计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(3):134–145 [doi: 10.15888/j.cnki.csa.009424]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于三支决策的多视图低秩稀疏子空间聚类算法①

方英杰[1], 贾天夏[1], 徐　怡[2], 骆　帆[2]

[1](安徽大学 纽约石溪学院, 合肥 230039)
[2](安徽大学 计算机科学与技术学院, 合肥 230601)
通信作者: 方英杰, E-mail: yingjie0418@126.com

摘　要: 多视图子空间聚类是一种从子空间中学习所有视图共享的统一表示, 挖掘数据潜在聚类结构的方法. 作为一种处理高维数据的聚类方法, 子空间聚类是多视图聚类领域的研究热点之一. 多视图低秩稀疏子空间聚类是一种结合了低秩表示和稀疏约束的子空间聚类方法. 该算法在构造亲和矩阵过程中, 利用低秩稀疏约束同时捕捉了数据的全局结构和局部结构, 优化了子空间聚类的性能. 三支决策是一种基于粗糙集模型的决策思想, 常被应用于聚类算法来反映聚类过程中对象与类簇之间的不确定性关系. 本文基于三支决策的思想, 设计了一种投票制度作为决策依据, 将其与多视图稀疏子空间聚类组成一个统一框架, 从而形成一种新的算法. 在多个人工数据集和真实数据集上的实验表明, 该算法可提高多视图聚类的准确性.

关键词: 三支决策; 多视图聚类; 低秩表示; 稀疏约束; 子空间聚类

引用格式: 方英杰,贾天夏,徐怡,骆帆.基于三支决策的多视图低秩稀疏子空间聚类算法.计算机系统应用,2024,33(3):134–145. http://www.c-s-a.org.cn/1003-3254/9424.html

Multi-view Low-rank Sparse Subspace Clustering Algorithm Based on Three-way Decision

FANG Ying-Jie[1], JIA Tian-Xia[1], XU Yi[2], LUO Fan[2]

[1](Stony Brook Institute at Anhui University, Hefei 230039, China)
[2](School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: Multi-view subspace clustering is a method for learning a unified representation of all views from subspaces and exploring the latent clustering structure of data. As a clustering approach for processing high-dimensional data, subspace clustering has become a focal point in the field of multi-view clustering. Multi-view low-rank sparse subspace clustering method combines low-rank representation and sparse constraints. During the construction of the affinity matrix, this algorithm utilizes low-rank sparse constraints to capture both global and local structures of the data, thereby optimizing the performance of subspace clustering. The three-way decision, rooted in the rough set model, is a decision-making concept often applied in clustering algorithms to reflect the uncertainty relationship between objects and clusters during the clustering process. In this study, inspired by the idea of the three-way decision, a voting system is designed as the decision basis. The system is integrated with multi-view sparse subspace clustering to form a unified framework, resulting in a novel algorithm. Experimental results on various artificial and real-world datasets demonstrate that this algorithm can enhance the accuracy of multi-view clustering.

Key words: three-way decision; multi-view clustering; low-rank representation; sparse constraint; subspace clustering

① 基金项目: 安徽大学大学生科研训练计划 (SXKY32205)
收稿时间: 2023-09-06; 修改时间: 2023-10-08; 采用时间: 2023-10-20; csa 在线出版时间: 2024-01-17
CNKI 网络首发时间: 2024-01-19

观察样本的数据信息往往可以取自多个视图, 来获得不同特征描述. 例如, 一件艺术作品的价值可以从审美、思想和技巧等不同的角度进行分析, 而技巧又包含多种不同的表现手法. 这些分析结果都可以被称为多视图数据. 多视图数据通常具有一致性和互补性, 因而可以通过尝试在不同视图间寻找共享的类簇结构, 实现比单一视图聚类更加准确的聚类方式, 这样的聚类被称为多视图聚类[1].

由于多视图数据来源的多样性, 多视图数据通常表现为高维数据, 这意味着计算时需要面临成倍增加的时间与空间复杂度. 同时, 由于客观噪声与采样误差, 多视图数据在聚类时的准确性会受到极大的影响. 子空间聚类可以将高维数据映射到低维的子空间中, 该聚类算法假设所有高维数据可以看作是低维子空间的并集, 通过对子空间分配类簇, 可以解决高维数据的聚类难题[2]. 由于高维数据点分布较为任意且稀疏, 传统的基于质心分布假设的算法在子空间聚类的结果上往往表现不佳[3]. 谱聚类[4,5]是一种解决子空间聚类问题的有效方法, 它因在任意形状类簇上的良好聚类表现, 受到了广泛关注. 在谱聚类过程中, 如何构建合适的相似度矩阵是主要的难题. 其中, 低秩稀疏子空间聚类[6]被认为是一种有效可靠的解决方案. 该方案结合了低秩子空间聚类[7,8]与稀疏子空间聚类[9]. 通过低秩约束来捕获数据的全局结构, 在假设子空间相互独立且采样数据充分的情况下, 保证了聚类的精度, 使得对象能被同一子空间中的其他对象线性表示. 同时, 借助稀疏表示, 对象能够被表示为其他对象的稀疏线性组合. 相比“纯低秩表示”, 稀疏约束的假设相对宽松, 但该方法受数据点维度的影响较大, 存在对子空间过分割的可能性. 因此, 对于子空间聚类来说, 结合低秩表示与稀疏约束来捕捉对象的全局结构与局部结构是必要的. 低秩稀疏子空间聚类被众多学者拓展研究, 应用在不同的领域. 其中, 多视图数据处理领域是许多学者研究的热点之一. Brbić等[3]将低秩稀疏子空间聚类推广到多视图领域, 用基于成对与基于质心两种正则化方法对多视图数据进行了处理, 并尝试通过解决核希尔伯特空间中的相关优化问题, 将算法扩展到非线性子空间中; Tian 等[10]将基于成对正则化的多视图低秩稀疏子空间聚类应用于高光谱图像分类, 在非重叠的三维块上实现了该算法在大尺度遥感图像上的应用, 进一步证明了 MLRSSC 算法巨大的发展前景. 王丽娟等[11]将视图多样性概念应用于多视图低秩稀疏子空间聚类算法中. 通过挖掘多视图特征的多样性信息, 确保了不同视图下子空间表示矩阵的多样性.

