# 数据流频繁模式挖掘及预测技术研究

陈　辉　著

江西高校出版社

**图书在版编目(CIP)数据**

数据流频繁模式挖掘及预测技术研究/陈辉著. —南昌:江西高校出版社,2010.6

ISBN 978-7-81132-932-2

Ⅰ.①数… Ⅱ.①陈…Ⅲ.①数据采集—研究
Ⅳ.①TP311.13

中国版本图书馆 CIP 数据核字(2010)第 113664 号

| | |
|---|---|
| 出版发行 | 江西高校出版社 |
| 社　　址 | 江西省南昌市洪都北大道 96 号 |
| 邮政编码 | 330046 |
| 总编室电话 | (0791)8504319 |
| 销售电话 | (0791)8513417 |
| 网　　址 | www.juacp.com |
| 印　　刷 | 南昌市光华印刷有限责任公司 |
| 照　　排 | 江西太元科技有限公司照排部 |
| 经　　销 | 各地新华书店 |
| 开　　本 | 850mm×1168mm　1/32 |
| 印　　张 | 4.875 |
| 字　　数 | 130 千字 |
| 版　　次 | 2010 年 6 月第 1 版第 1 次印刷 |
| 印　　数 | 1～200 册 |
| 书　　号 | ISBN 978-7-81132-932-2 |
| 定　　价 | 18.00 元 |

赣版权登字-07-2010-12

**图书在版编目(CIP)数据**

数据流频繁模式挖掘及预测技术研究/陈辉著. —南昌:江西高校出版社，2010.6

ISBN 978-7-81132-932-2

Ⅰ.①数… Ⅱ.①陈… Ⅲ.①数据采集-研究 Ⅳ.①TP311.13

中国版本图书馆 CIP 数据核字(2010)第 113664 号

| | |
|---|---|
| 出版发行 | 江西高校出版社 |
| 社　　址 | 江西省南昌市洪都北大道 96 号 |
| 邮政编码 | 330046 |
| 总编室电话 | (0791)8504319 |
| 销售电话 | (0791)8513417 |
| 网　　址 | www.juacp.com |
| 印　　刷 | 南昌市光华印刷有限责任公司 |
| 照　　排 | 江西太元科技有限公司照排部 |
| 经　　销 | 各地新华书店 |
| 开　　本 | 850mm×1168mm　1/32 |
| 印　　张 | 4.875 |
| 字　　数 | 130 千字 |
| 版　　次 | 2010 年 6 月第 1 版第 1 次印刷 |
| 印　　数 | 1～200 册 |
| 书　　号 | ISBN 978-7-81132-932-2 |
| 定　　价 | 18.00 元 |

赣版权登字-07-2010-12

# 前　言

在过去的几年里，数据流（Data Stream）广泛出现在传感器网络、金融证券管理、网络监控、Web 日志以及通信数据在线分析等应用领域中。由于数据流中数据的规模一般都十分庞大、且增长迅速，因此，有限的存储空间根本无法完整地保存数据流的全部数据，这给数据流的数据处理带来了巨大的挑战。此外，由于数据流数据的连续性与流动性，随着流数据连续到达，数据流所包含的知识信息总是在连续不断地变化。而对于实际的数据流应用而言，挖掘出数据流上知识的变化趋势往往比挖掘知识本身更为重要。因此，人们往往更希望挖掘出数据流上最近的某个滑动时间窗口内交易数据所包含的知识信息。

挖掘数据流上的频繁模式对于数据流的应用有着重要研究意义，例如，在网络监控中，对应于异常流量的频繁模式可能意味着存在网络攻击或者网络拥塞；在商业销售记录中，频繁模式总是反映那些热门销售的产品以及它们之间的关联关系；而在传感器网络数据管理中，挖掘其中的频繁数据集可以有助于去估计那些丢失的数据值。然而，由于流数据的特点，传统的静态数据库挖掘方法不可能直接应用流数据的频繁模式挖掘，而必须研究新的数据流频繁模式挖掘方法。此外，很多数据流应用不仅要分析数据流上当前的模式信息，还要能够预测和评估数据流未来流数据的模式信息，研究适合数据流的预测查询算法也是数据流研究中的重要工作之一。

