博士文库

数据流上频繁模式和高效用模式挖掘

Frequent Pattern & High Utility Pattern Mining Over Data Streams

王 乐◎著

知识产权出版社
全国百佳图书出版单位

博士文库

数据流上频繁模式和高效用模式挖掘

Frequent Pattern & High Utility Pattern Mining Over Data Streams

王 乐◎著

知识产权出版社
全国百佳图书出版单位

图书在版编目(CIP)数据

数据流上频繁模式和高效用模式挖掘 / 王乐著. —北京：
知识产权出版社，2014.9
ISBN 978-7-5130-2982-7

Ⅰ.①数… Ⅱ.①王… Ⅲ.①数据采集 Ⅳ.①TP274

中国版本图书馆CIP数据核字(2014)第209812号

内容提要

本书以数据流上的频繁模式和高效用模式挖掘计算为背景，介绍该领域相关的概念、理论及近年来相关的最新研究成果，内容包括传统数据集中的频繁模式挖掘及其大数据集下的频繁模式挖掘算法、不确定数据流中的频繁模式挖掘算法、具有效用值的数据流中的高效用模式挖掘算法。

本书可作为经济学、统计学、管理科学与工程、计算机科学与技术等学科高年级的本科生和研究生的参考用书，也可供商务数据挖掘、金融数据分析等相关研究人员参考。

责任编辑：吴晓涛　　**责任出版**：谷　洋

数据流上频繁模式和高效用模式挖掘

SHUJULIU SHANG PINFAN MOSHI HE GAOXIAOYONG MOSHI WAJUE

王 乐　著

出版发行：知识产权出版社有限责任公司		**网　　址**：http://www.ipph.cn http://www.laichushu.com	
电　　话：010-82004826			
社　　址：北京市海淀区马甸南村1号		**邮　　编**：100088	
责编电话：010-82000860转8533		**责编邮箱**：sherrywt@126.com	
发行电话：010-82000860转8101/8029		**发行传真**：010-82000893/82003279	
印　　刷：北京中献拓方科技发展有限公司		**经　　销**：各大网上书店、新华书店及相关专业书店	
开　　本：720mm×1000mm　1/16		**印　　张**：9.5	
版　　次：2014年9月第1版		**印　　次**：2014年9月第1次印刷	
字　　数：150千字		**定　　价**：28.00元	

ISBN 978-7-5130-2982-7

前　言

数据和信息正以前所未有的速度增长。正如Kevin Kelly在著名的*What Technology Wants*里面提到的那样，人类几百万年的基因变异，平均速度大约是每年1bit；而现在信息社会每年新增的信息量为400艾（exa，IE=10^{18}），即人类1s内处理数据的总量，等于我们的DNA用10亿年处理的数据量。在这样的滔天数据洪流面前，如何及时地对已产生的数据进行挖掘和分析，从中提取我们关心的、与企业产能和效益有密切关系的潜在信息，是信息时代的企业需要特别关注的问题；其中一个重要的方面，就是对关联关系（频繁模式）和高效用模式的挖掘。

由于数据流具有海量性、实时性和动态变化性的特点，这就要求数据流上的挖掘算法有较高的时空效率。尽管数据流上模式挖掘技术取得了一定的进展，但是挖掘算法的时空效率仍然是当前数据挖掘领域中的研究焦点之一。

本书以数据流上的频繁模式和高效用模式挖掘计算为背景，介绍该领域相关的概念、理论及近年来相关的最新研究成果，内容包括传统数据集中的频繁模式挖掘及其大数据集下的频繁模式挖掘算法、不确定数据流中的频繁模式挖掘算法、具有效用值的数据流中的高效用模式挖掘算法，以及包含相应静态数据集中的挖掘算法。全书共分为五章：第1章首先对已有的频繁模式和高效用模式挖掘算法进行了回顾，详细地介绍了算法Apriori和FP-Growth等；第2章探讨传统的动态数据中的频繁模式挖掘算法；第3章首先探讨不确定静态数据上的频繁模式挖掘算法，然后探讨了不确定数据流中的频繁模式挖掘算法；第4章探讨静态数据集上的高效用模式挖掘算法，然后基于静态数据集上的挖掘算法，介绍数据流中的高效用模式挖掘算法；第5章以传统数据集为例，介绍了MapReduce框架下的频繁模式挖掘算法。各章内容相对独立又相互联系，较

