邓家奇, 尹宏伟, 胡文军. 基于演化核嵌入的多视角数据流聚类算法[J]. 智能计算机与应用,2026,16(5):10-19. DOI:10.20169/j.issn.2095-2163.24090301

基于演化核嵌入的多视角数据流聚类算法

邓家奇[1,2], 尹宏伟[1,2], 胡文军[1,2]
(1 湖州师范学院 信息工程学院, 浙江 湖州 313000;
2 浙江省现代农业资源智慧管理与应用研究重点实验室, 浙江 湖州 313000)

摘 要: 与传统多视角聚类方法相比,多视角数据流聚类需要对持续到达的多视角数据进行实时融合与聚类分析,并要求聚类划分适应动态变化的数据分布。为实现此目标,本文提出了一种基于演化核嵌入的多视角数据流聚类算法(MSCEKE)。首先,通过设计多核融合机制,形成各视角聚类划分与共识核嵌入之间的交互协同,以提高多视角数据流的聚类性能。其次,结合支持向量域描述从历史数据中提取共识关键节点,利用共识关键节点的传播构建核嵌入的演化机制,从而适应多视角数据流动态变化的数据分布。最后,利用支持向量聚类从核嵌入中获取聚类划分。通过在多个真实数据上的测试,验证了本文所提方法的有效性。

关键词: 多视角聚类; 数据流聚类; 核学习; 支持向量域描述
中图分类号: TP301.6 **文献标志码:** A **文章编号:** 2095-2163(2026)05-0010-10

Multi-view streaming clustering with evolving kernel embedding

DENG Jiaqi[1,2], YIN Hongwei [1,2], HU Wenjun[1,2]
(1 School of Information Engineering, Huzhou University, Huzhou 313000, Zhejiang, China; 2 Zhejiang Province Key Laboratory of Smart Management & Application of Modern Agricultural Resources, Huzhou 313000, Zhejiang, China)

Abstract: Compared to traditional multi-view clustering methods, multi-view data streaming clustering requires real-time integration and clustering of continuously arriving multi-view data, while adapting to dynamically changing data distributions. To achieve this goal, this paper proposes a multi-view data streaming clustering algorithm based on evolving kernel embedding (MSCEKE). First, by designing a multi-kernel fusion mechanism, the interaction and collaboration between clustering partitions of different views and consensus kernel embedding are formed to improve the clustering performance of multi-view data streams. Secondly, by combining support vector domain description to extract consensus key nodes from historical data, and forming the evolution of kernel embedding through the propagation of these key nodes, the algorithm adapts to the dynamically changing data distribution of multi-view data streams. Finally, support vector clustering is utilized to obtain the clustering partitions. Extensive experiments on several real world datasets validate the effectiveness of our proposed method.

Key words: multi-view clustering; streaming clustering; kernel learning; support vector domain description

0 引 言

数据流是指在数据采集过程中,以极快的速度连续生成的大规模无界数据元素序列。例如,网络流[1]、图像流[2]、短文本流[3]、数值流[4]、事件流[5]和 Web 点击流[6]等。这些数据流的产生速度快、数量庞大,无法保留所有数据元素。多视角数据流则指从不同视角或维度同时采集和处理的数据流,每个视角描述同一现象的不同方面。例如,监控系统中[7],不同的摄像头会从不同的角度同时采集视频数据;网络安全中,不同的网络传感器会从不同的网络层面同时收集流量数据;气象卫星遥感中[8],多个卫星会从不同的轨道同时获取气象数据。多视角数据流能提供更丰富的信息,有助于更准确地进行

基金项目: 国家自然科学基金(62206094); 湖州市公益性应用研究项目(2021GZ05)。
作者简介: 邓家奇(1998—),男,硕士研究生。主要研究方向:数据挖掘,多视角聚类; 胡文军(1977—),男,博士,教授。主要研究方向:机器学习,模式识别。
通信作者: 尹宏伟(1990—),男,博士,副教授。主要研究方向:机器学习,数据挖掘,聚类分析。Email:02713@zjhu.edu.cn。
收稿日期: 2024-09-03

数据分析，但也显著增加了数据处理的复杂度，需要考虑各视角之间的关系和协调，开发相应的算法来整合和利用这些多维度的信息。

尽管单视角数据流聚类已取得了一定成功，但多视角数据流的高效聚类仍面临诸多挑战。现有多视角聚类方法多针对静态数据，难以直接处理流式数据。多视角数据流聚类需要在有限计算资源下有效捕捉簇的变化规律，发现任意形状的簇，并整合多个视角中的特征。已有多视角聚类研究[9]表明，不同视角间的信息往往具有互补性。简单地拼接特征会忽略了视角间的差异，进而影响聚类精度。现有的图数据流算法多依赖静态方法获取关键信息，或需人工选取关键信息，存在一定的局限性。

综上所述，本文提出了一种基于演化核嵌入的多视角数据流聚类算法。主要思想是设计多核融合机制获取数据流的共识核嵌入，利用位置正则化支持向量域描述[10]提取共识关键节点，随后通过支持向量聚类对每个视角进行聚类，再经过多数投票得到最终聚类标签，从而捕获任意形状的簇。通过跟踪块间共识关键节点的标签变化，可观察到簇的演化过程。由于共识关键节点只占数据的极小部分，该算法在有限计算资源下保持高效性。同时，算法将共识核嵌入学习与聚类过程进行协同优化，提升了聚类的有效性与灵活性，可更好地应对多视角数据流聚类的挑战。

本文的其余部分主要内容如下：第二节简要回顾相关工作，涵盖数据流聚类和多视角聚类两大方向；第三节将详细描述所提出的 MSCEKE 方法，具体包括多核融合机制、演化核嵌入、优化过程以及计算复杂度分析；第四节将报告和分析实验结果；第五节总结全文并展望未来工作。