本书分七章来研究数据流频繁模式挖掘及预测查询算法。第一章为绪论，介绍数据流产生的背景及特点，重点介绍数据流管理及数据流挖掘的国内外研究现状，最后还通过一系列的实际应用来阐述了数据流研究的应用意义。第二章介绍数据流处理模型及数据流处

理算法所必备的技术特点，归纳数据流处理的常用技术、数据流的窗口模型及常见的频繁模式挖掘方法，提出流数据频繁模式挖掘存在的技术难点，并对本书中所用到的专有名词进行定义与解释。第三章讨论一种快速挖掘数据流上最近的频繁模式的方法，该方法应用保守计算策略计算滑动时间窗口内模式的近似支持数，由于保守计算策略得到的近似支持数总大于或等于模式的真实支持数，因此，方法总能够保证模式挖掘的正确性。第四章研究一种挖掘大小可调的滑动时间窗口内频繁模式的方法，该方法应用时间衰减模型逐渐减小数据所包含模式支持数的权重，并以此来区分新产生流数据与历史流数据所包含的模式。为了保证模式挖掘的覆盖率和精度，方法还分析时间衰减模型对模式支持数的影响，并给出衰减因子在保证模式挖掘正确性条件下的临界值。当滑动时间窗口的大小改变时，仅需重新设定合适的衰减因子的值即可重新保证在新的滑动时间窗口下模式挖掘的正确性。第五章研究一种频繁模式支持度门限不定条件下挖掘滑动时间窗口内 Top - K 频繁模式的方法，该方法应用 Chernoff 边界理论估计窗口内第 K 频繁模式的支持度，并将其用于动态维护窗口内潜在频繁的模式信息；根据理论分析，Chernoff 边界理论能够为模式挖掘的正确性提供了概率保证。第六章研究一种基于马尔可夫模型的数据流预测查询算法，该方法将大小可能无限的流数据空间映射到一个有限的流数据状态空间中，从而将流数据变化序列转变成为一个流数据状态变迁序列。通过使用数据流状态变迁有向图(SSTD)维护流数据状态变迁序列的统计信息，可以得到流数据状态变迁的概率矩阵，从而应用马尔可夫模型可以去预测数据流在未来时刻的可能值。第七章总结全书的工作，并对未来研究工作进行展望。

本书由陈辉独立撰写，书中的部分实验内容得到了张琛硕士的大力相助。在写作与出版过程中，得到了江西财经大学科研处各位领导与老师的大力支持，也得到了杨兵博士、程远国博士、向军博士、陈刚博士、魏霞博士等老师和朋友的帮助，在此表示衷心的感谢。还

