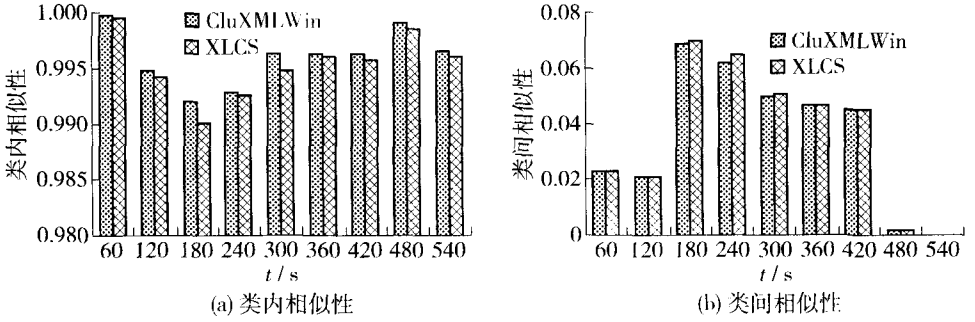

图 4 质量比较(真实数据集 XMLFile,$N = 100$,$\omega = 0.8$,$N_C = 50$)

Fig. 4 Quality comparison (XMLFile data set, $N = 100$, $\omega = 0.8$, $N_C = 50$)

3.3 内存开销

数据流处理算法的一个重要特征就是要求算法具有较小的内存空间开销,对于滑动窗口模式下的聚类来说,其内存主要用于维护窗口内的直方图(微簇). 一个 XML 聚类特征指数直方图维护着一组以层结

构为主的 XML 时间聚类特征. 从层结构的定义特点可知，一个类内相似度很高的微簇对应的层结构只和这组 XML 文档的 DTD 或 Schema 的复杂度相关，和其所包含的文档数目不成比例. 直方图的误差参数会影响直方图中时间聚类特征 F^{XML} 的数目，特别是对文档数目多的直方图. 在时标 T_1 时，假设一个直方图包括 10 个文档，如果 $\varepsilon = 0.5$，则根据直方图约束条件可知，此直方图维护了 3 个等级(1,2,2)，共 5 个 F^{XML}；如果 $\varepsilon = 0.125$，则该直方图维护了 2 个等级(1,8)，共 9 个 F^{XML}. 随着 ε 的增长，F^{XML} 的个数在减少，内存开销在下降.

为了验证上述分析结果，作者采用 XML 自动生成工具 oxygen 产生了一个文档数目是 20 838 的模拟数据流用于实验. 模拟数据集是基于一些成熟的行业 XML Schema 随机产生的，文档范围从 1 kB 到几百 kB. 图 5 是内存开销随窗口大小的变化. 从图中可以看出，对于一个分布稳定的数据流，当窗口中直方图个数恒定时内存开销基本固定，并不随窗口大小而变化.

图 6 是内存开销随直方图误差参数的变化情况. 尽管随着 ε 的增长，内存开销呈现下降趋势，但是由于文档数目多的直方图个数相对较少，内存开销的下降绝对值并不是很明显.

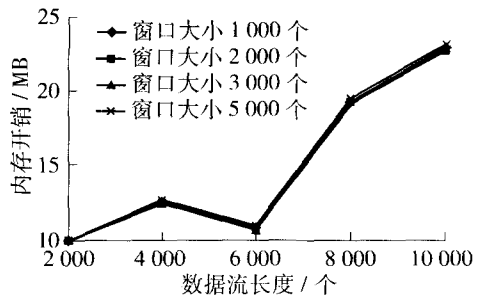

图 5　内存开销随窗口大小变化($\varepsilon = 4, \omega = 0.8, N_C = 130$)

Fig. 5　Memory vs. window size ($\varepsilon = 4, \omega = 0.8, N_C = 130$)

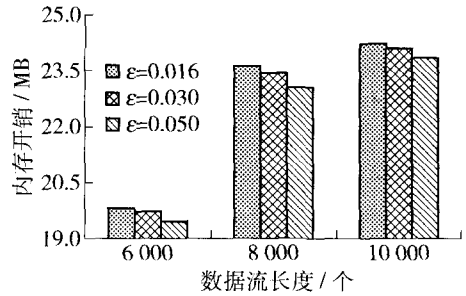

图 6　内存开销随误差参数的变化 ($N = 5\,000$, $\omega = 0.8, N_C = 130$)

Fig. 6　Memory vs. error parameter ε ($N = 5\,000$, $\omega = 0.8, N_C = 130$)

4　结束语

本文提出一种基于 XML 层结构表示的 XML 聚类特征指数直方图，用于在滑动窗口中动态聚类异构 XML 数据流. 该结构根据 XML 数据流的实际情况，采用 XML 聚类特征指数直方图作为微簇的概要结构，较好地保存了当前窗口内的 XML 文档流的分布状况，从而获取了较高质量的聚类结果.

