基于星型网络的分布式关联规则挖掘算法研究

黄贤英　王柯柯　范　伟
（重庆工学院计算机系　重庆 400050）

摘　要　随着 Internet 的迅猛发展，分布式数据库得到广泛应用。本文分析了一些主要的分布式数据挖掘算法的优缺点，提出了一种在星形结构下的分布式关联规则挖掘算法(SDAM)。该算法改进了 FDM 算法，具有通讯量低、并行性及可扩展性好等优点。

关键词　数据挖掘，关联规则，分布式数据库，并行计算

Study of Star-based Distributed Association Rules Mining Algorithm

HUANG Xian-Ying　WANG Ke-Ke　FAN Wei
(Department of Computer, Chongqing Institute of Technology, Chongqing 400050)

Abstract　With the rapid development of Internet, distributed database has been become a broadly used environment. The advantage and disadvantage of the main distributed association rules mining algrithms are analyzed, a new distributed association rules mining algrithm based on star network struture is proposed. This algorithm has the advantage of high efficiency in communication, good extension in parallel computation.

Keywords　Data mining, Association rules, Distributed, Parallel computation

随着 Internet 的发展，分布式数据库是一种极其广泛的应用环境。如银行的存取款数据、大型超市的客户销售记录都保存在不同的区域服务器中。因此在分布式环境下利用 Internet 网络环境进行数据挖掘是目前数据挖掘领域研究的主要问题之一。

关联规则是数据挖掘的一个重要研究领域，它能发现不同商品(项)之间的联系，发现顾客购买的行为模式，有利于有效设计商品货架、对用户分类等。Agrawal 等早在 1993 年[1]提出了挖掘顾客交易数据库中项集间的关联规则问题，之后人们在关联规则领域进行了大量深入研究，尤其是在分布式并行挖掘方面，出现了数据分布算法(DD)和记数分布算法(CD)。DD 算法的着眼点是优化分区，CD 算法的重点是在任意水平分区的基础上进行并行计算。CD 算法在每一次扫描后结点之间通讯量为 $O(n^2)$，因此它不是一种扩展性好的分布式算法[4,7]。

为了减少分区间的通讯量，D. W. Cheung 在 1996 年提出了对 CD 算法进行改进的 FDM 算法[4]。FDM 算法在每个分区运行 apriori 算法，找出局部大项集，然后在每个结点全局大项集之间进行支持度合计数交换，每一次扫描后结点间的通讯量为 $O(n)$。FDM 算法是在一种无共享资源的环境下设计的，即各个结点之间形成一个网状拓扑结构的网络环境[4]。

然而，在实际应用中，网络拓扑结构通常是星型结构，如银行系统总行和分行之间、大型超市的总店和分店数据库等。本文在 FDM 算法的基础上提出了一种基于星型网络的分布式关联规则挖掘算法 SDAM(Star-based Distributed Association Rules Mining Algrithm)。该算法具有实用性强、并行可扩展性好等优点。

1　SDAM 算法的基本思想

SDAM 算法的基本思想是在星型结构下讨论分布式挖掘算法。基于 Aprioi 算法，首先将项目长度为 1 项作为候选集，扫描数据库，找出支持度大于最小支持度 minsup 的项集，称为局部大项集。然后将局部大项集发送到中心结点，判断其是否为全局大项集。在长度为 k 的大项集的基础上生成长度为 $k+1$ 的大项集，再次进行数据库扫描，最后找出所有的大项集。该算法同 *FDM* 的区别在于，发现一个结点中局部大项集并不是发往各个结点，而是同中心结点进行信息交换。