1　相关工作

BIRCH 算法[11]是最早用于处理大型数据的聚类算法之一，通过 CF 树执行分层聚类。但由于缺乏处理概念漂移的机制，并不适用于数据流聚类。Guha 提出的 StreamKM 算法[12]扩展了 K-median 算法，能够一次遍历数据流完成聚类，但仍无法处理概念漂移，难以捕捉簇的演化。为此，Aggarwal 提出了 CluStream 框架[13]，该框架通过引入微簇概念，在在线和离线组件之间实现摘要统计信息的更新，可有效应对概念漂移。

在高维数据流聚类方面，Aggarwal 提出了 HPStream[14]，通过降维实现高效聚类。Alaettinoğlu 提出的 EMCStream 算法[15]结合流形嵌入和投影算法来处理高维数据流，并检测和适应概念漂移，以保持聚类质量的稳定性。Yoo 提出的 StrmSC 算法[16]，通过流式拉普拉斯矩阵近似和矩阵缩略图技术进行谱聚类，使其能高效处理高维数据流并捕捉任意形状的簇。

针对任意形状簇的捕获。在文献[17]中，Cao 提出将 DBSCAN[18]算法扩展到数据流聚类任务中，产生了一种新的数据流聚类算法 DenStream。该算法通过引入核心微簇的概念，可以处理任意形状的簇并及时识别异常值。Chen 等[19]开发了 D-Stream 框架，该框架可以利用网格划分空间并估计密度，实现数据流聚类。Wang 等[20]利用支持向量域描述和支持向量聚类提出的 SVStream 算法[21]，通过支持向量聚类高效捕捉任意形状簇，且内存占用较小。

针对多视角数据流聚类，Chen 等在文献[22]中提出多视角表示学习方法，通过生成公共稀疏亲和矩阵并应用谱聚类算法完成聚类。Huang[23]等提出 MVStrean 算法，该算法通过将多个视角映射到公共高维空间，利用支持向量构建任意形状簇边界，并捕捉簇的演化过程。

尽管上述数据流聚类算法已取得一定进展，但大多数算法都是针对单视角数据流设计的。多视角数据流聚类算法中，MVRL 算法依赖人工选择代表集，影响聚类效果。MVStream 算法在视角融合时未考虑数据分布差异，使用统一参数进行聚类也会降低精度。其他多视角聚类方法，如多视角 EM 算法[24]，多视角亲和传播[25]，基于可变权重和多核学习的多视角聚类算法[26]，全局和局部一致的多视角子空间聚类[27]，带特征选择的加权多视图聚类[28]，结构化低秩矩阵分解的多视角谱聚类[29]，以及用于多视角聚类的统一图学习框架[30]等，多为静态数据设计，难以应对数据流聚类的挑战。为此，本文提出一种基于演化核嵌入的多视角数据流聚类算法以解决上述问题。

2　提出方法

在多视角数据流中，样本以数据块形式按时间持续到达。在 t 时刻到达的样本集合组成数据块 $X_t=\{X_t^1,X_t^2,\cdots,X_t^m\}$，其中，$X_t^m=\mathbb{R}^{n^t\times d^m}$，$n^t$ 为第 t 块中样本数量，d^m 是第 m 视角的特征维度。首先设计多核融合机制获取多视角数据流的共识核嵌入，然后结合位置正则化支持向量域描述，构建演化学习

模型以捕获共识关键节点,通过共识关键节点对当前样本的数据分布进行概要描述,最后利用支持向量聚类实现聚类划分。

2.1 多核融合机制

在多视角核学习中,共识核嵌入的学习过程与聚类过程处于分离状态,导致两者之间缺乏协同交互,降低了多视角核学习方法的聚类性能。为解决此问题,本文引入文献[31]的框架,以消除过程分离导致的次优解问题,目标函数如下:

$$\begin{aligned}&\max_{H,Q^v,\boldsymbol{P}^v}\mathrm{Tr}(H^{\mathrm{T}}\sum_{v=1}^{m}Q^v\boldsymbol{P}^v)\\&\text{s.t. } H^{\mathrm{T}}H=I,K=HH^{\mathrm{T}}\\&(Q^v)^{\mathrm{T}}Q^v=I,(\boldsymbol{P}^v)^{\mathrm{T}}\boldsymbol{P}^v=I\end{aligned}\tag{1}$$

其中,Q^v 为第 v 视角的聚类划分;$\boldsymbol{P}^v$ 为第 v 视角的投影矩阵;K 为多视角数据的共识核;H 为多视角数据的共识核嵌入。

在以上目标函数中,通过投影矩阵驱使各视角的聚类划分与共识核嵌入之间保持一致,使得共识核嵌入引导各视角的聚类,同时各视角的聚类也能监督共识核嵌入的学习。

2.2 基于演化核嵌入的多视角数据流聚类

在多视角数据流中,为对当前时刻所有样本 X_t 的数据分布进行概要描述,采用位置正则化支持向量域描述(PSVDD)[32],从共识核空间中提取关键节点。基于公式(1)获取 X_t 的共识核嵌入,定义包含所有样本的最小球体如下:

$$\begin{aligned}&\min_{r,\mu,\xi_i}(r^2+\sum_{i=1}^{n}w_i\xi_i)\\&\text{s.t. } \|h_i-\mu\|^2\leqslant r^2+\xi_i,\forall i=1,\cdots,n\end{aligned}\tag{2}$$

其中,r 为球体半径;μ 为球体中心;h_i 为样本的共识核嵌入;$\xi_i\geqslant 0$ 为松弛变量;w_i 为位置权重。通过引入松弛变量允许一定程度偏差,避免最小球的过拟合问题。w_i 的计算方法如下:

$$w_i=\max_{l=1,\cdots,n}\{d_l^{\phi}\}-d_i^{\phi},w_i=\frac{w_i}{\max\limits_{l=1,\cdots,n}w_l}\tag{3}$$

其中,d_l^{ϕ} 为数据对象 l 到核空间中所有数据对象质心的距离,计算方式如下:

$$d_l^{\phi}=\|h_l-\frac{1}{n}\sum_{i=1}^{n}h_i\|^2=K_{l,l}+\frac{1}{n^2}\sum_{i,j=1}^{n}K_{i,j}-\frac{2}{n}\sum_{j=1}^{n}K_{l,j}\tag{4}$$

