摘 要

数据挖掘是研究如何从大量的数据中获取潜在的有用信息和知识。而关联规则挖掘是数据挖掘中最成熟、最主要、最活跃的研究内容。随着证券市场的不断发展，在证券信息数据库中积累了大量历史交易数据，如何充分利用这些历史数据探寻证券市场自身的运行规律，成为人们关心的问题。特别是2005年下半年，适逢股改大潮的来临，中国股市重新振作，掀起了一波接一波的热潮，从中留下的数据又为数据挖掘提供了良好的挖掘对象，可以总结出大量有价值的规律指导投资者操作。

在整个数据挖掘的研究中，算法的研究占有特别重要的地位。数据挖掘面对的是大量数据集，算法的效率起到决定性的作用，因此，研究和改进现有的算法，有着十分重要的意义。鉴于此，本文对关联规则挖掘算法进行了研究。首先对数据挖掘作了一般性介绍，包括数据挖掘的概念、模式、挖掘的主要问题、数据挖掘系统的分类以及应用和发展趋势。然后，对数据挖掘中重要的关联规则挖掘算法做了深入的研究，分析了关联规则中经典的 Apriori 算法、AprioriTid 算法和 AprioriHybird 算法及其他学者对 Apriori 算法的改进算法，总结了算法中存在的问题；接着，详细介绍了本文内容的重点之一，一种 AprioriHybird 算法的改进算法，并把它与 AprioriHybird 算法进行了详细比较。

为了更好地挖掘股市信息，就必须结合股市的特点，特别是股票自身的运作规律，股票的走势包含了数以万计人的思维和智慧，必须通过详细和耐心的观察才能学之一二。经过长期学习、跟踪股市及模拟演练，本文决定从宏观和微观两个方面来描述股票。宏观上，把近数月的股票数据通过模糊时间序列匹配的方法转化为股票的长期参数；微观上，把近数日的股票数据通过相关实战书籍经验和模拟实战经验转化为短期参数，从而形成一套完整的参数集，为挖掘工作打下坚实的基础。这是本文内容的重点之二。

最后在 Microsoft Visual C++ 6.0 环境下完成了对股票数据的处理、算法的改进及挖掘工作。实验验证了改进的 AprioriHybird 算法的效率在一定程度上优

于 AprioriHybird 算法；同时挖掘出了大量关联规则，其中一些颇具指导意义。

关键词 关联规则；AprioriHybird 算法；股票；模糊时间序列

ABSTRACT

Data Mining aims to get previously unknown and potentially useful knowledge from a large amount of data. Association rule mining is the most developing, main and vigorous research content in Data Mining. With the development of the stock market, lots of history exchange data have been stored in database. It attracts more and more attention that how to use these history exchange data to discover the rules of the stock market. Especially at the latter half of 2005 year, stock market was happened to reform. Chinese stock market renerves and surges high tide one wave after another. The stock data of this period time become well excavating object for Data Mining. A mass of valuable rules will be discoveried to direct investors.

Exploration of algorithms plays an important role in all Data Mining research. Data Mining faces large database. The efficiency of algorithms is the most important, so it is very significant to research and improve the existing algorithms. Based on above, this thesis mainly studies the algorithms of association rule mining. Firstly, it generally introduces Data Mining, including the concepts and the patterns, main mining problems, system classifications, and the application and development trend. Secondly, this thesis researches the Association Rule Algorithm totally, which is important in Data Mining. It analyses the classical algorithms that are Apriori, AprioriTid, AprioriHybird algorithms and the improved algorithms of Apriori, and it summarizes existing problems in these algorithms. Then this thesis presents an improved AprioriHybird algorithm in detail, which is one of the key contents, and compares it with the AprioriHybird algorithm.

In order to discovery the stock market information well, we must combine stock market characteristic, especially operational rules of stock itself. The movement of stock includes thinking and wisdom of tens of thousands of people. We want to study it only through detailed and patient observation. By a long time studying and

tracking stock market and simulated operation, this thesis determines to describe stock from macroscopical and microcosmic aspects. On the macroscopical aspect, data of latest months is transformed to obtain the long-term parameters of the stock through the fuzzy time series match method. On the microcosmic aspect, data of latest days is transformed to obtain the short-term parameters of the stock through the correlative books and the simulated combat. They form a set of integrated parameters sets, and build the solid foundation for the mining work. This is the second key content of this thesis.

Finally the disposal of stock data, the improvement of algorithm and mining were completed under VC++6.0 platform. The experiments show that the efficiency of the improved AprioriHybird was superior to AprioriHybird algorithm to a certain extent. And a lot of association rules were extracted, some of them have fine instructional significance.

Keywords Association Rule; AprioriHybird Algorithm; Stock; Fuzzy Time Series

第 1 章 绪论

1.1 课题的研究背景

近十几年来，随着数据库技术的飞速发展以及人们获取数据手段的多样化，人类所拥有的数据量急剧增加，据美国 GTE 研究中心统计，全国范围内仅科研机构每天存储的新的信息量大约有 1TB(terabytes)！大量的信息给人们带来方便的同时也带来了很多问题,主要的问题就是信息过量，难以消化理解。传统的数据库系统所能做到的只是对数据库中的已有数据进行存取和简单的操作，人们通过这些数据所获得的信息量仅仅是整个数据库所包含的信息量的很少的一部分。这样，收集在大型数据库中的数据就变成了“数据坟墓”。正像 John Naisbett 的那句名言:" We are drowning in information ,but starving for knowledge"(人类正被数据淹没，却饥渴于数据)。这种状况发生的根本原因是人们创建一个数据集时往往把精力都集中在数据的存储效率的问题上，而没有去考虑数据最终是怎样使用和分析的。

“数据海洋”是一个巨大的宝库，当其积累到一定程度时，必然会反映出规律性的东西。如果数据仅仅表现为存储，那么不经过任何分析和处理的原始数据是没有价值的。只有将这些数据转化为有用的信息和知识，它们的价值才能真正体现出来。因此，从大量的、复杂的、信息丰富的数据集中挖掘隐藏在其中的有用的知识逐渐成为所有商业、科学、工程领域的迫切需要[1]。知识发现(Knowledge Discovery in Databases)和数据挖掘(Data Mining)的概念与技术就在这样的需求推动下应运而产生，并得到了迅速发展。

经过几十年的研究和实践，数据挖掘技术吸收了许多学科的研究成果，形成了独具特色的研究分支。勿容置疑，数据挖掘研究和应用具有很大的挑战性。目前，大多数学者认为数据挖掘处于广泛研究和探索阶段。一方面数据挖掘概念已经被广泛接受，而且相关的研究成果和产品得到了学者的认可，吸引了越来越多的研究者;另一方面，目前的数据挖掘研究还存在许多有待研究和探索的问题。

1.2 本文的主要研究内容及安排

随着我国经济体制改革和金融体制改革的深入，证券投资已成为社会生活的一个重要部分，股票交易作为证券投资的一种，是现代经济生活中最常见的风险投资活动。投资股票离不开股票的分析与预测，早期发展的技术分析理论是股票预测的最初代表，如道氏理论、平均线理论、江恩理论等，在此基础上发展了众多的技术指标及分析方法，加上改进的指标，已经不计其数，面对如此众多的技术分析指标，一个投资者必然无所适从，因此研究能够预测股市、辅助投资者投资的方法，帮助投资者预测和分析股市，选择股票进行投资，优化组合投资，降低投资风险，获得最大收益是非常有意义的。

证券分析主要可以从基本面和技术面着手。投资专家往往也从这两方面入手进行分析，但他们的高明之处在于对股票走势模式的识别，在他们的脑海中敏锐的洞察力和丰富的先验知识形成了一类基本模式，所以对后市的判断会相当准确。应用知识发现方法的目的是用计算机模拟人类思维、推理方式对证券进行分类、预测，其关键就是模式获取。

本文在介绍数据挖掘、关联规则基本概念的基础上，强调了挖掘对象——中国股市相关特征。并对关联规则进行了归纳和总结，对关联规则的典型挖掘算法及其基本思想进行了详细地归纳、分析和研究，对各算法之间的差别进行了客观地比较，并提出了新的改进方法。同时为了挖掘股市的实用规律，又特别研究了股市，庄股的特点，对股票数据用模糊方法提取有意义的参数，为关联规则的挖掘提供了强有力的后盾。最后通过实验来验证挖掘结果及改进算法的可行性。

本论文的主要内容共分为六个部分：

第一章 绪论，介绍研究背景、内容及工作安排情况。

第二章 介绍数据挖掘和关联规则挖掘相关概念与信息，同时介绍了本篇关联规则挖掘的对象——中国股市的有关信息。

第三章 先对 Apriori 算法及性质进行了解释，并详细介绍了 Apriori 算法的

过程及 Apriori 算法存在的性能瓶颈问题，同时比较了几种相关改进算法。在 Apriori 的改进算法——AprioriHybird 算法上又提出新改进算法，并与做了比较。

第四章 针对股市这个特殊的研究对象，为了更好地挖掘出有价值的东西，用模糊化方法对股市信息提取出一些新参数，进而转化成关联规则挖掘需要的参数，使关联规则挖掘更具有实用价值。

第五章 创建了实验平台，对股票数据进行实际挖掘，给出了结果，并对改进的新算法与以前的算法重新做了比较。

结论 对本论文进行总结，并给出了一些可以进行后续研究的建议。

第 2 章 数据挖掘、关联规则及股市简介

2.1 数据挖掘技术

计算机网络与数据库技术的发展和广泛应用，使得信息在企业发展中的重要作用越来越得到人们的认同。人们利用信息技术生产和搜集数据的能力也大幅度提高，巨量的数据库被用于商业管理、政府办公、科学研究和工程开发等，这一势头仍将持续发展下去。在这些数据背后隐藏着极为重要的商业知识，但是这些商业知识是隐含的、事先未知的。于是，如何才能不被信息的汪洋大海所淹没，从中及时发现有用的知识，提高信息利用率就显得尤为重要。在这样的背景下，新的数据处理技术——数据挖掘（Data Mining）技术便应运而生了。

2.1.1 数据挖掘的概念、功能、步骤

2.1.1.1 数据挖掘的概念

KDD (Knowledge Discovery in Databases，数据库中的知识发现)一词首次出现是在 1989 年 8 月在美国底特律召开的第 11 届国际人工智能联合会议的专题讨论会上。随后在 1991 年、1993 年和 1994 年都举行 KDD 专题讨论会，集中讨论数据统计、海量数据分析算法、知识表示、知识运用等问题。

与 KDD 意义相近的一个术语是数据挖掘(Data Mining，简称 DM)。数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的数据集中识别有效的、新颖的、潜在有用的以及最终可理解的模式的过程。模式可以看作是我们所说的知识，它给出了数据的特性或数据之间的关系，是对数据包含的信息的更抽象的描述。一般说来，KDD 意为从数据库中获取知识，它代表从低层次数据中提取高层次知识的全过程，主要流行于人工智能和机器学习界。而数据挖掘是指从数据中自动地抽取模型，主要用于统计界(最早出现于统计文献中)、数据分析、数据库和管理信息系统界。在一般的定义中，数据挖掘被看作是知识发现过程中的一个核心部分[1-3] 。
过程中的一个核心部分[1-3] 。

2.1.1.2 数据挖掘的功能

利用数据挖掘技术可以海量数据中获得决策所需的多种知识。在许多情况下，用户并不知道数据存在哪些有价值的信息知识，因此，对于一个数据挖掘系统而言，它应该能够同时搜索发现多种模式的知识，以满足用户的期望和实际需要。此外，数据挖掘系统还应该能够挖掘多种层次的模式知识。数据挖掘系统还应允许用户来指导挖掘搜索有价值的模式知识。

特别要指出的是，数据挖掘技术从一开始就是面向应用的。它不仅是面向特定数据库的简单检索查询调用，而且要对这些数据进行微观、中观乃至宏观的统计分析、综合和推理，以指导实际问题的求解，企图发现事件间的相互关联，甚至利用已有的数据对未来的活动进行预测。例如美国著名国家篮球队NBA的教练，利用某公司提供的数据挖掘技术，临场决定替换队员，一度在数据库界被传为佳话。

数据挖掘的功能概括起来有以下几个方面[3-5]：

1） 预测（Prediction）：

数据挖掘自动在大型数据库中寻找预测性知识。若预测的变量是离散的，这类问题称为分类(Classification);如果预测的变量是连续的，这类问题称为回归(Regression)。一个典型的例子是市场预测问题。

2） 聚类(Clustering)

数据库中的记录可被划分为一系列有意义的子集，即聚类。与预测模型不同，聚类中没有明显的目标变量作为数据的属性存在。聚类算法通过监测数据判断“隐藏属性”。

3） 关联分析(Association Analysis)

数据关联是数据库中存在的一类重要的可被发现的知识。若两个或多个变量的取值之间存在某种规律性，就称为关联。本领域最常见的技术是利用关联规则。关联规则的计算依赖于识别在相关数据中频繁出现的数据集。频繁出现的数据由在某事务中同时出现的数据组成。

4） 时间序列分析(Time Sequence Analysis)

时间序列数据库内某个字段的值是随着时间而不断变化的，例如股票价格每天的涨跌，科学实验，浏览网页的次序等。时间序列分析通过对时间序列的搜索，发现重复发生概率较高的模式。

5） 偏差分析(Deviation Analysis)

用来发现与正常情况不同的异常和变化，并进一步分析这种变化是否是有意的诈骗行为，还是正常的变化。如果是异常行为，则提示预防措施;如果是正常的变化，那么就需要更新数据库记录。

6）孤立点分析(Outlier Analysis)

数据库中可能包含一些数据对象，它们与数据的一般行为或模型不一致，这就是孤立点。大部分数据挖掘方法将孤立点视为噪声或异常而丢弃。而在一些应用中(如信用卡欺骗检测)，罕见的事件可能比正常出现的那些更有价值。

7）概念描述(Concept Description)

概念描述就是对某类对象的内涵进行描述，并概括这类对象的有关特征。概念描述分为特征性描述和区别性描述，前者描述某类对象的共同特征，后者描述不同类对象之间的区别。

2.1.1.3 数据挖掘的过程

在传统的决策支持系统中，知识库中的规则是由专家或程序人员建立的是由外部输入的。而数据挖掘的任务是发现大量数据中尚未被发现的知识，是从系统内部自动获取知识的过程。对于那些决策者应明确了解的信息，可以用查询、联机分析处理(OLAP)或是其它工具直接获取。而另外一些隐藏在大量数据中的关系、趋势，即使是管理这些数据的专家也是没有能力发现，那么这些信息就可以让数据挖掘来处理。

数据挖掘发现的知识通常可以表示为：概念(Concept)，规则(Rules)，规律(Regulation)，模式(Patterns)，约束(Constrains)，可视化(Visualization)。

数据挖掘过程一般由3个主要的阶段组成:数据准备、数据挖掘、结果表达和解释。知识的发现可以描述为这3个阶段的反复过程。

数据准备:这个阶段又可以分成3个子步骤:数据集成、数据选择、数据预处理:数据集成将多文件或多数据库运行环境中的数据进行合并处理，解决语义模糊性、数据处理中的遗漏和清洗脏数据等，数据选择的目的是辨别出需要分析的数据集合，缩小范围，提高数据挖掘的质量。预处理是为了克服目前数据挖掘工具的局限性。

数据挖掘：该阶段首先根据对问题的定义明确挖掘的任务或目的，如分类、聚类、关联规则发现或序列模式发现等，之后要决定使用什么样的算法。选择实现算法要考虑两个因素:首先，不同的数据有不同的特点，因此，需要采用与之相关的算法来挖掘;其次，要根据用户或实际运行系统的要求来选择。例如，有的用户可能希望获取描述型的(descriptive)、容易理解的知识(采用规则表示的挖掘方法要好于其他方法)，而有的用户只是希望获取预测准确度尽可能高的预测型(predictive)知识，并不在意获取的知识是否易于理解。

结果表达和解释：数据挖掘阶段发现的模式，经过评估，可能存在冗余或无关的模式，这时需要将其剔除。此外，还需要对结果进行可视化处理。在上述过程中，对数据挖掘质量起决定性作用的是一个速度快、伸缩性好、结果容易理解和使用并且符合用户需求的算法。当然，数据挖掘的其他步骤也是非常重要的，每一步都是下一步的基础。数据挖掘是一个反复循环的过程，如果用户对结果不满意，可以在任何时候退回到前一阶段，如重新选取数据、采用新的数据变换方法、设定新的参数值，甚至换一种挖掘算法。

