计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20200432
57(8): 1683-1696, 2020

基于随机投影的高维数据流聚类

朱颖雯[1,2,3]　　陈松灿[1,2]

[1](南京航空航天大学计算机科学与技术学院　南京　211106)

[2](模式分析与机器智能工业和信息化部重点实验室(南京航空航天大学)　南京　211106)

[3](三江学院计算机科学与工程学院　南京　210012)

(yingwen.zhu@nuaa.edu.cn)

High Dimensional Data Stream Clustering Algorithm Based on Random Projection

Zhu Yingwen[1,2,3] and Chen Songcan[1,2]

[1](*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106)

[2](*MIIT Key Laboratory of Pattern Analysis and Machine Intelligence (Nanjing University of Aeronautics and Astronautics), Nanjing* 211106)

[3](*College of Computer Science and Engineering, Sanjiang University, Nanjing* 210012)

Abstract　High dimensional data streams emerge ubiquitously in many real-world applications such as network monitoring. Clustering such data streams differs from traditional data clustering algorithm where the given datasets are generally static and can be read and processed repeatedly, thus facing more challenges due to having to satisfy such constraints as bounded memory, single-pass, real-time response and concept-drift detection. Recently many methods of such type have been proposed. However, when dealing with high dimensional data, they often result in high computational cost and poor performance due to the curse of dimensionality. To address the above problem, in this paper we present a new clustering algorithm for data streams, called RPFART, by combining the random projection method with the adaptive resonance theory (ART) model that has linear computational complexity, uses a single parameter, i.e., the vigilance parameter to identify data clusters, and is robust to modest parameters setting. To gain insights into the performance improvement obtained by our algorithm, we analyze and identify the major influence of random projection on ART. Although our method is embarrassingly simple just by incorporating the random projection into ART, the experimental results on variety of benchmark datasets indicate that our method can still achieve comparable or even better performance than RPGStream algorithm even if the raw dimension is compressed up to 10% of the original one. For ACT1 dataset, its dimension is reduced from 67 500 to 6 750.

Key words　high dimensional data; data stream clustering; random projection; adaptive resonance theory; clustering

摘　要　高维数据流在许多现实应用中广泛存在,例如网络监控.不同于传统的静态数据聚类问题,数据流聚类面临有限内存、单遍扫描、实时响应和概念漂移等问题.然而现有许多数据流聚类算法在处理高维数据时,常常因产生维数灾难而导致高计算复杂度和较差的性能.为了解决此问题,基于随机投影和自适应谐振理论(adaptive resonance theory, ART)提出了一种针对高维数据流的高效聚类算法 RPFART.该算法具有线性计算复杂度,仅包含 1 个超参数,并对参数设置鲁棒.详细分析了随机投影对 ART 的主要影响,尽管该

收稿日期:2020-06-16;**修回日期**:2020-07-03

基金项目:国家自然科学基金重点项目(61732006)

This work was supported by the Key Program of National Natural Science Foundation of China (61732006).

通信作者:陈松灿(s.chen@nuaa.edu.cn)

算法仅简单地将随机投影与 ART 方法进行了结合,但在多个数据集上的实验结果表明:即使将原始尺寸压缩到 10%,该方法仍可以达到与 RPGStream 算法相当的性能.对于 ACT1 数据集,其维数从 67 500 减少到 6 750.

关键词 高维数据;数据流聚类;随机投影;自适应谐振理论;聚类

中图法分类号 TP391

随着云计算、物联网的快速发展,许多新型的应用领域,诸如网络入侵检测、视频监控、气象卫星遥感以及电力供应网等,每时每刻都在产生大量的数据.这些数据并不事先存放在存储介质中,而是像水流一样不断出现,它们具有快速(high speed)、时序(temporally ordered)、海量(massive)等特征,被称作数据流(data stream).

越来越多数据流的产生和应用需求使得对于数据流的挖掘变得炙手可热.挖掘数据流[1-9]的目的是从这些连续不断的流数据中提取隐藏的知识结构.数据流挖掘技术包括数据流分类、数据流聚类、数据流关联规则挖掘等.其中,数据流聚类是数据流学习的一项重点任务,它是将数据对象集合中相似对象划分为一个或多个组(称为"簇",cluster)的过程,划分后同一簇中元素彼此相似,但与其他簇中元素相异.不同于传统的静态数据聚类,数据流聚类面临许多问题,例如:1)有限内存(bounded memory),数据流中的数据常是海量,所以不可能在内存及硬盘上存储整个数据流集;2)一次扫描(single-pass),同样因为数据量巨大,传统的多遍扫描方法不再适用,对其挖掘应该是一个单遍扫描过程,且对流中数据元素的访问只能单次线性(linear scan),即只能按照流入顺序依次读取一次,无法进行随机访问;3)实时响应(real-time response),多数应用要求快速响应,因此挖掘应该是一个连续在线的过程;4)概念漂移(concept-drift detection),数据分布常随着时间的推移而发生变化.

目前,对于数据流聚类算法的研究已在学术界和工业界得到了广泛关注,许多相关算法已被提出[10-26].现有数据流聚类算法均由传统聚类算法扩展而来,根据其所扩展的传统算法不同,我们可以将其分为 5 类:基于划分的方法(STREAM[10]);基于层次的方法(CluStream[11],HPStream[12],SWClustering[13],E-Stream[14],REPSTREAM[15]);基于密度的方法(DenStream[16],ACSC[17],OPTICS-Stream[18],incPre-Decon[19]);基于网格的方法(D-Stream[20],MR-Stream[21],CellTree[22]);基于模型的方法(SWEM[23],GCPSOM[24],G-Stream[25],RPGStream[26]).表 1 分别针对 6 个特性对现有方法进行总结:1)基算法;2)所用计算策略(在线学习或两步学习);3)类簇个数是否自适应;4)是否可挖掘拓扑结构;5)是否可检测概念漂移;6)是否适合高维数据.

如表 1 所示,基于划分的数据流聚类方法相对简单并易于实现,但其需要预先定义类簇个数,然而由于数据分布未知,类簇个数通常无法直接得到.此外,该方法无法检测概念漂移.基于层次的数据流聚类方法虽然能够发现有意义的类簇结构,但其一般具有较高的计算代价,而且对流数据到达的顺序敏感.基于密度的数据流聚类方法可以发现任意形状的类簇,但是算法需要预设较多参数.基于网格的数据流聚类方法运行速度较快,也可以发现任意形状的类簇,但是其聚类质量取决于选取的网格粒度.基于模型的数据流聚类方法包含了很多领域知识并强依赖于假设模型,例如 SWEM 算法基于 EM 模型、GCPSOM 算法基于 SOM 模型、G-Stream 和 RPG-Stream 算法均基于 GNG 模型.从表 1 我们发现,在线(联机)学习算法是处理数据流聚类的一个很好策略,可以解决数据流约束中的一次扫描,实时响应和有限内存问题. STREAM[10], REPSTREAM[15], ACSC[17], incPre-Decon[19], SWEM[23], GCPSOM[24], G-Stream[25]和 RPGStream[26]均为在线学习算法,但只有 REPSTREAM, ACSC, SWEM, GCPSOM, G-Stream 和 RPGStream 可以处理概念漂移,即此算法能够随着数据的流动更新新来的概念并移除旧的概念.GCPSOM, G-Stream 和 RPGStream 不仅可以解决数据流挖掘中的各类约束,同时可以发现数据的拓扑结构,它们分别基于 SOM(self-organizing maps)和 GNG(growing neural gas)模型.但 GCPSOM 和 G-Stream 面对高维数据无能为力,据现有资料显示只有 HPStream[12], incPre-Decon[19]和 RPGStream[26]可以处理高维数据.RPGStream 虽然可以处理高维数据,但因其基于 GNG 模型,超参数较多,调节参数对算法性能影响较大.故本文的直接动机是设计一个可在单机执行、适用于高维数据流的高效数据流聚类算法.

