概念漂移数据流挖掘算法综述

丁 剑 韩 萌 李 娟
(北方民族大学计算机科学与工程学院 银川 750021)

摘 要 数据流是一种新型的数据模型,具有动态、无限、高维、有序、高速和变化等特性。在真实的数据流环境中,一些数据分布是随着时间改变的,即具有概念漂移特征,称为可变数据流或概念漂移数据流。因此处理数据流模型的方法需要处理时空约束和自适应调整概念变化。对概念漂移问题和概念漂移数据流分类、聚类和模式挖掘等内容进行综述。首先介绍概念漂移的类型和常用概念改变检测方法。为了解决概念漂移问题,数据流挖掘中常使用滑动窗口模型对新近事务进行处理。数据流分类常用的模型包括单分类模型和集成分类模型,常用的方法包括决策树、分类关联规则等。数据流聚类方式通常包括基于k-means的和非基于k-means的。模式挖掘可以为分类、聚类和关联规则等提供有用信息。概念漂移数据流中的模式包括频繁模式、序列模式、episode、模式树、模式图和高效用模式等。最后详细介绍其中的频繁模式挖掘算法和高效用模式挖掘算法。

关键词 数据流挖掘,分类,聚类,模式挖掘,概念漂移
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.004

Review of Concept Drift Data Streams Mining Techniques

DING Jian HAN Meng LI Juan
(School of Computer Science and Engineering, Beifang University of Nationalities, Yinchuan 750021, China)

Abstract Data stream is a new data model proposed in recent years. It has different characteristics such as dynamic, infinite, high dimensional, orderly, high speed and evolving. In some data stream applications, the information embedded in the data is evolving over time that has the characteristics of concept drift or change. These data streams are known as evolving data streams or concept drift data streams. Therefore, the algorithms that mine data streams have space and time restrictions, and need to adapt change automatically. In this paper, we provided the survey of concept drift and classification, clustering and pattern mining on concept drift data streams. Firstly, we introduced the types and detection methods about concept drift. In order to deal with the concept drift, the sliding window model is used to mining data stream. The data stream classification model includes single model and ensemble model. The common methods include decision tree, classification association rules and so on. Data stream clustering methods can be divided into k-means based method and not. Pattern mining can provide useful patterns for classification, clustering, association rules and so on. Patterns include frequent patterns, sequential patterns, episode, sub-tree, sub-graph, high utility patterns and so on. Finally, we introduced the frequent patterns and high utility patterns in detail.

Keywords Data stream mining, Classification, Clustering, Frequent pattern mining, Concept drift

1 概述

近年来数据流作为一种新型数据模型广泛出现在多种应用领域。与传统的数据集不同,数据流具有时间顺序性、快速变化、海量和潜在无限等特性。数据流主要产生于网络,如Web点击流分析、网络日志、交通流量监控与管理、电力供应管理与预测、电信数据管理、金融服务及商业交易管理和分析等。专用网络同样会产生大量的数据流,如基于卫星的高分辨率测量地球测地学数据、雷达衍生的气象数据、连续的大型天文光学、调查红外线和无线电波长及大气辐射测量等。

数据流挖掘目前成为了一项新兴的智能信息处理技术,引起了广大研究工作者的关注,不论是在国内还是国外都得到了广泛重视,许多文献对该领域的研究进展进行了报告。数据流模型不同于传统的数据库,它具有一定的约束:

(1)数据流具有无限性,即包含的数据个数是无限的。因此使得存储受到限制,只能存储概要信息,其余信息被丢弃。

(2)数据到达速度快,需要实时处理,处理后即被丢弃。

(3)产生的数据项的分布会随着时间而改变,因此历史数据可能无关甚至有害。

(4)项集的组合爆炸会加剧挖掘任务的困难程度[1]。

到稿日期:2016-01-18 返修日期:2016-06-14 本文受国家自然科学基金项目(61563001),北方民族大学科研基金项目(2014XYZ13)资助。
丁 剑(1977-),男,硕士,副教授,主要研究方向为计算机应用与数据挖掘,E-mail:ychdj@163.com;韩 萌(1982-),女,博士生,副教授,主要研究方向为数据挖掘与机器学习,E-mail:compute2006_2@126.com;李 娟(1975-),女,硕士,副教授,主要研究方向为网络安全与数据挖掘。

从挖掘功能的角度考虑，目前数据流的挖掘主要包括数据流模式挖掘、数据流分类、数据流聚类和数据流查询等技术。由于数据流模型的特性，对其进行挖掘时需要考虑其时空约束，同时还要考虑因数据变化而带来的概念漂移问题。设计数据流挖掘算法时需要自适应概念的变化，因此数据流的处理应该是增量更新处理过程。

2 概念漂移问题

数据流的特点是观测到数据的潜在分布会随着时间改变。处理此类数据流时分类器应能挖掘概念改变的信息，并快速调整分类模型以适应概念变化[2]。研究中出现了很多处理数据流分类中的概念漂移问题的方式，包括使用滑动窗口和实例权重[3]、检测概念改变点[4]、监控两个不同时间窗口内分布[2]等。如 Gama 提出基于错误率的概念检测分类方法[5]；Baena 提出基于分类错误距离的概念检测方法[6]；Gama 提出一种两层学习系统来解决周期性概念问题[2]等。