为系统地阐述了数据流中几种模式挖掘算法的研究现状。

本书主要内容为作者在攻读博士学位期间的研究成果，其中部分工作得到国家自然科学基金项目“大数据环境下高维数据流挖掘算法及应用研究”（61370200）、宁波市自然科学基金项目“面向大数据的高频金融时间序列高效用时态频繁模式挖掘研究”（2013A610115）和“多重不确定数据流上模式挖掘的建模及算法研究”（2014A610073）等项目的支持，并得到宁波大红鹰学院优秀博士计划资助。书稿的撰写过程中，大连理工大学的冯林教授、杨元生教授、金博博士等老师给予了大力支持和热心指导，同时也得到姚远、刘胜蓝、张晶、姜玫、吴明飞、王辉兵、蔡磊等同学的关心和合作，在此一并感谢！

作者

2014年7月于宁波大红鹰学院

主要符号表

符　号	含　义	单　位
t	事务	
minSup	最小支持度	%
minSN	最小支持数	
minExpSup	最小期望支持度	%
minExpSN	最小期望支持数	
minUT	最小效用阈值	%
minUti	最小效用值	
w	窗口宽度	批
p	每批数据中事务项集个数	个

目 录

第1章 绪 论

1.1 背景和意义

智能终端、互联网及无线传感网络的发展将我们带入了一个数据的时代，据市场研究公司Strategy Analytics的分析师预测称：在未来5年内，全球移动用户基数将增加到89亿；中国三家电信运营商的各省份公司也都在构建着自己的数据仓库，而这些数据仓库的总体规模已达到数十PB的水平；腾讯微博每天约有4000万条微博信息；YouTube每月上传的视频近100万h。此外，传感器网络、移动网络、电子邮件、社会网络以及生物信息等领域每天都会产生海量数据，在此推动下，数据流成为未来数据发展的一个主要趋势，而从数据流中挖掘有用的知识得到广泛的重视。

数据挖掘（Data Mining，DM）是从大量的、不完全的、有噪声的、模糊的、随机的数据中提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。当积累的数据越来越多，如何从积累的数据中提取有用的知识成为很多行业的当务之急。数据挖掘的技术主要有关联规则挖掘、聚类分析、分类、预测、时序模式和偏差分析等。

自从数据挖掘技术出现以来，关联规则挖掘一直是数据挖掘领域中的一个最基本和最重要的研究方向。关联规则挖掘的重要工作就是挖掘频繁项集（频繁模式），因此关联规则挖掘也常常称为频繁模式挖掘。根据处理的事务数据集的类型不同，存在传统数据集上的频繁模式挖掘、不确定数据集上的频繁模式挖掘和具有内外部效用值数据集中的高效用模式挖掘等。传统的数据集仅仅考虑了事务项集中的项是否出现，而没有考虑事务项集中的项集效用值；高效用模式挖掘将事务项集中的效用值也考虑到模式的挖掘模型中；不确定事务数据

集中的频繁模式挖掘考虑了事务项集中项对应值的不确定性。以上不同类型中的模式挖掘已被广泛应用在商业、企业、过程控制、政府部门及科学研究等领域。如在移动通信数据中，可以通过频繁模式挖掘出高消费客户群的消费规则、不同客户群之间的关系、增值较高的业务组合、客户的消费推荐等；在关联规则产生的过程中，可以同时利用频繁模式和高效用模式来产生利润最大的规则。另外频繁模式挖掘也被扩展到了聚类、分类、预测、序列模式、异常检测等其他数据挖掘技术中。

本书分别对传统数据流、不确定数据流中的频繁模式挖掘算法及数据流中高效用模式挖掘算法进行了分析与研究，分别介绍新的挖掘算法或者对已有算法的改进算法；同时本书也对大数据集中的频繁模式挖掘算法进行了分析与研究，并介绍基于MapReduce并行框架的大数据的频繁模式挖掘算法。

