第16卷 第12期
2006年12月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.12
Dec. 2006

一种基于规则的离群挖掘算法

张璐璐，贾瑞玉，李 杰
（安徽大学 计算机科学与技术学院，安徽 合肥 230039）

摘 要：离群数据挖掘是指从大量数据中挖掘明显偏离、不满足一般行为模式的数据。现有的离群数据挖掘算法大多对密集的交易数据库缺乏有效的处理，文中提出了一种高效的基于规则的离群挖掘算法。该算法使用了多层最大离群支持度及最小离群兴趣度，计算1－离群条件集的幂集，并在数据结构中存储了交易标识符链表，使得扫描数据库的次数仅为一次，从而提高了挖掘的速度、效率且使得结果更具有决策意义。文中使用此算法对某一商场的部分销售数据库进行了实验，结果表明该算法能有效、迅速地发现密集数据库中的离群数据。

关键词：数据挖掘；离群数据；离群挖掘；支持度；兴趣度
中图分类号：TP311.138　　**文献标识码**：A　　**文章编号**：1673－629X(2006)12－0073－03

An Algorithm for Outliers Mining Based on Rule

ZHANG Lu-lu, JIA Rui-yu, LI Jie
(Department of Computer Science and Technology, Anhui Univ., Hefei 230039, China)

Abstract: Outlier mining refers to the mining of the data with obvious departure, and with no general behaviors patterns from large amounts of data. Most existing algorithms don't have good performance when dealing with data－intensive transaction databases. This paper presents a highly efficient rule－based outlier mining algorithm. This algorithm uses multi－layer maximum outlier support and minimum outlier interest, calculates the power of 1－outlier set, and uses linked list to store the transaction identifier in data structure. All these make the time of scanning database only once. So the mining speed and efficiency are enhanced, the result are more useful to decision－making. In this paper, an experiment using this algorithm was carried out on part of a shopping center's scales database. The result presents that this algorithm can mine data－intensive transaction database more effectively and rapidly.

Key words: data mining; outlier data; outlier mining; support; interest

0 引 言

数据挖掘(data mining)[1]是一个从大量的数据中发现潜在知识的过程，是半自动或自动地从海量数据中发现模式、相关性、变化、反常规律性的过程。一般来说，数据挖掘的任务可以分成4类[2]：相关、依赖关系的发现；类别的判定；类别的描述；离群数据的挖掘(outlier mining)。

离群数据[3]就是明显偏离其它数据，不满足数据的一般模式或行为，与存在的其它数据不一致的数据。离群数据通常来源于测量错误、键入错误等人为错误，这些数据要被指出并加以修改或删除，否则，影响数据分析结果。另外，它也可能就是数据的真实性质的反映，可能比一般数据的所包含的信息更有价值，数据应予以保留。

离群数据挖掘是指从大量数据中挖掘明显偏离、不满足一般行为模式的数据，简称离群挖掘。离群挖掘已在各个领域得到广泛研究[4]，如防信用卡欺诈、天气预测、电力和用户分类等方面。但其算法众多，对于不同的应用范围其实现效果各异，如基于统计的方法需用户建立数据点的概率分布模型，应用时需事先知道数据集的分布和分布参数等信息，对数据要求比较严格，因此适用范围不大；Knorr和Ng[5]提出基于距离的离群数据挖掘方法，但此方法中的距离难以确定；Arning等[6]提出一种基于偏离的离群数据发现方法，需确定相异函数进行离群数据挖掘，但若其相异函数的选取不合适，则得不到满意结果。

文中在已有的离群挖掘算法的基础上针对交易数据库的基本特征提出了一种基于规则的离群数据挖掘算法，实验结果证明它可以为商业系统分析提供较好的数据，并为商业决策提供有用信息。

1 离群数据发现过程

为使离群数据同其他数据挖掘方法，如序列模式分析(分析异常访问模式的特征)、关联规则分析(分析非频繁、高兴趣度的规则)、链接分析(识别不同人和活动之间的联系)自然接轨，更便于专业人员对离群数据的理解分析，提出如图1所示的离群数据发现过程。首先对数据进行预处理，包括空值、数据规范化等处理，从而得到便利的观测序列，再利用离群数据发现算法对其进行挖掘，由