Table 1 Comparison of Various Data Stream Clustering Algorithms

表 1 数据流聚类算法比较

Algorithm	Base Algorithm	Online	Adaptive Cluster Number	Topology	Detect Concept Drift	High-Dimensional
STREAM[10]	k-medians	√	×	×	×	×
CluStream[11]	BIRCH	×	√	×	√	×
HPStream[12]	BIRCH	×	√	×	√	√
SWClustering[13]	BIRCH	×	√	×	√	×
E-Stream[14]	BIRCH	×	√	×	√	×
REPSTREAM[15]	CHAMELEON	√	√	×	√	×
DenStream[16]	DBSCAN	×	√	×	√	×
ACSC[17]	DBSCAN	√	√	×	√	×
OPTICS-Stream[18]	OPTICS	×	√	×	√	×
incPre-Decon[19]	PreDecon	√	√	×	×	√
D-Stream[20]	DENCLUE	×	√	×	√	×
MR-Stream[21]	STING	×	√	×	√	×
CellTree[22]	STING	×	√	×	√	×
SWEM[23]	EM	√	√	×	√	×
GCPSOM[24]	SOM	√	√	√	√	×
G-Stream[25]	GNG	√	√	√	√	×
RPGStream[26]	GNG	√	√	√	√	√

目前解决高维问题的基本方法大致有 2 种:1)特征选择[27-28],即从样本集中选取重要的特征子集,不改变原有特征空间;2)特征提取[29-30],主要通过属性间关系,如组合不同属性得到新的属性,改变了原有特征空间.其中特征提取可进一步分为 2 类,即数据依赖(data-dependent)和数据独立(data-independent).例如主成分分析(principal component analysis, PCA)作为一种数据依赖的方法,虽可以用于数据流聚类,但时间复杂度通常较高.而随机投影(random projection)作为一种数据独立的方法[31-32],将高维欧氏空间里的点集映射到低维空间,使得相对距离得到一定误差范围内的保持.考虑到数据流本身的特性,如无界性和动态变化,随机投影可以独立于数据,无疑是高维数据流聚类的首选方法.更重要的是,随机投影不仅计算效率高,而且可以通过使用随机矩阵,例如高斯随机矩阵等,有效降低高维数据的维度.低计算复杂度和对度量结构的近似表达使得将随机投影应用到聚类分析更加有效.近来,许多工作聚焦于将随机投影用于聚类问题.文献[33]将随机投影与 K-Means 算法结合聚类;文献[34]证明了任何含有 n 个样本 d 维特征(矩阵 $\boldsymbol{A}=\boldsymbol{R}^{n\times d}$)的集合都可以被投影到 $t=\Omega(k/\varepsilon^2)$ 维空间,对于任何 $\varepsilon\in(0,1/3)$,在 $O(nd\times\lceil\varepsilon^2 k/\log(d)\rceil)$ 时间下,$2+\varepsilon$ 范围内最优 k-划分被保留.文献[35-36]给出了如何利用 JL 引理(Johnson-Lindenstrauss)将数据投影到 $O(nd\times\lceil\log(k)/\varepsilon^2\rceil)$ 维,并完成一个 $(9+\varepsilon)$ 的近似保持.然而他们均聚焦于处理高维数据(d 很大),并没有考虑 n 也很大的场景.此外,基于随机投影,文献[37]探索了基于 SLC 和 ALC 链接以及最小扩展树(minimum spanning tree, MST)的快速层次聚类.同时文献[38]将随机投影与密度聚类相结合.文献[39-41]使用随机投影集成和迭代的方法对高维数据聚类.

为解决高维数据流聚类问题(n 和 d 均很大),本文提出了一种基于随机投影的高维数据流聚类算法 RPFART.首先通过随机投影将原始高维数据映射到低维数据空间,再使用 ART 模型[42]进行数据流聚类.ART 具有线性计算复杂度,且仅使用 1 个超参数,并对参数设置鲁棒.虽然将随机投影用于 K-Means 算法可以分析理论最差界,但由于 ART 本身的复杂性我们无法分析 RPFART 的最差界.所以,本文中我们使用大量实验分析 RPFART 算法的聚类性能.多个数据集上的实验结果表明:即使原始尺寸压缩到 10%,RPFART 算法仍可以达到与 RPGStream 算法相当甚至更好的性能.对于 ACT1 数据集,其维数从 67 500 减少到 6 750.

1 相关工作

1.1 数据流聚类与 ART

设数据流 $\mathcal{DS}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 为一个带有时间戳(time stamp)的多维数据点集合,实际应用中 n 的取值可以无限大,其中每个数据点 $\boldsymbol{x}_i=(x_i^1,x_i^2,\cdots,x_i^d)$ 是一个包含 d 维的数据记录,其到达时间为 t_i.数据流本身具有无限、时序、动态性.数据流聚类将数据对象集合中相似对象划分为一个或多个簇,划分后同一簇中元素彼此相似,但与其他簇中元素相异.

自组织神经网络是人工智能领域应用最为广泛的一种学习模型.为解决大部分神经网络模型遭遇的"稳定性-弹性问题",美国 Boston 大学的 Grossberg 和 Carpenter 于 1976 年提出了一种无监督竞争型神经网络模型,即自适应谐振理论网络(adaptive resonance theory, ART)[43],可在稳定原有模式类的前提下学习新的模式.ART 模拟了人类大脑如何捕捉、识别、记忆关于事物和事件的信息.随着理论的不断完善,学者们提出了大量基于 ART 改进的无监督学习模型,如 ART1[44],ART2[45],ART2-A[46],ART3[47] 和模糊 ART(fuzzy ART)[42].模糊 ART 通过在类别空间实时搜索和匹配现有类簇,增长式地逐步处理每一个输入模式,是本文研究的基本模型.警戒参数(vigilance parameter)用于约束在同一个类簇中模式的最小相似度.当输入模式与现有类簇都不相似时,则生成一个新的类簇来编码这个新模式.模糊 ART 已用于解决图像和文本挖掘问题,如 Web 文档管理、基于标记的 Web 图像组织、图像-文本关联,但还未用于数据流聚类.

模糊 ART 模型由输入层 F1 和识别层 F2 组成,如图 1 所示.输入层 F1 包含的输入向量 $\boldsymbol{I}$ 被提交到网络,与识别层 F2 中各个类簇的权值向量进行相似度比较并归类.

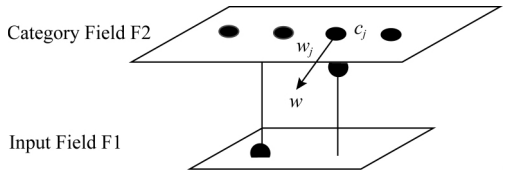


Fig.1 Fuzzy ART architecture

图 1 模糊 ART 结构

1) 输入向量(input vector).设 $\boldsymbol{I}=\boldsymbol{x}_i$ 表示输入层 F1 的输入模式,其中 $\boldsymbol{x}_i=(x_i^1,x_i^2,\cdots,x_i^d)$,$x_i^m\in[0,1]$($m=1,2,\cdots,d$).通过补编码(complement coding),$\boldsymbol{x}_i$ 与它的补向量 $\boldsymbol{x}_i=1-\bar{\boldsymbol{x}}_i$ 共同构成了 $\boldsymbol{I}=(\boldsymbol{x}_i,\bar{\boldsymbol{x}}_i)$.

2) 权值向量(weight vector).设 w_j 表示识别层 F2 中第 j 个类 c_j($j=1,2,\cdots,J$)的权值.

3) 参数(parameter).模糊 ART 随着 3 个参数动态改变,它们分别是选择参数 $\alpha>0$、学习参数 $\beta\in[0,1]$、以及警戒参数 $\rho\in[0,1]$.

模糊 ART 聚类过程包含 3 个关键步骤:

步骤 1. 类别选择(category choice).对每个输入模式 $\boldsymbol{I}$,模糊 ART 根据选择函数计算此输入对识别层 F2 中的每个类簇的选择值,并取具有最大值的类簇作为获胜类簇 c_{j*}.第 j 个类簇 c_j 的选择函数定义为

$$T_j=\frac{|\boldsymbol{I}\wedge w_j|}{\alpha+|w_j|}. \tag{1}$$

这里,模糊与操作 $\wedge$ 定义为 $(p\wedge q)_i\equiv\min(p_i,q_i)$,范式 $|\cdot|$ 定义为 $|p|\equiv\sum_i p_i$.

步骤 2. 模板匹配(template matching).输入模式 $\boldsymbol{I}$ 与获胜类簇 c_{j*} 使用匹配函数 M_{j*} 进行评估,M_{j*} 定义为

$$M_{j*}=\frac{|\boldsymbol{I}\wedge w_{j*}|}{|\boldsymbol{I}|}. \tag{2}$$

如果获胜类簇 c_{j*} 的 $M_{j*}\geqslant\rho$,则发生共振(resonance),引发步骤 3——中心学习.否则,返回步骤 2,继续在剩下的类簇中寻找一个获胜类簇.如果所有选出的获胜类簇均不满足 $M_{j*}\geqslant\rho$,则生成一个新的类簇来编码这个输入模式 $\boldsymbol{I}$.

步骤 3. 中心学习(prototype learning).如果获胜类簇 c_{j*} 的 $M_{j*}\geqslant\rho$,根据式(3)更新其权值向量 w_{j*}.

$$w_{j*}^{(\text{new})}=\beta(\boldsymbol{I}\wedge w_{j*})+(1-\beta)w_{j*}. \tag{3}$$

1.2 随机投影

随机投影中,原始的 d 维数据通过随机矩阵 $\boldsymbol{R}_{d\times d_c}$ 被投影到 d_c($d_c\ll d$)维子空间.样本矩阵 $\boldsymbol{X}_{n\times d}$($n$ 个样本,d 维特征)投影到 $\boldsymbol{X}_{n\times d_c}^{RP}$:

$$\boldsymbol{X}_{n\times d_c}^{RP}=\boldsymbol{X}_{n\times d}\boldsymbol{R}_{d\times d_c}. \tag{4}$$

随机投影的理论依据是 JL 引理[31]:高维欧氏空间里的点集映射到低维空间,其相对距离得到一定误差范围内的保持.

