计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(8): 1833−1849, 2025

基于双端联合学习的多视图聚类

杜 亮[1] 李晓东[1] 陈 艳[2] 周 芃[3] 钱宇华[4]
[1]（山西大学计算机与信息技术学院 太原 030006）
[2]（太原理工大学计算机科学与技术学院 山西晋中 030600）
[3]（安徽大学计算机科学与技术学院 合肥 230601）
[4]（山西大学大数据科学与产业研究院 太原 030006）
（duliang@sxu.edu.cn）

Double-Ended Joint Learning for Multi-View Clustering

Du Liang[1], Li Xiaodong[1], Chen Yan[2], Zhou Peng[3], and Qian Yuhua[4]
[1]（*School of Computer and Information Technology*, *Shanxi University*, *Taiyuan* 030006）
[2]（*College of Computer Science and Technology*, *Taiyuan University of Technology*, *Jinzhong*, *Shanxi*, 030600）
[3]（*School of Computer Science and Technology*, *Anhui University*, *Hefei* 230601）
[4]（*Institute of Big Data Science and Industry*, *Shanxi University*, *Taiyuan* 030006）

Abstract Multi-anchor graph approaches have attracted more and more attention for their potential in addressing the challenges of large-scale multi-view clustering. However, existing methods leveraging multi-anchor graphs encounter several hurdles when they come to tackling this challenge. The consistency-anchored graph learning methods struggle with handling misaligned anchor graphs and necessitates additional post-processing with consistency graph, thereby constraining the accuracy and reliability of clustering outcomes. And the anchor graph ensemble clustering method fails to harness the complementary information from different views during the independent generation of candidate base clustering and overlooks the original anchor graphs during fusion, thus impacting the effectiveness and stability of clustering results. To address these challenges, we propose a novel approach based on double-ended joint learning for multi-view clustering. The method fully considers the duality between multi-anchor information and samples in multi-anchor graphs, achieving synchronized clustering between anchor-end and sample-end. Moreover, under the guidance of multi-anchor information, it achieves joint alignment between sample-end clustering and multiple anchor-end clustering. Unlike existing methods, the approach does not require direct learning of consistent anchor graph, thus can handle any type of anchor misalignment issues, and mitigate the negative impact of separate graph learning and partition on clustering performance. Additionally, it utilizes multiple anchor graphs for anchor-end clustering and sample-end clustering within a unified optimization framework, effectively addressing the limitations of base clustering and the ensemble stage in leveraging multiple anchor graphs. Experimental results demonstrate that the proposed method outperforms several comparative methods in terms of clustering performance and time consumption, effectively enhancing the clustering performance of multi-view data. The relevant code for the proposed method and

收稿日期：2024-03-16；**修回日期**：2024-12-09
基金项目：国家自然科学基金项目（62376146, 62176001）；山西省中央引导地方科技发展专项项目（YDZJSX20231D003）；安徽省高校优秀青年科学项目（2023AH030004）；山西省重点研发计划项目（202202020101004）；山西省自然科学基金项目（20210302123455）
This work was supported by the National Natural Science Foundation of China (62376146, 62176001), the Shanxi Provincial Central Guidance for Local Science and Technology Development Special Project (YDZJSX20231D003), the Outstanding Young Science Program of Anhui Higher Education Institustions (2023AH030004), the Key Research and Development Program of Shanxi Province (202202020101004), and the Natural Science Foundation of Shanxi Province (20210302123455).
通信作者：钱宇华 (jinchengqyh@126.com)

comparative methods is provided in the supplementary material: http://github.com/lxd1204/DLMC.

Key words multi-view clustering; anchor end; sample end; alignment; ensemble

摘 要 在应对大规模多视图聚类这一挑战时都面临多个问题. 其中, 一致性锚点图学习方法难以处理锚点图不对齐问题, 并且过度依赖一致性图, 限制了其聚类结果的准确性和可靠性; 锚点图集成聚类方法则是在基聚类器的生成与融合过程中割裂了不同锚点图之间的联系, 影响了其聚类效果的有效性和稳定性. 为解决这些问题, 提出了一种基于双端联合学习的新型多视图聚类方法. 该方法充分考虑了多锚点图信息和锚点端聚类对样本端聚类的联合作用, 实现了锚点端聚类和样本端聚类同步进行, 并通过对多锚点图信息的综合实现了样本端聚类与多个锚点端聚类的集成对齐. 与现有方法不同, 该方法无需直接学习一致性锚点图, 可以处理任意类型的锚点不对齐问题, 并且规避了图学习与图划分分步处理对聚类性能的不利影响. 此外, 其在一个完整的优化框架中同时利用多个锚点图进行锚点端聚类和样本端聚类, 有效解决了基聚类器生成阶段无法利用除自身外的其他锚点图和集成阶段无法充分利用所有锚点图的问题. 实验结果表明, 所提出的方法在聚类性能和时间消耗方面均优于多个对比方法, 有效增强了多视图数据的聚类性能. 所提出方法以及所采用对比方法的相关代码附可在 http://github.com/lxd1204/DLMC 中查询.

关键词 多视图聚类; 锚点端; 样本端; 对齐; 集成

中图法分类号 TP181

DOI: 10.7544/issn1000-1239.202440175 **CSTR:** 32373.14.issn1000-1239.202440175

在当今数据科学领域, 随着数据规模的不断增大和数据来源的多样化, 多视图(数据的一组不同表示或特征集合)聚类作为一种有效的数据分析工具受到了广泛关注. 在众多应用场景中, 特别是在大规模数据集上, 多视图聚类的需求愈发迫切. 然而, 针对大规模多视图数据进行高效而准确的聚类仍然是一个具有挑战性的问题. 近年来, 基于锚点图(锚点则是从数据中所选择的具有代表性的样本, 利用锚点与所有样本之间的关系构建的图称之为锚点图)的多视图聚类方法应运而生, 为解决这一问题提供了新的思路和方法[1]. 这些方法通常包含多锚点图构造和多锚点图信息挖掘 2 个关键过程.

在多锚点图构造过程中, 传统方法通常是从每个视图中提取一个锚点图. 例如, 可以利用随机子集或 K 均值等获取一组锚点集[2], 然后通过概率邻近图[3]构建或子空间学习[4]等方法形成一个锚点图. 然而, 由于这样得到的不同视图的锚点图使用的锚点集合可能不同, 这给后续的多锚点图挖掘带来了困难[5]. 为了应对多视图数据的挑战, 一些研究[6-7]提出了直接拼接不同视图的原始特征的方法, 以促进多视图之间的信息共享, 并引入了直接交替采样和基于方差的去相关多视图锚点选择方法. 然而, 这种锚点选择策略可能存在一些问题[8]. 首先, 考虑到数据隐私保护问题, 多视图原始特征往往并不适宜直接共享. 其次, 不同视图原始特征的维度分布结构差异较大, 因此拼接的锚点选择方法往往容易受到部分视图特征的影响, 导致选择的锚点不够准确或代表性不足.

多视图多锚点图构造方法并不局限于 1 个视图 1 个锚点图的方式, 也可以采用 1 个视图多个锚点图的策略. 例如, Lao 等人[9]提出在每个视图上进行多次锚点选择, 以生成多个多样化的锚点集合, 然后利用基于锚点的子空间表示在每个视图上获得多个锚点图. 这种 1 个视图多个锚点图的策略有助于提高整体锚点图的多样性, 并减轻了锚点图构建过程中超参数的选择难度. 此外, 为了进一步提高在大规模数据上的锚点选择效率, 研究人员还提出了基于随机选择与 K 均值相结合的混合式锚点选择策略[10], 以及基于平衡 K 均值的层次 K 均值选择策略[11]等.

在多锚点图信息挖掘过程中, 基于多锚点图的多视图聚类方法可选择多种策略来合并这些锚点图. 例如, 对于以视图独立方式构建的多锚点图, Kang 等人[4]提出了采用直接拼接的方式将不同锚点图的信息融合以解决存在的锚点不对齐问题. 而 Wang 等人[5]则进一步提出先对锚点图逐个进行结构化对齐, 然后再叠加合并. 在以视图共享方式联合学习多锚点图时, Zhang 等人[12]提出在联合学习多个锚点图的同时学习它们对应的权重, 并将其用于加权拼接这些锚点图. 对于以多视图拼接形式获得的锚点对齐的多锚点图, 现有方法利用基于秩约束的结构化自适应图学习机制生成一致性锚点图[6-7,12]. 尽管基于多锚

点图的方法为大规模多视图聚类提供了新技术，但在多锚点图的挖掘过程中仍面临诸多挑战. 一方面，锚点不对齐(即不同视图锚点选取的数目不同或选取数目相同但代表性样本点选取不同)问题使得多锚点图与一致性锚点图联合学习变得相当棘手[5]. 另一方面，这些方法往往只聚焦于一致性锚点图的学习[13]，而未能直接关联到最终的聚类结果，这可能影响到下游聚类任务的准确性和一致性.

基于多锚点图的集成聚类方法的关键在于构建高质量且多样性丰富的多个基聚类器(即单独的聚类算法的实例)，并将它们有效地集成融合. 为此，Huang 等人[14]采用多种层面的随机化组合策略，包括不同的特征子集、锚点选择方法、锚点数量、样本与锚点的近邻数量以及视图组合等，以构建具有多样性的锚点图集合. 随后对每个锚点图独立进行图划分以得到对应的基聚类器. 最后，通过一致性集成方法从这些基础聚类器中得到最终的聚类结果. 然而，这些方法通常会将基聚类器的独立生成和集成学习过程分为 2 个阶段[10,15]. 这种做法一方面导致基聚类器在独立生成阶段未能有效地利用多锚点图的潜在信息，另一方面这种划分限制了最终聚类结果对多锚点图潜在信息的充分利用，进而影响最终聚类结果的准确性.

针对基于多锚点图的快速多视图聚类方法和大规模集成聚类方法所面临的关键问题，本文提出了基于双端联合学习的多视图聚类(double-ended joint learning for multi-view clustering, DLMC)算法. 该方法首先利用现有的多视图多锚点构图方法构造多个锚点图. 然后通过同时进行锚点端聚类和样本端聚类，并充分利用来自多个锚点图的信息实现样本端聚类与多个锚点端聚类的集成对齐. 在该方法中，多个锚点端聚类与样本端聚类在一个完整的框架中迭代进行，因而锚点端的聚类会受到样本端聚类以及其他锚点图的联合作用，而样本端聚类则同时受到多个锚点图和它们对应的锚点端聚类的联合作用. 为求解对应的优化问题，本文设计了一种线性迭代式更新算法并对其正确性与收敛性进行了验证.

综上，本文所提的方法优势包括：

1)在解决锚点图不对齐问题的同时避免了依赖一致性图等问题. 此外，该方法也克服了在集成聚类过程中基聚类器的生成阶段与融合阶段未能充分利用多个锚点图信息的局限.

2)将锚点端聚类问题与样本端聚类问题集成到一个统一的优化框架中，使得锚点聚类与样本聚类 2 个内在一致的任务可以在多锚点图的关联下互相提升，最终获得更好的聚类性能.

3)本文方法形式简洁，不需要额外的超参数，适用于广泛的多锚点图大规模聚类任务.

4)对每个锚点图的聚类仅在少量锚点集合上进行，保证了算法的高效性和可扩展性.

5)与表 1 中的 14 种聚类方法在 13 个多视图基准数据集上进行实验比较后，结果显示本文提出的方法在聚类性能和时间消耗方面均优于多个对比方法.

