**哈尔滨工业大学**

**硕士学位论文开题报告**

**题 目： 多元时间序列流跨事务关联规则的挖掘**

**院 （系） 计算机科学与技术学院**

**学 科 计算机技术**

**导 师 张春慨**

**研 究 生 孙雅琦**

**学 号 14S151001**

**开题报告日期 2015/9/22**

**研究生院制**

**二〇一五年九月**

目录

[1课题来源及研究的背景和意义 1](#_Toc430427542)

[1.1课题来源 1](#_Toc430427545)

[1.2研究的背景和意义 1](#_Toc430427549)

[2国内外在该方向上的研究现状及分析 2](#_Toc430427543)

[2.1关联规则的挖掘 2](#_Toc430427545)

[2.2时间序列上的规则挖掘 3](#_Toc430427549)

[2.3多元时间序列流上时间相关的关联规则的挖掘 5](#_Toc430427550)

[3主要研究内容 5](#_Toc430427551)

[3.1数据的预处理 5](#_Toc430427552)

3.1.1分段线性化近似………………………………………………………….…...6

3.1.2时间序列分割 ..…………………………………………………………….....6

3.1.3时间序列符号化………………………………………………………...….....6

[3.2关联规则的挖掘 .....7](#_Toc430427556)

3.2.1概要存储结构……………………………………………………………........7

3.2.2剪枝策略…………………………………………………….……………........7

3.2.3具体挖掘算法…………………………………………………………..…......8

3.2.4负关联规则.........................................................................................................8

[4研究方案及进度安排，预期达到的目标和取得的研究成果..............................10](#_Toc430427557)

[4.1研究方案 ..............10](#_Toc430427558)

[4.2预期达到的目标和取得的研究成果 ...10](#_Toc430427559)

[4.3进度安排 ..11](#_Toc430427560)

[5为完成课题已具备和所需的条件和经费 ..11](#_Toc430427561)

[6预计研究过程中可能遇到的困难和问题以及解决的措施 .11](#_Toc430427562)

[7参考文献 11](#_Toc430427563)

## 1课题来源及研究的背景和意义

## 1.1课题来源

随着科技的进步，经济的发展，数据的来源变得多样，但是对于海量的数据中蕴藏的意义的理解却不多。如何从海量的数据中及时发现有用的信息和知识，提高信息的利用率成为一个迫切需要解决的问题，并产生了数据挖掘技术[6]。数据挖掘是指对复杂的数据进行处理之后获得有用的规则，提出蕴藏在数据中的信息。常见的数据挖掘的方向有：频繁模式，关联规则，分类预测，聚类分析，离群点发现等等。

## 1.2研究的背景和意义

在诸多的数据类型中，时间序列流数据在实际生活中应用十分广泛。时间序列就是按照时间顺序排列的记录序列。他们广泛存在于社会生活的各个领域，如：金融证券的股票变化，生物医学中人体心电的变化等等。时间序列流数据的特点有海量性、生成速率快、潜在的无边界及不可控性[1]的特点给时间序列流数据的挖掘增加了难度。

正是由于挖掘时间序列流的难度，吸引了诸多的研究学者，关联规则挖掘是流数据挖掘的一个热门方向。通过时间序列流数据关联规则的挖掘可以应用到找到有效的商业促进方案，对未来的时间序列的趋势进行预测，提升工业控制水平的方法等等。

由于时间序列数据本身所具备的特性[22]，直接在时间序列上进行数据挖掘是不现实的，会浪费太多的时间和空间代价，并且可能会对算法的准确性和可靠性产生影响。所以对时间序列流挖掘之前对原始数据进行预处理是非常必要的。预处理就是用数据的某种形式对原始数据进行高效率表示。时间序列流数据的线性化近似就是一种预处理，能够对原始的时间序列流数据起到压缩，平滑，去噪的作用。在文献[9]中是对单个和多个时间序列流进行关联规则挖掘的。文献[5]则考虑到了时间延迟的因素对规则的影响。不同的研究人员对时间序列流数据的的处理不同，挖掘出来的关联规则的结果也是不同的。