聚类的目的是对相似的对象进行划分归类. 在聚类算法中, 若每个对象只属于单一类簇, 则被称为硬聚类; 若每个对象可归属于多个类簇, 则被称为软聚类. 从决策角度讲, 硬聚类是一种非黑即白的二支决策聚类; 软聚类虽然存在归属多个类簇的可能, 但这种聚类的表达方式模糊了对象与类簇间的不确定性. 为了区分出对象与类簇间“是”“否”“不确定”的关系, 需要引入 Yao 提出的三支决策[12]的思想, 用核心域与边缘域的组合结构来表示类簇, 将确定的对象存入核心域中, 将不确定的对象存入边缘域中. 这种聚类表示方式被称为三支聚类, 由 Yu 等[13–15]首次提出. 三支决策的核心思想是在没有把握作出决策的情况下, 延迟决策的进行, 等至有足够把握再执行决策. 在聚类过程中, 该决策可以在一定程度上避免高风险决策的发生, 以此来达到提高聚类性能的目的. 自提出三支聚类后, 许多学者进行了拓展研究. 于洪等[16]将三支决策应用于传统的 K-means 算法, 提出了一种考虑 q 近邻的分离性指数来作为决策的主要依据进行三支聚类. 夏月月等[17]将网格聚类算法中划分网格的思想加入三支决策下的 K-means 聚类, 用网格密度来快速确定核心域与边缘域, 有效解决 K-means 算法初始聚类中心随机选择导致的聚类结果不准确问题; 徐天杰等[18]将人工蜂群与三支 K-means 算法结合来解决聚类中心初始化的问题, 通过人工蜂群算法寻求最优蜜源的位置, 将其作为初始聚类中心, 并在此基础上重复迭代聚类算法. Yu 等[19]将三支决策与主动学习策略相结合, 引入多视图聚类. 在基于熵概念测量对象的不确定性之后, 学习重要信息的成对约束, 来提高高维多视图数据的聚类精度.

在通过谱聚类获得数据点隶属度的过程中, K-means 聚类作为其中一环, 将被应用于对图拉普拉斯矩阵的特征向量[3]进行聚类, 而 K-means 聚类是一种硬聚类算法, 只考虑了对象属于类簇或者不属于类簇这两种关系, 直接对所有样本进行了一次性划分. 因此, 多视图稀疏子空间聚类得到的是二支聚类的结果. 这种聚类结果没有反映出类簇与对象的不确定性关系[19]. 对此, 本文提出了一种基于投票的三支决策聚类, 将其与谱聚类算法结合, 进一步反映聚类过程中对象和类簇的不确定性关系. 通过优先对确定对象进行聚类, 并

修正相关参数, 来逐步完善各类簇的主体结构与内部信息. 投票制度保证了尽可能多的可靠数据点参与聚类时的划分决策, 在减弱聚类随机性的同时, 降低了错误决策带来的负面影响.

因此, 本文引入三支决策思想, 提出了一种基于三支决策的多视图稀疏子空间聚类方法. 该方法对多视图稀疏子空间聚类进行了拓展, 在一些数据集中, 表现出了更优的聚类性能.

1 相关理论基础

本节主要介绍了多视图低秩稀疏子空间聚类 (MLRSSC) 与三支决策聚类 (TWDC). 低秩稀疏子空间聚类的主要任务是借助数据在低维子空间中的稀疏表示, 为一组给定的数据集构造出合适的亲和矩阵, 最后通过谱聚类的方法得到基于子空间的聚类结果. 三支决策聚类则是通过构造一种新的类簇内部结构, 对样本对象重新进行了划分, 筛选出核心对象与边缘对象, 从而挖掘出样本对象潜在的不确定性. 在一定范围内, 借助三支决策可以更新相似度矩阵, 继而迭代谱聚类过程, 以此获得更加准确聚类结果.

1.1 低秩表示与稀疏约束

考虑一个样本数量为 n 的数据集$\mathbf{X}=\{\mathbf{x}_i\in\mathbb{R}^D\}_{i=1}^n$, 目标是根据数据样本所属的子空间对其进行聚类划分. 这个过程的关键是求出反映数据点之间相似程度的矩阵 $\mathbf{W}$, 即相似度矩阵.

低秩表示[7,8]的思想是寻找数据集的最低秩表示, 试图从数据集中找到一个低秩表示矩阵$\mathbf{C}\in\mathbb{R}^{\mathrm{N}\times\mathrm{N}}$, 其目标函数为:

$$\min_{\mathbf{C}} rank(\mathbf{C}),\ \text{s.t. } \mathbf{X}=\mathbf{XC} \tag{1}$$

由于秩函数的离散性质, 上述公式的优化是 NP 难问题. 一个有效的方法是引入核范数代替秩函数, 将其转化为凸优化问题. 考虑到细微噪声 (如高斯噪声) 的影响, 一个相对合理的策略是加入一个松弛约束系数来放宽等式约束, 将目标函数转化为:

$$\min_{\mathbf{C}}\|\mathbf{C}\|_*+\lambda\|\mathbf{E}\|_F,\ \text{s.t. } \mathbf{X}=\mathbf{XC}+\mathbf{E} \tag{2}$$

其中, $\|\cdot\|_*$为核函数, 用来近似矩阵 $\mathbf{C}$ 的秩, λ为噪声的平衡参数, $\|\cdot\|_F$为 Frobenius 范数.