根据拉格朗日乘子法,令 β_i 和 α_i 为拉格朗日乘子,则公式(2)中的最小化问题可转化为:

$$\begin{aligned}\min_{r,\mu,\alpha_i,\xi_i,\beta_i}[&r^2+\sum_{i=1}^{n}w_i\xi_i-\sum_{i=1}^{n}\alpha_i\xi_i\\&-\sum_{i=1}^{n}\beta_i(r^2+\xi_i-\|h_i-\mu\|^2)]\end{aligned}\tag{5}$$

将公式(5)分别对 r,μ,ξ_i 求偏导并令偏导数为0,得到以下结果:

$$\sum_{i=1}^{n}\beta_i=1,\ \mu=\sum_{i=1}^{n}\beta_ih_i,\ \beta_i=w_i-\alpha_i\tag{6}$$

根据 KKT 条件以及 Wolfe 对偶可消除其中变量 r、μ、ξ_i、α_i,从而将以上拉格朗日函数转化为关于 β_i 的沃夫对偶形式

$$\begin{aligned}&\max_{\beta_i}\sum_{i=1}^{n}\beta_ih_ih_i-\sum_{i,j=1}^{n}\beta_i\beta_jh_ih_j\\&\text{s.t. }\sum_{i=1}^{n}\beta_i=1,\ 0\leqslant\beta_i\leqslant w_i,\forall i=1,\cdots,n\end{aligned}\tag{7}$$

将公式(1)中多核融合机制与公式(7)中的支持向量域描述相结合,构造演化核嵌入学习的目标函数如下:

$$\begin{aligned}\max_{H,Q^v,P^v,\beta_i}[&\mathrm{Tr}(H^{\mathrm{T}}\sum_{v=1}^{m}Q^vP^v)-\lambda(\sum_{i=1}^{n}\beta_i(HH^{\mathrm{T}})_{i,i}-\\&\sum_{i,j=1}^{n}\beta_i\beta_j(HH^{\mathrm{T}})_{i,j})]\end{aligned}\tag{8}$$

其中,λ 为多核融合与支持向量域构造的权衡参数。通过该目标函数将数据从数据原始空间映射到共识核空间。在此空间中形成包含所有映射数据对象的最小球体,并从中提取共识关键节点。这些共识关键节点提供了灵活且准确的多视角数据描述。由于共识核嵌入与聚类划分之间的协同交互,能够得到最佳共识核空间,从而实现了演化核嵌入。本文采用交替优化策略对上述目标函数进行求解,优化过程如下:

1)更新 $\{P^v\}_{v=1}^m$:在固定 H,$\{Q^v\}_{v=1}^m$,β_i 的情况下,公式(1)中关于 $\{P^v\}_{v=1}^m$ 的优化问题等价于:

$$\begin{aligned}&\max_{P^v}(P^v)^{\mathrm{T}}(Q^v)^{\mathrm{T}}H\\&\text{s.t. }H^{\mathrm{T}}H=I\end{aligned}\tag{9}$$

2)更新 $\{Q^v\}_{v=1}^m$:在固定 H,$\{P^v\}_{v=1}^m$,β_i 的情况下,公式(1)中关于 $\{Q^v\}_{v=1}^m$ 的优化问题等价于:

$$\max_{Qv}Q^vH(P^v)^{\mathrm{T}}\tag{10}$$

3)更新 H:在固定 $\{Q^v\}_{v=1}^m$,$\{P^v\}_{v=1}^m$,β_i 的情况下,对 H 按其元素进行优化。令:

$$B=\sum_{v=1}^{m}Q^vP^v\tag{11}$$

通过设置目标函数对 $H_{i,a}$ 求偏导并置为0,可以得到共识核嵌入的更新规则:

$i\neq a$ 时,

$$H_{i,a} \leftarrow \frac{B_{i,a} + \beta_i \sum_{j=1}^{n} \beta_j H_{j,a}}{2\lambda\beta_i} \tag{12}$$

$j = a$ 时，

$$H_{a,a} \leftarrow \frac{B_{a,a} + \beta_a \beta_j H_{j,a}}{2\lambda\beta_a} \tag{13}$$

更新共识核嵌入 H 后，通过二次规划求解 β_i 以及按照公式(3)求解 w_i 的值。其中，$\{Q^v\}_{v=1}^m$ 通过对每个视角进行 Kmeans 得到初始化，$\{\beta_i\}$ 的初始值通过随机初始化得到。在第 t 个块到来时，将前 $t-1$ 个块的共识关键节点与第 t 个块一起组成新的数据块开始新的一轮迭代。

根据拉格朗日乘子 β_i 的值和相应的权重 w_i，将数据样本分为以下 3 种类型，其中支持向量即为共识关键节点，目标函数的详细求解过程如算法 1 所示：

1)内点：$\beta_i = 0$，内点位于超球体内部；

2)支持向量：$0 < \beta_i < w_i$，支持向量位于超球体表面；

3)边界支持向量：$\beta_i = w_i$，边界支持向量位于超球体外部。

算法 1　演化核嵌入

输入　$X_t = \{X_t^1, X_t^2, \cdots, X_t^m\}$

输出　β_i, w_i

初始化：$\{Q^v\}_{v=1}^m$，β_i

1 通过公式(1)获取 H

2 通过公式(2)构造最小球体

重复以下步骤直到收敛：

3 通过公式(9)更新 $\{P^v\}$

4 通过公式(10)更新 $\{Q^v\}$

5 通过公式(11)、公式(12)更新 H

6 通过二次规划以及公式(3)求出 $\{\beta_i\}$ 和 w_i

通过演化核嵌入学习获取多视角数据流的共识关键节点后，利用经典支持向量聚类[20]作用于核嵌入，可获取各视角的硬聚类划分，并通过投票方式获取最终的聚类标签。算法总体流程如算法 2 所示。