在现实生活中，对于时间序列的分析，一般是在多元的时间序列流数据上进行的。多元时间序列的关联规则挖掘问题，可以分为多序列的事务内关联规则挖掘，多序列的跨事务关联规则挖掘。对于多序列的事务内关联规则挖掘，使用传统的关联规则挖掘算法，但是传统的事务内关联规则分析发现的规则，只能揭示多个时间序列的各个事务在同一时间点的相互关联性，对于时间序列的最终目的-----预测的贡献很少。

跨事务的关联规则描述的是在不同事务之间的关系，可以分析不同时间，不同时间序列之间的关系，分析多元时间序列不同时刻观察值集合的关联规则。如：平安银行第一天上涨，浦发银行第二天下跌=>中国银行在第三天上涨（5%，80%）[23]此关联规则的前后件有时间前后关系，对于我们利用此规则预测事件发生具有重要意义。

综上所述，多元时间序列流数据大量出现在我们的实际生活中，多元时间序列流跨事务关联规则的挖掘是一个当前比较热门的研究课题，而对于这个问题的研究具有重要意义。

2国内外在该方向上的研究现状及分析

目前国内外都有一些关于时间序列的关联规则挖掘研究，由于他们研究的侧重点不同，用的关联规则的挖掘算法不同，挖掘到的关联规则的特点也不同。一般的关联规则挖掘的步骤分为两个阶段，一是预处理，在海量的原始数据上进行压缩，平滑，去噪，线性化近似，分割时间序列，聚类等，然后在已经处理过的数据集上进行关联规则挖掘算法的实施。

## 2.1关联规则的挖掘

最经典的关联规则挖掘算法是Agrawal提出的Apriori算法，Apriori算法是一种挖掘关联规则的频繁项集算法，通过逐层搜索的迭代算法，在每次生成候选的频繁项集的时候都要经过扫描，计数，比较，连接，剪枝这几个步骤。 Apriori算法的应用广泛，消费市场价格分析，猜测顾客的消费习惯等。但是利用Apriori算法挖掘关联规则在验证候选频繁K项集的时候要对整个数据集进行扫描不止一遍，其时间效率很低。因此后来出现了许多基于Apriori的改进算法。如EH-Apriori挖掘算法在Apriori算法的基础上进行了两点改进，一是挖掘过程进行了预处理，二是将数据集的数据Hash到一个很大的表[18]。AprioriTid[19]通过在第一次扫描数据库时，仅仅保留含有频繁一项集的事务，并将在以后的挖掘频繁2项集的时候仅仅扫描此数据集，从而有效减少后期挖掘长项集时扫描数据集的大小，进而来提高数据挖掘效率。Apriori算法只有在挖掘长项集的时候才体现出其优越性。ES-Apriori算法是扩展的分步Apriori算法，减少了单次调入内存的数据量。FITI(First Intra then Inter)是跨事务关联规则的挖掘算法，它是在先求得事物内频繁项集的基础上，以数据集的形式进行转换成跨事务关联规则。。

后来学者们研究了关联规则的相关性质提出了FP-growth算法[8]。FP-growth算法是通过建立具有前缀性质的FP-tree来仅仅遍历一遍数据库，就可以挖掘到频繁模式，从而提高挖掘效率。实验证明FP-growth算法的性能比Apriori快了一个数量级。在FP-growth的基础上也出现了很多改进的算法。CFP-mine[8]算法是基于压缩的FP树，基于约束子树的方法，减少内存的调用，并且用了数组的方法，减少了FP-tree的遍历次数。ITARM算法采用分而治之的挖掘跨事务关联规则的方法，在CFPmine的基础上进行改进，对挖掘的任务进一步分解。ITF-tree是在FP-growth算法的基础上改进了概要存储结构。