稀疏子空间聚类[9]的思想是将数据表示为同一子空间中其他数据的线性组合, 在子空间彼此独立的条件下, 通过引入了ℓ_1范数来求得稀疏解, 其目标函数为:

$$\min_{\mathbf{C}}\|\mathbf{C}\|_1,\ \text{s.t. } \mathbf{X}=\mathbf{XC},\mathrm{diag}(\mathbf{C})=0 \tag{3}$$

类似地, 在受到噪声污染的情况下, 重新考虑目标函数为:

$$\min_{\mathbf{C}}\|\mathbf{C}\|_1+\lambda\|\mathbf{E}\|_F,\ \text{s.t. } \mathbf{X}=\mathbf{XC}+\mathbf{E},\ \mathrm{diag}(\mathbf{C})=0 \tag{4}$$

其中, λ是平衡噪声与表示矩阵的参数, 一般基于两个范数的性质进行选择, 也可以根据经验进行调整[7].

在受到噪声污染的情况下, 低秩稀疏子空间聚类的目标函数如下:

$$\begin{cases}\min\limits_{\mathbf{C}}\beta_1\|\mathbf{C}\|_*+\beta_2\|\mathbf{C}\|_1+\lambda\|\mathbf{E}\|_F\\ \text{s.t. } \mathbf{X}=\mathbf{XC}+\mathbf{E},\ \mathrm{diag}(\mathbf{C})=0\end{cases} \tag{5}$$

当通过低秩稀疏子空间聚类求解出矩阵 $\mathbf{C}$ 后, 可以通过式 (6) 求得相似度矩阵:

$$\mathbf{W}=|\mathbf{C}|+|\mathbf{C}|^{\mathrm{T}} \tag{6}$$

利用谱聚类的方法, 对得到的相似度矩阵 $\mathbf{W}$ 进行聚类, 可以得到数据集 $\mathbf{X}$ 内部对象的初步划分.

1.2 三支决策聚类

一般来说, 聚类即是对 n 个待划分的样本对象 $\{x_1,x_2,x_3,\cdots,x_n\}$, 依据其特征关系划分入 K 个类簇 $\{Z_1,Z_2,Z_3,\cdots,Z_K\}$的过程. 对于一个待划分对象$x_i$, 传统的二支聚类决策会将其划入单一的聚类集合中, 但这无法反映对象与类簇间的不确定性关系. 在一些实际情况中, 由于信息不足, 算法难以对当前样本的归属做出一个相对准确的判断. Yao 等[12]在决策粗糙集理论中提出了三支决策的概念, 即将原本单一的聚类集合 Z 划分为互不相交的 3 个部分: 正域、负域、边界域. 近年来, 不少研究人员对三支决策理论展开了研究, 拓宽了其应用领域. Yu 等[13−15]重新定义了三支决策中聚类的类簇表示, 在多视图聚类、集成聚类[20]、软增量聚类[21,22]等领域实现了有效的落地应用.

三支决策的思想是用一组三元集合$\{CoreArea,\ FriArea,\ TriArea\}$来表示类簇 Z_i 的内部结构, 其中 $CoreArea$ 为类簇的核心域, 其间的元素对象明确属于该类簇. $FriArea$ 为类簇的边缘域, 其间的对象暂时不确定是否属于该类簇. $TriArea$ 为类簇的琐碎域, 表达式为 $U-(CoreArea\cup FriArea)$, 其间的对象不属于该类簇. 同时, 3 个域之间互不相交, 且它们的并集为样本对象的全集. 我们分别用 $Co(Z_i)$ 与 $Fr(Z_i)$ 来分别指代

某一具体类簇 Z_i 的核心域与边缘域, 将类簇 Z_i 简化为一个二元集合$\{Co(Z_i), Fr(Z_i)\}$. 核心域与边缘域满足如下性质: (1) 每个类簇的核心域内至少有一个对象, 不能为空; (2) 任一对象无法存在于多个核心域内, 但可以同时存在于多个边缘域; (3) 所有对象都必须划入某个类簇中; (4) 任一对象一旦进入某个类簇的核心域, 则无法存在于其他类簇的边缘域.

三支聚类的结果将表示为: $\{(Co(Z_1), Fr(Z_1)), \cdots, (Co(Z_K), Fr(Z_K))\}$.

当边缘域的结果全部划入核心域, 则聚类结果以二支的方式呈现: $\{Co(Z_1), \cdots, Co(Z_K)\}$.

2 多视图低秩稀疏子空间的三支聚类

2.1 多视图聚类模型

对一个有 n 个视角的多视图数据进行子空间聚类, 首先需要对不同视图的数据表示矩阵进行正则化. 而在聚类时所选择的正则化方式往往对最终聚类结果有重要影响. 本文将以基于质心的正则化方式进行多视图子空间聚类, 在无噪声的情况时, 其目标函数如下:

$$\begin{cases}\min\limits_{\mathbf{C}^{(1)},\cdots,\mathbf{C}^{(n_v)}}\sum\limits_{v=1}^{n_v}(\beta_1\|\mathbf{C}^{(v)}\|_*+\beta_2\|\mathbf{C}^{(v)}\|_1+\lambda^{(v)}\|\mathbf{C}^{(v)}-\mathbf{C}^*\|_F^2)\\ \text{s.t. } \mathbf{X}^{(v)}=\mathbf{X}^{(v)}\mathbf{C}^{(v)},\ \text{diag}(\mathbf{C}^{(v)})=0,\ v=1,2,\cdots,n_v\end{cases} \tag{7}$$

其中, $\mathbf{C}^{(v)}$为第 v 个视角的数据表示矩阵, β_1与 β_2分别为平衡低秩矩阵与稀疏矩阵的平衡系数, $\lambda^{(v)}$为视图间的一致性权重参数, $\mathbf{C}^*$是一致性变量, 表示不同视图间的共生子空间.