算法 2　基于演化核嵌入的多视角数据流聚类算法

输入　$X_t = \{X_t^1, X_t^2, \cdots, X_t^m\}$，前 $t-1$ 个块的共识关键节点

输出　当前块的聚类标签，共识关键节点

初始化　$\{Q^v\}_{v=1}^m$，$\{\beta_i\}$

重复以下步骤直到没有数据到达：

1 获取第 t 个数据块中的数据对象，并结合第 $t-1$ 个块中的共识关键节点组成临时数据 Z_t

2 通过公式(1)学习到共识核嵌入 H

3 通过公式(2)构造最小球体

4 通过算法 1 得到 $\{\beta_i\}$ 和 w_i，选择共识关键节点

5 对每个视角结合 SVC 进行聚类

6 对所有视角聚类标签进行多数投票

2.3　算法复杂度分析

本文所提算法的主要复杂度包括以下几个部分：

1)通过 Kmeans 计算聚类划分并学习到共识核嵌入，时间复杂度为 $O(|Z_t|k^2 + v(k^3 + |Z_t|k^2))$。

2)再通过二次规划求解 $\{\beta_{i,t}, i = 1, \cdots, |Z_t|\}$，时间复杂度为 $O(|Z_t|^3)$。

3)对 $\{Z_t^v, v = 1, \cdots, m\}$ 中的数据通过 SVC 进行聚类，时间复杂度为 $O(mh|Z_t|^3)$。

所以 MSCEKE 算法的总时间复杂度为 $O(|Z_t|k^2 + v(k^3 + |Z_t|k^2) + (mh+1)|Z_t|^3)$，完成 MSCEKE 算法复杂度分析后，通过 MSCEKE 聚类算法流程示意图直观呈现多视角数据块、共识核嵌入、共识关键节点的演化聚类过程(见图 1)。

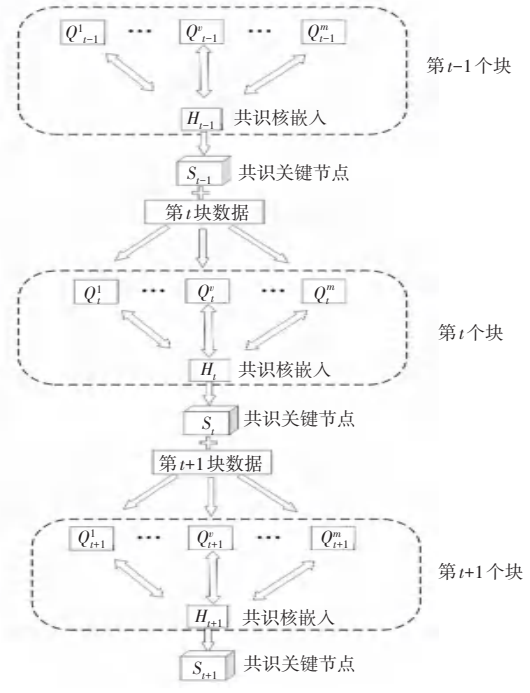


图 1　MSCEKE 聚类算法流程示意图

Fig. 1　The flow diagram of MSCEKE clustering algorithm

3　仿真实验

本节通过在 4 个数据集上的实验，评估所提出的多视角数据流聚类算法的性能及效率。实验主要内容包括对比实验、参数实验以及对算法运行时间

进行对比。实验的硬件环境为 Windows 10, Intel Core i7,内存为 16 GB。需要注意的是 CluStream, DenStream 以及 EmcStream 的编程环境为 PyCharm,其余算法的编程环境为 MATLAB2020。随后采用了聚类精度(Accuracy, Acc)和归一化互信息(Normalized Mutual Information, NMI)作为验证指标,其取值范围均为[0,1]。数值越接近于 1 则代表聚类结果与真实标签的一致性越高。

3.1 实验数据

为验证本文算法有效性,采用表 1 中的 4 个公开数据集进行实验。

表 1 数据集信息

Table 1 Dataset information

数据集	视角数	类簇个数	样本数量
Caltech101-20	6	20	2 386
Scene	4	8	2 688
Animals	4	20	11 673
NUSWIDEOBJ	5	31	30 000

(1) Caltech101-20 数据集:该数据集是来自多个内核学习存储库的数据集。包含 102 个类的 9 144 张图像,其中每个样本包含 6 个视角,分别是 LBP、GABOR、HOG、Wavelet moments、Cenhist 和 GIST。

(2) Scene 数据集:其包含了 8 个类别共 2 688 个样本。一共分为 4 个视角,每个视角的维度分别为 512、432、256、48。

(3) Animals 数据集:该数据集包含了动物的特征和类别。由 11 673 张图像组成,可分为 20 类动物。该数据集中的每个图像都有 4 个视角,其维度分别为 2 689、2 000、2 001 和 2 000。

(4) NUSWIDEOBJ 数据集:该数据集包含了 31 个类别共 30 000 张图像。包括 65 维的颜色直方图(CH)、226 维的颜色矩(CM)、74 维的边缘分布(ED)、129 维的小波纹理(WT)和 145 维的颜色相关性(CORR)。

3.2 对比算法

EmcStream[15]:该方法是一种在线嵌入与聚类方法。可实时检测并适应概念漂移,进而完成数据聚类。

DenStream[17]:该方法是对 DBSCAN 的扩展,属于基于密度的两阶段数据流聚类算法。其中在线部分维护潜在微簇和孤立点微簇。离线阶段执行聚类操作。

SVStream[21]:该方法是一种基于支持向量的数据流聚类方法。先对数据进行预处理,再利用 SVC(支持向量分类)完成聚类。

MVRL[22]:该方法是一种多视角数据流聚类方法,通过融合稀疏关联矩阵实现谱聚类。能够有效捕获数据的全局与局部结构。

MVStream[23]:该方法是一种基于支持向量的多视角数据流聚类算法。在提取支持向量后,对每个视角分别进行支持向量聚类。

3.3 参数分析

本节将分析两个参数对所提出的 MSCEKE 算法性能的影响,分别为块大小 m 和权衡参数 λ。在分析其中一个参数时,另一个参数被固定。

(1)块大小:首先分析当将权衡参数 λ 固定为 1.0 时,块大小 m 对 MSCEKE 性能的影响。在 Animal 和 NUSWIDEOBJ 数据集上,块大小 m 的取值范围为 {50, 100, 150, 200, 250, 300}。在 Caltech101-20 和 Scene 数据集上,块大小 m 的取值范围设为{50,60,70,80,90,100,110,120,130,140,150}。图 2 展示了 NUSWIDEOBJ、Caltech101-20、Animal 和 Scene 数据集上不同分块大小的平均 NMI 和 ACC 指标。实验结果表明,随着分块大小 m 的增加,MSCEKE 的性能呈持续下降趋势。