传统的Apriori算法和FP-growth算法都是有一定局限性的，它需要满足数据集中所有项均衡分布，并且各项是同等重要的，但是现实生活中这样的前提并不总是能满足。为了解决所有项的重要性并不一定是完全相同的问题，文献[10,11,12]提出了带权关联规则挖掘的方法，为不同的项赋予不同的权重，从而区分各项之间的重要性。为了有效应对各项在数据集中分布不平衡的问题，文献[13,14]提出了多支持度阈值关联规则挖掘方法，就是为数据集中每个项设置一个最小支持度阈值，从而应对各个项分布不平衡的问题。文献[15]从以上两个方面对关联规则挖掘算法进行了改进。

## 2.2时间序列上的规则挖掘

利用滑动窗口[10]从时间序列中挖掘规则的方法被广泛应用到时间序列流关联规则的挖掘中。这种方法首先研究的是从单时间序列中挖掘出规则，然后将挖掘的方法扩展到多元时间序列。

还有一种被广泛用于挖掘多元时间序列跨事务的关联规则的算法是ES-Apriori[14]，这种算法通过从数据库中检测是否有符合最小支持度要求的频繁项。并删除该频繁项集的真子集。循环操作直到读完数据库中的数据为止。这个算法通过只保留最大的频繁项集，从而压缩了搜索空间，提高数据挖掘的效率。该方法在Apriori算法的基础上从减少数据库扫描次数和内存数据分而治之这两个方面进行了改进，使规则挖掘算法在时间和空间性能上都得到了极大地提升，可以很好地用于挖掘静态的多元时间序列。该方法将数据值映射到一定的区间来减少值的多样性，对于挖掘时间序列流来说这是很大的局限性，因为流中的数据是源源不断产生的，数据值也是无法确定的，属性的数据区间也不好确定。

因此，时间序列的处理方式可以有多种，不同的处理方式得到的规则类型不同，但对于数值型的时间序列一定要减少数据值的多样性，因为数据值太多对于挖掘来说效果肯定是不佳的，一些真正有意义的规则也很可能挖掘不出来。而如何运用特殊的数据结构来节省对数据库的扫描次数以提升“跨事务性”关联规则的算法性能，仍然是一个需要解决的问题。

而且Apriori等算法大多是对静态事务数据中的关联规则进行挖掘，而实际生活中的数据是随着时间变化而变化的，大多是以时间序列流的形式存在的，因此针对时间序列流数据的特点设计特定的算法来挖掘是首要任务。

## 2.3多元时间序列流上时间相关的关联规则的挖掘

文献[11] 中考虑到一组时间序列流的变化给另一组时间序列流带来的影响可能不是同步的，应有一个时间延迟。因此，其挖掘出来的规则带有时间特性。该方法挖掘出的规则类型：A上升超过5%那么B在两天内将会上升超过10%，其中A和B表示不同的时间序列。

## 3主要研究内容

在时间序列关联规则的挖掘过程中主要包括两个步骤：一是数据的预处理，二是具体的数据挖掘。

本文挖掘任务是基于时间序列流数据的。因为时间序列流数据具有的一系列的特性，实时产生使得我们不可能将数据进行完全存储然后多次遍历进行挖掘，我们计划利用滑动窗口对流数据进行约束和限制，但是滑动窗口的长度并不能确定，采用动态滑动窗口的方法进行跨事务的限制，然后挖掘关联规则的算法在FP-growth的基础上进行改进，并增加用负关联规则剔除对正常关联规则产生误导的项，提高关联规则挖掘的准确性。