2.2 多视图聚类算法求解

基于质心正则化的目标函数中包括多个变量, 同时求解难度很大. 因此, 需要通过交替最小化各视图的表示矩阵$\mathbf{C}^{(v)}$与一致性变量$\mathbf{C}^*$来优化目标函数, 其主要思想如下.

(1) 固定一致性变量$\mathbf{C}^*$, 用 ADMM 交替方向乘子法[23]求解各视图$\mathbf{C}^{(1)},\mathbf{C}^{(2)},\cdots,\mathbf{C}^{(v)}$.

(2) 固定各视图$\mathbf{C}^{(1)},\mathbf{C}^{(2)},\cdots,\mathbf{C}^{(v)}$, 求解$\mathbf{C}^*$.

式 (7) 的求解是一个受线性约束的凸优化问题, 所有子问题都能被精确求解[3], 文献[24]保证了 ADMM 算法的全局收敛性. 根据文献[3], 当迭代次数到达最大值 T 或迭代的变量符合收敛条件时, 迭代停止. 迭代停止时, 需要将 $\mathbf{C}^*$带入式 (6) 得到相似度矩阵 $\mathbf{W}$, 最后对相似度矩阵 $\mathbf{W}$ 应用谱聚类的方法完成多视图部分的聚类.

(1) 固定$\mathbf{C}^*$, 求解$\mathbf{C}^{(v)}$.

当固定除某一特定视图$\mathbf{C}^{(v)}$以外的全部变量后, 目标函数如下:

$$\begin{cases}\min\limits_{\mathbf{C}^{(v)}}\beta_1\|\mathbf{C}^{(v)}\|_*+\beta_2\|\mathbf{C}^{(v)}\|_1+\lambda^{(v)}\|\mathbf{C}^{(v)}-\mathbf{C}^*\|_F^2\\ \text{s.t. } \mathbf{X}^{(v)}=\mathbf{X}^{(v)}\mathbf{C}^{(v)},\ \text{diag}(\mathbf{C}^{(v)})=0\end{cases} \tag{8}$$

通过引入辅助变量$\mathbf{A}^{(v)},\mathbf{C}_1^{(v)},\mathbf{C}_2^{(v)},\mathbf{C}_3^{(v)}$,将式 (8) 变为:

$$\begin{cases}\min\limits_{\mathbf{A}^{(v)},\mathbf{C}_1^{(v)},\mathbf{C}_2^{(v)},\mathbf{C}_3^{(v)}}\beta_1\|\mathbf{C}_1^{(v)}\|_*+\beta_2\|\mathbf{C}_2^{(v)}\|_1+\lambda^{(v)}\|\mathbf{C}_3^{(v)}-\mathbf{C}^*\|_F^2\\ \text{s.t. } \mathbf{A}^{(v)}=\mathbf{C}_2^{(v)}-\text{diag}(\mathbf{C}_2^{(v)}),\ \mathbf{X}^{(v)}=\mathbf{X}^{(v)}\mathbf{A}^{(v)}\\ \mathbf{A}^{(v)}=\mathbf{C}_1^{(v)},\mathbf{A}^{(v)}=\mathbf{C}_3^{(v)}\end{cases} \tag{9}$$

将式 (9) 写为拉格朗日增广形式:

$$\begin{aligned}L(\mathbf{A}^{(v)},\{\mathbf{C}_i^{(v)}\}_{i=1}^3,\{\mathbf{\Lambda}_i\}_{i=1}^4)&=\beta_1\|\mathbf{C}_1^{(v)}\|_*\\&+\beta_2\|\mathbf{C}_2^{(v)}\|_1+\lambda\|\mathbf{C}_3^{(v)}-\mathbf{C}^*\|_F^2\\&+\frac{\mu_1}{2}\|\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{A}^{(v)}\|_F^2\\&+\frac{\mu_2}{2}\|\mathbf{A}^{(v)}-\mathbf{C}_2^{(v)}+\text{diag}(\mathbf{C}_2^{(v)})\|_F^2\\&+\frac{\mu_3}{2}\|\mathbf{A}^{(v)}-\mathbf{C}_1^{(v)}\|_F^2+\frac{\mu_4}{2}\|\mathbf{A}^{(v)}-\mathbf{C}_3^{(v)}\|_F^2\\&+\text{tr}\left[(\mathbf{\Lambda}_1^{(v)})^{\mathrm{T}}(\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{A}^{(v)})\right]\\&+\text{tr}\left[(\mathbf{\Lambda}_2^{(v)})^{\mathrm{T}}(\mathbf{A}^{(v)}-\mathbf{C}_2^{(v)}+\text{diag}(\mathbf{C}_2^{(v)}))\right]\\&+\text{tr}\left[(\mathbf{\Lambda}_3^{(v)})^{\mathrm{T}}(\mathbf{A}^{(v)}-\mathbf{C}_1^{(v)})\right]+\text{tr}\left[(\mathbf{\Lambda}_4^{(v)})^{\mathrm{T}}(\mathbf{A}^{(v)}-\mathbf{C}_3^{(v)})\right]\end{aligned} \tag{10}$$

其中, μ_i是需要进一步调整的惩罚参数, $\{\mathbf{\Lambda}_i\}_{i=1}^4$是拉格朗日对偶变量.

