软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.006995]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于二部图的联合谱嵌入多视图聚类算法*

赵兴旺[1,2], 王淑君[1,2], 刘晓琳[3], 梁吉业[1,2]

[1](山西大学 计算机与信息技术学院, 山西 太原 030006)

[2](计算智能与中文信息处理教育部重点实验室 (山西大学), 山西 太原 030006)

[3](中北大学 计算机科学与技术学院, 山西 太原 038507)

通信作者: 梁吉业, E-mail: ljy@sxu.edu.cn

摘　要: 多视图聚类在图像处理、数据挖掘和机器学习等领域引起了越来越多的关注. 现有的多视图聚类算法存在两个不足, 一是在图构造过程中只考虑每个视图数据之间的成对关系生成亲和矩阵, 而缺乏邻域关系的刻画; 二是现有的方法将多视图信息融合和聚类的过程相分离, 从而降低了算法的聚类性能. 为此, 提出一种更为准确和鲁棒的基于二部图的联合谱嵌入多视图聚类算法. 首先, 基于多视图子空间聚类的思想构造二部图进而产生相似图, 接着利用相似图的谱嵌入矩阵进行图融合, 其次, 在融合过程中考虑每个视图的重要性进行权重约束, 进而引入聚类指示矩阵得到最终的聚类结果. 提出的模型将二部图、嵌入矩阵与聚类指示矩阵约束在一个框架下进行优化. 此外, 提供一种求解该模型的快速优化策略, 该策略将优化问题分解成小规模子问题, 并通过迭代步骤高效解决. 提出算法和已有的多视图聚类算法在真实数据集上进行实验分析. 实验结果表明, 相比已有方法, 提出算法在处理多视图聚类问题上是更加有效和鲁棒的.

关键词: 多视图聚类; 子空间聚类; 二部图; 谱嵌入矩阵; 聚类指示矩阵

中图法分类号: TP311

中文引用格式: 赵兴旺, 王淑君, 刘晓琳, 梁吉业. 基于二部图的联合谱嵌入多视图聚类算法. 软件学报. http://www.jos.org.cn/1000-9825/6995.htm

英文引用格式: Zhao XW, Wang SJ, Liu XL, Liang JY. Joint Spectral Embedding Multi-view Clustering Algorithm Based on Bipartite Graphs. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6995.htm

Joint Spectral Embedding Multi-view Clustering Algorithm Based on Bipartite Graphs

ZHAO Xing-Wang[1,2], WANG Shu-Jun[1,2], LIU Xiao-Lin[3], LIANG Ji-Ye[1,2]

[1](School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China)

[2](Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education (Shanxi University), Taiyuan 030006, China)

[3](School of Computer Science and Technology, North University of China, Taiyuan 038507, China)

Abstract: Multi-view clustering has attracted more and more attention in the fields of image processing, data mining, and machine learning. Existing multi-view clustering algorithms have two shortcomings. One is that in the process of graph construction, only the pairwise relationship between each view data is considered to generate an affinity matrix, which lacks the characterization of neighborhood relationships; the second is that existing methods separate the process of multi-view information fusion and clustering, thereby reducing the clustering performance of the algorithm. Therefore, this study proposes a more accurate and robust joint spectral embedding multi-view clustering algorithm based on bipartite graphs. Firstly, based on the multi-view subspace clustering idea,bipartite graphs are constructed, and similar graphs are generated.Then the spectral embedding matrix of similar graphs is used to perform graph fusion. Secondly, by considering the importance of each view during the fusion process, weight constraints are applied, and an indicator matrix is introduced to

* 基金项目: 国家自然科学基金 (62072293, U21A20473); 山西省自然科学基金 (202203021222048)
收稿时间: 2023-03-31; 修改时间: 2023-05-06; 采用时间: 2023-06-27; jos 在线出版时间: 2023-11-15

obtain the final clustering result. A model is proposed to optimize the bipartite graph, embedding matrix, and clustering indicator matrix within a single framework. In addition, a fast optimization strategy for solving the model is provided, which decomposes the optimization problem into small module subproblems and efficiently solves them through iterative steps. The proposed algorithm and existing multi-view clustering algorithms have been experimentally analyzed on real data sets. Experimental results show that the proposed algorithm is more effective and robust in dealing with multi-view clustering problems compared with existing methods.

Key words: multi-view clustering; subspace clustering; bipartite graph; spectral embedding matrix; clustering indicator matrix

聚类分析是数据挖掘、机器学习领域中一种经典且基础的无监督学习方法. 无监督学习的目标是通过对无标记训练样本的学习, 发掘和揭示数据集本身潜在的结构与规律, 聚类试图将数据集的样本划分为若干个互不相交的类簇, 使得每个簇对应一个潜在的类别[1]. 例如, 在复杂网络分析[2]中, 人们希望发现具有内在紧密联系的社团; 在图像分析[3]中, 人们希望将图像分割成具有类似性质的区域; 在文本处理[4]中, 人们希望发现具有相同主题的文本子集; 在顾客行为分析中, 人们希望发现消费方式类似的顾客群, 以便制订有针对性的客户管理方式来提高营销效率. 聚类分析作为一种典型的无监督学习方法, 过去的几十年里在众多实际场景中取得了长足的发展.

然而, 随着信息技术的发展, 数据采集方式的多样化, 数据逐渐呈现出多源异构的特点. 称这种不同来源或不同模态的数据为多视图数据[5−7]. 例如, “横看成岭侧成峰”表示不同的方位产生不同的视图, 但都描述的同一物体; 对于网页数据, 可以通过文本或者网页链接的形式获取数据来产生两个视图数据; 新闻可用多种语言报道, 每种语言表示一个视图等. 在聚类领域, 随着多视图数据的大量涌现, 多视图聚类已经成为一类重要的聚类方法[8−10]. 多视图聚类作为一种新的机器学习范式, 旨在通过联合学习多个视图的特征信息, 将视图中相似的样本划分到同一个簇, 将视图中不相似的样本划分到不同的类簇, 并且要求多个视图之间具有一致的划分结果[11]. 一致性和互补性是多视图聚类算法的两大原则, 如何充分利用多个视图的$n\times n$信息成为多视图聚类算法共同面临的挑战.

近年来, 研究者已提出了多种多视图聚类算法, 并在计算机视觉、模式识别和生物医学等领域得到了广泛应用[12,13]. 提出的算法主要分为: 基于图学习的算法[14−19]、基于子空间学习的算法[20]、基于集成学习的算法[21]和基于深度学习的算法[22−24]等. 但是, 大多数现有算法存在以下问题: 在图构造阶段, 学习的亲和矩阵时利用了样本对之间的成对关系, 造成了空间的极大浪费, 且缺少邻域信息的刻画, 未能充分捕获各视图的类簇信息. 在图融合阶段, 平等地对待每个视图, 将多个图或者表示矩阵融合为一个共享的矩阵, 在相似性或者表示级别上融合多视图信息, 缺少了划分级别信息的利用. 在图聚类阶段, 现有算法在得到共享亲和矩阵后, 执行 K-means 或其他聚类算法得到最终聚类结果, 将相似性融合和聚类过程相分离, 使得聚类结果不能反馈给融合过程, 从而产生次优解, 导致无法获得最优的聚类结果.

针对以上问题, 本文提出了一种基于二部图的联合谱嵌入多视图聚类算法. 其主要目的旨在利用联合嵌入矩阵探索噪声较少的共享空间, 提高多视图聚类算法的准确性和可靠性. 该算法在图构造阶段利用效率较高的 K-means 算法选出锚点, 基于多视图子空间聚类思想构造二部图, 利用二部图构造出的相似图的谱嵌入矩阵进行图融合. 同时考虑每个视图的重要性进行权重约束, 引入聚类指示矩阵得到最终的聚类结果. 提出模型将二部图、嵌入矩阵、聚类指示矩阵约束在同一个框架下进行优化, 有效提升了算法的稳定性, 且聚类指示矩阵直接输出聚类结果而不需要额外的算法, 有效提升了算法效率.

本文第 1 节简要介绍了多视图聚类相关工作. 第 2 节提出了基于二部图的联合谱嵌入多视图聚类算法. 第 3 节对提出算法和已有算法进行了实验分析. 最后, 第 4 节对本文的工作进行了总结.

1 相关工作

1.1 基于子空间学习的算法

子空间聚类使用数据样本的自表达特性, 即每个样本都可以由少数其他数据样本的线性组合表示. 多视图子空间学习的核心思想在于尽可能保留每个视图特有分布信息的情况下, 寻找多个视图共享的表示空间. 经典的子空间学习方法有: 典型相关分析、矩阵分析、自表示、主题模型、字典学习等[25,26]. Huang 等人[27]从多个不充分

的视图中同时恢复潜在的完整空间, 然后采用矩阵分解的方法, 将潜在的完整空间表示分解为聚类质心和聚类分配. Zhang 等人[28]提出了一种基于张量的多视图子空间表示学习方法, 将不同视图的子空间表示矩阵视为一个低阶张量. 采用低秩约束的子空间表示张量, 减少子空间表示的冗余, 提高了后续任务的准确性. Zhang 等人[29]不是直接融合多个亲和矩阵, 而是优化集成判别分区级信息, 有助于消除数据之间的噪声, 在统一的框架下同时学习了亲和矩阵、共识表示和最终聚类标签矩阵. Wang 等人[30]提出了一种新的子空间聚类方法, 将锚点选择和子空间图构造成一个统一的优化框架, 这两个过程相互调整, 以提高聚类质量. 此外, 该方法可以自动学习一个最优的锚点图, 而不需要任何额外的超参数. 基于子空间的聚类算法具有一定的稳定性和鲁棒性, 但是产生的低维潜在表示存在着难以解释的问题.

1.2 基于图学习的算法

基于图学习的算法思想是找出一个由所有视图共享, 且能够刻画所有视图相互关系的融合图, 在融合图上使用图切割算法或其他谱图技术, 最终获得聚类结果. 该类算法的 3 个关键步骤: 1) 基于单视图分别构造初始图; 2) 学习融合全部视图拓扑结构的一致图; 3) 基于图分割获得最终聚类结果. 其中如何构造高质量的初始图和如何设计有效的图融合策略是该类算法的关键所在. Zhan 等人[31]将所有视图的数据图矩阵进行融合, 生成一个统一的图矩阵, 进而改进了每个视图的数据图矩阵, 并直接给出了最终的聚类. 算法的关键之处在于它的学习方法可以帮助学习每个视图图矩阵和以相互强化的方式学习统一的图矩阵. Wang 等人[32]考虑到不同视图的重要性, 自动生成权重来处理视图间的差异性, 直接产生融合图的聚类无需额外步骤. Li 等人[33]利用少量的有代表性的统一锚点来构造二部图, 降低了空间复杂度, 在一个框架中共同学习每个视图的相似图、统一的二部图和具有代表性的统一锚点集. 该类算法具有可解释性强的特点, 在聚类结果的可解释性方面具有一定优势. 然而, 该类算法的性能大多依赖于图的初始化, 而初始图的质量通常难以得到保障.

