基于交互基函数的数据流聚类算法研究

黄承宁[1],李 莉[1],姜丽莉[1],徐平平[2]
(1. 南京工业大学浦江学院,江苏 南京 211222;
2. 东南大学 信息科学与工程学院,江苏 南京 210096)

摘 要:聚类是数据挖掘的有效工具,数据流聚类成为当前研究热点,目前很多数据流聚类算法已经被提出,但大部分算法将距离作为相似度度量标准,存在对噪点敏感问题,且聚类效果不理想。为了增强数据流聚类算法的灵活性并提升聚类质量,该文将分数阶交互基函数(IBFs)引入数据流聚类,结合模糊 ART 算法对其进行了扩展,生成柔性决策边策略,提出了新颖的数据流聚类算法 IBFs_ART。该算法首先对到达的数据点根据特征之间的相关性通过预计算函数特征扩展,并对原有特征进行分数阶变换,之后再基于交互基函数进行数据流聚类。交互基函数可生成灵活的决策边界且不需要指定软件,预计算函数可以在任何算法中实现,其可用于数据流聚类算法的任何扩展。经过实验表明,使用 IBFs 实现了较低计算成本生成灵活决策边界来找到最优聚簇,在相同警戒参数下实现了更高聚类质量和纯度,较传统聚类算法拥有更高的聚类精度、对称度量和更小的错误率。

关键词:聚类;数据流;数据流聚类;交互基函数;模糊自适应谐振理论
中图分类号:TP311.13 **文献标识码**:A **文章编号**:1673-629X(2024)03-0028-07
doi:10.3969/j.issn.1673-629X.2024.03.005

Research on Data Stream Clustering Algorithm Based on Interactive Basis Function

HUANG Cheng-ning[1], LI Li[1], JIANG Li-li[1], XU Ping-ping[2]
(1. Pujiang College of Nanjing University of Technology, Nanjing 211222, China;
2. School of Information Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: Clustering is an effective tool for data mining, and data stream clustering has become a hot topic in current research. Currently, many data stream clustering algorithms have been proposed, but most of them use distance as a similarity metric, which is sensitive to noise, and not ideal in clustering effect. In order to enhance the flexibility and improve the clustering quality of data flow clustering algorithms, we introduce fractional order interactive basis functions (IBFs) into data flow clustering, and combine them with fuzzy ART algorithm for expansion to generate flexible decision edge strategies. A novel data flow clustering algorithm, IBFs_ART, is proposed. The algorithm first expands the arrived data points through a pre calculated function based on the correlation between features, and performs fractional transformation on the original features. Then, it clusters the data streams based on interactive basis functions. Interactive basis functions can generate flexible decision boundaries without specifying software. Precomputing functions can be implemented in any algorithm, and can be used for any extension of data stream clustering algorithms. Experiments have shown that using IBFs can achieve lower computational costs and generate flexible decision boundaries to find the optimal clustering, achieve higher clustering quality and purity under the same alert parameters, and have higher clustering accuracy, symmetry metrics, and smaller error rates compared with traditional clustering algorithms.

Key words: cluster; data stream; data stream clustering; interactive basis function; fuzzy adaptive resonance theory

0 引 言

数据流是一种潜在的海量、连续、快速的数据信息序列,引起了数据挖掘领域的极大关注和研究热潮[1]。在人类生活的各个方面都存在数据流,如网络媒体传输的监测信息、煤矿传感器传输的信息、网站信息、金融和证券公司产生的经济信息、天气预报信息等。由

收稿日期:2023-03-17 **修回日期**:2023-07-19
基金项目:国家自然科学基金项目(61702229);南京工业大学浦江学院科研重点培育课题(njpj2022-1-07);南京工业大学浦江学院青年教师发展基金(PJYQ03)
作者简介:黄承宁(1985-),男,副教授,博士,研究方向为知识发现、人工智能、计算机教育应用。

于这种形式的数据海量且实时更新,传统的聚类方法无法对其进行处理,因此迫切需要新的聚类方法[2]。目前,已经有很多数据流聚类方法被提出,不过均根据传统数据的聚类算法扩展而来,且均没有考虑到特征之间的关系。

该文提出将交互基函数(IBFs)引入数据流聚类,结合模糊 ART 算法,考虑特征的自交互与交叉交互,以相对较低的计算成本生成灵活决策边界来找到最优聚簇,实现了聚类高精度与低错误率,提高了算法的数据流聚类质量。