要感谢江西财经大学软件与通信工程学院的各位领导和老师，尤其要感谢夏家莉院长、尹爱华副院长、张弛副院长、刘细发博士、黄茂军博士等老师，感谢他们的关心和帮助。

由于作者学术水平有限，书中难免存在疏漏和错误，诚恳希望读者批评指正。

陈　辉

2010 年 4 月

# 符　号

| 序号 | 符号 | 含　义 |
|---|---|---|
| 1 | $DS$ | 数据流 |
| 2 | $N_{DS}$ | 数据流的大小 |
| 3 | $SW$ | 数据流滑动时间窗口 |
| 4 | $N$ | 数据流滑动时间窗口的大小 |
| 5 | $A$ | 数据流数据项的集合 |
| 6 | $I$ | 模式 |
| 7 | $l$ | 模式 $I$ 中所包含数据项的个数 |
| 8 | $I_p^l$ | 模式 $I$ 中长度为 $l$ 的前缀模式 |
| 9 | $T$ | 流数据，也称事务 |
| 10 | $L$ | 流数据 $T$ 中所包含数据项的个数 |
| 11 | $T_i$ | 数据流上第 $i^{th}$ 到达的流数据 |
| 12 | $T_c$ | 当前流数据 |
| 13 | $\prec$ | 一种确定的全序排列关系 |
| 14 | $T'$ | 事务 $T$ 按照全序 $\prec$ 关系的投影 |
| 15 | $\theta$ | 最小支持度门限 |
| 16 | $\varepsilon$ | 最大许可误差 |
| 17 | $sup(I)$ | 模式 $I$ 的支持数 |
| 18 | $freq(I)$ | 模式 $I$ 的支持度 |
| 19 | $TDM$ | 时间衰减模型 |

| 序号 | 符号 | 含　义 |
|---|---|---|
| 20 | $f$ | 衰减因子 |
| 21 | $sup_d(I)$ | 模式 $I$ 衰减后的支持数 |
| 22 | $freq_d(I)$ | 模式 $I$ 衰减后的支持度 |
| 23 | $DW$ | 数据窗口 |
| 24 | $CDW$ | 当前的数据窗口 |
| 25 | $PDW$ | 当前数据窗口之前的那个数据窗口 |
| 26 | $RFP-tree$ | 数据流上最近的频繁模式树 |
| 27 | $SW-tree$ | 数据流上的滑动时间窗口树 |
| 28 | $SWTP-tree$ | 数据流滑动时间窗口内的 $Top-k$ 频繁模式树 |
| 29 | $node$ | 数据流频繁模式树节点 |
| 30 | $IIT$ | 数据项索引表 |
| 31 | $e$ | 数据项索引表表项 |
| 32 | $Root$ | 数据流频繁模式树根结点 |
| 33 | $\delta$ | 可信度参数，为大于 0 小于 1 的数 |
| 34 | $sup_k$ | 数据流滑动时间窗口内第 $k^{th}$ 频繁模式的支持数 |
| 35 | $pr$ | 概率 |
| 36 | $p$ | 概率矩阵 |
| 37 | $SSTD$ | 统计状态变迁图 |
| 38 | $R$ | 实数集 |
| 39 | $N$ | 整数集 |

# 1 绪 论

## 1.1 数据流产生背景

自20世纪70年代以来，数据库技术得到了迅速的发展和广泛的应用，传统数据库获得了巨大的成功。但是到了20世纪末，随着信息技术的飞速发展，一种新的称为数据流(data stream)[1]的应用模型却对它提出了有力的挑战。这种应用模型广泛出现在众多应用领域中，例如金融数据管理、网络监控、通信数据管理、传感器网络等。这类数据的共同特点是：数据连续不断地产生且在时间维度上严格有序，数值不断地变化，数据的规模一般都很庞大且增长迅速[1,2]。利用传统的数据库技术管理这种数据，需要将数据全部保存在物理介质中，然后再进行各类操作。但是，由于数据流数据源源不断，在有限的存储空间中无法保存数据流的全部数据；而且，数据流应用对数据处理的实时性往往要求很高，这也决定了不可能将数据完全地保存下来，然后再进行处理。因此，现有的数据库技术无法有效处理数据流数据，必须研究新的数据流处理技术[3]。

数据挖掘、联机分析处理、内存数据库、实时数据库、主动查询处理技术和数据库近似查询等技术是数据库领域中当前最为活跃的研究方向。这些技术大都依然以传统数据库为研究基点，适用于持久稳固的数据存储。在数据库管理系统中，插入、更新、删除等操作都没有查询发生频繁，查询结果反映了当前的数据库状态。但是这样的存储方式以及对基于时间的数据非常有限的管理能力无法满足数据流应用的需求。数据流系统中的数据有量大、快速和时变的特点，所以不能仅仅采用传统方式来处理它们。简单地将数据放到传统的