Table 1 A Summary of Abbreviations of Method Used in This Paper

表 1 本文所用方法缩写名称总结

大规模多视图子空间聚类（large-scale multi-view subspace clustering，LMVSC）[4]
可扩展且无需参数的多视图图聚类（scalable and parameter-free multi-view graph clustering，SFMC）[6]
基于张量二部图学习的多视图聚类（tensorized bipartite graph learning for multi-view clustering，TBGL）[7]
具有多个锚图的高效多视图 *K*-means 聚类 (efficient multi-view *K*-means clustering method with multiple anchor graphs，EMKMC）[8]
基于多二部图的可扩展多视图聚类（scalable multi-view clustering via many bipartite graphs，SMCMB)[9]
超可扩展的集成聚类（ultra-scalable ensemble clustering，U-SENC）[10]
面向可扩展多视图聚类的灵活多样锚图融合（flexible and diverse anchor graph fusion for scalable multi-view clustering，FDAGF）[12]
通过集成方法实现快速多视图聚类（fast multi-view clustering via ensembles，FastMICE）[14]
单遍多视图聚类（one-pass multi-view clustering，OPMC）[16]
多视图结构图学习（multi-view structured graph learning，MSGL）[17]
快速且无需参数的多视图子空间聚类（fast parameter-free multi-view subspace clustering，FPMVS）[18]
高效单遍多视图子空间聚类（efficient one-pass multi-view subspace clustering，EOMSC）[19]
统一且离散的二部图学习（unified and discrete bipartite graph learning，UDBGL）[20]
高效单遍的后融合多视图聚类（efficient one-pass late fusion multi-view clustering，OPLFMVC）[21]
通过非负正交分解的快速多视图聚类模型（fast multi-view clustering model via nonnegative and orthogonal factorization，FMCNOF）[22]

1 相关工作

本节将重点讨论快速多视图聚类和大规模集成聚类两大类方法，并在表 2 中从锚点集构建、锚点图构建以及聚类器构建 3 个角度出发列出主要相关方法，其中视图独立、视图共享的区别在于构建时是否挖掘其他视图信息，视图一致指的是构建共识表示.

Table 2 Attributes Comparison of Related Methods

表 2 相关方法属性对比

对比方法	年份	锚点集构建学习			锚点图构建学习			聚类器构建学习		
		视图独立	视图共享	视图一致	视图独立	视图共享	视图一致	视图独立	视图共享	视图一致
LMVSC[4]	2020	√			√		√			
SFMC[6]	2022		√		√		√			
TBGL[7]	2023		√		√		√			
SMCMB[9]	2024	√			√			√		√
U-SENC[10]	2020	√			√			√		√
FDAGF[12]	2023		√			√	√			
FastMICE[14]	2023	√			√				√	√
OPMC[16]	2021									√
MSGL[17]	2022	√					√			
FPMVS[18]	2022			√			√			
EOMSC[19]	2022			√			√			
UDBGL[20]	2023		√			√	√			
OPLFMVC[21]	2021	√			√			√		√
DLMC（本文）		√	√		√	√			√	√

1.1 快速多视图聚类方法

由于传统的多视图聚类方法具有较高的计算复杂度，难以处理大规模的数据集，因此出现了许多面向多视图的快速聚类方法. 这些方法可以分为 2 类：即基于 K 均值的方法和基于锚点图的方法.

基于 K 均值的方法继承了原始 K 均值方法简单高效的特点，并可以证明其等价于基于矩阵分解的方法. 这些方法继承了标准 K 均值的高效性，能够直接从原始数据中提取类别结构，无需任何预处理. 其中代表性方法包括：Cai 等人[23] 通过鲁棒的 L21 范数刻画多视图数据上的一致性聚类误差. 而 Xu 等人[24] 进一步引入降维学习以更好地处理高维多视图数据.Nie 等人[25] 提出了基于样本和特征双边离散划分的多视图 K 均值与三因子矩阵分解方法. 另外，Liu 等人[16] 提出了带有正交约束系数矩阵和一致性离散划分的多视图矩阵分解方法 OPMC. 尽管多视图 K 均值方法在许多情况下表现高效，但仍面临 2 个主要挑战. 首先，多视图数据可能存在质量问题，因为不同视图的数据分布和噪声水平可能不一致，从而影响算法的聚类性能和准确性. 其次，处理高维特征时会导致算法效率下降，并且降低其可扩展性.

基于锚点图的方法利用样本与少量锚点的关系来提高计算效率，避免了传统谱聚类方法使用的全样本图. 这类方法通常包括 4 个主要步骤：锚点的选择与生成、候选锚点图的构造与学习、一致性锚点图的构造与学习以及一致性锚点图的划分与聚类. 在锚点选择与生成过程，一些方法如 LMVSC[4], EMKMC[8], MSGL[17] 采用 K 均值方法在每个视图上独立选择锚点. 而 SFMC[6], TBGL[7], FMCNOF[22] 等则首先将多视图特征进行拼接，然后选择对齐的共享锚点. 另一些方法如 FPMVS[18] 和 EMOSC[19] 则在统一的框架中学习多视图一致的隐式锚点矩阵. 在候选锚点图构造或学习方面，一些方法如 LMVSC[4], SFMC[6], TBGL[7], FMCNOF[22] 通过概率邻近图构造或子空间学习在每个视图上生成锚点图. 另一些方法如 FDAGF[12] 和 UDBGL[20] 则在统一的框架中学习多个锚点图. 在多锚点图融合方面，一些方法如 SFMC[6], TBGL[7], UDBGL[20] 利用一致性学习机制生成一致性锚点图来综合多个视图的信息，而 LMVSC[4] 和 FDAGF[12] 则采用多锚点图拼接的方式，将不同锚点图的信息融合在一起. 另一些方法如 MSGL[17], FPMVS[18], EOMSC[19] 则在统一的框架中直接学习一致性锚点图，以得到最终的一致性聚类结果. 多视图聚类的最终结果可以通过在一致性锚点图上进行划分和聚类来获得. 尽管基于锚点图的多视图聚类方法受到了广泛关注，但仍存在一些待解决的问题. 首先，采用视图拼接的锚点选择方法往往容易受到不同视图特征的影响，可能导致选择的锚点不够准确或代表性不足. 其次，在单个视图上独立进行锚点选择可能会导致不同视图之间的锚点不对齐，给后续的一致性锚点图学习带来挑战. 此外，现有方法通常仅能获得一致性锚点图，而在实际应用中，还需要进一步进行图划分和聚类，这种割裂可能影响聚类性能并增加计算开销.

1.2 大规模集成聚类方法

集成聚类（ensemble clustering, EC）的目的是将多个基聚类组合成更好、更稳健的一致性聚类. 目前的集成聚类方法主要分为 3 类：基于共现矩阵的方法、基于标签对齐的方法和基于图划分的方法.

基于共现矩阵的方法通过捕捉基聚类器中的成对共现关系来构建共现矩阵，然后利用聚类算法在

共现矩阵上进行操作以获取一致的聚类结果. 如 Fred 等人[26]提出了证据累积的聚类方法，通过在共现矩阵上实施层次聚类来实现. Iam-On 等人[27]利用共协矩阵构建基于连接的相似度矩阵，以获得一致性的最终聚类结果；Tao 等人[28]则在共协矩阵上使用谱聚类算法得到一致性的聚类结果；Huang 等人[29]通过基于熵的局部加权策略对共现矩阵进行优化，并提出了局部加权证据累积方法. 然而，这类方法通常具有较高的复杂性，需要更多的计算资源.

基于标签对齐的方法通过调整基聚类器的标签以重新标记数据，以获得一致性的最终聚类结果. 例如，Topchy 等人[30]将集成聚类问题建模为最大似然问题，并采用 EM(expectation maximization)算法进行求解. Bai 等人[31]提出了基于局部可信标签的重标签聚类集成学习算法用以处理复杂数据. Liu 等人[21]则提出了一种后融合方法，将每个视图的谱嵌入进行旋转对齐以获得更一致的聚类结果. 然而，由于不同基本聚类器生成的簇可能存在语义上的差异或不一致，这使得在进行集成聚类时难以完全实现簇级语义的对齐，从而影响最终聚类结果的质量.

基于图划分的方法将多个基聚类器表示为一种图结构，通过对该图进行图分割来获得一致的聚类结果. 例如，Strehl 等人[32]利用基聚类器的结果构建超图，将集成聚类问题转化为超图分割问题. Zhou 等人[33]将集成聚类任务转化为多图学习任务，并提出了基于多图学习的 3 层鲁棒集成聚类方法. Lao 等人[9]构建了无向图来描述样本之间的关系，并提出了用于集成聚类的大规模谱聚类方法. 此外，为进一步提高大规模多视图数据上集成聚类的性能，Huang 等人[14]提出使用锚点图提高基聚类器的生成效率，并采用多种策略提高基聚类器集合的多样性，最终的集成聚类结果可以通过基于基聚类器构建的锚点图进行分割获得. 然而，这些方法往往将基聚类器的独立生成和集成学习划分为 2 个独立的阶段，这种做法存在明显的局限性. 首先，独立生成阶段未能充分利用多锚点图所蕴含的丰富信息，导致基聚类器的性能受限. 其次，由于 2 个阶段的划分，最终聚类结果可能无法充分融合多锚点图的潜在信息，从而影响聚类的准确性. 值得注意的是，为了克服现有技术的不足，本文方法直接利用现有多锚点图构建方法获得多个锚点图，然后对锚点图开展大规模集成对齐聚类.

2 本文方法

本节首先介绍了后续将使用的符号，并在图 1 中展示了本文方法的整体流程示意图. 接着简要介绍了常见的多锚点图构造方法，并着重介绍了本文提出的基于锚点端与样本端联合学习的聚类方法，以及相应的迭代式快速求解方法. 最后，对该方法的收敛性和复杂性进行了分析.

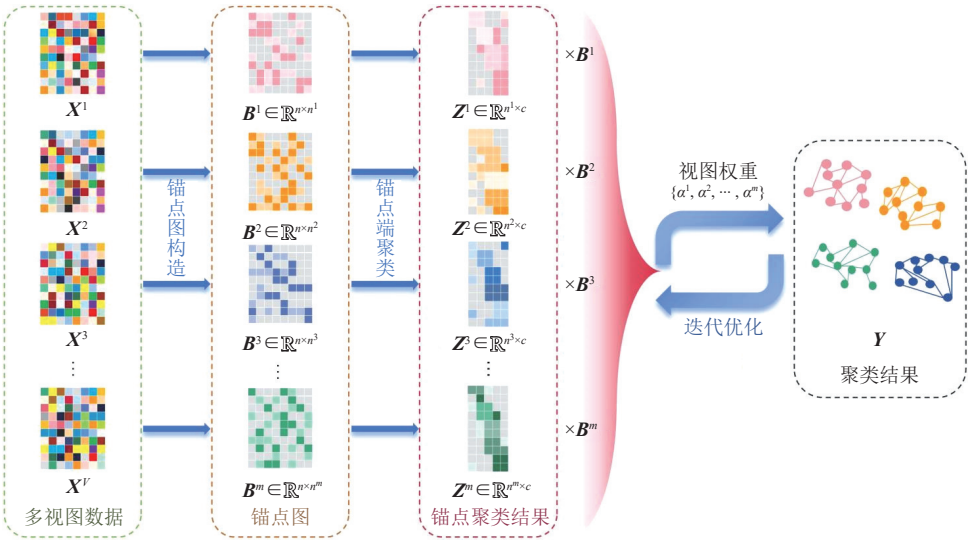


Fig. 1 Schematic diagram of multi-view clustering method based on double-ended joint learning

图 1 基于双端联合学习的多视图聚类方法示意图

2.1 符号定义

我们使用符号$\{\boldsymbol{X}^v\}_{v=1}^V$表示$V$个视图对应的多视图数据集合，其中$\boldsymbol{X}^v \in \mathbb{R}^{n\times d^v}$代表由$n$个样本和$d^v$个特征组成的第$v$个视图. 用$\{\boldsymbol{B}^r\}_{r=1}^m$表示构造的$m$个锚点图集合，其中$\boldsymbol{B}^r \in \mathbb{R}^{n\times n^r}$代表$n$个样本与$n^r$个锚点构造的锚点图. 用$\{\boldsymbol{Z}^r\}_{r=1}^m$表示锚点的聚类结果，其中$\boldsymbol{Z}^r \in \mathbb{R}^{n^r\times c}$

代表第r个锚点图对应的n^r个锚点在c维类簇上的聚类结果，其中c是最终类簇个数. 此外，我们用$\boldsymbol{x}_i^v$表示第v个视图中的第i个样本，用x_{ij}^v表示第v个视图中第i个样本的第j个特征. $\boldsymbol{Y}\in\mathbb{R}^{n\times c}$表示多视图样本端共识聚类结果，$\boldsymbol{1}$和$\boldsymbol{I}$则分别表示单位向量和单位矩阵.