为了解决式 (9) 的凸优化问题, 需要用 ADMM 算法进行求解, 分别更新各个辅助变量.

将式 (10) 中关于$\mathbf{A}^{(v)}$的偏导置为 0, 以得到其在 n+1 轮迭代的更新公式:

$$\begin{aligned}\mathbf{A}^{(v)}=&[\mu_1(\mathbf{X}^{(v)})^{\mathrm{T}}\mathbf{X}^{(v)}+\mu_2\mathbf{I}+\mu_3\mathbf{I}+\mu_4\mathbf{I}]^{-1}\\&\times[\mu_1(\mathbf{X}^{(v)})^{\mathrm{T}}\mathbf{X}^{(v)}+\mu_2\mathbf{C}_2^{(v)}+\mu_3\mathbf{C}_1^{(v)}+\mu_4\mathbf{C}_3^{(v)}\\&+(\mathbf{X}^{(v)})^{\mathrm{T}}\mathbf{\Lambda}_1^{(v)}-\mathbf{\Lambda}_2^{(v)}-\mathbf{\Lambda}_3^{(v)}-\mathbf{\Lambda}_4^{(v)}]\end{aligned} \tag{11}$$

其中, $\{\mathbf{C}_i^{(v)}\}_{i=1}^3,\{\mathbf{\Lambda}_i^{(v)}\}_{i=1}^4$为第 n 轮的变量.

在更新$\mathbf{C}_1^{(v)}$时[25], 需要对$(\mathbf{A}^{(v)}+\mu_3^{-1}\mathbf{\Lambda}_3^{(v)})$进行奇异值分解, 用软阈值函数$\eta_s$对其奇异值进行修正:

$$\begin{cases}\mathbf{A}^{(v)}+\dfrac{\boldsymbol{\Lambda}_3^{(v)}}{\mu_3}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}\ rank\left(\mathbf{A}^{(v)}+\dfrac{\boldsymbol{\Lambda}_3^{(v)}}{\mu_3}\right)=r\\ \eta_s\left(\boldsymbol{\Sigma},\dfrac{\beta_1}{\mu_3}\right)=\mathrm{diag}\left(\left\{sgn(\sigma_i)\left(|\sigma_i|-\dfrac{\beta_1}{\mu_3}\right)_+\right\}\right)\\ \boldsymbol{\Sigma}\ =\ \mathrm{diag}\left(\{\sigma_i\}1\leqslant i\leqslant r\right)\end{cases}\tag{12}$$

其中, $\mathbf{A}^{(v)}$为第 n+1 轮的变量, $\boldsymbol{\Lambda}_3^{(v)}$为第 n 轮的变量; $\mathbf{U},\boldsymbol{\Sigma},\mathbf{V}^{\mathrm{T}}$这 3 个矩阵取自矩阵$(\mathbf{A}^{(v)}+\mu_3{}^{-1}\boldsymbol{\Lambda}_3^{(v)})$的奇异值分解, t_+的具体表达式为$t_+=\max(0,t)$. 最终通过式 (13) 对$\mathbf{C}_1^{(v)}$进行更新:

$$\mathbf{C}_1^{(v)}=\mathbf{U}\eta_s\left(\boldsymbol{\Sigma},\frac{\beta_1}{\mu_3}\right)\mathbf{V}^{\mathrm{T}}\tag{13}$$

在文献[3]中, 将该过程表述为式 (14):

$$\mathbf{C}_1^{(v)}=\pi_{\frac{\beta_1}{\mu_3}}\left(\mathbf{A}^{(v)}+\frac{\boldsymbol{\Lambda}_3^{(v)}}{\mu_3}\right)\tag{14}$$

根据文献[26,27], 类似地, $\mathbf{C}_2^{(v)}$的更新规则如下:

$$\begin{cases}\mathbf{C}_2^{(v)}=\pi_{\frac{\beta_2}{\mu_2}}\left(\mathbf{A}^{(v)}+\dfrac{\boldsymbol{\Lambda}_2^{(v)}}{\mu_2}\right)\\ \mathbf{C}_2^{(v)}=\mathbf{C}_2^{(v)}-\mathrm{diag}\left(\mathbf{C}_2^{(v)}\right)\end{cases}\tag{15}$$

通过将式 (9) 中关于$\mathbf{C}_3^{(v)}$的偏导数设置为 0, 求得$\mathbf{C}_3^{(v)}$的更新规则:

$$\mathbf{C}_3^{(v)}=(2\lambda^{(v)}+\mu_4)^{-1}(2\lambda^{(v)}\mathbf{C}^*+\mu_4\mathbf{A}^{(v)}+\boldsymbol{\Lambda}_4^{(v)})\tag{16}$$

其中, $\mathbf{A}^{(v)}$为第 $n+1$ 轮的变量, $\mathbf{C}^*$与$\boldsymbol{\Lambda}_4^{(v)}$为第 n 轮的变量.

