大型决策表分解方法研究*)

王加阳 刘柳明 罗 安
(中南大学信息科学与工程学院 长沙 410083)

摘 要 数据的海量性和复杂性是当前决策表数据分析中面临的难题,分解是处理大型决策表复杂特性、提高分析效率和质量的有效手段。讨论了大型决策表分析存在的问题和决策表分解的必要性,提出了评价分解方法的三条标准,重点对几种决策表分解方法进行了分析和比较,指出了其特点与不足,提出了进一步研究的方向。

关键词 大型决策表,分解,粗糙集理论

Research on Decomposition of Large Decision Table

WANG Jia-Yang LIU Liu-Ming LUO An
(School of Information Science and Engineering, Central South University, Changsha 410083)

Abstract The great quantity and complexity of data are difficulties in analysis of decision table. Decomposition is an effective tool to deal with large decision table, it can improve the efficiency and quality of data analysis. The problem in large decision table analyzing and the necessity of decomposition are discussed in this paper, three standards are proposed for evaluating the decomposition methods. Typical methods for decomposition of decision table are analyzed and compared, several problems are pointed out for further research.

Keywords Large decision table, Decomposition, Rough set theory

1 引言

现实应用中,数据量的不断增大给现有的数据分析和处理技术提出了挑战,许多实际的决策表包含大量的对象和属性,结构复杂,若直接对其进行挖掘,计算复杂度高,而所得到的规则质量较低。

分解是解决大型决策表问题的一种有效方法,将一个大型决策表分解为若干规模较小且易于处理的子表,在子表中进行规则获取,可以减少每次处理的数据量,避免直接在复杂系统中建模的困难和缺陷,提高数据分析的效率和质量。目前,大型决策表的分解已得到学术界的广泛重视,并被认为是数据转换最有效的方式之一。

本文针对决策表分析存在的问题,对决策表分解的必要性及其优势进行论述,提出衡量分解方法质量的标准,从规则质量与分类精度、决策等价性和分解算法的效率等方面分析比较当前存在的决策表分解方法。

2 决策表分析存在的问题

决策表是一种将条件属性和决策属性区分开来的知识表示系统,由对象集、条件属性集和决策属性集组成,是信息系统的一种特殊情况,为数据集中的规则推导和知识发现提供基础,广泛应用于各种数据分析理论与方法中。

决策表可以表示为一个四元组:$T=(U,R,V,f)$,其中论域U是对象的集合,$R=C\cup D$是属性集合,子集C和子集D分别是条件属性集与决策属性集,$V=\bigcup_{a\in R}V_a$是属性值的集合,V_a表示属性a的值域,$f:U\times A\rightarrow V$是指定U中对象的属性值的信息函数。对于含有多个决策属性的决策属性集D,可以将其转换为单一的决策属性,本文中只考虑决策属性集只包含一个决策属性d的情况,即$D=\{d\}$,$T=(U,C\cup\{d\},V,f)$。

大型决策表的特点是包含大量的对象或属性,或者属性的取值较多,使得现有的数据分析方法适用范围受到限制。现实问题中待处理的原始数据量已经达到太比特(1000千兆字节)数量级,数据规模庞大、结构复杂,带来了决策表分析与处理上的诸多问题:

(1)计算复杂度上升,由于大多数规则推导和归纳学习算法的计算复杂度都随着属性或对象数量的增加而增大,大型决策表必然导致算法执行所需时间增多;

(2)分类精度降低,搜索空间的增大给建立正确有效的分类模型带来困难。若属性数量庞大而相应的对象数量较少,学习过程中可能出现过度拟合现象,影响分类精度;

(3)存储容量受到限制,学习算法执行前需要将大量的训练数据由外存读入主存,若数据量持续增加,主存容量小将成为制约算法效率的瓶颈。

现实应用中,决策表的复杂性主要来自于属性数量的增长。随着属性集的不断扩大,为了建立有效的分类模型,避免出现过度拟合现象,训练集中所需的对象数需呈指数级增长[1],另外,根据规则的最短描述长度原理[2],若得到的分类规则前件中属性过多,将影响规则质量。所以针对决策表的复杂性首先考虑对属性集的处理,力求减小属性集的规模。