因此,针对这些数据集,分块大小分别设置为 100、80、50 和 60。需要注意的是,尽管参数分析提示了分块大小如何影响性能的见解,但分块大小应根据 MSCEKE 聚类任务的具体情况确定,而不是为了产生最佳结果而选择。分块大小在不同的时间段可能会变化。例如,根据任务需求确定的一个小时、一天或其他特定时间段内到达的数据对象视为一个数据块。

(2)权衡参数:分析在固定块大小 m 的情况下,权衡参数 λ 对 MSCEKE 性能的影响。将 4 个测试数据集 Caltech101-20、Scenes、Animal 和 NUSWIDEOBJ 的块大小分别设置为 80、60、50 和 100,权衡参数 λ 选择在 $\lambda = 0.5、0.7、\cdots、1.5$ 的范围内。图 3 中报告了所有分块的 NMI 和 ACC 的平均值作为 λ 的函数。从图 3 中可以看出,当设置 λ 在 $\lambda = 0.5、0.7、\cdots、1.5$ 的范围内时,可以产生相对稳定的结果。特别是当设置 λ 约为 1.0 时,所提出的 MSCEKE 算法在 NMI 和 ACC 方面均取得了最佳结果,λ 取 1 时也就代表着应该平等对待核构建模型和领域描述构建模型。因此,在本节的实验中,将 4 个测试数据集上的权衡参数 λ 设置为 1.0。需要注意的是,根据本节的研究发现,当将 λ 设置为较

小的值，例如 $\lambda = 0.1$，或将 λ 设置为较大的值，例如 $\lambda = 500$ 时，反而会生成较差的聚类结果。这意味着非常不均衡地处理核融合模型和支持向量域描述构建模型的话，会严重影响聚类性能。在极端情况下，当设置 $\lambda = 0.01$ 时，即几乎忽略支持向量域描述模型时，会生成非常不准确的聚类结果。另一方面，当设置 $\lambda = 1\,000$ 时，即几乎忽略核融合模型时，也会生成非常不准确的聚类结果。通过实验发现在 $[0.5, 1.5]$ 的范围内测试 λ 是一个不错的选择。

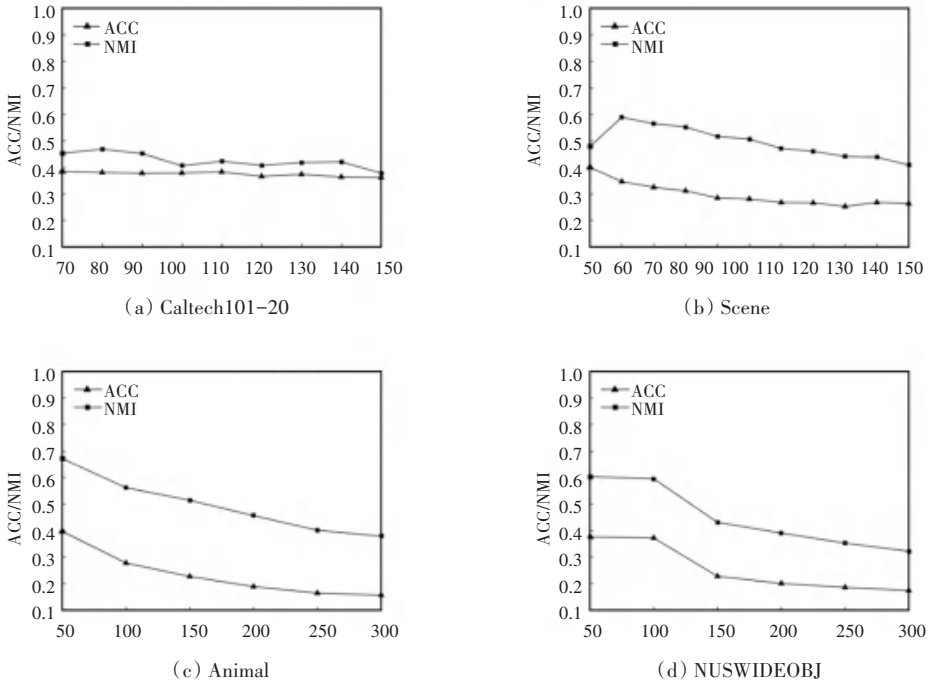


图 2　块大小的参数分析：不同块大小下 4 个数据集中所有块的 NMI 和 ACC 平均值

Fig. 2　Parameter analysis of chunk size: Average NMI and ACC of all chunks in four datasets under different chunk sizes chunk sizes chunk sizes