2.1.2 数据挖掘的国内外现状

国际上，数据挖掘已成为当前计算机科学界的一大热点。数据挖掘应用十分广泛。目前热点集中在科学、生物医学、零售业、电信业、金融业、Web挖掘、文本挖掘等诸多方面。在科学上，美国加州理工学院喷气推进实验室的Kayyad研究开发了用于大规模天文测量数据分析的SKICAT系统[61]，已经帮助

天文学家发现了 16 颗新的极遥远的类星体。在生物医学上，用数据挖掘处理 DNA 数据，在癌症治疗、大规模序列模式和基因功能的发现等方面取得了众多突破。而在金融业的应用主要体现在数据仓库和 OLAP 服务、客户信用评定与贷款偿还预测、目标客户的分类和聚类、金融犯罪探测等方面，在零售业的应用主要体现在销售、产品和顾客的多维分析、促销活动效果分析、顾客忠诚度分析、购物篮设计等方面，在电信业的应用主要体现在电信数据的多维分析、盗用模式分析和异常模式识别、序列关联规则分析等方面，等等。数据挖掘所取得的成效举不胜举。

随着数据挖掘理论研究的逐步成熟，数据挖掘产品也应运而生。目前，世界上比较有影响的典型数据挖掘产品有 SAS 公司的 Enterprise Miner, IBM 公司的 Intelligent Miner、SGI(Silicon Graphics Inc.)公司的 Mineset、，加拿大 SimonFraser 大学的 DBMiner、SPSS 公司的 Clementine、,SYBASE 公司的 Warehouse Studio, RuleQuest Research 公司的 Sees, IBM 公司 Almaden 研究中心的 Quest，还有 verstory, Exploar, Knowledge Discovery Workbenrch 等。

国内数据挖掘研究开始于 90 年代中期，到 90 年代中后期，初步形成了知识发现和数据挖掘的的基本框架。研究重点从发现方法转向系统应用，并且注重多种发现策略和技术的集成，以及多种学科之间的相互渗透。与国外相比，国内对 DMKD 的研究稍晚，没有形成整体力量[23-24]，进行的大多数研究项目是由政府资助进行的。国内从事数据挖掘研究的人员主要在大学，也有部分在研究所或公司，所涉及的研究领域很多，一般集中于学习算法的研究、数据挖掘的实际应用以及有关数据挖掘理论方而的研究。其中，华中理工大学、复旦大学、浙江大学、中国科技大学、中科院数学研究所、吉林大学等单位较好地开展了对关联规则开采算法的优化和改造。南京大学的徐洁磐等人开发了一个数据挖掘原型系统 Knight 作为挖掘工具。我国的李德毅院士、施伯乐教授等在数据挖掘领域也取得了显著的成果。目前，国内对非结构化数据包括文本数据、图形图象数据、多媒体数据的知识发现和 Web 数据挖掘做了较深的研究。在时

序数据的挖掘及可视化数据挖掘方面也取得了一定进展。在应用方面也较为广泛。以在工业的应用宝钢为例，它多年来坚持计算机化管理，积累了大量数据，为解决配矿问题，宝钢采用了数据挖掘系统，应用 SAS 全套的数据挖掘和数据分析软件产品，取得了较好的成果。

2.2 关联规则挖掘

2.2.1 关联规则挖掘的概念

关联规则挖掘(Association Rule Mining)是数据挖掘研究中的一个重要分支，关联规则是数据挖掘的众多知识类型中最为典型的一种。该问题是 Agrawal 等在 1993 年在对市场购物蓝问题(Market Basket Analysis)进行分析后首次提出的，用以发现商品销售中的顾客购买模式。购物蓝问题源于这样一个普通的例子:美国加州某个超级连锁店对记录着每天销售信息和顾客基本情况的数据库中的数据进行分析，发现在下班后前来购买婴儿尿布的顾客多数是男性，而且往往也同时购买啤酒。于是这个连锁店的经理当机立断，重新布置货架，把啤酒类商品布置在婴儿尿布货架附近，并在二者之间放上土豆之类的佐酒小食品，同时把男士们的日常生活用品也就近布置。这样一来，上述几种商品的销量大大增加了。

关联规则挖掘可以发现交易数据库中项目(Items)或属性(Attributes)之间的有趣联系，这些联系是预先未知的，不能通过数据库的逻辑操作(如表的联接)或统计的方法得出。这说明它们不是基于数据自身的固有属性(如函数依赖关系)，而是基于数据项目的同时出现的特征。关联规则的特点是形式简洁、易于解释和理解，并可以有效地捕捉数据间的重要关系。最为典型的例子是“在购买面包的顾客中有 80%也购买了黄油”。大型商场和超市的数据库中保存了大量的顾客的购买信息，从中发掘黄油——面包这类有趣的关联关系，可以指导商家制定正确的销售决策，又如通过交叉购物、贱卖分析、目录设计、商品陈列等，使他们在市场竞争中取得更大的主动权。其实，关联规则的应用不仅仅

局限于市场菜篮分析，它有着广泛的应用领域，如商业与金融、人口普查数据分析、工程技术数据分析、医疗[25-26]、财政[27]、宏观决策支持、电子商务、CRM[28]、网站设计[29]、互联网[30]等等。理论上讲，关联规则挖掘是指从一个大型的数据集(Data set)中发现有趣的关联(Association)或相关(Correlation)关系，即从数据集中识别出频繁出现的属性值集(Sets of Attrib-Value)，也称为频繁项集(Frequent Itemsets，简称频繁集)，然后再利用这些频繁集创建描述关联关系的规则的过程。

2.2.2 关联规则形式和分类

关联规则的形式为X =>Y，其中X称为规则的前项项集(Antecedent Itemsets，简称前项)，Y称为后项项集(Consequent Itemsets，简称后项)。它说明数据库中的某一条记录如果包含了X，那么也倾向于包含Y。或者说，如果数据库中的某条记录使X中的属性值为真，那么也倾向于使Y中的属性值为真。下面用实例进一步说明关联规则。

例 :contains(T ，“面包”) =>contains(T ，“黄油”) [support=2% ,confidence=50%]

在这里，T是表示事务记录(Transaction Record)的变量。该规则表明，如果事务T中包含“面包”，则它同时包含“黄油”的可能性为50%，并且所有事务中有2%包含了两者。

关联规则挖掘就是从事务数据库中找出上述形式的规则。

根据不同的情况，关联规则有多种分类方法：

1.根据规则中所处理的值的类型划分

a 布尔关联规则(Boolean association rule)

布尔关联规则处理的值都是离散的、种类化的，规则表达的是项的存在与不存在。例如：面包=>黄油 [support =3%, confident=60%]

b 量化关联规则(Quantitative association rule)

关联规则描述的是量化的项或属性之间的关联。规则中的项或属性的量化值被划分为不同区间。例如:

age(x,"25..45")∧income(x,"1500..6000")=>buys(x,"TV")

其中，x 是代表顾客的变量，量化属性 age 和 income 已离散化。

2.根据规则中的数据维数划分

a 单维关联规则(single-dimensional association rule)

从事务数据库中挖掘出的规则通常只涉及一个维或一个属性。如:规则面包=>黄油可写作 buys(x, "面包")=>buys(x, "黄油")。该规则中只有一个维 buys.。

b 多维关联规则(multidimensional association rule)

前面例子中，age, income 和 buys 是三个不同的维(属性)，其相应的规则就是多维关联规则。多维关联规则中可以多次出现同一个维。

3.根据规则所涉及的抽象层划分

a 单层关联规则(single-level association rule)

单层关联规则中，所有变量都没有考虑现实数据具有多个不同层次。

b 多层关联规则(multilevel association rule)

由于在一些挖掘关联规则的方法引入了概念分层，这样就可以在不同的抽象层得到关联规则。例如，一个关联规则集包含下面的规则:

age(x,"30..39")=>buys(x,"Good Computer") [support=2%, confidence=30%]

age(x,"30..39") =>buys(x, "computer") [support=1% , confidence=60%]

其中，"computer"比"Good compute"具有更高的抽象层。那么，这个规则集就是多层关联规则集。对于事务数据库，在其多个概念层的各项之间寻找有趣

的关联规则比仅在原始层数据之间寻找更容易，并且在较高概念层发现的关联规则更具普遍意义。

2.2.3 关联规则算法综述及研究方向

由于挖掘算法在数据挖掘过程中起着至关重要的作用，因此自从 Agrawal 等提出挖掘交易据库中项集间的关联规则问题以后，很多人对此进行了研究，这些研究包括:关联规则的挖掘理论的探索、原有算法的改进和新算法的设计、并行关联规则挖掘(Parallel Association Rule Mining)以及量化关联规则挖掘(QuantitiveAssociation Rule Mining)等。

关联规则数据挖掘首先由 Agrawal, Imiehski 和 Swami[6]提出，著名的 Apriori 算法由 Agramal 和 srikant[7]提出。很多又在 Apriori 算法基础上改进，用以提高效率和伸缩性。如利用采样(Samplingof Database)的方法[57][58]对数据库进行挖掘，可大大减少对数据库的扫描次数，提高计算效率，这样做的后果可能产生遗漏的频繁项，如何找回部分遗漏的频繁项目集也是一个需要解决的问题，因此，对随机采样的方法进行了进一步的讨论，给出了利用二项分布(Binomial Distribution)和契尔诺夫边界(Chernoff Bounds)[58]处理采样的方法来解决上述问题。有人提出了一种无需生成候选集的频繁集生成算法—频繁模式增长(FP-growth)方法[59]，它采用新颖的数据结构和分治策略，无需产生候选项集，从而大大降低了搜索开销，比 Apriori 算法速度提高一个数量级。还有讨论基于垂直数据库结构的关联规则挖掘方法的文章[60]，这种方法不受数据库的大小、形状、内容等限制，可以有效地发掘最大频繁项集，对于挖掘低支持度和长模式的关联规则特别有效。类似的剪枝方法的算法变形由 Mannila、Toivonen 和 Verkamo[8]研究。其他一些新的技术，如，散列技术即 Hash 技术被 Park、 Chen 和 Yu[9]研究，通过事务压缩技术减少数据的访问的方法被 Agramal 和 srikant[10]，Han 和 Fu[11]，以及 Park、Chen 和 Yu 研究，划分技术被 Savasere、Omiecinski 和 Navathe[12]提出，其主要思想是把数据库分为几个相互独立的块，再采取分而治之的策略。还有基于临时生成项集的动态项集技术被 Brig、Motwani,Ullman

和 Tsur[14]提出等。另外许多新的方法被提出以扩充关联规则挖掘，包括 Agramal 和 srikant 的序列模式挖掘[14], Zaki、Lesh 和 Ogihara 的对 plan failure 的序列模式挖掘[15], Guha、Rastogi 和 Shim 的基于约束的序列模式挖掘[16]， Mannila、Toivonen 和 Verkamo 的 episodes 挖掘[17]，Koperski 和 Han 的空间关联规则挖掘[18]，Ozden、Ramaswamy 和 Silberschatz 的有环关联规则挖掘[19]，Savasere、Omiecinski 和 Navathe 的否定关联规则挖掘[20]，Lu、Han 和 Feng 的事务间关联规则挖掘[21], Ramaswamy、Mahajan 和 Silberschatz 的日历购物篮分析[22], Bayard 的最大模式挖掘等等。

目前，关联规则挖掘方面的研究已经取得了较大的进展，但对下列问题仍有待于进一步研究，如挖掘算法高效性，挖掘对象的广泛性，挖掘的可视化问题，模式评估等。

2.3 中国股市简要分析

2.3.1 中国股市简介

自 1990 年沪深证交所成立，发展至今已趋于成熟。中国的各行各业均有上市公司。目前交易品种有股票，国债，债券等。其中 A 股市场有近 1700 只股票可供中国绝大多数投资者投资。中国股票的总市值已达几万亿。股市被誉为经济的晴雨表，可见其对中国经济的重要性。适逢 2005 年下半年，中国股市又经历了重大改革——非流通股转变为流通股即国有股减持政策的推出，从而为中国股市带来了良好的发展契机。

专家认为，影响股票涨跌的主要因素是市场内部原因——股票的供求情况，外部因素只起辅助作用。那么供求情况又是怎么样表现出来的呢?谁为供，谁为求呢？市场将供方比喻成空方，将求方比喻成多方。多方占优势时，股票就上涨；反之，就下跌。其实，供求情况只是一个表象，供求的背后—— 资金才是市场的关键。没有资金的支持，股市形同虚设，更谈不上供求关系。市场上的资金主要分为两部分：第一部分是少部分人控制的大资金即庄家，他们通过大资金买入某股票的大量股份，成为该股票的幕后操纵者，他们的操作都是进行

过周密部署的。第二部分是大部分人控制的小资金即散户，他们资金量少而且分散不足以控制某只股票，而且思维不统一，跟风气氛浓厚，容易被庄家利用。庄家和散户是矛盾的，他们共处股市却暗中较量，这两个矛盾体构成了股市。

目前，投资者主要关心的问题是（1）如何买入一只即将上涨的或涨幅不是很高以后还有上涨空间的股票。（2）一只股票获利后如何在保持住最大胜利果实的同时将股票卖掉。

2.3.2 目前分析和研究股市的方法

股票价格指数和平均数仅仅为人们提供了一种衡量股票价格变动历史的工具，然而，人们更关心的是如何预测股票价格的未来趋势，以及买卖股票的适当时机。多少年来，人们不断地对股价走势进行研究，产生了种种方法。现在大多数人采用基本面分析法和技术分析法预测股市的走势。

基本面分析法主要是分析股票自身的特点及经济，政治对股票的影响。股票自身的特点主要是指股票上市公司所属行业的性质，公司经营业绩，财务指标等，它们属于相对稳定的数据。中国证监会要求上市公司每季度公布一次（季度报），对半年报和年报要求公布的数据更为详细。除了股票自身的特点外，经济，政治对其也有较深的影响。国际形势的变化，战争的爆发，国内重大政治事件的出现都会给股市带来突如其来的影响，投资者往往还没有反应过来股市就开始了暴涨或暴跌；国家的重大经济政策，如产业政策、税收政策、货币政策，则会给相关行业的股票带来较为深远的影响。

所谓股价的技术分析，是相对于基本分析而言的。正如上一部分所述，基本分析法着重于对一般经济情况以及各个公司的经营管理状况、行业动态等因素进行分析，以此来研究股票的价值，衡量股价的高低。而技术分析则是透过图表或技术指标的记录，研究市场过去及现在的行为反应，以推测未来价格的变动趋势。其依据的技术指标的主要内容是由股价、成交量或涨跌指数等数据计算而得的，我们也由此可知—技术分析只关心证券市场本身的变化，而不考虑会对其产生某种影响的经济方面、政治方面等各种外部的因素。

基本分析的目的是为了判断股票现行股价的价位是否合理并描绘出它长远的发展空间，而技术分析主要是预测短期内股价涨跌的趋势。通过基本分析我们可以了解应购买何种股票，而技术分析则让我们把握具体购买的时机。在时间上，技术分析法注重短期分析，在预测旧趋势结束和新趋势开始方面优于基本分析法，但在预测较长期趋势方面则不如后者。大多数成功的股票投资者都是把两种分析方法结合起来加以运用。他们用基本分析法估计较长期趋势，而用技术分析法判断短期走势和确定买卖的时机。

股价技术分析和基本分析都认为股价是由供求关系所决定。基本分析主要是根据对影响供需关系种种因素的分析来预测股价走势，而技术分析则是根据股价本身的变化来预测股价走势。技术分析的基本观点是:所有股票的实际供需量及其背后起引导作用的种种因素，包括股票市场上每个人对未来的希望、担心、恐惧等等，都集中反映在股票的价格和交易量上。

技术面分析作为证券投资分析的工具之一，与基本面分析结合在一起，构成了证券投资分析的左右手，因此，在证券投资分析中不能片面使用技术面分析。事实上，在中国的证券市场上，技术面分析依然有较高的预测成功率。这里，成功的关键在于不能机械地使用技术面分析。要全面考虑技术分析的各种方法对未来的预测，综合这些方法得到的结果，最终得出一个合理的多空双方力量对比的描述。实践证明，单独使用一种技术分析方法具有相当的局限性和盲目性。如果每种方法得到同一结论，那么这一结论出错的可能性就很小，这是已经被实践证明了的真理。如果仅靠一种方法得出的结论，出错的概率就大得多了。为了减少自己的投资失误，应尽量多掌握一些技术分析方法，而且在分析市场趋势时尽量用各种技术分析方法都去分析一下，掌握地越多，分析地越全面，对投资都是有好处的。除了在实践中不断修正技术面分析外，还必须结合基本面分析来使用技术面分析。