在动态变化环境中，随着时间的推移，数据的潜在分布会动态变化，这被称为概念漂移。一般情况下，概念改变是不可预测的。给定预测目标变量 y 和条件变量 X，则一个实例可以表示为 (X,y)。概念改变可以定义为多个形式，在时间点 t_0 和 t_1 出现的概念漂移可以定义为式(1)所示[7]。其中 P_{t_0} 表示在时间点 t_0 时输入变量 X 和目标变量 y 之间的联合分布。

$$\exists X: P_{t_0}(X,y) \neq P_{t_1}(X,y) \tag{1}$$

概念漂移还可以表示为其他形式：

1)类的先验概率 $P(y)$可能改变；

2)类的条件概率 $P(X|y)$可能改变；

3)类的后验概率 $P(y|X)$可能改变，这会影响分类预测。

对于预测分类来说，其关注的是两类变化：第一类是数据分布 $P(y|X)$是否改变，是否影响预测结论；第二类是在不知道真实类标的情况下，数据分布的改变是否可见。当概念改变影响了预测结论时，其需要被处理。

现有的概念漂移问题分为两类：真实概念漂移(Real Concept Drift)和虚假漂移(Virtual Drift)。真实概念漂移是指无论 $P(X)$是否发生改变，$P(y|X)$均发生改变。虚假漂移是指输入数据改变，即 $P(X)$发生改变，但 $P(y|X)$没改变。如图 1 所示，图 1(a)是原始数据，不同形状代表不同的类；图 1(b)表示发生了真实概念漂移，$P(y|X)$发生改变；而图 1(c)表示发生了虚假漂移，即 $P(X)$发生改变，但 $P(y|X)$没改变。

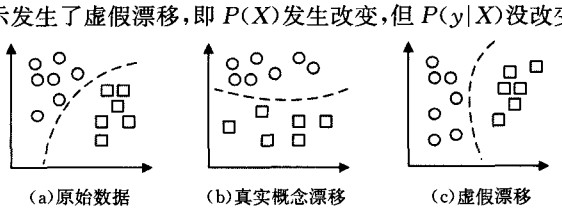

图 1 真实概念漂移和虚假漂移

概念漂移的检测方法通过识别变化点或发生小变化的时间间隔来标识和量化改变的发生。本节介绍 3 类常用的概念漂移检测方法，包括基于序列的分析方法、基于控制图的方法和基于两个分布之间不同的方法。这 3 类方法的比较如表 1 所列，其给出了在某一时刻 t 内存、时间的消耗比较，并给出了常用的算法实例，假定窗口大小为 N。

表 1 概念检测方法的复杂度比较

方法	内存	时间	算法示例
基于序列	O(1)	O(1)	PH
基于控制图	O(1)	O(1)	SPC
基于两个分布之间不同	O(logN)－O(N)	O(logN)－O(N)	ADWIN

2.1 基于序列的分析方法

PH(Page-Hinckley)方法是一种基于序列分析的检测器[7,8]。这是一种连续分析技术，被用于信号处理过程的概念改变检测。它可以有效地检测模型建立的正常行为中的概念改变。PH 检测可以连续适应且检测高斯信号平均值突变。

这种检测包含两个变量，测试变量 m_T 定义了至今为止观测值和平均值之间的累计差，如式(2)所示，其中 δ 是可以容忍的变化幅度。m_T 的最小值定义为 $M_T=\min(m_t, t=1,\cdots,T)$。PH 检测 m_T 和 M_T 之间的不同，如式(3)所示。当这种不同高于用户定义阈值 η 时，标记改变。定义较大的 η 值可以得到较少的误改变，但可能错过一些改变。

$$m_T=\sum_{t=1}^{T}(x_t-\bar{x}_T-\delta) \tag{2}$$

$$\bar{x}_T=\frac{1}{T}\sum_{t=1}^{T}x_t$$

$$PH_T=m_T-M_T \tag{3}$$

2.2 基于控制图的方法

SPC(Control Charts or Statistical Process Control)是一种基于统计过程控制的检测器[7,9]。它是标准的统计技术，被用于检测和控制连续过程生产的产品质量。SPC 把学习看作过程并且监视整个过程的演变。

给定预测目标变量 y 和条件变量 X，则一个实例可以表示为 (X,y)。对每个实例进行处理，分类模型预测为 y'，得到的结论可以是 true($y=y'$)或 flase($y\neq y'$)。给定一组实例，则错误率是一个随机变量且满足伯努利分布。二项式分布给出了表示 n 个实例错误率随机变量的一般形式。在时间点 i，假定观测到的值为 false 的错误率是概率 P_i($y\neq y'$)，则其标准偏差是：

$$\sigma_i=\sqrt{P_i(1-P_i)/i}$$

概念漂移检测器在模型操作过程中记录两个值：$P_{\min}$ 和 $\sigma_{\min}$。在时间点 i，对当前实例做过预测后更新预测错误，如果 $P_i+\sigma_i<P_{\min}+\sigma_{\min}$，则令 $P_{\min}=P_i$ 且 $\sigma_{\min}=\sigma_i$。

假定在时间点 i，实例 (X_i,y_i) 到达，模型预测后得到 P_i 和 σ_i，则系统处理实例的过程可以定义为以下 3 个状态。当出现不可控状态时说明出现了概念漂移，需要做处理。

(1)可控(In-Control, IC)。如果 $P_i+\sigma_i<P_{\min}+2\times\sigma_{\min}$，则称为 IC。即系统是稳定的，实例 (X_i,y_i) 与之前的实例是同分布的。

(2)不可控(Out-of-Control, OC)。如果 $P_i+\sigma_i\geq P_{\min}+3\times\sigma_{\min}$，则称为 OC。即错误与之前的实例相比有明显的增加。最新实例与历史实例有 99%的概率是不同分布的。