设 $I=\{i_1, i_2, \cdots, i_m\}$ 项目的集合，事务 t 是 I 的子集。设事务数据库 DB 是由事务 t 组成的集合，数量为 D。$\overline{DB}=\{DB^1, DB^2, \cdots, DB^n\}$，大小分别为 $\overline{D}=\{D^1, D^2, \cdots, D^n\}$。项集 $X_i \in I$ 在第 j 个分区的支持度记为 $\sup(X_i, DB^j)$，并用 x_i^j 表示。对于用户给定的支持度 $0 \leqslant \delta \leqslant 1$，我们说 X_i 是全局大的，当且仅当 $\sup(X_i, DB) \geqslant \delta \times D$。$X_i$ 在 j 结点是局部大的，当且仅当 $x_i^j \geqslant \delta \times D^j$。各结点与中心结点间的通讯是依靠传送$\langle i, x_i^j \rangle$的消息来实现的。$i$ 表示项集 X_i，而 j 表示传送信息的结点号。在描述算法之前，给出几个定义和定理，在此不做证明。

定理 1　若项集 X_i 在节点 i 是局部大的，则所有的 X_i 的子集在节点 j 也是局部大的。

定理 2　若项集 X_i 是全局大的，则存在一个结点 $p(1 \leqslant p \leqslant n)$，$X_i$ 在结点 p 是局部大的。

定义 1　若项集 X_i 是全局大的，在结点 p 是局部大的，则称 X_i 是在 p 结点上的强项集。

定理 3　如果一个项集 X_i 是全局大的，则存在一个结点 $p(1 \leqslant p \leqslant n)$，使得 X_i 是在结点 p 上的强项集。

用 GL_i 表示在结点 i 的全局大的数据项集的集合，$GL_{i(k)}$ 表示在结点 i 的长度为 k 的全局大的 k 项集集合，$L_{(k)}$ 表示第 k 次迭代后产生的全局大项集集合。在每个结点，设 $CG_{i(k)}$ 为对 $CL_{i(k)}$ 运行 Apriori_gen 算法生成的候选项集集合，也就是：

黄贤英　副教授，硕士，主要研究方向：软件工程应用、数据挖掘。

$CG_{i(k)}$ = Apriori_gen($GL_{(k-1)}$)，用 $CG_{(k)}$ 来表示集合 $\bigcup_{i=1}^{n} CG_{i(k)}$。

定理 4 假设 $CG_{i(k)}$ = Apriori_gen($GL_{(k-1)}$)，对于每一个 $k>1$，所有的全局大的 k 项集集合 $L_{(k)}$ 是 $CG_{(k)}=\bigcup_{i=1}^{n} CG_{i(k)}$ 的子集。

定理 4 说明只要在每个结点的强项集的基础上产生候选集，即可进行下一次迭代，此定理是 SDAM 算法的基础。

2 SDAM 算法描述

在 SDAM 中，运行在各结点的算法实现局部剪枝，在每个结点 $j(1\leqslant j\leqslant n)$ 执行以下步骤：

(1)生成候选集。根据在结点 j 经过 $k-1$ 次迭代生成强项集的基础上，利用公式

$CG_{j(k)}$ = Apriori_gen($GL_{(k-1)}$)

生成 $CG_{j(k)}$。

(2)局部剪枝。对于每一个数据项集 $X_i \in CG_{j(k)}$，扫描数据库 DB^j，计算局部支持度合计数 x_i^j。如果 X_i 在结点 i 不是局部大的，那末将其从候选数据项集 $LL_{i(k)}$ 中删除。

(3)支持度交换。$LL_{j(k)}$ 发往中心结点。

(4)接收由中心节点发来的全局大的数据项集。

运行在中心结点的算法实现全局剪枝。假设 X 是 k 次迭代结束后大小为 k 的候选数据项集。在中心结点都已收到所有 X 的大小为 $k-1$ 的子集的局部支持度合计数。对于一个结点数据库 DB^i，用 maxsupi(X)来表示 X 的所有大小为 $k-1$ 的子集的最小的局部支持度的合计数，即：$\text{minsup}_i(X)=\min\{Y.\,sup_i \mid Y<X$ 且 $|Y|=k-1\}$。所有分支数据库中这类上界函数的和就是 $X.\,sup_i$ 的上界，用 maxsup(X)来表示，即：$X.\,\text{sup}\leqslant \max\,\text{sup}(X)=\sum_{i=1}^{n}\max\,\text{sup}_i(X)$，可以用它来进行全局剪枝。即如果 $\text{maxsup}(X)<\delta * D$，那末 X 就不可能成为一个候选数据集。在合计数开始交换之前，结点 i 对余下的候选元进行全局剪枝。候选数据集 X 的一种可能的全局支持度合计数上界为：

