一种基于分布式数据库的关联规则挖掘新算法

黄 勇,赵 靖
(安徽科技学院 计算机系,安徽 凤阳 233100)

摘 要:分布式系统下关联规则挖掘算法的挖掘效率取决于频繁项目集的确定和网络各站点间的通讯量。为提高频繁项目集的生成效率,提出了关系数据库下一种新的数据预处理方法以及一种基于数组形式的频繁项目集生成算法。新的数据预处理方法可以降低候选项目集的数量,基于二进制的数组只需进行逻辑与运算便可生成频繁项目集,将该算法结合星型网络结构下的分布式挖掘算法SDMA应用于实验挖掘,理论分析与实验结果表明,算法提高了挖掘效率,是可行的。

关键词:分布式系统;关联规则;算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2011)02-0147-04

An Innovation Algorithm of Association Rules Mining for Distributed Database

HUANG Yong, ZHAO Jing
(Department of Computer Science, Anhui Science and Technology University, Fengyang 233100, China)

Abstract: The performance of association rules mining for distributed database mainly depends on the computation of frequency item sets and communication of networking. In order to enhance the generation efficiency of frequent items sets, put forward a new data preprocessing method which can reduce the size of candidate items sets, as well as a generation algorithm for frequent items sets with arrays which presents items to binary forms and uses some simple logical operations to generate frequent items sets. Theory analysis and experiments on dataset mining in distributed database with radical network structure show that SDMA algorithm is quite effective and feasible.

Key words: distributed system; association rule; algorithm

0 引 言

关联规则挖掘是从大数据集中提取潜在的关联关系或模式,从而认识事物客观规律的一种方法。关联规则挖掘的研究是数据挖掘领域一个较为重要的研究课题[1~3],它在货篮计划、市场营销、商品广告邮寄分析、保险业等方面具有重要的应用价值。随着网络技术的发展,分布式数据库已成为一种极其广泛的应用环境,如大型超市的客户销售记录都保存在不同的区域服务器中。考虑到安全性、通信效率等多方面因素的影响,将分散的数据集中到一起来存储存在很多困难,在这样的情况下,分布式环境下关联规则挖掘的研究逐渐成为了数据挖掘领域一个新的研究课题[4~8]。

在分布式环境下的关联规则挖掘方面,学者们提出了CD、DD、FDM算法。CD算法的基本思想是在各站点上都存储全局的候选项目集和频繁大项集,利用Apriori算法计算出候选集在本站点数据集上的支持数,然后做一次同步,各站点交换本站点候选项目集的支持数,它是典型的基于Apriori的并行算法。结点之间通讯量为$O(n^2)$,它的扩展性不好。DD算法的基本思想是将候选项目集在各个站点之间均匀分布,可以通过增加站点来减少扫描的候选项集数目,提高算法的并行效率。为了减少各站点间的通讯量,D. W. Cheung在1996年在CD算法的基础上提出了FDM算法,FDM算法在各个站点运行Apriori算法,找出各站点的局部频繁项目集,然后在各个站点全局频繁项目集之间进行支持度合计数交换,FDM算法的通讯量为$O(n)$,比较适于网状拓扑结构的网络环境。然而,在一些实际应用中,星型结构是常用的网络拓扑结构,如大型超市的总店和分店的数据库等。黄贤英等人在FDM算法的基础上提出了一种基于星型网络拓扑结构的分布式关联规则挖掘算法SDAM[4]。

分布式关联规则挖掘算法的效率主要由频繁项目

收稿日期:2010-05-04;修回日期:2010-09-17
基金项目:安徽省自然科学研究项目(KJ2009B033Z)
作者简介:黄 勇(1974-),男,安徽肥东人,硕士,副教授,研究方向为数据库技术与数据挖掘。

集的挖掘效率以及各站点之间网络通讯量来确定。因此,算法的优化方法主要集中在如何提高频繁项目集的挖掘效率和降低网络站点间的网络通讯量。为了降低挖掘频繁项目集的时间,文章提出了一种基于二进制形式的只需进行逻辑运算的频繁项目集生成算法,将该算法结合星型网络结构下分布式挖掘算法 SDMA 应用于实验挖掘,实验结果表明,算法提高了挖掘效率,是可行的。

1 相关理论基础

1.1 相关定义及定理

定义 1: DB 是在一个项目集合 $I = \{i_1, i_2, \cdots, i_n\}$ 上的交易集合,一个项目集合 $I_n = \{i_1, i_2, \cdots, i_k\}$, $I_n \subset I$,是一个交易,也称为项集。每个交易中包含项目 i 的个数称为该项集的长度,k-项集是一个长度为 k 的项集[9]。