定理 1[41]. 设样本矩阵 $\boldsymbol{X}\in\mathbb{R}^{n\times d}$ 包含 n 个样本 d 维特征.给定 $\varepsilon,\beta>0$,则:

$$k_0=\frac{4+2\beta}{\varepsilon^2/2-\varepsilon^3/3}\log n. \tag{5}$$

这里参数 ε 控制距离保持的精度,β 控制投影成功的概率.d_c 是一个正整数,且 $d_c \geqslant k_0$,随机矩阵 $\boldsymbol{R}$ 是一个 $d \times d_c$ 矩阵,$R(i,j)=r_{ij}$,r_{ij} 是一个独立的随机变量,可以由 3 种概率分布生成:

$$r_{ij} \sim N(0,1); \tag{6}$$

$$r_{ij}=\begin{cases}+1, \text{概率为 } 1/2,\\ -1, \text{概率为 } 1/2;\end{cases} \tag{7}$$

$$r_{ij}=\sqrt{3}\times\begin{cases}+1, \text{概率为 } 1/6,\\ 0, \text{概率为 } 2/3,\\ -1, \text{概率为 } 1/6.\end{cases} \tag{8}$$

设 $E=\frac{1}{\sqrt{d_c}}\boldsymbol{XR}$,$f:\mathbb{R}^d \to \mathbb{R}^{d_c}$ 将 $\boldsymbol{X}$ 的第 i 行映射到 E 的第 i 行.

对所有的 $\boldsymbol{u},\boldsymbol{v} \in \boldsymbol{X}$,在至少 $1-n^{-\beta}$ 概率下:

$$(1-\varepsilon)\|\boldsymbol{u}-\boldsymbol{v}\|^2 \leqslant \|f(\boldsymbol{u})-f(\boldsymbol{v})\|^2 \leqslant (1+\varepsilon)\|\boldsymbol{u}-\boldsymbol{v}\|^2. \tag{9}$$

从式(9)可以看出,理论上 JL 界(k_0)不依赖于原始空间的维度 d,为了得到定理 1 的结果,我们只需要通过一个简单的概率分布生成随机矩阵 $\boldsymbol{R}$,同时进行投影计算.

通过假设输入数据的期望为 0,在主成分分析的激励下,文献[41]给出结论:根据概率上的方差分析,压缩后的数据获得了原始数据的全部可变性.首先,压缩后的数据可以从低维数据中获得很多信息,因为这些低维都是线性无关的.其次,原始数据维度的方差之和等于投影后数据维度的方差之和.

定理 2[41]. 设矩阵 $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ 是包含 n 个 d 维独立样本的随机向量$(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_d)$,$\boldsymbol{S}$ 表示 $\boldsymbol{X}$ 的样本协方差矩阵.一个随机投影矩阵 $\boldsymbol{R} \in \mathbb{R}^{d \times d_c}$ 将 d 维随机向量映射到 d_c 维随机向量 $\boldsymbol{Y}$,$(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_{d_c})=(1/\sqrt{d_c})(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_d)\boldsymbol{R}$,$\boldsymbol{S}^*$ 表示的投影后数据 $\boldsymbol{Y}$ 的协方差矩阵.如果随机投影矩阵 $\boldsymbol{R}$ 服从定理 1 所要求的分布,且与随机向量$(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_d)$相互独立,则:

1) 投影数据的维度是相互独立的.

$Cov(Y_i,Y_j)=0, \forall i \neq j$;

2) 随机投影保持了可变性.

$\sum_{i=1}^{d_c} var(Y_i)=\sum_{i=1}^{d} var(X_i)$;当 $d_c \to \infty$,概率为 1,$tr(\boldsymbol{S}^*)=tr(\boldsymbol{S})$.

① http://www.cse.fau.edu/~xqzhu/stream.html

② http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data

③ https://archive.ics.uci.edu/ml/datasets/Covertype

④ http://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities

利用上述性质,已有相关工作验证了将随机投影应用于聚类问题的可行性.Boutsidis 等人[33]首次将随机投影与 k-Means 结合进行聚类;吴等人[34]和 Schneider 等人[37]针对 SLC 和 ALC 聚类以及最小生成树(MST)问题,探索了基于随机投影的快速层次聚类算法.同时 Schneider 和 Vlachos[38]通过使用随机投影来扩展基于密度的聚类,并提出了显著提高学习效率的算法.Ferns 和 Brodley[39]、Cardoso 和 Wichert[40]、叶等人[41]提出了使用随机投影对高维数据聚类的集成模型和迭代模型.

2 基于随机投影的数据流聚类算法

2.1 RPFART 算法

鉴于随机投影的优良性质,我们提出了一种基于随机投影的数据流聚类算法 RPFART.直觉上,通过在低维空间对数据进行聚类,不仅可以大大降低每次模糊 ART 迭代的计算成本,还可以在原始高维空间中找到与之相关的解决方案.假设有一个数据流序列 $\mathcal{DS}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$(实际应用中 n 的取值可以无限大),其中每个数据点 $\boldsymbol{x}_i=(x_i^1,x_i^2,\cdots,x_i^d)$是一个包含 d 维的数据记录,其到达时间为 t_i.RPFART 生成了包含一系列节点(nodes)的结构,其中节点代表类簇(cluster),且每个节点均包含相关的权值向量 $\boldsymbol{w}_c=(w_c^1,w_c^2,\cdots,w_c^d)$.

RPFART 算法首先依据 d_c 的大小生成随机投影矩阵 $\boldsymbol{R}$,当一个新的数据点 x_i 到达时,用 $\boldsymbol{R}$ 将 x_i 投影到对应的 d_c 维($d_c \ll d$)点 y_i.从而整个数据流 $\mathcal{DS}_{n \times d}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$被投影到 $\mathcal{Y}_{n \times d_c}=\{y_1,y_2,\cdots,y_n\}$.再对 $\mathcal{Y}_{n \times d_c}$ 使用模糊 ART 进行聚类.具体见算法 1.

算法 1. RPFART 算法.

输入:$\mathcal{DS}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$;

输出:节点集合 $C=\{c_1,c_2,c_3,\cdots\}$及其权值 $\mathcal{W}=\{w_{c_1},w_{c_2},w_{c_3},\cdots\}$.

① 生成满足定理 1 的随机矩阵 $\boldsymbol{R}_{d \times d_c}$($d_c \ll d$);

② for each $\boldsymbol{x}_i$

③ $y_i=\boldsymbol{x}_i \times \boldsymbol{R}$;

④ 对 y_i 使用模糊 ART 算法进行聚类;

⑤ end for

2.2 算法复杂度分析

算法 1 中最耗时的运算是步骤③和步骤④.可以看出随机投影在计算上非常简单,可以快速生成,故步骤③的投影时间复杂度为 $O(ndd_c)$.模糊 ART 算法包含了类别选择、模板匹配、中心学习 3 个主要步骤,每个步骤的时间复杂度分别为 $O(d)$, $O(md)$,$O(d)$,给定 n 个输入样本,总体时间复杂度为 $O(nmd)$,其中 m 是聚类结果中节点个数.故步骤④的时间复杂度为 $O(nmd_c)$.RPFART 算法的总体时间复杂度为 $O(ndd_c+nmd_c)$.

3 实验与结果

为了验证本文提出算法的有效性,我们在 5 个数据集上与现有数据流聚类算法 RPGStream 进行了比较.实验使用的计算机配置为 Intel Core i5-6300U 2.4 GHz 处理器和 8 GB 内存,Windows 10 操作系统,所有比较程序均在 MATLAB R2015a 上设计和运行.

3.1 聚类评价指标

为了对各种聚类算法性能进行评价,我们引入了 3 项评价指标[26]:1) accuracy (purity);2) NMI (normalized mutual information);3)RI(rand index).

1) Accuracy(Purity)

$$Acc=\frac{1}{K}\sum_{i=1}^{K}\frac{|C_i^d|}{|C_i|}\times 100\%, \tag{10}$$

其中,K 表示类簇个数,$|C_i^d|$ 表示在类簇 i 中的样本点数,$|C_i|$ 表示类簇 i 中真实的样本个数.因此,Accuracy 度量了聚类的平均纯度,Accuracy 越大表明聚类纯度越高.其取值范围在 0～1 之间.