对偶变量$\{\boldsymbol{\Lambda}_i^{(v)}\}_{i=1}^4$与惩罚系数$\mu_i$采用以下的更新规则:

$$\begin{cases}\boldsymbol{\Lambda}_1^{(v)}=\boldsymbol{\Lambda}_1^{(v)}+\mu_1(\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{A}^{(v)})\\ \boldsymbol{\Lambda}_2^{(v)}=\boldsymbol{\Lambda}_2^{(v)}+\mu_2(\mathbf{A}^{(v)}-\mathbf{C}_2^{(v)})\\ \boldsymbol{\Lambda}_3^{(v)}=\boldsymbol{\Lambda}_3^{(v)}+\mu_3(\mathbf{A}^{(v)}-\mathbf{C}_1^{(v)})\\ \boldsymbol{\Lambda}_4^{(v)}=\boldsymbol{\Lambda}_4^{(v)}+\mu_4(\mathbf{A}^{(v)}-\mathbf{C}_3^{(v)})\\ \mu_i=\min(\rho\mu_i,\mu_{\max}),\ i=1,\cdots,4\end{cases}\tag{17}$$

其中, $\mathbf{A}^{(v)}$与$\{\mathbf{C}_i^{(v)}\}_{i=1}^3$均为第 n+1 轮变量. $\mu_{\max}$为惩罚参数的最大值, ρ为惩罚参数的正系数, 本文参考文献[3], 保持视图间的惩罚系数一致.

在噪声影响下, 式 (9) 的目标函数修改为:

$$\begin{cases}\min\limits_{\mathbf{C}^{(1)},\cdots,\mathbf{C}^{(n_v)}}\sum\limits_{v=1}^{n_v}\left(\dfrac{1}{2}\left\|\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{A}^{(v)}\right\|_F^2+\beta_1\|\mathbf{C}^{(v)}\|_*\right.\\ \qquad\left.+\beta_2\|\mathbf{C}^{(v)}\|_1+\lambda^{(v)}\|\mathbf{C}^{(v)}-\mathbf{C}^*\|_F^2\right)\\ \text{s.t. }\mathbf{X}^{(v)}=\mathbf{X}^{(v)}\mathbf{C}^{(v)},\ \mathrm{diag}(\mathbf{C}^{(v)})=0,\ v=1,2,\cdots,n_v\end{cases}\tag{18}$$

在被噪声污染时, $\{\mathbf{C}_i^{(v)}\}_{i=1}^3$与$\{\boldsymbol{\Lambda}_i^{(v)}\}_{i=1}^4$的更新过程不发生变化, 同式 (14)–式 (17); 但矩阵$\mathbf{A}^{(v)}$的更新公式与式 (11) 有所不同, 其式如下:

$$\begin{aligned}\mathbf{A}^{(v)}=&[(\mathbf{X}^{(v)})^{\mathrm{T}}\mathbf{X}^{(v)}+\mu_2\mathbf{I}+\mu_3\mathbf{I}+\mu_4\mathbf{I}]^{-1}\\ &\times[(\mathbf{X}^{(v)})^{\mathrm{T}}\mathbf{X}^{(v)}+\mu_2\mathbf{C}_2^{(v)}+\mu_3\mathbf{C}_1^{(v)}+\mu_4\mathbf{C}_3^{(v)}\\ &+(\mathbf{X}^{(v)})^{\mathrm{T}}\boldsymbol{\Lambda}_1^{(v)}-\boldsymbol{\Lambda}_2^{(v)}-\boldsymbol{\Lambda}_3^{(v)}-\boldsymbol{\Lambda}_4^{(v)}]\end{aligned}\tag{19}$$

(2) 固定$\mathbf{C}^{(v)}$, 求解$\mathbf{C}^*$.

更新完每一个视图的自表示矩阵后, 将$\mathbf{C}^*$相对于目标函数 (10) 的偏导置 0, 得到$\mathbf{C}^*$的解为:

$$\mathbf{C}^*=\frac{\sum_{v=1}^{n_v}\lambda^{(v)}\mathbf{C}^{(v)}}{\sum_{v=1}^{n_v}\lambda^{(v)}}\tag{20}$$

算法 1. 基于质心的正则化多视图低秩稀疏子空间聚类

输入: 各视图矩阵$\{\mathbf{X}^{(v)}\}_{v=1}^{n_v}$, 平衡系数$\beta_1$, β_2, 权重参数$\{\lambda^{(v)}\}_{v=1}^{n_v}$, 惩罚系数$\mu_i$, $\mu_{\max}$, ρ

输出: 各视图间的一致性相似矩阵 $\mathbf{W}$

步骤:

1. 初始化各系数:
   $\{\mathbf{C}_i^{(v)}\}_{i=1}^3=0$, $\mathbf{C}^*=\mathbf{A}^{(v)}=0$, $\{\boldsymbol{\Lambda}_i\}_{i=1}^4=0$, $\varepsilon=10^{-3}$, $T=100$, $i=1,\cdots,n_v$
2. 未收敛时或未超过迭代次数上限 T 时循环执行:
3. 　遍历各视图v: $v=1,\cdots,n_v$
4. 　　固定其他变量, 根据式 (11) 更新$\mathbf{A}^{(v)}$
5. 　　固定其他变量, 根据式 (14) 更新$\mathbf{C}_1^{(v)}$
6. 　　固定其他变量, 根据式 (15) 更新$\mathbf{C}_2^{(v)}$
7. 　　固定其他变量, 根据式 (16) 更新$\mathbf{C}_3^{(v)}$
8. 　　固定其他变量, 根据式 (17) 更新对偶变量$\{\boldsymbol{\Lambda}_i^{(v)}\}_{i=1}^4$
9. 　结束遍历
10. 　根据式 (17) 更新惩罚系数$\mu_i=\min(\rho\mu_i,\mu_{\max})$
11. 　固定各视图变量, 根据式 (20) 更新$\mathbf{C}^*$
12. 　判断收敛条件:

$$\left\|\mathbf{A}^{(v)}-\mathbf{C}_1^{(v)}\right\|_\infty\leqslant\varepsilon$$
$$\left\|\mathbf{A}^{(v)}-\mathbf{C}_2^{(v)}\right\|_\infty\leqslant\varepsilon$$
$$\left\|\mathbf{A}^{(v)}-\mathbf{C}_3^{(v)}\right\|_\infty\leqslant\varepsilon$$
$$\left\|\mathbf{A}_k^{(v)}-\mathbf{A}_{k-1}^{(v)}\right\|_\infty\leqslant\varepsilon$$