目前,大多数研究集中于属性约简方法[3~5],在保持分类能力不变的前提下删除决策表中不相关或不重要的属性,从

*)基金项目:湖南省自然科学基金资助项目(06JJ20075);国家自然科学基金资助项目(60474041);国家十五攻关计划资助项目(2002BA218C)。王加阳 博士,教授,主要研究方向为粗糙集理论、智能计算与决策支持。刘柳明 硕士研究生,主要研究方向为智能信息处理、粗糙集理论及应用。罗 安 教授,博士生导师,主要研究方向为智能控制、电力系统自动化。

而减小属性集的规模，其中一些算法取得了较好的效果。但属性约简技术仍存在一些弊端：某些情况下，决策属性与大多数条件属性都密切相关，必要的条件属性很多，经过约简后的属性集可能仍然庞大；约简算法的结果依赖于训练集中对象的数量，若对象较少，约简的质量将受到影响；另外，某些约简算法对于大型决策表效率较低，计算复杂度高。

3 决策表的分解

针对大型决策表的复杂性和属性约简等技术存在的问题，分解是一种较好的处理方法，其基本思想是将原问题分解为若干规模较小、互不相同但存在联系的子问题，子问题更加简单易处理，可用现有的工具和方法解决，然后将它们组合起来解决原问题。分解方法在运筹学[6]、人工智能[7]和工程设计[8]等领域中得到关注和应用，但并没有引起知识发现和机器学习界的重视。

决策表的分解是从对象集或属性集的角度将原决策表分解为若干规模较小的子表的过程。决策表分解后，各子表本身仍然是一个完整的决策表，但对象集或条件属性集的规模较小，分析与处理较原决策表简单。相对于其它的数据转换方法，分解方法的优势在于：

(1)提高数据分析效率：针对大型决策表的海量性，分解减少了每次处理的数据量，使得适合普通决策表的算法也能适用于复杂的大型决策表，各子表之间可以进行并行计算，减少时间复杂度；

(2)增强可理解性和数据挖掘过程的透明度：分解后在子表内进行小样本建模，子模型更易于理解，适合由用户驱动的数据挖掘，还可能通过分解发现数据中隐藏的规律(如发现不同属性或对象之间的关系等)；

(3)提高分类精度和算法的性能：在分解后的子表中针对其各自的特点使用不同的归纳学习方法，克服每种方法各自的缺点，算法的性能和规则的分类精度可得到提高；

(4)局部模型的独立性：数据动态更新的过程中，重新构建全局分类模型的代价很大，而分解后只需要对部分子模型(局部模型)进行重构。

大型决策表分解中需要考虑的关键问题是，对于给定的问题，如何选择最合适的分解方法。因为后续数据分析和规则获取的质量主要取决于分解的效果，所以需要对各种不同分解方法的特点进行分析和比较，供实际应用时选择。

4 决策表分解方法分析

Kusiak 将决策表分解方法大致分为两类[9]：基于属性集的纵向分解方法和基于对象集的横向分解方法，实际上这两种分解也可以同时发生，例如某些对象集分解过程中也包含对属性集的分解。对不同分解方法的质量和特征，主要从以下三个方面进行评价：

(1)规则质量与分类精度：分解后各子表进行规则推导，由合成的知识或者局部知识对新对象分类，规则质量和分类精度是否得到提高；

(2)分解前后的信息无损和决策等价性：某些分解过程中可能存在原决策表重要信息的损失，综合子表归纳的规则得到的决策结果也可能与原决策表不一致，分解方法应尽量保证分解前后决策分类上的等价性；

(3)分解算法的时间复杂度：分解中要寻找对属性集或对象集进行合理划分的依据，这将占用分解过程的大部分时间，由于分解的主要目的之一是减小原决策表分析的时间复杂度，分解算法本身的时间复杂度必须控制在可接受的范围内。

4.1 基于函数分解的方法

Blaz Zupan 等将函数分解的思想应用于决策表的分解问题[10]，假设 T_F 是一个初始决策表，有条件属性集 $C=\langle c_1,\cdots,c_n\rangle$ 以及决策属性 d，条件属性集与决策属性的关系用函数表示为 $d=F(C)$，一次分解的过程是将该函数分解为 $d=G(A,H(B))$ 的形式，其中 $A\cup B=C$，$A\cap B=\varnothing$，分解的过程产生了两个子决策表 T_G 和 T_H，函数 G 和 H 分别表示 T_G 和 T_H 中决策属性与条件属性的关系。该分解过程得到一个中间决策属性 $m=H(B)$，当 A 中属性的取值一定时，对于 B 中各属性取值的任意两种组合，若对应的决策属性 d 值相等，则 B 上的这两种属性值组合是相容的，对应有相同的中间决策值 m。这样，子集 B 中属性的个数直接决定了算法的复杂度，为减少属性值的组合次数，属性子集 B 的规模将受到限制。