2) NMI(normalized mutual information)

归一化互信息 NMI 是一个量化 2 个分布之间共享统计信息的对称策略.当类簇标签和真实样本类别一对一映射时 NMI 值到达最大值 1.0.给定真实类簇 $A=\{A_1,A_2,\cdots,A_k\}$和某聚类算法得到的类簇 $B=\{B_1,B_2,\cdots,B_h\}$,混淆矩阵 $\boldsymbol{C}$ 中的元素 C_{ij} 表示即在 A_i 又在 B_j 中的样本个数.NMI 计算为

$$NMI(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A}C_{i.}\log(C_{i.}/N)+\sum_{j=1}^{C_B}C_{.j}\log(C_{.j}/N)}, \tag{11}$$

其中,C_A(C_B)表示 A(B)中样本个数,$C_{i.}$($C_{.j}$)表示 C 中 i 行元素和(C 中 j 列元素和),N 表示样本个数.

3) RI(rand index)

RI 比较 $n\times(n-1)/2$ 个数据对,其中 n 为数据集中样本个数,P_1,P_2 为 2 种聚类算法,n_{11}为数据对(x_i,x_j)在 P_1,P_2 中划分为同一类的数据对数,n_{00}则为(x_i,x_j)隶属不同类的数据对数,RI 错误率计算为

$$RI=\frac{n_{11}+n_{00}}{C_n^2}. \tag{12}$$

由式(12)可得 $RI\in[0,1]$,当 P_1 与 P_2 划分完全一致时 $RI=1$.

3.2 数据集和参数设置

为了对 RPFART 算法的聚类有效性进行评价,实验中我们使用了人工和真实数据集,表 2 给出数据集的相关信息:

Table 2 Statistics of Five Datasets

表 2 数据集

Data Set	# Samples	# Features	# Classes
HyperPlan①	100 000	10	5
KddCup99②	494 021	41	23
CoverType③	581 012	54	7
ACT2④	9 120	5 625	19
ACT1	760	67 500	19

HyperPlan 数据集是人工模拟数据集.HyperPlan 是一个含有概念漂移的数据流,包含 5 个类共 10 万个样本,每个样本 5 维.KddCup99,CoverType 和 ACT 均来自 UCI.KddCup99 数据集最早来源于 MIT 林肯实验室的一项入侵检测评估项目,记录了 9 周内 TCP 网络连接和系统审计数据,仿真各种不同的用户类型、网络流量和攻击手段.这些原始数据包含约 50 万条连接记录的训练集.每个连接记录包含 41 个属性,这些连接记录含 1 种正常的标识类型和 22 种训练攻击类型共 23 个类别.CoverType 数据集来源于 US Geological Survey (USGS)和 US Forest Service (USFS)对位于 Roosevelt 国家森林的四片荒野区域的观测.数据集中包含 581 012 条记录,这些记录最终被分为 7 种类型.每条观测记录包含 54 个地质学和地理学属性.ACT(The Daily and Sports Activities Data Set)数据集包含 45 个传感器在 5 min 内以 25 Hz 的采样频率收集的 19 项活动的数据.为了获得高维数据集,我们分别将 1 min 和 5 s 的活动数据处理为一个样本,结果得到了 760×67 500(ACT1)和 9 120×5 625(ACT2)数据矩阵.

由算法 1 所示,RPFART 算法需要设置警戒参数 ρ 和压缩率 r.RPGStream 算法设置 $\varepsilon_b=0.01$、

$\varepsilon_n=0.001$、$\beta=300$(ACT1 取 30)、$\lambda_1=0.2$、$\lambda_2=0.2$、$|windows|=600$(ACT1 取 60)、$|reservoir|=400$(ACT1 取 50)、$agemax=250$(代表边年龄的最大值)、$weightmin=2$(代表神经元节点权值的最小值),并且每次插入新节点的个数 $NbNodesInserted=3$.

3.3 聚类性能比较

首先评估 RPFART 的聚类质量,并将其与 RPGStream 算法在 5 个数据集上进行比较.每个算法重复实验 10 次.聚类结果如表 3～5 所示.参数 r 表示压缩率,例如对于 KddCup99 数据集,$r=90\%$ 表示通过随机投影将特征数减少到 $54\times0.9=48$.

从表 3～5 中我们可以发现:1)RPFART 在使用了随机投影后,总体上与 RPGStream 的结果相当,特别是 *NMI* 和 Rand 指数在所有数据集上均超过了 RPGStream.2)RPFART 在 HyperPlan 和 ACT2 数据集上的聚类纯度略低于 RPGStream.3)即使设置一个小的 r,例如 $r=10\%$,RPFART 在 ACT1 上的

Table 3 The Comparison Results of RPFART and RPGStream in Terms of Accuracy

表 3 RPFART(RPF)和 RPGStream(RPG)在不同数据集上的聚类性能 Accuracy 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	0.368 1±0.000 0	0.370 9±0.000 0	0.389 7±0.000 0	0.383 9±0.000 0	0.385 6±0.000 0
	RPG	**0.424 3±0.000 0**	**0.405 8±0.000 0**	**0.408 5±0.000 0**	**0.396 1±0.000 0**	**0.394 9±0.000 0**
KddCup99	RPF	**0.982 6±0.000 0**	**0.981 0±0.000 0**	**0.978 2±0.000 0**	**0.980 2±0.000 0**	**0.976 8±0.000 0**
	RPG	0.981 9±0.000 0	0.979 5±0.000 0	0.977 5±0.000 0	0.979 3±0.000 0	0.976 7±0.000 0
CoverType	RPF	**0.539 5±0.000 0**	**0.543 5±0.000 0**	**0.559 0±0.000 0**	**0.580 3±0.000 0**	**0.575 8±0.000 0**
	RPG	0.521 3±0.000 0	0.521 3±0.000 0	0.520 3±0.000 0	0.522 5±0.000 0	0.525 3±0.000 0
ACT2	RPF	0.556 9±0.000 2	0.537 6±0.000 2	0.532 0±0.000 2	0.547 3±0.000 2	0.551 8±0.000 2
	RPG	**0.654 1±0.000 2**	**0.653 6±0.000 3**	**0.659 1±0.000 2**	**0.658 1±0.000 4**	**0.647 5±0.000 3**
ACT1	RPF	**0.832 9±0.000 3**	**0.819 7±0.000 3**	**0.826 3±0.000 5**	**0.823 7±0.000 3**	**0.809 2±0.000 3**
	RPG	0.520 3±0.000 5	0.515 1±0.000 2	0.516 7±0.000 7	0.520 4±0.000 4	0.499 6±0.001 7

Note: The best results are shown in bold.

Table 4 The Comparison Results of RPFART and RPGStream in Terms of *NMI*

表 4 RPFART(RPF)和 RPGStream(RPG)在不同数据集上的聚类性能 *NMI* 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	**0.666 7±0.000 0**	**0.457 0±0.000 0**	**0.343 5±0.000 0**	**0.162 3±0.000 0**	**0.006 5±0.000 0**
	RPG	0.018 1±0.000 0	0.010 8±0.000 0	0.013 8±0.000 0	0.006 2±0.000 0	0.006 4±0.000 0
KddCup99	RPF	**0.772 6±0.000 3**	**0.741 1±0.000 4**	**0.751 0±0.000 3**	**0.705 9±0.000 3**	**0.692 5±0.000 3**
	RPG	0.646 7±0.000 3	0.642 7±0.000 4	0.646 4±0.000 2	0.637 6±0.000 3	0.629 3±0.000 5
CoverType	RPF	**0.353 9±0.000 0**	**0.343 0±0.000 0**	**0.245 7±0.000 0**	**0.163 6±0.000 0**	**0.161 1±0.000 0**
	RPG	0.093 4±0.000 0	0.093 5±0.000 0	0.092 8±0.000 0	0.095 5±0.000 1	0.091 0±0.000 2
ACT2	RPF	**0.735 3±0.000 0**	**0.720 7±0.000 0**	**0.740 3±0.000 0**	**0.732 8±0.000 0**	**0.727 0±0.000 0**
	RPG	0.599 7±0.000 0	0.595 9±0.000 0	0.599 6±0.000 0	0.598 3±0.000 0	0.585 0±0.000 0
ACT1	RPF	**0.748 1±0.000 0**	**0.735 7±0.000 0**	**0.741 3±0.000 0**	**0.741 0±0.000 0**	**0.737 8±0.000 0**
	RPG	0.576 2±0.000 3	0.574 9±0.000 2	0.570 8±0.000 4	0.576 7±0.000 0	0.561 4±0.000 6

Note: The best results are shown in bold.