为了克服上述方法存在的不足, 本文采用二部图替代亲和矩阵来捕获视图的划分信息, 并利用噪声及冗余较少的嵌入矩阵进行融合, 将图融合过程与聚类过程约束在一个统一框架下, 因而可以产生更好的聚类性能.

2 基于二部图的联合谱嵌入多视图聚类算法

本节首先对多视图聚类问题进行了描述, 接着对本文提出的算法进行了介绍, 并给出了详细的优化过程和算法流程, 最后对算法的时间复杂度进行了分析.

2.1 问题描述

给定一个包含 n 个样本、m 个视图的多视图数据集, 用 $\boldsymbol{X}^1,\ldots,\boldsymbol{X}^m$ 表示 m 个视图的数据矩阵. 其中, $\boldsymbol{X}^v=\{x_1^v,\ldots,x_n^v\}\in\mathbb{R}^{d_v\times n}$ 表示含有特征个数为 d_v 的 n 个样本的数据矩阵. 多视图聚类的目标是: 将原始的 n 个样本划分为 c 个簇, 并且遵循两个原则, 即一致性和互补性, 来强调视图间存在共性和差异性.

本文中矩阵用粗体大写字母表示 (例如 $\boldsymbol{X}$). 文中所使用的重要符号见后文表 1 所示.

2.2 基于二部图的联合谱嵌入多视图聚类算法

本文提出算法的主要框架如图 1 所示, 主要包括初始二部图构造和联合谱嵌入融合 2 个步骤, 下面将详细描述.

2.2.1 初始二部图构造

传统的图构造方法对所有视图生成 $n\times n$ 的亲和矩阵来反应样本对之间的关系, 但缺乏了邻域关系间信息的刻画. 考虑到 $t\times n$ 的二部图首先可以反映邻域关系, 其次还可以间接反应样本对之间的关系, 且可以更好地捕获视图的划分结构, 因此本文用二部图代替了全样本图. 所提算法首先对原始数据进行采样, 生成 t 个锚点, 然后利用锚点和每个视图的原始数据构造初始二部图矩阵 $\tilde{\boldsymbol{Z}}^v\in\mathbb{R}^{t\times n}$. 具体地, 由于 K-means 对初始聚类中心敏感, 因此针对各个视图分别采用 K-means++聚类算法[34]选择 t 个初始聚类中心, 并对各个视图数据进行聚类, 收集这些簇的中心来形成对应视图的锚点矩阵 $\boldsymbol{A}^v\in\mathbb{R}^{d_v\times t}$. 进而, 采用一种基于 k 近邻的方法进行二部图构造, $\tilde{z}_{ij}^{(v)}$ 可以用公式 (1)

得到.

$$\tilde{z}_{ij}^{(v)}=\begin{cases}\dfrac{d\left(a_i^{(v)},x_j^{(v)}\right)}{\sum_{j'\in\langle i\rangle}d\left(a_i^{(v)},x_{j'}^{(v)}\right)}, & \text{if } i\in\langle j\rangle\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

其中, $i\in\langle j\rangle$ 表示锚点 $a_i^{(v)}$ 属于数据点 $x_j^{(v)}$ 的 k 近邻 ($k<t$), 通过归一化使得矩阵 $\tilde{\boldsymbol{Z}}$ 的列和为 1. 在本文中邻域个数 k 设置为 5.

表 1 符号及含义

符号表示	符号说明
$\boldsymbol{X}^v\in\mathbb{R}^{d_v\times n}$	第 v 个视图的数据矩阵
$\boldsymbol{A}^v\in\mathbb{R}^{d_v\times t}$	第 v 个视图的锚点矩阵
$\boldsymbol{Z}^v\in\mathbb{R}^{t\times n}$	第 v 个视图的二部图矩阵
$\boldsymbol{F}^v\in\mathbb{R}^{n\times c}$	第 v 个视图的嵌入矩阵
$\boldsymbol{F}^*\in\mathbb{R}^{n\times c}$	联合谱嵌入矩阵
$\boldsymbol{S}^v\in\mathbb{R}^{n\times n}$	第 v 个视图的亲和矩阵
$\boldsymbol{G}\in\mathbb{R}^{n\times c}$	聚类指示矩阵
α_v	第 v 个视图的权重
t	锚点个数
λ	正则化参数
$\boldsymbol{I}$	单位矩阵
$\mathbf{1}$	所有项都是1的列向量
$\mathrm{tr}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 的迹
$\|\boldsymbol{X}\|_F$	矩阵 $\boldsymbol{X}$ 的F范数

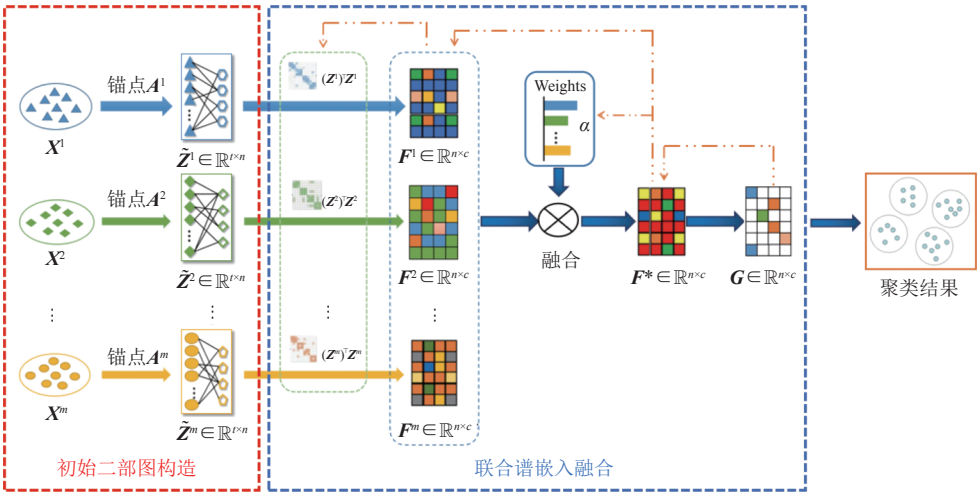


图 1 算法流程示意图

2.2.2 联合谱嵌入融合

得到初始二部图之后, 接下来将进行联合谱嵌入融合. 考虑到初始二部图是一种静态的构图方法, 具有一定的局限性, 因此本文将对二部图进行动态更新. 基于多视图子空间聚类的思想, 通过解决以下问题, 可以得到每个视图的二部图 $\boldsymbol{Z}^v$.

$$\begin{cases}\min\limits_{\boldsymbol{Z}^v}\sum\limits_{v=1}^{m}\|\boldsymbol{X}^v-\boldsymbol{A}^v\boldsymbol{Z}^v\|_F^2+\lambda\|\boldsymbol{Z}^v\|_F^2\\ \text{s.t. } 0\leqslant Z_{ij}^v\leqslant 1,\ (\boldsymbol{Z}^v)^{\top}\boldsymbol{1}=\boldsymbol{1}\end{cases}\tag{2}$$

其中, $\boldsymbol{Z}^v$ 表示锚点和数据点之间的相似性. $\boldsymbol{Z}^v$ 中的元素应该是非负的, 并且列和为 1.

在得到每个视图的动态二部图后将进行谱划分, 理想的聚类相似结构应该具有亲和矩阵中的连通分量个数等于类簇个数的性质, 该相似性结构将有助于后续的聚类[35]. 即如果亲和矩阵有 c 个连通分量, 通过 $rank(\boldsymbol{L}^v)=n-c$ 的约束, 则可以将样本直接划分为 c 个簇. 自然地, 为了达到这个条件可以在优化目标中添加一个秩约束. 因此, 最优化问题将变成:

$$\begin{cases}\min\limits_{\boldsymbol{Z}^v}\sum\limits_{v=1}^{m}\|\boldsymbol{X}^v-\boldsymbol{A}^v\boldsymbol{Z}^v\|_F^2+\lambda\|\boldsymbol{Z}^v\|_F^2\\ \text{s.t. } 0\leqslant Z_{ij}^v\leqslant 1,\ (\boldsymbol{Z}^v)^{\top}\boldsymbol{1}=\boldsymbol{1},\ rank(\boldsymbol{L}^v)=n-c\end{cases}\tag{3}$$

其中, $\boldsymbol{L}^v=\boldsymbol{D}^v-\boldsymbol{S}^v$, $\boldsymbol{D}^v$ 是 $\boldsymbol{S}^v$ 的度矩阵, $\boldsymbol{S}^v$ 可以通过 $(\boldsymbol{Z}^v)^{\top}\boldsymbol{Z}^v$ 获得.