收稿日期：2006－03－31
作者简介：张璐璐(1983－)，女，安徽阜阳人，硕士研究生，研究方向为数据挖掘、机器学习。

离群条件集得到相应的离群数据。专家链接分析平台将分析得到的离群数据同其他相关数据库进行链接分析，利用反馈结果来进行相关参数的调节。

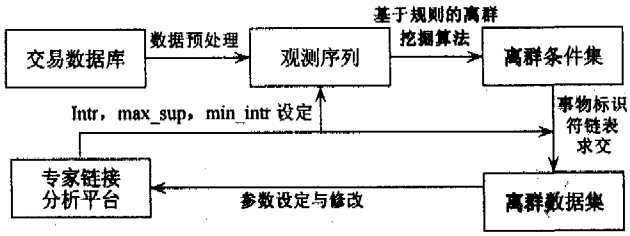

图 1 离群数据发现过程

2 基于规则的交易数据库离群挖掘

离群数据挖掘的方法比较多，主要有统计学的方法、基于距离的方法、基于偏差的方法等。这里采用基于规则的离群挖掘。关联规则挖掘的目标是满足最小支持度、最小信任度的属性值及属性值组合，而离群数据挖掘的目标是搜索小于某一阈值(即最大离群支持度 max_sup)的数据项集。

2.1 预备知识

定义 1 条件项 c 定义为由属性、属性值组成的合取式形式：$c_{i1} \wedge c_{i2} \wedge \cdots \wedge c_{ik} \wedge \cdots c_{ij}$，其中 c_{ik} 的意义为 $a_{ik} = v_{ij}$，a_{ik} 为属性；v_{ij} 为属性取值；$k = 1,2,\cdots,j$；j 为条件项的长度。条件项的集合组成条件项集 C，$c \in C$。

例如：交易数据库 D 中，一项交易的属性 a_1，含义为“咖啡”；属性 a_2，含义为“面包”，条件项 c：$(a_1 = 0) \wedge (a_2 = 1)$ 表示顾客在此项交易中没有购买咖啡，购买了面包。

定义 2 假设 A 是一个集合，A 的所有子集的集合称为集合 A 的幂集，记作 power(A)。

例如：$A = \{a,b,c\}$，则 power(A) = $\{\Phi,\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\}$。

定义 3 如果交易集合 D 中满足条件项 c 的交易有 s 项，则称条件项 c 的支持度为 s，即条件项 c 出现的次数记为 sup(c)。

例如：在交易数据库 D 中，有 50 项交易中购买了面包，没有购买咖啡，则 sup(c) = 50。

定义 4 只有条件项 o_i 的支持度小于某一阈值，则 o_i 才有可能是离群条件项，这一阈值为最大离群支持度，记为 max_sup；条件项 o_i 的支持度小于 max_sup，表示为 $c <$ max_sup。

在文中，i 不同，o_i 对应的 max_sup(o_i) 也不相同，i 越大，max_sup(o_i) 越小。其具体的计算方式如下：

$i = 1$ 时，max_sup(o_i) = 30(用户定义的确定值)；

$i \neq 1$ 时，max_sup(o_i) = max_sup(o_1) × ($n - i$)/(n 为候选 1 - 项集的元素个数)

定义 5 在不同的环境中，人们对属性的兴趣程度不同，因此有必要对每一项属性都赋予一个确定的值以表示人们感兴趣的程度，记为 intr(a_i)。对于条件项，规定兴趣度按如下方式计算：

$\text{intr}(c_i) = \sum_{k=1}^{j} a_{ik} \times \text{intr}(a_{ik}) / \sum_{k=1}^{j} a_{ik} (k = 1,\cdots, j)$；$j$ 为条件项 c_i 的长度。

兴趣度值越大表示该属性或条件项更重要。只有条件项 o_i 的兴趣度值大于某一阈值，则 o_i 才有可能是离群条件项，这一阈值为最小离群兴趣度，记为 min_intr。

例如：intr(a_1) = 3，intr(a_2) = 5，表示属性 a_2 比 a_1 更应被重视。

c：$(a_1 = 1) \wedge (a_2 = 1)$，则 intr($c$) = 4；

在算法中，由于对各个属性均引入了兴趣度值，则只有同时满足 $0 <$ sup(o_i) $\leqslant$ max_sup(o_i) 且 intr(o_i) $\geqslant$ min_intr 的条件项才被称为离群条件项。O_{set} 为离群条件项集。

定义 6 离群规则可表示为：$O_{set} \Rightarrow O$，O 为离群条件集，O_{set} 为真时，数据库中对应的数据对象集称之为离群数据，在交易数据库中即为离群条件集 O_{set} 为真时对应的交易集合。