基本面分析与技术面分析各有千秋，各有所长。技术面分析的理论基础是统计学和数量经济学，而基本面分析法的理论基础是货币经济理论。这种理论把经济理论与货币银行学理论揉合在一起，是当今经济货币化，金融证券化趋

势的反映。尽管两种分析法各有特色，各有专业的研究人员和信奉者，我们应该结合基本面分析与技术面分析的观点和操作策略以投身于股市实战，从而收到最大限度的积极效果。

2.3.3 相关软件的介绍及评价

我国股票市场使用的分析软件的代表是钱龙，这是在上海和深圳市场刚刚开始的时候从台湾引进的。此后的几年，市场上相继出现了多种其他种类的分析软件。目前市面上常用的估计有20多种。因为钱龙先入为主，所以占据了相当大的市场。就分析的功能看，钱龙差不多“划定了框框”。此后的分析软件的功能基本上是在其基础上进行较小改进的产物。

当前的分析软件提供的功能大致有如下几个方面[31-34]：实时接收数据，智能盯盘报警与备忘录(对每只股票建立信息库)，综合绘线功能，投资组合分析，主力进出和主力动向，区域统计—可统计任意时间段内股票的涨跌幅，成交量，成交金额，换手率等，强势股跟踪与投资实习，自由组织技术指标、条件选股公式，智能选股，动态选股——技术指标与财务指标选股，抓住实时量价计算，第一时间发现机会，人工智能自动投资组合功能，资金管理—对投资者的入市资金进行风险分离，每次可按比例进行投资。

这些功能各有千秋，但又各有局限性，主要是不能把炒股专家实战的经验加进去，而且对证券基本面的分析都很浅显;对于技术面分析智能性不够，简单的指标罗列使一些不懂这些指标的投资者望而却步。有一些股票分析软件名义上提供专家系统、数据挖掘等分析项目，但实际上只是对股票数据进行简单的统计分析，没有深入发掘各支行情记录本身所蕴藏的信息，智能化水平较低。现有的分析软件还有一个很重要的不足就是没有验证功能。假设我们用现在的分析软件进行选股，就会遇到一个很大的问题。选股条件是人们根据自己的要求输入的，这些输入的条件是否合理还是个问题。由此可知，根据这些条件选出的股票究竟有多大的可用成份。无论是自己总结的方法，还是从别人那里学习来的方法，都必须经过实践的检验才能使用。如果我们总结得到了一套投资

的方法，我们当然要关心它的实战效果。我们希望知道，如果严格按照这种方法可能给我们带来什么样的后果。从纯理论的观点讲，任何方法都不可能是永远有效，都应该随时间和环境的改变而进行相应的修正。软件的“验证功能”将提供根据环境改变对策的依据。如果发现原来的计算方法不灵了，就要即时调整对策以适应新的环境。

让人满意的智能型软件在我国还没有出现(或者出现了没有公开)。我国股票市场起步晚，投资者的分析“水平”参差不齐，不理智的因素较多。随着股票市场的发展和时间的推移，为数众多的不合格投资者将被市场淘汰，留下来参与投资的人将越来越理智。投资效果的好坏越来越依靠“真正的分析”。越来越多的投资者将会更多地借助计算机进行分析，而目前分析软件的功能明显不能适应今后投资分析人员的需要，今后要求分析软件做的事情肯定不会像现存这么简单。因此，我国股票分析软件的发展方向就是可验证与修正的智能型软件。

在这种情况下引入代表数据库和人工智能最新技术的“真正”的数据挖掘技术进行股票分析与预测成为一个必然的选择。

第 3 章 关联规则的基本算法及相关改进

3.1 关联规则的基本概念

关联规则的挖掘是对给定的一个交易数据库 D，求出所有满足最小支持度和最小置信度的关联规则的过程。该问题可分解为两个子问题:(1)根据给定的最小支持度，按项目数自小而大的顺序找出数据库 D 中频繁项目集;(2)根据频繁项目集和指定的最小置信度生成关联规则。

设有 $I=\{I_1,I_2,...I_m\}$是由 m 个不同的项组成的集合。给定一个事务数据库 D，其中每一个事务 T 是 I 中一组项的集合，即 $T\subseteq I$, T 有一个唯一的标识符 TID。若项集 $A\subseteq I$ 且 $A\subseteq T$，则称事务 T 包含项集 A。如果项集 A 中包含 K 个项，则称为 K-项集。

定义 3.1 关联规则是形如 A=>B 的蕴涵式，其中 $A\subset I$, $B\subset I$, $A\cap B=\varnothing$。关联规则 A=>B 在事务数据库 D 中成立，具有支持度 *s*，其中 *s* 是 D 中事务包含 AUB 的百分比，记作:support(A=>B) =P(AUB)。

通常用户指定最小支持度,记为 minsupport。

定义 3.2 关联规则 A=>B 在事务数据库 D 中的置信度是 D 中包含 A 的事务同时也包含 B 的百分比，它是条件概率 P(B|A)，记作:confidence(A=>B) = P(B|A)。

通常用户指定最小置信度,记为 minconfidence。

定义 3.3 若 support(A=>B)≥minsupport,且 confidence(A=>B)≥minconfidence,则称关联规则 A=>B 为强关联规则。

关联规则挖掘的任务就是在数据库中挖掘出所有强关联规则。即在事务数据库中找出所有具有用户给定的最小支持度 min_sup 和最小置信度 min_conf 的关联规则。这样，每一条被挖掘出来的关联规则就可以用一个蕴含式，两个阈值唯一标识。

置信度是对关联规则正确程度的衡量，表示规则的强度;支持度是对关联规则重要性的衡量，表示规则的频度。规则的支持度说明它在所有事务中有多大的代表性，其值越大，关联规则越重要。如果关联规则的置信度很高，但支持

度很低，说明该关联规则实用机会很小;如果支持度很高，而置信度很低，则说明该规则不可靠。

定义 3.4 如果一个项目集 A 满足最小支持度阈值 min_sup，即 support(A)≥min_sup，则称它为频繁项集(frequent itemset)。频繁 k-项集通常记为 L_K。反之，如果一个项目集 A 不满足最小支持度，则称为非频繁项集。

定义3.5候选项集是潜在的频繁项集的集合，是频繁K-1项集的超集(superset)，含有K项的候选项集表示为C_k，由它构成频繁k-项集L_K。

3.2 经典 Apriori 算法分析

单维、单层、布尔关联规则挖掘是最简单形式的关联规则挖掘，最著名、最有影响的关联规则挖掘算法是 R. Agrawal 等人提出的 Apriori 算法[35]，该算法是一种挖掘布尔关联规则频繁项集的算法。它利用频繁项集性质的先验知识，使用一种称为逐层搜索的迭代方法来找出所有的频繁项集。首先扫描事务数据库 D，统计库中事务的数量和各个不同的项(1-项集)所出现的次数，进而根据最小支持度 min_sup 获得所有的频繁 1-项集 L_1。然后用 L_1 查找频繁 2-项集 L_2，如此下去，直到不能找到频繁 k-项集。找每个 L_k 需要一次数据库扫描。

Apriori 算法的具体描述如下:

输入:事务数据库 D,最小支持度阈值 min_sup;

输出:D 中的频繁项集 L。

L_1＝find_frequent_1-itemsets (D);　//频繁 1-项集

for (k=2; $L_{k-1} \neq \varnothing$;k++)

　begin

　C_k=apriori_gen (L_{k-1}，min_sup);　//新的候选项集

　for each transaction t∈D　//扫描 D 中项集

　　begin

　　C_t=subset (C_k, t);　//事务 t 中包含的候选项集

　　for each candidate c∈C_t

```
            c.count++;
            end
        Lk={c∈Ck| c.count≥min_sup }
        end
return L=U k Lk;

procedure  find_frequent_1-itemsets(D: transaction  database)
//找频繁 1-项集
    begin
        for  each item ik∈D
        begin
          if  ik .count/ |D|≥min_sup  then
          add  ik to L1;
        end
    end

procedure apriori_gen (Lk-1:frequent (k-1)-item set; min_sup: support)
// apriori_gen 算法
begin
    for each item set I1 ∈Lk-1
        for each item set I2 ∈Lk-1
            if (I1 [1]= I2 [1]) ∧(I1 [2]=I2 [2]) ∧…∧(I1 [k-2]=I2 [k-2]) ∧
                (I1 [k-1]< I2 [k-1]) then
            begin
                    c= I1 ∞I2;  //连接步:产生候选项集集合
                if  has_infrequent_subset(c,Lk-1) then
                        delete c; //剪枝步:删除非频繁候选项集
```

```
                    else add c to Ck;
                end
        return Ck;
end

procedure   has_infrequent_subset(c:   candidate   k-item   set;
                                  Lk-1:frequent (k-1)-item set)
    // 判断候选 k 项集的 k－1 子项是否都在频繁 k－1 项集中
  begin
      for each (k-1)-subsets of c
      if s∉Lk-1 then
          return TRUE;
      return FALSE;
  end
```

该算法中有两个关键步骤:连接步和剪枝步。

(1)连接步:为找出 L_k(频繁 k 一项集),通过 L_{k-1} 与自身连接(L_{k-1} ∞L_{k-1})产生候选 k-项集,该候选项集记作 C_k:其中,L_{k-1} 的元素是可连接的。

(2)剪枝步: C_k 是 L_k 的超集,即它的成员可以是也可以不是频繁的,但所有的频繁 k-项集都包含在 C_k 中。扫描数据库,确定 C_k 中每一个候选的计数,从而确定 L_k (计数值不小于最小支持度计数的所有候选是频繁的,从而属于 L_k)。然而, C_k 可能很大,这样所涉及的计算量就很大。为压缩 C_k,使用 Apriori 性质:任何非频繁的(k-1)-项集都不可能是频繁 k 项集的子集。因此,如果一个候选 k-项集的(k-1)-项集不在 L_{k-1} 中,则该候选项也不可能是频繁的,从而可以由 C_k 中删除。这种子集测试可以使用所有频繁项集的散列树快速完成。

一旦从数据库 D 中的事务中找出频繁集,就可以由它们的产生强关联规则。下面通过一个具体的示例来说明用 Apriori 算法来找寻一个事务数据库中的频繁项集的过程,以及 apriori-gen 函数中的连接和剪枝两个步骤。假设事务数据库

D 如图 3.1 所示，D 中有 9 个事务，假定事务中的项按字母次序存放。

事务标识号(TID)	项 ID 的列表
T_1	A,B,E
T_2	B,D
T_3	B,C
T_4	A,B,D
T_5	A,C
T_6	B,C
T_7	A,C
T_8	A,B,C,E
T_9	A,B,C

图 3.1 一个事务数据库

经过算法的第一次迭代，对事务数据库进行一次扫描，得到候选项集集合 C_1，如图 3.2 所示。

项集	支持度计数
A	0.67
B	0.78
C	0.67
D	0.22
E	0.22

图 3.2 候选项集集合 C_1

假定最小事务支持计数为 2(即 min-sup=2/9=0.22)。通过比较可以得到频繁 1-项集的集合 L_1，如图 3.3 所示。

项集	支持度计数
A	0.67
B	0.78
C	0.67
D	0.22
E	0.22

图 3.3 频繁 1-项集集合 L_1

使用 $L_1 \infty L_1$，产生候选 2-项集集合 C_2，如图 3.4 所示。

项集	支持度计数
A,B	0.44
A,C	0.44
A,D	0.11
A,E	0.22
B,C	0.44
B,D	0.22
B,E	0.22
C,D	0
C,E	0.11
D,E	0.11

图 3.4 候选 2-项集集合 C_2

通过候选项集集合 C_2 的支持度和支持度阈值进行比较，得到频繁 2-项集集合 L_2，如图 3.5 所示。

项集	支持度计数
A,B	0.44
A,C	0.44
A,E	0.22
B,C	0.44
B,D	0.22
B,E	0.22

图 3.5 频繁 2-项集集合 L_2

使用 $L_2 \infty L_2$ 产生候选 3-项集集合 C_3={{A,B,C}, {A,B,E}, {A,C,E}, {B,C,D}, {B,C,E}, {B,D,E}}。根据 Apriori 性质，频繁项集的所有子集必须是频繁的，我们可以确定后 4 个候选项不可能是频繁的。因此，我们把它们从 C_3 删除，这样，在此后扫描 D 确定 L_3 时就不必再求它们的计数值。得到候选项集集合 C_3，如图 3.6 所示。

项集	支持度计数
A,B,C	0.22
A,B,E	0.22

图 3.6 候选 3-项集集合 C_3

通过候选项集集合 C_3 的支持度和支持度阀值进行比较，得到频繁 3-项集集合 L_3，如图 3.7 所示。

项集	支持度计数
A,B,C	0.22
A,B,E	0.22

图 3.7 频繁 3-项集集合 L_3

使用 $L_3 \infty L_3$，与产生候选 4-项集集合 C_4={A,B,C,E}，由于该集合的子集{B,C,E}不是频繁的，所以，这个项集被剪去。这样，$C_4=\varnothing$，因此 Apriori 算法终止，我们就找出了所有的频繁项集。

在找到了事务数据库中的所有频繁项集后，利用这些频繁项集可以产生关联规则，产生关联规则的步骤如下:

(1)对于每个频繁项集 l，产生 l 的所有非空子集。

(2)对于 l 的每个非空子集 s，如果 support-count(l)/support-count(s) ≥ min_conf，则输出规则“s =>(l-s)”。其中，min_conf 是最小置信度阀值。

例如，我们假定频繁项集 l={A,B,E}}，l 的非空子集有{A,B}, {A,E}, {B,E}, {A}, {B}, {E}，则运用上述产生关联规则的方法可以得到以下关联规则。

A∧B =>E, confidence=2/4=0.5

A∧E =>B, confidence = 2/2=1

B∧E =>A, confidence=2/2=1

A =>B∧E, confidence=2/6=0.33

B =>A∧E, confidence=2/7=0.29

E =>A∧B, confidence=2/2=1

如果最小置信度阀值为 0.7，则上述规则中只有以下三条规则是强规则，可以输出。

A∧E =>B, confidence = 2/2=1

B∧E =>A, confidence=2/2=1

E =>A∧B, confidence=2/2=1

Apriori 算法的缺点:(1)由频繁 k-1 项集进行自连接生成的候选频繁 k 项集数量巨大。(2)在验证候选频繁 k 项集的时候需要对整个数据库进行扫描，非常耗时[36]。

3.3 Apriori 改进算法：AprioriTid 和 AprioriHybird 算法的分析

为了避免每次都要扫描数据库，于是产生了 Apriori 的改进算法：AprioriTid 算法。该算法也使用了 Apriori-gen 函数以便在遍历之前确定候选项目集。这个算法的新特点是在第一次遍历之后就不使用数据库 D 来计算支持度，而是用集合 C_k 来完成。集合 C_k 每个成员的形式为（TID，$\{X_k\}$），其中每个 X_k 都是一个潜在的大型 k 项集，在标识符为 TID 的事务中。对于 k=1，C_1 对应于数据库 D，虽然在概念上每个项目 I 由项目集{1}代替。对于 k>1，有算法产生 C_k。与事务 t 相应的 C_k 的成员是（t.TID,{c∈C_k|t 中包含的 c}）。若某个事务不包含任何候选 k 项目集，那么 C_k 对应这个事务就没有条目（Entry）。这样，C_k 中条目数量比数据库中的事务数量少，尤其对于大值的 k 而言。另外，对于大值的 k，每个条目比相应的事务要小，这是因为几乎没有什么候选能包含在此事务中。但是，对于小值的 k，每个条目比相应的事务要大，因为 C_k 中的一个条目包括了此事务中的所有候选 k 项目集。

AprioriTid 算法的具体描述如下:

(1) L_1={大项目集 1 项目集};

(2) C_1=数据库 D;

(3) for(k=2;L_{k-1} ≠∅ ;k++) do begin

(4) C_k=apriori-gen(L_k-1);