然而, 直接采用秩约束 $rank(\boldsymbol{L}^v)=n-c$ 会使得优化问题难以解决. 根据 K-F 定理[36] $\sum_{i=1}^{c}\sigma_i(\boldsymbol{L}^v)=\min_{(\boldsymbol{F}^v)^{\top}\boldsymbol{F}^v=\boldsymbol{I}_c}\mathrm{tr}((\boldsymbol{F}^v)^{\top}\boldsymbol{L}^v\boldsymbol{F}^v)$, $\sigma_i(\boldsymbol{L}^v)$ 是 $\boldsymbol{L}^v$ 的第 i 个最小特征值, $\boldsymbol{F}^v\in\mathbb{R}^{n\times c}$ 是第 v 个视图的嵌入矩阵. 因此可以把秩约束问题转化为最小化 $\mathrm{tr}((\boldsymbol{F}^v)^{\top}\boldsymbol{L}^v\boldsymbol{F}^v)$. 很明显, 当 $\sum_{i=1}^{c}\sigma_i(\boldsymbol{L}^v)=0$ 时秩约束成立, 因此公式 (3) 可以转化成以下形式:

$$\begin{cases}\min\limits_{\boldsymbol{Z}^v,\boldsymbol{F}^v}\sum\limits_{v=1}^{m}\|\boldsymbol{X}^v-\boldsymbol{A}^v\boldsymbol{Z}^v\|_F^2+\lambda\|\boldsymbol{Z}^v\|_F^2+\mathrm{tr}((\boldsymbol{F}^v)^{\top}\boldsymbol{L}^v\boldsymbol{F}^v)\\ \text{s.t. } 0\leqslant Z_{ij}^v\leqslant 1,\ (\boldsymbol{Z}^v)^{\top}\boldsymbol{1}=\boldsymbol{1},\ (\boldsymbol{F}^v)^{\top}\boldsymbol{F}^v=\boldsymbol{I}_c\end{cases}\tag{4}$$

如公式 (4) 所示, 每个视图都可以得到其单独的谱嵌入表示 $\boldsymbol{F}^v$. 考虑多视图聚类算法一致性原则, 将多个视图的 $\boldsymbol{F}^v$ 融合成一个联合谱嵌入矩阵 $\boldsymbol{F}^*$, 以跨视图集成多个视图的划分信息. 由于谱嵌入融合对比相似矩阵融合而言, 可以提供更多的鉴别信息和更少的冗余及噪声, 因此, 融合 $\boldsymbol{F}^v$ 可以获得优于相似矩阵融合的性能. 在融合时考虑不同视图的重要性引入视图权重 α_v, 可以将上述融合过程表述为:

$$\begin{cases}\min\limits_{\boldsymbol{F}^v,\boldsymbol{F}^*}\sum\limits_{v=1}^{m}\alpha_v\|\boldsymbol{F}^v-\boldsymbol{F}^*\|_F^2\\ \text{s.t. } (\boldsymbol{F}^v)^{\top}\boldsymbol{F}^v=\boldsymbol{I}_c,\ (\boldsymbol{F}^*)^{\top}\boldsymbol{F}^*=\boldsymbol{I}_c,\ \alpha_v\geqslant 0,\ \boldsymbol{1}^{\top}\boldsymbol{\alpha}=\boldsymbol{1}\end{cases}\tag{5}$$

其中, $\boldsymbol{\alpha}=[\alpha_1,\ldots,\alpha_m]^{\top}$, 表示视图权重, 其和为 1.

虽然引入视图权重可以获得更好的聚类性能, 但是公式 (5) 会得到一个平凡解, 即将最佳视图的权值分配为 1, 其他视图权值为 0. 本文引入了一个新的正则化项 $\alpha\ln\alpha$, 它不仅可以避免平凡解, 而且可以产生更稳定的最优谱嵌入. 因此, 公式 (5) 可以写成如下:

$$\begin{cases}\min\limits_{\boldsymbol{F}^v,\boldsymbol{F}^*}\sum\limits_{v=1}^{m}\alpha_v\|\boldsymbol{F}^v-\boldsymbol{F}^*\|_F^2+\gamma\sum\limits_{v=1}^{m}\alpha_v\ln\alpha_v\\ \text{s.t. } (\boldsymbol{F}^v)^{\top}\boldsymbol{F}^v=\boldsymbol{I}_c,\ (\boldsymbol{F}^*)^{\top}\boldsymbol{F}^*=\boldsymbol{I}_c,\ \alpha_v\geqslant 0,\ \boldsymbol{1}^{\top}\boldsymbol{\alpha}=\boldsymbol{1}\end{cases}\tag{6}$$

其中, $\gamma<0$ 是一个自由参数, 控制不同视图权重对融合过程的影响, 以探索更好的统一表示.

在得到联合谱嵌入矩阵 $\boldsymbol{F}^*$ 后, 现有方法通常将 $\boldsymbol{F}^*$ 输入到 K-means 聚类算法中, 得到最终的聚类结果. 然而, 这种策略将融合过程和最终的聚类过程相分离, 从而产生了一个次优解. 为了简化模型且提高计算效率, 本文引入了一个离散聚类指示矩阵 $\boldsymbol{G}\in\mathbb{R}^{n\times c}$ 来直接指示样本类别, 从而将联合谱嵌入矩阵的优化过程与聚类过程融合在一个优化目标中. 即 $\boldsymbol{G}$ 的每一行都包含一个值为 1 的元素, 而其余元素值为 0, 公式如下:

$$\begin{cases}\min\limits_{\boldsymbol{G}}\|\boldsymbol{F}^*-\boldsymbol{G}\|_F^2\\ \text{s.t. } \boldsymbol{G}\in\{0,1\}\end{cases}\tag{7}$$

为了保持各步骤之间的语义相关性, 提出算法结合公式 (4)、公式 (6)、公式 (7) 构成一个统一的多视图聚类模型, 如下:

$$\begin{cases} \min\limits_{\boldsymbol{Z}^v,\boldsymbol{F}^v \atop \boldsymbol{F}^*,\boldsymbol{G},\boldsymbol{\alpha}} \sum\limits_{v=1}^{m} \|\boldsymbol{X}^v-\boldsymbol{A}^v\boldsymbol{Z}^v\|_F^2+\lambda\|\boldsymbol{Z}^v\|_F^2+\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{L}^v\boldsymbol{F}^v)+\alpha_v\|\boldsymbol{F}^v-\boldsymbol{F}^*\|_F^2+\|\boldsymbol{F}^*-\boldsymbol{G}\|_F^2+\gamma\sum\limits_{v=1}^{m}\alpha_v\ln\alpha_v \\ \text{s.t. } 0\leqslant Z_{ij}^v\leqslant 1,(\boldsymbol{Z}^v)^\top\boldsymbol{1}=\boldsymbol{1},rank(\boldsymbol{L}^v)=n-c,(\boldsymbol{F}^v)^\top\boldsymbol{F}^v=\boldsymbol{I}_c,(\boldsymbol{F}^*)^\top\boldsymbol{F}^*=\boldsymbol{I}_c,\alpha_v\geqslant 0,\boldsymbol{1}^\top\boldsymbol{\alpha}=1,\boldsymbol{G}\in\{0,1\} \end{cases} \tag{8}$$

这样, 在一个统一的框架中同时学习二部图矩阵、联合谱嵌入和聚类指示矩阵. 这 3 个矩阵在统一框架下交替进行优化, 直到目标函数收敛, 最终可以获得良好的聚类结果.

2.3 优化与算法流程

公式 (8) 中的约束问题难以直接解决. 在本节中提供了一种求解该模型的快速优化策略, 将优化问题分解成小规模子问题, 并通过迭代步骤高效解决.

2.3.1 更新二部图矩阵 $\boldsymbol{Z}^v$

固定 $\boldsymbol{F}^v,\boldsymbol{F}^*,\boldsymbol{G},\boldsymbol{\alpha}$, 对 $\boldsymbol{Z}^v$ 的优化可以转化为求解:

$$\begin{cases} \min\limits_{\boldsymbol{Z}^v} \sum\limits_{v=1}^{m} \|\boldsymbol{X}^v-\boldsymbol{A}^v\boldsymbol{Z}^v\|_F^2+\lambda\|\boldsymbol{Z}^v\|_F^2+\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{L}^v\boldsymbol{F}^v) \\ \text{s.t. } 0\leqslant Z_{ij}^v\leqslant 1,\ (\boldsymbol{Z}^v)^\top\boldsymbol{1}=\boldsymbol{1} \end{cases} \tag{9}$$

本文采用两步近似的方法对 $\boldsymbol{Z}^v$ 进行优化. 首先得到无任何约束的闭式解, 然后用对 $\boldsymbol{Z}^v$ 的约束来逼近最优解. 由于每个视图都是独立的, 因此每个 $\boldsymbol{Z}^v$ 都可以单独求解. 在第 1 步中, 忽略上标和约束条件, 公式 (9) 得到:

$$\min_{\boldsymbol{Z}}\|\boldsymbol{X}-\boldsymbol{A}\boldsymbol{Z}\|_F^2+\lambda\|\boldsymbol{Z}\|_F^2+\mathrm{tr}(\boldsymbol{F}^\top\boldsymbol{L}\boldsymbol{F}) \tag{10}$$

已知, $\mathrm{tr}(\boldsymbol{F}^\top\boldsymbol{L}\boldsymbol{F})=\sum_{i,j=1}^{n}\frac{1}{2}\boldsymbol{S}_{i,j}\left\|\boldsymbol{F}_{i,:}-\boldsymbol{F}_{j,:}\right\|^2$, 令 $\boldsymbol{Q}_{i,j}=\|\boldsymbol{F}_{i,:}-\boldsymbol{F}_{j,:}\|$, 则通过设置公式 (10) 对 $\boldsymbol{Z}$ 的导数为 0, 可以得到以下形式的封闭解:

$$\hat{\boldsymbol{Z}}=(\boldsymbol{A}^\top\boldsymbol{A}+\lambda\boldsymbol{I}+\tfrac{1}{2}\boldsymbol{Q}^\top)^{-1}\boldsymbol{A}^\top\boldsymbol{X} \tag{11}$$

在第 2 步中, 将 $\hat{\boldsymbol{Z}}$ 投影到一个受约束的空间中, 对于 $\boldsymbol{Z}^v$ 的每一行, 可以得到:

$$\min_{\boldsymbol{Z}_{i,:}^v\geqslant 0,(\boldsymbol{Z}_{i,:}^v)^\top\boldsymbol{1}=\boldsymbol{1}}\|\boldsymbol{Z}_{i,:}^v-\hat{\boldsymbol{Z}}_{i,:}^v\|_F^2 \tag{12}$$

然后可以得到公式 (12) 的拉格朗日函数:

$$\mathcal{L}(\boldsymbol{Z}_{i,:}^v,\beta,\varphi)=\|\boldsymbol{Z}_{i,:}^v-\hat{\boldsymbol{Z}}_{i,:}^v\|_F^2-\beta_i(\boldsymbol{Z}_{i,:}^{v\top}\boldsymbol{1}-1)-\varphi_i^\top\boldsymbol{Z}_{i,:}^v \tag{13}$$

其中, β_i 和 $\varphi_i^\top$ 是拉格朗日乘数. 然后根据 KKT 条件可以得到:

$$\boldsymbol{Z}_{i,:}^v=\max(\hat{\boldsymbol{Z}}_{i,:}^v+\eta_i\boldsymbol{1},0),\ \eta_i=\frac{1-\boldsymbol{1}^\top\hat{\boldsymbol{Z}}_{i,:}^v}{n} \tag{14}$$