1　数据流聚类与自适应谐振理论

1.1　数据流聚类

数据流具有内在的特性,包括无限大小、时间顺序和动态变化。与传统的数据挖掘相比,数据流挖掘只是在满足单次通过、实时响应、有界内存和概念漂移检测等约束条件下产生近似结果。

数据流(DS)定义为数据对象或样本序列或为一个带有时间戳(Time Stamp)的多维数据点集合: DS = $\{x_1, x_2, \cdots, x_n\}$,其中 x_n 为第 n 个到达的数据对象(实际应用中 n 的取值可以为无限大[3-4])。其中每个数据点是一个 d 维的数据记录,其到达时间为 t_i 。

数据流聚类将 DS 中的相似对象划分为一个或多个组(称为"簇",Cluster),划分后,同一簇中的元素彼此相似,但相异于其他簇中的元素。

针对高维、动态、实时的特点,目前不少研究者都已经提出了许多有效的数据流聚类算法,但数据流信息是不确定的,总是存在离群点且包含噪声[5],传统的聚类方法无法对其进行处理,因此发现新的数据流聚类方法越来越迫切。

目前从实际应用看,数据流聚类基本都面临着许多共性问题[6-7]:(1)内存有限:数据流中的数量往往是庞大的,不可能在内存和硬盘中存储整个数据流;(2)一次扫描:因为巨大的数据量,传统的扫描方法不再适用,在对数据的访问只能单次线性,也就是只按顺序依次读取一次,不能进行随机访问;(3)实时响应:大多数应用程序要求快速响应,因此挖掘应该是一个连续的在线过程;(4)概念漂移:数据分布经常随时间变化。目前典型的数据流聚类算法包括 REPSTREAM, ACSC, G-Stream, MR-Stream, CellTree 以及 RPGStream 等[8]。

1.2　自适应谐振理论

自适应谐振理论(ART)[9]是一种学习模型,它模拟人脑捕获,识别和记忆有关对象和事件的信息,既是一种认知理论,也是一种关于大脑如何在不断变化的世界中快速学会分类、识别和预测物体和事件的神经理论。该文提出的算法便是在模糊自适应谐振理论基础上引入交互基函数(IBFs)[10]扩展进行数据流聚类,从而提升聚类精度与质量。

模糊自适应谐振理论的体系结构由用于接收输入模式的输入层 F1 和用于聚类的类别层 F2 组成[11],如图 1 所示,输入层 F1 包含的输入向量被提交到网络,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。

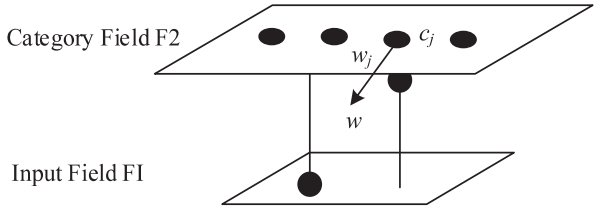

图 1　模糊 ART 结构

模糊 ART 使用模糊算法并引入一个"补编码"[12]来解决"类别扩散"问题。模糊 ART 执行步骤如下:

(1)类别选择:对于每个输入模式 I ,模糊 ART 根据选择函数为识别层 F2 中的每个聚簇计算一个选择值,并标识具有最大值的聚簇为获胜聚簇,第 j 个簇的选择函数定义为:

$$T_j = \frac{|I \wedge W_j|}{\partial + |W_j|} T_j \tag{1}$$

(2)模板匹配:使用匹配函数 M_{j^*} 评估输入模式 I 与获胜聚簇 C_{j^*} 之间的相似性,该函数定义为:

$$M_j = \frac{|I \wedge W_{j^*}|}{|I|} M_j \tag{2}$$

如果获胜聚簇 C_{j^*} 满足警戒标准 $M_{j^*} \geqslant \rho$,会发生谐振,从而导致步骤 3 的中心学习。否则,将在其余聚类中选择新的获胜聚簇。如果没有获胜聚簇满足警戒标准,则将生成一个新的聚簇来对输入模式进行编码。

(3)中心学习:如果 C_{j^*} 满足警戒标准,其对应的权重向量 W_{j^*} 将通过函数进行更新,定义为:

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

模糊 ART 中基于警戒准则计算的簇的 VR 是由特征空间中与簇关联的区域几何定义的,它从几何上解释了模糊 ART 的警戒准则,认为落入 VR 的输入模式与相应的簇相似,而 VR 的形状和功能行为则取决于补编码的使用[13]。

2　基于交互基函数的数据流聚类

2.1　交互基函数

如前所述,用于训练的特征构成了问题的基础向量。例如,当特征数量 $p = 2$ 时,搜索空间是由特征的正交轴形成的平面,每个特征都是一个基向量。三个特征形成三维基础,以此类推。如果把一个特征看作一个基向量,基函数就是一个变换。在最简单的情况

下,基函数可以是等式:

$$f(X) = X \tag{4}$$

多项式函数的一个特殊情况,即当 $a = 1$:

$$f(X) = X^a \tag{5}$$

$$f(X) = (1 - X)^a \tag{6}$$

也可以定义其它基函数,例如指数:

$$f(X) = (e^X)^a \tag{7}$$

回归分析中常用的是基函数,它们具有改变回归平面性质的作用。例如,从恒等式到变量的平方的转换具有将回归线变为抛物线的效果。但 DTs(决策树)[14]中基函数的使用并没有同样的效果。考虑 K 个实函数 b_i 的一般情况:$R \to R$,$i = 1,2,\cdots,K$,称 $\{f_1, f_2,\cdots,f_K\}$ 为一组基函数。然后利用基函数得到的 T 个新特征扩充 p 个特征集:

$$X^* = (X_1,\cdots,X_p,X_{p+1},\cdots,X_{p+T}) \tag{8}$$

并且 $X^* \in R^{P+T}$,$X_{p+i} = f_{s_i}(X_{j_i})$,$i = 1,2,\cdots,T$,$s_i \in \{1,2,\cdots,K\}$,$j_i \in \{1,2,\cdots,p\}$。

值得注意的是,增广集合 T 中应用于任意 X_p 的决策树的标准递归划分机制导致了($p + T$)维投影的划分。此外,这些投影在 X 定义的 R^p 子空间中的投影是原空间上线性正交划分的结果。在 $p = 2$ 的情况下,即 $X = \{X_1, X_2\}$。另外,当 $T = 1$, $K = 1$, $b_1(x) = x^2$ 使得 $X^* = \{X_1, X_2, X_3 = X_1^2\}$。当划分机制选择在 X_3 中拆分 s,该分裂在 X 中投影为一个常数 $X_1 = \sqrt{X_3}$。在基函数维数上的任何划分都等价于在原基上找到一个正交决策边界[15]。

由于基函数在原基中仍然产生正交划分,笔者的建议是在构造 X^* 时使用两个或多个特征之间的交互信息。这些交互不同于自交互,可以通过一组 D 函数来识别,这些 M 函数通过基函数来再现特征变换的功能交互,这些交互函数被定义为:

$$hh_i: R^{pk} \to Rh \tag{9}$$

$$(b_1(X_1), b_1(X_2),\cdots,b_k(X_p)) \tag{10}$$

此设置下,定义:

$$X^* = (X_1,\cdots,X_p,X_{p+1},\cdots,X_{p+D}) \tag{11}$$

$$X_{p+i} = h_i(b_1(X_1), b_1(X_2),\cdots,b_k(X_p))$$
$$i = 1,2,\cdots,D \tag{12}$$

通过将标准递归划分方法应用 X^* 上,并考虑到特征之间的相互作用,在 X 上的投影将提供一个斜划分(最终也可能是非线性的)。

IBFs 提供的框架不仅允许诱导出斜划分,还允许诱导出非线性决策边界[16-17]。这是通过在数据集中特征生成的子空间 $X = (X_1, X_2,\cdots,X_p)$ 中投影方程 $h_i(b_1(X_1), b_1(X_2),\cdots,b_k(X_p)) = a$ 来完成的。

2.2 基于交互基函数的数据流聚类算法

基于交互基函数的特性,在实验中将 IBFs 引入模糊 ART,提出 IBFs_ART 算法,用于对数据流进行聚类。通过对原始输入特征进行分数阶变换,诱导出单一的超参数,在实现上比模糊 ART 更具灵活性,且进一步提升聚类精度。