(5) $\overline{C}_k$ = ∅;

(6) for 所有条目 t∈$\overline{C}_{k-1}$ do begin

(7) //确定事务 t。TID 中包含的候选

Ct＝{c∈C_k | (c-c[k]) ∈t.项目集的集合∧(c-c[k-1]) ∈t.项目集的集合};

(8) for 所有候选 c∈Ct do

(9) c.count++;

(10) if(Ct≠∅) then $\overline{C}_k$ +=<t.TID,C_t>;

(11) end

(12) $L_k=\{c\in C_k \mid c.count\geq min\_sup\}$

(13) end

(14) 答案= $\bigcup_{k=1}^{m} L_k$;

现举一例来说明，如图 3.8 数据库，假设最小支持度是 2 个事务。在第（4）步对 L_1 调用 apriori-gen，给出候选项目集 C_1。在第（6）步到第（10）步，在 $\subseteq_1$ 中条目上不断重复计算 C_2 中候选的支持度，并产生 $\overline{C}_2$。$\overline{C}_1$ 中第一个条目是{{1}{3}{4}},与事务 100 响应。第（7）步上与这个条目 t 相应的是{{1 3 }},因为{1 3 }是 C_2 的一个成员，（{1 3 }-{1}）和（{1 3 }-{3}）都是 t 项目集的集合的成员。对 L_2 调用 apriori-gen，给出 C_3。遍历 $\overline{C}_2$ 中的数据并利用 C_3 产生 $\overline{C}_3$。注意在 $\overline{C}_3$ 中对于 TID 为 100 和 400 的事务没有条目，因为它们不包括任何 C_3 中的项集。C_3 中的候选{2 3 5}是大的，而且是 L_3 的唯一成员。用 L_3 产生 C_4 时，它是空的，于是结束。

AprioriTid 算法的优点：仅在计算频繁 1 项集对数据库进行一次扫描，以后对频繁 k 项集的计算都是用上次生成的 C_{k-1} 来计算项集的支持度，随着 k 的增加，C_{k-1} 的大小越来越小于原始数据库，减少了 I/O 操作时间和需要扫描的数据库的大小。ApririTid 算法的缺点：$\overline{C}_1$ 和 $\overline{C}_2$ 数据量庞大，在求频繁 2 项集 L_2 和频繁 3 项集 L_3 的时候非常耗时。

由于 AprioriTid 算法在生成 $\overline{C}_1$，$\overline{C}_2$，…，$\overline{C}_n$ 时，产生大量的项目集合，特别是 1-项目集合和 2-项目集合，它们占据内存空间巨大，影响计算速度，所以又产生一种混合算法 AprioriHybird，它的想法是：在初始的遍历中使用 Apriori 算法，当希望在遍历末尾处集合 C_k 能适合内存时就转换到 AprioriTid 算法。用某种方法来估计下一次遍历中 C_k 是否适合内存。这种方法可以是这样的：在当前遍历的末尾，可以得到 C_k 中候选的计数（Counts）。从这点估计如果产生了 C_k，那么它的大小尺寸是（$\sum_{候选\ c\in Ck}$ 支持度（C）＋事务的数量）。若这次遍历中

数据库

TID	项目
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$\overline{C}_1$

TID	项目的集合
100	{{1},{3},{4}}
200	{{2},{3},{5}}
300	{{1},{2},{3},{5}}
400	{{2},{5}}

L_1

项集	支持度
{1}	2
{2}	3
{3}	3
{5}	3

C_2

项集	支持度
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$\overline{C}_2$

TID	项目的集合
100	{{1 3}}
200	{{{2 3}{2 5}},{3 5}}
300	{{{1 2}{1 3}{3 5}},
	{{2 3}{2 5}},{3 5}}
400	{{2 5}}

L_2

项集	支持度
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	3

C_3

项集	支持度
{2 3 5}	2

$\overline{C}_3$

TID	项目的集合
200	{{2 3 5}}
300	{{2 3 5}}

L_3

项集	支持度
{2 3 5}	2

图 3.8 实例数据库

的 C_k 小到可以适合内存，而且在当前遍历中的大型候选比前次遍历要少，那么就转换成 AprioriTid。

通过性能测试，几乎在所有的情况下，算法 AprioriHybird 比 Apriori 和 AprioriTid 要完成的更好。当转换出现的遍历是最后一次遍历时，AprioriHybird 比 Apriori 要差点；AprioriHybird 要承担转换成本，而没有收益。AprioriHybird 比 Apriori 好 30%，比 AprioriTid 好 60%。但 AprioriHybird 的实现比 Apriori 更复杂。

3.4 其他几种改进算法简介

1.基于哈希(hash)表技术：利用 hash 表技术[55]可以帮助有效减少候选 k-项集 C_k （k>1）所占用的空间。例如:在扫描交易数据库以便从候选 1-项集 C:中产生频繁 1-项集 L_1 时，就可以为每个交易记录产生所有的 2-项集并将它们哈希(hash)到 hash 表的不同栏目中，且增加相应栏目的技术。如果 hash 表中一个存放 2-项集的栏目技术低于最小支持频度，则表示相应 2-项集为非频繁项集而被移出候选集。利用这样 hash 表技术可以帮助有效减少需要检查的候选项集。

2.基于划分的方法：Savasere 等[37]设计了一个基于划分(partition)的算法，这个算法先把数据库从逻辑上分成几个互不相交的块，每次单独考虑一个分块并对它生成所有的频集，然后把产生的频集合并，用来生成所有可能的频集，最后计算这些项集的支持度。这里分块的大小选择要使得每个分块可以被放入主存，每个阶段只需被扫描一次。而算法的正确性是由每一个可能的频集至少在某一个分块中是频集保证的。上面所讨论的算法是可以高度并行的，可以把每一分块分别分配给某一个处理器生成频集。产生频集的每一个循环结束后，处理器之间进行通信来产生全局的候选 k-项集。通常这里的通信过程是算法执行时间的主要瓶颈;而另一方面，每个独立的处理器生成频集的时间也是一个瓶颈。其他的方法还有在多处理器之间共享一个杂凑树来产生频集。更多的关于生成频集的并行化方法可以在中找到。

3.采样技术[13]：所谓采样技术就是对给定数据集的一个 r 集进行挖掘。采样方法的核心是随机从数据集 D 中采集 S'样本集，然后搜索 S'中频繁项集而不是 D 中的。这样就以效率换取准确性。因此有可能漏掉一些全局频繁项集。为减少这种可能性，这里利用了一个比最小支持度阈值要小的支持度阈值来挖掘局部频繁项集。采样方法在对效率要求较高的应用场合是极具意义和重要的:尤其是在需要频繁进行这种密集计算的应用场合。

4.基于动态项集计数的技术[14]：动态项集计数技术将数据库划分为标记开始点的块，在扫描的不同点添加候选项集。不象 Apriori 仅在每次完整的数据库扫描之前确定新的候选，在这种方法中，可以在任何开始点添加新的候选项集。该技术动态地评估已被计数的所有项集的支持度，如果一个项集的所有子集已被确定为频繁的，则将它添加为新的侯选。结果算法需要的数据库扫描比 Apriori 少。

5.基于事务压缩[54]：该技术减少用于未来扫描的事务集的大小。一个基本的原理就是不包含任何 k-项集的事务不可能包含任何(k+1)-项集，从而就可以将这些事务在其后的考虑时加上标记或删除，因为 j-项集(j>k)的产生，扫描数据库时不再需要它们，这样在下一遍扫描时就可以减少要进行扫描的事务集的个数。

3.5 对 AprioriHybird 算法的新改进

3.5.1 改进思路

AprioriHybird 算法前半段使用的是 Apriori 算法，后半段使用的是 AprioriTid 算法。整个算法的花费时间主要是在前半段，而前半段中以生成频繁 2 项集和频繁 3 项集所费时间最多，占据绝大部分[56]，如图 3.9 所示。同时多次扫描数据库也占去了不少运行时间。于是可以用减少扫描数据库次数和减少求解频繁 2 项集和频繁 3 项集的时间的方法来改进算法。另外在求解频繁 2 项集和频繁 3

项集时不断压缩数据库，从而在转为 AprioriTid 算法时，可以少生成$\overline{C}_k$中的项，减少了查询比较时间。

我们用一个上三角矩阵来直接记录候选频繁 2 项集 C_2 的所有元素。该方法不用生成 L_1。例如有 A, B, C，D ，4 个项 ，则 C_2 可用如图 3.10 表示。这个矩阵称为：2 项集支持度矩阵，给 A, B, C, D 分别赋予顺序号 0，1，2，3，则使用下标(i，j)就可以访问到任何一个候选频繁 2 项集，比如(A, B)使用(0，1)就可以访问到。

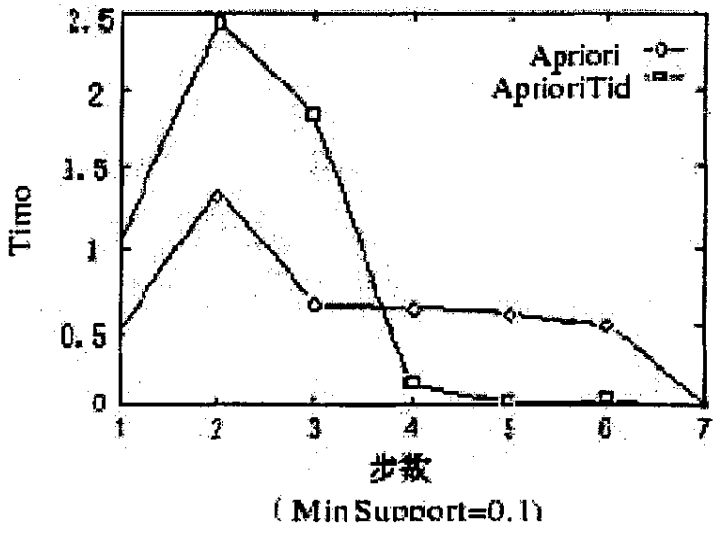

图 3.9 Apriori 和 AprioriTid 的每步执行时间

扫描一边数据库用以确定 C_2 中各项的支持度。先对矩阵清零，对于每一条交易，把交易项目排序，生成所有可能的顺序 2 项组合，访问矩阵中相应的元素，把该元素的计数加 1，就完成了对一条交易的处理。比如第一条交易项目为{A，B，D}，则所有的顺序 2 项组合为{A, B}、{A, D}、{B, D}，它们对应的矩阵元素为{0, 1}, {0, 3}, {1, 3}，把这些元素的计数加 1，就完成了对第一条交易项目的处理。一直扫描完数据库的所有交易，然后用矩阵中各元素值除以数据库的总条目数就得到了矩阵各元素的支持度,再分别写回原处。另外设置一个数组 Z 记录每条事务中的项目个数。

	A	B	C	D
A		AB	AC	AD
B			BC	BD
C				CD
D				

图 3.10 候选频繁 2 项集

上述方法可获得候选频繁 2 项集的支持度，下述方法可直接获得频繁 3 项集及其支持度，不用生成候选频繁 3 项集。首先查找矩阵第一行，找出所有支持度大于 min_sup 的项，然后用第一项与第二项组合生成 3 项集，同时记录下第一项与第二项分别所在的列号，再查找由这两个列号组成的矩阵元素的值，若大于 min_sup，则用第一项与第二项组合生成的 3 项集就是频繁 3 项集，其支持度为第一项，第二项，两个列号组成的矩阵元素中支持度的最小值。例如：AB 的支持度为 2，大于 min_sup，其所在列号为 B，AC 的支持度为 2，也大于 min_sup，其所在列号为 C，则查看由 B，C 组成的矩阵元素值，例如为 3，则项 ABC 是频繁项，支持度为三者中最小的，为 2。这样就完成了一个项的组合。依次把矩阵第一行中所有支持度大于 min_sup 的项组合起来，按上述方法计算支持度。第一行处理完成后，再以同样的方法处理第二行，等等，直到处理完所有行。

现在考虑压缩数据库：(1)对于 2 项集支持度矩阵，若某行中支持度没有大于 min_sup 的项，则删去该行或给该行打上标记，也就是一条事务。因为由该行任意两项组成的 3 项集的支持度不可能大于 min_sup。(2)按照如下性质：一条事务若包含 k 频繁项集，则它本身至少要包含 k 个项目。这样我们在生成 k(>3) 候选频繁项集时，只要查看数组 A 对应值大于 k 的事务，同时给小于 k 的事务打上标记，下次不予查找，从而大大减轻了查找数据库的工作量。

3.5.2 改进算法描述

(1) 定义 2 项集支持度矩阵，数组 Z，初始化它们。

(2) 扫描数据库每条事务，分别生成 2 项集。将相应信息写入矩阵和数组中。

(3) 由 2 项集支持度矩阵生成 2 项频繁项集和 3 项频繁项集，并计算支持度。

(4) 转入 AprioriTid 算法，计算 k(>3) 项频繁项集。

3.5.3 改进算法的挖掘示例

现举一例来说明新算法的处理过程，如图 3.11 所示数据库。min_sup＝2

TID	项目
100	A C D E F
200	B C E
300	A B C D E
400	B E

图 3.11 数据库

	A	B	C	D	E	F
A		1	2	2	2	1
B			2	1	3	0
C				2	3	1
D					2	1
E						1
F						

图 3.12 2 项集支持度矩阵

(1) 先生成 2 项集支持度矩阵，上三角矩阵元素置 0。再逐行扫描数据库的条目来确定各项集的支持度。例如首先把 TID＝100 的事务拆成 2 项集{{ A C },{ A D },{ A E },{ A F },{C D},{C E},{C F},{D E},{D F},{E F}},在矩阵对应位置加 1，再按同样方法处理下面的事务。如图 3.12 所示。同时用数组 Z 记录每条交易中的项目个数和。例如 TID＝100 的事务中有 5 个项目，于是 Z[0]＝5。

(2)查看矩阵中值≥2 的元素，可以得到 2 项频繁项集，如图 3.13 所示。

(3) 生成 3 项频繁项集的过程。首先在矩阵第一行中按顺序找出两个元素值大于 min_sup 的元素，刚好找到 AC 和 AD 对应的项，于是再找到由这两项列

号组成的元素 CD 的值，等于 2≥min_sup，所以由 AC 和 AD 组成的项 ACD 是频繁的，其支持度为三者中支持度最小者 2。继续查找第一行支持度≥min_sup 的元素。找到 AE，再由 AC 和 AE 组合生成 ACE，再查 CE 项，满足条件，于是 ACE 是频繁的，支持度为 2。再向后查找，没有满足条件的。组合 AD 和 AE 生成 ADE，查找 DE 项，满足条件，于是 ADE 也是频繁的，支持度为 2。 第一行扫描结束。以同样方法扫描第二行，等等，直到扫描完数据库。生成 3 项频繁项集。如图 3.14 所示。

(4) 转入 AprioriTid 算法，由 3 项频繁项自连接生成 4 项候选项集{A C D E}，通过剪枝判断后，决定把它留下。开始计算它的支持度。在数组 Z 中只查找元素值≥4 的数据库条目，于是只考察 TID＝100 和 TID＝300 的两条事务，由这两条事务生成$\overline{C}_4$，找到每条事务中各有 1 项与{A C D E}完全相同，于是生成 4 项频繁项{A C D E},支持度为 2。如图 3.15 所示。

(5) 4 项频繁项集中只有一项，不能生成 5 项候选项集。算法结束。

项集	支持度
{A C}	2
{A D}	2
{A E}	2
{B C}	2
{B E}	3
{C D}	2
{C E}	3
{D E}	2

图 3.13 2 项频繁项集

项集	支持度
{A C D}	2
{A C E}	2
{A D E}	2
{B C E}	2
{C D E}	2

图 3.14 3 项频繁项集

项集	支持
{A C D E}	2

图 3.15 4 项频繁项集

3.6 改进 AprioriHybird 算法的特点及与 AprioriHybird 的比较

改进的 AprioriHybird 算法的前半部分改进中，通过以空间换时间的方法，减少了 Apriori 部分数据库的扫描次数，同时所需要的辅助空间也是有限的。采用矩阵形式有以下好处：(1)占用内存空间少。在传统的改进方法——Hash 树方法中，每一个桶占用的空间是候选项集结构的 N 倍，N 为桶中的候选项集的个数，而且存在没有包含候选项集的桶；矩阵方法只需要候选项集的个数 int (或 long)类型的长度空间。(2)快速计算支持度。Hash 树方法中，需要把每一个交易使用一定的 Hash 函数，分配到 Hash 树上，这是个复杂的递归过程；而在矩阵方法中，只需要组合成所有可能的 2 项集，把相应的矩阵元素加 1 就可以了。(3)易于理解，方便实现。矩阵是个很简单明了的存储 C_2 的方法，编程实现只需要二维数组，相对于 Hash 树方法，更易于理解和实现。