参考文献：

[1] BRAY T, PAOLI J, SPERBERG-MCQUEEN C M, et al. Extensible markup language (XML) 1. 0[S/OL]. 5th ed [2009-07-08]. http: //www. w3. org/TR/REC-xml/.

[2] ALGERGAWY A, SCHALLEHN E, SAAKE G. A schema matching-based approach to XML schema clustering[C] // Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services 2008. New York: ACM Press, 2008: 131-136.

[3] LIAN W, WAI-LOK Cheung D, MAMOULIS N, et al. An efficient and scalable algorithm for clustering XML documents by structure[J] //IEEE Transactions on Knowledge and Data Engineering, 2004, 16(1): 82-96.

[4] COSTA G, MANCO G, ORTALE R, et al. A tree-based approach to clustering XML documents by structure[C] //Knowledge Discovery in Databases: PKDD 2004. Berlin: Springer-Verlag, 2004: 137-148.

[5] NIERMAN A, JAGADISH H V. Evaluating structural similarity in XML documents[C] //Proceedings of the 5th International Workshop on the Web and Databases, Madison: ACM Press, 2002: 61-66

[6] 郑仕辉，周傲英，张龙. XML 文档的相似测度和结构索引研究[J]. 计算机学报，2003，26 (9)：1116-1122.
ZHENG Shi-hui, ZHOU Ao-ying, ZHANG Long. Similarity measure and structural index of XML documents [J]. Chinese

Journal of Computers, 2003, 26 (9): 1116-1122. (in Chinese)

[7] DALAMAGAS T, CHENG T, WINKEL K J, et al. Clustering XML documents by structure[C]// SETN 2004. Berlin: Springer-Verlag, 2004: 112-121.

[8] YOON J, RAGHAVAN V, CHAKILAM V. BitCube: clustering and statistical analysis for XML documents[C]// Thirteenth International Conference on Scientific and Statistical Database Management. Fairfax: IEEE Computer Society, 2001: 18-20.

[9] 杨建武, 陈晓鸥. 基于核矩阵学习的 XML 文档相似度量方法[J]. 软件学报, 2006, 17(5): 991-1000.
YANG Jian-wu, CHEN Xiao-ou. Similarity measures for XML documents based on kernel matrix learning [J]. Journal of Software, 2006, 17(5): 991-1000. (in Chinese)

[10] BERTINO E, GUERRINI G, MESITI M. Measuring the structural similarity among XML documents and DTDs, DISI 2TR 202202[R]. Genova: Department of Computer Science, University of Genova, 2002.

[11] FLESCA S, MANCO G, SCIARI E M, et al. Detecting structural similarities between XML documents [C]// Proceedings of the 5th International Workshop on the Web and Databases, WebDB. Madison: ACM Press, 2002: 55-60.

[12] NAYAK R. Fast and effective clustering of XML data using structural information [J]. Knowl Inf Syst, 2008, 14: 197-215.

[13] 姚文集, 高明霞, 毛国君, 等. 基于滑动窗口的 XML 数据流聚类算法 [J]. 计算机工程, 2010, 36(13): 87-89.
YAO Wen-ji, GAO Ming-xia, MAO Guo-jun, et al. Algorithm for clustering XML data stream using sliding window [J]. Computer Engineering, 2010, 36(13): 87-89. (in Chinese)

[14] 金澈清, 钱卫宁, 周傲英. 流数据分析与管理综述[J]. 软件学报, 2004, 15(8): 1172-1180
JIN Che-qing, QIAN Wei-ning, ZHOU Ao-ying. Analysis and management of streaming data: a survey [J]. Journal of Software, 2004, 15(8): 1172-1180. (in Chinese)

[15] 常建龙, 曹 锋, 周傲英. 基于滑动窗口的进化数据流聚类[J]. 软件学报, 2007, 18(4): 905-918.
CHANG Jian-long, CAO Feng, ZHOU Ao-ying. Clustering evolving data streams over sliding windows[J]. Journal of Software, 2007, 18(4): 905-918. (in Chinese)

Exponential Histogram of Cluster Feature for XML Stream

GAO Ming-xia, YAO Wen-ji, MAO Guo-jun

(College of Computer Science, Beijing University of Technology, Beijing 100124, China)

Abstract: To mine XML stream in an online way, a data structure named exponential histogram of cluster feature for XML is proposed. The structure is based on the temporal cluster feature and can be maintained according to the exponential histogram rule. The experiment results for a real data set and a synthetic data set show that the structure is of higher quality than the method offline.

Key words: extensible markup language (XML); exponential histogram; temporal cluster feature

(责任编辑 梁 洁)