该方法的关键问题是在一次分解过程中，如何恰当地选择对属性集的划分，选择产生的中间决策属性 m 取值最少的划分，可以减少组合计算次数和子决策表 T_H 的规模，降低复杂度。上述分解过程保证了函数 G 和 H 与函数 F 的一致性，即由子决策表 T_G 和 T_H 得到的决策结果与原决策表 T_F 相同。该分解过程递归地应用于 T_G 和 T_H，直到满足一定的分解终止条件为止(如子表的个数、子表中的属性数量达到要求)，产生一系列具有层次关系的属性子集。

该方法以函数分解理论为基础，将大型决策表分解为具有层次关系且规模较小易于分析的子表，子表中的属性数量和对象数量均得到减少，各子决策表的复杂度低于原决策表的复杂度，提高了原决策表数据分析和规则获取的效率，并且通过建立属性集上的层次结构，得到有意义的中间决策概念，从而描述复杂决策逻辑。将分解算法应用于现实数据集，再进行分类规则获取，与普通分类算法(如决策树算法 C4.5)的实验比较结果[11]充分表明，该分解方法有利于提高分类精度，当数据集规模较大时效果更为显著。

但是，属性值域的大小直接影响算法的复杂程度，若属性取值较多，不同属性值之间组合次数的增多会增大计算开销，所以该算法的时间复杂度通常较大。另外，该分解方法仅适用于相容决策表，容噪能力较差，当条件属性之间缺乏内在层次关系时，显露其弊端。

4.2 属性聚类分解方法

文[12]提出一种基于属性聚类的决策表分解方法，利用粗糙集理论[13]计算条件属性之间的依赖度，从而对条件属性聚类，形成多个独立的条件属性子集来分解决策表。属性依赖度借鉴粗糙集理论中分类质量的思想，根据粗糙集理论，决策表 $T=(U,C\cup\{d\},V,f)$ 的分类质量定义为：

$$\gamma=\frac{|POS_C(d)|}{|U|} \tag{1}$$

其中 $POS_C(d)=\bigcup_{X\in U/d}\underline{C}(X)$ 是决策表的相对正域，即 $\gamma=\sum_{X\in U/d}\frac{|\underline{C}(X)|}{|U|}$，(1)式也可看作条件属性集 C 与决策属性 d 之间的依赖度，同理，条件属性集 C 中任一属性 c_i 都可确定一个等价关系，将条件属性 c_i 对 c_j 的依赖度定义为：

$$k(c_i,c_j)=\sum_{X\in U/c_j}\frac{|\underline{c_i}(X)|}{|U|} \tag{2}$$

对于决策表 T 的条件属性集 C，计算任意两个条件属性

之间的依赖度，根据属性依赖度构造的相关矩阵进行谱系聚类，聚类结果中每个条件属性子集与决策属性结合构成一个子决策表，完成对原决策表的分解，该算法的时间复杂度为 $O(|C|^2|U|^2)$。

通过属性聚类进行分解的优点是基于粗糙集理论和聚类技术，可行性强，得到的子决策表之间相对独立。但是根据(2)式属性 c_i 对 c_j 的依赖度 $k(c_i,c_j)$ 和属性 c_j 对 c_i 的依赖度 $k(c_j,c_i)$ 不一定相等，给聚类过程带来一定困难。聚类得到的子决策表中各条件属性之间依赖程度较大，各属性对论域的划分存在较高的相似度，使得子决策表不能体现原决策表的分类特性，得到的规则支持度偏低，质量较差。

4.3 基于属性核的分解方法

根据粗糙集理论，核是指决策表中对决策起关键性作用的重要属性。文[14]根据决策表中核属性和非核属性对于决策的区别，给出了一种决策表分解方法，将原决策表分解为两个子表，并产生一个中间决策属性。其中第一个子表的条件属性集由原决策表的核属性组成，决策属性即为新产生的中间决策属性，对于原决策表中核属性上的相容对象，中间决策属性的值为原决策表中的决策属性值，对于不相容对象，中间决策属性值由核属性对论域的划分确定。第二个子表的条件属性由原决策表中核属性之外的条件属性以及中间决策属性组成，决策属性即为原决策表的决策属性。