另外，采取的事务压缩方法，对以后数据库的扫描也起到了减少候选项的作用；在后半部分改进中，由于前边的事务压缩，从而在 AprioriTid 部分减少了不必要候选项的产生，另外，通过检查支持度的方法，也减少了不必要候选项的产生。

之前，已做了 AprioriHybird 与 Apriori 的比较，现仅对改进的 AprioriHybird 算法与 AprioriHybird 算法进行比较。

(1)在 Apriori 部分的改进：改进算法仅扫描一遍数据库，无需生成候选频繁 1 项集 C_1，频繁 1 项集 L_1，候选频繁 3 项集 C_3，通过一个矩阵直接生成频繁 3 项集 L_3，大大减少了运行时间。而原 AprioriHybird 算法的 Apriori 部分需扫描 3 次数据库，每次需生成 C_k， L_k。换取时间大幅下减的代价是生成一个数据库所有项目个数之和*数据库所有项目个数之和的上三角矩阵，由于所有项目个数之和（一般少于 30 个）远小于数据库的条目数（一般大于 1 万条），所以新算法在占用空间方面内存完全可以接受；新算法的主要时间花费在对每一条事务的详细处理上(时间复杂度接近 AprioriTid 生成 $\overline{C}_2$ 时的复杂度)及计算矩阵各元素值上。

(2) 在 AprioriTid 部分的改进：通过数据库压缩的方法，在减少了要查询事务条数的同时，也减少了一些不必要候选项的产生。原 AprioriTid 部分也采用数据库压缩的方法，但在候选项的生成问题上解决的不是太好，存在明显的漏洞。

第4章 关联规则挖掘应用于股市的新参数的确定

4.1 使用关联规则挖掘新参数的意义

目前学术界对股票挖掘所用的参数都较为简单，例如神经网络用股票的开盘价，收盘价等一些直接的股票信息，还有些较为高明的方法用到些复杂的参数，如用到了股市参数KDJ等技术指标。然而股市是一个相当复杂的系统，单用一个或几个指标恐怕揭示不出其中的奥秘。因为一只股票的操纵者——庄家，本身就是由数人组成的团体，他们中间又代表了不同团体的意志，他们本身就有很多矛盾性就连他们自己也解决不了。何况是外人又怎么能看懂。中国股市有上千只股票，可以说绝大部分的股票中都有庄家藏身，这些庄家由数以万计的人组成，代表着十分广泛的思想，要想琢磨清楚恐怕是天方夜谈。于是我们只能根据他以往的操作风格或思路来预测将来他怎么操作。那么现在问题就转到怎么样寻找及总结以往他的思路和手法，再通过计算机编程表现出来。至于一些基本信息如开盘价，收盘价等，还有投资者常用的技术指标，它们都很普通，其意义已经不是很明显了，因为如果庄家用这些大家都看得懂的东西，那么钱就是大家一起赚，而不是庄家赚钱了。庄家常使用大家都看得懂的技术指标，引诱散户进行买卖，时机成熟时，他就撕下面具反手操作，使得无数股民只能望股兴叹，感叹股市的多变。市场就是这样的，庄家和散户永远是对立的。所以庄家使用的方法一定是和大家不一样的方法。至于怎么个不一样，那就看庄家对股市的认识程度了，他对大多数投资者认识的越深，他就越反其道行之；他认识的不深，他的思路跟大家差不多，他就要像大多数投资者一样，给市场教学费，再提高自己，股市中庄家自身被套的案例也不少。

如何把庄家以往的手法寻找出来，并转化为计算机语言，从而再变为适合关联规则挖掘的参数正是本章所要解决的问题。

4.2 支持新参数的股市理论

笔者对目前股市的庄家做了一个总结，主要分为三种。第一种是对于刚上市的新股就入驻的庄家，这类庄股在股票的前几个交易日里，成交量巨大，换手率充分，庄家买走了该股80%以上的股份，从此以后数年内，股价决不会跌回到原位。这种股票只要投资者长期持有（二三年），资金增幅在100%，200%都是常事，这类股票的钱最好赚，一旦看好买入后就不动，适合上班族。例如股票002024苏宁电器，002048宁波华翔。第二种是某个股票前期被炒作过，现在已经跌幅很深，而且一、二年内没有太大动静，于是有大资金可能看上，开始悄悄买入。这类庄股面临的主要问题是要获得大利润，就的要解决前期套牢盘的压力，所以拉升幅度不会像第一种情况那么高。第三种是庄家自救行为，由于各方面原因，有些庄家在买入股票后还没来得及拉升，股市就开始下跌，他也不想赔钱出局，于是就在更低位再收集些筹码，等机会拉高出局。因为是自救，所以这种股票日后拉升幅度不会太高，可能还高不过他被套时的价位，对于这类庄股我们要小心他可能随时会出货。这类情况目前在股市里还挺多。

庄家对于股票的操作方法一般是：在合适的价位悄悄地买入股票，一般需要较长时间，至于买入的方法则各有不同。在买入足量的股票后，等待大盘转好，只要大盘不坏，庄家瞄准时机就准备拉升，当然在拉升前，庄家要清理一下其中混杂的散户，能清理多少就清理多少。随后庄家一般采用迅猛而强有力的手段，在几个交易日内将股票迅速拉升到一个较高价位，使得散户还没弄清是怎么回事。等到散户稍清醒时，股价已经高的有些不敢买了。之后庄家要么继续拉升，要么来个横盘，总之使股价不会有较大回调。再拉升，拉升到满意价位时，他们就开始出货，至于出货手法也是多种多样。当然整个的操作手法不是三言两语就能说清的，遇见具体情况庄家还得自己分析。

我们的任务是找出涨幅满足要求的股票，查看其涨升前一年的原始数据（对于有的股票时间有可能不够长），主要是开盘价，收盘价，成交量等基本信息，把原始数据转化成股市的信号（主要依据笔者对股市的认识），如吸货，放量

等概念，之后再转化成关联规则能挖掘的模式。

4.3 新参数的确定

由于原始数据都是纯数字形式，所以要把这些数字转化成股市的语言，股市用精确的定量性质的语言描述的效果不太好，而用一种模糊的语言描述就显得易于理解。比如昨日成交量是 1000 手，今日成交量是 10000 手，我们就说放量了或放巨量了。所以我们准备用模糊化的方法把原始数据转化成股市的信号[41]。

4.3.1 模糊理论相关概念

1965 年，Zadeh 和 Goguen 等人第一次提出模糊集合理论[38]，扩展了传统的集合理论(经典集合)。在普通的集合论中，某一事物要么属于某集合，要么就不属于，这里没有模棱两可的情况，然而在现实生活中，却充满了模糊事务、模糊概念，像“年轻，老，现在”等这些模糊的概念，由于它没有明确的内涵和外延，所以无法用普通集合论来加以描述，也就是说，在这样的集合中，一个元素是否属于某集合，不能简单地用“是”或“否”来回答，有一个渐变的过程。

在描述一个模糊集合时，我们可以在普通集合的基础上，把特征函数的取值范围从集合{0, 1 }扩大到在[0, 1]区间连续取值，这样一来，就能借助经典数学这一工具，来定量地描述模糊集合。并把模糊集合中的特征函数称为隶属函数。在下文中，首先介绍特征函数的定义，以此为原型，给出模糊集合中隶属函数的定义。

特征函数： 对于一般意义上的集合，我们可以通过一个特征函数 $\mu_A(x)$来判断论域 X 上的一个元素 x 是否属于集合 A,当且仅当 x∈A 时，$\mu_A(x)=1$；当且仅当 x∉A 时，$\mu_A(x)=0$。即 x∈A 和 x∉A 有且仅有一个成立，其界清晰，毫不含糊。

隶属函数： 对于模糊集合$\tilde{A}$，不能严格的判断元素x是否属于$\tilde{A}$，中间有一个过渡区域，为了刻划模糊现象，有必要将离散的0, 1两点扩充为连续状态的区间[0, 1]。这样，普通集合的特征函数就扩展为模糊集上的隶属函数。在模糊集合$\tilde{A}$上，可以通过一个隶属函数$\mu_{\tilde{A}}(x)$来表示元素x属于$\tilde{A}$的隶属度。

α截集： $\tilde{A}$的α截集取所有元素x对$\tilde{A}$的隶属度大于等于阈值α的集合，记为:

$$\tilde{A}_{\alpha}=\{x\in X|\mu_{\tilde{A}}(x)\geq\alpha\}$$

标量基数 定义在论域X上模糊集合$\tilde{A}$的标量基数是指X中所有元素在$\tilde{A}$上隶属度的和。即$|\tilde{A}|=\sum_{x\in X}\mu_{A}(x)$

模糊集的运算:在模糊集的所有运算中，补集、并集和交集是最常用的，下面分别介绍这几种运算:

1)补集: 模糊集A的补集用$\neg A$表示，补集$\neg A$的隶属函数为:

$$\mu_{\neg A}(x)=1-\mu_{A}(x),\ \forall x\in X$$

2)并集: 设两个模糊集合$\tilde{A}$与$\tilde{B}$的并集记为$\tilde{A}\cup\tilde{B}$，其隶属函数定义为:

$$\mu_{\tilde{A}\cup\tilde{B}}(x)=\max\{\mu_{\tilde{A}}(x),\ \mu_{\tilde{B}}(x)\},\ \forall x\in A$$

3)交集: 设两个模糊集合A与B的交集记为A∩B，其隶属函数定义为:

$$\mu_{\tilde{A}\cap\tilde{B}}(x)=\min\{\mu_{\tilde{A}}(x),\ \mu_{\tilde{B}}(x)\},\ \forall x\in A$$

4.3.2 通过模糊时间序列匹配方法获得的模糊参数

4.3.2.1 时间序列简介

股票数据是含有时间属性的按一定顺序存放的数据集合。所以在引入模糊概念的同时，我们按照处理时间序列的方法来处理股票数据。目前，对时间序列的研究大致集中在以下几方面：

1.时间序列的相似性研究：一般有两个研究方向，一种是将时间序列从时域映射到频域后再进行相似性匹配，一种是直接在时域内进行研究。主要应用包括：

a 从股票数据中识别具有相似变化趋势的模式，以预测新数据的在未来的发展行为。

b 超市中具有相似销售模式商品的进货预测等。

2.时间序列的值预测：包括多步预测和单步预测，将时间序列视为一个动力系统，认为在其过去的波动中蕴涵有可用于预测未来的信息，并以此为基础进行下一步或多步的值预测[39]，可用于股票在今后一天或几天的价格预测。

3.时间序列关联规则的抽取：通过固定长度的窗口将时间序列离散化成一系列子序列，研究子序列之间的相似性，然后将相似的子序列进行聚类形成模式，应用关联规则的研究方法从各种模式中抽取关联规则，可以得到一个时间序列内部不同模式之间的关联规则或不同时间序列之间模式的关联规则。其规则形如“如果第一天Microsoft上涨而且Intel下降，则IBM第二天上涨”。

传统上定义时间序列的数据挖掘是从时间序列数据中抽取分类规则。方法是首先通过极值法将时间序列在极值点处分割开，形成一系列模式，则每种模式内部的行为趋势不变（上升或下降），把决定各种模式的条件属性和分类属性组成一个信息表，这个信息表将与时间无关。然后通过关联规则方法或其他方法从信息表中抽取规则形成规则集，用该规则集可以对时间序列进行趋势预测（以股票数据为例，是指用规则集对股票的某种上涨或下跌的行为趋势进行持续时间长或短的预测其规则形式如“如果目前股票形势波动比较厉害，则本次行情持续时间将很短”）。这种预测结果对股票投资者的长期投资行为将更具指导意义。

本篇所用的方法与传统方法有以下几处不同之处：1.除了用极值法形成一系列模式外，还另外加进些其他指标，这些指标不是用极值法得到的。2.预测的目标是针对中短期投资的。本文的研究内容包括了时间序列的相似性研究和时间序列关联规则的抽取两个方面。

4.3.2.2 模糊时间序列的建模

证券市场中被认为最重要的三个因素是：时间，价格，成交量。实战家，

理论家在研究股票时除了要考虑不可测因素外，恐怕最多涉及的也就是这三个因素了。经典的量价理论认为价格是由成交量支持的，没有成交量支持的价格是不可靠。只有价涨量升才是真实的，才适合操作。至于价格与时间方面的联系，目前论述的书籍相对较少，也没比较公认的理论。就笔者认为价格与时间方面的关系可以总结为：相互转化。例如，理论上某只股票需要5天下跌7%的回调，而实际可能演化成10天下跌4%的情况。这就是以时换空或以空换时的表现。所以本文突出把三者都作为重点结合起来考虑。当然股市的各种消息很多，除了其自身原因外，其它非自身因素也起了相当作用，这里无法考虑到。

首先，我们按极值法的思路把股票数据序列分隔成一系列内部行为趋势不变模式，即把股票数据序列划成几段。然而若用股票每日的收盘价或开盘价等数据来划段，由于它们的波动性太随机，所以必然产生巨量的段，可是其中很多段又表现不出特定的意义，而且处理起来也相当费事。于是我们采用另一个工具来建模[40]，这个工具就是5日平均线。5日平均线不但能滤去过于凌乱的信息，起到平滑的作用，而且信息也相对保存地比较好，遗失的信息较少，能够基本反映股价每天运动的情况，保留了主要的短期趋势。更重要的是它在股票实战中起着至关重要的作用，绝大多数短线高手都是以5日均线来作为买卖信号。不少实战家，理论家在他们的著作中也都强调了5日均线作用[43-45]。所以从理论和实战的角度讲，5日平均线是最好的建模工具。然而在实际建模中我们发现还是有些杂波没有滤除干净。如图4.1，AB段就是杂波，我们认为AB段时间短，斜率小，作用不大，要把它合并到其邻段中去。这里有个合并原则，我们稍后再说。这样我们就可以得到一系列内部行为趋势不变的段。为其记录先作如下定义：

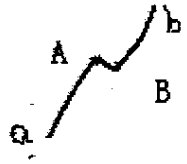

图4.1 实际存在的一种情况

设某个基本形态L是由连续的n条首尾相接的线段组成$l_1, l_2, \cdots l_n$，设S是每条线段所包含的属性集合，含有3个属性S_1, S_2, S_3,每个属性S_i (i=1,2,3)有各自论域。

例如每条线段有成交量，时间，斜率这3个属性（实际中换手率更能反映交易的情况，所以用换手率来代替成交量），而换手率的论域为：{很大，大，正常，小}；时间的论域为：{超常，长，正常，短,超短}；斜率的论域为：{正向陡峭，正向正常，平缓，负向正常，负向陡峭}，其中{正向陡峭，负向陡峭}为同等最高级，{正向正常，负向正常}为同等次高级。为了简便期间我们当隶属函数值大于某个阈值时，就肯定它的性质，即隶属度取值为1，这样可以定义换手率值>=5时为很大；值在[2,5)为大；值在[1,2)为正常；值<1为小等等，于是该基本形态可用如下形式表示:{{大,正常,正向陡峭}$_1$,{正常,短,负向陡峭}$_2$,…,{正常,长, 正向陡峭}$_n$}。其中：某条线段的换手率＝该时间段换手率之和/该时间段长度。

现在来说一下刚才提到的合并原则。具体的合并原则为：时间为超短，斜率为平缓，则把该段与其邻段斜率小的合并，合并后的新段信息为：换手率取两者的较大值；时间记法：若两段都为超短，则记为短，否则以时间长者记；斜率则以没合并前的起始点与终点连线的斜率记。

具体的建模过程如下：

1）记录 5 日平均价格线信息：价格 $p[i]$，换手率 $m[i]$。i 表示时间。

2）若有连续几天(包括 1 天)的涨或走平或跌，称之为事件，则记录发生 g 该事件的首尾 2 个时间，把这些时间放入一个数组 $p1[50]$中。

3）用数组 $p1$,p,m 计算每个事件的相关信息。可用下面的方法：

a.计算第 i+1 个事件的时间:$p1[i+1]-p1[i]$；将其转化为相应的模糊信息；

b.计算第 i+1 个事件的斜率:$(p[p1[i+1]]-p[p1[i]])/(p1[i+1]-p1[i])$; 将其转化为相应的模糊信息；

c.计算第 i+1 个事件的换手率: $\sum_{k=p1[i]}^{p1[i+1]} m[k]/(p1[i+1]-p1[i])$; 将其转化为相应的模糊信息；