引理 1 如果 o_i 是离群条件项，c_i 为另外的条件项，则 $p_i = o_i \cup c_i$ 也是离群条件项。

引理 2 对于条件项 o 和 c，若满足 o 的事务标识符链表为 Tid_list(o)，满足 c 的事务标识符链表为 Tid_list(c)，则同时满足 o 和 c 的事务标识符链表为{Tid_list($o \cup c$) = Tid_list(o) and intr(item_set)$\geqslant$min_intr}。

2.2 算法描述及复杂度分析

(1)首先逐个扫描交易数据库，产生 1 - 候选集 C_1，在扫描每一项交易时，除了对每一项进行计数以外，还要记录包含该项的交易标识符 Tid。这样，扫描完一遍数据库之后，在得到的候选集 C_1 中，每个项都包含了一个相应的交易标识符链表。C_1 的每一个元素的结构如下：

(项 item，支持度 sup(item)，兴趣度值 intr(item)，交易标识符列表 Tid_list(item))

然后，从 C_1 中删除大于最大离群支持度的条件项，则此时 C_1 即为 1 - 离群条件集 O_1。

(2)求出项集 O_1 的幂集，此时幂集中的每一个元素结构如下：

(项集 item_set，支持度 sizeof(Tid_list(item_set))，兴趣度值 intr(item_set)，交易标识符列表 Tid_list(item_set))

然后，从幂集中删除：大于最大离群支持度的条件项、支持度为 0 的条件项、兴趣度值小于最小离群兴趣度的条件项。此时密集中的项集为 O_{set}，且 O_{set} 中各元素的交易标识符链表 Tid_list(item_set) 的并即为所求离群数据。

算法具体描述如下：

```
Input:交易数据库 TDB,最大离群支持度 max_sup,最小离群兴趣度 min_intr
Out:Result = 离群条件集;o = 离群数据集
Begin
Result = Ø;
C1 = create_candidate_1(TSB);//创建 1 - 候选集
```

$O_1 = \{c \in C_1 \mid 0 < \sup(c) \leqslant \max\_sup\}$;//生成 1 - 离群条件集

$V = \text{power}(O_1)$;//求出 1 - 离群条件集的幂集,且按照数据结构,计算每一元素的属性值

If(V could be inserted in EMS memory whole)

Then { Scan(V);}

Else{divide V into V_i; | $V_i \in V$, $V_i$could be inserted in EMS memory whole}

For($V_i \in V$)

Scan(V_i)

Endif;

Result = {item_set | item_set $\in V$, $0 < \sup(\text{Item_set}) \leqslant \max_sup(\text{Item_set})$and $\text{intr}(\text{item\_set}) \geqslant \min\_\text{intr}$};//得到离群条件集

$O = \text{Tid\_list}(r_1) \cap \text{Tid\_list}(r_2) \cdots \cap \text{Tid\_list}(r_k) \mid r_1, r_2, \cdots, r_k$ 为 Result 中的所有元素;

//得到离群数据集

End

基于规则的交易数据库离群挖掘算法计算复杂度,与交易数据库中交易个数 n 有关,与 1 - 离群条件集 O_1 中元素的个数 k 有关($k \ll n$),其计算复杂度为 $O(n + 2^k)$。由于 k 很小,当 n 很大时,算法复杂度是与 n 线性相关的。k 的数量与数据集的属性个数有关,由于利用了引理 1 、引理2的特性,k 的数量还与数据库本身的特性(如数据的分布),以及最大支持度、最小兴趣度的选取有关。如果 k 较大,可以通过相应的办法进行控制,如,降低最大支持度、增加最小兴趣度等。

利用本算法计算条件项的支持度,效率是很高的。例如,假设有一个 8 项元素的 O_1,有一条交易记录含有 O_1 中的 4 项。当扫描 TDB 时,用一般的离群挖掘算法计算至少要检验 15 次,计数 15 次。而采用本算法,则只需检验 1 次,计数 16 次,效率得到很大提高。

2.3 实验与结果

为了检测算法是否有效,应用某食品超市交易数据库作为数据源,目标是找出兴趣度高的离群数据,并由决策者对其进行分析。首先扫描数据库,得到 1 - 离群条件集,然后求出 1 - 离群条件集的幂集(包括各个项集的属性),最后根据判定条件得出离群条件集和离群数据集。数据如表 1、表 2 所示。

表 1　交易数据库

Tid1	牛奶	面包	水果	肉类	速冻品	调味料	禽蛋	…
Tid2	面包	肉类	速冻品	调味料	巧克力			…
Tid3	酒水	蔬菜	水果	肉类	饼干	巧克力		…
Tid4	牛奶	水果	调味料	禽蛋				…
Tid5	牛奶	面包	瓜子	肉类	速冻品	禽蛋		…
Tid6	面包	水果	速冻品	巧克力	调味料			…
…								…