1.2 国内外研究现状

由于静态数据集和动态数据流的数据特征不同，从中挖掘频繁模式或高效用模式的算法也有所不同；但是静态数据集的挖掘算法是动态数据流中挖掘算法的基础，本节分别从静态数据集、动态数据流两方面介绍频繁模式和高效用模式挖掘算法的研究现状。

1.2.1 传统数据集中频繁模式挖掘算法的研究

1. 静态数据集上频繁模式挖掘算法

Agrawal等[1, 2]首先提出了频繁模式挖掘问题的原始算法，并给出了著名的Apriori算法。该算法的主要理论依据是频繁项集的两个基本性质：①频繁项集的所有非空子集都是频繁项集；②非频繁项集的超集都是非频繁项集。算法Apriori首先产生频繁1-项集L_1，然后利用频繁1-项集产生频繁2-项集L_2，直到有某个r值使得L_r为空为止。在第k次循环中，算法先产生候选k-项集的集合C_k，C_k中每一个项集是用两个只有一项不同的L_{k-1}中进行并集产生的。C_k中的项集是用来产生频繁k-项集的候选项集，即频繁项集L_k是C_k的一个子集。C_k中的每个项集需要统计在数据集中的个数，从而来决定其是否加入L_k，即需要扫描一遍数据集来计算C_k中的每个项集的支持度。

Apriori是首次提出采用逐层挖掘的算法，并且是逐层挖掘算法中的代表算法，之后的很多算法都是在此基础上进行改进，如算法DHP[3]采用Hash技术来优化Apriori算法中候选项集的产生过程。Cheung等[4]采用并行方式对Apriori算法进行改进，将数据集划分为多个小数据块，在每次迭代产生频繁项集过程中，首先并行计算所有候选项集在各个数据块的支持数，然后汇总每个候选项集的总支持数（即可从候选项集中找到频繁项集），最后再利用当前层产生的频繁项集来产生新的候选项集来进行下次的迭代。算法Apriori的主要缺点：①扫描数据集的次数至少等于最长的频繁项集的长度；②需要维护算法过程中产生的候选项集（中间结果）。

算法Apriori在挖掘过程中产生了大量的候选项集，并且需要反复扫描数据集，严重影响了算法效率。为此，Han等人提出了一种无须产生候选项集的算法FP-Growth [5]，该算法只需要扫描数据集两次：第一次扫描数据集得到频繁1-项集；第二次扫描数据集时，利用频繁1-项集来过滤数据集中的非频繁项，同时生成FP-Tree，然后在FP-Tree上执行递归算法，挖掘所有的频繁项集。实验分析表明FP-Growth算法比Apriori算法快一个数量级。

算法FP-Growth是采用模式增长的方式直接生成频繁模式，该算法是模式增长算法中的典型算法，同时也是第一个模式增长算法，之后提出的模式增长挖掘算法[6-18]都是在此基础上进行改进的，包括不确定数据集中频繁模式挖掘和高效用模式挖掘中的很多算法。

算法COFI[19]不需要递归的构建子树，该算法通过项集枚举的方法来挖掘频繁模式；在稀疏数据集上，该算法的时间和空间效率优于算法FP-Growth，但在处理稠密数据集或长事务数据集的时候，该算法的处理效率比较低。

之后提出了很多频繁项集挖掘算法[20-30]，包含完全频繁项集挖掘[5, 19, 21, 22, 24, 26, 31-35]、闭项集挖掘[20, 28, 36]和最大频繁项集挖掘[23, 25, 27, 29, 37]；其中国内在这领域研究也有较大的进展[7, 38-74]，包括TOP-K频繁项集挖掘[75-77]、负关联规则挖掘算法[78, 79]等。