2.2 多锚点图构造

本文的重点在于如何在多锚点图上进行高性能聚类. 为了使本文内容更加完整，本节简要介绍了一些常见的多锚点图构造方法. 如 Cai 等人[2]所述，一种常见的锚点选择方法是在某个视图上运行 K 均值算法将数据聚集到指定个数的类簇，并选择与类簇中心最近的点作为锚点集合. 鉴于多样化机制构建的多个锚点图有助于提高后续聚类性能[14]，因此在实际应用中可以从某个视图部分特征上、不同锚点数量下和不同的 K 均值初始化条件下等多个角度选择多组多样性锚点集合.

获得多个锚点集合后，我们同样可以使用多种方法构建样本与锚点的关系，比如概率邻近图[4]或者子空间学习[5]等. 本文考虑到概率邻近图可以刻画样本与锚点集合间的局部邻近信息以保持数据潜在的流形结构，并且其仅包含易于设置的邻近个数 k，因此被选为本文的锚点图构造方法. 在计算第r个锚点图$\boldsymbol{B}^r$时，其中每个元素b_{ij}^r表示第i个样本和第j个锚点之间的相似性，其计算方法为：

$$b_{i,j}^r=\begin{cases}\dfrac{h(\boldsymbol{x}_i^r,\boldsymbol{x}_{k+1}^r)-h(\boldsymbol{x}_i^r,\boldsymbol{x}_j^r)}{\sum\limits_{j'=1}^{k}\left(h(\boldsymbol{x}_i^r,\boldsymbol{x}_{k+1}^r)-h(\boldsymbol{x}_i^r,\boldsymbol{x}_{j'}^r)\right)}, & j\leqslant k,\\ 0, & j>k,\end{cases}\tag{1}$$

其中$h=(\cdot,\cdot)$是计算 2 个样本之间的欧氏距离，k为其邻域大小.

2.3 基于锚点图的集成对齐聚类

基于锚点图的数据表示形式在多视图聚类任务中具有明显优势. 与原始特征相比，锚点图能更清晰地展现数据的关键结构信息，有助于深入理解数据的内在性质. 每个锚点图都由不同的锚点集合构成，每个集合都包含一定数量的代表性点. 从锚点图的角度来看，锚点集合与样本集合有相似之处，甚至在某些情况下，锚点集合可以被视为样本集合的一个子集. 这种相似性使得我们可以直观地将锚点集合划分为与样本相同的c个类簇. 即使锚点并非真实样本，但它们在数据空间中仍代表着重要的结构信息，例如可能代表聚类中心. 通过对锚点集合进行聚类，我们能更好地理解数据的聚类模式和分布特征，进而提高聚类的准确性和稳定性. 因此，本文提出对每个锚点图中的锚点集合进行聚类，并学习锚点端的谱嵌入. 以第r个锚点图$\boldsymbol{B}^r\in\mathbb{R}^{n\times n^r}$为例，其锚点集合对应的聚类嵌入可通过以下优化问题获得：

$$\begin{aligned}&\max_{\boldsymbol{Z}^r}\ \mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r),\\&\text{s.t. }\ \boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r=\boldsymbol{I}_c,\end{aligned}\tag{2}$$

其中$\boldsymbol{Z}^r\in\mathbb{R}^{n^r\times c}$代表第$r$个锚点图对应的$n^r$个锚点在$c$维类簇空间上的表示结果. 通过式(2)构建的$\boldsymbol{Z}^r$可以被视为锚点在$c$维类簇空间的嵌入，其中每一列代表一个类簇. 同时，对于任意一个样本点，它都可以用$\boldsymbol{B}^r\boldsymbol{Z}^r$的线性组合来表示，这意味着它在$c$维类簇空间的位置可以由$\boldsymbol{B}^r\boldsymbol{Z}^r$所定义的子空间中的线性组合来表示. 这种分析建立了样本在任一锚点图下的类簇空间与锚点集合获得的类簇空间之间的关系. 因此，样本端的聚类结果$\boldsymbol{Y}$与第r个锚点图$\boldsymbol{B}^r$和$\boldsymbol{Z}^r$间的一致性对齐就可以通过以下公式刻画：

$$\begin{aligned}&\max_{\boldsymbol{Y}}\ \mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r),\\&\text{s.t. }\ \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}_c.\end{aligned}\tag{3}$$

借助式(3)，本文方法仅通过对锚点进行聚类就可获得样本在该锚点图下对应的类簇表示. 这种分析不仅建立了样本与锚点的内在联系以提升样本聚类性能，而且保持了非常低的计算复杂性(仅与锚点大小有关). 假设每个锚点图$\{\boldsymbol{B}^r\}_{r=1}^m$都通过上述方式获得了对应的$\{\boldsymbol{Z}^r\}_{r=1}^m$，考虑到样本与所有的锚点集合都对应于相同的类簇空间，那么样本端在多锚点图下的聚类结果$\boldsymbol{Y}$与所有锚点图$\{\boldsymbol{B}^r\}_{r=1}^m$及其对应的锚点集合嵌入$\{\boldsymbol{Z}^r\}_{r=1}^m$间的一致性集成对齐可表示为

$$\begin{aligned}&\max_{\boldsymbol{Y}}\ \sum_{r=1}^{m}\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r),\\&\text{s.t. }\ \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}_c.\end{aligned}\tag{4}$$

基于上述分析，本文提出在一个联合学习的框架下对所有锚点图上的锚点与样本同时进行聚类，进而得到基于双端联合学习的多视图聚类方法，具体优化问题如下所示：

$$\begin{aligned}&\max_{\boldsymbol{Z}^r,\boldsymbol{Y},\alpha^r}\ \sum_{r=1}^{m}\alpha^r\left[\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)+\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)\right]\\&\qquad\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}_c,\ \ \boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r=\boldsymbol{I}_c,\\&\text{s.t.}\quad 1\geqslant\alpha^r\geqslant 0,\ \ \sum_{r=1}^{m}\alpha^{2r}=1,\forall r,\end{aligned}\tag{5}$$

其中$\{\alpha^r\}_{r=1}^m$表示锚点图在集成对齐时对应的不同权重. 与现有的锚点图学习方法不同，该方法不需要直接学习一致性锚点图，因而避免了锚点不对齐带来的困难. 同时也规避了图学习与图划分分步处理对聚类性能的不利影响. 与现有的集成聚类方法不同，

该方法在一个完整的优化框架中同时利用多个锚点图进行锚点端聚类和样本端聚类，在保证计算效率的同时，还解决了基聚类器在独立生成阶段无法利用其他锚点图以及在集成阶段无法充分利用多锚点图的问题. 因此，本文方法能够填补现有多视图聚类方法与集成聚类方法之间的差距并提高多视图数据上的聚类性能.

2.4 模型优化求解

式(5)所述的优化问题是包含3组变量，即样本端聚类结果 $\boldsymbol{Y}$、每个锚点图对应的锚点端聚类结果 $\{\boldsymbol{Z}^r\}_{r=1}^m$ 和每个锚点图对应的权重 $\{\alpha^r\}_{r=1}^m$ 的非凸优化问题. 为此，我们设计了分块坐标轮换法进行逐个迭代求解，算法优化过程如下所示.

2.4.1 优化锚点端聚类 $\boldsymbol{Z}^r$

当变量 $\boldsymbol{Y}$ 和 $\{\alpha^r\}_{r=1}^m$ 固定时，关于 $\boldsymbol{Z}^r$ 的子优化问题可以简写为

$$\begin{aligned}&\min_{\boldsymbol{Z}^r}\ \mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{W}^r\boldsymbol{Z}^r)+\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{P}^r),\\&\text{s.t.}\ \ \boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r=\boldsymbol{I}_c,\end{aligned}\tag{6}$$

其中辅助变量 $\boldsymbol{W}^r=-\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r$，$\boldsymbol{P}^r=-\frac{1}{2}\boldsymbol{B}^r\boldsymbol{Y}$. 通过引入拉格朗日乘子 $\boldsymbol{\Lambda}^r$，得到如下拉格朗日函数：

$$L^r=\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{W}^r\boldsymbol{Z}^r)+2\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{P}^r)-\mathrm{tr}(\boldsymbol{\Lambda}^r(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r-\boldsymbol{I})).$$

对 L^r 函数中变量 $\boldsymbol{Z}^r$ 求偏导并令其等于0，可得

$$\frac{\partial L^r}{\partial \boldsymbol{Z}^r}=2(\boldsymbol{W}^r\boldsymbol{Z}^r+\boldsymbol{P}^r-\boldsymbol{Z}^r\boldsymbol{\Lambda}^r)=0.\tag{7}$$

在式(7)两边同时乘以 $\boldsymbol{Z}^{r\mathrm{T}}$，结合 $\boldsymbol{Z}^r$ 的正交约束条件可以得到 $\boldsymbol{\Lambda}^r=\boldsymbol{Z}^{r\mathrm{T}}(\boldsymbol{W}^r\boldsymbol{Z}^r+\boldsymbol{P}^r)$，且 $\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r$ 对称的，因此 $\boldsymbol{\Lambda}^r$ 也满足对称性质，故 $\boldsymbol{\Lambda}^r$ 可以被重新表示为 $\boldsymbol{\Lambda}^r=(\boldsymbol{W}^r\boldsymbol{Z}^r+\boldsymbol{P}^r)^{\mathrm{T}}\boldsymbol{Z}^r$，然后将其带回式(7)得到

$$\frac{\partial L^r}{\partial \boldsymbol{Z}^r}=2(\boldsymbol{W}^r\boldsymbol{Z}^r\boldsymbol{Z}^{r\mathrm{T}}+\boldsymbol{P}^r\boldsymbol{Z}^{r\mathrm{T}}-\boldsymbol{Z}^r(\boldsymbol{W}^r\boldsymbol{Z}^r+\boldsymbol{P}^r)^{\mathrm{T}})\boldsymbol{Z}^r.$$

令 $\boldsymbol{R}=2\left(\boldsymbol{W}^r\boldsymbol{Z}^r\boldsymbol{Z}^{r\mathrm{T}}+\boldsymbol{P}^r\boldsymbol{Z}^{r\mathrm{T}}-\boldsymbol{Z}^r(\boldsymbol{W}^r\boldsymbol{Z}^r+\boldsymbol{P}^r)^{\mathrm{T}}\right)$，容易证明 $\boldsymbol{R}$ 是一个斜对称矩阵，且满足 $\boldsymbol{R}^{\mathrm{T}}=-\boldsymbol{R}$. 根据文献[32]中定理3可知式(6)中带正交约束的二次优化问题对应的更新公式为

$$\boldsymbol{Z}^{r(t+1)}=\left(\boldsymbol{I}+\frac{\eta}{2}\boldsymbol{R}^r\right)^{-1}\left(\boldsymbol{I}+\frac{\eta}{2}\boldsymbol{R}^r\right)\boldsymbol{Z}^{r(t)},\tag{8}$$

其中 η 为梯度下降法的步长，并且可以验证 $\boldsymbol{Z}^{r(t+1)}$ 满足正交约束条件.