IBFs_ART 算法通过分数阶交互基函数(IBFs)对模糊 ART 进行了扩展,提出了一种新的生成柔性决策边界的策略。目标是评估 IBFs 在 IBFs_ART 中的表现。当样本 $x = \{x^1, x^2,\cdots,x^d\}$ 即将到来时,每个特征在[0,1]中被归一化。对于 IBFs,用 d 个新特征来扩大 d 个特征的集合:

$$x^* = (x^1, x^2,\cdots,x^d,x^{d+1},x^{d+2},\cdots,x^{2d}) \tag{13}$$

其中,使用自交互时 $x^* \in R^{2d}$,$x^{d+j} = f_p(x^j)$,$p \in \{1, 2,\cdots,K\}$。使用交叉交互时 $x^{d+j} = g_1(f_1(x^1), f_2(x^2),\cdots,f_K(x^d))$。

考虑如下函数:

$$f_1(x^j) = (x^j)^a \tag{14}$$

$$f_2(x^j) = (1 - x^j)^a \tag{15}$$

$$f_3(x^j) = (e^{xj})^a \tag{16}$$

$$x^{d+j} = g_1(f_1(x^1), f_2(x^2),\cdots,f_k(x^d)) = \max_{m\in[1,d]m\neq j} \min_{1}(f_1(x^j), f_1(x^m)) \tag{17}$$

当新的数据点 x_n 到达时,提出的 IBFs_ART 首先使用 IBFs 对 d 个新特征的集合进行放大。因此,将 $DS = \{x_1, x_2,\cdots,x_n\}$ 逐一变换为 $X^* = \{x_1^*, x_2^*,\cdots, x_n^*\}$,$x_n^* = [0,1](n = 1,2,\cdots)$。通过补编码,$x_n^*$ 与它的补向量 $\overline{x_n^*}$ 进一步级联,使得 $I = (x_n^*, \overline{x_n^*})$,其中 $\overline{x_n^*} = 1 - x_n^*$。然后,利用模糊 ART 对这一变换后的数据流进行聚类。算法 IBFs_ART 给出了详细的实现过程。

IBFs_ART 算法如下所示:

输入:$DS = \{x_1, x_2,\cdots,x_n\}$

输出:节点集合 $C = \{c_1, c_2,\cdots,c_n\}$ 和权值 $W = \{W_{c_1}, W_{c_2},\cdots, W_{c_n}\}$

(1):for each x_n

(2):用公式 11 ~ 14 计算 x_n^*

(3):补编码 x_n^* 得到 $I = (x_n^*, \overline{x_n^*})$

(4):使用公式 1 计算选择函数,求出活动节点 $\Lambda(\Lambda \in C)$

(5):从活动节点中查找获胜聚簇 J:$J = \arg_{j\in\Lambda}\max(T_j)$

(6):使用公式 2 计算匹配函数;

(7):if 获胜聚簇 J 满足 $M_j \geqslant \rho$

(8):使用学习函数(3)更新获胜聚簇 J

(9):else

(10):类别 J:$\Lambda \leftarrow \Lambda - J$

(11):if 活动节点 $\Lambda \neq \varnothing$ then

(12):返回执行第 5 步

(13):else

(14):$J = |C| + 1$

(15):创建新的聚类: $C \leftarrow C \cup J$

(16):初始化新的聚类: $w_j = I$

(17):end if

(18):end if

(19):end if

3　实验与结果

3.1　实验环境

本次实验计算机配置为 Inter Core i7-7500U 2.90 GHz 处理器和 4 GB 内存,Windows10 操作系统,所有比较程序均在 MATLAB 上设计和运行。

3.1.1　数据集

为了对聚类的有效性进行更好的评价,在实验中采用了人工数据集和真实数据集,见表 1。

表 1　数据集

Datasets	#records(样本)	#features(维度)	#classes(类)
Letter4	9 344	2	7
KddCup99	494 021	41	23
CoverType	581 012	54	7
Powersupply	29 928	2	24

Letter4 由 Java 代码 https://github.com/feldob/ 生成。它包括 9 344 个样本,2 个维度和 7 个类。

KddCup99 来源于林肯实验室的一项入侵检测评估项目,仿真各种不同的用户类型、网络流量和攻击手段,记录了 9 周内 TCP 网络连接和系统审计数据。包含约 50 万条连接记录,这些记录含 1 种正常的标识类型和 22 种训练攻击类型,共有 23 个类,每个连接记录包含 41 个维度。