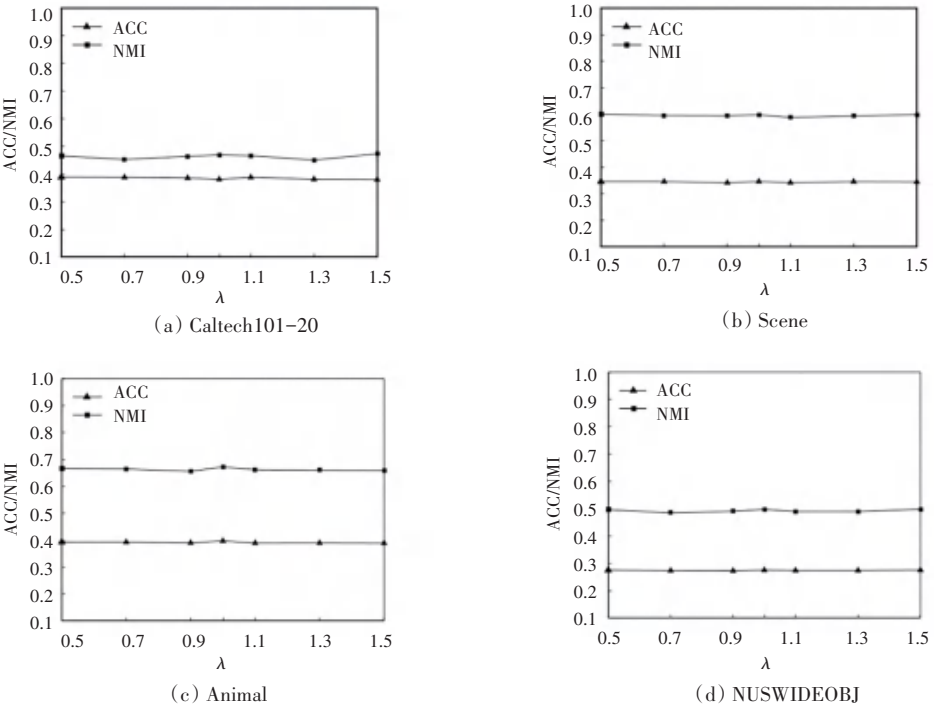


图 3　权衡参数分析：不同权衡参数下 3 个数据集中所有块的 NMI 和 ACC 平均值

Fig. 3　Analysis of trade-off parameters: Average NMI and ACC of all chunks in three datasets under different trade-off parameters

3.4 对比试验

本节将通过表 2 和表 3 中的 ACC、NMI 指标比较所提出的 MSCEKE 方法与现有的 5 种方法在聚类性能上的表现。

表 2 四个数据集上的 ACC 比较

Table 2 Comparison of ACC on four datasets (Mean 代表平均值,Std 代表标准差)

Method	Caltech101-20		Scene		Animal		NUSWIDEOBJ	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
EmcStream	0.493	0.041	0.450	0.083	0.243	0.051	0.181	0.039
DenStream	0.406	0.033	0.442	0.070	0.165	0.030	0.156	0.027
SVStream	0.523	0.074	**0.458**	0.050	0.460	0.016	0.356	0.016
MVRL	**0.561**	0.097	0.438	0.063	0.545	0.025	0.260	0.016
MVStream	0.494	0.136	0.398	0.056	**0.574**	0.045	0.339	0.004
Our	0.502	0.072	0.421	0.022	0.504	0.012	**0.381**	0.013

表 3 四个数据集上的 NMI 比较

Table 3 Comparison of NMI on four datasets(Mean 代表平均值,Std 代表标准差)

Method	Caltech101-20		Scene		Animal		NUSWIDEOBJ	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
EmcStream	0.229	0.033	0.468	0.059	0.199	0.019	0.070	0.016
DenStream	0.590	0.057	0.411	0.157	0.339	0.111	0.191	0.069
SVStream	0.620	0.075	0.511	0.055	0.673	0.023	0.566	0.053
MVRL	0.426	0.109	0.357	0.064	0.662	0.035	0.224	0.017
MVStream	0.308	0.295	0.268	0.260	0.327	0.221	0.090	0.057
Our	**0.625**	0.072	**0.658**	0.009	**0.774**	0.017	**0.685**	0.013

对于 SVStream、MVStream 和 MSCEKE,由于算法中使用了高斯核函数,其核参数设置为当前数据块中所有数据点之间平方距离的平均值。对于所有数据集,将多个视角的特征连接在一起作为一个视角,结合所有视角将数据以流式方式输入到算法中,并报告所有视角中的最佳聚类结果。

对 4 个测试数据集以分块的方式作为流式输入到数据流聚类算法中,根据每个块的聚类结果评估 NMI 和 ACC,并报告所有块的平均值、标准差。最后将 NMI 最大值和最小值体现在图 4 和图 5 中。