2.4.2 优化样本端聚类 $\boldsymbol{Y}$

当变量 $\{\boldsymbol{Z}^r\}_{r=1}^m$ 和 $\{\alpha^r\}_{r=1}^m$ 固定时，关于 $\boldsymbol{Y}$ 的子问题可以简化为

$$\begin{aligned}&\max_{\boldsymbol{Y}}\ \mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{H}),\\&\text{s.t.}\ \ \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I},\end{aligned}\tag{9}$$

其中 $\boldsymbol{H}=\sum_{r=1}^m\boldsymbol{B}^r\boldsymbol{Z}^r$. $\boldsymbol{Y}$ 的最优解为

$$\boldsymbol{Y}=\boldsymbol{G}\boldsymbol{V}^{\mathrm{T}},\tag{10}$$

其中 $\boldsymbol{G}$ 和 $\boldsymbol{V}$ 分别为对 $\boldsymbol{H}$ 进行SVD分解得到的左右奇异矩阵.

2.4.3 优化锚点图权重 $\{\alpha^r\}_{r=1}^m$

当变量 $\{\boldsymbol{Z}^r\}_{r=1}^m$ 和 $\boldsymbol{Y}$ 固定时，关于 $\{\alpha^r\}_{r=1}^m$ 的子优化问题可以简化为

$$\max_{\alpha^r}\sum_{r=1}^m\alpha^r h^r,\ \text{s.t.}\ 1\geqslant\alpha^r\geqslant 0,\ \sum_{r=1}^m\alpha^{2r}=1,\tag{11}$$

其中 $h^r=\mathrm{tr}\left(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r\right)+\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)$. 可得到 α^r 的闭式解，即 $\alpha^r=h^r\Big/\sqrt{\sum_{r=1}^m h^{2r}}$. 本文目标函数的完整优化过程如算法1所示.

算法 1. 基于双端联合学习的多视图聚类算法.

输入：多锚点图 $\{\boldsymbol{B}^r\}_{r=1}^m$ 以及类簇数 c；

输出：c 个相互独立的簇.

① 初始化 $\{\boldsymbol{Z}^r\}_{r=1}^m$ 和 $\boldsymbol{Y}$；

② 初始权重赋值为 $\alpha^r=\frac{1}{\sqrt{m}}$；

③ while 目标函数不收敛 do

④ 通过求解式(8)更新 $\{\boldsymbol{Z}^r\}_{r=1}^m$；

⑤ 通过求解式(10)更新 $\boldsymbol{Y}$；

⑥ 更新 $\alpha^r=h^r\Big/\sqrt{\sum_{r=1}^m h^{r2}}$；

⑦ end while

⑧ until 目标函数收敛；

⑨ 对 $\boldsymbol{Y}$ 进行归一化处理后运行 K 均值得到最终聚类结果.

2.5 收敛性分析

引理 1. 针对式(5)所设计的求解算法1可以收敛到局部最优解.

证明. 我们首先证明式(5)中的损失函数是有上界的. 为此，我们引入关于锚点图的常见假设，即 $\boldsymbol{B}^r\geqslant\boldsymbol{0}$，$\boldsymbol{B}^r\boldsymbol{1}=\boldsymbol{1}$. 事实上，通过概率邻近图和子空间学习这2种常见方式获得的锚点图都满足这一性质. 令 $\boldsymbol{A}^r=\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r$，可知 $\boldsymbol{0}\leqslant\boldsymbol{A}^r\leqslant\boldsymbol{1}$，且 $\boldsymbol{A}^r$ 是对称半正定的，即可以将其表示为 $\boldsymbol{A}^r=\boldsymbol{U}^r\boldsymbol{\Gamma}^r\boldsymbol{U}^{r\mathrm{T}}$，其中 $\boldsymbol{U}^r$ 是单位正交矩阵，$\boldsymbol{\Gamma}^r$ 是特征值组成的对角矩阵，由此可得

$$\begin{aligned}&\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{B}^{r\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)=\mathrm{tr}(\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{A}^r\boldsymbol{Z}^r)=\mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{U}^r\boldsymbol{\Gamma}^r\boldsymbol{U}^{r\mathrm{T}}\boldsymbol{Z}^r)=\\&\sum_{i=1}^c\lambda_i\|(\boldsymbol{U}^r\boldsymbol{Z}^r)_i\|^2=\sum_{i=1}^c\lambda_i(\boldsymbol{U}^r\boldsymbol{Z}^r)_i^2=\\&\sum_{i=1}^c\lambda_i\leqslant c\lambda_{\max}\leqslant c\,\|\boldsymbol{A}^r\|_{\mathrm{F}}\leqslant cn^r.\end{aligned}\tag{12}$$

此外，根据Cauchy-Schwarz不等式可得

$$\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)\leqslant\|\boldsymbol{Y}^{\mathrm{T}}\|_{\mathrm{F}}\|\boldsymbol{B}^r\boldsymbol{Z}^r\|_{\mathrm{F}}. \tag{13}$$

由于$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}$, 则$\|\boldsymbol{Y}^{\mathrm{T}}\|_{\mathrm{F}}=\|\boldsymbol{Y}\|_{\mathrm{F}}=1$和$\boldsymbol{Z}^{r\mathrm{T}}\boldsymbol{Z}^r=\boldsymbol{I}$. 因此, 可以得到

$$\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^r\boldsymbol{Z}^r)\leqslant\|\boldsymbol{B}^r\|_{\mathrm{F}}. \tag{14}$$

由于$\boldsymbol{B}^r$的每一行元素之和都为1, 可以得到

$$\|\boldsymbol{B}^r\|_{\mathrm{F}}\leqslant nn^r. \tag{15}$$

结合式(12)和式(15)可得式(5)的上界为

$$\sum_{r=1}^{m}a^r(c+n)n^r\leqslant(c+n)n^r_{\max}. \tag{16}$$

我们进一步说明, 在算法1的迭代过程中式(5)中的损失函数是单调递增的. 每次迭代, 算法1分别优化3个子问题. 这些子问题在优化一个变量并保持其他变量固定时都是严格凸的. 因此, 在每次迭代中, 当优化一个变量时, 算法的目标函数在固定其他变量的情况下是单调增加的. 综上所述, 由于式(5)的目标函数在每次迭代中单调递增且受到上界的限制, 我们可以验证算法1是收敛的. 证毕.

2.6 复杂性分析

本节从时间复杂度和空间复杂度2个方面对本文所提出算法的进行总结. 该算法的复杂度主要包括3个部分. 在更新$\boldsymbol{Z}^r$时, 需要消耗$O(t(nkc+nc^2+c^3)$的时间和$O(nc+nk)$的空间, 其中t表示迭代次数; 更新$\boldsymbol{Y}$时, 需要消耗$O\left(nc^2\right)$的时间和$O(nc)$的空间; 更新α^r时需要消耗$O(1)$的时间和$O(V)$的空间. 在聚类任务中, 通常$n\gg c$以及$n\gg k$, 因此, 本文算法的时间复杂度以及空间复杂度近似为$O(n)$, 即保持线性复杂度.

3 实验分析

本节将介绍实验中所采用的13个数据集和14种基线方法, 以及它们在聚类结果、聚类t-SNE可视化结果以及运行时间方面的表现. 我们还将探讨不同锚点图配置方式对实验结果的影响, 并通过对实验数据的分析来深入评估这些方法的性能.

3.1 数据集

本文采用了13个广泛使用的多视图数据集[14,23]作为聚类方法的基准评估. 这些数据集包含了多个图像数据集、2个文本数据集、2个网页数据集和1个基因序列数据集. 这些数据集在样本规模、类别数量、维度和视图数量等方面具有不同的特点, 能够全面评估多视图聚类方法在各种场景下的性能和鲁棒性.

在表3详细列出了数据集的关键属性, 包括样本大小、特征的维度、类别数量和视图数量. 表4则通过归一化熵(normalized entropy, NE)[34]、聚类大小的标准差(standard deviation of cluster sizes, SDCS)、最小聚类与期望聚类大小的比率RME(rand measure expected)以及最大聚类与最小聚类的比率(Max/Min)这4个度量指标来评估数据集的平衡性. 当NE, RME, Max/Min值为1而SDCS值为0时说明该数据集为平衡数据集, 反之则为不平衡数据集, 且当NE值和RME值越小而SDCS和Max/Min值越大时说明数据集越不平衡.

Table 3 Dataset Description

表3 数据集描述

编号	数据集	样本个数	视图数量	类别数量	维特征度
D1	NEWSGROUPS	500	3	5	2 000, 2 000, 2 000
D2	BBCSPORT	544	2	5	3 183, 3 203
D3	WEBKB	1 051	2	2	1 840, 3 000
D4	REUTERS	1 200	5	6	2 000, 2 000, 2 000, 2 000, 2 000
D5	100LEAVES	1 600	3	100	64, 64, 64
D6	BDGP	2 500	3	5	1 000, 500, 250
D7	CCV	6 773	3	20	20, 20, 20
D8	MNIST	10 000	3	10	30, 9, 30
D9	ALOI100	10 800	4	100	77, 13, 64, 125
D10	AWA	30 475	6	50	2 688, 2 000, 252, 2 000, 2 000, 2 000
D11	YTF10	38 654	4	10	944, 576, 512, 640
D12	YOUTUBEFACE	101 499	5	31	64, 512, 64, 647, 838
D13	ALOI1000	110 250	4	1 000	125, 77, 113, 64

Table 4 Corresponds to Four Balance Index Values for Thirteen Datasets

表4 对应13个数据集的4个平衡性指标值

编号	度量指标			
	NE	SDCS	RME	Max/Min
D1	1.00	0.00	1.00	1.00
D2	0.94	54.36	0.56	3.16
D3	0.76	417.90	0.44	3.57
D4	1.00	0.00	1.00	1.00
D5	1.00	0.00	1.00	1.00
D6	1.00	0.00	1.00	1.00
D7	0.98	134.24	0.31	6.73
D8	1.00	0.00	1.00	1.00
D9	1.00	0.00	1.00	1.00
D10	0.98	253.59	0.15	12.70
D11	0.98	1 237.17	0.70	2.25
D12	0.89	4 432.03	0.46	18.10
D13	1.00	1.30	0.98	1.03

3.2 对比方法

本文选择了 14 种具有代表性的多视图聚类和集成聚类方法. OPMC[16] 是基于多视图矩阵分解的方法并去除非负性约束. U-SENC[10] 是基于锚点图生成的基聚类器，并在此基础进行大规模集成聚类. OPLFMVC[21] 为每个视图生成正交嵌入，然后进行后融合以集成聚类. LMVSC[4] 为每个视图学习锚点图，拼接后进行图划分. EOMSC[19] 联合学习一组隐锚点与一致性锚点图. FPMVS[18] 将锚点选择和子空间图构造集成到统一的优化问题. EMKMC[8] 为每个视图构造锚点图，使用改进的 K 均值集成聚类. MSGL[17] 选择多个锚点集，直接学习一致性锚点图. SFMC[6] 选择多视图一致性锚点集，构建多锚点图并学习共识锚点图. FastMICE[14] 构建构建多视图下多样化锚点图以生成基聚类器并融合. FDAGF[12] 学习多锚点图及其权重，加权拼接多锚点图并进行划分聚类. TBGL[7] 选择视图一致性锚点集，构建多锚点图并利用高阶信息学习共识锚点图. UDBGL[20] 选择多视图一致性锚点，联合学习多锚点图和一致性锚点图. SMCMB[9] 基于多样化锚点图生成基聚类器并进行集成聚类.

3.3 主要实验设置

为保证实验的公平性，本文实验遵循各自论文中的参数建议来调整参数，对于公开源代码的方法采用源代码进行实验. 对于只给定单个数据集参数范围的方法，在使用原文中未使用的数据集进行实验时，参考论文中同等规模数据集的参数范围并将其扩大进行网格状搜索. 此外，对于 OPLFMVC 方法我们采用与本文相同的锚点图构造方法，将其作为该算法的输入.