这种属性集的分解过程同时也是一个分步决策过程，核属性在决策过程中起重要作用，所以首先根据核属性所在的子决策表产生的规则对新对象分类，但该子表产生的规则并不能保证给所有的对象准确分类，其中存在不相容情况，而核以外的属性对决策可以起辅助作用。该分解算法的时间复杂度为 $O(|C|^2|U|^2)$，通过分解得到的知识和从原决策表直接归纳的知识是等价的，对新对象的分类可保证决策一致，实验证明，子决策表进行约简的复杂度较小。该分解方法存在的最大问题是只能将原决策表分为两个子表，不能完成更加细致的分解，若原决策表的核为空，就面临如何对条件属性集进行划分的问题。

4.4 对象集分解方法

在某些应用中，决策表的条件属性并不复杂，但存在庞多的对象实例，这种情况下需要进行对象集的分解。

Nguyen, S. H. 等用模板表示条件属性与其取值的对应关系，在决策表中寻找最优模板，将其作为决策表对象集的分解基准[15]。模板是若干形如“条件属性＝属性值”表达式的合取形式，是一组特定的条件属性及其取值的描述，决策表 $T=(U,C\cup\{d\},V,f)$ 的一个模板 L 可表示为 $L=(c_{l1}=v_{l1})\wedge(c_{l2}=v_{l2})\wedge\cdots\wedge(c_{lk}=v_{lk})$，若某对象满足模板 L 所包含的所有 k 个属性取值，则称该对象与模板 L 匹配。根据 Nguyen, S. H. 的观点，模板的质量由与该模板匹配的对象数和该模板中“属性＝值”表达式的个数的乘积来定义[16]，其原因是从模板的长度和支持度两方面综合考虑，避免模板在属性选择上的局限性和后续分解中子表之间的不平衡。从决策表中产生最优模板通常比较困难，但存在一些求次优模板的方法，如最大长度法、对象权值法、属性权值法、利用遗传算法产生模板等[17]。对决策表 T，首先求出 T 的最优或次优模板 L，按该模板将 T 的对象域分解为两个子域，分别对应两个子表 T_1 和 T_2：T_1 包含所有与 L 匹配的对象，$T_2=T-T_1$，若所得到的子表大小在合理的范围内，则停止分解，否则对不符合要求的子表重复上述步骤。该分解过程产生一个树形结构，根为原始决策表，树枝结点为不同的子表，每个子表都有相应的模板与之对应。

该方法突出的优点体现在规则的动态扩展和新对象的局部决策分类上。由于分解过程根据不同的模板产生了若干子决策表，若数据动态更新的过程中有新样例加入原决策表，从分解树的根结点开始将其与各子表对应的模板进行匹配，加入到相应的子表中，只需要更新该子表的规则库，而不需重新对整个决策表进行规则获取。对新对象进行分类时，将其与各子表对应的模板匹配，选择最接近的子表，根据该子表产生的规则进行分类。这种局部决策的方式不仅缩短计算时间，还可以避免原决策表中不相关因素对分类的干扰，实验证明其分类精度优于普通的算法[18]。通过信息熵的方法可证明分解过程不会引起决策表的信息损失。该方法的缺点是最佳模板缺乏统一完善的定义，最佳模板求取困难，模板太复杂会使子表数量过多，而每个子表中对象数量少，缺乏代表性，无法反映原决策表的信息。

文[19]用粗糙集的观点计算条件属性相对于决策的权重，根据权重较小的次要属性取值对决策表的对象集进行分解。在决策表 $T=(U,C\cup\{d\},V,f)$ 中，定义条件属性对决策属性 d 的重要度为：

$$w_c(d)=\frac{card(U)-card(pos_{C-c}(d))}{card(U)} \tag{3}$$

(3)式反映了条件属性 c 对决策表分类能力的影响，删除次要的条件属性不会对决策产生太大影响，因此计算各条件属性 $w_c(d)$ 的值，选择重要度最小的条件属性，按照该属性的取值来分解决策表对象域，比例最大的属性值所对应的对象被划分到一个子表中，其它对象在另一个子表中，若子表仍未达到要求，则重复上述分解步骤，一次分解过程的时间复杂度为 $O(|C|^2|U|^2)$。文[20]从分类精度的提高等方面将属性选择度量标准进行改进，并结合规则的支持度，提出了分解终止的判断标准。