(3)警告(Warning)。状态在以上两个状态之间。这不是一个决定状态。

2.3 基于两个分布之间不同的方法

最近常用的监控分布方式是可变窗口方法(ADaptive WINdow, ADWIN)[10]，这是一种基于两个部分之间的不同

进行检测概念改变的方法。它使用 Hoeffding 边界来保证窗口的最大宽度，且在窗口内没有概念改变。ADWIN 是一种概念漂移检测器和评估器，它是一种可以很好地捕获流平均数的方法。它保留一个可变长度窗口大小的最新实例，窗口足够大使得这个窗口内不存在概念漂移。ADWIN 主要工作思路是：若最新窗口 W 中两个足够大的子窗口 W_1 和 W_2 可以展示足够明显的平均数，并且可以推断出相应的预测值是不同的，则窗口中较旧的部分可以删除。其中足够大和足够明显可以用 Hoeffding 边界定义，即两个子窗口的平均值大于变量 ε_{cut}，如式(4)所示。其中 $|W|$ 为最新窗口 W 的大小，$|W_1|$ 和 $|W_2|$ 是两个子窗口 W_1 和 W_2 的大小，且满足 $|W|=|W_1|+|W_2|$。m 是 W_1 和 W_2 的调和平均数(Harmonic Mean)。

$$m=\frac{2}{\frac{1}{|W_1|}+\frac{1}{|W_2|}}$$
$$\varepsilon_{cut}=\sqrt{\frac{1}{2m}\times\ln\frac{4|W|}{\delta}} \tag{4}$$

3 数据流分类

数据流分类模型主要分为单分类模型和集成分类模型。单分类模型技术维护和增量更新单个分类模型，有效地对概念漂移做出回应。集成模型需要比单个模型相对简单的技术更新概念，且同样有效地处理概念漂移。提出处理概念漂移时，集成分类器优于单个分类器；同时易于扩展和并行，通过剪枝整合中的某些部分可以快速适应漂移，它们可以得到更准确的概念描述。并且，通常基础分类器的训练速度要快于单一模型的更新速度，因此它也更加适合处理高速产生的数据流。由于数据流的特征，对其进行处理时主要采用的是增量算法[7]。增量算法是指按照顺序一个接一个(或一批接一批)地处理实例，每次处理一个(一批)实例后更新模型。

数据流分类方法包括神经网络[11,12]、支持向量机[13,14]、关联/分类规则及决策树等。本节对常用的分类方法进行介绍。

3.1 分类关联规则

关联规则挖掘是一种基于频繁模式的分类方式，在传统数据库中得到广泛应用。近年来，出现了针对数据流的规则分类算法。

如 CAPE[15] 是最早的基于频繁模式处理数据流的规则分类算法，它利用衰减窗口方法处理概念漂移问题，取得了比较好的实验结果。CBC-DS[16] 算法采用闭合频繁模式来挖掘类关联规则，使得算法具有较高的效率。PNRMXS[17] 算法发现 XML 数据流中的正、负关联规则。Esper[18] 在 Aprior 算法和 FPGrowth 算法的基础上发现数据流中的关联规则并做相关分析。该算法对不同特征的数据流使用滑动窗口模型和倾斜窗口模型进行规则挖掘。AMRules[19] 发现数据流中的规则用于解决回归问题。规则的前件是属性值的条件连接，后件是属性值的线性组合。它设计一种策略以检验数据的改变，并通过剪枝规则集合对改变做出反应。FRBCs[20] 在 FP-Growth 算法的基础上进行了模糊扩展，挖掘模糊频繁模式，从而生成模糊关联分类模型。其中模糊的项是由离散化输入变量和从这些离散间隔中定义强模糊划分产生的。MapReduce[21] 在 FPGrowth 算法的基础上进行分布式设计挖掘分类关联规则。一旦挖掘出分类关联规则，则进行分布式规则剪枝。得到的规则集合可以用于分类未标记的模式。

3.2 决策树分类模型

决策树模型被广泛用于创建分类器处理数据流，原因在于决策树模型类似于人类的推理，很容易被理解[22]。其中基于 Hoeffding 的决策树算法是从数据流中学习树的最受欢迎的方法之一。如快速决策树(Very Fast Decision Tree，VFDT)[23]、概念自适应快速决策树(Concept-adapting Very Fast Decision Tree，CVFDT)[24]、VFDTc[25]、二元回归树(Binary Regression Trees)[26]、模糊模式树(Fuzzy Pattern Trees)[27]、哈稀树[28]、Hoeffding 选择树(Hoeffding Option Trees，HOT)[29]、Hoeffding 自适应树(Hoeffding Adaptive Tree，HAT)[30]、自适应 Hoeffding 选择树(Adaptive Hoeffding Option Tree，AdoHOT)[31] 和自适应大小 Hoeffding 树(Adaptive-Size Hoeffding Tree，ASHT)[31] 等。其中 VFDT 是一种基于 Hoeffding 不等式针对数据流挖掘环境建立分类决策树的方法。它通过不断地将叶节点替换为分支节点而生成，即在每个决策节点保留一个重要的统计量，当该节点的统计量达到一定阈值时进行分裂测试。其最主要的创新是利用 Hoeffding 不等式确定叶节点变为分支节点所需要的样本数目与分裂点。该算法仅需对数据流扫描一次，具有较高的时空效率，且其分类器性能近似于传统算法生成的分类器。其不足在于不能很好地处理概念漂移问题。CVFDT 对 VFDT 进行了扩展以快速解决概念漂移数据流的分类。其核心思想是当新子树分类更准确时，用新子树替换历史子树。它维持一个滑动训练窗口，并通过在样本流入和流出窗口时更新已生成的决策树，使其与训练窗口内的样本保持一致。