4）检查所有事件的时间和斜率，如果某些事件的时间为超短而且斜率为平缓，则把该事件与其前事件或后事件合并，合并原则按前述的方法进行。

4.3.2.3 模糊时间序列建模及其应用结果

我们先在 2004.7.1——2004.11.23 这一时段 1436 只股票（去除 ST 类，基金，高价股）中找出 10 个交易日内有 25%或以上涨幅的股票，然后以上涨前 90 个交易日里的 5 日平均价格线建模,合并满足要求的线段。对于结果，先检查整条线段中换手率是否都是正常或小，若是，则把该整条线段删除。因为经验告诉我们没有成交量支持的股票多数都不可靠。随后又发现即使这样后，整条线段所含的线段仍然较多，于是进一步合并。合并原则如下：换手率为正常或小，时间为短，斜率为平缓，则把该段与其邻段斜率小的合并，合并后的新段信息为：换手率取两者的较大值，时间为正常，斜率则以没合并前的起始点与终点连线的斜率记。该原则主要是突出成交量的作用，因为实际上是成交量的放大才支持股价的上涨。最后得到了 143 个基本模型。其中有一些文献[40] 中提到的模型，如上升三角形，“w”底，还有一些变形的模型，如旗形整理中夹杂着平衡三角形，更有很多新模型，如“大螃蟹”，形成大概 50 个交易日，大型“M”底，形成大概 80 个交易日左右，还有些模型不好总结。

我们选某段时间如 2005.1.14——2005.4.20 的 1436 只股票作为验证，找出有 15%或以上上涨或下跌的股票，一只股票可能有几次满足要求，则要重复记录。所有满足上涨或下跌的股票总数记为 s，上涨的股票总数记为 z，下跌的股票总数记为 d，然后用其上涨或下跌前 100 个交易日的 5 日平均价格线建模，方法同上，建模好后，则与 143 个基本模型比较，匹配。匹配原则如下：

1）比较的同段线段，若换手率一个是很大或大，另一个是正常或小，则认为不匹配。

2）换手率都是很大或大，或都是正常或小时，查看时间与斜率，若相差和>=2 时（例如一个时间为长，另一个为短，则差为 2；又如一个斜率为正向正常，另一个为平缓，则差为 1），则认为不匹配。

3）若整条线段相差和>4,则认为不匹配。

4）其余情况则认为是匹配的。

相关算法可参考文献[42]。

匹配出的结果中，若其后走势是涨的总数记为 zm，是跌的总数记为 dm，于是可以定义支持度为：

$$sup=（zm+dm）/s$$

支持度的实际意义是模型出现次数的多少，支持度如果较低，则实际意义不大。

置信度定义为：

$$con=zm/（zm+dm）$$

其意义是代表模型的成功率。

结果：找出了2060只股票有15%或以上上涨或下跌，其中 $z=813$，$d=1247$，$zm=656$，$dm=230$，支持度 $sup=0.43$，置信度 $con=74\%$。上述结果说明在2005.1.14——2005.4.20这62个交易日中共有 $zm+dm=886$ 次操作机会，每次胜率为74%。

自2004.4.7股市形成中期头部后，股指一直运行在下降通道内，期间有很少的几次反弹，很多股票的跌幅都比大盘的跌幅大。在这种情况下建模、选股，难度自然加大，从而影响了准确度，然而从另一个方面看说明参数的选择还是有一定意义的，较为正确的。

4.3.3 模糊参数的转化及其它参数的添加

现在把每只股票通过几段信息来表示，然而每只股票被划分成了多少段并不知道，或者说股票被划分的段数可能不一样，有的可能划成5段，有的可能划成8段。关联规则挖掘要求属性固定，所以上面的参数显然不适合直接用于挖掘，于是我们要对其作个转化，变成适合关联规则挖掘的形式。我们的目的是尽可能地保留下转换前的信息，少丢失信息。

分两个区段记录股票，第一区段记录上涨段，第二区段记录下跌段，首先，出于成交量重要的原则，先找出是否有放量段。

（1）若无则计为0，从第一段开始，进行以下步骤，a.以时间为属性继续查找，若时间是长或超常，则该属性值计为1，在此基础上，再查看斜率，若为正向陡

峭段或负向陡峭段，则对应属性值计为1；b.若以时间为属性查找时，该属性值为0，则查看斜率，若为正向陡峭段或负向陡峭段，则对应属性值计为1。

（2）若有则计为1，以该几段为基础进行以下步骤，a.以时间为属性继续查找，若时间是长或超常，则该属性值计为1，在此基础上，再查看斜率，若为正向陡峭段或负向陡峭段，则对应属性值计为1；b.若以时间为属性查找时，该属性值为0，则查看斜率，若为正向陡峭段或负向陡峭段，则对应属性值计为1。

例如：某股票a有2段放量上涨，这2段放量段中有一段的时间是长或超常，而这1段的斜率不为正向陡峭；下跌的4段中有2段放量，在这2段放量段中有1段时间是长或超常，而这1段的斜率为负向陡峭。于是它们的记录可由表4.1来表示：

表4.1 股票a的参数的表示方法

放量段之和1	时间段1	斜率段1	放量段之和2	时间段2	斜率段2
1	1	0	1	1	1

又如：某股票b有2个上涨段，但该2段没有放量，于是从第一段开始查看时间属性，发现2段时间属性值为0，于是从第一段开始查看斜率属性，发现第2段斜率属性值为1。于是它的记录可由表4.2来表示：

表4.2 股票b的参数的表示方法

放量段之和1	时间段1	斜率段1
0	0	1

在对股票时间序列建模后的应用结果分析中，我们发现结果还不是十分令人满意，于是有必要对原方法进行重新分析。在重新参考了大量股票理论，实战的书籍后[46-53]，发现了改进的地方。前面的建模方法是从整体上对股票进行的描述，可能把庄家长时间的吸货过程描述的较清晰，然而对吸完货后如何驱

除散户似乎描述的不太清楚。庄家在拉升前期往往用一些短期K线组合如横盘等来打消散户的跟风意念或驱逐出意志不坚定的人，从而减轻拉升时的压力，节约资金。于是我们把思路转移到如何描述短期的行为上。通过书籍及模拟实战，我们找到了一些性质并把它们转化成关联规则挖掘的属性形式。它们是：上影线，下影线，长实体，横盘，连续阳，连续阴，涨幅百分比数量。这样一共设置13个属性来作为关联规则挖掘的新参数。它们是：放量段之和（上涨段），时间段（上涨段），斜率段（上涨段），放量段之和（下跌段），时间段（下跌段），斜率段（下跌段），上影线，下影线，长实体，横盘，连续阳，连续阴，涨幅百分比数量。分别用 $I_1 \cdots I_{13}$ 对应。

4.4 新参数与以往参数的比较

以往参数所考虑的仅是股票的部分表象特征，没有深入研究股票的内在实质，重视这个的同时必然忽视了那个，缺乏对股票的深层认识。参数的局限性较大，挖掘出的结果对于应用起不到较好的决定作用。

新参数充分考虑到股票的长期走势与短期走势相互结合，兼顾到股票运作的特点，从大局和细节两处入手，体现了庄家的运作思路。在关联规则挖掘及其应用模拟实战中起到了良好效果。

第 5 章 实验及结果

5.1 实验环境

实验基于 Windows 2000 Professional 操作系统，在 Microsoft Visual C++ 6.0 环境下编程;后台数据库采用 Access 2000。 数据库接口采用 ODBC(开放数据库互连)方式。各功能主要通过菜单实现。在输入置信度时有人机交互界面。

5.2 运用于股票数据的挖掘结果及解释

为了验证本文所述的算法的可行性，采用上海证交所和深圳证交所 2004 年 12 月 20 日至 2005 年 11 月 25 日 1000 多支股票价格数据库作为测试集。数据库信息是由上海大智慧－Internet 版炒股软件导出的。

首先找出在 10 个交易日内有 25%以上上涨幅度的股票（找到近 400 支股票），再把每支股票上涨前 100 个交易日的数据通过第 4 章的方法转化为长期参数，再把每支股票上涨前 10 个交易日的详细数据，包括开、收盘价，最高、低价，每日成交量等转化成短期参数，最后用 Apriori 算法、AprioriHybird 算法和改进的 AprioriHybird 算法分别对这些参数进行挖掘，得到较多令人满意的关联规则。现举一例进行说明。其中 I1 表示放量段之和（上涨段），I2 表示时间段（上涨段），I3 表示斜率段（上涨段），I4 表示放量段之和（下跌段），I5 表示时间段（下跌段），I6 表示斜率段（下跌段），I7 表示上影线，I8 表示下影线，I9 表示长实体，I10 表示横盘，I11 表示连续阳，I12 表示连续阴，I13 表示涨幅百分比数量。

R1：I1,I4,I5,I6,I7,I9,I11 （26.5%，86.2%）

其意思是：有过多次带量上涨，但时间每次不长，上涨幅度不大；也有过多次带量下跌，下跌力度大，下跌时间长，在真正上涨前曾出现过上影线，长

实体，连续阳的特征。在所有股票中有 26.5%的股票能产生该规则，其中能上涨的占 86.2%。

其股市解释为：在该时间段内，有一批股票经过了大幅放量下跌，当然在下跌的过程中也出现过几次反弹，反弹的性质多为庄家出货逃命，也包括散户。当股票跌到一定价位时，庄家觉得可以买入了，于是出现一些放量，长实体等信号，它们标志着庄家的买入，由于庄家也不敢确定他买入的时机就是对的，所以买的并不多，从而没有出现时间长、幅度大的上涨。

还有些代表性的规则，在此仅列出其中部分，稍作解释。

R2：I2,I4,I6,I8 （21.6%，70.1）

解释：没有 I1 说明上涨时无量，I2 则表示有一些长时间的反弹，没有 I3 说明反弹力度不大，I4 和 I6 则表示下跌时凶狠，I8 是上涨前有下影线的标志。

R3：I1,I2,I3,I4,I5,I6,I7,I8 （18.8%，65.3%）

解释：I1,I2,I3,I4,I5,I6 表示上涨和下跌都较有力度且时间较长，I7,I8 则表示上涨前有上影线和下影线同时出现。

R4：I5,I6,I9,I10,I11,I12（5.6%，78.3%）

解释：没有 I1,I2,I3 意为反弹十分虚弱，I5,I6 表示下跌时间长，力度大，I9,I10,I11,I12 说明上涨前有长实体，横盘，连续阳，连续阴的特征。

5.3 使用挖掘出的关联规则模拟买卖股票的情况

笔者在 www.gwgz.com 网站上注册了几个帐号用于模拟买卖股票。买卖股票的原则有的根据第 4 章的结果，有的是根据挖掘出的规则，还有些是根据笔者的经验，主要是根据挖掘出的规则。现将结果列出供大家参考。有兴趣的读者可以打开炒股软件进行跟踪。由于该系统每个月进行一次数据删除，所以只能记录一个月的买卖情况。笔者的每次买卖都是正常的交易情况，没有利用系统的缺陷进行作弊。

图 5.1 2005 年 9 月模拟买卖战况

图5.1是笔者注册的网名“非广通3”在2005年9月的模拟买卖情况。其中根据第4章得到的模型“W”型底，及连续几日的持续放量的特征，于8月22日至26日持有000693聚友网络，持有价格为6.05元—6.77元，涨幅为11.9%。根据第4章得到的模型“V”型底，及数个交易日的持续放量的特征，于9月2日至15日分别持有600421春天股份和600644乐山电力，持有价格分别为4.63元—4.76元，3.9元—4.17元，涨幅分别为2.8%和6.9%。根据板块联动的经验，当时ST板块表现的最为突出，所以于9月15日至20日持有000034ST深泰，持有价格为2.09元—2.4元，涨幅为14.3%。该段时间模拟盈利30.3%，而同期大盘从8月22日的1148.97点上涨至9月20日的1223.56点，涨幅为6.5%。

图5.2是笔者注册的网名“非广通3”在2006年3月的模拟买卖情况。其中根据得到的关联规则：缩量洗盘，及突破大阳线，有跳空等特点，于3月2日至21日持有600439G瑞贝卡，持有价格为9.42元—9.91元，涨幅为5.2%。根据关联规则：平台缩量，带量突破，于3月21日至29日持有600663G陆家嘴，持有价格分别为8.35元—8.8元，涨幅为5.4%。分别根据第4章模型：楔型整理和关联规则：平台缩量，带量突破，于3月29日至31日分别持有600627上电股份和000616亿城股份，持有价格分别为12.27元—12.27元，5.63元—6.05元，涨幅分别为0%和7.5%。该段时间模拟共盈利9.97%，而同期大盘从3月2日的1308.2点下跌至3月31日的1298.3点，涨幅为—0.76%。

图5.3是笔者注册的网名“非广通1”在2006年3月的模拟买卖情况。其中根据模型：三角形整理，于3月6日至29日持有000612焦作万方，持有价格为5.04元—5.4元，涨幅为7.1%。根据关联规则：平台突破时有上影线，于3月29日至30日持有000612焦作万方，持有价格为5.18元—5.33元，涨幅为2.9%。根据关联规则：旗型整理突破，带有放量大阳线，于3月30日至31日持有600309烟台万华，持有价格为17.05元—17.55元，涨幅为2.9%。该段时间模拟共盈利9.76%，而同期大盘从3月6日的1288.26点上涨至3月31日的1298.3点，涨幅为0.78%。

图5.4是笔者注册的网名“非广通”在2006年3月的模拟买卖情况。其中根据关联规则：大型“W”底突破，带有10%幅度的大阳线，于3月6日至7日持有000534汕电力A，持有价格为3.52元—3.47元，涨幅为—1.4%。根据