Table 5 The Comparison Results of RPFART and RPGStream in Terms of Rand Index

表 5 RPFART(RPF)和 RPGStream(RPG)在不同数据集上的聚类性能 RI 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	**1.000 0±0.000 0**	**0.989 7±0.000 0**	**0.967 0±0.000 0**	**0.877 1±0.000 0**	0.558 9±0.000 0
	RPG	0.703 1±0.000 0	0.702 6±0.000 0	0.703 3±0.000 0	0.702 6±0.000 0	**0.701 8±0.000 0**
KddCup99	RPF	**0.962 0±0.000 4**	**0.940 6±0.000 5**	**0.945 7±0.000 4**	**0.862 1±0.000 4**	**0.884 2±0.000 4**
	RPG	0.823 2±0.000 4	0.815 5±0.000 5	0.818 4±0.000 2	0.814 2±0.000 6	0.811 0±0.000 8
CoverType	RPF	**0.909 2±0.000 0**	**0.901 6±0.000 0**	**0.805 0±0.000 0**	**0.690 5±0.000 0**	**0.680 9±0.000 0**
	RPG	0.615 5±0.000 0	0.615 7±0.000 0	0.615 2±0.000 0	0.616 4±0.000 0	0.615 2±0.000 0
ACT2	RPF	**0.978 0±0.000 0**	**0.977 2±0.000 0**	**0.974 7±0.000 0**	**0.976 3±0.000 0**	**0.977 4±0.000 0**
	RPG	0.947 7±0.000 0	0.947 4±0.000 0	0.947 7±0.000 0	0.947 4±0.000 0	0.947 7±0.000 0
ACT1	RPF	**0.961 0±0.000 0**	**0.959 5±0.000 0**	**0.959 9±0.000 0**	**0.960 4±0.000 0**	**0.958 7±0.000 0**
	RPG	0.929 8±0.000 0	0.926 9±0.000 0	0.927 9±0.000 0	0.925 9±0.000 0	0.921 3±0.000 0

Note: The best results are shown in bold.

聚类纯度、*NMI* 和 Rand 指数仍然是最好的.4)我们的算法不限于海量数据,即使对高维小样本也可以得到很好的结果,如在 ACT1 数据集上取得了较好的效果.

为了进一步证明 RPFART 的性能,我们分别在 5 个数据集上与离线聚类算法 RPK-Means 和 PCAFART 进行了比较.PCAFART 算法是将模糊 ART 与 PCA 结合对数据流进行聚类.其结果如表 6～8 所示.

从表 6～8 中我们可以发现:1)RPFART 在精度、*NMI*、Rand 指数都优于 KddCup99, CoverType, ACT2, ACT1 数据集上的 RPK-Means.2)RPFART 与 PCAFART 具有相当的聚类结果,但后者在处理高维 ACT1 数据集时,出现内存耗尽溢出问题.因此,

Table 6 The Comparison Results of RPFART, RPK-Means and PCAFART in Terms of Accuracy

表 6 RPFART(RPF)和 RPK-Means(RPK), PCAFART(PCAF)在不同数据集上的聚类性能 Accuracy 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	0.368 1±0.000 0	0.370 9±0.000 0	**0.389 7±0.000 0**	0.383 9±0.000 0	0.385 6±0.000 0
	RPK	**0.415 3±0.000 0**	**0.382 4±0.000 0**	0.380 3±0.000 0	0.379 3±0.000 0	**0.395 9±0.000 0**
	PCAF	0.367 9±0.000 0	0.368 9±0.000 0	0.382 2±0.000 0	**0.399 3±0.000 0**	0.377 7±0.000 0
KddCup99	RPF	**0.982 6±0.000 0**	**0.981 0±0.000 0**	**0.978 2±0.000 0**	**0.980 2±0.000 0**	**0.976 8±0.000 0**
	RPK	0.814 5±0.000 1	0.812 2±0.000 0	0.814 2±0.000 0	0.754 0±0.009 7	0.785 5±0.005 9
	PCAF	0.874 6±0.000 0	0.878 6±0.000 0	0.901 9±0.000 0	0.791 9±0.000 0	0.569 0±0.000 0
CoverType	RPF	**0.539 5±0.000 0**	**0.543 5±0.000 0**	**0.559 0±0.000 0**	**0.580 3±0.000 0**	**0.575 8±0.000 0**
	RPK	0.520 3±0.000 0	0.520 9±0.000 0	0.513 7±0.000 0	0.521 4±0.000 0	0.524 9±0.000 0
	PCAF	0.502 0±0.000 0	0.495 3±0.000 0	0.489 9±0.000 0	0.478 9±0.000 0	0.500 7±0.000 0
ACT2	RPF	**0.556 9±0.000 2**	**0.537 6±0.000 2**	**0.532 0±0.000 2**	**0.547 3±0.000 2**	**0.551 8±0.000 2**
	RPK	0.438 8±0.001 8	0.435 5±0.001 0	0.457 0±0.000 6	0.446 4±0.000 9	0.434 0±0.000 9
	PCAF	0.256 1± 0.000 3	0.236 0±0.000 3	0.191 6±0.000 3	0.268 8±0.000 3	0.434 3±0.000 3
ACT1	RPF	**0.832 9±0.000 3**	**0.819 7±0.000 3**	**0.826 3±0.000 5**	**0.823 7±0.000 3**	**0.809 2±0.000 3**
	RPK	0.400 7±0.000 8	0.414 1±0.001 7	0.376 2±0.002 0	0.393 2±0.000 8	0.391 3±0.002 7
	PCAF			Out of memory		

Note: The best results are shown in bold.

Table 7 The Comparison Results of RPFART, RPK-Means and PCAFART in Terms of *NMI*

表 7 RPFART(RPF)和 RPK-Means(RPK), PCAFART(PCAF)在不同数据集上的聚类性能 *NMI* 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	**0.666 7±0.000 0**	0.457 0±0.000 0	**0.343 5±0.000 0**	**0.162 3±0.000 0**	0.006 5±0.000 0
	RPK	0.017 1±0.000 0	0.003 6±0.000 0	0.003 3±0.000 0	0.003 0±0.000 0	**0.009 3±0.000 0**
	PCAF	0.649 2±0.000 0	**0.527 6±0.000 0**	0.295 2±0.000 0	0.027 8±0.000 0	0.004 1±0.000 0
KddCup99	RPF	**0.772 6±0.000 3**	**0.741 1±0.000 4**	**0.751 0±0.000 3**	**0.705 9±0.000 3**	**0.692 5±0.000 3**
	RPK	0.643 7±0.003 1	0.648 2±0.002 8	0.645 5±0.000 7	0.531 0±0.051 8	0.534 6±0.027 2
	PCAF	0.504 8±0.000 3	0.536 7±0.000 4	0.564 3±0.000 3	0.479 1±0.000 3	0.018 3±0.000 3
CoverType	RPF	0.353 9±0.000 0	0.343 0±0.000 0	0.245 7±0.000 0	0.163 6±0.000 0	0.161 1±0.000 0
	RPK	0.088 6±0.000 0	0.083 6±0.000 1	0.081 3±0.000 2	0.089 9±0.000 2	0.092 3±0.000 4
	PCAF	**0.493 5±0.000 3**	**0.528 5±0.000 3**	**0.559 8±0.000 3**	**0.459 8±0.000 3**	**0.432 1±0.000 3**
ACT2	RPF	**0.735 3±0.000 0**	**0.720 7±0.000 0**	**0.740 3±0.000 0**	**0.732 8±0.000 0**	**0.727 0±0.000 0**
	RPK	0.581 3±0.000 7	0.575 2±0.000 6	0.587 6±0.000 2	0.574 8±0.000 5	0.577 2±0.000 3
	PCAF	0.707 4±0.000 0	0.689 8±0.000 0	0.591 5±0.000 0	0.599 1±0.000 0	0.667 1±0.000 0
ACT1	RPF	**0.748 1±0.000 0**	**0.735 7±0.000 0**	**0.741 3±0.000 0**	**0.741 0±0.000 0**	**0.737 8±0.000 0**
	RPK	0.575 4±0.000 3	0.573 5±0.000 4	0.574 2±0.000 7	0.579 3±0.000 3	0.560 4±0.001 5
	PCAF			Out of memory		

Note: The best results are shown in bold.

