

基于矩阵的数据流频繁模式预测算法

任永功, 李雪兰

(辽宁师范大学 计算机与信息技术学院, 辽宁 大连 116029)

摘 要: 随着数据挖掘的广泛应用,许多实际的数据挖掘应用需要用过去和当前数据对未来数据状态进行预测,针对这一现状,文中提出基于矩阵的数据流频繁模式预测算法(MFP). MFP 算法可预测在下一时间窗口中可能性较大的频繁项集,以满足用户需要. 该算法首先将数据转换为 0—1 矩阵;然后通过矩阵剪裁和位运算更新矩阵,并从中挖掘频繁项集;最后,利用当前窗口数据预测下一时间窗口中可能出现的频繁项集. 实验结果表明, MFP 算法在不同实验环境下能有效预测频繁项集,该算法是可行的.

关键词: 频繁项集;数据流;数据挖掘;矩阵;预测

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2012)10—0060—04

Matrix-based Algorithm for Predicting Frequent Patterns over Data Streams

REN Yong-gong, LI Xue-lan

(School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)

Abstract: With the wide application of data mining, many practical data mining applications need to use past and current data to predict the future state of the data. To solve this problem, we propose a new method (MFP) for predicting frequent patterns over data streams. MFP algorithm can predict those frequent itemsets that have high potential to become frequent in the subsequent time windows, to meet users' needs. Firstly, the algorithm converts the data to 0—1 matrix. Then it will update the matrix by tailoring it and bit operations, from which mine frequent itemsets as well. Finally, it will predict possible frequent itemsets that may appear in the next time window by using the current data. Experimental results show that MFP algorithm can predict the frequent itemsets in different experimental conditions, therefore, the algorithm is feasible.

Key words: frequent itemset; data stream; data mining; matrix; predict

1 引言

随着数据流应用领域的不断扩大,数据流挖掘已成为研究的热点问题. 现有的基于数据流的数据挖掘算法已有很多^[1-4], 主要包括多数据流和单数据流挖掘. 多数据流挖掘的主要目的是分析多条并行到达的数据流之间相互关系, 关联规则. 单数据流挖掘主要包括数据流分类^[5]、聚类^[6]、频繁模式挖掘^[7]、一些突变^[8]等. 其中, 数据流频繁模式挖掘在

商务决策、知识库的应用中有着重要作用.

在数据流频繁模式挖掘方面, 许多学者在 Apriori 算法^[9] 和 FP-growth 算法^[10] 等经典算法的基础上提出了自己的算法^[11-13]. 然而, 这些算法主要集中在对当前的数据进行分析, 对数据流变化趋势的预测研究较少, 难以满足当前水灾预报、股票分析等应用的需要.

针对上述问题, 本文在单数据流挖掘的基础上提出了基于矩阵的数据流频繁模式预测算法. 相比

收稿日期: 2012—03—15; **修回日期:** 2012—04—20

基金项目: 国家自然科学基金项目(60603047); 教育部留学回国人员科研启动基金项目; 辽宁省科技计划项目(2008216014); 辽宁省教育厅高等学校科研基金(L2010229); 大连市优秀青年科技人才基金(2008J23JH026)

现有的频繁模式挖掘算法,该算法不仅可以挖掘当前时间窗口内的频繁模式,还可有效的对下一时间窗口内频繁模式进行预测。

2 问题相关定义

定义 1 (频繁项目集) 设 X 是一个项目集, 当 $\text{sup}(X) \geq s \times \omega$ 时 (s 是用户定义的范围在 $[0, 1]$ 上的最小支持数), X 被称为一个频繁项目集。

定义 2 (频繁 k -项集) 包含 k 个项目的频繁项集称为频繁 k -项集。

定义 3 (矩阵) 矩阵 A_{ij} 记为:

$$A_{ij} = \begin{matrix} & 0 & A_{01} & A_{02} & \cdots & A_{0n} \\ & \vdots & & & & \vdots \\ A_{i0} & A_{i1} & A_{i2} & \cdots & A_{in} \\ & \vdots & & & & \vdots \\ & 0 & A_{(m+1)1} & A_{(m+1)2} & \cdots & A_{(m+1)n} \end{matrix}$$