## 3.1数据的预处理

在跨事务分析中，假设每一个频繁项将扩展为3(滑动窗口)个。如果采用类似于Apriori算法的先产生候选集，再进行测试的方法，假设有100个频繁项，在传统的关联规则分析中，最多会产生100\*99/2=4950个候选频繁2项集，而在跨事务分析中，会产生300\*299/2=44850个，成指数增加，增加了9倍，而继续构造3频繁项集时数目还会继续增加。随着数据增加，系统内存将会逐渐被耗尽，效率会明显降低。这时，数据的预处理就显得尤为重要，通过数据预处理压缩数据，使后续关联规则的挖掘过程更加高效。

### 3.1.1分段线性化近似

时间序列流数据由于其范围不可估，数据量大的特点为挖掘带来了许多困难，所以为了克服在时间序列挖掘中遇到的这些困难，我们必须在对时间序列流数据正式挖掘之前对时间序列流数据进行压缩处理。所以采用分段线性化近似[19]的方法来对数据进行压缩，平滑，去噪[21]。线性化方法主要就是按照时间序列的变化形态将时间序列转换为不同线段，每个线段可以直观地表示该时间段内时间序列的变化特征[23]，按照时间序列的升降态势该特征总体上可分为3类：上升、平缓、和下降，不同的变化形态表示数据间不同变化趋势。通过分段线性化处理时间序列可以被压缩为多个线段，一般情况下原数据流到线段的压缩率与设置的压缩误差有关系。

### 3.1.2时间序列分割

经过线性化近似处理之后的多元时间序列流数据的各时间序列由长短不一的变化形态，即那些线段组成；且在同一时间段内各时间序列流可能含有多个线段，即一个时间序列流在这一时间段内存在多种变化形态，对新进入滑动窗口的正在分析的基本窗口内的多元时间序列使同一个时间段内各时间序列只有一种变化形态。

时间序列分割的算法主要有：滑动窗口算法、自顶向下算法和自底向上算法[19]。这三种算法具有不同的特点，其中滑动窗口方法效率最低，而自顶向下算法偏重于全局最优化，自底向上算法偏重局部最优化。

#### 3.1.3时间序列符号化表示

经过分割后的时间序列流数据，每个时间序列在同一个时间段内只有一个变化趋势，但是现在直接挖掘线段中的关联关系还是很麻烦的，因为每条线段的长度，斜率，或者是起点都是不一样的，如果用三个参数来表示没那么会产生很多的项，因此，采用聚类的概念，将它们聚类成一个簇，然后用这个簇的中心来表示那些在这个簇中的相似的线段，这样可以压缩挖掘时的数据项的数目，挖掘的结果可能更有用。用一个聚类方法来聚类这些线段，使相似的线段在一个类内，对于流数据来说该聚类方法应该是可增量聚类的。聚类后为每个类分配一个符号，用该符号代表该类内的所有元素，这样时间序列就可由几个符号来表示。常用的聚类方法有K-means,贪婪算法等。

## 3.2关联规则的挖掘

现在所有的时间序列流数据已经是转化了的事务间数据集，因此下面主要任务就是关联规则的挖掘了。针对不同特点是时间序列流采用不同的挖掘算法，并在最经典的算法基础上改进，提高关联规则挖掘的准确性。

### 3.2.1概要存储结构

在时间序列流数据处理过程中，由于数据量太大，如果将扫描过的数据保存在内存中是不现实的会占用太多的内存，如果将其保存在磁盘中，在时间序列流数据查询和挖掘过程中经常要读取已扫描过的数据，通过磁盘来存取花费的时间代价太大，因此，流数据处理系统[21]一般在内存中维护一个概要存储结构来保存那些扫描过的数据的一些概要信息，概要存储结构[17]是关联规则挖掘中的概念，特别是在流数据挖掘的算法中，概要存储结构是一个非常重要的概念，用于保存已扫描过的数据集中有用的信息。本文虽然研究的是时间序列流，但也具有流数据的特征，因此，应该在内存中维护一个概要存储结构。概要存储结构有多种，要能高效地保存数据集中有用的信息，便于挖掘关联规则，提高算法的效率。