定义 2:设项集 X、Y, $X \subset I$, $Y \subset I$,且 $X \cap Y = \varnothing$。支持度定义为 $S\%$,$S\%$ 是事务集 D 中包含 XUY 的百分比,即事务数据库 D 中至少有 $S\%$ 的事务包含 XUY;可信度定义为 $C\%$,$C\%$ 是事务数据库 D 中包含事务 X 同时又包含事务 Y 的百分比,即在事务数据库 D 中包含了 X 记录的同时至少有 $C\%$ 也包含 Y[4,5]。关联规则形如 $X \Rightarrow Y$ 的蕴含式,它成立的条件有以下两点:首先它具有支持度 $S\%$,且 $S\% \geqslant$ 用户设定的最小支持度;其次它具有可信度 $C\%$,$C\% \geqslant$ 用户设定的最小支持度。

定义 3:在 DB 中,若一项目集合出现频率大于等于用户定义的最小支持度,则该项目集合称为频繁项集,若同时可信度大于等于用户定义的可信度阈值,则为强关联规则。

定义 4:现设有 M 个站点,DB_i 为各站点上的局部数据库,且各局部数据库在逻辑上同构,DB_1,DB_2,$\cdots$,DB_m 的集合 DB 称为全局数据库。设用户定义的最小支持度为 min - sup,若项集 X 满足 $X.\sup_i \geqslant$ min - sup $\times DB_i$,则 X 为局部频繁项目集,若 X 满足 $X.\sup_i \geqslant$ min - sup $\times$ DB,则为全局频繁项目集。

定理 1:若项集 X 在站点 i 是局部频繁项目集,则所有的项集 $Y(Y \subset X)$ 在节点 j 也是局部大的。

定理 2:若项集 X 是全局频繁项目集,则在全局数据库中一定存在一个站点 p, X 在结点 p 是局部频繁项目集 。

1.2 几种经典的分布式挖掘算法

1.2.1 CD 算法

CD 算法的基本思想是将全局候选项目集和全局频繁项目集存储在分布式环境的每个站点上,利用 Apriori 算法计算出候选集在本站点数据集上的支持数,在做一次同步后,各站点之间交换本地的候选项目集的支持数,目的是使得每个分布式环境中的站点都得到候选项目集全局支持数,全局频繁项目集 L_k 也就产生了。

CD 算法是在 Apriori 算法基础上的一种较为典型的并行算法。首先,每个站点各自独立地、并行地进行本地数据库扫描,计算出各个候选项目集在局部数据库上的支持度。然后各站点交换各候选集的支持度,得到全局支持度,若候选集的全局支持度大于用户定义的最小值,则该项集为全局频繁项目集。CD 算法需要在每一次扫描结束后进行一次同步,以获得全局支持度,同步操作的网络量很大,其复杂性为 $O(n^2)$。显然,随着站点数的增多,该算法的网络通讯量将快速增加。因此 CD 算法的扩展性需要进一步提高。

1.2.2 DD 算法

DD 算法也是基于 Apriori 的并行算法,为计算本地候选项目集的全局支持数,每个站点的处理机需计算候选项目集在本地的支持数以及在所有其它站点上的支持数,在这一步既要将本地计算出的候选项目集的支持度广播到其它站点,同时还要接收从其它站点发来的支持度计数,网络通信开销很大,对站点间通信速度要求很高。为了更有效地利用存储空间,每个站点中存储不同的候选项目集,这样当站点数量增加时,可增加一步计算的候选项目集数量。为了提高算法的并行效率,该算法将候选项目集均匀分布在各个站点,这样可以通过增加站点来减少扫描的候选项集数目。但随之而来的是提高了站点间的数据传送量,算法的效率明显降低。

1.2.3 FDM 算法

分步式环境下局部频繁项目集与全局频繁项目集之间存在着一定的关系,FDM 算法利用了这种关系,在计算频繁项目集减少了站点间信息传输量,降低了系统的网络通信开销。在各站点候选数据集计算出后,在各站点都可以利用项目集间的关系,采用剪枝技术对候选数据集进行裁剪。在这种情况下候选数据集的数据量可以减少到 CD 算法数据量的 10% ~25%。

2 一种新的频繁项目集挖掘方法

2.1 数值属性或分类属性的离散化或概化

本研究对象主要针对已得到了广泛应用的分布式关系数据库。在关系数据库中包含数值属性、分类属性等多个不同属性的字段,挖掘前首先要将这些数值属性(或分类属性)的字段离散化(或概化),离散化(或概化)是根据预定义的概念分层离散化到一个普通的字符集,如 $\{a, b, c, \cdots\}$,每一个字符被看成事务数据库中的项,这样,关系数据库就转换为项目数等长