2.3.2 更新各视图谱嵌入矩阵 $\boldsymbol{F}^v$

通过固定其他变量并去除与 $\boldsymbol{F}^v$ 无关的项, 可以将 $\boldsymbol{F}^v$ 的优化转化为解决以下问题:

$$\begin{cases} \min\limits_{\boldsymbol{F}^v}\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{L}^v\boldsymbol{F}^v)-2\alpha_v\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{F}^*) \\ \text{s.t. } (\boldsymbol{F}^v)^\top\boldsymbol{F}^v=\boldsymbol{I}_c \end{cases} \tag{15}$$

公式 (15) 可以放宽为以下形式, 其中 $\lambda_{\max}$ 为 $\boldsymbol{L}^v$ 的最大特征值.

$$\begin{cases} \max\limits_{\boldsymbol{F}^v}\mathrm{tr}((\boldsymbol{F}^v)^\top(\lambda_{\max}\boldsymbol{I}-\boldsymbol{L}^v)\boldsymbol{F}^v)+2\alpha_v\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{F}^*) \\ \text{s.t. } (\boldsymbol{F}^v)^\top\boldsymbol{F}^v=\boldsymbol{I}_c \end{cases} \tag{16}$$

如果 $\boldsymbol{A},\boldsymbol{B}$ 是半正定矩阵, 那么 $f(\boldsymbol{W})=\mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{A}\boldsymbol{W}\boldsymbol{B})+\mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{C})$ 是一个凸函数. 问题可以通过迭代优化 $\max\limits_{\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}}\mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{M})$ 来解决, 其中 $\boldsymbol{M}=f'(\boldsymbol{W})=2\boldsymbol{A}\boldsymbol{W}\boldsymbol{B}+\boldsymbol{C}$[37]. 因此, 公式 (16) 可以通过算法 1 求解.

算法 1. 更新各视图嵌入矩阵 $\boldsymbol{F}^v$.

输入: $\lambda_{\max},\boldsymbol{L}^v,\alpha_v,\boldsymbol{F}^*$;

输出: 各视图嵌入矩阵 $\boldsymbol{F}^v$.

1. While not converged do
2. $\boldsymbol{M}=2(\lambda_{\max}\boldsymbol{I}-\boldsymbol{L}^v)+2\alpha_v\boldsymbol{F}^*$.
3. 对 $\boldsymbol{M}$ 进行 SVD, $\boldsymbol{M}=\boldsymbol{U}_1\Sigma\boldsymbol{V}_1^\top$.
4. $\boldsymbol{F}^v=\boldsymbol{U}_1\boldsymbol{V}_1^\top$.
5. end while

2.3.3 更新联合谱嵌入矩阵 $\boldsymbol{F}^*$

固定 $\boldsymbol{Z}^v,\boldsymbol{F}^v,\boldsymbol{G},\boldsymbol{\alpha}$, 对 $\boldsymbol{F}^*$ 的优化可以转化为求解:

$$\begin{cases}\min\limits_{\boldsymbol{F}^*}\sum\limits_{v=1}^{m}\alpha_v\|\boldsymbol{F}^v-\boldsymbol{F}^*\|_F^2+\|\boldsymbol{F}^*-\boldsymbol{G}\|_F^2\\ \text{s.t. }(\boldsymbol{F}^*)^\top\boldsymbol{F}^*=\boldsymbol{I}_c\end{cases}\tag{17}$$

公式 (17) 的问题可以转换为以下优化形式:

$$\begin{cases}\min\limits_{\boldsymbol{F}^*}\sum\limits_{v=1}^{m}-2\alpha_v\mathrm{tr}((\boldsymbol{F}^v)^\top\boldsymbol{F}^*)-2\mathrm{tr}((\boldsymbol{F}^*)^\top\boldsymbol{G})\\ \text{s.t. }(\boldsymbol{F}^*)^\top\boldsymbol{F}^*=\boldsymbol{I}_c\end{cases}\tag{18}$$

公式 (18) 通过简单的转换可以得到:

$$\begin{cases}\max\limits_{\boldsymbol{F}^*}\mathrm{tr}((\boldsymbol{F}^*)^\top\boldsymbol{N})\\ \text{s.t. }(\boldsymbol{F}^*)^\top\boldsymbol{F}^*=\boldsymbol{I}_c\end{cases}\tag{19}$$

其中, $\boldsymbol{N}=2\Sigma_{v=1}^{m}\alpha_v\boldsymbol{F}^v+2\boldsymbol{G}$.

当受约束问题 $\max\limits_{\boldsymbol{A}}\mathrm{tr}((\boldsymbol{A})^\top\boldsymbol{B})$ s.t. $(\boldsymbol{A})^\top\boldsymbol{A}=\boldsymbol{I}$ 已知 $\boldsymbol{B}=\boldsymbol{U}\Sigma\boldsymbol{V}^\top$, 则有闭式解 $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{V}^\top$. 因此, 可以得到公式 (19) 的闭式解 $\boldsymbol{F}^*=\boldsymbol{U}_2\boldsymbol{V}_2^\top$, 其中 $\boldsymbol{U}_2$ 和 $\boldsymbol{V}_2$ 分别为矩阵 $\boldsymbol{N}$ 的左奇异矩阵和右奇异矩阵.

2.3.4 更新视图权重 α_v

固定 $\boldsymbol{Z}^v,\boldsymbol{F}^v,\boldsymbol{F}^*,\boldsymbol{G}$, 令 $h_v=\|\boldsymbol{F}^v-\boldsymbol{F}^*\|_F^2$, 对 α_v 的优化可以简化为:

$$\min_{\alpha_v\geqslant 0,\,\mathbf{1}^\top\boldsymbol{\alpha}=\mathbf{1}}\sum_{v=1}^{m}\alpha_v h_v+\gamma\sum_{v=1}^{m}\alpha_v\ln\alpha_v\tag{20}$$

为了解决上述问题, 公式 (20) 的拉格朗日函数为:

$$\mathcal{L}(\alpha_v,\eta)=\sum_{v=1}^{m}\alpha_v h_v+\gamma\sum_{v=1}^{m}\alpha_v\ln\alpha_v-\eta\left(\sum_{v=1}^{m}\alpha_v-1\right)\tag{21}$$

其中, η 是拉格朗日乘数, 令 $\mathcal{L}$ 对 α_v 求偏导为 0, 即:

$$\frac{\partial\mathcal{L}}{\partial\alpha_v}=h_v+\gamma\,(\ln\alpha_v+1)-\eta=0\tag{22}$$

得到 α_v 的解为:

$$\alpha_v=\exp\left(\frac{\eta-h_v}{\gamma}-1\right)\tag{23}$$

将得到的 α_v 添加约束 $\sum_{v=1}^{m}\alpha_v=1$, 得到:

$$\alpha_v=\frac{\exp(-h_v/\gamma)}{\sum_{v=1}^{m}\exp(-h_v/\gamma)}\tag{24}$$

2.3.5 更新聚类指示矩阵 $\boldsymbol{G}$

固定其他变量, 目标函数可以转换为以下形式:

$$\begin{cases}\min\|\boldsymbol{F}^*-\boldsymbol{G}\|_F^2\\ \text{s.t. }\boldsymbol{G}\in\{0,1\}\end{cases}\tag{25}$$

很容易得到$\boldsymbol{G}$:

$$g_{ij}=\begin{cases}1, & \text{if } j=\arg\max_{j^*=1,2,\ldots,c} f_{ij^*}\\ 0, & \text{otherwise}\end{cases} \tag{26}$$

迭代更新所有变量, 直到目标函数公式 (8) 收敛, 样本类别可以直接从$\boldsymbol{G}$中获得, 而不需要任何后期的操作. 整个过程的详细步骤见算法 2.

算法 2. 基于二部图的联合谱嵌入多视图聚类算法.

输入: 多视图数据集$\{\boldsymbol{X}^v\}_{v=1}^m$, 簇数$c$, 参数$\lambda$, 参数$\gamma<0$, 锚点数$t$;
输出: 聚类指示矩阵, 直接获得聚类结果.

1. 利用 K-means++聚类算法初始化锚点矩阵$\boldsymbol{A}^v$;
2. 利用公式 (1) 初始化二部图矩阵$\tilde{\boldsymbol{Z}}^v$;
3. 用拉普拉斯矩阵的特征向量初始化嵌入矩阵$\boldsymbol{F}^v$;
4. 随机初始化正交矩阵$\boldsymbol{F}^*$, 利用公式 (26) 初始化聚类指示矩阵$\boldsymbol{G}$;
5. 初始化视图权重$\alpha_v=1/m$;
6. While not converged do
7. 通过公式 (14) 更新二部图矩阵$\boldsymbol{Z}^v$;
8. 通过算法 1 更新各视图嵌入矩阵$\boldsymbol{F}^v$;
9. 通过解决公式 (19) 更新联合嵌入矩阵$\boldsymbol{F}^*$;
10. 通过公式 (26) 更新聚类指示矩阵$\boldsymbol{G}$;
11. 通过公式 (24) 更新视图权重α_v;
12. end while

2.4 时间复杂度分析

根据算法 2 描述, 本文算法的时间复杂度主要包括矩阵初始化和优化过程. 基于 K-means 聚类算法选择锚点并构造二部图, 其复杂度是O(ntd), 其中t为锚点的数量且远小于n, d为数据特征个数之和. 在优化部分, $\boldsymbol{Z}^v$的更新需要矩阵逆运算, 计算复杂度为O(mn^3). 更新$\boldsymbol{F}^*$的复杂度为O(n^3), 而更新$\boldsymbol{F}^v$的复杂度为O(t_1mnc^2), 其中t_1为算法 1 的迭代次数. 特别地, 由于变量$\boldsymbol{G}$被约束为一个聚类指示矩阵, 可以在优化结束时直接获得所有样本的类别而不需要执行额外操作. 总的来说, 算法整体复杂度为O(Tmn^3), 其中T表示算法 2 总的迭代次数.

3 实验分析

在本节中, 在 6 个广泛使用的基准数据集上评估了所提算法的聚类性能. 将所提算法与 2 种单视图聚类算法和 5 种先进的多视图聚类算法在 3 个评价指标上进行了实验比较分析.

3.1 数据集

本实验选取 6 个真实多视图数据集作为实验数据, 数据集的详细情况如表 2 所示.

3sources: 该数据集包含 BBC, Reuters 和 The Guardianz 这 3 大新闻资源的 169 篇文章, 这 169 篇文章被分为商业、娱乐、政治、健康、体育和科技 6 个类.

bbcsport: 该数据集从 5 个主题领域的原始体育新闻数据库中选择了 116 个样本. 每个文档被分成 4 个相关的部分作为特征, 每个部分由连续的文本段落组成.

webkb: 该数据集由高校计算机科学系搜集的网页组成, 有 203 个网页和 4 个类. 每个网页都由页面的内容、超链接的锚文本和其标题中的文本来表述.