在本文方法中采用了 1 个视图 1 个锚点图的策略，在每个视图上将锚点数量作为 K 均值的簇数，并选择距离聚类中心最近的点作为锚点. 通过小范围的网格搜索确定最优的锚点个数. 具体而言，在不同样本数的情况，我们设置了不同的锚点搜索范围. 例如，当样本数超过 50 000 时，锚点为 1 000, 2 000, 3 000, 4 000, 步长为 500；当样本数在 10 000~50 000 时，锚点为 500, 800, 1 000, 1 200, 1 500, 2 000；当样本数在 5 000~10 000 时，锚点为 100, 200, 500, 800, 1 000；当样本数少于 5 000 时，锚点为 100, 200, 300, 400, 500, 步长为 100. 确定锚点数目之后我们利用式(1)中的概率邻近图来描述样本与锚点之间的关系，并构建相应的锚点图，其中近邻值被设置为 20.

本文使用聚类准确性(accuracy, ACC)、归一化互信息(normalized mutual information, NMI)、调整兰德指数(adjusted rand index, ARI)、纯度(Purity)和运行时间作为性能评估指标. 所有实验均在配备 i7-12700CPU 和 64 GB 内存的计算机上运行.

3.4 聚类性能比较与分析

在表 5~8 中给出了 DLMC 与其他 14 个基线方法在上述 13 个数据集上的实验结果，其中包括 ACC,

Table 5 DLMC and All Comparative Methods on Thirteen Benchmark Datasets for ACC Values

表 5 DLMC 与所有对比方法在 13 个基准数据集上的 ACC 值 %

对比方法	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
U-SENC	93.66	93.57	83.54	23.93	80.93	47.00	17.74	69.84	68.22	7.85	78.38	12.76	39.64
LMVSC	82.40	94.30	88.20	43.42	64.38	49.28	16.86	68.61	56.61	6.73	69.12	12.77	42.19
OPMC	83.74	76.54	92.40	43.67	54.70	45.23	15.82	56.38	49.75	9.34	69.23	15.21	23.62
FPMVS	72.30	40.63	94.96	44.33	33.01	42.59	24.32	81.64	31.53	8.88	71.49	23.35	-
OPLFMVC	66.40	80.88	94.10	30.42	80.25	52.04	17.70	72.57	72.27	8.51	73.41	19.98	49.75
EOMSC	68.60	38.42	89.44	45.92	39.88	42.08	22.52	64.91	31.63	8.74	81.19	26.66	-
MSGL	51.60	77.57	77.35	31.83	14.13	38.64	14.81	44.49	13.82	6.96	68.08	26.63	-
SFMC	76.20	70.22	78.21	23.92	70.25	23.28	14.53	66.86	67.20	3.90	68.47	N/A	N/A
EMKMC	87.40	91.18	93.34	41.33	45.56	49.88	15.86	64.95	45.54	6.00	74.23	21.60	28.51
TBGL	32.20	74.26	80.59	24.92	63.06	20.08	10.75	75.48	65.44	N/A	N/A	N/A	N/A
FDAGF	93.60	86.76	94.58	52.67	60.19	54.08	19.99	67.24	53.51	7.44	82.35	15.12	42.11
FastMICE	70.96	94.05	94.90	28.68	83.53	43.45	20.10	78.71	75.12	8.68	77.28	20.79	47.74
UDBGL	51.60	49.63	91.82	31.17	63.50	39.44	20.74	70.99	58.02	8.36	60.35	24.85	-
SMCMB	89.20	92.87	96.67	51.50	83.00	49.68	19.92	75.41	75.57	9.01	81.86	23.15	-
DLMC（本文）	**94.40**	**95.59**	**98.19**	**57.83**	**89.69**	**57.68**	**24.46**	**82.28**	**77.35**	**9.89**	**82.83**	**30.30**	**58.42**

注：黑体数值表示最佳结果；“N/A”表示由于内存不足导致算法在该数据集上无法运行；“-”表示由于运行时间过长（超 48 h 未完成）而终止了算法的运行.

Table 6 DLMC and All Comparative Methods on Thirteen Benchmark Datasets for NMI Values

表 6 DLMC 与所有对比方法在 13 个基准数据集上的 NMI 值 %

对比方法	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
U-SENC	82.49	81.46	18.61	9.71	90.53	27.45	13.71	66.40	77.98	9.14	82.81	11.45	67.14
LMVSC	60.54	83.92	33.49	28.07	83.25	28.01	11.87	63.02	74.95	7.35	74.43	5.27	73.34
OPMC	73.57	66.87	62.24	25.85	76.65	25.89	12.88	52.70	68.84	**11.77**	74.13	13.05	58.78
FPMVS	57.06	10.37	66.69	20.64	61.90	19.73	18.59	67.78	55.51	9.11	76.86	21.87	-
OPLFMVC	53.71	66.36	59.47	15.02	90.86	30.47	13.38	64.48	83.78	7.72	76.21	17.62	77.81
EOMSC	60.42	17.03	36.93	24.26	66.24	14.45	15.36	56.16	55.64	8.94	81.81	9.64	-
MSGL	37.57	61.26	16.33	15.18	41.34	11.48	8.67	41.80	36.70	7.64	73.44	0.07	-
SFMC	70.79	60.66	0.28	11.28	82.70	4.26	8.13	61.89	69.21	0.16	76.35	N/A	N/A
EMKMC	69.13	78.13	58.50	18.13	70.29	27.45	11.06	59.43	64.74	5.03	75.88	18.82	63.50
TBGL	9.35	51.82	7.37	11.30	70.50	0.16	2.05	60.93	68.80	N/A	N/A	N/A	N/A
FDAGF	82.71	77.22	60.11	34.57	81.13	35.35	16.82	60.02	70.86	6.91	83.36	12.43	71.79
FastMICE	52.82	85.43	63.36	15.31	93.45	22.20	15.44	68.44	83.36	10.34	81.58	17.37	73.25
UDBGL	26.28	19.40	52.38	10.29	78.80	16.85	16.08	61.21	68.58	8.92	67.40	16.47	-
SMCMB	74.07	84.18	73.15	29.51	92.13	29.70	15.53	63.67	83.20	10.34	85.46	19.56	-
DLMC（本文）	**83.49**	**86.57**	**82.81**	**35.35**	**95.78**	**36.48**	**18.94**	**70.13**	**88.54**	11.51	**85.59**	**25.74**	**80.35**

注：黑体数值表示最佳结果；“N/A”表示由于内存不足导致算法在该数据集上无法运行；“-”表示由于运行时间过长（超 48 h 未完成）而终止了算法的运行.

Table 7 DLMC and All Comparative Methods on Thirteen Benchmark Datasets for ARI Values

表 7 DLMC 与所有对比方法在 13 个基准数据集上的 ARI 值 %

对比方法	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
U-SENC	84.74	83.58	29.10	1.96	73.95	21.63	5.41	57.35	55.87	1.78	72.31	1.82	26.65
LMVSC	59.83	84.62	47.85	17.97	54.33	24.74	5.05	52.35	38.97	1.11	60.92	0.96	24.93
OPMC	72.41	64.46	74.47	17.71	44.48	20.61	4.20	41.10	44.95	2.49	60.83	2.22	11.81
FPMVS	55.58	10.28	79.40	16.91	19.96	15.91	8.52	64.84	16.07	**2.77**	65.04	5.83	-
OPLFMVC	46.68	68.99	75.57	4.22	74.68	29.59	4.90	56.34	65.05	2.01	66.69	3.41	36.01
EOMSC	56.34	10.04	56.49	18.41	20.22	9.79	7.09	49.15	16.09	2.53	71.37	2.36	-
MSGL	28.16	56.12	27.45	9.12	2.40	8.64	2.91	27.76	7.86	1.27	58.42	0.06	-
SFMC	66.43	51.19	0.49	2.18	40.33	0.62	3.22	45.37	10.44	0.00	58.22	N/A	N/A
EMKMC	70.51	78.77	73.34	13.63	31.55	24.43	3.92	49.33	33.82	0.85	66.49	4.67	17.21
TBGL	9.32	45.72	12.40	1.90	10.57	0.00	−0.05	56.81	10.40	N/A	N/A	N/A	N/A
FDAGF	84.62	81.13	76.74	23.13	49.91	29.41	6.70	51.36	30.16	1.09	76.94	1.99	23.24
FastMICE	42.42	86.32	78.48	5.82	79.96	18.40	6.76	63.11	66.30	2.18	71.00	4.01	33.71
UDBGL	23.34	24.25	67.93	7.15	41.14	12.55	6.23	53.06	31.90	1.83	51.92	3.35	-
SMCMB	74.57	87.11	85.82	22.14	77.77	23.31	6.32	56.84	65.40	2.44	77.96	4.94	-
DLMC（本文）	**86.51**	**89.40**	**92.03**	**27.72**	**87.03**	**33.53**	**8.70**	**66.52**	**68.66**	2.69	**79.60**	**7.29**	**36.26**

注：黑体数值表示最佳结果；“N/A”表示由于内存不足导致算法在该数据集上无法运行；“-”表示由于运行时间过长（超 48 h 未完成）而终止了算法的运行.

NMI, ARI, Purity 在内的 4 个评估指标值. 观察表 5~8 以及图 2 的结果可以得到 4 个结论：

1)与传统的基于锚点图的多视图聚类方法相比，本文提出的双端联合聚类算法(DLMC)在所采用的 13 个数据集上性能都有一定的提升，并且在 12 个数据集上 4 个指标值都达到最佳. 该结果表明，DLMC 是一种有效的多锚点图聚类算法.

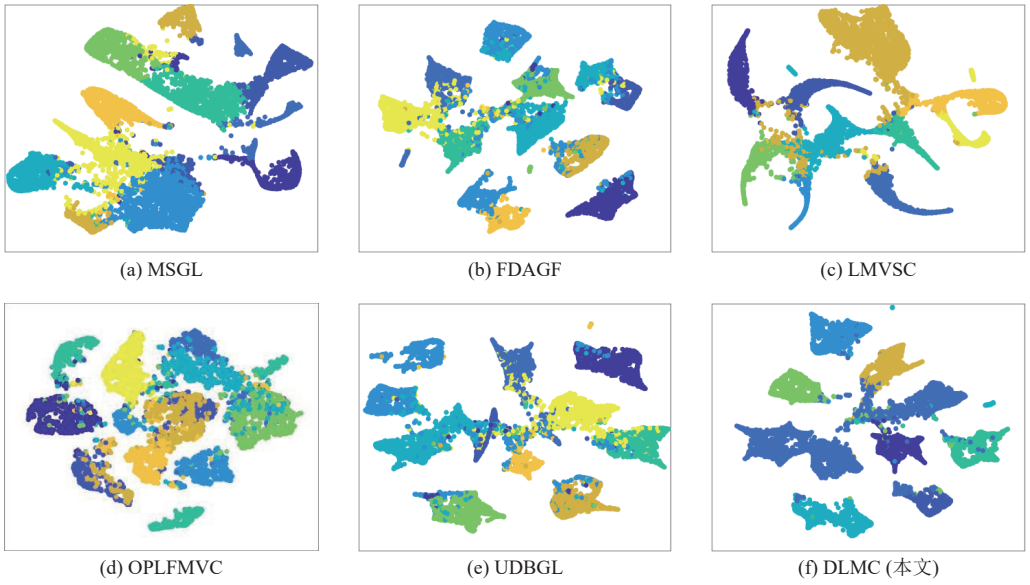
2)在与基于多锚点图的大规模多视图聚类方法如 SFMC, MSGL, UDBGL, TBGL, EOMSC 的比较中，DLMC 在除了 D10 之外的所有数据集上均展现出了更优的性能表现. 深究这一现象，可以发现这些算法普遍存在对一致性锚点图的过度依赖问题. 特别是

Table 8 DLMC and All Comparative Methods on Thirteen Benchmark Datasets for Purity Values