Table 8 The Comparison Results of RPFART, RPK-Means and PCAFART in Terms of RI

表 8 RPFART(RPF)和 RPK-Means(RPK), PCAFART(PCAF)在不同数据集上的聚类性能 RI 比较

Datasets	Algorithms	Rate r/%				
		90	70	50	30	10
HyperPlan	RPF	**1.000 0±0.000 0**	0.989 7±0.000 0	**0.967 0±0.000 0**	**0.877 1±0.000 0**	0.558 9±0.000 0
	RPK	0.682 2±0.000 0	0.681 7±0.000 0	0.682 0±0.000 0	0.680 8±0.000 0	**0.681 1±0.000 0**
	PCAF	0.999 9±0.000 0	**0.998 5±0.000 0**	0.957 7±0.000 0	0.724 3±0.000 0	0.573 7±0.000 0
KddCup99	RPF	**0.962 0±0.000 4**	**0.940 6±0.000 5**	**0.945 7±0.000 4**	**0.862 1±0.000 4**	**0.884 2±0.000 4**
	RPK	0.814 5±0.002 6	0.810 0±0.002 1	0.812 9±0.002 2	0.752 5±0.032 6	0.760 0±0.016 1
	PCAF	0.815 6±0.000 4	0.834 3±0.000 5	0.845 6±0.000 4	0.746 8±0.000 4	0.409 9±0.000 4
CoverType	RPF	0.909 2±0.000 0	0.901 6±0.000 0	0.805 0±0.000 0	0.690 5±0.000 0	0.680 9±0.000 0
	RPK	0.609 7±0.000 0	0.609 2±0.000 0	0.609 6±0.000 0	0.609 8±0.000 0	0.610 2±0.000 0
	PCAF	**0.973 7±0.000 0**	**0.987 9±0.000 0**	**0.993 3±0.000 0**	**0.970 3±0.000 0**	**0.960 6±0.000 0**
ACT2	RPF	0.978 0±0.000 0	0.977 2±0.000 0	**0.974 7±0.000 0**	**0.976 3±0.000 0**	**0.977 4±0.000 0**
	RPK	0.850 5±0.000 5	0.852 4±0.001 0	0.868 1±0.000 7	0.859 3±0.000 9	0.855 3±0.000 8
	PCAF	**0.988 1±0.000 0**	**0.981 9±0.000 0**	0.951 7±0.000 0	0.953 7±0.000 0	0.954 2±0.000 0
ACT1	RPF	**0.961 0±0.000 0**	**0.959 5±0.000 0**	**0.959 9±0.000 0**	**0.960 4±0.000 0**	**0.958 7±0.000 0**
	RPK	0.830 0±0.001 1	0.840 1±0.001 1	0.828 6±0.000 6	0.828 1±0.000 2	0.855 3±0.000 8
	PCAF			Out of memory		

Note: The best results are shown in bold.

我们可以得出 RPFART 更适合于高维数据,同时结合随机投影和模糊 ART 是可行和有前途的.此外,RPFART 的方差比其他方法略小,这表明它相对稳定.除了随机投影,PCA 也可以用来降维,但当我们想

将 PCA 与模糊 ART 结合用于 ACT1 数据集时,算法因为内存不足而停止.

3.4 运行时间比较

图 2 显示了 $r=50\%$时的 5 个数据集上 RPFART 和 RPGStream 的运行时间.

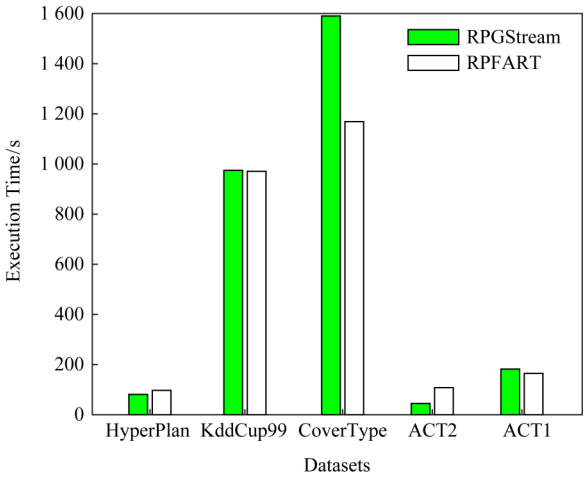


Fig. 2 Execution time(in seconds)

图 2 运行时间比较

从图 2 可以看出:1)RPFART 和 RPGStream 的执行时间都随着数据量的增加而增加.2)随着样本数的增加 RPFART 比 RPGStream 更快.研究表明,RPFART 算法对大规模、高维数据的处理效率更高.

3.5 随机矩阵的选择

由于 RPFART 是基于随机投影的,所以直观地说,随机矩阵 R 的类型选择将在一定程度上对聚类性能产生影响.因此,为分析其影响,我们使用不同类型的随机矩阵进行实验,典型的有高斯分布(式(6))、均匀分布(式(7))和稀疏分布(式(8))随机矩阵.我们还利用 Gram Schmidt 方法对上述随机矩阵进行正交实验.所有实验均在 HyperPlan,Kddcup99,CoverType,ACT2 上进行,重复 10 次,$r=50\%$.结果如表 9 所示.结果表明,正交后 RPFART 算法在聚类纯度、*NMI* 和 Rand 指数上比非正交算法较优.然而,正交化并不免费,计算上十分昂贵.有趣的是,Hecht-Nielsen[48] 证明高维空间中存在大量几乎正交(而不是严格正交)的方向,即具有随机方向的一系列向量同样可以是有效正交,从而其可作为一组基的近似.

Table 9 Performance of Clustering Algorithms with Different Random Matrix

表 9 不同随机矩阵对 RPFART 的影响

Datasets	Random Matrix	Accuracy	*NMI*	Rand Index
HyperPlan	GaussRP	**0.389 7±0.000 0**	0.343 5±0.000 0	0.967 0±0.000 0
	GaussRP-Orth	0.374 9±0.000 0	**0.347 5±0.000 0**	**0.975 0±0.000 0**
	uniformRP	**0.388 8±0.000 0**	0.211 5±0.000 0	0.911 4±0.000 0
	uniformRP-Orth	0.383 0±0.000 0	**0.268 5±0.000 0**	**0.945 1±0.000 0**
	SparseRP	0.372 0±0.000 0	0.386 6±0.000 0	0.985 4±0.000 0
	SparseRP-Orth	**0.373 6±0.000 0**	**0.430 4±0.000 0**	**0.991 1± 0.000 0**
KddCup99	GaussRP	0.978 2±0.000 0	0.751 0±0.000 3	0.945 7±0.000 4
	GaussRP-Orth	**0.979 1±0.000 0**	**0.7789±0.0003**	**0.962 0±0.000 4**
	uniformRP	**0.979 9±0.000 0**	**0.792 9±0.000 3**	**0.964 8±0.000 4**
	uniformRP-Orth	0.958 1±0.000 0	0.7200 ±0.000 3	0.929 4±0.000 4
	SparseRP	**0.988 2±0.000 0**	**0.740 5±0.000 3**	**0.933 1±0.000 4**
	SparseRP-Orth	0.982 1±0.000 0	0.682 6±0.000 3	0.842 4±0.000 4
CoverType	GaussRP	**0.561 8±0.000 0**	0.244 0±0.000 3	0.806 9±0.000 0
	GaussRP-Orth	0.550 8±0.000 0	**0.355 2±0.000 3**	**0.900 8±0.000 0**
	uniformRP	**0.553 8±0.000 0**	**0.269 1±0.000 3**	0.831 8±0.000 0
	uniformRP-Orth	0.542 4±0.000 0	0.261 8±0.000 3	**0.832 9±0.000 0**
	SparseRP	0.489 6±0.000 0	**0.562 5±0.000 3**	**0.994 6±0.000 0**
	SparseRP-Orth	**0.536 4±0.000 0**	0.378 2±0.000 3	0.923 1±0.000 0
ACT2	GaussRP	0.532 0± 0.000 2	0.740 3±0.000 0	0.974 7±0.000 0
	GaussRP-Orth	**0.545 9±0.000 2**	**0.741 2±0.000 0**	**0.975 8±0.000 0**
	uniformRP	0.5305±0.000 2	**0.734 4±0.000 0**	0.975 5±0.000 0
	uniformRP-Orth	**0.548 1±0.000 2**	0.727 1±0.000 0	**0.976 9±0.000 0**
	SparseRP	0.535 0±0.000 2	**0.745 2±0.000 0**	0.974 9±0.000 0
	SparseRP-Ort	**0.543 7±0.000 2**	0.733 1±0.000 0	**0.975 9±0.000 0**

Note: The best results are shown in bold.

3.6 处理非平稳数据能力

本节研究 RPFART 在非平稳数据流聚类中的有效性.许多实际应用程序中,数据通常随着时间演变,即具有非平稳性.例如,第 1 个类的数据点全部到达后,第 2 个、第 3 个类的数据点才依次按类别到达.这种情况下,旧的概念消失,同时新的概念随着新的数据点的到来而出现,从而导致概念漂移.因此我们分别将 RPFART 在类排序(按类标签)和类未排序的数据流上进行聚类,重复实验 10 次.图 3～5 显示了 RPFART 的聚类纯度、*NMI* 和 Rand 指数.