### 3.2.2剪枝策略

剪掉对后续挖掘无用的信息。减小问题的处理规模。

根据多元时间序列流跨事务关联规则的性质：组成事务间的所有关联规则的频繁项集的项，必定也是事务内关联规则的频繁项集的项。

利用这个性质在做跨事务关联规则挖掘之前可以先对多元时间序列进行数据预处理，利用传统的事务内关联规则挖掘方法将事务内非频繁项集的项去掉，这样可以有效的减少事务间无效关联规则的生成，进而降低算法所需的时间与空间复杂度。但这个性质只能用在具体挖掘之前，也就是仅仅能排除非频繁一项集，不能再挖掘频繁多项集时起作用，但是经过推证得出，在跨事务关联规则挖掘过程中还有一个类似的性质可以对挖掘频繁多项集时进行剪枝，从而提高挖掘效率。

跨事务关联规则中如果存在某两项或多项同属于某一个事务，那么由这两项或多项一定也是事务内频繁项集。

由这个性质我们可以在挖掘过程中对挖掘的中间结果进行筛选剪枝，从而提高挖掘效率。一般情况下挖掘关联规则最常用方法是Apriori。其思路就是在挖掘频繁k项集的基础上生成候选k+1项集，在生成候选k+1项集时可以仅对其项集内存在的事务内的频繁项集进行测试，如果存在事务内的项集，但是该项集不是事务内频繁项集，那么我们就可以将此候选项集删除，从而减少遍历数据库的次数。本方法要遍历事务内频繁项集库要花费一定的时间，不过统计表明频繁项集库的大小比原始数据库的大小降低的不止一个数量级，因此用事务内频繁项集库进行验证还是能提升效率的。

### 3.2.3具体的挖掘算法

在经过预处理之后，基本窗口内的多元时间序列流段被转换成了事务集，只需要在这个事务集上运行关联规则挖掘算法就可以挖掘出相关的关联规则。我们要研究的是时间序列流数据中的关联规则，因此采用传统的Apriori算法或者是FP-growth算法是不可行的。

#### 3.2.4负关联规则

传统算法生成的关联规则，研究重点大部分在正关联规则上，用支持度，置信度衡量关联规则的重要性，而对于负关联规则的研究非常少。负关联规则是数据集中数据项互斥关系的体现，对于决策支持中能提供很多有用的信息，有的关联规则是无用的信息，甚至是误导的规则，通过负关联规则的挖掘对其进行度量识别正负关联规则，得到有用的关联规则，去除无用的误导规则。目前现有的识别负关联规则的方法，兴趣度，相关性，相关系数等，在挖掘负关联规则时仍然存在不足。

#### 举一个简单的例子,假设有1000条交易记录,分别是买A和B,只买A,只买B两者都不买。假定给定最小支持度minsup=0.3,minconf=0.6。

**表1负关联规则**

|  |  |  |  |
| --- | --- | --- | --- |
|  | B | ¬B | 行总计 |
| A | 400 | 200 | 600 |
| ¬A | 350 | 50 | 400 |
| 列总计 | 750 | 250 | 1000 |

利用 Apriori 算法计算得出 supp(A=>B)=0.4, conf(A=>B) =0.66>minconf,则我们认为A=>B是有效的,购买A和购买B之间存在关系。

不同于传统的关联规则,负关联规则的研究对象是项目和项目之间的互斥关系,比如当某些事件发生的时候,会影响甚至抑制另外一些事件的发生。当决策者想要了解“客户在购买产品A之后会不会对其买产品B产生影响”之类的问题的时候,负关联规则具有很大的信息参考价值。

引入相关性(corrA，B)来判断所挖掘的关联规则的正负性,则可以通过比较

conf(A=>B)与supp(B)来度量A与B的相关性:

corrA,B=

其取值范围为(l,+∞)为正相关,1为相互独立,[0,1)为负相关。上个例子中corrA,B=400/600\*750/1000=0.88,所以A和B是负相关,A出现得越多会导致B出现得越少[24]。