表 8 DLMC 与所有对比方法在 13 个基准数据集上的 Purity 值 %

对比方法	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
U-SENC	93.66	93.57	83.54	23.93	80.93	47.00	17.74	69.84	68.22	7.85	78.38	12.76	39.64
LMVSC	82.40	94.30	88.20	43.42	64.38	49.28	16.86	68.61	56.61	6.73	69.12	12.77	42.19
OPMC	83.74	76.54	92.40	43.67	54.70	45.23	15.82	56.38	49.75	9.34	69.23	15.21	23.62
FPMVS	72.30	40.63	94.96	44.33	33.01	42.59	24.32	81.64	31.53	8.88	71.49	23.35	-
OPLFMVC	66.40	80.88	94.10	30.42	80.25	52.04	17.70	72.57	72.27	8.51	73.41	19.98	49.75
EOMSC	68.60	38.42	89.44	45.92	39.88	42.08	22.52	64.91	31.63	8.74	81.19	26.66	-
MSGL	51.60	77.57	77.35	31.83	14.13	38.64	14.81	44.49	13.82	6.96	68.08	26.63	-
SFMC	76.20	70.22	78.21	23.92	70.25	23.28	14.53	66.86	67.20	3.90	68.47	N/A	N/A
EMKMC	87.40	91.18	93.34	41.33	45.56	49.88	15.86	64.95	45.54	6.00	74.23	21.60	28.51
TBGL	32.20	74.26	80.59	24.92	63.06	20.08	10.75	75.48	65.44	N/A	N/A	N/A	N/A
FDAGF	93.60	86.76	94.58	52.67	60.19	54.08	19.99	67.24	53.51	7.44	82.35	15.12	42.11
FastMICE	70.96	94.05	94.90	28.68	83.53	43.45	20.10	78.71	75.12	8.68	77.28	20.79	47.74
UDBGL	51.60	49.63	91.82	31.17	63.50	39.44	20.74	70.99	58.02	8.36	60.35	24.85	-
SMCMB	89.20	92.87	96.67	51.50	83.00	49.68	19.92	75.41	75.57	9.01	81.86	23.15	-
DLMC（本文）	**94.40**	**95.59**	**98.19**	**57.83**	**89.69**	**57.68**	**24.46**	**82.28**	**77.35**	**9.89**	**82.83**	**30.30**	**58.42**

注：黑体数值表示最佳结果；“N/A”表示由于内存不足导致算法在该数据集上无法运行；“-”表示由于运行时间过长（超 48 h 未完成）而终止了算法的运行.



(a) MSGL (b) FDAGF (c) LMVSC

(d) OPLFMVC (e) UDBGL (f) DLMC (本文)

Fig. 2 Visualization results of DLMC and other five comparative methods on D5

图 2 DLMC 与其他 5 个对比方法在 D5 上的可视化结果

SFMC 和 TBGL 在初始阶段为每个视图指定了相同的锚点，这种做法限制了算法充分利用视图间的一致性和差异性信息的能力. 此外，这些算法将图学习过程与图划分过程分离处理，可能引入了额外的误差，从而对聚类性能产生了不利影响. 相对而言，DLMC 的设计有效地规避了这些问题，其算法框架允许更灵活地处理视图间的信息，同时将图学习与图划分过程更紧密地结合起来，减少了因分离处理而产生的误差，从而在多视图聚类任务中取得了更优秀的性能.

3）与基于多锚点图的大规模集成聚类方法，如 U-SENC, OPLFMVC, FastMICE, SMCMB 相比，DLMC 在所有数据集上的聚类性能都得到了显著的提升. 追溯其原因，主要在于这些方法将基聚类器的生成与

集成学习划分为 2 个独立的阶段. 这种分离策略在基聚类器的生成阶段难以充分发挥多锚点图的潜在信息价值. 同时, 这种分阶段的处理方式也限制了最终聚类结果对多锚点图信息的深度挖掘和利用. 与之相反, 本文提出的 DLMC 通过统一的优化框架, 实现了对多个锚点图信息的全面利用, 其聚类过程不仅考虑了锚点端的特征, 也同时对样本端进行了综合分析. 这种端到端的聚类策略有效地避免了信息利用不足的问题, 从而在聚类性能上超越了传统的集成聚类方法.

4)与 14 种对比方法相比, DLMC 在大规模数据集 D12 和 D13 上的最优结果显著高于次优结果(超过 4%), 进一步证明了 DLMC 在大规模数据集上的有效性.

3.5 锚点图构造策略的影响

本节探讨了 2 种不同的锚点图构建策略对 DLMC 算法的影响.

3.5.1 1 个视图 1 个锚点图, 锚点个数一样但不对齐

在采用 1 个视图 1 个锚点图的构造策略时, 我们在不同视图上获取了锚点数量相同但不对齐的锚点集, 并进一步分析了锚点数量对聚类性能的影响. 图 3 展示了 DLMC 在不同数据集上的聚类性能随着锚点数量变化的情况. 观察图 3 可知, 在数据集 D9, D10, D12 上, 锚点数量的变化对聚类性能的影响微乎其微. 然而, 在 D4, D6, D7 数据集上, 随着锚点数目的增加, 聚类性能受到了数据集特性的影响而发生变化. 因此, 对于复杂的数据集可能需要更多的锚点来捕捉其复杂性, 而对于相对简单的数据集较少的锚点已经足够, 通过选取合适数量的锚点能够更好地捕捉数据的全局结构和模式, 从而提高聚类效果.

3.5.2 1 个视图 1 个锚点图, 锚点个数不一样不对齐

实际应用中, 不同视图的数据分布和特征重要性往往存在差异. 为了提高聚类效果, 根据每个视图的特性选择最合适的锚点显得尤为关键. 这样做不仅能增加聚类过程的多样性, 还有助于更全面地揭示数据中不同类别和结构的信息.

为了验证 DLMC 的普适性与有效性, 我们实施了一种策略, 即针对每个视图构建具有特定锚点的图, 从而在不同视图上得到数量各异的锚点结果(我们称之为 DLMC-d 策略). 通过对比表 9 中锚点数量相同与数量不同 2 种情况下的性能表现, 我们发现 DLMC 能够灵活应对不同配置下的多视图聚类问题. 更重要的是, 通过精细调整各视图的锚点数量, DLMC 进一步提升了多视图聚类的准确性. 这一实验结果不仅强调了处理多锚点图时采用不同锚点数量的必要性, 也凸显了 DLMC 在解决多视图聚类问题时的独特优势和应用价值.

3.6 收敛性分析

我们在图 4 中描绘了 DLMC 在 6 个基准数据集

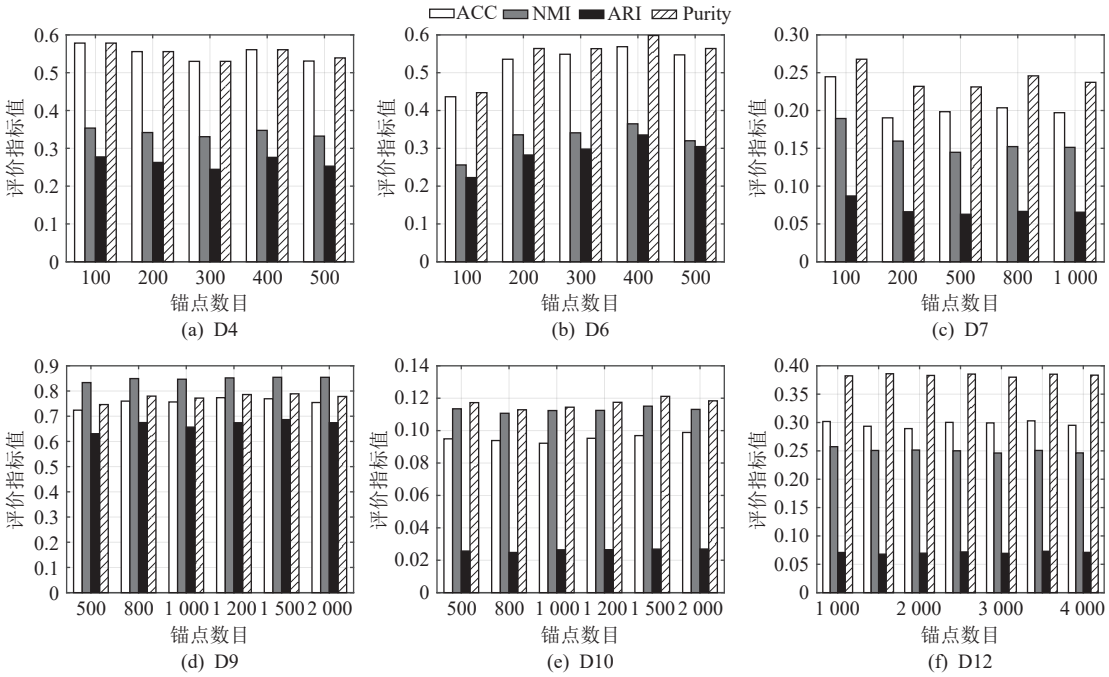


Fig. 3 Impact of consistent numbers but misaligned anchor on DLMC

图 3 数目一致但不对齐锚点对 DLMC 的影响

Table 9 Impact of the Consistency or Inconsistency of the Number of Anchors on DLMC in Different Views

表 9 不同视图下数目一致与不一致锚点对 DLMC 的影响

评价指标	方法	D1	D2	D4	D5	D6	D7	D8
ACC	DLMC	0.944 0	0.955 9	0.578 3	0.576 8	0.576 8	0.244 6	0.822 8
	DLMC-d	**0.958 0**	**0.968 8**	**0.628 3**	**0.913 8**	**0.592 8**	**0.249 4**	**0.835 0**
NMI	DLMC	0.834 9	0.865 7	0.353 5	0.364 8	0.364 8	0.189 4	0.701 3
	DLMC-d	**0.875 6**	**0.896 9**	**0.408 0**	**0.966 7**	**0.385 7**	**0.206 5**	**0.713 6**
ARI	DLMC	0.865 1	0.894 0	0.277 2	0.335 3	0.335 3	0.087 0	0.665 2
	DLMC-d	**0.897 2**	**0.917 9**	**0.326 7**	**0.894 7**	**0.328 4**	**0.088 4**	**0.682 4**
Purity	DLMC	0.944 0	0.955 9	0.578 3	0.598 4	0.598 4	0.267 8	0.822 8
	DLMC-d	**0.958 0**	**0.968 8**	**0.628 3**	**0.932 5**	**0.592 8**	**0.277 4**	**0.835 0**

注：黑体数值表示最优结果.