2. 数据流中频繁项集挖掘算法

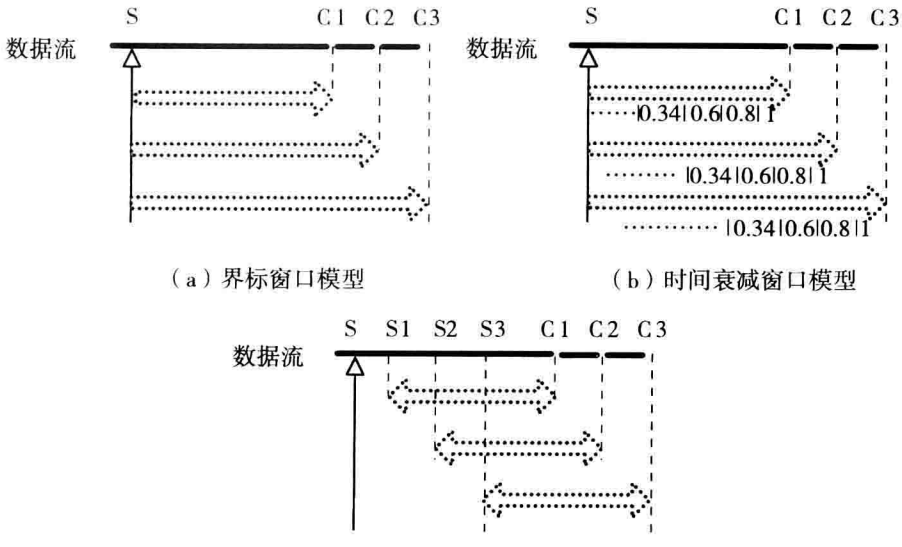
和静态数据集相比，动态数据流上有更多的信息需要跟踪，如以前频繁模式后来变为非频繁项集，或以前非频繁模式后来变为频繁模式；另外，由于数据的流动性，当前内存中维护的数据要不断地调整。数据流中的频繁模式挖掘

算法一般采用窗口方法获取当前用户关注的数据；然后基于已有的静态数据集上的频繁模式挖掘算法，提出可以挖掘数据流中被关注数据的算法。目前存在3种典型的窗口模型[80]：界标窗口模型（Landmark Window Model）、时间衰减窗口模型（Damped Window Model）和滑动窗口模型（Sliding Window Model）。

界标窗口模型中的窗口指特定一时间点（或数据流中一条特定的数据）到当前时间（或当前条数据）之间的数据，界标窗口模型如图1.1（a）所示，在C1、C2和C3时刻，窗口中的数据分别包含了从S点到C1点、C2点和C3点之间的数据。文献［81-85］中频繁项集挖掘算法都是基于界标窗口，文献［82］提出算法DSM-FI（Data Stream Mining for Frequent Itemsets）是基于界标窗口，它以数据开始点为界标点，该算法有三个重要特征：①整个挖掘过程只需要一遍数据集扫描；②扩展前缀树存储挖掘的模式；③自上而下的方式挖掘频繁项集。文献［83］提出一个基于界标窗口的频繁闭项集挖掘算法FP-CDS，该算法将一个界标窗口划分为多个基本窗口，每个基本窗口作为一个更新单元（每个基本窗口中的数据也可以称为一批数据）：首先从每个基本窗口中挖掘出潜在的频繁闭项集，同时存储在FP-CDS树上，最终从FP-CDS树上挖掘出所有的频繁闭项集。文献［84］提出一个近似算法Lossy Counting，该算法以批为处理单元，每来一批就更新一次已有频繁项集的支持数，频繁的项集被保留下来，不频繁的被删除，同时也将当前批中新的频繁项集保留下来。

时间衰减窗口模型和界标窗口模型所包含的数据是相同的，只是衰减窗口中的每条数据有不同的权重，距离当前时间越近，数据的权重越大，如图1.1（b）所示；实际上，时间衰减窗口模型是界标窗口模型的一个特例。文献［86-90］中算法都是基于时间衰减窗口模型。文献［86］提出一个基于衰减窗口模型的近似算法，该算法用一个树结构FP-stream来存储两类项集：频繁项集和潜在频繁项集。当新来一批数据的时候，更新树结构上这两类项集的支持数，如果更新后的项集既不是频繁项集，也不是潜在频繁项集，则将这类项集从树上删除；同时新来一批数据中新产生的频繁项集或潜在频繁项集也要存储在这个树结构上。文献［87］引入一个时间衰减的函数来计算项集支持数以及总的事务支持数。文献［88］采用固定的衰减值，当新来一个事务项集的时候，已有的