$$X.\,\text{sup}_i+\sum_{j=1,j\neq i}^{n}\max\,\text{sup}_j(X)$$

因为 $X.\,\text{sup}_i$ 在局部剪枝后获得，所以上界可以在中心结点被计算出来，并用于对数据集进行剪枝。带全局剪枝的中心结点算法可描述为：

```
输入：各个结点的 LL_k
输出：传送 HL_k 到各个节点
k=1;
HLL_ki=Φ;
all=0;
do {
  for i=1 to n 接收 i 结点的 LL_k;
  if LL_k==Φ then all=all++;
  if all== n then exit;
  if 接收到 i 结点的 LL_k and LL_k≠Φ then
  for p=1 to n { //全局剪枝
      if p<> i then 在 HLL_(k-1)I求 maxsup_p(X_k);
      sum=sum+maxsupp(X_k);
      }
if sum>δ.D then {
  for all x_k^i∈LL_k^i{
      for j =1 to n {
          if x_k^i∉ HLL_ki then{
            RCV(j,k)=RCV(j,k)∪X_k^i;
            向 j 结点发送 RCV(j,k)收集 j 结点的所有 x_k^i;
            将所有的 x_k^i 进行累加结果存入相应的 x_k^i;
            }
          将所有的 x_k^i∈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_k^i-X_k^i;
    }
  }
  insert LL_k^i into HLL_ki;
  将 LL_k^i 发送到各个结点;
  k=k+1;
}
```

3 算法分析

(1)算法的复杂度分析

在 SDAM 算法中，如果项集 X 在结点 i 是局部大的，则其通信量的复杂性为 $O(n)$，这与 FDM 算法是相同的。而 CD 算法需要通讯量为 $O(n^2)$。

(2)算法的并行代价分析

假设每个结点的分区具有相同的大小，具有 D/n 个事务，假设需要挖掘的项目数为 m，则最多可有 2^m 个项集。在最坏的情况，每一次对数据库 D 的扫描时间为 t_s，则串行情况下，算法的扫描时间为 $2^m\times t_s$，即串行复杂度为 $O(2^m)$。各个分区的并行运行时间为 $t_s\times 2^m/n$。

并行算法的代价 $c=t*n$，这里 t 为并行算法所需的时间，n 为结点的数量。显然 $c=2^m\times t_s$，即 SDAM 算法在挖掘关联规则的并行算法执行代价在阶的意义上等于最坏情形下串行求解此问题所需的运行时间，可见 SDAM 算法具有代价最佳的并行性。

(3)算法的并行伸缩性分析

一般并行算法的效率为 $E_p=S_p/n$，S_p 为算法的加速比，n 为并行结点数。所谓并行算法的可伸缩性是指当处理机数目不变的情况下，如果 E_p 随着问题规模的扩大而单调递增，则称这种算法是具有可伸缩性的并行算法，而并行算法的效率表示为：

$E_p=S_p/n=1/(1+T_r/T_c)$

由于 $T_r=T_f\times m$，$T_c=T_s\times 2^m$，因此

$E_p=1/(1+T_f\times m/T_s\times 2^m)$

随着数据库规模的变化，E_p 的分母将严格减小，即 E_p 单调递增，所以 SDAM 算法具有良好的可伸缩性。

(4)算法的加速比分析

在各个结点需要等待的时间为中心结点在每次迭代结束后向各结点通讯的时间，考虑最坏的情况，每次中心结点向各个结点的发送时间为 T_f，则总的发送时间为 $m\times T_f$。根据阿达尔(Amdahl)定律，SDAM 的最大可能加速为

$$S_p<\frac{1}{f+(1-f)/p}$$

其中 f 为串行执行部分的时间，p 为结点数目，S_p 为最大加速比。

设 T_c 表示并行算法所需要的时间，T_r 表示并行算法所需要的额外时间(包括通讯、同步和空闲等待时间等)，则算法的加速比为 $S_p=T_c/(T_r(T_c+T_r)/n)$。由于