数据库中并对其进行操作是不切实际的，因为大量的数据会造成数据库无法正常使用，而且大部分数据可能很快就会被删除，不需要永久保存，数据更新和查询的效率也非常低，因此如何对数据流进行有效的管理与挖掘是当前亟待解决的问题。

自2000年以来，数据流管理技术已经引起了数据库界的广泛关注。许多著名的研究机构和大学都开始了这方面的研究，提出了一系列的理论、方法和技术，有力地推动了数据流管理技术的发展。目前的研究工作主要包括两个方面：数据流管理系统（Data Stream Management System，DSMS）[1,4,5]和数据流挖掘。前者侧重于数据流管理系统的开发和相关技术的实现，如数据流的连续查询、内存管理和系统调度等；后者侧重于数据流的在线分析，从流数据聚类[6-8]、流数据分类[9-11]、频繁项计数[12]、频繁模式[13-16]挖掘、序列模式挖掘[17,18]、关联规则挖掘[19]、趋势分析[20,21]、流数据预测[22,23]与数据流变化检测[24-26]等方面的研究。

## 1.2 数据流的特点

数据流可以看作是连续的数据序列，其中的数据元素按照一定的时间顺序依次到达，并且暂时驻留在内存缓冲区中，所有的数据处理工作都在这段时间内完成。当内存缓冲区填满之后，随着新数据的到达，内存缓冲区中最陈旧的数据将被删除而不可再访问。因此，数据流形式的数据不同于传统的基于集合的相对静止的数据。数据流中的数据是连续变化的，数据量的多少事前是不确定的，也可能是无限的。归纳起来，数据流的共同特点是：

（1）无限性

数据流数据连续不断地到达，若要将其全部存储，需要的存储空间可能是无限的。

（2）未知性

数据流数据的值是不断改变的,即使是利用预测方法也不可能准确地预测下一时刻将到达数据流数据的值。

(3)不可再现性

对于数据流上的数据,一旦流过处理节点就不会再次出现。

## 1.3 数据流管理系统现状

随着网络与信息技术的发展,许多应用领域都出现了大量的流数据,为了管理这些数据并从中获取有用的知识信息,用户迫切需要研究能够对高速流数据进行有效管理的系统,即数据流管理系统(Data Stream Manage System,DSMS)。如表 1.1 所示,数据流管理系统与传统的数据库管理系统存在重要的区别,它能够提供对高速的数据流进行实时连续地查询、特征提取、聚类分析以及实时监控等功能。

表 1.1 数据流管理系统与数据库管理系统的比较

| 数据流管理系统(DSMS) | 数据库管理系统(DBMS) |
|---|---|
| 时变的、连续的数据 | 稳定的数据 |
| 连续查询 | 一次性查询 |
| 顺序存取 | 随即存取 |
| 有限的内存空间 | 海量的磁盘存储空间 |
| 历史数据和当前数据都重要 | 当前数据比较重要 |
| 实时性、随时性 | 一般无实时性要求 |
| 原始数据的概念层次较低且具有多维性 | 数据可具有多个概念层次 |
| 近似的处理结果 | 精确的处理结果 |

在国外,许多大学和科研机构已对数据流管理系统开展了大量的研究工作;而在国内,针对这方面的研发却比较少。表 1.2 中列举了目前已经取得的数据流管理原型的主要研究成果。下面,我们对

其中的几个进行简要的介绍。文献[1]介绍了斯坦福大学研制的STREAM(Stanford Stream Data Manager)系统,该系统是一个以关系为基础的数据流管理系统,主要提供了内存管理和近似查询等功能;可以处理快速的、易变的、大量涌入的数据流信息,具有较好的连续查询能力;具有对连续的数据流监控和优化功能,能够根据应用环境需求,合理地进行资源配置。文献[27]介绍了施乐公司的 Tapestry 项目,该项目研发了 Tapestry 数据流管理系统,该系统支持连续查询处理,并且其查询语言支持 SQL 语言的子集,能够进行增量式的有效查询处理。