的事务数据库[10]。

普通字符集中的字符来表示的项没能反映项与项之间存在的联系,项目集的字符项有的是同一字段转换来的,而同一字段转换来的项不可能存在于频繁项目集中,这种性质的存在可用于候选项集的"剪枝",从而在很大程度上降低了候选项集数量。即如果属性离散化(或概化)后的项其本身能反映它们是同一字段,后续的挖掘算法可以将这些项直接删除,从而减少候选集的个数,提高挖掘效率。

基于以上分析,可对关系数据库中的数值型字段的离散化采用如下方法:对于布尔属性的字段将它映射到如 $B1$、$B2$ 两个字符串集,对于分类属性的字段,根据它的分类值的个数 n 映射到如 $C1,C2,\cdots,Cn$ 等 n 个字符串集;对于数值属性的字段,使用用户定义的概念分层对数值属性离散化到如 $S1,S2,\cdots,Sm$ 等 m 个字符串集,可以发现,第一个字母相同的项为同一字段转换的项,它们之间不可能形成关联规则。在确定候选项集时,可以直接删除。

2.2　频繁项集的挖掘

(1)为降低多次扫描数据库带来的时间开销,将等长的事务数据库中每个记录项映射成存放于内存的二维数组 $S[n,m]$。其中,m 由事务数据库中的项目数多少来确定,n 由事务数据库中的记录条数来确定,各二维数组元素的值由该项是否出现在某一记录项来确定,若某一记录中出现该项,则其值取逻辑值 $t.$,否则取逻辑值 $f.$。

(2) 首先将所有项作为候选项,计算每个项对应的数组元素中逻辑值为 $t.$ 的数组元素的个数,即计算 $S[n,i]$,i 为一常量。若其中为 $t.$ 值的元素个数大于定义的最小值,则该项为频繁 1 - 项集,所有的频繁 1 - 项集构成了频繁 2 - 项集的候选项集。

(3) 频繁 k - 项集的挖掘。频繁 k - 项集的候选集 C_k 由 L_{k-1} 产生,候选集的产生分为二步:先采用 Apriori 算法[3] 中的"连接" 步算法,再对候选集进行"剪枝";然后再将包含同一字符开头的项删除,进行二次"剪枝"。然后,对剪枝后的候选集进行判断,判断其是否为频繁项集,即将候选集中的项所对应的数组元素 $S[n,i]$、$S[n,j]$ 按次序进行逻辑与运算,即 $S[n,i] \wedge S[n,j]$,运算结果中为 $t.$ 值的总个数大于用户定义的最小支持度计数的该候选项为频繁 k - 项集。

(4) 考虑到"频繁项集的子集必定也是频繁项集"这一特性,当 L_{k-1} 的个数不大于 k 时,算法终止。这是因为,若 L_k 是频繁项目集,则 L_k 所包含的 k 个 $k-1$ 元素子集必定包含在 L_{k-1} 子集,即如果 L_{k-1} 中频繁项目集的个数若小于 k,则不可能再产生频繁 k-项集。

2.3　算法描述

设分布式数据库中各站点的数据的预处理已完成。

算法描述:

输入:离散化后的数据库 D;最小支持度阈值 min_sup

输出:D 中的频繁项集 L

方法:

1) ARRAY = F(D) //将每一事务项转换成数组

2) L_1 = find_frequent_1-itemsets(ARRAY) //根据 min_sup 得出频繁 1-项集

3) for(k = 2, $|L_{k-1}|$ < k; k++) {

4) C_k = generate_attribute_combinations(L_{k-1}) //产生候选集,根据 Apriori 性质及属性约束关系进行两次"剪枝",产生候选集

5) for each C_k {

6) if $A_i[i,j] \wedge A_j[i,j] \wedge \cdots \wedge A_k[i,j]$ =. t. (i = 1,2,…,n)

7) c. count ++; } // A_i、A_j、A_k 表示候选集中项所对应的二维数组,对候选集中的所有项所对应的数组元素按次序进行逻辑与运算,统计结果为. t. 的总个数。

7) $L_k = \{c \in C_k \mid$ c. count ≥ min_sup$\}$ }

8) RETURN $L = \cup_k L_k$

3　SDAM 算法

SDAM 算法的基本思想是各局部站点产生局部大项集并将其发送到中心站点,由中心站点来判断其是否为全局大项集。

3.1　局部站点算法思想

在任一站点经过 $k-1$ 次迭代生成候选集。对于每一个候选集中数据项集候选集,扫描本站点数据库,计算局部支持度合计数。如果该数据项集在该站点不是局部频繁项集,则将其从候选数据项集中删除,将局部频繁项集发往中心节点,接收由中心节点发来的全局大的数据项集。