在负关联规则研究领域,针对项目间的正负相关性,许多专家学者提出了提出了很多有建设性的算法。

P-S兴趣度:interest = (A => B)-sup(A)sup(B),即通过两项独立性来判断其关联性。置信度:杨建林综合考虑了规则准确度、规则相关度、用户主观偏好及兴趣度后,对其进行了改进,提出了置信度validity = p(AB)-p(B)。

匹配度：match = conf(A=>B) - conf (=> B)。

Brin S首次提出通过的A=>B的卡方值来判断AB的相互独立性,如果是则两者的存在对对方都没有影响,反之,调用二阶矩阵来计算它们之间的正负相关性。

这些方法都能生成正负规则,并在一定范围内削弱无用规则,但产生的规则仍有一定的误导性,不能准确的度量项目相关性程度。

## 4研究方案及进度安排，预期达到的目标和取得的研究成果

## 4.1研究方案

数据预处理过程中分析线性化近似方法，优化已有方法，增强近似度。自底向上，自顶向下相结合。在生成拟合线段时改变传统算法中均值法和插值法，用线性回归法和最小二乘法结合，能够达到更好的结果。

数据预处理改进相似度度量方法，研究分析现有的聚类方法，选取合适的聚类方法。常用的方法有K-means和贪婪算法，K-means算法需要事先确定簇的个数和簇的中心。所以相比而言，贪婪算法更加高效。

关联规则挖掘中算法的优化主要是优化概要存储结构，节省数据库的扫描次数，提升跨事务关联规则的算法性能，结合负关联规则挖掘算法提高关联规则的准确度。然后针对改进的关联规则挖掘算法采用合适的剪枝策略，减小数据的规模，从而达到提高效率的目的。

## 4.2预期达到的目标和取得的研究成果

1.优化线段线性化近似方法。

2.改进相似度度量方法，研究分析现有的聚类方法，选取合适的聚类方法。

3.目前通过调查分析，发现FP-growth算法相比Apriori算法在关联规则挖掘方面效率要更高，而FP-growth算法对于跨事务关联规则挖掘有较好的性能，若直接使用FP-growth算法进行挖掘，当频繁项集规模比较大的时候，FP-tree将可能会由于规模太大无法存于内存中，因此要进行相应的改进。期望在未来能够将FP-growth算法有一定的改进。