其中 $A_{0j} (1 \leq j \leq n)$ 为 1-项集, $A_{i0} (1 \leq i \leq (m+1))$ 为事务 T_i 包含的项数, $A_{(m+1)k} (1 \leq k \leq n)$ 为相应列 1-项集的支持数。

性质 1 如果项目集 $\{x_1, x_2, \dots, x_k\}$ 是一个频繁 k -项集, 那么它的所有非空子集也频繁。

性质 2 设 X_k 为频繁 k -项集, 则项目数小于 k 的事务不可能产生 k -项集。

性质 3 设 X_k 为频繁 k -项集, 则在窗口中出现次数小于 k 的项集不可能产生 k -项集。

3 预测算法

基于矩阵的数据流频繁项集更新算法将挖掘频繁项集的过程分为以下三步: 首先, 利用事务数据生成待挖掘矩阵; 然后更新矩阵, 删除矩阵中不能产生频繁项集的行、列, 同时利用更新后的矩阵挖掘频繁项集; 最后, 在窗口滑动前, 利用当前窗口数据预测窗口滑动后可能产生的频繁项集。

3.1 生成矩阵

首先, 将数据转换为 0-1 矩阵。若当前窗口大小 $N=4$, 最小支持数 $\text{min_sup}=3$, 挖掘 W_1 中数据如表 1 所示。

挖掘结果为 a, b, c, d, e, g , 6 个 1-项集。接着利用定义 3 生成矩阵。

$$A_{DB} = \begin{matrix} & 0 & a & b & c & d & e & g \\ 4 & 1 & 1 & 1 & 1 & 0 & 0 \\ 4 & 0 & 1 & 1 & 1 & 1 & 0 \\ 4 & 1 & 0 & 1 & 1 & 1 & 0 \\ 4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 4 & 4 & 3 & 1 \end{matrix}$$

表 1 事务数据

TID	Itemset
1	abcd
2	bced
3	acde
4	cdeg
5	abeg

其中, $A_{0j} (1 \leq j \leq n)$ 存储所有 1-项集; $A_{5j} (1 \leq j \leq n)$ 存储对应列 1-项集的支持数; $A_{i0} (1 \leq i \leq n)$ 存储对应行事务中得项目数; $A_{ij} (1 \leq i \leq n, 1 \leq j \leq n)$ 记录 1-项集在对应事务中是否存在, 若存在, 则值为 1, 否则, 值为 0。

生成矩阵的过程如算法 1 所描述。

算法 1 生成矩阵

输入: 事务数据集

输出: 矩阵

//str[]存放 W_1 中数据, a[][]为矩阵

```

1: 初始化 a[][] 各项为 0;
2: a[0][1] = str[0][0];
3:  $W_1$  中每个项目依次判断
4: for(s=1; s<q)
5:   if(str[m][n] == a[0][s]) {a[m+1][s] = 1;
6:     a[m+1][0]++; //统计事务包含项的个数;
7:     a[t+1][s]++; //支持数+1}
8:   else if(a[0][s] == null)
9:     a[0][s] = str[m][n]; //存储 1-项集;
10:    else s++;

```

3.2 矩阵更新和频繁项集挖掘

矩阵生成后, 挖掘频繁项集前要对矩阵进行更新, 删除非频繁 1-项集和性质 2 所述项目数小于 k 的事务。更新表 1 中第一窗口所得矩阵 A_{DB} , 并从对应矩阵中挖掘频繁 k -项集。矩阵更新过程如图 1 所示。

$$A^1 = \begin{matrix} & 0 & c & d & e \\ 2 & 1 & 1 & 0 \\ 3 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 0 & 4 & 4 & 3 \end{matrix} \quad A^2 = \begin{matrix} & 0 & c & d & e \\ 2 & 1 & 1 & 0 \\ 3 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 0 & 4 & 4 & 3 \end{matrix}$$

$$A^3 = \begin{matrix} & 0 & c & d & e \\ 3 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 0 & 3 & 3 & 3 \end{matrix}$$

图 1 矩阵变化

图 1 所示 A^1 为删除支持度小于 3 的 1-项集, 从矩阵 A^1 中挖掘频繁 1-项集. A^2 为删除项目数小于 2 的事务后得到的矩阵, 从中挖掘频繁 2-项集. A^3 为删除项目数小于 3 的事务后得到的矩阵, 从中挖掘频繁 3-项集.