频繁项集的支持数都乘以固定的衰减值，如果新来的事务包含某一频繁项集，则该项集的支持数再加上1。



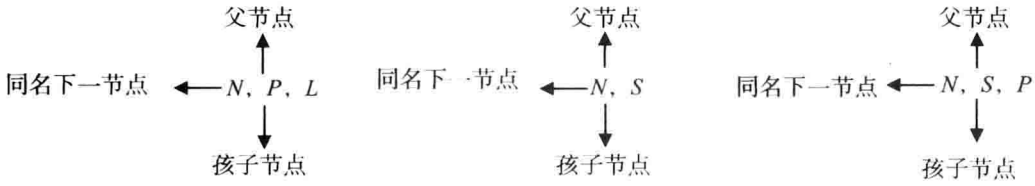
（a）界标窗口模型　　（b）时间衰减窗口模型

（c）滑动窗口模型

图1.1　三种窗口模型

S—数据流中指定的起点；C1，C2，C3—3个不同的处理点；
S1，S2，S3—3个不同的起点。

滑动窗口模型中当前处理数据的个数固定，或者是当前处理数据的时间段长度固定，如图1.1（c）所示。基于滑动窗口模型的频繁项集挖掘算法研究比较多[91-100]。文献［92］提出一个挖掘算法DST，该算法指定一个窗口中有固定批的数据，并且每批数据中事务个数也是固定的，每新到一批数据就更新一次窗口；DST将窗口中事务数据保存到树DST-Tree上，DST-Tree的节点结构如图1.2（a）所示，节点上的P字段记录节点当前窗口中每批数据中的支持数，同时每个节点还用L字段记录更新该节点的最后批次；每新来一批数据，将新到的数据添加到树DST-Tree上，同时修改相应节点上的P值和L值。算法DST在挖掘窗口中频繁项集之前，先把树上不再有用的节点（垃圾节点）从树上删除，然后用算法FP-Growth挖掘每个窗口中的频繁项集。文献［92］的作者后来又提出一个算法DSP[93]，算法DSP和DST的主要区别是DSP采用COFI来挖掘窗口中的频繁项集，而DST是用FP-Growth挖掘窗口中的频繁项集；算法DSP存储事务项集的树结构和DST的相同。



（a）DST-Tree上的节点结构 （b）CPS-Tree上的一般节点的结构 （c）CPS-Tree上的尾节点的结构

图1.2 树DST-Tree和CPS-Tree的节点结构

N—节点名；*S*—总支持数；*L*—最后更新批次；

P—pane-counter $[V_1, V_2, \cdots, V_w]$（V_i—第i批中的支持数；w—窗口中的总批数）。

文献［95］提出一个基于滑动窗口的频繁项集挖掘算法CPS，在算法CPS中，每个窗口是由固定批的数据组成，以批为单位更新窗口中数据，该算法将窗口中事务项集保存到一棵CPS-Tree树上，树CPS-Tree上有两类节点：正常节点（normal node）和尾节点（tail-node），正常节点上只记录该节点总的支持数*S*，如图1.2（b）所示；而尾节点要记录总的支持数*S*，同时还用一个数组*P*记录一个节点当前窗口中每批数据上的支持数，如图1.2（c）所示。每新来一批数据，该算法会将树中的垃圾节点删除，采用FP-Growth算法挖掘每个窗口中的频繁项集。

1.2.2 不确定数据集中的频繁模式挖掘算法的研究

随着数据挖掘技术的广泛应用、数据采集中的不确定性和误差性等原因，现实中会产生很多不确定的数据，例如一个病人在问诊中，往往并不能根据病人的症状而被百分之百地确诊为某一病；通过RFID或者GPS获取的目标位置都有误差[101, 102]；用商业网站或历史数据中挖掘到的购物习惯来预测某些顾客下一步购买的商品都存在一定的不确定性。表1.1是一个不确定数据集的例子，每个事务表示某一顾客最近要买哪些商品以及买这些商品的可能性（概率）。因此随着不确定数据在很多领域的产生，对该类数据进行挖掘分析又成为数据挖掘领域一个新的研究问题[103-124]。由于不确定数据集和传统数据集的数据结构不同，并且两类数据集上的频繁模式挖掘模型也不同，因此不能用传统数据集中的算法来挖掘不确定数据集中的频繁模式。本节根据数据集的静态和动态特性，分别描述不确定数据集上频繁模式挖掘算法的研究现状。