表 2　货品兴趣度

名称	牛奶	方便面	酒水	面包	咖啡	蔬菜	水果	肉类	速冻品	巧克力	瓜子	饼干	调味料	食用油	禽蛋
intr	2	2	5	2	6	1	3	3	2	7	3	2	3	3	2

max_sup = 30;当未考虑兴趣度时:

离群条件项集:饼干 ∧ 瓜子(support = 2)、饼干 ∧ 咖啡(support = 1)、饼干 ∧ 食用油(support = 1)、饼干 ∧ 瓜子 ∧ 咖啡(support = 1)。

离群数据集:Tid8,酒水,蔬菜,肉类,速冻食品,饼干,调味料,食用油。

Tid13,酒水,咖啡,蔬菜,水果,肉类,瓜子,饼干。

Tid88,酒水,蔬菜,水果,肉类,速冻食品,瓜子,饼干,调味料,禽蛋。

当考虑了兴趣度,且 min_intr = 3 时:

离群条件项集:饼干 ∧ 咖啡(support = 1, intr = 4)、饼干 ∧ 瓜子 ∧ 咖啡(support = 1, intr = 3.7)。

离群数据集:Tid13,酒水,咖啡,蔬菜,水果,肉类,瓜子,饼干。

实验表明文中的方法是正确的,算法中加入了属性的兴趣度,能够更加突出地显现使决策者感兴趣的离群数据,从而更好地起到帮助决策的作用。

为了更好地应用文中的方法,应注意以下几个方面:

(1) 如果 max_sup 较大,离群数据较多,计算时间增加,反之运算时间明显加快。

(2) 如果 min_intr 较小,离群数据较多,计算时间增加,反之运算时间明显加快。

(3) 可以使用增量技术,用户选择某个支持度,计算了一次结果后,改变支持度时,不用全部从头开始计算,可以大大减少运行时间。对于兴趣度亦可如此。

3　结论与展望

现在的离群数据的挖掘方法大部分是从距离的角度查找离群数据,而文中从另一角度:基于规则来着手离群数据发现的研究。提出了一种新的基于规则的离群挖掘算法。其主要思想是离群数据在某些属性上总是远离大部分的正常数据。通过设置和两个阈值有效的挖掘离群数据,算法中数组方式存储 1 - 条件集的幂集,并采取了特殊的数据结构,使得算法效率大大提高。通过对食品数据库中的离群数据的发现及分析,证明了算法的有效性及正确性。实验表明:应根据数据的特性选取渐变的最大离群支持度,随着条件项数增加,最大离群支持度变小,属性的值域范围增大,更易发现离群数据及属性建有逻辑关系的数据。该算法的提出有效解决了密集的交易数据库的离群挖掘问题,今后将对算法作进一步的改进,以便用于其他各种类型的数据库。

参考文献:

[1] Fayyad U, Piatetsky - Shapiro G, Smyth P. From data mining to knowledge discovery: An overview[C] // Advances in Knowledge Discovery and Data Mining. USA: AAA I/M IT Press, 1996.

(下转第 78 页)

表 1 自定义 session 监视器中的属性及属性值

属性	代码	默认属性值
确定框标题	ajaxTimer.title="Your Title";	超时通知
确定框内容	ajaxTimer.message="Your session is ending!";	你目前的 session 已经超时……
更新按钮文本	ajaxTimer.confirm="Update";	确定
取消按钮文本	ajaxTimer.ignore="Cancel";	忽略
最大空闲时间	ajaxTimer.maxWait=1000 * 60 * 5;	5 分钟
空闲消息	ajaxTimer.extendedMessage="You lost your data!"	你的 session 已经到期……

2.4 同时刷新引起对 session 请求的竞态条件

竞态条件是一个在设备或者系统试图同时执行两个操作的时候出现的不希望的状况,但是由于设备和系统的自然特性,为了正确地执行,操作必须按照合适顺序进行。在网络中,竞态条件会在两个用户同时试图访问同一个可用信道的时候发生,在系统同意访问之前没有计算机能得到信道被占用的提示。

编写传统的 WEB 页时,每个页面都有自己唯一的请求、唯一的执行环境,并且用户每一次都只访问一个页面。每个页面的请求都来自于一个唯一的用户,而且每个请求到达服务器的时间都有先后顺序,不和其他的页面请求共享数据。