$T_r=T_f\times m$，$T_c=T_s\times 2^m$

因此并行算法的效率表示为

$$S_p=\frac{T_s*2^m*n}{T_f*m*(T_s*2^m+T_f*m)}$$

$$=\frac{n}{m*T_f/T_s}*\frac{2^m}{(2^m+T_f/T_s*m)}$$

其中 $T_f/T_s<1$，当 $m\to+\infty$ 时，S_p 将逼近 n，所以 SDAM 算法具有较好的算法加速比。

结束语 具有中心结点的星型网络拓扑结构的分布式关联规则挖掘算法，充分利用了局部剪枝和全局剪枝技术。在各个结点利用局部剪枝技术，而在中心结点利用全局剪枝技

(下转第 188 页)

其拒绝域应为：$W=\{F>c\}$，对给定的显著性水平 $\alpha(\alpha=0.05)$，c 由 F 分布的 $1-\alpha$ 分位数 $F_{1-\alpha}(r-1,n-r)$ 确定。

当各正态均值有显著差异时，假设发生 $F>F_{1-\alpha}(r-1,n-r)$ 的概率为 p，则称 p 为显著性概率[3]。若 p 值小于 α，则拒绝零假设，各组均值有显著差异，即因子与考察指标有关。反之，若 p 值大于 α，则不能拒绝零假设，各组均值无显著差异，即因子与考察指标无关。

将客户流失样本中的各基本属性与目标变量输入程序，即可得出图 2 的分析结果。

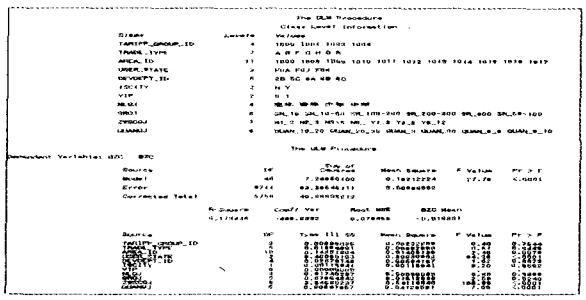

图 2　因子分析：目标维变量_各特征分量

由图 2 可以得出，与目标变量无关的属性有：TRAIFF_GROUP_ID(费率类型)、TRADE_TYPE(行业)、DEVDEPT_ID(营销、统计属性)、ISCITY(地处)、VIP(VIP 客户)、NLQJ(年龄区间)；与目标变量有关的属性有：AREA_ID(地域)、USER_STATE(用户状态)、SRQJ(消费额区间)、ZWSCQJ(在网时长区间)、QUANQJ(话务量区间)。

3　客户流失聚类模型的建立

聚类分析是数据挖掘中的一种主要技术，是把一组个体按照相似性分成若干类别，使得属于同一类别的个体之间的距离尽可能地小，而不同类别上的个体间的距离尽可能地大。

聚类分析主要分为层次聚类和迭代的平方误差分区聚类。层次方法按群组的嵌套顺序组织数据，以树状图或树形结构来表示。平方误差分区算法试图得到一个使类内分散最小而类间分散最大的分区[3]。

在客户流失分析中采用的就是迭代的平方误差分区聚类。设 m 是样本参与聚类的属性个数，n 是样本的个数，S 是由用户预先设定的分类数目，聚类分析问题可描述为：给定 m 维空间 R^m 中的 n 个向量，把每个向量归属到 S 聚类中的某一个，使得每个向量与其聚类中心的“距离”最小[4]。