Cal7: 该数据集在 101 个类别的图像数据集中选择了 7 个广泛使用的类别, 包含 1474 个样本, 提取 6 个常用

的图像特征以形成不用的视图特征.

Handwritten[38]: 该数据集有 6 个视图, 每个视图包含 2000 个来自 “0–9”的数字图像.

MSRC-V1[39]: 该数据集由来自 210 个图像的 7 个类组成, 分别用 5 种不同的特征提取方法将图像构成不同的视图.

表 2　数据集的统计信息

统计项	3sources	bbcsport	webkb	Cal7	Handwritten	MSRC-V1
#d1	3560	1991	1703	48	240	24
#d2	3631	2063	230	40	76	576
#d3	3068	2113	230	254	216	512
#d4	—	2158	—	1984	47	256
#d5	—	—	—	512	64	254
#d6	—	—	—	928	6	—
#view	3	4	3	6	6	5
#instances	169	116	203	1474	2000	210
#class	6	5	4	7	10	7

3.2 评价指标

为了评估聚类性能, 本文选择了 3 种类型的外部评价指标: *ACC* (准确性)、*NMI* (归一化互信息)、*F*-score, 下面将详细进行介绍.

数据点 x_i 的聚类结果和真实标签分别为 q_i 、 p_i, 则准确性 (*ACC*) 定义如下:

$$ACC = \frac{\sum_{v=1}^{m} \delta(p_i, map(q_i))}{n} \tag{27}$$

其中, $\delta(x,y)$ 函数表示如果 $x=y$, 则 $\delta(x,y)=1$, 否则 $\delta(x,y)=0$. $map(q_i)$ 是映射函数, 使用 KM 算法排列聚类标签以匹配真实的标签.

归一化互信息 (*NMI*) 定义如下:

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{ij} \log \frac{n \times n_{ij}}{n_i \times n_j}}{\sqrt{\sum_{i=1}^{c} n_i \log \frac{n_i}{n} + \sum_{j=1}^{c} n_j \log \frac{n_j}{n}}} \tag{28}$$

其中, n 表示样本个数, n_i 表示类 i 中的数据点数, n_j 表示分配给类簇 j 的数据点数, n_{ij} 表示类 i 和类 j 共享的数据点数, *NMI* 反映了聚类结果与真实标签之间的一致性.

F-score 是信息检索中使用的精度 (*P* 值) 和查全率 (*R* 值) 的同等加权组合. 给定一个聚类方案, *F*-score 被定义为:

$$F\text{-score} = \sum_{i=1}^{c} \frac{n_i}{n} \max_j (F_{ij}) \tag{29}$$

其中, $F_{ij} = \frac{1 \times P_{ij} \times R_{ij}}{P_{ij} + R_{ij}}$, $P_{ij} = \frac{n_{ij}}{n_j}$, $R_{ij} = \frac{n_{ij}}{n_i}$.

上述 3 个评价指标的范围均为 0–1, 值越大表示聚类结果越好.

3.3 比较的方法及参数设置

在实验过程中, 本文将和以下算法进行聚类性能方面的比较.

(1) 单视图聚类算法: 单视图聚类算法 K-means 和单视图谱聚类 (SC) 分别运行在数据集的每个视图上, 并记录这些视图中最好的聚类结果.

(2) 基于二部图的多视图聚类算法: EOMSC-CA 算法[40]对锚点选择和子空间构造进行联合优化以提高聚类性

能; EMKMC 算法[41]首先为每个视图构造锚点图, 然后使用改进的 K-means 进行集成, 并设计了两种算法来求解模型; LMVSC 算法[42]在原始数据点和生成的锚点之间构造一个更小的图, 并设计了一种新的集成方法来合并这些图.

(3) 基于完全图的多视图聚类算法: GMC 算法[32]关键新颖之处在于其以相互强化的方式学习每个视图的图矩阵和统一图矩阵, 并用一种新的多视图融合技术自动地对每个视图矩阵进行加权; COMVSC 算法[29]集成划分级信息, 有助于消除数据之间的噪声.

根据已有论文实验分析中的建议, 对比较方法的参数进行了设置. 其中, 本文提出的方法与其他对比方法的共同参数为簇数 c, 为了比较的公平性, 将所有算法的 c 值均设置为真实的簇数. 其他参数设置如下, 对于单视图聚类算法 K-means 和 SC, 本文将该算法分别运行在不同视图的数据中并记录最优结果. EOMSC-CA 方法需要确定锚点数和锚点矩阵的维度, 根据建议, 实验通过遍历 $[c, 2c,\ldots, 7c]$ 记录最佳结果. EMKMC 方法根据论文设置自由参数 $1<\gamma<2$, 并根据论文设置锚点集记录最佳结果. LMVSC 在{c, 50, 100}的集合中搜索锚点数, 在集合{0.001, 0.01, 0.1, 1, 10}中选择参数 α. GMC 根据论文设置近邻数 k 为 15, 初始参数 λ 为 1. COMVSC 有两个参数 λ 和 γ, 根据文献 [29] λ 在$\{2^3, 2^5,\ldots, 2^{13}\}$的集合中进行调优, 而 γ 在{1.3, 1.5,…, 2.7}的集合中进行调优, 增量步骤为 0.2. 本文所提方法为了学习最优图, 引入了一个自由参数 γ 来约束每个视图的权重, 并设置在{−1E0, −1E1,…, −1E7}的集合中进行调优, 参数 λ 在$\{2^3, 2^5,\ldots, 2^{13}\}$的集合中进行调优, 锚点数 t 在$\{c+1, c+3, c+5, c+7, c+9, 50, 100\}$的集合中进行调优.

3.4 实验结果

3.4.1 真实数据集聚类结果

在真实多视图数据集上, 本文提出算法及比较的其他算法在 3 个经典的聚类评价标准 (*ACC*、*NMI* 和 *F*-score) 下的聚类结果如表 3–表 5 所示, 其中最优以粗体显示, 次优用下划线显示.

从表 3–表 5 可以看出, 本文提出的算法除了在 Handwritten 数据集上得到次优解, 其他所有数据集上都有最好的性能. 结果表明, 提出的方法是一种有效的多视图聚类方法.

表 3 不同聚类算法在多视图数据集上的 *ACC* 值

Methods	3sources	bbcsport	webkb	Cal7	Handwritten	MSRC-V1
K-means	0.4961	0.2772	0.4276	0.3726	0.5038	0.5911
SC	0.4142	0.4224	0.4631	0.4403	0.5485	0.5429
EOMSC-CA	0.5728	0.5001	0.6108	<u>0.8351</u>	0.9340	0.6714
EMKMC	0.6568	0.6662	0.5299	0.5160	0.5727	0.6337
LMVSC	0.4994	0.5060	0.4634	0.7266	0.9165	0.7204
GMC	<u>0.6923</u>	0.5603	0.7635	0.6920	0.8820	0.7476
COMVSC	0.6864	<u>0.6983</u>	<u>0.7882</u>	0.8134	**0.9450**	<u>0.7714</u>
本文算法	**0.8935**	**0.8534**	**0.8079**	**0.8365**	<u>0.9375</u>	**0.8619**

表 4 不同聚类算法在多视图数据集上的 *NMI* 值

Methods	3sources	bbcsport	webkb	Cal7	Handwritten	MSRC-V1
K-means	0.5356	0.4336	0.5301	0.4710	0.6108	0.6505
SC	0.4414	0.2895	0.3717	0.4868	0.7067	0.4853
EOMSC-CA	0.5194	0.4747	0.4488	0.5219	0.7767	0.5608
EMKMC	<u>0.6737</u>	<u>0.7438</u>	<u>0.7272</u>	0.5440	0.6183	0.6851
LMVSC	0.5754	0.5443	0.6770	0.5193	0.8443	<u>0.7596</u>
GMC	0.6216	0.4771	0.4164	<u>0.6595</u>	0.8041	0.7509
COMVSC	0.5301	0.5346	0.4856	0.5311	<u>0.8925</u>	0.7040
本文算法	**0.7911**	**0.7692**	**0.7667**	**0.6804**	**0.9021**	**0.7614**

表 5　不同聚类算法在多视图数据集上的 *F*-score 值

Methods	3sources	bbcsport	webkb	Cal7	Handwritten	MSRC-V1
K-means	0.4556	0.4569	0.4828	0.4654	0.6320	0.6952
SC	0.4866	0.3909	0.5143	0.4278	0.6808	0.4568
EOMSC-CA	0.5962	0.5776	0.5515	0.7967	0.8738	0.5475
EMKMC	0.7160	0.7017	0.7438	0.8130	0.7125	0.7095
LMVSC	0.5207	0.6034	0.6946	0.6947	0.8540	0.7143
GMC	0.6047	0.4439	0.6933	0.7217	0.8653	0.6968
COMVSC	0.6788	0.5322	0.7251	0.7728	**0.8934**	0.6776
本文算法	**0.8533**	**0.7911**	**0.7655**	**0.8856**	0.8813	**0.7520**

与基于原始构图的多视图聚类算法 GMC 和 COMVSC 相比, 所提算法显示了更高的聚类性能, 分析原因在于所提算法对数据运行 K-means 聚类算法, 选出簇中心作为锚点进而产生二部图, 且结合其他矩阵信息进行动态更新, 相较 $n\times n$ 的完全图可以更好地捕获视图的划分结构信息.

与基于二部图的多视图聚类算法 EOMSC-CA、EMKMC 和 LMVSC 相比, 聚类性能有很高提升, 分析原因在于本文算法进行谱嵌入融合, 对比相似矩阵融合可以提供更多的鉴别信息和更少的噪声, 且将联合谱嵌入矩阵的优化过程与聚类过程融合在一个优化目标中, 避免了次优解. EMKMC 在 3sources、bbcsport 和 webkb 数据集上均获得了次优解, 分析原因在于 EMKMC 为同一数据集的不同视图设置不同的锚点个数, 这更准确地结合了不同视图数据, 但由于算法需要手动输入锚点集, 因此导致结果不稳定.

3.4.2　运行时间分析

在相同的计算环境中对每种方法进行了实验, 并记录了每种方法在每个数据集上的运行时间, 统计结果如图 2 所示. 可以看出, 单视图方法通常比多视图方法运行得更快, 主要因为多视图聚类方法需要同时处理多个视图.