使用 AJAX 的时候,这一切都改变了,由于 AJAX 的多次刷新,一次页面访问就可能同时产生多次的请求。不同页面的进程不能共享数据,但是同一个页面的异步请求则可以共享数据。如图 2 所示,一个页面装载时产生了有效的 session 数据,在初始化时产生了两个异步的 AJAX 请求,几乎在同一时间,一个请求想要读取 session 中的某一数据,另一个请求想要向 session 中改写这个数据的值,这就产生了竞态条件,这时,你不知道是哪个请求最先到达服务器端,最后这个 session 读取的数据的值只能由到达服务器的时间的先后来决定了,造成了 session 的数据值不确定[5]。

产生这个问题的原因是网络传输的不确定性,这个问题的解决有两种方法。

第一,可以把这个问题的决定权交给用户。如上所示,可以给每个 Request 添加一把锁,对 session 中要访问的资源锁定,如果 Request1 先到达服务器,当 Request2 到达服务器时,会发现 Request2 所要的变量已经被 Request1 锁定,这时,Request2 返回一条消息,让用户决定是否继续 Request2 的写操作,同样,如果 Request2 先到达,则返回消息,让用户决定是否继续 Request1 的读操作。但是,这种解决方法牺牲了用户的友好性,如果产生的消息太多的话,用户会很反感,可行性不大。

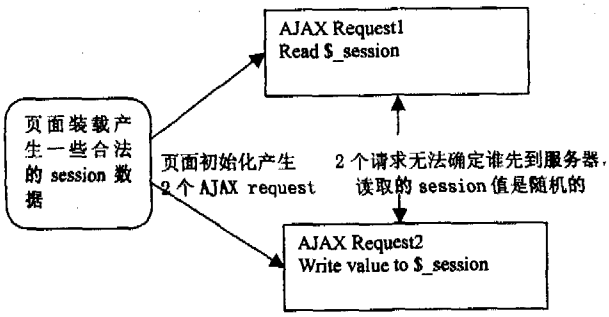

图 2 AJAX 引起 session 随机取值

第二,既然问题产生的根源是网络传输的不确定性,那么,可以去掉网络传输,把 session 从服务器端搬到客户端来,这样,所有的问题就都迎刃而解了。事实上,现在 Client/SOA 模式就是让 session 在客户端进行管理,只不过现在所用的大多是 HTML + + 模式而 Client/SOA 模式离成熟应用还需要不少努力。所以,在目前最好的方法还是避免同时的产生请求对 session 的同一数据进行操作。

3 总 结

文中主要研究了 AJAX 技术中对原有的 session 所造成的危险,从原理上进行了介绍并指出了解决的办法。对 AJAX 技术进行了总体的介绍,对 AJAX 的优点做出了概括。相信随着时间的推移,AJAX 技术会进一步得到改进,越来越多的网站将会使用 AJAX 技术,使得 AJAX 能够在 WEB 中站有一席之地。

参考文献:

[1] Grance D, Pascarello E, James D. Ajax in Action[M]. USA: Manning, 2005.

[2] 夏 桅. AJAX with ASP.NET[J]. MSDN 开发精选, 2005(4): 230 - 231.

[3] Session 详解[EB/OL]. 2005 - 03 - 12. http://dev2dev.bea.com.cn/bbs/jishudata/ArticleShow.jsp? Id=10.

[4] 徐雪霖. Web 数据库访问技术探悉[J]. 微计算机信息, 2004, 20(20): 86 - 87.

[5] Troubles with Asynchronous Ajax Requests and PHP Sessions [EB/OL]. 2005 - 03 - 22. http://www.chipmunkninja.com/article/asyncsessions.

(上接第 75 页)

[2] Knorr E, Ng R. Algorithms for mining distance - based outliers in large datasets[C]// In: Proc of the 24 VLDB Conf. New York, USA: [s.n.], 1998: 392 - 403.

[3] 史东辉, 蔡庆生, 倪志伟, 等. 基于规则的分类数据离群挖掘方法[J]. 计算机研究与发展, 2000, 37(9): 1094 - 1100.

[4] 范 明, 孟小峰. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2001: 223 - 261.

[5] Edwin K, Roymond N. Algorithms for mining distance - based outliers in large database[C]// In: Proc of the VLDB Conf. New York: [s.n.], 1998: 392 - 403.

[6] Arning A, Agrawal A, Raghavan P. A linear method for deviation detection in large database[C]// Int Conf on knowledge Discovery in Databases and Data Mining. Portland: [s.n.], 1996: 169 - 184.