表 1.2 数据流管理的原型系统

| 系统名称 | 研发机构 | 应用领域 |
|---|---|---|
| Stream | Stanford University | General - purpose DSMS |
| Cougar | Comell University | Sensor Networks |
| Aurora | MIT/Brown/Brandies University | Surveillance |
| Giga Scope/Hancock | AT&T | Telecom Data Streams |
| Niagara | University of Wisconsin - Madision | Internet XML Database |
| OpenCQ | Georgia Tech | Triggers, Incr View Maintenance |
| Tapestry | Xerox | Pub/Sub Content - based Filtering |
| Telegraph(CQ) | U.C.Berkeley | Adaptive Enginefor Sensor Networks |
| TinyDB | U.C.Berkeley | Sensor Networks |
| Tradebot | www.tradebot.com | Stock Tickers & Streams |
| Tribeca | Bellcore | Network Monitoring |
| Medusa | Brown University | Distributed Stream Processing |

| 系统名称 | 研发机构 | 应用领域 |
| --- | --- | --- |
| StatStream | New York University | Statistical Monitoring |
| Streaminer | UIUC | New Projectfor Stream Data Mining |

文献[28]介绍了加州大学伯克力分校研发的 TelegrpahCQ 系统，该系统用于处理连续的数据流，它主要用于处理对数据流进行大量连续查询时产生的查询流。文献[29]介绍了布朗大学和麻省理工学院合作研发的 Aurora 项目，其主要目的是构建一个新型的数据处理系统，以专门用于对数据流的监控处理。Aurora 系统可以满足三种不同的应用需求，即实时监控、流数据历史信息的存储管理和针对历史数据以及当前数据的综合应用。

## 1.4 数据流挖掘技术现状

数据流挖掘技术的研究成果主要集中在数据流的聚类、分类、预测、频繁模式挖掘或者关联规则挖掘等方面。

### 1.4.1 数据流聚类技术现状

尽管聚类问题在数据库、数据挖掘和统计等领域得到了广泛研究，但是，流数据的特点仍为聚类算法提出了前所未有的挑战。由于完整甚至部分地存储数据流上历史数据的方法是不可行的，因此，需要设计能够只根据新产生的流数据就能够追踪聚类变化的算法。这要求数据流聚类算法必须是增量式的[30,31]，聚类的表示要简洁，新数据的处理速度要快，且对噪音和异常数据是稳健的。由于数据流可看成是随时间不断变化的无限过程，其隐含的聚类可能随时间动态变化而导致聚类质量降低。近年来，有学者提出了应用于大规模数据集的一趟聚类算法，如 Squeezer 算法[32]和 BIRCH[33]算法，它们可以应用于解决某些数据流问题。此外，也有学者提出了针对流数据的聚类算法，典型的有 STREAM 算法[6,34]和 CluStream 算法[7]。下

面，我们对这些算法进行简要地介绍。

（1）最小距离原则聚类算法

文献[32，35]分别提出了针对分类属性数据和数值属性数据的最大相似度或最小距离原则的聚类算法，不需要聚类个数的先验知识，扫描数据集一趟即将数据分割为半径几乎相同的超球体。所不同的是：Squeezer 算法[32]采用不同属性值的取值频度来表示类，而文献[35]使用质心来表示类。