根据上述因子方差分析的结果，剔除样本数据中与分析主题无关的属性，保留相关属性参与聚类算法，得出聚类模型，如图 3。

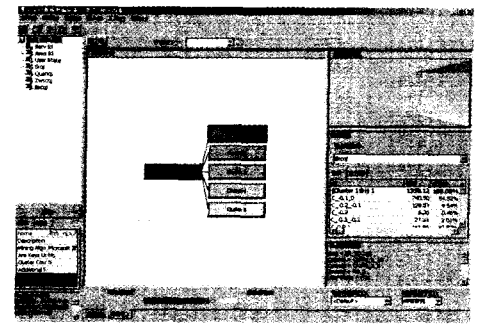

图 3　针对相关特征分量的客户流失聚类模型

由图 3 可以看出，聚类时选取了相关属性，把集合聚合为 5 类。例如，第一类节点的数据，根据每个字段代表的意义，可以解释为：消费额区间为 50～200 元，话务量区间为 3～20 个小时，在网时长区间为 1～5 年，用户状态为正常流失，地域为营销中心。从特性数据区中，可以看出这类人群的消费比在－0.1～0 区间时的流失的可能性为 54.52%，消费比在－0.1～－0.2 时的流失的可能性为 9.54%，……。

这类人群已经有了较为显著的特征：消费额、话务量、在网时长都处于中等水平，且为主动流失的，而且限定了地域范围。基本达到了客户细分的功能，为营销策略的构建提供了依据。

总结　本文针对电信客户流失主题，采用了聚类方法对特征属性进行分类，鉴于聚类算法的无监督性，结合因子方差分析方法对聚类进行监督，构建的聚类模型，基本达到了细分客户的目的，能有效地反映流失客户不同群体的共同特征。从实践中证明，用该方法构建聚类模型是可行的方案。

参 考 文 献

1　邵峰晶，于忠清. 数据挖掘原理与算法. 中国水利水电出版社，2004. 2～15

2　高惠璇. 实用统计方法与 SAS 系统. 北京大学出版社，2001. 20～30

3　Chen M S, Han J, Yu P S. Data mining: An overview from database perspective. IEEE Transac-tions on Knowledge and Data Eng, December 1996. 866～883

4　Fayyad U. Data Mining and Knowledge Discovery. Making Sense Out of Data. IEEE Expert Intelligent Systems and their Applications, 1996

(上接第 181 页)

术。该算法具有很强的实用性，因为网络的拓扑结构采用星型结构，这样网络结构更加适合现实系统的应用现状。另外，从网络的建设的成本以及网络的管理和维护方面比 FDM 系统采用的网状结构相比较，具有成本低及易于管理的优点。

在大规模的分布式数据库中进行关联规则的挖掘是数据库技术与并行处理技术两者相结合的一种方法。由于数据规模的进一步扩大，分布式关联规则的网络结构将会变化成为树形结构。本算法经改变，也可以适合树形结构的网络拓扑结构。

参 考 文 献

1　Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conference on Management of data, 1993. 207～216

2　宋宝莉，覃征. 分布式环境下关联规则的安全挖掘算法[J]. 计算机工程，2006，21

3　陈涛，石伟胜，陈启买. 关联规则的并行挖掘算法研究[J]. 现代计算机，2006(7)

4　Bayardo R, Agrawal R. Mining the most interesting rules. In: Proc. of the ACM SIGKDD Conf on knowledge Discovery and Data mining, San Diego, CA USA, 1999. 145～154

5　Cheung D W, han J, Ng V, et al. A fast distributed algorithm for mining association rules. In: Proc. 1996 Int Conf. Parallel and Distributed Information Systems. Miami Beach, Florida, Dec. 1996. 31～44

6　陈耿，倪巍伟，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)

7　Schuster A, Wolff R. Communication-efficient Distributed Mining of Association Rules. In: Proc. 2001 ACM SIGMOD Int Conf. Santa Barbara, California, May 2001. 473～484