3.3 集成分类模型

集成学习使得分类器具有更高精度的特性，可以很好地适应概念的变化，将概念漂移的影响削弱在共同决策中。常用的集成方式包括 Boosting 和 Bagging。Bagging 方式是将原始数据集通过 T 次随机采样，从而得到 T 个与原始数据集大小相同的子数据集，再分别训练得到 T 个单分类器，然后结合(如投票)为一个集成分类器。Boosting 也是通过重采样得到多个单分类器，最后得到一个集成分类器。区别在于 Bagging 是基于权值的分类器集成。

如 Bifet[31] 提出了一种基于 ADWIN 对多个单分类器投票而得到的集成分类器，用于处理概念漂移数据流。数据随着时间发生变化，分类器自动调整。Grossi[32] 通过对每个子分类方法的权重进行衡量生成集成分类方法。由于没有处理过期事务的方法得到的权重较大，因此每个权重起到了处理概念漂移的关键作用。Farid[33] 提出了自适应集成方法来进行概念漂移数据流的分类和新类检测等。Brzezinski[34] 提出了一种基于数据块的在线处理概念漂移数据流的集成分类方法。它将数据流划分成多个数据块，每个块训练的分类器具有不同的权重，然后对块的特征进行分析来产生新类型的集成分类器。Czarnowski[35] 同样对数据块进行分块处理，每块生成基分类器，而后赋予不同权重进行分类器集成。Ikonomovska[36] 采用了在线权重集成方法和在线随机森林方法来设计集成分类方法，其可用于分类和回归。Abdallah[37] 设计了一种可变的轻量框架用于移动用户行为分类。这种框架使用一种集成分类模型，它对每个窗口内的预测值进行投票，从而得到最终的分类结果。Hosseini[38] 提出一种半监督学习的

集成分类方法来处理可变数据流。设定缓冲池存储多个分类算法，每个分类器处理独立的概念。每个分类器赋予不同的权重，集成方法根据权重函数最终给出分类结果。ZareMoodi[39]将集成分类方法用于发现数据流中出现的新类。其采用局部集成方式，每个局部集成分类器都是针对某一个类值的。

4 数据流聚类

聚类问题可以认为是将数据集分成若干个相似对象构成的子集的过程，这里的子集称为簇或类别。聚类的结果是使簇内的对象尽量相似，同时与其它簇的对象尽量不同[40]。由于聚类时的数据对象没有类别标签，因此聚类是无监督的学习过程。聚类分析是重要的和基本的数据挖掘方法。它可以单独用来分析获取数据的分布情况，将数据分成不同的簇，观察每一个簇的特性，然后对特定的目标簇做进一步的处理。聚类还可以作为其它数据处理方法的预处理技术，比如产生类别标签给分类提供支持，提取特征以支持相关分析、频繁项挖掘、预测异常值检测等。

研究者针对数据流提出了许多聚类算法。如算法 Clustream[41]是采用界标窗口，并以应用中心请求为导向的数据流聚类方法。它把聚类过程分成在线部分和离线部分。在线部分周期地存储概要统计信息，离线部分仅使用这些概要信息。分析者利用离线部分的各种输入来快速理解数据流中的广泛簇群。如何有效选择、存储和使用这些统计数据是难题。为了解决这个问题，该算法设计使用椎体时间框架(Pyramidal Time Frame)并结合微聚类方法。Clustream 算法是基于 k-means 的，不能很好地发现任意形状的簇，且不能处理异常，需要先验知识和用户定义的时间窗口。DGClust[42]用于处理全网络产生的数据流的聚类问题。它是一种分布式算法，可以减少维度和通信负担，允许每个本地传感器保持其数据流的在线离散化。DGClust 使用网格存储数据，在界标窗口中处理最新数据，它也是基于 k-means 的。D-Stream[43]算法能解决基于 k-means 算法的不足。它是一种基于密度的数据流聚类算法，分为在线和离线两部分，在线部分将每个输入数据映射到一个网格，而离线部分计算网格密度并依据密度对网格聚类。D-Stream 使用密度衰减技术来捕获数据流中的动态改变。该算法利用衰减因子、数据密度和簇结构之间的复杂关联可以实时生成和调整簇。ClusTree[44]采用衰减窗口自动适应数据流速度，是无参数的单遍扫描数据流的聚类方法。它总是维持最新的聚类模型，并且报告概念漂移以及新的和异常的值。该算法不是采用先验知识假设聚类模型尺寸，而是自适应调整的。StreamKM++[45]是采用欧氏距离处理数据流聚类问题的算法。它使用点树存储数据信息，也是一种基于 k-means 的方法。该算法使用界标窗口处理最新的数据。StreamKM++算法对数据流的小权重抽样数据进行处理，为此提出了两种新的结构。首先使用自适应的非均匀采样方法从数据流中获得一些小的点；其次，该算法设计数据结构点树用于显著加快自适应的时间和非均匀采用点的过程。

5 数据流模式挖掘

频繁模式(Frequent Patterns)或频繁项集(Frequent Itemsets)是指在数据集合中出现次数多于用户定义最小支持度阈值的项集。频繁模式挖掘可以看作许多数据挖掘任务的基础，如关联分析、相关性分析、序列挖掘、分类和聚类等。

数据流中挖掘的模式类型主要包括：频繁序列(Sequential Patterns)、高效用(High Utility Patterns)、子图(Subgraphs)、子树(Subtree)、episode 和频繁模式等。