（2）BIRCH 算法

BIRCH 算法[33]试图利用有限的资源来生成最好的聚类结果，尽可能减少 I/O 请求。BIRCH 算法采用聚类特征树（CFTree）来表示聚类，CF 树是高度平衡树，采用分层数据结构存储聚类特征。聚类特征 CF（ClusterFeature）是聚类信息的三元组：CF =（N，LS，SS），这里 N 是类中对象个数，LS、SS 分别是这 N 个对象的属性值之和与平方和，用于计算属性均值和方差。CF 树的大小由两个参数确定：分支因子 B 和阈值 T，分支因子定义了每个非叶节点孩子的最大数目，而阈值给出了叶节点中聚类的最大直径。BIRCH 算法包括两个阶段：第一阶段扫描数据库，建立初始存于内存的 CF 树；第二阶段采用某个聚类算法对 CF 树的叶节点进行聚类，以进一步改进聚类质量。由于 CF 树的每个节点只能包含有限数目的条目，节点并不总是对应于自然聚类，而且，由于 BIRCH 算法用直径的概念控制聚类的边界，如果聚类的边界不是球形的，BIRCH 算法则不能很好地工作。

（3）STREAM 算法

STREAM 算法[6，34]采用基于 K - 均值的思想，使用质心和权值（类中数据个数）表示聚类。它采用批处理的方式对流数据进行聚类处理，对于每一批数据 Bi，STREAM 算法对其进行聚类，得到加权的聚类质心集 Ci。在处理每一批数据进行聚类时，STREAM 算法采用分级聚类的方法，首先对最初的 $m$ 个输入数据进行聚类得到 $O(k)$

个一级带权质心，然后将上述过程重复$\frac{m}{O(k)}$次，得到 $m$ 个一级带权质心，然后对这 $m$ 个一级带权质心再进行聚类得到 $O(k)$个二级带权质心；同理，每当得到 $m$ 个 $i$ 级带权质心时，就对这些质心进行一次聚类得到 $O(k)$个 $i+1$ 级带权质心；重复这一过程直到得到最终的$O(k)$个质心。对于每个第 $i+1$ 级带权质心而言，其权值是与它对应的 $i$ 级质心的权值之和。

(4)CluStream 算法

C.Aggarwal，J.Han 等人在文献[7]中提出了流数据聚类算法 CluStream，该算法首次把数据流看成一个随时间变化的过程，而不是一个整体进行聚类分析。该算法有很好的可扩展性，可产生高质量的聚类结果，尤其是在数据流随时间变化较大时，可产生比其它算法质量更高的聚类。CluStream 算法不仅能给出整个数据流聚类的结果，还可以给出任意时间范围内的聚类结果。该算法由在线和离线两部分构成，在线部分用 micro - cluster 定时存储数据流的摘要信息，对数据的处理和更新是增量式的，离线部分 macro - cluster 通过对在线部分保存的中间结果的再处理得到用户感兴趣的不同时间范围内数据流的聚类结果。为了既体现数据流进化的过程又不消耗过多的存储空间，C.Aggarwal 等人提出了倾斜时间窗口[36]的概念，用不同的时间粒度对数据流信息进行存储和处理，最近的数据变化以较细的时间粒度刻画，而离现在较远的数据以较粗的时间粒度刻画。

### 1.4.2 数据流分类技术现状

数据流分类算法的研究成果主要体现在文献[9，10]中，P.Domingos 等在文献[9]中提出了一种改进的 Hoeffding 决策树分类算法 VFDT(Very Fast Decision Tree)，使用恒定的内存大小和时间处理每个样本，有效地解决了时间、内存和样本对数据挖掘，特别是在高速数据流上数据挖掘的限制。VFDT 使用信息熵选择属性，通过建立 Hoeffding 树来进行决策支持，并使用 Hoeffding 约束来保证高精度地

处理高速数据流;既可连续处理数据,也可通过二次抽样,重新扫描数据集,因此可以处理非常庞大的数据集。VFDT 的另一个特点是增量式学习及随时可用性,它是一个实时算法,在学习了最初的一些样本之后,就提供了一个随时可用、不断优化的决策树。