CoverType 来源于某国家森林的四片荒野区域的观测。共包含 581 012 条记录,分为 7 种类型,每条观测记录包含 54 个维度。

Powersupply 来源于意大利某电力公司的供电数据,记录两个电能:来自主电网的电能和来自其他电网的电能。该流包含 2015 年至 2018 年三年供电记录。数据变化主要来自季节、天气、一天的时间(例如早晚),以及工作日和周末的差异。它由 29 928 个样本,2 个维度,24 个类组成。

3.1.2　聚类评价指标

为了评价算法性能,引入了三种评价指标:

(1)Accuracy(purity)。

$$\mathrm{Acc} = \frac{\sum_{i=1}^{K} \left| \frac{c_i^d}{c_i} \right|}{k} \tag{18}$$

K:聚簇个数, $|C_i^d|$:聚簇 i 中的样本点数, $|C_i|$:聚簇 i 中的真实样本个数。直观地来看,聚类精度度量了聚簇的纯度。Acc 的取值在[0,1]之间,取值越大代表聚类精度越高。

(2)NMI(normalized mutual information)。

NMI 是一个量化两个分布之间共享的统计信息的对称度量,当类簇标签和样本类别之间存在一对一的映射时,NMI 值达到最大为 1.0。A 为真实聚簇 $A = \{A_1, A_2, \cdots, A_k\}$, B 为通过某个聚类算法得到的聚簇 $B = \{B_1, B_2, \cdots, B_h\}$, $\boldsymbol{C}$ 为混淆矩阵, $\boldsymbol{C}$ 中的元素 C_{ij} 表示既在 A 中又在 B 中的样本个数。

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_iC_j)}{\sum_{i=1}^{C_A} C_i\log(C_i/N) + \sum_{j=1}^{C_B} C_j\log(C_j/N)} \tag{19}$$

其中, $C_A(C_B)$ 为聚簇 A,B 同时在矩阵 $\boldsymbol{C}$ 中的簇数目, $C_{i.}(C_{.j})$ 为 $\boldsymbol{C}$ 中第 i 行的元素和; N 为样本个数。

(3)RI(rand index)。

RI(兰德指数)的计算公式为:

$$\mathrm{RI} = \frac{n_{11} + n_{00}}{C_n^2} r_i \tag{20}$$

其中, n 为样本个数, n_{11} 是两聚类中划分为同一类的数据对数, n_{00} 则是划分为不同类的数据对数,并且 $C_n^2 = n(n-1)/2$。RI 的取值范围为[0,1],取值越大,聚类性能越好。当两种聚类算法划分一致时,RI=1。

3.2　实验结果

首先评价 IBFs_ART 的聚类质量,并从 Acc,NMI 和 RI 三个方面与 G-Stream(警戒参数较多)以及模糊 ART(Fuzzy ART,只有一个警戒参数)进行了比较。对于自交互,使用公式 5~7,对于特征交互,使用以下三个函数:

$$X^{d+j} = g_1(f_1(x^1), f_1(x^2), \cdots, f_1(x^d)) = \max_{m\in[1,d]m\neq j} \min (f_1(x^j), f_1(x^m)) \tag{21}$$

$$X^{d+j} = g_1(f_2(x^1), f_2(x^2), \cdots, f_2(x^d)) = \max_{m\in[1,d]m\neq j} \min (f_2(x^j), f_2(x^m)) \tag{22}$$

$$X^{d+j} = g_1(f_3(x^1), f_3(x^2), \cdots, f_3(x^d)) = \max_{m\in[1,d]m\neq j} \min (f_3(x^j), f_3(x^m)) \tag{23}$$

每个算法重复实验 10 次,聚类结果如表 2~4 所示。通过实验,发现取不同的 ρ 值,IBFs_ART 算法从三个方面的评价几乎都可以找到一个 a 值(选取 1/2 和 1/4 值),使其性能指标均优于模糊 ART,且性能指标得到不小提升,验证了 IBFs_ART 算法的优越性。

通过实验评估了不同警戒参数 ρ 的 IBFs_ART 的性能,该参数控制了当输入样本与类别发生共振时,随后是否允许该类别学习样本。实验中选择合理的警戒值 ρ 可以允许发现有用的簇,而不需要对许多敏感参数值进行微调。图 2~5 显示了 IBFs_ART 在 4 个数