5.1 频繁模式挖掘

频繁模式挖掘是数据挖掘的热点问题，在传统数据库中挖掘频繁模式这个问题已经被广泛地研究和应用。然而，在数据流环境下挖掘频繁模式给研究者带来了更大的机遇和挑战。

近年来，在数据流中发现频繁模式通常采用的方式是使用滑动窗口模型来避免概念漂移问题。发现的频繁模式主要分为两类：完全频繁模式集合和压缩频繁模式集合。算法 SWCA[46]，MSW[47]，EclatDS[48]和 SA-Miner[89]采用滑动窗口发现完全模式结果集，由此带来的不足是会产生大量无用的、短的模式。为了减少模式的数量，需要挖掘压缩模式，如最大频繁模式(Maximal Frequent Pattern)、闭合频繁模式(Closed Frequent Pattern)和 top-k 频繁模式(top-k Frequent Pattern)等。其中最大模式和 top-k 模式为有损压缩模式，压缩程度一般都强于闭合模式。算法 Max-FISM[50]采用滑动窗口，AMFIoDS[51]采用界标窗口挖掘数据流中的最大模式。Top-k-FCI[52]和 FCI_max[53]等算法挖掘 top-k 频繁模式。闭合模式是强大的频繁模式表现方式之一，因为它能消除冗余信息。一般来说，闭合频繁模式比频繁模式全集中的模式数量少得多，且闭合模式包含了频繁模式全集中的全部信息。Moment[54]，Stream_FCI[47]，TMoment[55]，IncMine[56]，CloStream+[57]和 TDMCS[58]等算法采用滑动窗口挖掘数据流的闭合频繁模式。其中，算法 CFI-Stream 存储了全部的闭合模式，但该模式不满足最小支持度的要求，因此会付出一些存储代价。Moment 算法比当前频繁闭合模式存储的信息更多，但是会用最小支持度阈值来降低模式的数量。IncMine 提出了半频繁闭合模式的概念，增加了一个模式的最小支持度阈值，使得它在窗口内保留时间更长。TDMCS 采用衰减滑动窗口模型发现满足最大误差阈值的闭合频繁模式和半频繁闭合模式，可以保证不丢失可能的模式。

5.2 高效用模式挖掘

频繁模式挖掘的一个主要的限制在于它假定每个项(Item)只能在每个事务中出现一次，且这些项被赋予相同的重要性。这些假定使得它不能满足真实应用的需求。频繁模式挖掘算法会丢弃项的权重信息，发现许多低权重的频繁项集，且仅能发现少量高权重的频繁项集。为此，应采用高效用模式挖掘方法来考虑项在每条事务中出现多于一次和每个项有权重的问题。效用挖掘具有广泛的应用，例如网站的点击流分析、交叉销售的零售商店销售和生物医学应用等。若发现高效用模式，重要商务便可以决定收益最大化、最小化营销或减少库存；生物基因数据可以发现重要的基因片段等。

已有的高效用模式挖掘方法主要采用两阶段方式和一阶段方式。前者在挖掘过程中需要产生大量的候选高效用项集，一阶段方式不需要产生候选高效用项集，但是需要消耗大量的连接操作来评估每个项集的高效用。如 Two-Phase[59]，IHUP[60]，UMining[61]和 UP-Growth[62]采用不同的剪枝策略减小搜索空间，使用两阶段方法发现高效用模式。第一阶段用于产生候选效用项集，第二阶段精确地计算每个候选项集

的效用，从而找到高效用模式。最近，研究者在 UHI-Miner 算法[63]中提出了一种更有效的方法，即采用一阶段直接发现高效用模式。它相比之前的两阶段方法表现更优。但是，发现高效用模式依然非常消耗执行时间，因此还需要设计更加有效的方法。

以上这些算法适用于传统频繁模式挖掘，适用于数据流的高效用模式挖掘算法较少。最早使用滑动窗口发现高效用模式的算法是 THUI-Mine[64]，它使用两阶段方法发现了高效用模式。基于滑动窗口发现高效用模式的算法 MHUI-BIT 和 MHUI-TID[65]被提出，它们被证实优于之前的算法。其主要的优势在于它们使用了一种树结构来存储长度为 1 和 2 的候选模式，可以减少候选模式的数量。一些研究成果也用于发现高效用压缩模式，如 GUIDE[66]采用树形结构挖掘数据流的最大高效用模式，T-HUDS[67]采用一种压缩树结构来发现滑动窗口内可能的 top-k 高效用模式，AprioriCH，AprioriHC-D 和 CHUD[68]采用不同的策略挖掘闭合高效用模式。这些发现数据流中高效用模式的方法的不足在于会产生大量的候选集合，导致需要大量的内存来存储这些候选集，并且产生候选集和计算这些候选集的效用也会消耗大量的执行时间；发现的模式数据量依然很大，不利于用户的使用，尤其是对长事务或长高效模式，情况更加糟糕。

6 进一步的研究方向

概念漂移数据流的挖掘方法在过去的十多年中得到了一定的发展，但是现有的方法仍然存在着不足之处，这为研究者提供了进一步的研究方向。

(1)大规模分布式数据流的计算模型、连续计算理论和随机数据算法的研究。其包括针对 TCP/IP 网络监控的算法和计算模型研究，电信通信网络数据流的计算，虚拟天文等科学数据流的存储、查询和挖掘，为了提高搜索引擎效率对 Web 数据流的挖掘研究等。科学和技术数据收集的性质正在迅速演变，例如数据量和数据产生速度呈指数方式增长，伴随着不断增加的复杂性和信息内容，因此有趣的或者异常显现必须被快速地测量和处理。

(2)大规模数据流中模式发现问题研究。其面临的主要问题是发现的模式数量巨大，其中存在大量无用的模式。当事务长或最小支持度阈值低时，这个问题尤其严重。为了减少模式的数量，需要挖掘压缩模式，进一步研究压缩的频繁模式，并在压缩模式的基础上再次筛选，得到更加精简的模式集合。