更新矩阵的过程如算法 2 所描述.

算法 2 更新矩阵

输入: 由数据集产生的矩阵

输出: 更新后矩阵和频繁项集

```
1: for(i=1; i<q){
2:   if(项集支持数<min_sup){
3:     for(m=1; m<t; m++){ if(a[m][i]==1)
4:       a[m][0]=a[m][0]-1; } //事务包含项的个数-1
5:     delete 第 j 列; //删除不频繁列}
6:   else i++;
7:   for(s=1; s<=m; s++){
8:     if(a[s][0]<k){ if(a[s][m]==1)
9:       a[t+2-flag][m]--; //支持数-1
10:    delete 第 j 行; //由性质 2, 删除第 0 列小于 k 的行}
```

更新矩阵的同时, 从对应矩阵 A^k 中挖掘频繁 k -项集. 对应的矩阵 A^k 中保存着可产生 k -项集的 1-项集, 频繁 k -项集的产生就是对对应的矩阵 A^k 进行组合得到的结果, 对 1-项集进行组合并判断支持数, 得到频繁 k -项集. 如图 1 中, A^1 产生频繁 1-项集 c, d, e ; A^2 产生频繁 2-项集 cd, ce, de ; A^3 产生频繁 3-项集 cde , 是 c, d, e 组合的结果.

挖掘频繁项集过程如算法 3 所描述.

算法 3 挖掘频繁项集

输入: 更新后矩阵和频繁项集

输出: 频繁项集

```
fun(flag, i, k, sflag, a[], B[])
1: if(flag!=0){
2:   for(m=k+1; m<q; m++){ i++;
3:   if(a[0][m]==0){
4:   else{ for(s=0; s<k; s++){
5:   生成 k-项集并判断支持数;
6:   if(flag+1==k){
7:   else fun(); //递归调用 fun()
8:   flag--;
9:   fun(); //递归调用 fun()
10:  else{sflag++; i++;
11:  if(a[0][k]!=0)
12:  for(s=0, j=sflag+1; s<k, j<k+sflag+1; s++, j++)
13:  B[i][s]=a[0][j]}
```

3.3 挖掘窗口滑动后可能出现的频繁项集

在矩阵生成和频繁项集挖掘算法基础上, 利用

当前窗口数据预测新事务到来后可能出现的频繁项集, 预测算法分为以下几步:

(1) 在新事务到来前删除当前窗口中最旧事务, 窗口滑动后, 它将不再影响项集的频繁度;

(2) 将更新后的窗口数据划分为 DB 和 db 两部分, 其中, $\text{Sup}(\text{DB}) + \text{Sup}(\text{db}) = \text{Sup}(\text{DB} + \text{db})$;

(3) 在数据集 DB 和 db 中预测下一窗口频繁项集. 数据 db 中项集若为频繁项集, 应满足 $\text{Sup}(\text{db}) \geq \min_sup(\text{db})$; 当 $\text{Sup}(\text{db}) = |\text{db}|$ 时, 数据 DB 中项集若为频繁项集, 应满足 $\text{Sup}(\text{DB}) \geq \min_sup(\text{DB}) + \min_Sup(\text{db}) - |\text{db}|$.

在下一滑动窗口中可能频繁的项目集应该满足: $\text{Sup}(\text{DB}) + \text{Sup}(\text{db}) \geq 2\min_sup(\text{db}) + \min_sup(\text{DB}) - |\text{db}|$.

当支持度 $s=60\%$, 滑动窗口 $N=4$ 时, 以表 1 中数据为例, 用 W_1 中数据预测 W_2 中频繁项集. $|\text{DB}|=3, \min_sup(\text{DB})=s * |\text{DB}|=1.8; |\text{db}|=1, \min_sup(\text{db})=s * |\text{db}|=0.6$.

挖掘数据集 db 中满足 $\text{Sup}(\text{db}) \geq 0.6$ 的项集, 得到频繁项集: $c, d, e, g, cd, ce, cg, de, dg, eg, cde, cdg, deg, cdeg$; 挖掘数据集 db 中频繁项集支持数在数据集 DB 中 $\text{Sup}(\text{DB}) \geq 1.8 + 0.6 - 1 = 1.4$ 的频繁项集, 得到频繁项集: c, d, e, cd, ce, de, cde ; 项集在数据集 DB 和 db 中频繁, 应两条条件同时满足, 将 DB 和 db 中所得频繁项集相交得: c, d, e, cd, ce, de, cde , 为预测结果. 经验证, 所得结果与表 1 中 W_2 所得频繁项集结果相同.