据集上使用 Acc,NMI 和 RI 三个评价指标展示警戒参　　数 ρ 的敏感性。

表 2　IBFs_ART 和其他数据流聚类算法 Acc 比较结果

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply
G-Stream	-	0.951 5	0.981 4	0.520 1	0.175 2
Fuzzy ART	-	0.96 (ρ =0.85)	0.982 7 (ρ =0.8)	0.593 5 (ρ =0.8)	0.174 8 (ρ =0.95)
IBFs_ART(使用公式 5)	a =1/2	0.989 1	0.900 9	0.601 4	0.174 8
	a =1/4	0.972	0.901 0	0.568 9	0.172 6
IBFs_ART(使用公式 6)	a =1/2	0.983	0.985 5	0.590 8	0.177 4
	a =1/4	0.983 2	0.985 8	0.587 6	0.174 8
IBFs_ART(使用公式 7)	a =1/2	0.992 1	0.980 2	0.596 3	0.173 5
	a =1/4	0.992 4	0.984 9	0.596 2	0.176 1
IBFs_ART(使用公式 21)	a =1/2	0.912 9	0.983 5	0.601 4	0.175 1
	a =1/4	0.961	0.983 3	0.568 9	0.174 7
IBFs_ART(使用公式 22)	a =1/2	0.960 5	0.985 5	0.590 8	0.174 7
	a =1/4	0.999 8	0.985 8	0.587 6	0.171 1
IBFs_ART(使用公式 23)	a =1/2	0.980 6	0.980 5	0.593 6	0.175 1
	a =1/4	0.981 1	0.983 8	0.596 2	0.176 5

表 3　IBFs_ART 和其他数据流聚类算法 NMI 比较结果

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply
G-Stream	-	0.599 1	0.173 2	0.186	0.173 2
Fuzzy ART	-	0.689 5 (ρ =0.9)	0.760 1 (ρ =0.9)	0.183 3 (ρ =0.8)	0.186 (ρ =0.95)
IBFs_ART(使用公式 5)	a =1/2	0.693 1	0.684 0	0.170 9	0.185 1
	a =1/4	0.712 7	0.680 9	0.154 8	0.185 4
IBFs_ART(使用公式 6)	a =1/2	0.678 6	0.772 8	0.155 8	0.186 5
	a =1/4	0.693 3	0.678 5	0.149 4	0.186 3
IBFs_ART(使用公式 7)	a =1/2	0.679 4	0.751 4	0.159 3	0.185 6
	a =1/4	0.676 6	0.693 9	0.165 0	0.186 0
IBFs_ART(使用公式 21)	a =1/2	0.696 2	0.696 7	0.170 9	0.182 6
	a =1/4	0.709 5	0.706 5	0.154 8	0.186 7
IBFs_ART(使用公式 22)	a =1/2	0.677 9	0.772 8	0.155 8	0.182 4
	a =1/4	0.672 8	0.678 5	0.149 4	0.187 2
IBFs_ART(使用公式 23)	a =1/2	0.688 9	0.744 3	0.159 3	0.183 7
	a =1/4	0.692 6	0.789 5	0.165 0	0.182 4

表 4　IBFs_ART 和其他数据流聚类算法 RI 比较结果

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply
G-Stream	-	0.809 5	0.817	0.615 8	0.946 4
Fuzzy ART	-	0.834 3 (ρ =0.9)	0.952 6 (ρ =0.9)	0.649 2 (ρ =0.8)	0.941 2 (ρ =0.95)
IBFs_ART(使用公式 5)	a =1/2	0.848 7	0.918 6	0.648 2	0.938 7
	a =1/4	0.860 2	0.911 1	0.638 3	0.932 4
IBFs_ART(使用公式 6)	a =1/2	0.827 2	0.954 7	0.644 9	0.933 0
	a =1/4	0.833 3	0.909 2	0.638 6	0.930 3