(3)数据流中的概念漂移问题是研究的热点，虽已有大量研究工作及成果，但缺少有效的概念漂移检测及处理方法，因此可以对模糊周期或模糊可预测的概念漂移问题进行研究。大多数概念漂移是假设在隐藏背景下发生的，是不可观测到的。但是，在某些应用中概念漂移是可以预测的，它沿时间轴或在不同对象中的模型化区域可能重新出现。例如粮食需求预测，可以用模糊周期性季节的影响为对象设定特定子群。

(4)基于模式的分类方法研究。数据流中包含无限的数据，这些数据包含大量的冗余信息甚至是噪声，而模式发现可以去除数据中的冗余信息且不受噪声的影响。因此，挖掘有趣的、频繁的和有区分力的模式以用于有效的分类。基于模式的分类具有更高的准确性，并且可以很好地解决缺失值的问题。因此可以进一步对基于模式的数据流分类方法进行研究。

结束语 由于数据流中包含的知识会随着时间的发展而发生改变，因此挖掘数据流时需要考虑时空效率和概念漂移问题。文中首先对概念漂移的类型和解决方法进行了整理和比较。接着介绍了数据流挖掘的常用技术，包括数据流分类、聚类和模式挖掘。数据流分类常用的方法包括单分类模型和集成分类模型，文中分别对不同的分类方式进行综述。数据流聚类方式包括基于 k-means 的和非基于 k-means 的，文中对相关方式进行了介绍。数据流中的模式类型很多，最常见的是频繁模式和高效用模式，文中介绍了相关的挖掘方法。数据流除了包含概念漂移问题，还有不确定性、多维、噪声等问题，还需对这些特征数据流做一进步研究。

参 考 文 献

[1] Cheng J, Ke Y, Ng W. A survey on algorithms for mining frequent itemsets over data streams[J]. Knowledge and Information Systems, 2008, 16(1): 1-27

[2] Gama J, Kosina P. Recurrent concepts in data streams classification[J]. Knowledge and Information Systems, 2014, 40(3): 489-507

[3] Klinkenberg R. Learning drifting concepts: example selection vs. example weighting[J]. Intelligence Data Analysis, 2004, 8(3): 281-300

[4] Kosina P, Gama J. Very fast decision rules for classification in data streams[J]. Data Mining and Knowledge Discovery, 2015, 29(1): 168-202

[5] Gama J, Medas P, Castillo G, et al. Learning with drift detection[C]// Proceedings of the 17th Brazilian Symposium on Artificial Intelligence. Berlin, Germany: Springer-Verlag, 2004: 286-295

[6] Baena G M, Campo A J, Fidalgo R, et al. Early drift detection method[C]// Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams. Berlin, Germany, 2006: 77-86

[7] Gama J, Zliobaite I, Bife A, et al. A survey on concept drift adaptation[J]. ACM Computing Surveys, 2014, 46(4): 1-37

[8] Ikonomovska E, Gama J, Dzeroski S. Learning model trees from evolving data streams[J]. Data Mining Knowledge Discovery, 2011, 23(1): 128-168

[9] Gomes J B, Menasalvas E, Sousa P A C. Learning recurring concepts from data streams with a context-aware ensemble[C]// Proceedings of the 26th Annual ACM Symposium on Applied Computing. New York, United States: Association for Computing Machinery, 2011: 994-999

[10] Bifet A, Gavaldá R. Learning from time-changing data with adaptive windowing[C]// Proceedings of the Seventh SIAM International Conference on Data Mining, Minnesota, USA, 2007. Philadelphia, United States: Society for Industrial and Applied Mathematics, 2007: 443-448

[11] Ghazikhani A, Monsefi R, Yazdi H S. Ensemble of online neural networks for non-stationary and imbalanced data streams[J]. Neurocomputing, 2013, 122: 535-544

[12] Cao K, Wang G, Han D, et al. An algorithm for classification over uncertain data based on extreme learning machine[J]. Neurocomputing, 2016, 174(Part A): 194-202

[13] Cervantes J, Lamont F G, Chau A L, et al. Data selection based on decision tree for SVM classification on large data sets[J]. Applied Soft Computing, 2015, 37: 787-798

[14] Kranjc J, Smailović J, Podpečan V, et al. Active learning for sentiment analysis on data streams: Methodology and workflow im-

plementation in the ClowdFlows platform[J]. Information Processing & Management, 2015, 51(2): 187-203

[15] Wang P, Wu X C, Wang C, et al. CAPE-A classification algorithm using frequent patterns over data streams[J]. Journal of Computer Research and Development, 2004, 41(10): 1677-1683 (in Chinese)
王鹏, 吴晓晨, 王晨, 等. CAPE-数据流上的基于频繁模式的分类算法[J]. 计算机研究与发展, 2004, 41(10): 1677-1683

[16] Ao F J, Wang T, Liu B H, et al. CBC-DS: A classification algorithm based on closed frequent patterns for mining data streams [J]. Journal of Computer Research and Development, 2009, 46 (5): 779-786(in Chinese)
敖富江, 王涛, 刘宝宏, 等. CBC-DS: 基于频繁闭模式的数据流分类算法[J]. 计算机研究与发展, 2009, 46(5): 779-786

[17] Çokplnar S, Gündem T I. Positive and negative association rule mining on XML data streams in database as a service concept [J]. Expert Systems with Applications, 2012, 39(8): 7503-7511

[18] Ari I, Olmezogullari E, Celebi O F. Data stream analytics and mining in the cloud[C]// Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science. Los Alamitos, USA: IEEE, 2012: 857-862

[19] Almeida E, Ferreira C, Gama J. Adaptive model rules from data streams[C]// Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Berlin, Germany: Springer-Verlag, 2013: 480-492