挖掘可能出现的频繁项集过程如算法 4 所描述.

算法 4 挖掘可能出现的频繁项集

输入: 当前滑动窗口数据

输出: 下一滑动窗口可能出现的频繁项集

//矩阵生成、更新和频繁项集挖掘如算法 1~3 所述

```
1: F<-挖掘 db 中频繁项集;
2: D<-挖掘 DB 所有项集;
3: E=D∩F;
4: 判断 E 在数据集 DB 中.
```

4 实验分析

为了评估 MFP 算法的有效性, 对它的精确度进行分析. 用 C++ 实现了 MFP 算法, 实验在相同环境下进行. 实验环境为: Windows XP 系统, 2.66GHz 奔腾四处理器, 512MB 内存. 实验采用文献[3]的标准测试集 T10. I4. D100K 进行测试, 数据

集特征如表 2 所示,滑动窗口包含 4 个基本窗口。

表 2 数据集特征

数据库	T10. I4, D100K
事务数	100K
项目数	870
平均长度	10.1

图 2 为算法 MFP 挖掘测试集所得到结果精确度的变化,随着支持度的不断上升,挖掘结果精确度逐渐提高,当支持度大于 60% 时,所得结果的精确度上升为 100%。

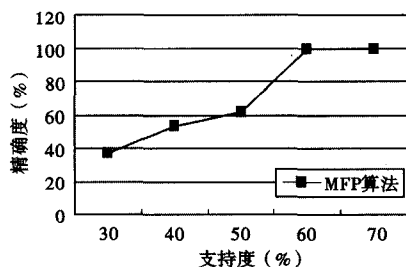


图 2 不同支持度下算法比较

如图 3 所示,随着窗口大小的增大,算法得到的精确度不断升高,在窗口大小大于 16 时,精确度接近 100%。实验说明, MFP 算法适合在大窗口下挖掘数据流。

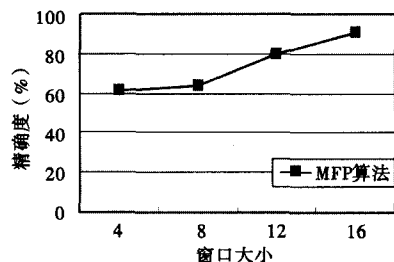


图 3 不同窗口大小下算法比较

如图 4 所示,随着项目数的增加, MFP 算法精确度逐渐降低,当项目数小时,算法精确度接近 100%。实验说明, MFP 算法适合对项目数少的数据流进行挖掘。

通过实验分析,算法 MFP 能够预测下一滑动窗口可能出现的频繁项集。

5 结束语

本文提出了基于矩阵的数据流频繁模式预测算法,不仅可以挖掘当前时间窗口中频繁项集,还可以预测数据流中下一窗口可能出现的频繁项集。实验

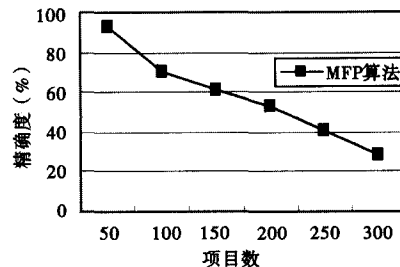


图 4 不同项目数下算法比较

结果表明, MFP 算法能够在不同实验条件下有效预测频繁项集,该算法对支持度高、窗口大且项目数少的数据流挖掘效果理想。

参考文献:

- [1] 张忠平,王浩,薛伟,等. 动态滑动窗口的数据流聚类方法[J]. 计算机工程与应用, 2011, 47(7): 39-41.
- [2] Haifeng Lathing Chen. Mining non-derivable frequent itemsets over data stream [J]. Data&Knowledge Engineering, 2009(68): 481-498.
- [3] Hua-Fu Li. Interactive mining of top-K frequent closed itemsets from data streams[J]. Expert Systems with Applications, 2009(36): 10779-10788.
- [4] Joong Hyuk Chang, Hye-Chung (Monica) Kum. Frequency-based load shedding over a data stream of tuples [J]. Information Sciences, 2009(179): 3733-3744.
- [5] Jin R, Agrawal G. Efficient decision tree construction on streaming data [C] // The ACM SIGKDD 9th International Conference on Knowledge Discovery and Data Mining. Washington, 2003: 2-6.
- [6] Raymond T Ng, Jiewei Han. Efficient and effective clustering methods of spatial data mining [C] // Proceedings of the 20th VLDB Conference. Chile: Santiago, 1994: 144-155.
- [7] Shichao Zhang, Zhi Jin, Jingli Lu. Summary queries for frequent itemsets mining [J] The Journal of Systems and Software, 2010(83): 405-411.
- [8] Tao Li, Shenghuo Zhu, Mitsunori Ogiwara. A new distributed data mining model based on similarity [C] // Proceedings of the 18th Annual ACM Symposium on Applied Computing. Canada: Vancouver, 2003: 432-436.
- [9] Agrawal R, Imilienski T, Swami A. Mining association rules between sets of items in large databases [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. Washington, DC, 1993: 207-216.

(下转第 68 页)

参考文献:

- [1] 万朋, 王建国, 赵志钦, 等. SAR 图像点目标的检测[J]. 电波科学学报, 2000, 15 (1): 55-59.
- [2] 曾生根, 王小敏, 范瑞彬, 等. 基于独立分量的遥感图像分类技术[J]. 遥感学报, 2004, 8(2): 150-157.
- [3] Dubuisson-Jolly M P, Gupta A. Color and texture fusion: application to imagesegmentation and GIS updating [J]. Image and Vision Computing, 2000(18): 83-832.
- [4] 黄启宏, 王帅, 刘钊. 改进的基于独立成分分析的图像颜色特征提取算法. [J]. 光电工程, 2007, 34(1): 121-125.
- [5] Oja E Principal Components, Minor Components, and linear neural networks [J]. Neural Networks, 1992, 5 (6): 927-935.
- [6] 何国辉, 甘俊英. PCA-LDA 算法在性别鉴别中的作用[J]. 计算机工程, 2006, 32(19): 208-210.
- [7] 余丰宁, 蔡启铭. 主成分监督分类及其在水质特征遥感图像识别中的应用. 湖波科学 [J], 1997, 9 (3): 261-

268.

- [8] 陈华, 陈书海. K-means 算法在遥感分类中的应用[J]. 红外与激光工程, 2000, 29 (2): 26-30.

作者简介:



赵 蕾 女, (1971—), 硕士, 副教授. 研究方向为图像处理、软件工程.

解争龙 男, (1961—), 硕士, 教授. 研究方向为信息安全.

李 红 女, (1976—), 博士研究生, 讲师. 研究方向为图像处理.

李小林 男, (1976—), 硕士, 讲师. 研究方向为图像处理、软件理论.

(上接第 63 页)

- [10] Han Jiawei, Pei Jian, Yin Yiwen. Mining frequent patterns without candidate generation[C]// Proceedings of the ACM SIGMOD'00. Dallas, TX, USA: ACM Press, 2000.
- [11] 刘洁, 杨路明, 毛伊敏, 等. 改进的数据流频繁闭项集挖掘算法 [J]. 计算机工程, 2011, 37(9): 75-77.
- [12] Show-Jane Yen, Cheng-Wei Wu, Yue-Shi Lee, et al. A fast algorithm for mining frequent closed itemsets over stream sliding window [C]// 2011 IEEE International Conference on Fuzzy Systems. Taipei, Taiwan, 2011: 996-1002.
- [13] Mina Memar, Mahmood Deypir. A block-based ap-

proach for frequent itemset mining over data streams [C]// 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) China: Shanghai, 2011: 1647-1651.

- [14] 徐嘉莉, 陈佳. 一种基于矩阵的频繁项集更新算法[J]. 计算机应用研究, 2010, 27(3): 837-840.

作者简介:

任永功 男, (1972—), 博士, 教授. 研究方向为数据挖掘、图像处理技术等.

李雪兰 女, (1986—), 硕士研究生. 研究方向为数据挖掘.