续表 4

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply
IBFs_ART(使用公式 7)	$a=1/2$	0.833 5	0.952 4	0.649 3	0.942 2
	$a=1/4$	0.835 8	0.912 6	0.644 1	0.937 6
IBFs_ART(使用公式 21)	$a=1/2$	0.849 3	0.919 8	0.648 2	0.938 9
	$a=1/4$	0.862 5	0.926 2	0.638 3	0.932 8
IBFs_ART(使用公式 22)	$a=1/2$	0.832 3	0.954 7	0.644 9	0.935 0
	$a=1/4$	0.827 3	0.909 2	0.638 6	0.926 9
IBFs_ART(使用公式 23)	$a=1/2$	0.837 1	0.948 5	0.649 3	0.938 8
	$a=1/4$	0.837 9	0.957 8	0.644 1	0.939 2

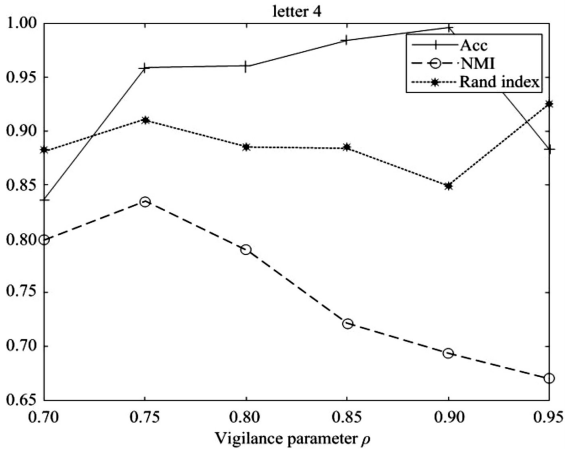

图 2　IBFs_ART 对于 Letter4 数据集 ρ 的敏感性

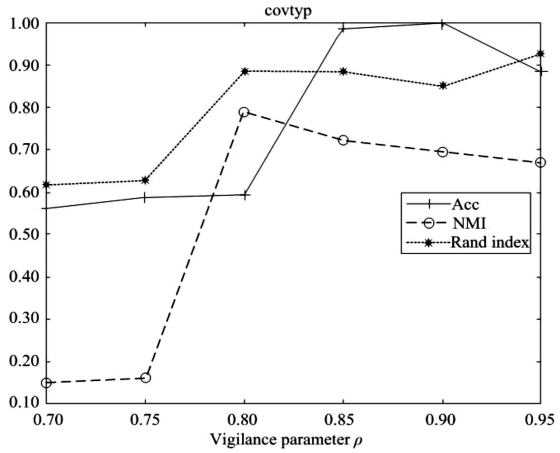

图 3　IBFs_ART 对于 Kddcup99 数据集 ρ 的敏感性

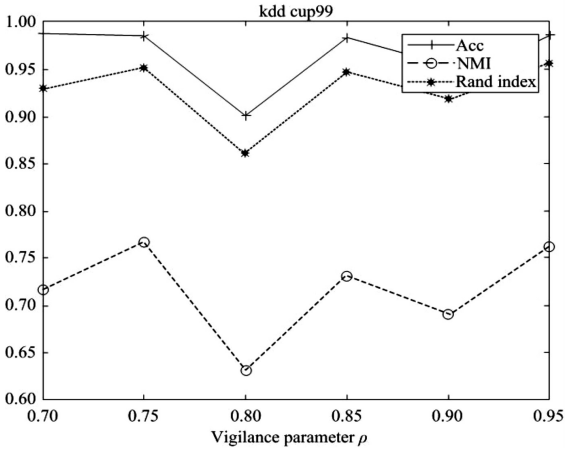

图 4　IBFs_ART 对于 CoverType 数据集 ρ 的敏感性

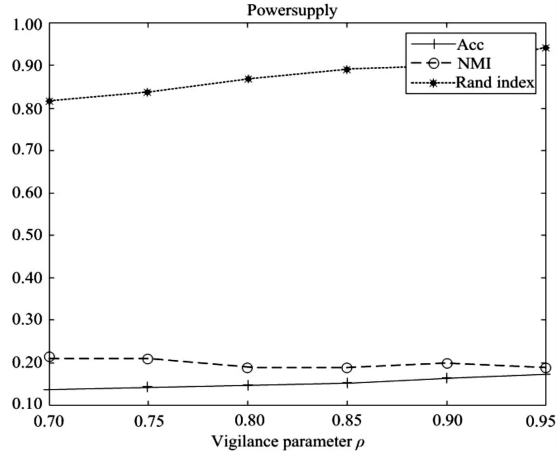

图 5　IBFs_ART 对于 Powersupply 数据集 ρ 的敏感性

通过实验,首先评价了 IBFs_ART 的聚类质量,并从 Acc,NMI 和 RI 三个方面将 G-Stream 以及模糊 ART 方法进行了比较,并且 IBFs_ART 同时采用了自交互与交叉交互。其次,采用不同的警戒参数值进行实验,证明了警戒参数对算法的影响。大量的数据结果证明,IBFs_ART 可以达到更好的聚类效果与更高性能。