[20] Antonelli M, Ducange P, Marcelloni F, et al. A novel associative classification model based on a fuzzy frequent pattern mining algorithm[J]. Expert Systems with Applications, 2015, 42(4): 2086-2097

[21] Bechini A, Marcelloni F, Segatori A. A MapReduce solution for associative classification of big data[J]. Information Sciences, 2016, 332(c): 33-55

[22] Kotsiantis S B. Decision trees: a recent overview[J]. Artificial Intelligence Review, 2013, 39(4): 261-283

[23] Domingos P, Hulten G. Mining high-speed data streams [C]// Proceedings of the Sixth ACM International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2000: 71-80

[24] Hulten G, Spencer L, Domingos P. Mining time-changing data streams[C]// Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2001: 97-106

[25] Gama J, Rocha R, Medas P. Accurate decision trees for mining high-speed data streams[C]// Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining. New York, United States: Association for Computing Machinery, 2003: 523-528

[26] Fidalgo-Merino R, Nunez M. Self-adaptive induction of regression trees[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1659-1672

[27] Shaker A, Senge R, Hüllermeier E. Evolving fuzzy pattern trees for binary classification on data streams[J]. Information Sciences, 2013, 220: 34-45

[28] Li B, Zhu X, Chi L, et al. Nested subtree hash kernels for large-scale graph classification over streams [C] // Proceedings of IEEE 12th International Conference on Data Mining. Piscataway, United States: Institute of Electrical and Electronics Engineers Inc., 2012: 399-408

[29] Pfahringer B, Holmes G, Kirkby R. New options for hoeffding trees[C]// Proceedings of the 20th Australian Joint Conference on Artificial Intelligence. Heidelberg, Germany: Springer Verlag, 2007: 90-99

[30] Bifet A, Gavaldá R. Adaptive learning from evolving data streams [C]// Proceedings of the 8th International Symposium on Intelligent Data Analysis. Berlin, Germany: Springer-Verlag, 2009: 246-260

[31] Bifet A, Holmes G, Pfahringer B. New ensemble methods for evolving data streams[C]// Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining. New York, United States: Association for Computing Machinery, 2009: 139-148

[32] Grossi V, Turini F. Stream mining: a novel architecture for ensemble-based classification[J]. Knowledge and Information Systems, 2012, 30(2): 247-281

[33] Farid D M, Zhang L, Hossain A, et al. An adaptive ensemble classifier for mining concept-drifting data streams[J]. Expert Systems with Applications, 2013, 40(15): 5895-5906

[34] Brzezinski D, Stefanowski J. Combining block-based and online methods in learning ensembles from concept drifting data streams[J]. Information Sciences, 2014, 265(5): 50-67

[35] Czarnowski I, Jędrzejowicz P. Ensemble Classifier for Mining Data Streams[J]. Procedia Computer Science, 2014, 35: 397-406

[36] Ikonomovska E, Gama J, Džeroski S. Online tree-based ensembles and option trees for regression on evolving data streams [J]. Neurocomputing, 2015, 150(Part B): 458-470

[37] Abdallah Z S, Gaber M M, Srinivasan B. Adaptive mobile activity recognition system with evolving data streams[J]. Neurocomputing, 2015, 150: 304-317

[38] Hosseini M J, Gholipour A, Beigy H. An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams[J]. Knowledge & Information Systems, 2016, 46(3): 1-31

[39] ZareMoodi P, Beigy H, Siahroudi S K. Novel class detection in data streams using local patterns and neighborhood graph[J]. Neurocomputing, 2015, 158: 234-245

[40] Silva J, Faria E R, Barros R C, et al. Data stream clustering: A survey[J]. ACM Computing Surveys, 2013, 46(1): 125-134

[41] Aggarwal C C, Han J, Wang J, et al. A framework for clustering evolving data streams[C]// Proceedings of the 29th Conference on Very Large Data Bases. Berlin, Germany, 2003, 29: 81-92

[42] Gama J, Rodrigues P P, Lopes L. Clustering distributed sensor data streams using local processing and reduced communication [J]. Intelligent Data Analysis, 2011, 15(1): 3-28

[43] Chen Y, Tu L. Density-based clustering for real-time stream data[C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, United States: Association for Computing Machinery, 2007: 133-142

[44] Kranen P, Assent I, Baldauf C, et al. The clustree: indexing microclusters for anytime stream mining[J]. Knowledge and Information Systems, 2011, 29(2): 249-272

[45] Ackermann M R, Märtens M, Raupach C, et al. StreamKM++: A clustering algorithm for data streams[J]. Journal of Experimental Algorithmics, 2010, 17(7): 173-187

(下转第 62 页)

[13] Cao Jie, Guo Lei-lei. Evidence combination rule based on local conflict distribution strategy[J]. Application Research of Computers, 2013, 30(7): 2033-2035(in Chinese)
曹洁, 郭雷雷. 一种基于局部冲突分配的证据组合规则[J]. 计算机应用研究, 2013, 30(7): 2033-2035

[14] Jousselme A L, Grenier D, Bosse E. A new distance between two bodies of evidence[J]. Information Fusion, 2001, 2(2): 91-101

[15] Lin Guo-ping, Liang Ji-ye, Qian Yu-hua. Multigranulation view based fusing strategy of D-S evidence[J]. Computer Science, 2014, 41(2): 45-48(in Chinese)
林国平, 梁吉业, 钱宇华. 基于多粒度视角下的 D-S 证据理论融合策略[J]. 计算机科学, 2014, 41(2): 45-48

[16] Jiang Wen, Zhang An, Deng Yong. A novel information fusion method based on our evidence conflict representation[J]. Journal of Northwestern Polytechnical University, 2010, 28(1): 27-32 (in Chinese)
蒋雯, 张安, 邓勇. 基于新的证据冲突表示的信息融合方法研究[J]. 西北工业大学学报, 2010, 28(1): 27-32