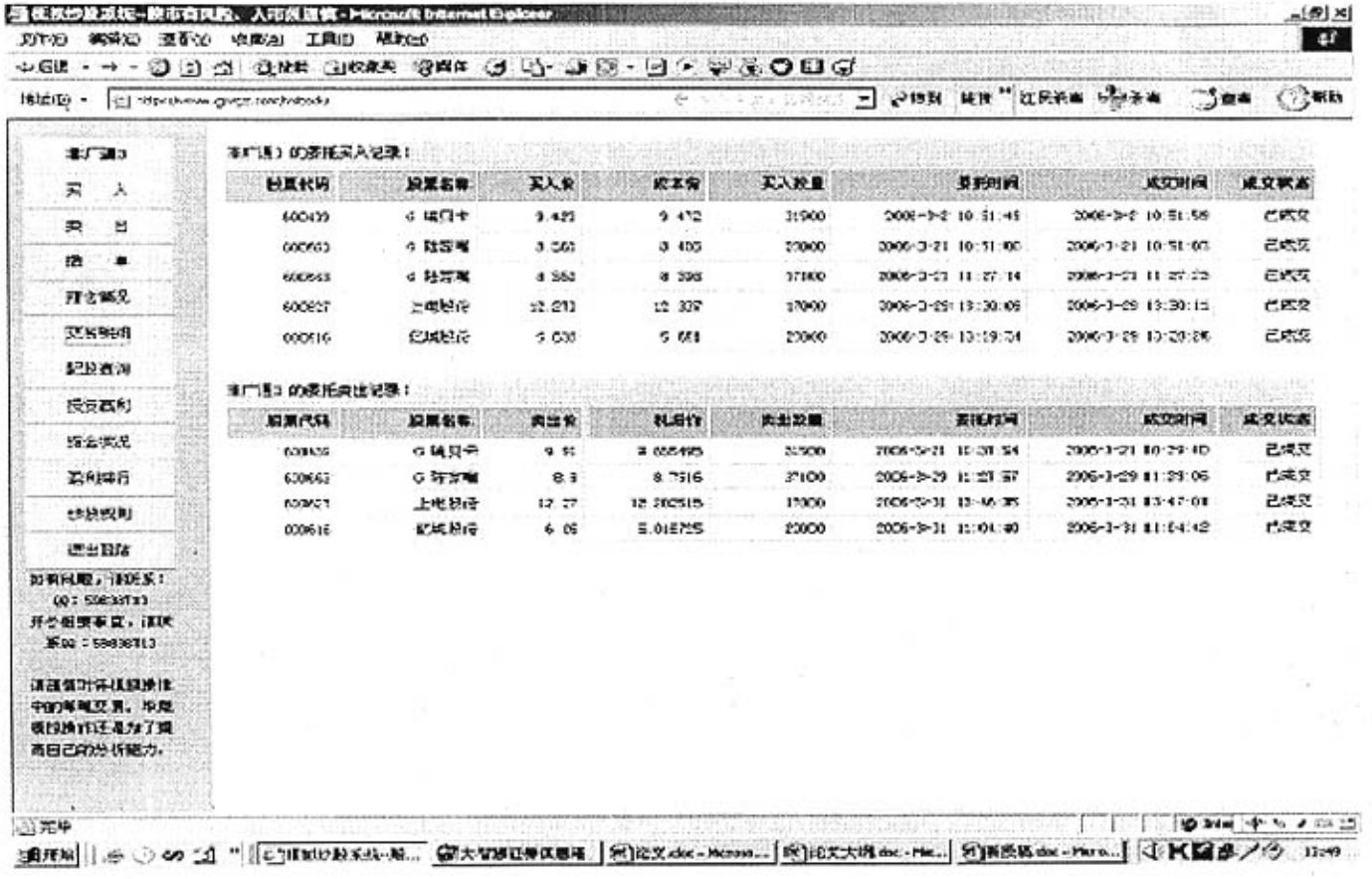

图 5.2 2006 年 3 月模拟买卖战况

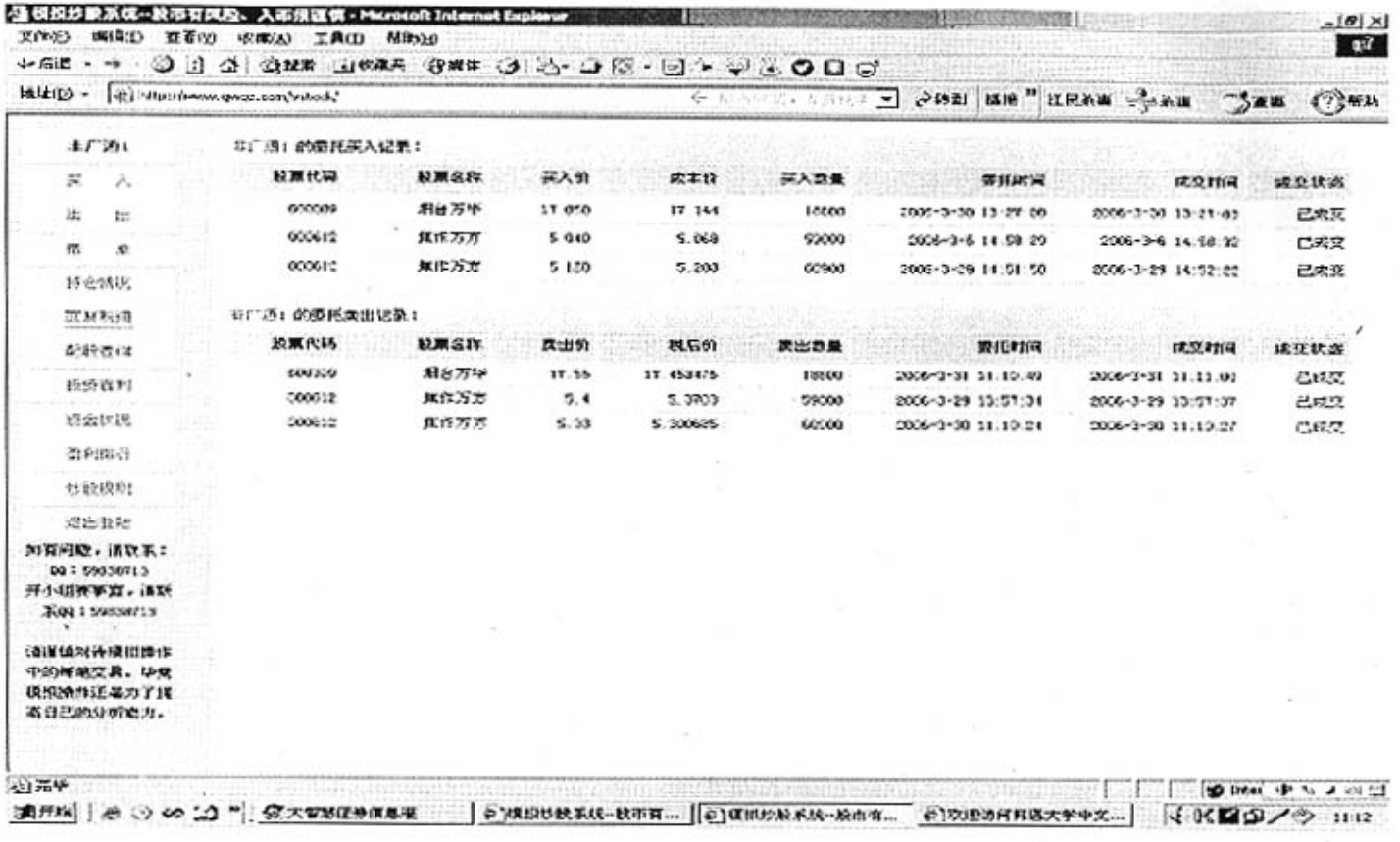

图 5.3　2006 年 3 月模拟买卖战况（非广通 1）

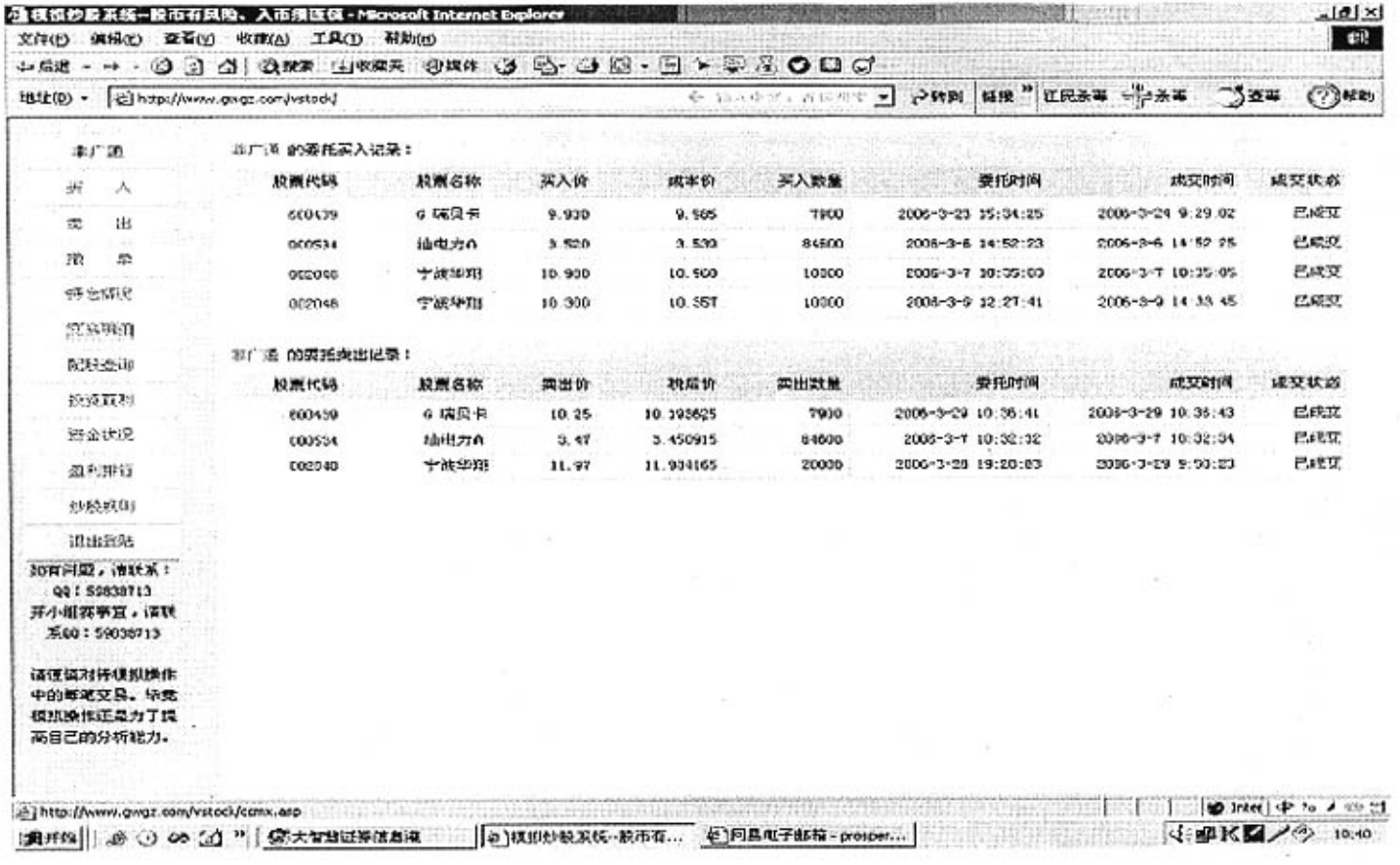

图 5.4　2006 年 3 月模拟买卖战况（非广通）

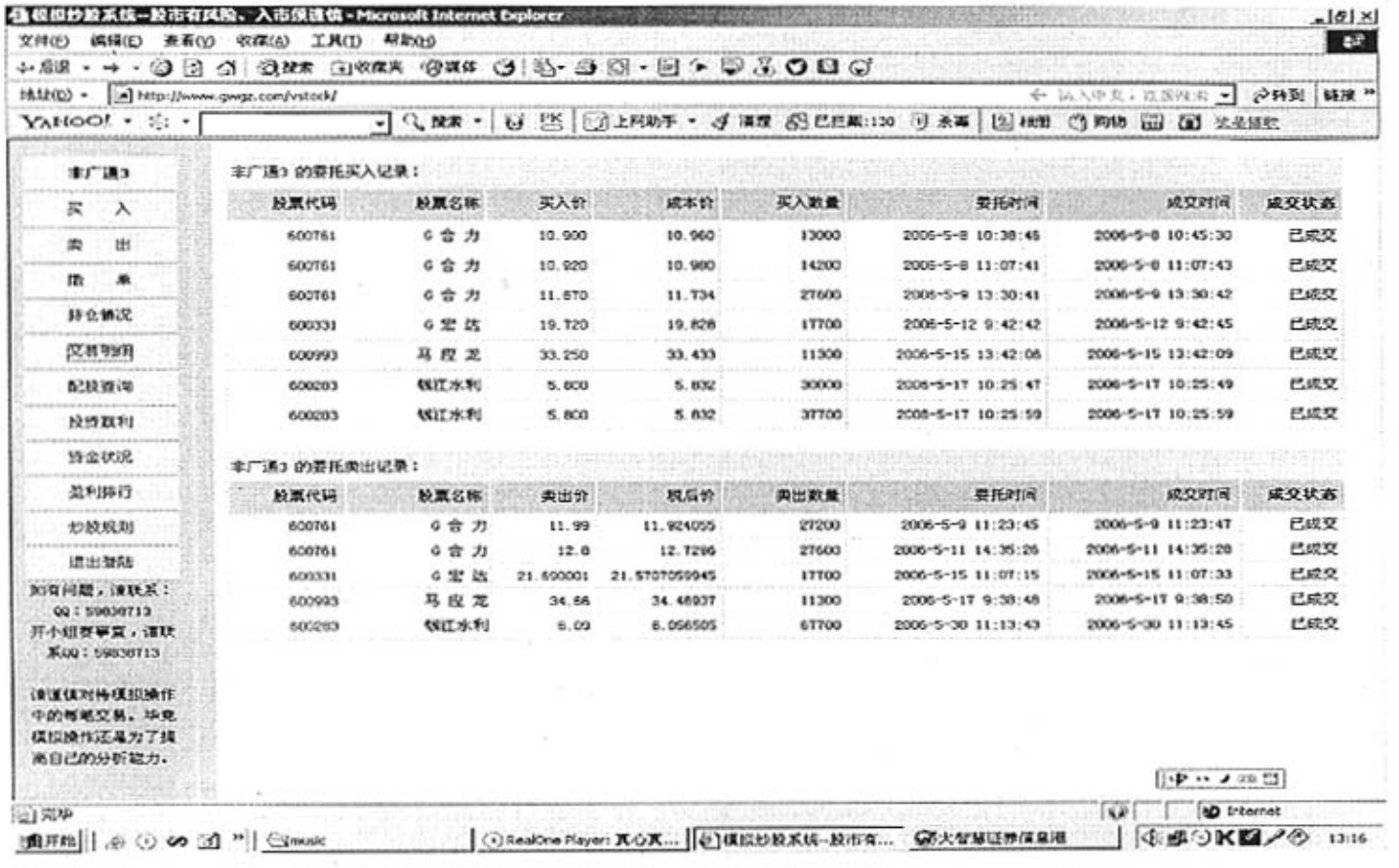

图 5.5　2006 年 5 月模拟买卖战况

经验：放量回调后能够稳住，于3月8日至28日持有002048宁波华翔，持有价格为10.6元－11.97元，涨幅为12.9%。根据关联规则：复合型“W”底突破，并有带量阴上影线，于3月23日至29日持有600439 G瑞贝卡，持有价格为9.93元－10.25元，涨幅为3.2%。该段时间模拟共盈利6.36%，而同期大盘从3月6日的1288.26点上涨至3月31日的1298.3点，涨幅为0.78%。

图5.5是笔者注册的网名“非广通3”在2006年5月的模拟买卖情况。其中根据关联规则：横盘突破，带有10%幅度的大阳线，于5月8日至9日持有600761G合力，持有价格为10.91元－11.99元，涨幅为10%。根据分时图经验：急涨后必有横盘，于5月9日至11日重新持有600761G合力，持有价格为11.67元－12.8元，涨幅为9.68%。根据关联规则：板块效应，并有10%幅度的大阳线，于5月12日至15日持有600331 G宏达，持有价格为19.72元－21.69元，涨幅为10%。根据经验：冲高的回调可入，于5月15日及17日分别持有600993马应龙和600283钱江水利，分别于5月17日及30日卖出该两支股票，分别盈利4.24%和5%。该段时间模拟共盈利37.1%，而同期大盘从5月8日的1440.22点上涨至5月30日的1657.3点，涨幅为15.1%。

综上所述，一共进行了19次模拟买卖（同天买入的同支股票属1次操作），其中有2次略亏操作（亏幅不超过1.5%），1次略赢操作，10次盈利在3%－7%之间，6次盈利在10%以上，最大盈利的股票为ST深泰，盈利值为14.3%。盈利比率为17/19＝89.5%。其中当天买入第2天即卖出者，共有5次，4次成功，分别是000612焦作万方，盈利为2.9%；600309烟台万华，盈利为2.9%；600761G合力，盈利为10%；600331 G宏达，盈利为10%；1次失败，000534汕电力A，盈利为－1.4%。

以上买卖操作证明了所得的关联规则及第4章建模结果的正确性和实用性及其对短线操作的指导意义。

5.4 改进算法与 Apriori，AprioriHybird 的实践比较

图 5.6 是 Apriori，AprioriHybird 及改进算法执行效率图，我们使用的数据是 5.2 节的数据，该数据库条目数近 400 条，每条事务有 13 个属性（字段），平均每个属性占3个字节。从图5.6可以看出随着最小支持度的逐渐减小，Apriori 算法由于不断重复扫描数据库，因而性能急剧降低，而 AprioriHybird 算法仅扫描数据库有限次，在一定程度上解决了部分问题，而改进的 AprioriHybird 算法由于采用了 2 项集支持度矩阵后，很好地解决了频繁 2 项集和频繁 3 项集的求解，从而又减少了数据库扫描次数。因此，随着支持度的增加，其性能非常稳定，需要的时间没有发生突变的增加，并且因为采取了更加有效地缩减数据库的方法：当 k>3 后，使用 AprioriTid 方法和减小数据库的策略可以更加有效地减小数据库，从而使求解频繁 k 项集所需要的时间相对于求解频繁 2 项集的时间更小。从实验结果可以看出来，改进的 AprioriHybird 算法的效率比 AprioriHybrid 算法的效率确实有实质性的提高。