4.总结现有的甄别负关联规则的方法，去除传统关联规则挖掘中无用的规则，提高关联规则的实用性和适用性。

## 4.3进度安排

2015年9月-2015年11月：全面查找材料，总结，分析已存在的方法、算法。

2015年11月-2016年5月：根据课题的目的和背景，提出新的算法和模型。

2016年5月-2016年7月：进行编码、实现以及测试，分析实验结果，得出结论。

2016年7月-2016年9月：进行系统总体测试，总结。

2016年9月-2016年12月：总结研究结论，完成论文，准备答辩。

## 5为完成课题已具备和所需的条件和经费

目前，主要包括计算机一台等，基本具备完成本课题的软硬件条件。

## 6预计研究过程中可能遇到的困难和问题以及解决的措施

在研究过程中，可能遇到的问题和相应解决方法如下。

（1）预期困难：在实现各个算法的过程中可能会发现算法描述不够清楚或者存在一些技术上的困难。

解决方法：查阅相关资料，或寻求原作者或指导老师帮助。

（2）预期困难：所运用的数据挖掘算法或者优化算法效果不好

解决方法：改进数据挖掘算法或优化算法，或采用其他方法重新进行试验，或向指导老师请教问题所在。

（3）预期困难：数据集太大，算法运行花费时间长

解决方法：可以采用并行、多处理器等方式运行处理数据集，或配置高配置电脑。

## 7参考文献

1. 朱小栋, 沈国华. 流式数据上关联规则挖掘研究综述[J]. 计算机应用研究. 2010.
2. R Agrawal, T Imielinski, and A Swami. Mining association rules between sets of items in large database[C]. Proceedings of the 1993 ACM-SIGKDD international conference on management of data(SIGMOD’93), New York, NY, USA: ACM, 1993.
3. R Agrawal, and R Srikant. Fast algorithms for mining association rules[C]. Proceeding of the 1994 international conference on very large database (VLDB’94), New York, NY, USA: ACM, 1994.
4. Han J, Pei J, and Yin Y. Mining frequent patterns without candidate generation[C]. Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD’00). New York, NY, USA: ACM, 2000.
5. Joong H C, and Won S L. Finding recent frequent itemsets adaptively over online data streams[C]. Proceeding of the 2003 ACM-SIGKDD international conference on knowledge discovery and data mining (KDD’03). 2003.
6. 张宏伟, 魏东平, 王金凤. 数据流建模方法概述[C]. //第九届计算机科学与技术研究生学术讨论会文集. 2006.
7. 廖国琼, 吴凌琴, 万常选. 基于概率衰减窗口模型的不确定数据流频繁模式挖掘[J]. 计算机研究与发展. 2012.
8. C. Giannella, J. Han, J. Peia, X. Yan, and P.S. Yu. Mining frequent patterns in data streams at multiple time granularities[C]. in: H. Kargupta, A. Joshi, K. Sivakumar, Y. Yesha (Eds.), Next Generation Data Mining, AAAI/ MIT, 2003.
9. Lee C H, Lin C R, and Chen M S. Sliding-window filtering: an efficient algorithm for incremental mining[C]//Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001.
10. Han J, Cheng H, Xin D, et al. Frequent pattern mining: current status and future directions[J]. Data Mining and Knowledge Discovery, 2007.
11. W. Cheung, O.R. Za, Incremental mining of frequent patterns without candidate generation or support constraint[J], in: Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS), 2003.
12. G. Cormode, and S. Muthukrishnan, What’s hot and what’s not: tracking most frequent items dynamically[J], ACM Transactions on Database Systems. 2005.
13. G. Grahne, J. Zhu, Fast algorithms for frequent itemset mining using FP-Trees[J], IEEE Transactions on Knowledge and Data Engineering. 2005.
14. J. Han, G. Dong, and G. Yin, Efficient mining of partial periodic patterns in time series database[C], in: Proceedings of IEEE International Conference on Data Mining, 1999.
15. 董泽坤, 李辉, 史忠植. 多元时间序列中跨事务关联规则分析的高效处理算法[J]. 计算机科学, 2004.
16. 钱爱玲.复杂结构的时间序列数据挖掘与预测方法研究[D].华中科技大学.2011
17. 赵传申, 何顺刚, 杨吉宏, 陈丽霞. 基于多分类-关联规则的数据流酚类算法[J]. 计算机工程. 2010.
18. 倪萍. 流数据挖掘关键技术研究[D]. 北京邮电大学, 2010.
19. 辛演. 基于点击流分析的电子商务智能决策支持系统[D]. 江苏大学, 2003.
20. 王喆. 商务数据中关联和聚类算法研究[D]. 吉林大学, 2005.
21. 丁艳辉, 王洪国, 高明, 谷建军. 一种发现有价值的稀有数据关联规则的算法[J]. 山东师范大学学报（自然科学版）. 2005.
22. Cheung D W, Han J, Ng V T, et al. Maintenance of discovered association rules in large databases: An incremental updating technique[C]//Data Engineering, 1996. Proceedings of the Twelfth International Conference on. IEEE, 1996.
23. 常睿 Apriori算法的一种改进方法[J].计算机光盘软件与应用.2012(19)
24. 孙帆. 事务间关联规则挖掘技术研究[D]. 辽宁师范大学, 2012.