[17] Song Ya-fei, Wang Xiao-dan, Lei Lei, et al. Measurement of evidence conflict based on correlation coefficient[J]. Journal of Communications, 2014, 35(5): 95-100 (in Chinese)
宋亚飞, 王晓丹, 雷蕾, 等. 基于相关系数的证据冲突度量方法[J]. 通信学报, 2014, 35(5): 95-100

[18] Zhang Z J, Liu T H, Chen D, et al. Novel algorithm for identifying and fusing conflicting data in wireless sensor networks[J]. Sensors, 2014, 14(6): 9562-9581

[19] Dempster A P. Upper and lower probabilities induced by a multivalued mapping[J]. Annals of Mathematical Statistics, 1967, 38(2): 325-339

[20] Shafer G. A mathematical theory of evidence[M]. Princeton: Princeton University Press, 1976

(上接第 29 页)

[46] Li C W, Jea K F. An adaptive approximation method to discover frequent itemsets over sliding-window-based data streams[J]. Expert Systems with Applications, 2011, 38(10): 13386-13404

[47] Li G H, Chen H. Mining the frequent patterns in an arbitrary sliding window over online data streams[J]. Journal of Software, 2008, 19(19): 2585-2596(in Chinese)
李国徽, 陈辉. 挖掘数据流任意滑动时间窗口内频繁模式[J]. 软件学报, 2008, 19(19): 2585-2596

[48] Deypir M, Sadreddini M H, Hashemi S. Towards a variable size sliding window model for frequent itemset mining over data streams[J]. Computer & Industrial Engineering, 2012, 63(1): 161-172

[49] Li C W, Jea K F. An approach of support approximation to discover frequent patterns from concept-drifting data streams based on concept learning[J]. Knowledge and Information Systems, 2014, 40(3): 639-671

[50] Farzanyar Z, Kangavari M, Cercone N. Max-FISM: Mining (recently) maximal frequent itemsets over data streams using the sliding window model[J]. Computers and Mathematics with Applications, 2012, 64: 1706-1718

[51] Li H F, Zhang N. Approximate maximal frequent itemset mining over data stream[J]. Journal of Information and Computational Science, 2011, 8(12): 2249-2257

[52] Li J, Gong S. Top-k-FCI: Mining top-k frequent closed itemsets in data streams[J]. Journal of Computational Information Systems, 2011, 7(13): 4819-4826

[53] Tsai C Y, Liou J J H, Chen C J, et al. Generating touring path suggestions using time-interval sequential pattern mining[J]. Expert Systems with Applications, 2010, 37: 6968-6973

[54] Chi Y, Wang H X, Yu P S, et al. Catch the moment: maintaining closed frequent itemsets over a data stream sliding window[J]. Knowledge and Information Systems, 2006, 10(3): 265-294

[55] Nori F, Deypir M, Sadreddini M H. A sliding window based algorithm for frequent closed itemset mining over data streams [J]. Journal of Systems and Software, 2013, 86(3): 615-623

[56] Cheng J, Ke Y, Ng W. Maintaining frequent closed itemsets over a sliding window[J]. Journal of Intelligent Information Systems, 2008, 31(3): 191-215

[57] Yen S J, Wu C W, Lee Y S, et al. A fast algorithm for mining frequent closed itemsets over stream sliding window[C]// Proceedings of 2011 IEEE International Conference on Fuzzy Systems. Taipei, Taiwan, 2011: 996-1002

[58] Han M, Wang Z H, Yuan J D. Efficient method for mining closed frequent patterns from data streams based on time decay model[J]. Chinese Journal of Computers, 2015, 38(7): 1473-1483(in Chinese)
韩萌, 王志海, 原继东. 一种基于时间衰减模型的数据流闭合模式挖掘方法[J]. 计算机学报, 2015, 38(7): 1473-1483

[59] Ahmed C F, Tanbeer S K, Jeong B S, et al. Efficient tree structures for high-utility pattern mining in incremental databases [J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(12): 1708-1721

[60] Liu Y, Liao W, Choudhary A. A two-phase algorithm for fast discovery of high utility itemsets[C]// Proceedings of PAKDD. 2005: 689-695

[61] Yao H, Hamilton H J, Geng L. A unified framework for utility-based measures for mining itemsets[C]// Proceedings of ACM SIGKDD(USA). 2006: 28-37

[62] Tseng V S, Wu C W, Shie B E, et al. UP-Growth: An efficient algorithm for high utility itemsets mining[C]// Proceedings of the 16th ACM SIGKDD. Washington, USA, 2010: 253-262

[63] Liu M, Qu J. Mining high utility itemsets without candidate generation[C]// Proceedings of CIKM12. 2012: 55-64

[64] Tseng V S, Chu C J, Liang T. Efficient mining of temporal high utility itemsets from data streams[J]. Information and Software Technology, 2006, 48(6): 357-369

[65] Li H F, Huang H Y, Chen Y C, et al. Fast and memory efficient mining of high utility itemsets in data streams[C]// Proceedings of the 8th ICDM. 2008: 881-886

[66] Shie B E, Yu P S, Tseng V S. Efficient algorithms for mining maximal high utility itemsets from data streams with different models[J]. Expert Systems with Applications, 2012, 39: 12947-12960

[67] Zihayat M, An A. Mining top-k high utility patterns over data streams[J]. Information Sciences, 2014, 285(1): 138-161

[68] Tseng V S, Wu C W, Fournier-Viger P, et al. Efficient algorithms for mining the concise and lossless representation of high utility itemsets[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(3): 726-739