这类基于属性度量的分解方法在数据量较小时适用性不强，但是在处理大型数据表时有明显的优势，可以提高计算速度。但是分解基准的选择标准还不完善，有待进一步改进。

4.5 几种分解方法的比较

上述决策表分解方法，无论是从属性的角度还是对象的角度分解决策表，都是产生属性集或对象集的一个划分，分解得到的属性子集或对象子集互不相交，一定程度上避免了子集之间相互关联带来的误差和分类时可能引起的矛盾。上述方法中，函数分解和基于核属性的分解两种方法产生了新的中间决策概念，其它方法则没有产生中间决策属性。函数分解、模板分解和基于属性度量的分解方法都是建立一个基本分解思想，然后递归地对原决策表进行分解，属性聚类方法则没有递归的过程，分解随着属性聚类一次性完成。表 1 从分类质量、决策等价性、分解效率等方面对上述几种决策表分解方法进行比较。

表 1 决策表分解方法的比较

	分类质量	决策等价性	时间复杂度	子表数量	产生新决策
函数分解	提高	等价	较高	一般	是
属性聚类	较差	不一定	较低	一般	否
基于核属性的分解	一般	等价	较低	较少	是
模板分解	提高	等价	一般	较多	否
基于属性量度的分解	一般	等价	较低	较多	否

根据 Wolpert 的"No Free Lunch"理论[21]，任何一种方法不可能对所有领域的问题都是最佳的，不同的分解方法在诸多特征和性能方面存在差异，在实际应用中要根据问题的具体情况和决策表数据本身的特点选择合适的分解方法，力求在保证决策等价的前提下，达到分解效率和分类质量的平衡。

总结与展望 分解是解决大型决策表数据海量性和复杂性问题的有效手段，本文分析了决策表分解的必要性，提出了评价分解方法的三条标准，对当前存在的几种决策表分解方法，从不同的角度分析其特征并进行比较。现有的决策表分解方法还未达到成熟完善的程度，进一步的研究方向包括建立完善的分解终止判断标准，对于递归分解的方法，目前多是结合后续分析的需要或由专家经验确定分解终止的条件，较为主观，应综合考虑多方面因素提出合理全面的标准。从运行效率和分类质量等方面对分解方法进行改进、建立决策等价性判断标准等问题也值得深入研究。

参 考 文 献

1 Jimenez L O, Landgrebe D A. Supervised Classification in High-Dimensional Space: Geometrical, Statistical, and Asymptotical Properties of Multivariate Data. IEEE Transactions on Systems Man, and Cybernetics—Part C: Applications and Reviews, 1998 (28):39～54

2 Zupan B, Bohanec M, Demsar J, et al. Learning by discovering concept hierarchies. Artificial Intelligence, 1999(109):211～242

3 Starzyk J, Nelson D E, Sturtz K. Reduct generation in information systems. Bulletin of international rough set society, 1998,3: 19～22

4 Kumar A. New Techniques for Data Reduction in a Database System for Knowledge Discovery Applications. Journal of Intelligent Information Systems,1998,10(1): 31～48

5 王清毅,范焱,蔡庆生. 知识的约简研究. 小型微型计算机系统, 2000,21(6):623～627

6 He D W, Strege B, Tolle H, et al. Decomposition in Automatic Generation of Petri Nets for Manufacturing System Control and Scheduling. International Journal of Production Research, 2000, 38(6): 1437～1457

7 Michie D. Problem decomposition and the learning of skills. In: Proceedings of the European Conference on Machine Learning, Springer-Verlag, 1995. 17～31

8 Szczerbicki K E, Park K. A Novel Approach to Decomposition of Design Specifications and Search for Solution. International Journal of Production Research,1991,29(7): 1391～1406

9 Kusiak A. Decomposition in data mining: An industrial case study. IEEE Transactions on Electronics Packaging Manufacturing, 2000,23(4):345～353

10 Zupan B, Bohanec M, Bratko I, et al. A dataset decomposition approach to data mining and machine discovery. In: Heckerman D, ed. Proc. of the Third International Conference on Knowledge Discovery and Data Mining. Irvine, CA: AAAI Press,1997. 299～303