我们再使用上述400条数据中的200条数据来作实验，得到图5.7所示结果。从图中可以看出三种算法的效率差别不是太大。因为数据库内容的减少，Apriori 算法需要扫描数据库的时间大为减少，而 AprioriHybrid 算法和改进的 AprioriHybrid 算法减少扫描数据库的优势体现的并不是很明显，所以造成了三者效率相差并不大的结果。

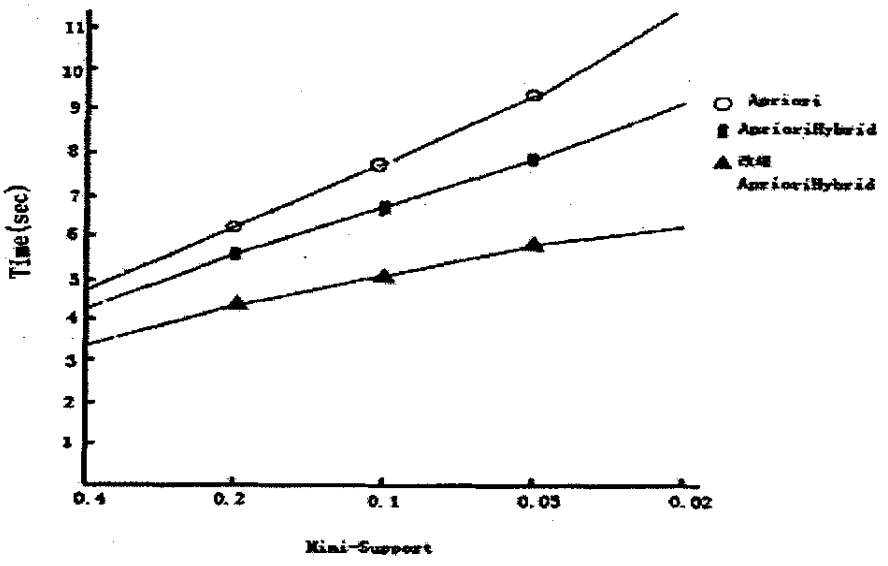

图 5.6 Apriori 算法，AprioriHybird 算法及改进算法执行效率图（400 条）

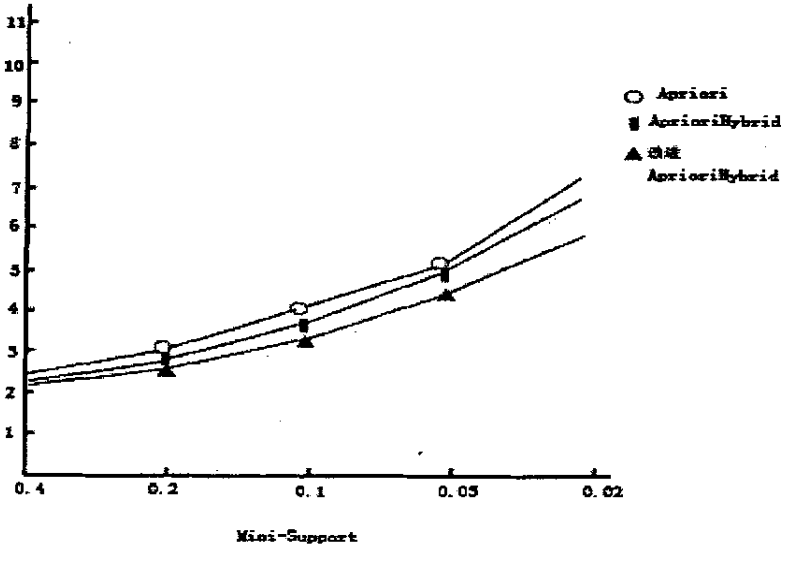

图 5.7 Apriori 算法，AprioriHybird 算法及改进算法执行效率图（200 条）

结束语

本文应用数据挖掘技术对股票进行分析与预测，为投资者提供了一种较新的方法。其主要工作在于:

1.对股票市场作了较为深入的研究，主要从技术层面分析了股票上涨的原因及实质。通过对股票数据的建模，重新定义了参数，为下一步的挖掘工作打下了良好的基础。

2.对关联规则挖掘算法提出了改进 AprioriHybrid。通过与 AprioriHybrid 算法从时间复杂度和空间复杂度两方面予以比较，可以看出，改进的 AprioriHybrid 算法的前半部分比 AprioriHybrid 算法的前半部分要占用更多的内存空间，但通过空间换时间的策略减少了这半部分的运行时间，后半部分通过减少数据库条目和减少生成项目集的方法进一步节约了运行时间。

3.挖掘出了较多实用的规则，对投资者有更好的指导作用，特别对买入时机给予了更多提示。

我国股市虽经多年发展，然而还有较多不成熟的地方，股市受政治、经济及人为因素干扰较大，上市企业所提供的财务数据不够透明，造假情况也时有发生，等等。这些因素都注定了无论用什么方法研究都是有缺陷的，得不出十分客观的规律。因此本文所用的方法并不是终结，还有值得推敲和改进之处。

另外，一些更为细腻的坐庄手法，本文还未能表示出来加以实现。

致 谢

本论文从选题到写作都是在导师杨燕副教授的悉心指导下完成的。在两年多的学习生活和论文的进行工作中，杨老师为学生的进步倾注了大量的心血和汗水，在此，向我的导师致以衷心的感谢和深深地敬意。导师高深的学术造诣、严谨的治学态度、一丝不苟的工作作风、渊博的知识以及诲人不倦的导师风范，使我在学习生活中受益非浅，并激励着我在今后的学习工作中不断进步，这里再次对导师致以最诚挚的谢意。

在这里我要感谢我的父母，他（她）们在学习和生活上给予我无私的鼓励和支持，使得我能够在学习中充满信心，克服困难，顺利完成学业。无论今后走到哪里我都不会忘记他（她）们。

在论文的写作中，我得到了计算机学院老师及同学的热心帮助和支持，在此，谨向杨燕老师以及所有给予我帮助和鼓励的老师及同学致以诚挚的谢意和由衷的敬意!

作者于 2006 年 5 月

参 考 文 献

[1] M.S. Chen, J.W. Han and S.Y. Philip, Data Mining: An Overview from a Database Perspective, IEEE Transaction on Knowledge and Data Engineering, 866-883, 1996.

[2] R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proc. 1993 ACM SIGMOD Int' 1 Conf.Management of Data, Washington, D. C., pp. 207-216, 1993.

[3] 范明，孟小峰等．数据挖掘概念与技术．机械工业出版社，2001．8

[4] 邵峰晶，于忠清．数据挖掘原理与算法．中国水利出版社，2003．8

[5] 数据挖掘资料汇编 http://www.dmgroup.org.cn.

[6] R.Agrawal, T.Imielinski and A. Swami, Mining association rules between sets of items in large databases, ACM Press Publisher, 207-216, 1993.

[7] R.Agrawal and R.Srikant, Fast Algorithms for Mining Association Rules in Large Databases, In Research Report RJ 9839, IBM Almaden Research Center, San Jose, Califomia, 1994.

[8] H.K. Mannila, H. Toivonen and A.I. Terkamo, Efficient Algorithms for Discovering Association Rules, Usama M.Fayyad and Ramasamy Uthurusamy, Proceedings of AAAI'94 Workshop on Knowledge Discovery in Databases (KDD94), Seattle Washington, 1994, AAAI Press Publisher, 181-192, 1994.

[9] J.S. Park, M.S. Chen and P.S. Yu, An Effective Hash Based Algorithm for Mining Association Rules.Michael, J. Carey and A.S. Donovan, Proceedings of the ACM-SIGMOD International Conference On Management of Data (SIGMOD 95), San Jose, California, 1995, ACM Press Publisher, 175-186, 1995.

[10] R.Agrawal and R.Srikant, Fast Algorithms for Mining Association Rules, J. B.Bocca, M.Jarke and C. Zaniolo, Proceedings of the 20th International

Conference On Very Large Databases (TLDB 94), Santiago, Chile, 1994, Morgan Kaufmann Publisher, 487-499, 1994.

[11] J. Han and Y.Fu, Discovery of Multiple-Level Association Rules from Large Databases, Proceedings, of the 21st International Conference On Very Large Databases (TLDB95), Zurich, Switzerland, Morgan Kaufmann Publisher, 420-431, 1995.

[12] A.Savasere, E.Omiecinski and S.Navathe, An efficient algorithm for mining association rules in large databases, Proceedings of the 21st International Conference On Very Large Databases (TLDB 95), Zurich, Switzerland, 1995, Morgan Kaufmann Publisher, 432 – 443, 1995.

[13] H.Toivonen, Sampling Large Databases for Association Rules, Proceedings of the 22nd International Conference On Very Large Databases (TLDB 96), Bombay, India, 1996, Morgan Kaufmann Publisher, 134-145, 1996.

[14] S. Brin, R. Motwani, J.D. Ullman and S.Tsur, Dynamic Itemset counting and implication rules for market basket data, Proceedings of the 1997 ACM-SIGMOD International Conference On Management of Data (SIGMOD '97), Tucson, Arizona, 1997, ACM Press Publisher, 255-264, 1997.

[15] R.Agrawal and R.Srikant, Mining sequential Patterns, Proceedings of the 11th International Conference on Data Engineering (ICDE95), Taipei, Taiwan, 1995, IEEE Computer Society Press Publisher, 3-14,1995.

[16] M.I. Zaki, N. Lesh and M. Ogihara, PlanMine: Sequence Mining for Plan Failures. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD 98), New York City, New York, 1998, AAA I Press Publisher, 369-374, 1998.

[17] S. Guha, R. Rastogi and K. Shim, ROCK: A Robust Clustering Algorithm for Categorical Attributes. Proceedings of the 15th International Conference on Data Engineering(ICDE 99), Sydney, Austrialia, 1999, IEEE Computer Society

Press Publisher, 512-521, 1999.

[18] H.Mannila, H. Toivonen and A.I. Terkamo, Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery, l (3): 259-289, 1997.

[19] K.Koperski and J.Han, Discovery of Spatial Association Rules in Geographic Information Databases, Proceedings of the 4th Internatioual Symposium on Large Spatial Databases (SSD95), Portland, Maine, 1995, Springer Publisher, 47-66, 1995.

[20] B.Ozden, S.Ramaswamy and A.Silberschatz, Cyclic Association Rules. Proceedings of the 14th International Conference on Data Engineering (ICDE'98), Orlando, Florida, 1998, IEEE Computer Society Press Publisher; 412-421, 1998.

[21] A.Savasere, E.Omiecinski and S.Navathe, Mining for Strong Negative Associations in A Large Database of Customer Transactions. Proceedings of the 14th International Conference on Data Engineering (ICDE'98), Orlando, Florida, 1998, IEEE Computer Society Press Publisher, 494-502, 1998.

[22] H.Lu, J.Han and L.Feng, Stock movement prediction and n-d imensional inter-transaction association rules, Proceedings of the 3rd ACM-SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD98), Seattle, Washington, 1998, ACM Press Publisher, 121-127, 1998.

[23] 李宝慧著. Data Mining 统计学的新进展. 中国学术期刊电子杂志社人物专访，84-86，2001

[24] 陈卫华，朱仲英. 数据挖掘在 CRM 中的应用. 微型电脑应用，17 (10):26-28，2001

[25] K. Ali, S. Manganaris and R. Srikant, Partial Classification using Association Rules, In Proceeding of 3rd International Conference on Knowledge Discovery and Data Mining, 1997.

[26] J. Nearthos, M. Rothman and M.Viveros, Applying Data Mining Techniques to

a health Insurance Information System, In Proc. of the 22th Int' 1 Conf. on Very Large Databasees, 1996.

[27] T.Rathburn, Data Mining in the Financial Markets: Data-Related Issues, PCAI, 11(4), 19, 1997.

[28] 李保东，宋汉涛. 数据挖掘在客户关系中的应用. 计算机应用研究，2002. 10

[29] R. Cooley, P.N. Tan and J. Srivastava, Discovery of International Usage Patterns from Web Data, Technical Report TR 99-022 University of Minnesota, 1999.

[30] B. Mobasher, N. Jain, E.Han and J. Srivastava, Web Mining: Pattern Discovery From World Wide Web Transaction, Department of Computer Science, University of Minnesota, Technical Report (TR96-050), 1996.

[31] www. stockstar. com 证券之星

[32] www.hxinfo. com 华夏证券资讯网

[33] 胡旭微证券组合投资的评价及选择. 中国统计，2000. 3

[34] 顾铭德长线是金——基本面股价走势分析. 四川人民出版社，1999. 2

[35] 蔡伟杰，张晓辉，朱建秋，朱扬勇. 关联规则挖掘综述. 计算机工程，5:31-49，2001

[36] 秦吉胜，宋瀚涛. 关联规则挖掘 AprioriHybird 算法的研究和改进. 计算机工程，17:7-9，2004

[37] M. J. Zaki, S. Parthasarathy and W. Li., A localized algorithm for parallel association mining, 9th Annual ACM Symposium on Parallel Algorithms and Architectures, Newport, Rhode Island, June 28-29, 1997.

[38] L A Zadeh, Fuzzy sets, Information and control, 8 (3): 338-353, 1965.

[39] 杨一文，刘贵忠等. 基于小波网络的非线性时间序列预测及其在股市中的应用. 模式识别与人工智能，14(2)，2001

[40] 李宏，陈松乔，王建新. 基于时序模式关联的股票走势分析研究. 计算机工程与应用，13:56－58，2001

[41] 张小刚，章兢，陈华. 模糊时间序列挖掘在复杂系统模糊建模中的应用. 控制理论与应用， 19(6):872－878，2002

[42] 黄河，黄轲，杭小树等. 时间序列中快速模式发现算法的研究. 计算机工程与应用，21:192－194，2003

[43] 唐能通. 短线是银之三——短线高手制胜的 54 张王牌. 四川人民出版社，2000. 2

[44] 唐能通. 短线是银之六——炒股实战真功夫. 四川人民出版社，2004. 5

[45] 只铁. 铁血战记。中国商业出版社，2002. 6

[46] 唐能通. 短线是银之四——十万到百万. 四川人民出版社，2000. 8

[47] 唐能通. 短线是银之五——头部不再套. 四川人民出版社，2001. 2

[48] 一阳. 短线必杀——职业操盘秘诀. 中国科学技术出版社，2003. 4

[49] 一阳. 短线必胜——盛世操盘宝典. 中国科学技术出版社，2004. 5

[50] 霍华德.王（Howard Wang）. 操盘 80 守则——华尔街分析师心诀. 百家出版社，2002. 2

[51] 黄栢中. 江恩理论——金融走势分析. 地震出版社，2004. 5

[52] 邵道明. 庄家克星——职业操盘手解析坐庄全过程. 经济管理出版社，2003. 6

[53] 只铁. 短线英雄. 中国科学技术出版社，2000. 2

[54] 马盈仓. 挖掘关联规则中 Apriori 算法的改进. 计算机工程与软件，11（21）：82-84，2004

[55] 陈文庆，许棠. 关联规则挖掘Apriori算法的改进与实现. 微机发展，8(15): 155-157，2005

[56] 秦吉胜，宋瀚涛. 关联规则挖掘AprioriHybird算法的研究与改进. 计算机工程，117(30):7-9，2004

[57] H. Mannila, H.Toivonen and Verkamo, Efficient Algorithms for Discovery Association Rules, Proc, AAAI Workshop Knowleage Discovery in Databases (KDD'94), Seattle, 181-192, 1994.

[58] M. J. Zaki, S. Parthasaraty, W.Li, M. Ogihara., Evaluation of Sampling for Data Mining of Association Rules, In Technical Report 617, Department of Computer Science, University of Rochester, Rechester, New York, May, 1996.

[59] J.w.Han, J. Pei and Y.w. Yn, Mining Frequent Patterns without Candidate Generation, In SIGMOD, l-12, 2000.

[60] P. Shenoy, J.Y Haritsa and S. Sudarshan, Turbo-charging Vertical Mining of Large Databases, In ACM SIGMOD Intl Conf Management of Data, 2000.

[61] 高文等. “KDD 的发展及应用”专题. 计算机世界，l-10，1998

攻读硕士学位期间发表的论文

主要发表论文

1 何云峰,杨燕.基于模糊时间序列——股票走势的建模与应用.微计算机信息.2006,12