13. 结束循环
14. 将$\mathbf{C}^*$带入式 (6) 计算一致性相似矩阵 $\mathbf{W}$

3 基于投票的三支聚类

通过谱聚类对得到的相似度矩阵 $\mathbf{W}$ 进行分割求解, 可以初步得到多视图聚类的解. 但获得的初步解是一种二支聚类解, 没有考虑到类簇与对象间的不确定性, 聚类结果存在一定的误差. 因此, 本文引入三支决策, 设计了一种基于投票的三支聚类, 将其应用于多视图聚类求解的谱聚类环节, 对得到的二支聚类的解作

进一步的修正.

三支决策聚类的主要思想为: 对于确定的解将其纳入核心域, 对于不确定的解将其纳入边缘域. 通过优先构造出由确定性较高的对象组成的核心域, 可以尽可能多地增加掌握的聚类信息, 提升对不确定样本成功聚类的可能性. 聚类基于 k 近邻与投票制度, 一定程度上降低了聚类过程中噪声的干扰. 同时, 考虑到错误划分样本带来的连锁效应, 我们设定了基于原始样本数量动态变化的迭代上限 Q, 通过控制其大小来取得相对较优的聚类解.

本文基于一些评价性指标将三支聚类结果与原来直接通过谱聚类得到的结果进行比对, 来验证其有效性.

3.1 算法框架图

图 1 为三支聚类算法的主要框架流程图. 框架以循环迭代的形式呈现, 其主要包括 3 个模块: 初始化域模块, 扩展域模块以及边缘域对象重划分模块. 其中, 初始化域模块只在第 1 次迭代时启动, 该模块的目的是将可信度较高的聚类结果作为各类簇的基本依据; 扩展域模块则会根据当前候选区情况选择性启动, 其主要思想是通过不断完善各类簇核心域内的主体信息, 重新考虑之前的未划分对象, 决定是否将其归入类簇; 而重划分模块在每轮迭代时都会启动. 这是考虑到聚类过程中各类簇的特征信息是实时变化的, 一些早先被划入边缘域的样本可能会因这些信息变动, 获得进入核心域的资格; 同时, 某些边缘域样本的重划分, 可能导致一些无法得到划分的对象重新获得划入某一类簇的资格. 重划分的目的就是在于尽可能谨慎地、及时地更新这些可能存在的变动.

三支决策聚类迭代结束条件为循环次数大于设定值 Q 或者无可划分目标对象, Q 值需要视具体数据样本对象的复杂度设定.

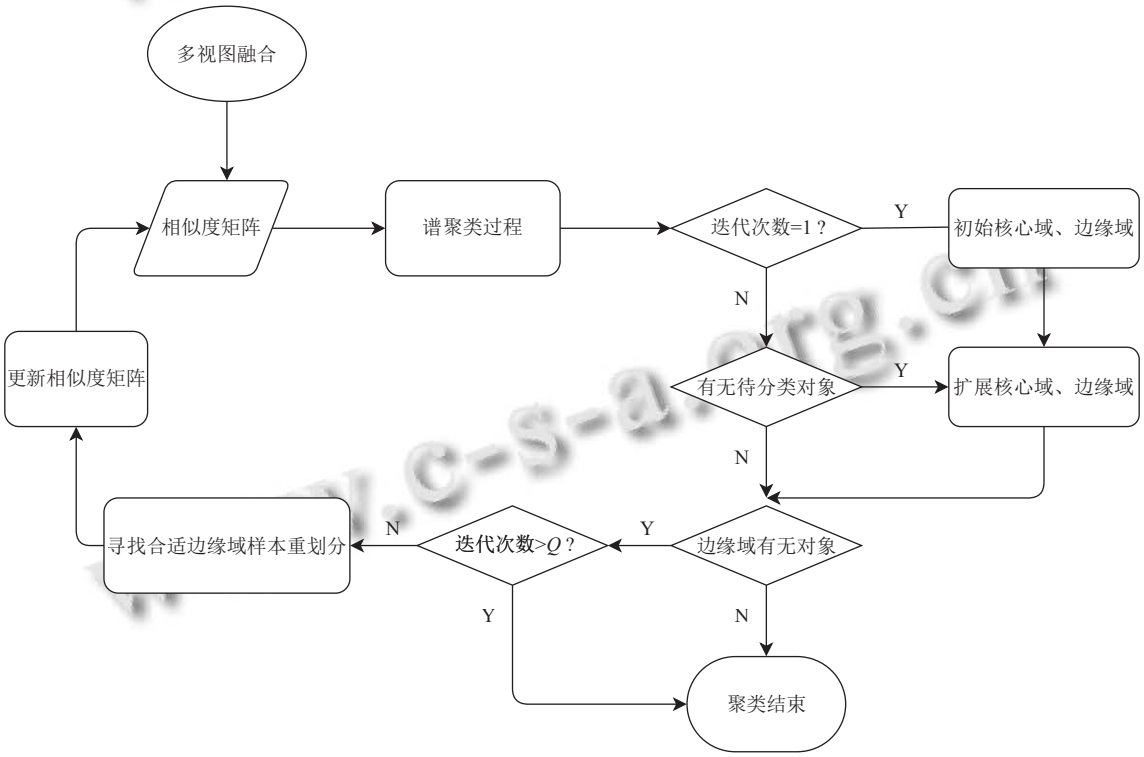

图 1 算法框架图

3.2 算法思想

为了得到一个三支聚类的结果, 需要经历初始、扩展、重划分 3 个主要阶段. 本文的三支决策思想基于文献[19]与文献[28]做出改进, 在初始化部分, 核心域基于原有的二支聚类结果进行重塑; 在扩展与重划分部分, 引入基于 k 近邻标准的投票制度, 在尽可能保证对聚类性能不产生负提升的情况下, 一定程度上对其结果进行部分修正, 提高聚类效率.