VFDT 和其他大多数机器学习方法一样,假设数据是从静态分布中随机获取的,不能反映数据随时间变化的趋势。因此,G.Hulten 等在文献[10]中引入了滑动窗口[37,38]技术,并对 VFDT 算法进行改进,提出了 CVFDT(Concept adapting Very Fast Decision Tree)算法。该算法除了保留 VFDT 算法在速度和精度方面的优点外,还增加了对数据产生过程中变化趋势的检测和响应,使得算法更好地适应对高速时变流数据的分类。CVFDT 利用样本窗口来有效维护决策树的更新和模式的一致性,然而它并不需要在每个新样本到来时就学习新的模式,而是通过增加相应新样本的总数来更新充分统计量,相应地减少滑动窗口中旧的节点数量。如果基本概念是静态的,将不会产生影响;然而,如果概念是动态变化的,由于一个可替换的属性具有更高的增益,使得某些先前通过 Hoeffding 测试的分支不再起作用。在这种情况下,CVFDT 开始在根节点用新的更好属性生成一棵用于替换的子树,当这棵替换的子树在新的数据上比旧的子树更精确时,就用新的子树代替旧的子树。

### 1.4.3 数据流频繁模式挖掘技术现状

在传统的静态数据库挖掘中,频繁模式挖掘技术[39-41]得到了广泛的研究,并取得了很多成果,其中的经典算法包括:Apriori[42]、FP-growth[43]、CLOSET[44]、CHARM[45]等。但是,这些算法难以增量式地处理流数据,不适合数据流挖掘。因为在这些经典的算法中,挖掘频繁模式的工作是建立在得到数据库全部数据的基础上,这使得在数据流环境中挖掘和更新频率模式变得十分困难。与对静态数据集的频繁模式挖掘相比,数据流上的频繁模式挖掘需要追踪更多信

息和处理更复杂的情况，例如，随着时间的推移，数据流上的频繁模式集会变化，并且，非频繁模式在未来时刻可能变为频繁模式。此外，在很多情况下，挖掘频繁模式集的变化趋势往往比挖掘频繁模式本身更令人感兴趣。因此，在数据流频繁模式挖掘中，人们更希望找出频繁模式随时间变化或进化的情况。

文献[36]提出了基于 FP - tree 模型的 FP - stream 算法来挖掘数据流上的频繁模式。算法采用批处理的方式分段处理数据流上的数据，当一个数据分段内的数据全部到达之后，算法首先使用 FP - tree 方法挖掘出该数据分段内的频繁模式，然后把它们增量更新到数据流全局的频繁模式树上。此外，该方法还应用倾斜时间窗口策略以精细的时间粒度保存数据流上最近的频繁模式信息，而以粗糙的时间粒度保存历史的频繁模式。

文献[46]提出了一种利用有限存储空间通过单遍扫描流数据来估计数据流中最大频繁项集的算法，该算法采用了称为“COUNTS-KETCH”的数据结构，使得可以可靠地估计数据流中频繁项集的频率。

刘学军等提出了一种挖掘数据流频繁模式方法：FP - DS 算法[47]，该方法针对数据流的特点，在借鉴 FP - growth 算法的基础上采用数据分段的思想，逐段挖掘频繁项集，用户既可以在线连续获得当前的频繁项集，又可以有效地挖掘所有的频繁项集。算法通过引入误差 $\varepsilon$，裁减大量的非频繁项集，可以大大减少算法所需的内存空间，同时，还能保证整个频繁模式集中模式支持度的误差不超过 $\varepsilon$。

刘学军等还提出了一种发现滑动窗口中频繁闭合模式的方法：DSC - FI 算法[48]，该方法将滑动窗口分割为若干个基本窗口，以基本窗口为更新单位，利用已有的频繁闭合模式挖掘算法计算每个基本窗口的潜在频繁闭合项集，将它们及其子集存储到一种能够增量更新的数据结构 DSC - FI 树中，利用 DSC - FI 树可以快速地挖掘滑