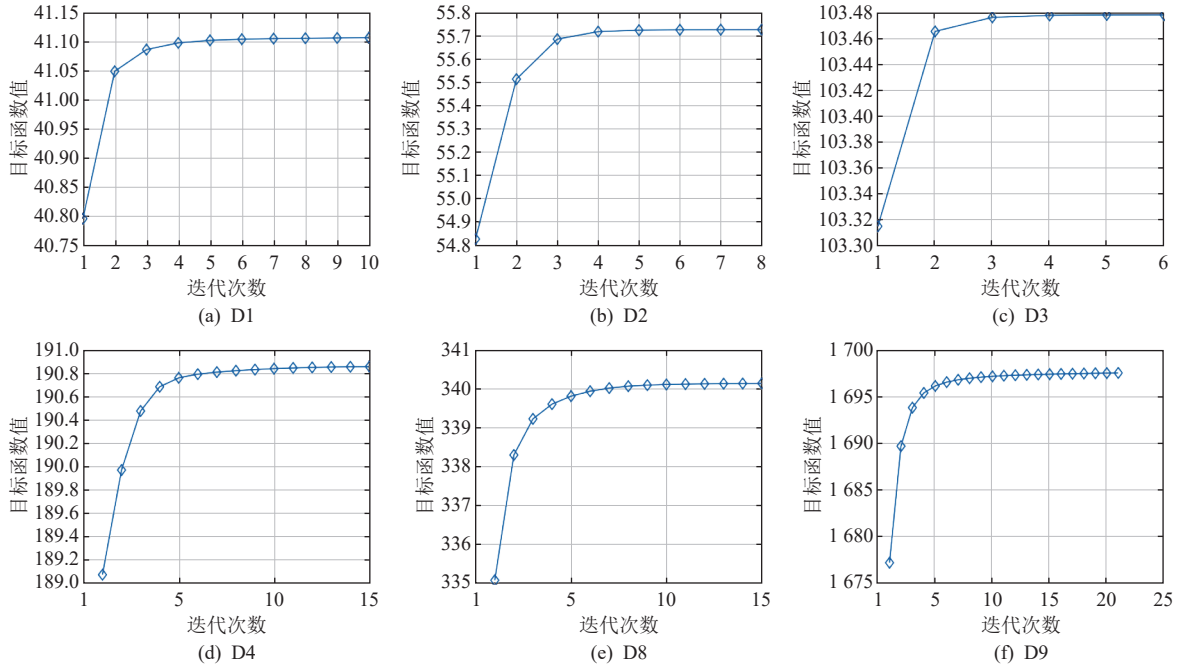


Fig. 4 Convergence curves of DLMC on six datasets

图 4 DLMC 在 6 个数据集上的收敛曲线

上随着迭代次数增加，其目标函数值的变化趋势，以直观展示每个数据集上的优化过程. 观察到在每个数据集上的迭代过程中，目标函数值都呈现出单调递增的趋势，并且很快收敛到稳定状态. 这说明 DLMC 在这些数据集上具有良好的收敛性能，能够有效地优化目标函数并得到稳定的聚类结果. 这进一步验证了 DLMC 的有效性和可靠性.

3.7 鲁棒性分析

在本节中我们通过 2 种特定的数据扰动情况即存在噪声的数据和不平衡的数据来评估 DLMC 的鲁棒性. 首先，我们在 BDGP 和 CCV 数据集中加入了不同水平(10%, 20%, 30%, 40%)的椒盐噪声(SP)、高斯噪声(GS)，以此产生 16 个不同噪声级别的数据集.

在表 10 和表 11 中，我们展示了 DLMC 与其他 6 种基线方法在这些噪声数据集上的聚类结果. 分析表明，DLMC 在不同类型的噪声不同程度的干扰条件下均表现出了卓越的聚类性能和鲁棒性. 而关于不平衡数据的鲁棒性测试则在表 4 中已对本文所采用的数据集的平衡性进行了展示，并且在 3.4 节中对其实验结果进行了分析，已经证明了 DLMC 在处理不平衡数据时的有效性，由于篇幅限制，此处不再重复. 总体而言，DLMC 与现有多锚点图聚类方法相比，在一个统一的框架内整合了多锚点图信息、锚点端的聚类信息以及样本端的聚类信息，确保了这三者之间的互补和深度融合，从而使得 DLMC 具有出色的鲁棒性.

Table 10 NMI Values of DLMC and Six Comparative Methods on D6 and D7 Datasets after Introducing Different Noises

表 10 DLMC 与 6 种对比方法在对 D6 和 D7 数据集上引入不同噪声后的 NMI 值 %

数据集	U-SENC	OPMC	MSGL	EMKMC	FastMICE	SMCMB	DLMC
D6 + 10%SP	25.0 ± 9.3	31.5 ± 4.2	25.0 ± 3.5	18.0 ± 11.3	27.1 ± 4.2	19.5 ± 5.3	**33.7 ± 3.7**
D6 + 20%SP	22.3 ± 11.5	33.2 ± 3.5	25.9 ± 2.6	16.5 ± 10.7	29.9 ± 6.6	19.7 ± 6.2	**28.8 ± 3.0**
D6 + 30%SP	28.8 ± 12.1	31.9 ± 4.8	28.8 ± 1.9	16.8 ± 11.3	26.2 ± 8.1	20.2 ± 5.9	**31.4 ± 2.2**
D6 + 40%SP	22.6 ± 11.3	33.1 ± 3.6	30.7 ± 1.9	15.1 ± 7.3	28.7 ± 3.1	20.2 ± 6.0	**32.6 ± 3.4**
D6 + 10%GS	19.5 ± 11.9	31.7 ± 3.8	25.7 ± 0.9	16.9 ± 11.5	24.7 ± 6.6	17.8 ± 6.1	**27.9 ± 4.3**
D6 + 20%GS	25.9 ± 10.7	32.5 ± 3.4	28.1 ± 1.1	18.2 ± 10.3	25.6 ± 9.1	19.3 ± 7.4	**30.1 ± 4.9**
D6 + 30%GS	25.7 ± 13.2	31.4 ± 4.5	25.8 ± 0.7	18.2 ± 10.1	26.1 ± 7.1	20.2 ± 7.5	**24.6 ± 3.9**
D6 + 40%GS	25.0 ± 9.3	32.7 ± 3.3	28.8 ± 3.1	18.0 ± 11.3	31.1 ± 3.2	19.1 ± 5.2	**29.5 ± 5.4**
D7 + 10%SP	14.7 ± 3.5	13.2 ± 0.5	3.2 ± 3.8	9.3 ± 1.4	15.3 ± 2.1	13.1 ± 2.0	**16.7 ± 1.1**
D7 + 20%SP	13.1 ± 3.0	13.2 ± 0.4	5.8 ± 3.9	9.2 ± 0.7	16.1 ± 1.7	13.0 ± 1.8	**16.6 ± 1.1**
D7 + 30%SP	12.9 ± 3.5	13.1 ± 0.4	6.5 ± 4.4	9.0 ± 1.2	16.2 ± 0.6	12.8 ± 2.0	**16.3 ± 1.0**
D7 + 40%SP	11.4 ± 1.6	13.2 ± 0.4	5.8 ± 3.9	8.9 ± 1.0	15.6 ± 1.1	12.9 ± 1.9	**16.2 ± 1.4**
D7 + 10%GS	13.5 ± 3.3	13.2 ± 0.7	5.9 ± 4.0	8.8 ± 1.0	15.2 ± 1.3	12.9 ± 2.0	**17.3 ± 1.0**
D7 + 20%GS	11.6 ± 2.9	12.9 ± 0.6	3.1 ± 3.5	9.1 ± 1.2	15.7 ± 1.3	12.9 ± 2.0	**17.0 ± 1.1**
D7 + 30%GS	11.6 ± 2.5	13.1 ± 0.6	5.7 ± 3.7	9.3 ± 0.9	16.0 ± 1.1	12.8 ± 1.8	**16.6 ± 1.1**
D7 + 40%GS	14.6 ± 4.0	12.7 ± 0.5	3.7 ± 4.7	9.2 ± 1.0	15.4 ± 1.8	13.0 ± 1.8	**16.8 ± 1.0**

注：黑体数值表示最优值, “±”前数值表示平均值; “±”后数值表示标准差.

Table 11 ACC Values of DLMC and Six Comparative Methods on D6 and D7 Datasets after Introducing of Different Noises

表 11 DLMC 与 6 种对比方法在对 D6 和 D7 数据集上引入不同噪声后的 ACC 值 %

数据集	U-SENC	OPMC	MSGL	EMKMC	FastMICE	SMCMB	DLMC
D6 + 10%SP	46.8 ± 8.2	47.9 ± 3.5	45.5 ± 4.2	41.2 ± 11.3	48.4 ± 3.6	41.0 ± 4.7	**54.4 ± 1.8**
D6 + 20%SP	44.2 ± 9.9	49.0 ± 2.7	44.3 ± 3.6	39.2 ± 10.6	47.7 ± 5.6	40.9 ± 5.6	**51.4 ± 4.1**
D6 + 30%SP	49.2 ± 9.5	47.7 ± 4.1	50.6 ± 2.8	39.0 ± 11.2	46.9 ± 6.9	41.2 ± 5.7	**53.8 ± 2.0**
D6 + 40%SP	43.2 ± 9.6	47.6 ± 4.2	51.2 ± 1.3	37.5 ± 9.2	48.7 ± 3.0	41.8 ± 5.9	**52.9 ± 3.0**
D6 + 10%GS	40.6 ± 9.1	47.9 ± 3.1	49.1 ± 1.1	40.4 ± 13.1	45.6 ± 6.1	40.5 ± 5.3	**49.0 ± 3.9**
D6 + 20%GS	43.8 ± 8.8	47.2 ± 4.1	50.3 ± 1.4	41.5 ± 11.2	46.6 ± 7.3	41.7 ± 6.2	**51.1 ± 4.2**
D6 + 30%GS	44.3 ± 10.9	47.4 ± 3.8	46.9 ± 1.2	41.6 ± 11.5	45.5 ± 5.5	42.2 ± 5.8	**50.2 ± 4.8**
D6 + 40%GS	46.8 ± 8.2	48.5 ± 3.4	49.4 ± 3.7	41.2 ± 11.3	50.2 ± 3.9	41.0 ± 4.9	**51.0 ± 4.1**
D7 + 10%SP	18.9 ± 3.6	15.8 ± 0.4	11.4 ± 1.5	14.1 ± 1.9	20.2 ± 2.1	17.3 ± 2.2	**21.5 ± 0.8**
D7 + 20%SP	17.2 ± 2.7	16.2 ± 0.5	13.0 ± 2.0	14.3 ± 0.8	20.8 ± 1.6	17.5 ± 2.2	**21.7 ± 0.7**
D7 + 30%SP	17.0 ± 3.1	16.0 ± 0.5	12.5 ± 1.9	14.2 ± 0.6	20.8 ± 0.7	17.2 ± 2.3	**21.0 ± 0.7**
D7 + 40%SP	15.4 ± 1.9	16.0 ± 0.5	13.7 ± 2.5	14.3 ± 1.2	20.0 ± 1.1	17.4 ± 2.3	**21.4 ± 1.1**
D7 + 10%GS	18.1 ± 2.8	16.0 ± 0.6	13.7 ± 2.6	14.2 ± 1.1	20.0 ± 1.0	17.4 ± 2.3	**22.7 ± 0.6**
D7 + 20%GS	15.8 ± 2.6	15.8 ± 0.6	11.9 ± 2.0	14.5 ± 1.3	20.3 ± 1.3	17.4 ± 2.3	**21.7 ± 1.1**
D7 + 30%GS	15.6 ± 2.3	16.0 ± 0.6	13.5 ± 2.5	14.3 ± 0.8	20.8 ± 1.2	17.4 ± 2.0	**21.5 ± 0.5**
D7 + 40%GS	18.4 ± 3.4	15.7 ± 0.4	11.7 ± 2.1	14.5 ± 0.6	20.0 ± 1.5	17.5 ± 2.1	**21.7 ± 0.8**

注：黑体数值表示最优值, “±”前数值表示平均值; “±”后数值表示标准差.

3.8 运行时间对比

为了更直观地评估 DLMC 的计算效率, 本文在图 5 展示了 DLMC 和所有对比方法在除 D1 外的 12 个数据集上对数形式化后的运行时间. 结果表明, 相较于其他方法, DLMC 在处理多视图大规模聚类任务时表现突出. 综合考虑实验结果和运行时间, DLMC 在处理多视图大规模聚类任务时具有明显的优势.