4　结束语

数据流是一种潜在的海量、连续、快速的数据信息序列,引起了数据挖掘领域的极大关注和研究热潮。而聚类又是数据挖掘的有效工具,因此数据流聚类无疑是数据流挖掘研究的重点。该文将交互基函数引入到模糊 ART 中,构造 IBFs_ART 算法,经过和原先算法的对比实验,验证了该算法能够提高聚类精度且只需要较低的计算成本,在 Acc,NMI 和 RI 三个方面都比传统算法模型更好,且底层模糊 ART 递增执行聚类的过程并没有改变,也就意味着 IBFs_ART 算法可以在任何算法中实现,可用于数据流聚类算法的任何扩展。

参考文献：

[1] IKOTUN A M, EZUGWU A E, ABUALIGAH L, et al. K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data[J]. Information Sciences, 2022, 622: 178-210.

[2] BATAINEH B. Fast component density clustering in spatial databases: a novel algorithm[J]. Information, 2022, 13(10): 477.

[3] 朱颖雯,陈松灿. 数据流聚类算法研究[J]. 数据采集与处理,2022,37(4):894-908.

[4] 陈 谦,徐兴梅,陈 帅. 基于文本挖掘的多用户投诉数据流聚类算法[J]. 计算机仿真,2022,39(5):423-426.

[5] 褚轲欣,荀亚玲. 基于相似度均值的分类数据层次聚类分析算法[J]. 计算机技术与发展,2022,32(11):154-163.

[6] NAKARIYAKUL S. High-dimensional hybrid feature selection using interaction information-guided search[J]. Knowledge-Based Systems, 2018, 145: 59-66.

[7] FAHY C, YANG S, GONGORA M. Ant colony stream clustering: a fast density clustering algorithm for dynamic data streams[J]. IEEE Transactions on Cybernetics, 2019, 49(6): 2215-2228.

[8] GHESMOUNE M, LEBBAH M, AZZAG H. A new growing neural gas for clustering data streams[J]. Neural Networks, 2016, 78: 36-50.

[9] ZHU Y, CHEN S. Growing neural gas with random projection method for high-dimensional data stream clustering[J]. Soft Computing, 2020, 24(13): 9789-9807.

[10] BRITO DA SILVA L E, ELNABARAWY I, WUNSCH D C. Dual vigilance fuzzy adaptive resonance theory[J]. Neural Networks, 2019, 109: 1-5.

[11] DA SILVA L E B, ELNABARAWY I, WUNSCH II D C. Distributed dual vigilance fuzzy adaptive resonance theory learns online, retrieves arbitrarily-shaped clusters, and mitigates order dependence[J]. Neural Networks, 2020, 121: 208-228.

[12] BAGOZI A, BIANCHINI D, DE ANTONELLIS V. Multi-level and relevance-based parallel clustering of massive data streams in smart manufacturing[J]. Information Sciences, 2021, 577: 805-823.

[13] MATIAS A L S, ROCHA NETO A R, MATTOS C L C, et al. A novel fuzzy ARTMAP with area of influence[J]. Neurocomputing, 2021, 432: 80-90.

[14] ARUN MANICKA RAJA M, SWAMYNATHAN S. Hierarchical stream clustering based NEWS summarization system[J]. Computers, Materials & Continua, 2022, 70(1): 1263-1280.

[15] KAUR A, KUMAR Y. A multi-objective vibrating particle system algorithm for data clustering[J]. Pattern Analysis and Applications, 2022, 25(1): 209-239.

[16] SILVA L, ELNABARAWY I, II D W. A survey of adaptive resonance theory neural network models for engineering applications[J]. Neural Networks, 2019, 120: 167-203.

[17] MENG L, TAN A H, WUNSCH D C. Adaptive scaling of cluster boundaries for large-scale social media data clustering[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(12): 2656-2669.