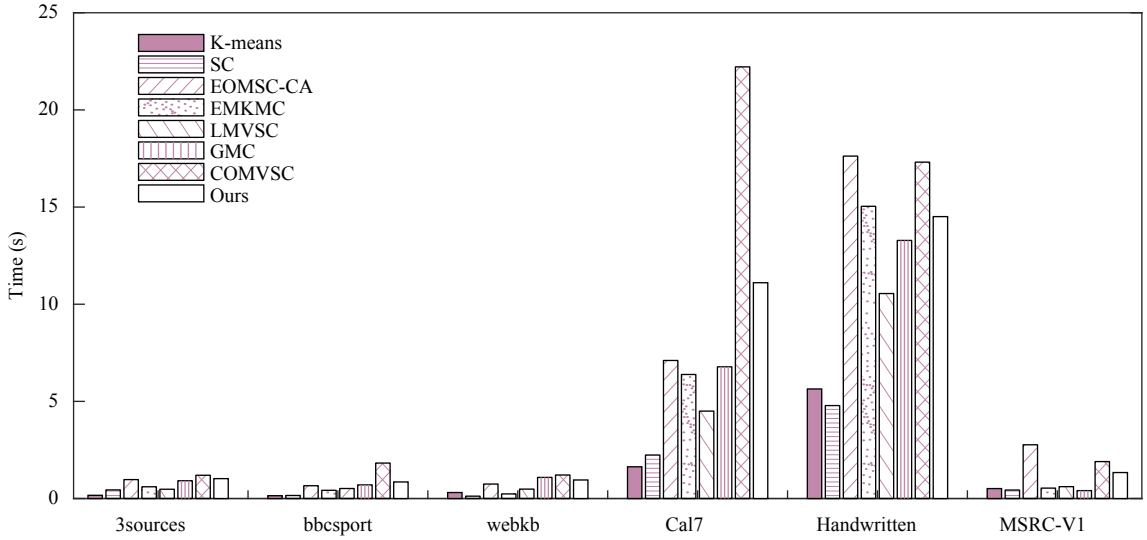


图 2　不同算法在 6 个数据集上的运行时间比较

当数据集较小时, 所有算法运行时间较短, 而当数据集较大时, 所有算法的运行时间都较长. 在比较的算法中, 最快和最适合用于大规模数据集的是 LMVSC 和 GMC, 时间复杂度分别为 $O(mnt^2)$ 和 $O(Tmn^2)$, 其中 m 和 t 分别表示视图个数和锚点个数, T 表示算法迭代次数. 这些方法旨在解决大规模的问题, 它们更关注效率而不是有效性. 虽然所提出的算法比专门为大规模场景设计的算法要慢, 但与它们相比, 本文算法可以在聚类性能方面实现显著的改进. COMVSC 时间复杂度为 $O(Tmn^3)$, 在 COMVSC 中利用了全样本图而不是锚点图, 这使得构造图部分的复杂度高于其他算法. 算法 EMKMC 和 EOMSC-CA 都为线性时间复杂度, 但是 EMKMC 需要输入每个视图的锚点数, 导致结果不稳定, 且两个算法效率在某些数据集上远不及本文所提算法. 可以看出, 所提算法兼顾了效率和

时间, 是一个比较有优越性的多视图聚类算法.

3.4.3 视图权重分析

所提方法为了学习最优图, 引入了一个自由参数γ来约束每个视图的权重. 以数据集 3sources 为例, 设置参数λ为2^7, 锚点数t为$2c$, 图 3(a)–(c) 分别为γ等于−1E0 ($\gamma\to 0$)、−1E5、−1E7 ($\gamma\to-\infty$) 时的视图权重, 可以看出, 当$\gamma\to 0$时, 将得到平凡解. 相反当$\gamma\to-\infty$时, 将得到相等的权重. 实验证明当γ处于中值即视图权重发挥作用时聚类结果最好. 因此, 一个合适的γ可以带来更好的聚类性能.

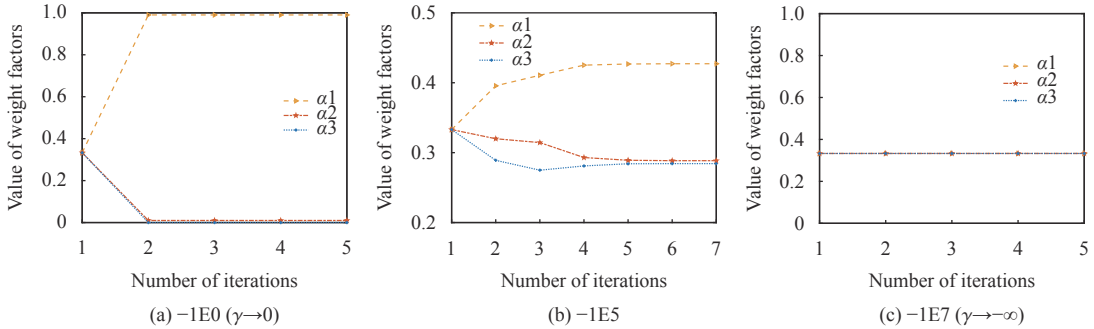


图 3 参数γ对 3sources 数据集视图权重的影响

为了进一步评估提出算法的聚类性能, 本文对每个多视图数据集进行了以下操作. 用本文算法对数据集的不同的视图进行聚类, 并记录性能最佳的聚类结果, 由 BestView 表示. 将所有视图的特征合并到一个视图中, 然后用本文方法对合并后的单视图数据进行聚类, 用 ConView 表示. 图 4 为它们在所有数据集上的比较结果.

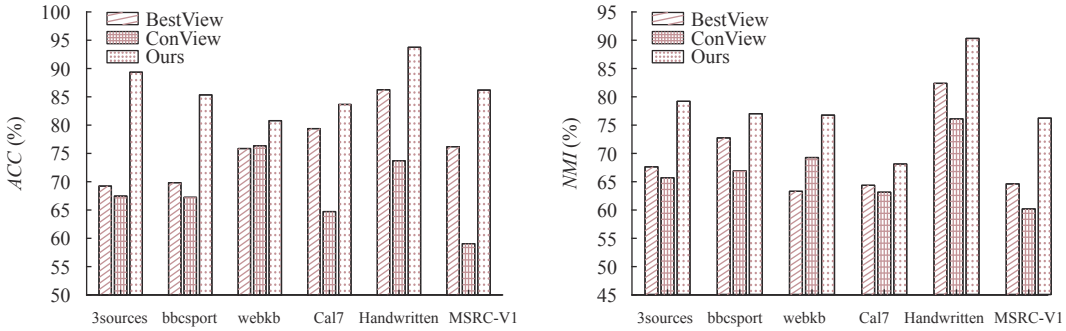


图 4 BestView、ConView 和本文方法在 6 个数据集上的性能比较

由图 4 可以看出, 所提算法总是在所有数据集上都具有最好的性能. 另外, 在大多数情况下, BestView 优于 ConView. 这表明, 简单地连接所有视图的特性并不能获得更好的性能. 而 webkb 数据集上 ConView 优于 BestView, 分析可能是因为该数据集的每个视图的聚类能力相似. 因此正如本文中所做那样, 多视图聚类方法需要使用视图加权技术, 才能获得更好的聚类性能.

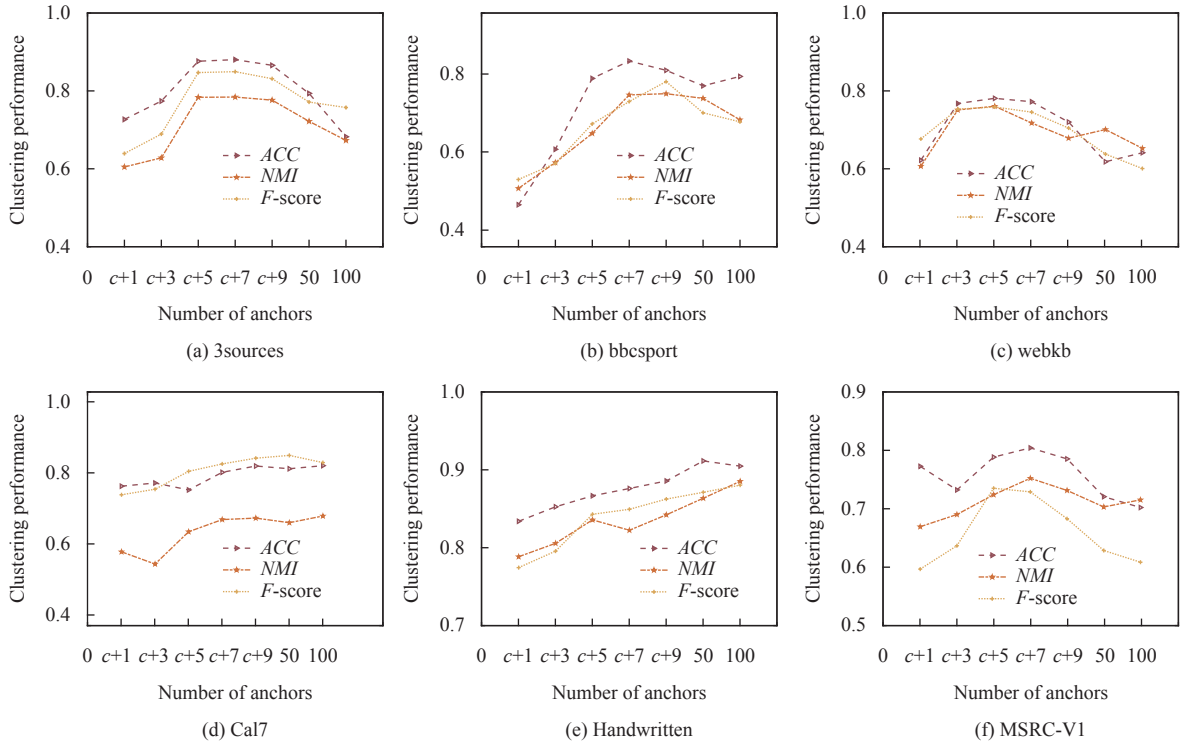
3.4.4 参数分析

本文还对参数进行了敏感性分析, 所提出的方法有 3 个参数t、λ和γ. 锚点数t在$\{c+1, c+3, c+5, c+7, c+9, 50, 100\}$的集合中进行调优, 参数$\lambda$在$\{2^3, 2^5, \ldots, 2^{13}\}$的集合中进行调优, 参数$\gamma$在$\{-1E0, -1E1, \ldots, -1E7\}$的集合中进行调优. 图 5, 图 6 显示了参数对聚类性能的影响.