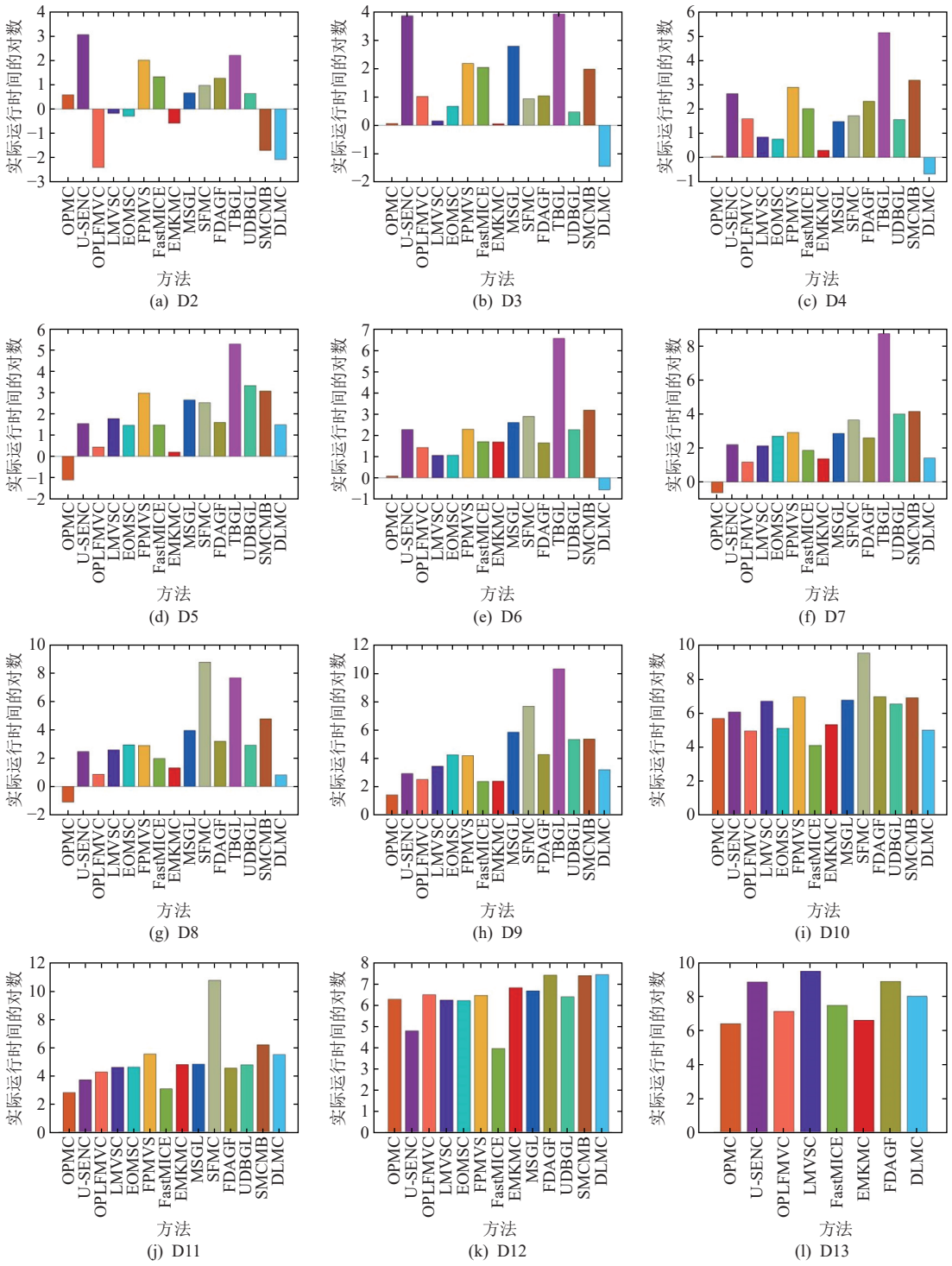


Fig. 5 Running time of all comparative methods on different datasets

图 5 所有对比方法在不同数据集上的运行时间

4 结 论

本文提出的 DLMC 算法通过双端联合学习框架，将锚点端聚类与样本端聚类有机结合，实现了在多锚点图协作下的相互增强. 尽管该方法在多视图聚类中表现优异，但是在锚点选择上仍面临很大的挑战，针对不同规模的数据集确定合适的锚点数目并选择最优的锚点集仍然是一大难题. 因此，未来我们会侧重于去探索更为可靠的锚点选择技术，以适应多样化数据集，提升聚类性能和鲁棒性. 此外，我们也会去开发更加灵活的算法变体，使其能够自动调

整以适应数据的不平衡性和复杂性并将其推广到生物信息学、社交网络分析等领域的实际应用中.

作者贡献声明: 杜亮负责算法和实验方案的设计; 李晓东和陈艳负责实验结果的整理与分析, 并参与论文的撰写与修订; 周芃负责改进实验过程; 钱宇华则把握论文的创新性, 并指导论文的修订工作.

参 考 文 献

[1] Chao Guoqing, Sun Shiliang, Bi Jinbo. A survey on multiview clustering[J]. IEEE Transactions on Artificial Intelligence, 2021, 2(2): 146−168

[2] Cai Deng, Chen Xinlei. Large scale spectral clustering via landmark-based sparse representation[J]. IEEE Transactions on Cybernetics, 2014, 45(8): 1669−1680

[3] Nie Feiping, Wang Xiaoqian, Huang Heng. Clustering and projected clustering with adaptive neighbors[C]//Proc of the 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2014: 977−986

[4] Kang Zhao, Zhou Wangtao, Zhao Zhitong, et al. Large-scale multi-view subspace clustering in linear time[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2020: 4412−4419

[5] Wang Siwei, Liu Xinwang, Liu Suyuan, et al. Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences[J]. Advances in Neural Information Processing Systems, 2022, 35: 5882−5895

[6] Li Xuelong, Zhang Han, Wang Rong, et al. Multiview clustering: A scalable and parameter-free bipartite graph fusion method[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(1): 330−344

[7] Wei Xia, Quanxue Gao, Wang Qianqian, et al. Tensorized bipartite graph learning for multi-view clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 5187−5202

[8] Yang Ben, Zhang Xuetao, Li Zhongheng, et al. Efficient multi-view k-means clustering with multiple anchor graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(7): 6887−6900

[9] Lao Jinghuan, Huang Dong, Wang Changdong, et al. Towards scalable multi-view clustering via joint learning of many bipartite graphs[J]. IEEE Transactions on Big Data, 2024, 10(1): 77−91

[10] Huang Dong, Wang Changdong, Wu Jiansheng, et al. Ultra-scalable spectral clustering and ensemble clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(6): 1212−1226

[11] Wei Zhu, Nie Feiping, Li Xuelong. Fast spectral clustering with efficient large graph construction[C]//Prof of the 2017 IEEE Int Conf on Acoustics, Speech and Signal Processing. Piscataway, NJ: IEEE, 2017: 2492−2496

[12] Zhang Pei, Wang Siwei, Li Liang, et al. Let the data choose: Flexible and diverse anchor graph fusion for scalable multi-view clustering[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2023: 11262−11269

[13] Yu Xiao, Liu Hui, Lin Yuxiu, et al. Consistency-guided adaptive weighted multi-view clustering[J]. Journal of Computer Research and Development, 2022, 59(7): 1496−1508 (in Chinese)
(于晓, 刘慧, 林毓秀, 等. 一致性引导的自适应加权多视图聚类[J]. 计算机研究与发展, 2022, 59(7): 1496−1508)

[14] Huang Dong, Wang Changdong, Lai Jianhuang. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(11): 11388−11402

[15] Zhou Peng, Du Liang, Liu Xinwang, et al. Partial clustering ensemble[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(5): 2096−2109

[16] Liu Jiyuan, Liu Xinwang, Yang Yuexiang, et al. One-pass multi-view clustering for large-scale data[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 12344−12353

[17] Kang Zhao, Lin Zhiping, Zhu Xiaofeng, et al. Structured graph learning for scalable subspace clustering: From single view to multiview[J]. IEEE Transactions on Cybernetics, 2022, 52(9): 8976−8986

[18] Wang Siwei, Liu Xinwang, Zhu Xinzhong, et al. Fast parameter-free multi-view subspace clustering with consensus anchor guidance[J]. IEEE Transactions on Image Processing, 2022, 31: 556−568

[19] Liu Suyuan, Wang Siwei, Zhang Pei, et al. Efficient one-pass multi-view subspace clustering with consensus anchors[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2022: 7576−7584

[20] Fang Siguo, Huang Dong, Cai Xiaosha, et al. Efficient multi-view clustering via unified and discrete bipartite graph learning[J]. arXiv preprint, arXiv: 2209.04187, 2023

[21] Liu Xinwang, Liu Li, Liao Qing, et al. One pass late fusion multi-view clustering[C]//Proc of the Int Conf on Machine Learning. New York: ACM, 2021: 6850−6859

[22] Yang Ben, Zhang Xuetao, Nie Feiping, et al. Fast multi-view clustering via nonnegative and orthogonal factorization[J]. IEEE Transactions on Image Processing, 2020, 30: 2575−2586

[23] Cai Xiao, Nie Feiping, Huang Heng. Multi-view *k*-means clustering on big data[C]//Prof of the 23rd Int Joint Conf on Artificial Intelligence. San Francisco, CA: Margan Kaufmann, 2013: 2598−2604

[24] Xu Jinglin, Han Junwei, Nie Feiping, et al. Re-weighted discriminatively embedded *k*-means for multi-view clustering[J]. IEEE Transactions on Image Processing, 2017, 26(6): 3016−3027

[25] Nie Feiping, Shi Shaojun, Li Xuelong. Auto-weighted multi-view co-clustering via fast matrix factorization[J]. Pattern Recognition, 2020, 102: 107207

[26] Fred A L N, Jain A K. Combining multiple clusterings using evidence accumulation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(6): 835−850

[27] Iam-On N, Boongoen T, Garrett S, et al. A link-based approach to the cluster ensemble problem[J]. IEEE Transactions on Pattern Analysis

and Machine Intelligence, 2011, 33(12): 2396−2409

[28] Tao Zhiqiang, Liu Hongfu, Li Sheng, et al. Robust spectral ensemble clustering[C]//Proc of the 25th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2016: 367−376

[29] Huang Dong, Wang Changdong, Lai Jianhuang. Locally weighted ensemble clustering[J]. IEEE Transactions on Cybernetics, 2017, 48(5): 1460−1473

[30] Topchy A, Jain A K, Punch W. Clustering ensembles: Models of consensus and weak partitions[J]. IEEE Transactions on Pattern Analysisand Machine Intelligence, 2005, 27(12): 1866−1881

[31] Bai Liang, Liang Jiye, Du Hangyuan. An information-theoretical framework for cluster ensemble[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(8): 1464−1477

[32] Strehl A, Ghosh J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions[J]. Journal of Machine Learning Research, 2002, 3: 583−617

[33] Zhou Peng, Du Liang, Liu Xinwang, et al. Self-paced clustering ensemble[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(4): 1497−1511

[34] Li Zhihui, Nie Feiping, Chang Xiaojun, et al. Balanced clustering via exclusive lasso: A pragmatic approach[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2018: 3596−3603



Du Liang, born in 1985. PhD, associate professor, PhD supervisor. Member of CCF. His main research interests include unsupervised learning and machine learning.

杜　亮, 1985 年生. 博士, 副教授, 博士生导师. CCF 会员. 主要研究方向为无监督学习、机器学习.



Li Xiaodong, born in 1999. Master. Member of CCF. His main research interest includes unsupervised learning.

李晓东, 1999 年生. 硕士. CCF 会员. 主要研究方向为无监督学习.



Chen Yan, born in 1994. PhD, lecturer. Her main research interest includes unsupervised learning.

陈　艳, 1994 年生. 博士, 讲师. 主要研究方向为无监督学习.



Zhou Peng, born in 1989. PhD, associate professor, PhD supervisor. Member of CCF. His main research interests include machine learning, data mining, and artificial intelligence.

周　芃, 1989 年生. 博士, 副教授, 博士生导师. CCF 会员. 主要研究领域为机器学习、数据挖掘、人工智能.



Qian Yuhua, born in 1976. Professor, PhD supervisor. His main research interests include pattern recognition, feature selection, rough set theory, granular computing, and artificial intelligence.

钱宇华, 1976 年生. 教授, 博士生导师. 主要研究方向为模式识别、特征选择、粗糙集理论、粒计算、人工智能.