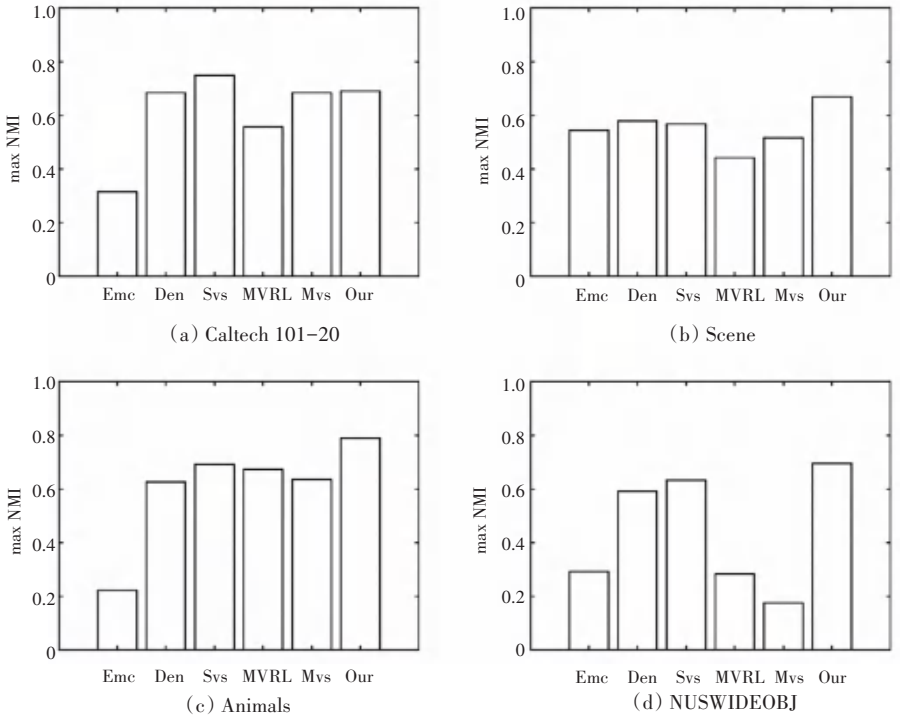


图 4 各算法在不同数据集上的 NMI 最大值比较

Fig. 4 Comparison of max NMI values of different algorithms on various datasets

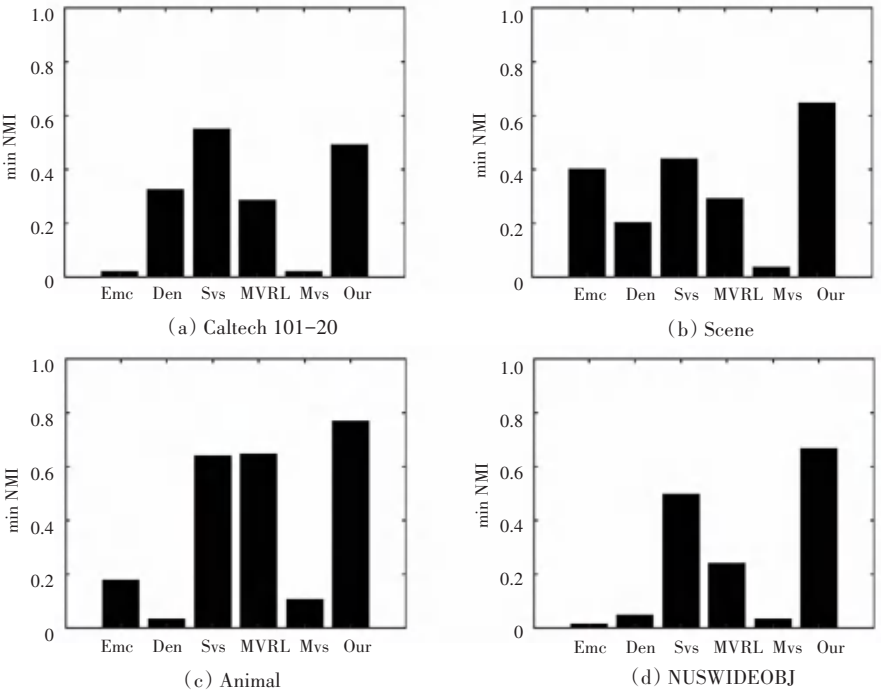


图 5　各算法在不同数据集上的 NMI 最小值比较

Fig. 5　Comparison of min NMI values of different algorithms on various datasets

在表 2、表 3 中，对于数据流聚类算法，分别计算了每个分块的 NMI 和 ACC 值，然后报告了所有分块的 NMI 和 ACC 的平均值、标准差。其中最大值用粗体标记，第二大的值用下划线标记。从表中可以看出，本文所提出的 MSCEKE 算法在 NUSWIDEOBJ 数据集上相较于 ACC 次优的 SVStream 算法提高了 7.02%，相较于 NMI 次优的 SVStream 算法提高了 21.02%，在 Animal 数据集上相较于次优的 SVStream 算法提高了 15.00%，在 Scene 数据集上相较于次优的 SVStream 算法提高了 28.70%。本文提出的 MSCEKE 在中大型数据集上表现出色。此外，从最大值和最小值对比来看，MSCEKE 算法在中大型数据集上也相对比其他数据流聚类方法更稳定，说明 MSCEKE 在理解数据的全局模式和簇间关系上有很大的优势。

3.5　运行时间比较

图 6 绘制了 8 种算法的执行时间(单位：s)。由于 SVC 在 Animal 和 NUSWIDEOBJ 数据集上因内存不足而未能分区，对应的条形图中未显示执行时间。从图中可以看出，总体而言，MSCEKE 的执行时间与现有方法相比是可以接受的。MSCEKE 总体时间花费较大。尤其是对于样本量较大的数据尤为明显。因为 MSCEKE 算法需要迭代优化寻找最优的共识核嵌入。整体结果验证了 MSCEKE 的效率，虽然时间花费较大，但是为了能得到更好的聚类结果是值得的。

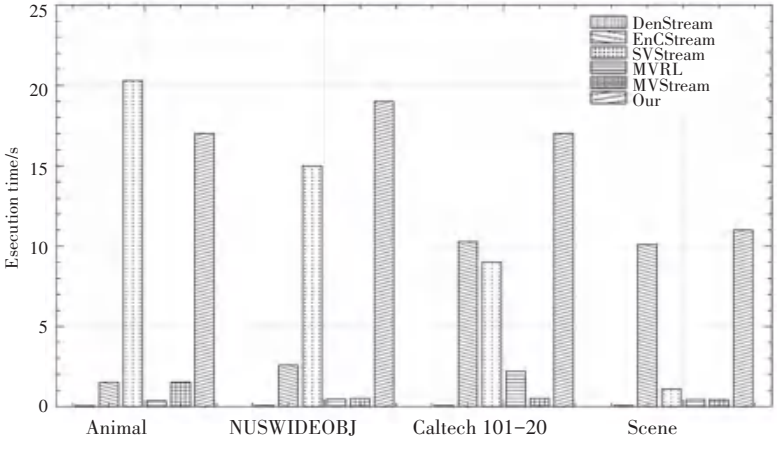


图 6　块平均运行时间

Fig. 6　Average run time of each chunk

4 结束语

本文提出了 MSCEKE 算法。通过设计多核融合机制,形成各视角聚类划分与共识核嵌入之间的交互协同,其次,结合支持向量域描述提取历史数据的共识关键节点,并通过共识关键节点的传播形成核嵌入的演化,从而适应多视角数据流动态变化的数据分布,最后结合支持向量聚类获取聚类划分。利用支持向量聚类,可以发现每个视角中任意形状的簇。通过跟踪两个连续块之间的共识关键节点的簇标签,可以捕获与概念漂移相关的簇演变,即新簇的出现、旧簇的合并、旧簇的分裂和旧簇的消失。通过在多个真实数据上的测试,验证了本文所提方法的有效性,能够高效捕捉多视角数据流的聚类结构。在未来的工作中计划将 MSCEKE 扩展到一些视角非完备的情况下。视角非完备指的是在多视角数据中,某些视角可能缺少完整的信息或者某些方面的信息不完整,从而导致整个数据集的信息不完整或不充分。包括但不限于以下几种情况。第一种是视角级别的非完备,即每个视角只包含部分信息,但所有视角一起包含了关于潜在完整表示的信息。第二种情况是特征级别的非完备,即一些视角对一些或所有数据对象的特征缺失。第三种情况是对象级别的非完备,即一些视角可能存在缺失的数据对象。在本文算法中,视角的核融合并未考虑视角非完备的情况。未来将设计一些有效的策略来处理非完备视角数据,进一步提高算法的性能和适用性。

参考文献

[1] MIAO X, LIU Y, ZHAO H, et al. Distributed online one-class support vector machine for anomaly detection over networks[J]. IEEE Transactions on Cybernetics, 2019, 49(4): 1475-1488.