11 Zupan B, Bohanec M, Bratko I, et al. Machine learning by function decomposition. In: Proceedings of the Fourteen International Conference on Machine Learning. Nashville, TN: 1997. 421～429

12 杨善林,刘业政,李亚飞. 基于 Rough Sets 理论的证据获取与合成方法. 管理科学学报,2005,8(5):69～75

13 Pawlak Z. Rough Sets. Theoretical Aspects of Reasoning about Data. Dortrecht: Kluwer Academic Publishers,1991

14 樊群,赵卫东,达庆利. 一种基于粗集的实例分解归纳学习方法. 管理工程学报,2001,15(2):79～81

15 Nguyen S H, Skowron A. Searching for Relational Pattern on Data. In: Komorowski J, Zytkow J, eds. In: Proceedings of First European Symposium on Principles of Data Mining and Knowledge Discovery. Trondheim, Norway: Springer Verlag,1997. 265～276

16 Nguyen S H, Nguyen T T, Polkowski L, et al. Decision rules for large data tables. In: Proceedings of Symposium on Modeling, Analysis and Simulation. France, Lille,1996. 942～947

17 Nguyen S H, Polkowski L, Skowron A, et al. Searching for Approximate Description of Decision Classes. In: Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery. Tokyo, Japan,1996. 153～161

18 Nguyen S H, Skowron A, Synak P, et al. Knowledge discovery in data bases: Rough set approach. In: Mares M, ed. Proceedings of the Seventh International Fuzzy Systems Association World Congress. Academia, rague,1997. 204～209

19 马昕,孙优贤. 面向大型数据表的粗分析方法. 计算机工程与应用,2003(16):198～200

20 王庆东,马昕,戴华平,等. 基于粗糙集属性量度的数据库分解方法. 浙江大学学报(工学版),2004,38(9):1196～1199

21 Wolpert D H. The lack of a priori distinctions between learning algorithms. Neural Computation,1996(8): 1341～1390

(上接第 183 页)

表 4 Web 数据库的误差率

minsup	BIRCH		MARC		Sampling	
	$e_F(B)$	$e_R(B)$	$e_F(M)$	$e_R(M)$	$e_F(S)$	$e_R(S)$
10%	0.361	0.412	0.021	0.012	0.031	0.022
20%	0.213	0.127	0.020	0.012	0.032	0.027
30%	0.288	0.256	0.020	0.009	0.033	0.028
40%	0.279	0.301	0.020	0.011	0.036	0.030
50%	0.261	0.204	0.022	0.010	0.037	0.029

表 5 关联规则的误差率

min-conf	Census			mushroom			Web		
	$e_R(B)$	$e_R(M)$	$e_R(S)$	$e_R(B)$	$e_R(M)$	$e_R(S)$	$e_R(B)$	$e_R(M)$	$e_R(S)$
10%	0.378	0.007	0.026	0.366	0.011	0.154	0.362	0.013	0.021
20%	0.391	0.005	0.027	0.198	0.009	0.159	0.471	0.009	0.022
30%	0.511	0.006	0.027	0.117	0.008	0.168	0.412	0.012	0.022
40%	0.414	0.006	0.028	0.293	0.009	0.171	0.402	0.009	0.022
50%	0.376	0.006	0.028	0.312	0.008	0.172	0.365	0.010	0.023

总结 实验结果表明，与 BIRCH 和 Sampling 算法相比，在运行时间和算法结果上综合比较，MARC 优势明显，虽然在运行时间比较上，BIRCH 和 MARC 耗时基本相同，但是由于 BIRCH 只适合开采数字数据，在交易数据上得到的结果并不理想。

参 考 文 献

1 Agrawal R, Imielinski T, Swami A N. Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD'93). 1993. 207～216

2 谢坤武,陈世强. 一种分类数据的聚类算法. 见:全国数据库学术会议(NDBC2006). 广州,2006. 332～327

3 Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94). 1994. 487～499

4 Toivonen H. Sampling Large Databases for Association Rules. In: Proceedings of 22nd International Conference on Very Large Data Bases (VLDB'96). 1996. 134～145

5 Brin S, Motwani R, Ullman J D, et al. Dynamic Itemset Counting and Implication Rules for Market Basket Data. In: Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'97). 1997. 255～264