图 5 显示了锚点数t对不同数据集聚类 *ACC*、*NMI* 和 *F*-score 的影响. 可以看出, 当锚点数量较小时, 随着锚点数量的增加, 聚类结果总体上呈上升趋势. 当锚点的数量增加到一定的值时, 随着锚点数量的增加, 过多的锚点会使代表性降低, 聚类性能反而下降. 但可以看出, Handwritten 数据集较其他数据集大, 对锚点数的敏感性不明显.

在图 6 中可视化了参数λ和γ对不同数据集的 *F*-score 的影响. 可以看出, 提出的算法在数据集的大范围内保持了相对稳定的聚类性能. 特别是, 数据集 3sources、bbcsport 和 webkb 在λ取其内容的中值时表现更好, 当λ取

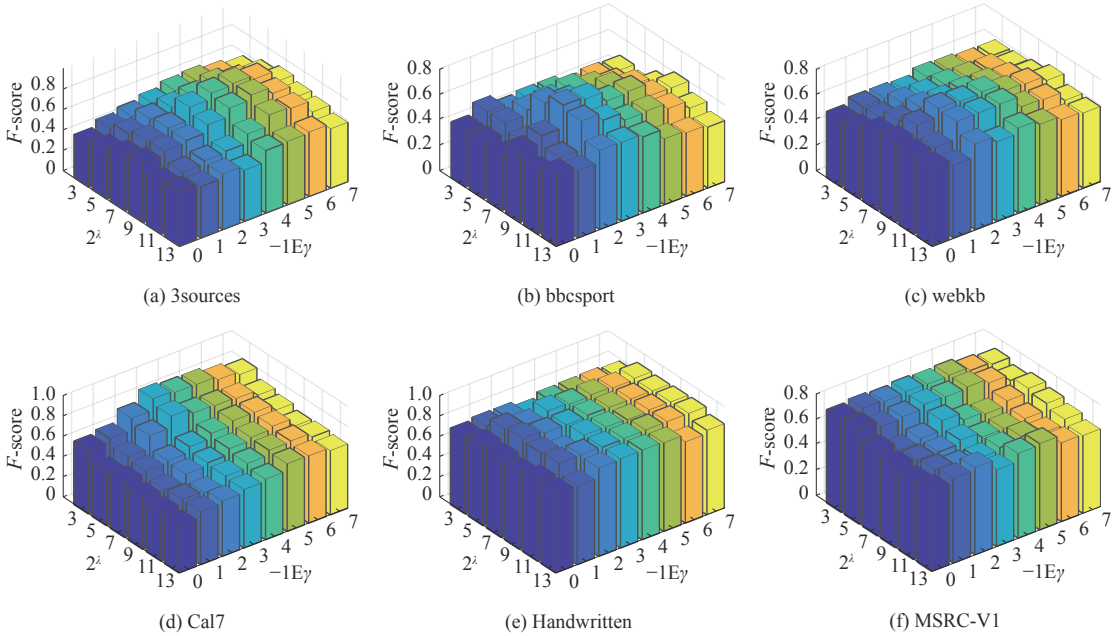
其范围的较小值时, 数据集 Cal7 和 MSRC-V1 表现更好. Handwritten 在大多数参数下都能保持良好的性能. 本文为了充分利用多视图数据, 添加参数γ控制视图权重, 提高聚类有效性. 因此, 为了获得最优的聚类有效性和效率, 需要选择合理的λ和γ.



(a) 3sources (b) bbcsport (c) webkb

(d) Cal7 (e) Handwritten (f) MSRC-V1

图 5 不同数据集对锚点数的敏感度分析



(a) 3sources (b) bbcsport (c) webkb

(d) Cal7 (e) Handwritten (f) MSRC-V1

图 6 使用 *F*-score 评估参数 γ 和 λ 的灵敏度

除了验证算法对参数的敏感性, 本文还进一步验证了算法的鲁棒性. 本文对 3sources、Cal7、MSRC-V1 这 3 个数据集各个特征值分别添加了均值为 0, 方差为 0.01 的高斯噪声, 形成噪声数据. 本文算法和对比算法在噪声数据集上的实验结果如图 7 所示, 图中 noisepre 和 noiseaft 的具体数值为与表 4 相比加噪声前后不同算法实验结果的 *NMI* 值. 由图 7 可以看出, 不同算法在添加噪声数据集上的聚类结果的有效性均出现了一定程度的下降, 相比而言, 本文所提算法的下降值更小, 更具鲁棒性.

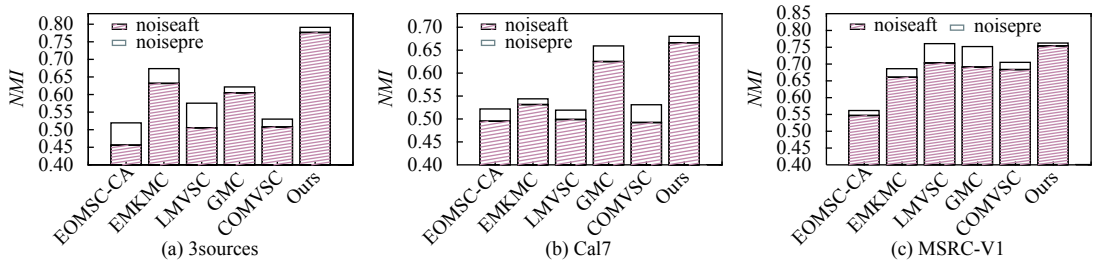


图 7 噪声数据集上不同算法的鲁棒性分析

3.4.5 收敛性分析

关于收敛性分析, 本文主要进行了收敛次数的对比. 图 8 为本文算法在 6 个数据集上的收敛曲线. 对于每个图, X 轴和 Y 轴分别表示迭代次数和目标函数值. 可以看出, 该算法的收敛速度非常快, 通常需要 10 次迭代就可以实现算法收敛. 根据已发表论文中对比算法的收敛次数, 其中 EOMSC-CA 在迭代 15 次左右得到了收敛, EMKMC 迭代 10 次得到了收敛, GMC 在迭代 10 次内收敛, COMVSC 在数据集 YaleA 上迭代 70 次得到收敛. 因此, 从收敛次数上分析, 本文算法可以达到平均水平.

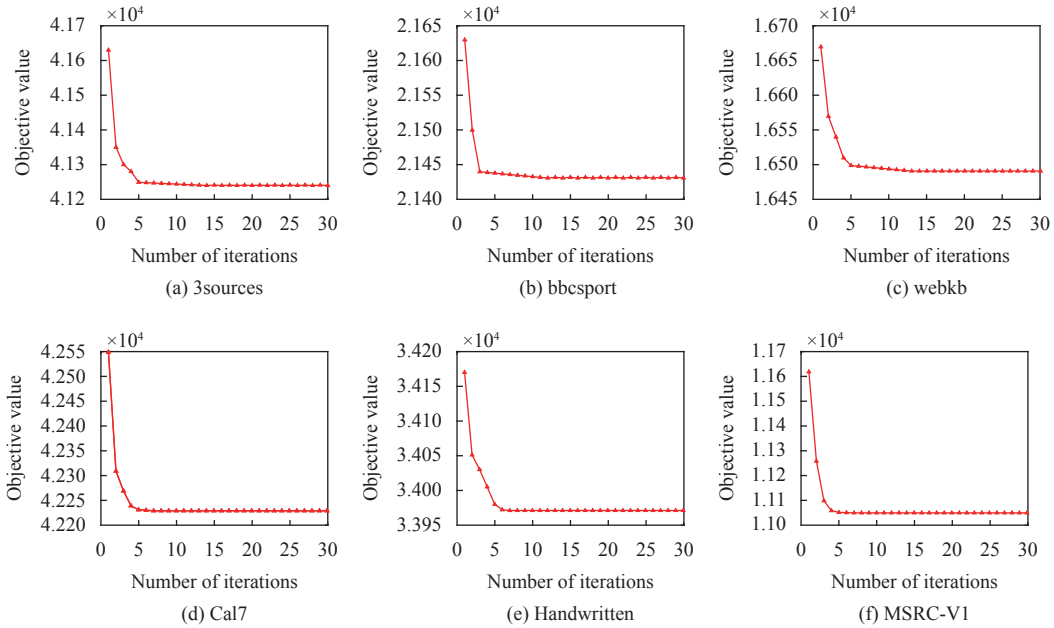


图 8 在不同数据集上的算法收敛性分析

4 总 结

近年来, 如何充分利用多视图数据进行聚类分析成为一个重要的研究内容. 本文提出了一种基于二部图的联

合谱嵌入多视图聚类算法. 该算法基于多视图子空间聚类的思想构造二部图进而产生相似图, 接着利用相似图的谱嵌入矩阵进行图融合. 同时考虑每个视图的重要性进行权重约束, 引入聚类指示矩阵得到最终聚类结果. 通过大量的实验, 对提出算法的有效性和鲁棒性进行了验证. 然而, 本文提出的算法存在时间复杂度较高的问题, 在未来的工作中将考虑如何提高算法的计算效率来处理大规模多视图数据.

References:

[1] Berlingerio M, Pinelli F, Calabrese F. ABACUS: Frequent pattern mining-based community discovery in multidimensional networks. Data Mining and Knowledge Discovery, 2013, 27(3): 294–320. [doi: 10.1007/s10618-013-0331-0]

[2] Hu SY, Gu ZY, Wang YF, Zhang XL. An analysis of the clustering effect of a jump risk complex network in the Chinese stock market. Physica A: Statistical Mechanics and Its Applications, 2019, 523: 622–630. [doi: 10.1016/j.physa.2019.01.114]

[3] Omran M, Engelbrecht AP, Salman A. Particle swarm optimization method for image clustering. Int'l Journal of Pattern Recognition and Artificial Intelligence, 2005, 19(3): 297–321. [doi: 10.1142/S0218001405004083]

[4] Liu L, Peng T, Zuo WL, Dai YK. Clustering-based PU active text classification method. Ruan Jian Xue Bao/Journal of Software, 2013, 24(11): 2571–2583 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4467.htm [doi: 10.3724/SP.J.1001.2013.04467]

[5] Nguyen DT, Chen LH, Chan CK. Clustering with multiviewpoint-based similarity measure. IEEE Trans. on Knowledge and data Engineering, 2012, 24(6): 988–1001. [doi: 10.1109/TKDE.2011.86]

[6] Zhao J, Xie XJ, Xu X, Sun SL. Multi-view learning overview: Recent progress and new challenges. Information Fusion, 2017, 38: 43–54. [doi: 10.1016/j.inffus.2017.02.007]

[7] Fu LL, Lin PF, Vasilakos AV, Wang SP. An overview of recent multi-view clustering. Neurocomputing, 2020, 402: 148–161. [doi: 10.1016/j.neucom.2020.02.104]

[8] Wang H, Yang Y, Liu B, Fujita H. A study of graph-based system for multi-view clustering. Knowledge-based Systems, 2019, 163: 1009–1019. [doi: 10.1016/j.knosys.2018.10.022]

[9] Kumar A, Daume III H. A co-training approach for multi-view spectral clustering. In: Proc. of the 28th Int'l Conf. on Machine Learning. Bellevue: Omnipress, 2011. 393–400.