[2] CHI L, LI B, ZHU X, et al. Hashing for adaptive real-time graph stream classification with concept drifts [J]. IEEE Transactions on Cybernetics, 2018, 48(5): 1591-1604.

[3] LI P, HE L, WANG H, et al. Learning from short text streams with topic drifts[J]. IEEE Transactions on Cybernetics, 2018, 48(9): 2697-2711.

[4] POULARAKIS S, KATSAVOUNIDIS I. Low-complexity hand gesture recognition system for continuous streams of digits and letters[J]. IEEE Transactions on Cybernetics, 2016, 46(9): 2094-2108.

[5] DECROOS T, DZYUBA V, VAN HAAREN J, et al. Predicting soccer highlights from spatio-temporal match event streams[C]// Proceedings of the AAAI Conference on Artificial Intelligence. AAAI, 2017: 1302-1308.

[6] ABOLHASSANI M, ESFANDIARI H, HAJIAGHAYI M T, et al. Pricing for data streams [C]//Proceedings of the AAAI Conference on Artificial Intelligence. AAAI, 2017: 291-297.

[7] DEVAGIRI V M, BOEVA V, ABGHARI S. A multi-view clustering approach for analysis of streaming data [C]// Proceedings of Artificial Intelligence Applications and Innovations. Cham: Springer, 2021: 169-183.

[8] SUN X, LI B, SHI T, et al. Real-time processing for remote sensing satellite data based on stream computing[C]//Proceedings of International Conference on Signal, Information and Data Processing. Piscataway, NJ: IEEE, 2019: 1-8.

[9] YANG Y, WANG H. Multi-view clustering: A survey[J]. Big Data Mining and Analytics, 2018, 1(2): 83-107.

[10] TAX D M J, DUIN R P W. Support vector data description[J]. Machine Learning, 2004, 54(1): 45-66.

[11] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases [C]// Proceedings of the 1996 International Conference on Management of Data. New York: ACM, 1996: 103-114.

[12] GUHA S, MEYERSON A, MISHRA N, et al. Clustering data streams: Theory and practice [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(3): 515-528.

[13] AGGARWAL C C, PHILIP S Y, HAN J, et al. A framework for clustering evolving data streams [C]// Proceedings of 29th International Conference on Very Large Data Bases. VLDB, 2003: 81-92.

[14] AGGARWAL C C, HAN J, WANG J, et al. A framework for projected clustering of high dimensional data streams [C]// Proceedings of the 30th International Conference on Very Large Data Bases. VLDB, 2004: 852-863.

[15] ZUBAROGLU A, ATALAY V. Online embedding and clustering of evolving data streams[J]. Statistical Analysis and Data Mining, 2023, 16(1): 29-44.

[16] YOO S, HUANG H, KASIVISWANATHAN S P. Streaming spectral clustering [C]//Proceedings of the International Conference on Data Engineering. Piscataway, NJ: IEEE, 2016: 637-648.

[17] CAO F, ESTERT M, QIAN W. Density-based clustering over an evolving data stream with noise [C]//Proceedings of the 6th International Conference on Data Mining. Piscataway, NJ: IEEE, 2006: 328-339.

[18] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proceedings of the 2th International Conference on Knowledge Discovery and Data Mining. AAAI, 1996: 226-231.

[19] CHEN Y, TU L. Density-based clustering for real-time stream data[C]//Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007: 133-142.

[20] WANG C D, LAI J H, HUANG D, et al. SVStream: A support vector-based algorithm for clustering data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 25(6): 1410-1424.

[21] BEN-HUR A, HORN D, SIEGELMANN H T, et al. Support vector clustering [J]. Journal of Machine Learning Research, 2001, 2(12): 125-137.

[22] CHEN J, YANG S, WANG Z. Multi-view representation learning for data stream clustering[J]. Information Sciences, 2022, 613: 731-746.

[23] HUANG L, WANG C D, CHAO H Y, et al. MVStream:

Multiview data stream clustering[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(9): 3482-3496.

[24] BICKEL S, SCHEFFER T. Multi-view clustering [C]// Proceedings of the 4th International Conference on Data Mining. Piscataway, NJ: IEEE, 2004: 19-26.

[25] WANG C D, LAI J H, PHILIP S Y. Multi-view clustering based on belief propagation[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(4): 1007-1021.

[26] ZHANG P, YANG Y, PENG B, et al. Multi-view clustering algorithm based on variable weight and MKL[C]//Proceedings of International Joint Conference on Rough Sets. Cham: Springer, 2017: 599-610.

[27] FAN Y, HE R, HU B G. Global and local consistent multi-view subspace clustering [C]//Proceedings of Asian Conference on Pattern Recognition. Piscataway, NJ: IEEE, 2015: 564-568.

[28] XU Y M, WANG C D, LAI J H. Weighted multi-view clustering with feature selection[J]. Pattern Recognition, 2016, 53: 25-35.

[29] WANG Y, WU L, LIN X, et al. Multiview spectral clustering via structured low-rank matrix factorization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(10): 4833-4843.

[30] LIANG Y, HUANG D, WANG C D. Consistency meets inconsistency: A unified graph learning framework for multi-view clustering[C]// Proceedings of International Conference on Data Mining. Piscataway, NJ: IEEE, 2019: 1204-1209.

[31] LIU X, ZHU X, LI M, et al. Late fusion incomplete multi-view clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(10): 2410-2423.

[32] WANG C D, LAI J H. Position regularized support vector domain description[J]. Pattern Recognition, 2013, 46(3): 875-884.