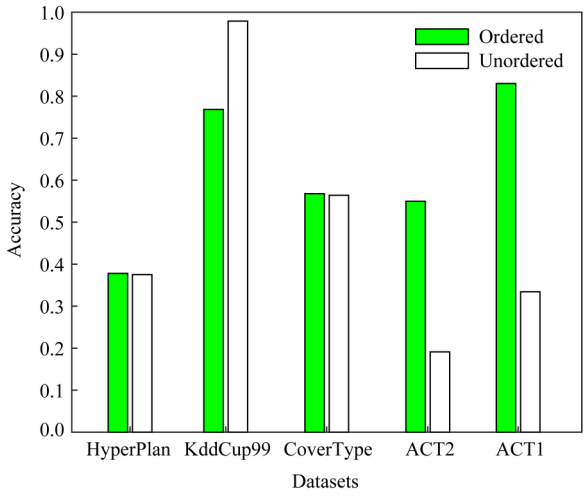


Fig. 3 Accuracy of RPFART with and without ordering of classes

图 3 RPFART 在类排序与类未排序数据集上的聚类纯度

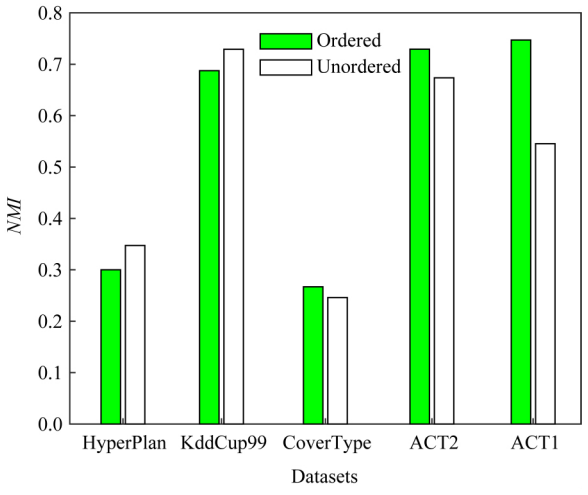


Fig. 4 *NMI* of RPFART with and without ordering of classes

图 4 RPFART 在类排序与类未排序数据集上的 *NMI*

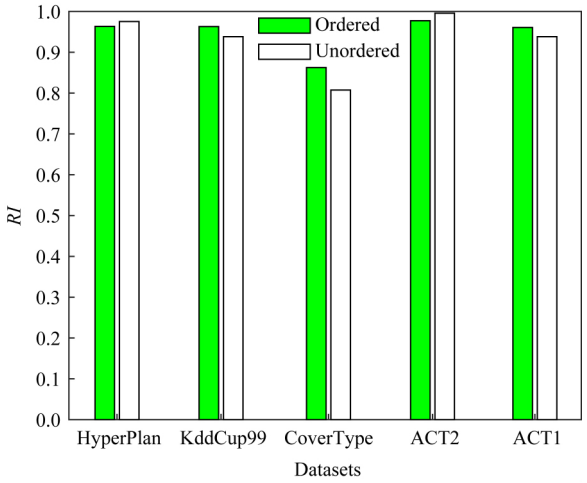


Fig. 5 Rand index of RPFART with and without ordering of classes

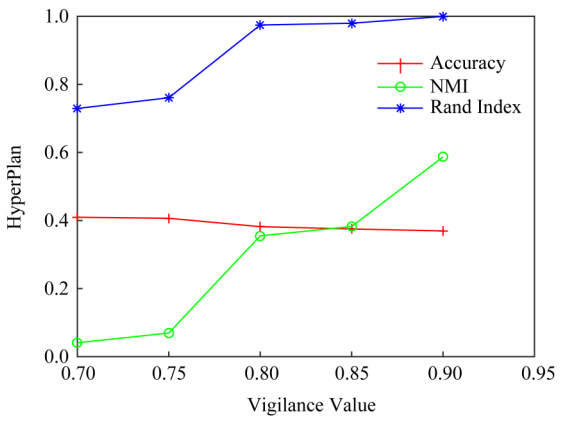
图 5 RPFART 在类排序与类未排序数据集上的 Rand 指数

从图中可以看出:1)RPFART 在类排序数据集上可以找到与类未排序数据集上相当的聚类纯度、*NMI* 和 Rand 指数.特别是 ACT2 和 ACT1 数据集上甚至更优.2)仅 KddCup99 数据集上 RPFART 的聚类纯度值略有下降.基于以上结果,我们可以得出结论,不管数据是否按类标签排序到达,RPFART 均可以有效地处理概念漂移问题.

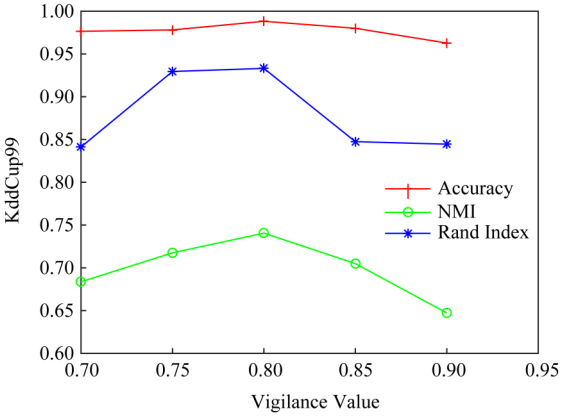
3.7 警戒参数 ρ 的变化

图 6 显示了 $r=50\%$ 时 RPFART 在 5 个数据集上随警戒参数 ρ 的变化聚类性能的变化.

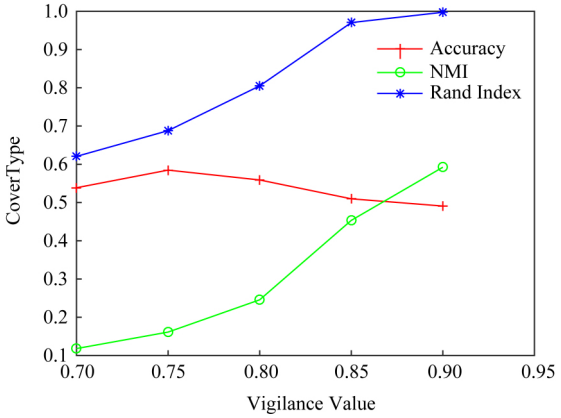
从图 6 可以看出:1)5 个数据集上聚类纯度均随参数 ρ 的增大到达一定值后有所下降;2)在 HyperPlan, CoverType, ACT2 这 3 个数据集上 *NMI* 和 Rand 指数都随参数 ρ 的增大稳步增长,但 KddCup99 和 ACT1 数据集有下降趋势.



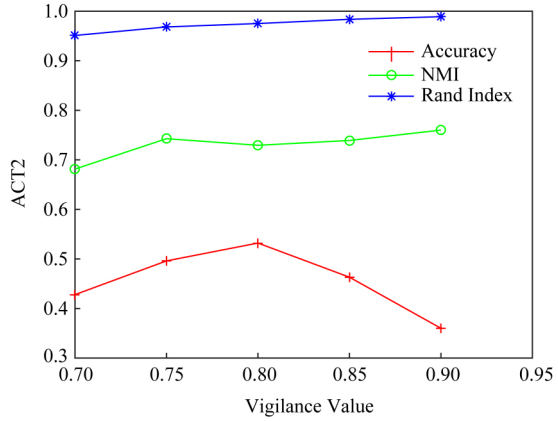
(a) HyperPlan



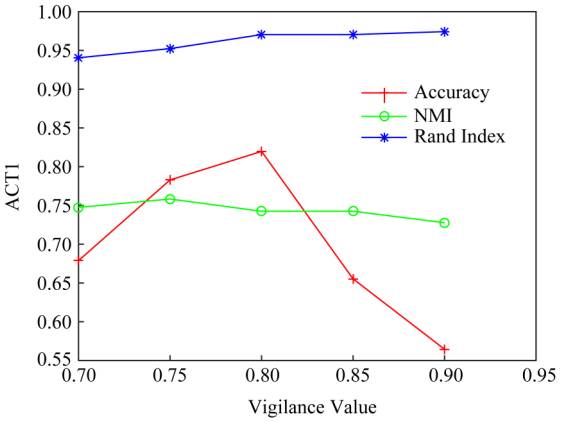
(b) KddCup99



(c) CoverType



(d) ACT2



(e) ACT1

Fig. 6 Sensitivity of RPFART to vigilance value

图 6 警戒参数对 RPFART 算法影响

4 结 论

本文基于随机投影提出了高维数据流聚类算法 RPFART.**首先通过随机投影将原始高维数据映射到低维数据空间,再使用** ART **模型进行数据流聚类.**ART **具有线性计算复杂度,仅使用** 1 **个超参数,并对参数设置鲁棒.文中使用大量实验分析** RPFART **算法的聚类性能.多个数据集上的实验结果表明,即使原始尺寸压缩到** 10%,RPFART **算法仍可以达到与** RPGStream **算法相当甚至更好的性能.对于** ACT1 **数据集,其维数从** 67 500 **减少到** 6 750.