3.2　中心结点算法

输入:各个结点的 LL_k^i

输出:传送全局大的数据项集到各个节点

k = 1;

$HLL_{ki} = \Phi$;

all = 0;

do {

for i = 1 to n 接收 i 结点的 LL_k^i;

if $LL_k^i == \Phi$ then all = all++;

if all = = n then exit;

else then

for p=1 to n{

if p<>I then 在 $HLL_{(k-1)I}$求 maxsupp(X_k);

sum=sum+maxsupp(X_k);

}

if sum>δ. D then {

for all $x_k^i \in LL_k^i$ {

for j=1 to n{

if $x_k^j \notin HLL_{ki}$ then{

$RCV(j,k)=RCV(j,k) \cup X_k^j$

向 j 结点发送 RCV(j,k)收集 j 结点的所有 x_k^j并将所有的 x_k^j进行累加结果存入相应的 x_k^i;

}

将所有的 $x_k^j \in HLL_{ki}$进行累加结果存放在相应的 x_k^i

}

If x_k^i<δ. D then $LL_k^i=LL^{ik}-X_k^i$;

}}

Insert LL_k^i into HLL_{ki}

将 LL_k^i发送到各个结点;

k=k+1;}

4 算法实验

为了验证算法的有效性,在以下环境对算法进行了测试。5 台 Pentium Ⅵ 1.5GHz 微机,内存配置为 512MB,作为局部站点存储局部数据库,1 台 Pentium Ⅵ1.8GHz 微机,内存配置为 512MB,作为服务器。每个站点使用的操作系统为 Windows XP,服务器的操作系统为 Windows 2000 Server,挖掘算法用 C++语言实现,数据库为 SQL Server 2000,实验数据从网上下载[11,12]。理论分析与实验结果表明,该算法的挖掘效率高于 SDMA 算法,是可行的。

5 结束语

分布式系统下关联规则挖掘算法的时间取决于频繁项目集的确定和站点间的网络通讯量[8]。基于二进制数组的频繁项目集挖掘方法提高了频繁项目集的挖掘效率,星型网络结构下的 SDAM 降低了网络的站点间的网络通讯量。理论分析与实验结果表明,算法提高了挖掘效率。

(1)数据预处理后,从转换后的数据库到二维数组表示时只需要一次扫描数据库,克服了经典 Apriori 算法多次扫描数据库所带来的时间开销;

(2)算法的主要运算是二维数组元素间的逻辑运算,相对于其它运算而言,逻辑运算的效率要高很多,且算法的时间复杂度为 $O(n)$;

(3)局部-全局的通信模式的采用减少了各个局部站点的通信量。

参考文献:

[1] Agrawal R, Imielinski T, Swami A. Mining Association Rules between Sets of Items in Large Databases[C]//Proceedings of the 1993 ACM SIGMOD Conference. Washington D C: [s. n.], 1993:207-216.

[2] Bayardo R, Agrawal R. Mining the most interesting rules [C]// In: Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. San Diego, CA, USA: [s. n.], 1999:145-154.

[3] Han J, Kamber M. DataMining: Concepts and Techniques [M]. Beijing: Higher Education Press, 2001.

[4] 黄贤英,王柯柯,范 伟. 基于星型网络的分布式关联规则挖掘算法研究[J]. 计算机科学,2007,34(12):180-181.

[5] 剧立伟,姜 浩,蒲安建. Web Service 架构下的分布式关联规则挖掘研究[J]. 计算机技术与发展,2009,19(4):31-34.

[6] 吉根林,韦素云. 分布式环境下约束性关联规则的快速挖掘[J]. 小型微型计算机系统,2007(5):882-885.

[7] 吴 青,傅秀芬. 水平分布数据库的正负关联规则挖掘[J]. 计算机技术与发展,2010,20(6):113-117.

[8] 孙 超,董一鸿,邰晓英. 改进的分布式关联规则安全挖掘算法[J]. 计算机工程,2009,35(12):109-110.

[9] 王爱平,王占凤,陶嗣干,等. 数据挖掘中常用关联规则挖掘算法[J]. 计算机技术与发展,2010,20(4):105-108.

[10] 王 芳,王万森. 关系数据库中关联规则挖掘的一种高效算法[J]. 微机发展(现更名:计算机技术与发展),2004,14(9):20-22.

[11] 邹 丽,孙 辉,李 浩. 分布式系统下挖掘关联规则的两种方案[J]. 计算机应用研究,2006(1):77-78.

[12] 陈 耿,倪巍伟,朱玉全,等. 基于分布数据库的快速关联规则挖掘算法[J]. 计算机工程与应用,2006(4):165-167.