表 1.1 不确定数据集

事务	事务项集
t_1	(a : 0.8), (b : 0.7), (d : 0.9), (f : 0.5)
t_2	(c : 0.8), (d : 0.85), (e : 0.4)
t_3	(c : 0.85), (d : 0.6), (e : 0.6)
⋮	⋮

1. 静态数据集

不确定事务数据集的频繁项集挖掘算法主要分为逐层挖掘（level-wise）和模式增长（pattern-growth）两种方法。逐层挖掘的算法基于算法 Apriori，模式增长方式的算法则基于算法 FP-Growth。表 1.2 列出了一些重要算法及其一些特征。

表 1.2 不确定数据集上的频繁模式挖掘的主要算法

时间	作者	出处	算法	方法	近似/精确
2007	Chui C K, Kao B, et al	PAKDD 2007	U-Apriori	逐层挖掘，候选项集筛选	精确
2007	Leung C K S, Carmichael C L, Hao B	ICDM Workshops 2007	UF-Growth	模式增长	精确
2009	Aggarwal C C, Li Y, et al	KDD 2009	HU-Mine	模式增长	精确
2009	Aggarwal C C, Li Y, et al	KDD 2009	UFP-Mine	模式增长，候选项集筛选	精确
2011	Wang L, Cheung D, Cheng R, et al	IEEE TKDE	MBP	逐层挖掘	精确
2012	Sun X, Liu L, Wang Sh	Journal of dvancements in Computing Technology	IMBP	逐层挖掘	近似
2012	Lin C W, Hong T P	Expert Systems with Applications	CUFP-Mine	候选项集筛选	精确

U-Apriori[121]是第一个不确定数据集上的频繁项集挖掘算法，该算法是基于Apriori提出的。该算法和Apriori的主要区别是前者扫描数据集是为了计算每个候选项集的期望支持数，而后者扫描数据集是为了计算候选项集的支持数。因此U-Apriori算法的缺陷同Apriori算法一样：产生候选项集，以及需要多遍扫描数据集来统计每层候选项集的期望支持数，如果最长的频繁项集长度是k，则最少需要扫描k次数据集。因此，如果数据集比较大、事务项集长度比较长或设定的最小期望支持数比较小，则U-Apriori的时间和空间性能都会受到很大的影响。

2011年，Wang等[115]提出一个不确定数据集的频繁项集挖掘算法MBP。该算法主要是对U-Apriori算法的改进，作者提出了两种策略来提高计算候选项集的期望支持数的效率：①在扫描数据集的过程中，如果一个候选项集提前被识别为非频繁项集，就停止计算该项集的实际期望支持数；②扫描数据集的过程中，如果一个候选项集的当前期望支持数已经大于预定义的最小期望支持数，就停止计算该项集的实际期望支持数。因此，算法MBP在时间和空间性能上得到了很大的提升。

2012年，Sun等[111]改善了算法MBP，基于MBP算法给出一个不确定数据集中频繁项集挖掘的近似算法IMBP。IMBP时间和空间性能优于MBP，然而，其准确性并不稳定，并且在稠密数据集中的精确度比较低。

2007年，Leung等[122]提出一个基于树的模式增长的算法UF-Growth，该算法中还提出一个新的树结构UF-Tree来存储不确定事务数据集，采用模式增长的方式从树上挖掘频繁项集。UF-Growth和FP-Growth的主要区别有两点：①UF-Tree树上每个节点除了保存和FP-Tree树上节点相同的信息外，还保存了每个节点的概率值，因此只有项相同，并且其相应的概率值相同的项才能共享同一个节点；而FP-Tree中，只要项相同就可以共享同一节点。②UF-Growth中计算频繁项集的时候都是统计项集的期望支持数，FP-Growth是计算项集的支持数。因此，UF-Tree树上的节点比较多，例如，对于两个不确定事务项集{a：0.50，b：0.70，c：0.23}和{a：0.55，b：0.80，c：0.23}，当按照字典顺序插入树中时，由于两个事务中项a的概率不相等，所以这两个项集不能共享同一个节点a，从而算法UF-Growth需要更多的空间和时间来处理UF-Tree。

Leung等[120]又改善UF-Growth以减小UF-Tree树的大小。改进算法的思想：先