参 考 文 献

[1] Nguyen H L, Woon Y K, Ng W K. A survey on data stream clustering and classification [J]. Knowledge and Information Systems, 2015, 45(3): 535-569

[2] Gaber M M, Zaslavsky A, Krishnaswamy S. Mining data streams: A review [J]. ACM Sigmod Record, 2005, 34(2): 18-26

[3] Gama J, Rodrigues P P. An overview on mining data streams [M] //Foundations of Computational, Intelligence Volume 6. Berlin: Springer, 2009: 29-45

[4] Aggarwal C C. Data streams: An overview and scientific applications [M] //Scientific Data Mining and Knowledge Discovery. Berlin: Springer, 2009: 377-397

[5] Silva J A, Faria E R, Barros R C, et al. Data stream clustering: A survey [J]. ACM Computing Surveys, 2013, 46(1): 1-31

[6] Li Yangyang, Yang Guoli, He Haiyang, et al. A study of large-scale data clustering based on fuzzy clustering [J]. Soft Computing, 2016, 20(8): 3231-3242

[7] Zhang Pu, Shen Qiang. Fuzzy c-means based coincidental link filtering in support of inferring social networks from spatiotemporal data streams [J]. Soft Computing, 2018, 22(21): 7015-7025

[8] Carnein M, Assenmacher D, Trautmann H. An empirical comparison of stream clustering algorithms [C] //Proc of the Computing Frontiers Conf. New York: ACM, 2017: 361-366

[9] Carnein M, Trautmann H. Optimizing data stream representation: An extensive survey on stream clustering algorithms [J]. Business & Information Systems Engineering, 2019, 61(3): 277-297

[10] O'callaghan L, Mishra N, Meyerson A, et al. Streaming-data algorithms for high-quality clustering [C] //Proc of the 18th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2002: 685-694

[11] Aggarwal C C, Han J, Wang J, et al. A framework for clustering evolving data streams [C] //Proc of the 29th Int Conf on Very Large Data Bases. Berlin: VLDB Endowment, 2003: 81-92

[12] Aggarwal C C, Han J, Wang J, et al. A framework for projected clustering of high dimensional data streams [C] // Proc of the 30th Int Conf on Very Large Data Bases. Berlin: VLDB Endowment, 2004: 852-863

[13] Zhou Aoying, Cao Feng, Qian Weining, et al. Tracking clusters in evolving data streams over sliding windows [J]. Knowledge and Information Systems, 2008, 15(2): 181-214

[14] Udommanetanakit K, Rakthanmanon T, Waiyamai K. E-stream: Evolution-based technique for stream clustering [C] //Proc of the Int Conf on Advanced Data Mining and Applications. Berlin: Springer, 2007: 605-615

[15] Lühr S, Lazarescu M. Incremental clustering of dynamic data streams using connectivity based representative points [J]. Data & Knowledge Engineering, 2009, 68(1): 1-27

[16] Cao F, Estert M, Qian W, et al. Density-based clustering over an evolving data stream with noise [C] //Proc of the 2006 SIAM Int Conf on Data Mining. Bethesda: Society for Industrial and Applied Mathematics, 2006: 328-339

[17] Fahy C, Yang S, Gongora M, et al. Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2019, 49(6): 2215-2228

[18] Tasoulis D K, Ross G, Adams N M. Visualising the cluster structure of data streams [C] //Proc of Int Symp on Intelligent Data Analysis .Berlin: Springer, 2007: 81-92

[19] Kriegel H P, Kröger P, Ntoutsi I, et al. Density based subspace clustering over dynamic data [C] //Proc of Int Conf on Scientific and Statistical Database Management. Berlin: Springer, 2011: 387-404

[20] Chen Yixin, Tu Li. Density-based clustering for real-time stream data [C] //Proc of the 13th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2007: 133-142

[21] Wan L, Ng W K, Dang X H, et al. Density-based clustering of data streams at multiple resolutions [J]. ACM Transactions on Knowledge Discovery from Data, 2009, 3 (3):1-28

[22] Park N H, Lee W S. Statistical grid-based clustering over data streams [J]. ACM Sigmod Record, 2004, 33(1): 32-37

[23] Dang X H, Lee V, Ng W K, et al. An EM-based algorithm for clustering data streams in sliding windows [C] //Proc of Int Conf on Database Systems for Advanced Applications. Berlin: Springer, 2009: 230-235

[24] Smith T, Alahakoon D. Growing self-organizing map for online continuous clustering [M] //Foundations of Computational Intelligence Volume 4. Berlin: Springer, 2009: 49-83

[25] Ghesmoune M, Lebbah M, Azzag H. A new growing neural gas for clustering data streams [J]. Neural Networks, 2016, 78(1): 36-50

[26] Zhu Yingwen, Chen Songcan. Growing neural gas with random projection method for high-dimensional data stream clustering [J]. Soft Computing, 2020, 24(13): 9789-9807

[27] Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications [C] //Proc of the 1998 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 1998: 94-105

[28] Dy J G, Brodley C E. Feature subset selection and order identification for unsupervised learning [C] //Proc of Int Conf on Machine Learning. New York: ACM, 2000: 247-254

[29] Webb A R. Statistical pattern recognition [M]. New York: John Wiley & Sons, 2003

[30] Keogh E, Chakrabarti K, Pazzani M, et al. Locally adaptive dimensionality reduction for indexing large time series databases [C] //Proc of the 2001 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2001: 151-162

[31] Johnson W B, Lindenstrauss J. Extensions of Lipschitz mappings into a Hilbert space [J]. Contemporary Mathematics, 1984, 26(1): 189-206

[32] Achlioptas D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins [J]. Journal of Computer and System Sciences, 2003, 66(4): 671-687

[33] Boutsidis C, Zouzias A, Drineas P. Random projections for k-means clustering [C] //Advances in Neural Information Processing Systems. New York: Curran Associates, 2010: 298-306

[34] Wu X, Kumar V, Quinlan J R, et al. Top 10 algorithms in data mining [J]. Knowledge and Information Systems, 2008, 14(1): 1-37

[35] Cohen M B, Elder S, Musco C, et al. Dimensionality reduction for k-means clustering and low rank approximation [C] //Proc of the 47th Annual ACM Symp on Theory of Computing. New York: ACM, 2015: 163-172

[36] Musco C N. Dimensionality reduction for k-means clustering [D]. Boston: Massachusetts Institute of Technology, 2015

[37] Schneider J, Vlachos M. On randomly projected hierarchical clustering with guarantees [C] //Proc of the 2014 SIAM Int Conf on Data Mining. Society for Industrial and Applied Mathematics. Philadelphia: SIAM, 2014: 407-415

[38] Schneider J, Vlachos M. Fast parameterless density-based clusteringvia random projections [C] //Proc of the 22nd ACM Int Conf on Information & Knowledge Management. New York: ACM, 2013: 861-866

[39] Fern X Z, Brodley C E. Random projection for high dimensional data clustering: A cluster ensemble approach [C] //Proc of the 20th Int Conf on Machine Learning (ICML-03). New York: ACM, 2003: 186-193

[40] Cardoso Â, Wichert A. Iterative random projections for high-dimensional data clustering [J]. Pattern Recognition Letters, 2012, 33(13): 1749-1755

[41] Ye Mao, Liu Wenfen, Wei Jianghong, et al. Fuzzy-means and cluster ensemble with random projection for big data clustering [J]. Mathematical Problems in Engineering, 2016, 2016(1): 1-13

[42] Carpenter G A, Grossberg S, Rosen D B. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system [J]. Neural Networks, 1991, 4(6): 759-771

[43] Grossberg S. How does a brain build a cognitive code [J]. Psychological Review, 1980: 87(1): 1-51

[44] Carpenter G A, Grossberg S. A massively parallel architecture for a self-organizing neural pattern recognition machine [J]. Computer Vision, Graphics, and Image Processing, 1987, 37(1): 54-115

[45] Carpenter G A, Grossberg S. ART 2: Self-organization of stable category recognition codes for analog input patterns [J]. Applied Optics, 1987, 26(23): 4919-4930

[46] Carpenter G A, Grossberg S, Rosen D B. ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition [J]. Neural Networks, 1991, 4(4): 493-504

[47] Carpenter G A, Grossberg S. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures [J]. Neural Networks, 1990, 3(2): 129-152

[48] Robert Hecht-Nielsen. Context vectors: General purpose approximate meaning representations self-organized from raw data [J]. Computational Intelligence: Imitating Life, 1994: 43-56

Zhu Yingwen, born in 1982. PhD candidate with the College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics (NUAA). Student member of CCF. Received her BSc and MSc degrees in computer science & technology from Nanjing Normal University in 2005 and 2008. Her main research interests include data mining and machine learning.

Chen Songcan, born in 1962. Professor and PhD supervisor. Senior member of CCF. His main research interests include pattern recognition, machine learning and neural computing.