[10] Nie FP, Li J, Li XL. Self-weighted multiview clustering with multiple graphs. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. Melbourne: AAAI Press, 2017. 2564–2570.

[11] Liang JY, Liu XL, Bai L, Cao FY, Wang DH. Incomplete multi-view clustering via local and global co-regularization. Science China Information Sciences, 2022, 65(5): 152105. [doi: 10.1007/s11432-020-3369-8]

[12] Liu XL, Bai L, Zhao XW, Liang JY. Incomplete multi-view clustering algorithm based on multi-order neighborhood fusion. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1354–1372 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6471.htm [doi: 10.13328/j.cnki.jos.006471]

[13] Zhang YP, Zhou J, Deng ZH, Zhong FL, Jiang YZ, Hang WL, Wang ST. Multi-view fuzzy clustering approach based on medoid invariant constraint. Ruan Jian Xue Bao/Journal of Software, 2019, 30(2): 282–301 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5625.htm [doi: 10.13328/j.cnki.jos.005625]

[14] Tang C, Zhu XZ, Liu XW, Li MM, Wang PC, Zhang CQ, Wang LZ. Learning a joint affinity graph for multiview subspace clustering. IEEE Trans. on Multimedia, 2019, 21(7): 1724–1736. [doi: 10.1109/TMM.2018.2889560]

[15] Zhan K, Nie FP, Wang J, Yang Y. Multiview consensus graph clustering. IEEE Trans. on Image Processing, 2019, 28(3): 1261–1270. [doi: 10.1109/TIP.2018.2877335]

[16] Ji GY, Lu GF. One-step incomplete multiview clustering with low-rank tensor graph learning. Information Sciences, 2022, 615: 209–225. [doi: 10.1016/j.ins.2022.10.026]

[17] Liang YW, Huang D, Wang CD. Consistency meets inconsistency: A unified graph learning framework for multi-view clustering. In: Proc. of the 2019 IEEE Int'l Conf. on Data Mining. Beijing: IEEE, 2019. 1204–1209. [doi: 10.1109/ICDM.2019.00148]

[18] Huang SD, Kang Z, Tsang IW, Xu ZL. Auto-weighted multi-view clustering via kernelized graph learning. Pattern Recognition, 2019, 88: 174–184. [doi: 10.1016/j.patcog.2018.11.007]

[19] Xia DX, Yang Y, Wang H, Yang SH. Late fusion multi-view clustering based on local multi-kernel learning. Journal of Computer Research and Development, 2020, 57(8): 1627–1638 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200212]

[20] Yin QY, Wu S, He R, Wang L. Multi-view clustering via pairwise sparse subspace representation. Neurocomputing, 2015, 156: 12–21. [doi: 10.1016/j.neucom.2015.01.017]

[21] Tumer K, Agogino AK. Ensemble clustering with voting active clusters. Pattern Recognition Letters, 2008, 29(14): 1947–1953. [doi: 10.

1016/j.patrec.2008.06.011]

[22] Chang S, Hu J, Li TR, Wang H, Peng B. Multi-view clustering via deep concept factorization. Knowledge-based Systems, 2021, 217: 106807. [doi: 10.1016/j.knosys.2021.106807]

[23] Wang SP, Xiao SX, Zhu W, Guo YY. Multi-view fuzzy clustering of deep random walk and sparse low-rank embedding. Information Sciences, 2022, 586: 224–238. [doi: 10.1016/j.ins.2021.11.075]

[24] Li JQ, Zhou GX, Qiu YN, Wang YJ, Zhang Y, Xie SL. Deep graph regularized non-negative matrix factorization for multi-view clustering. Neurocomputing, 2020, 390: 108–116. [doi: 10.1016/j.neucom.2019.12.054]

[25] Zhao XR, Evans N, Dugelay JL. A subspace co-training framework for multi-view clustering. Pattern Recognition Letters, 2014, 41: 73–82. [doi: 10.1016/j.patrec.2013.12.003]

[26] Jiang YZ, Deng ZH, Wang J, Qian PJ, Wang ST. Collaborative partition multi-view fuzzy clustering algorithm using entropy weighting. Ruan Jian Xue Bao/Journal of Software, 2014, 25(10): 2293–2311 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4510.htm [doi: 10.13328/j.cnki.jos.004510]

[27] Huang L, Chao HY, Wang CD. Multi-view intact space clustering. Pattern Recognition, 2019, 86: 344–353. [doi: 10.1016/j.patcog.2018.09.016]

[28] Zhang CQ, Fu HZ, Wang J, Li W, Cao XC, Hu QH. Tensorized multi-view subspace representation learning. Int'l Journal of Computer Vision, 2020, 128(8): 2344–2361. [doi: 10.1007/s11263-020-01307-0]

[29] Zhang P, Liu XW, Xiong J, Zhou SH, Zhao WT, Zhu E, Cai ZP. Consensus one-step multi-view subspace clustering. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(10): 4676–4689. [doi: 10.1109/TKDE.2020.3045770]

[30] Wang SW, Liu XW, Zhu XZ, Zhang P, Zhang Y, Gao F, Zhu E. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. IEEE Trans. on Image Processing, 2022, 31: 556–568. [doi: 10.1109/TIP.2021.3131941]

[31] Zhan K, Zhang CQ, Guan JP, Wang JS. Graph learning for multiview clustering. IEEE Trans. on Cybernetics, 2018, 48(10): 2887–2895. [doi: 10.1109/TCYB.2017.2751646]

[32] Wang H, Yang Y, Liu B. GMC: Graph-based multi-view clustering. IEEE Trans. on Knowledge and Data Engineering, 2020, 32(6): 1116–1129. [doi: 10.1109/TKDE.2019.2903810]

[33] Li LS, He HB. Bipartite graph based multi-view clustering. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(7): 3111–3125. [doi: 10.1109/TKDE.2020.3021649]

[34] Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. In: Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms. New Orleans: Society for Industrial and Applied Mathematics, 2007. 1027–1035.

[35] Chen MS, Huang L, Wang CD, Huang D. Multi-view clustering in latent embedding space. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI, 2020. 3513–3520. [doi: 10.1609/aaai.v34i04.5756]

[36] Fan K. On a theorem of Weyl concerning eigenvalues of linear transformations: II. Proc. of the National Academy of Sciences of the United States of America, 1950, 36(1): 31–35. [doi: 10.1073/pnas.36.1.31]

[37] Nie FP, Wang XQ, Huang H. Clustering and projected clustering with adaptive neighbors. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 977–986. [doi: 10.1145/2623330.2623726]

[38] van Breukelen M, Duin RPW, Tax DMJ, den Hartog JE. Handwritten digit recognition by combined classifiers. Kybernetika, 1998, 34(4): 381–386.

[39] Winn J, Jojic N. LOCUS: Learning object classes with unsupervised segmentation. In: Proc. of the 10th IEEE Conf. on Computer Vision. Beijing: IEEE, 2005. 745–763. [doi: 10.1109/ICCV.2005.148]

[40] Liu SY, Wang SW, Zhang P, Xu K, Liu XW, Zhang CW, Gao F. Efficient one-pass multi-view subspace clustering with consensus anchors. In: Proc. of the 36th AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI, 2022. 7576–7584. [doi: 10.1609/aaai.v36i7.20723]

[41] Yang B, Zhang XT, Li ZH, Nie FP, Wang F. Efficient multi-view K-means clustering with multiple anchor graphs. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(7): 6887–6900. [doi: 10.1109/TKDE.2022.3185683]

[42] Kang Z, Zhou WT, Zhao ZT, Shao JM, Han M, Xu ZL. Large-scale multi-view subspace clustering in linear time. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI, 2020. 4412–4419. [doi: 10.1609/aaai.v34i04.5867]

附中文参考文献:

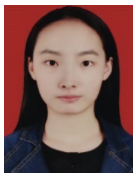
[4] 刘露, 彭涛, 左万利, 戴耀康. 一种基于聚类的PU主动文本分类方法. 软件学报, 2013, 24(11): 2571–2583. http://www.jos.org.cn/1000-9825/4467.htm [doi: 10.3724/SP.J.1001.2013.04467]

[12] 刘晓琳, 白亮, 赵兴旺, 梁吉业. 基于多阶近邻融合的不完整多视图聚类算法. 软件学报, 2022, 33(4): 1354–1372. http://www.jos.org.

cn/1000-9825/6471.htm [doi: 10.13328/j.cnki.jos.006471]
[13] 张远鹏, 周洁, 邓赵红, 钟富礼, 蒋亦樟, 杭文龙, 王士同. 代表点一致性约束的多视角模糊聚类算法. 软件学报, 2019, 30(2): 282–301. http://www.jos.org.cn/1000-9825/5625.htm [doi: 10.13328/j.cnki.jos.005625]
[19] 夏冬雪, 杨燕, 王浩, 阳树洪. 基于邻域多核学习的后融合多视图聚类算法. 计算机研究与发展, 2020, 57(8): 1627–1638. [doi: 10.7544/issn1000-1239.2020.20200212]
[26] 蒋亦樟, 邓赵红, 王骏, 钱鹏江, 王士同. 熵加权多视角协同划分模糊聚类算法. 软件学报, 2014, 25(10): 2293–2311. http://www.jos.org.cn/1000-9825/4510.htm [doi: 10.13328/j.cnki.jos.004510]

赵兴旺(1984－), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为数据挖掘与机器学习.

刘晓琳(1990－), 女, 博士, 讲师, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘.



王淑君(1999－), 女, 硕士生, 主要研究领域为多视图机器学习.

梁吉业(1962－), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为数据挖掘, 机器学习, 人工智能.