算法 2. 基于投票的三支决策算法

输入: 相似度矩阵 **W**, 类簇数量 K

输出: 三支表示最终的聚类结果$\{(Co(Z_1), Fr(Z_1)), \cdots, (Co(Z_i), Fr(Z_i)), \cdots, (Co(Z_K), Fr(Z_K))\}$

步骤:

1. 相似度矩阵 **W** 执行谱聚类过程, 得到初始二支解$\{Z_1^*, Z_2^*, \cdots, Z_K^*\}$

2. 判断是否为第 1 次迭代. 若是, 则在原二支聚类解中, 选取距离聚类中心最近的样本点与其周围样本点初始化核心域$\{Co(Z_1), \cdots, Co(Z_K)\}$; 否则进入步骤 3.

3. 判断当前待分类区是否划分完毕. 若未划分完毕, 则遍历每一个候选区的待分类样本, 使其参与投票. 对于某个待分类样本 **x**, 所有类簇的核心域样本$\mathbf{x}_j \in \cup_{k=1}^{K} Co(Z_k)$需依据式 (22) 进行投票. 若多个类簇的核心域样本参与竞争, 依据式 (23) 执行最终决策; 否则进入步骤 4.

4. 判断当前边缘域有无划分完毕. 若未划分完毕, 则通过式 (24)–式 (27), 在边缘域中选出重划分候选对象$\mathbf{x}_e$与$\mathbf{x}_s$; 否则结束算法.

5. 依据式 (29) 执行投票决策, 得出最终划分对象, 将其划入指定类簇的核心域, 同时删除存在于其他类簇边缘域的该对象.

6. 式 (31) 更新相似度矩阵, 判断迭代次数是否超过设定最大值 Q, 若超过则结束算法; 否则跳转至步骤 1.

3.2.1 初始化核心域

得到相似度矩阵后, 利用谱聚类可以得到一个原始的二支解. 三支聚类将原有的 n 个类簇的二支解$\{Z_1^*, Z_2^*, \cdots, Z_K^*\}$作为参考, 根据各个类簇的原始解确定核心域的初始空间$\{Co(Z_1), \cdots, Co(Z_K)\}$, 并且根据原始解的数量$\{m_1, m_2, \cdots, m_K\}$对初始核心域进行动态调整, 其数量通过系数 ω 决定, 设为 m_k 与 ω 的比值, 在一定程度上减少初始化的随机性.

初始化的核心域对象的选择依据为: 谱聚类过程中, 所有目标对象关于各自 K-means 聚类中心的距离. 其主要分为两个部分: 中心对象群与近中心对象群. 中心对象群由若干距离聚类中心最近的样本点组成, 其任务主要负责吸引核心样本点. 近中心对象群则由若干围绕聚类中心对象的样本点组成, 其主要负责吸引介于核心与边界域之间特征较为模糊的样本点.

3.2.2 扩展核心域与边缘域

本节将介绍扩展核心域与边缘域部分. 定义核心近邻 $IC(\mathbf{x}, cn)$, 边缘近邻 $IF(\mathbf{x}, fn)$ 分别为样本 **x** 在范围 cn 与 fn 个近邻的集合, **voteCo** 为 $1\times K$ 的矩阵, 用于储存各个类簇内对象的投票结果. 其中 $cn<fn$, 两个变量基于类簇个数 K 与样本总数 M 动态分配, 我们通过两个参数 θ_1 与 θ_2 来控制他们, 其定义主要如下:

$$cn = \left\lfloor \left\lfloor \frac{M}{K} \right\rfloor \times \frac{1}{\theta_1} \right\rfloor,\ fn = \left\lfloor \left\lfloor \frac{M}{K} \right\rfloor \times \frac{1}{\theta_2} \right\rfloor \tag{21}$$

对于待划分样本 **x** 与已划分样本$\mathbf{x}_j \in Co(Z_k)$, 有如下投票决策:

$$\begin{cases} (1)\ \text{if}\ (\mathbf{x} \in IC(\mathbf{x}_j, fn)\ \&\&\ \mathbf{x} \notin IC(\mathbf{x}_j, cn), \\ \text{then}\ \mathbf{voteCo}(1,k) = \mathbf{voteCo}(1,k) \\ (2)\ \text{if}\ (\mathbf{x} \in IC(\mathbf{x}_j, cn)\ \&\&\ \mathbf{x}_j \notin IC(\mathbf{x}, cn)), \\ \text{then}\ \mathbf{voteCo}(1,k) = \mathbf{voteCo}(1,k) \\ (3)\ \text{if}\ (\mathbf{x} \in IC(\mathbf{x}_j, cn)\ \&\&\ \mathbf{x}_j \in IC(\mathbf{x}, cn)), \\ \text{then}\ \mathbf{voteCo}(1,k) = \mathbf{voteCo}(1,k) + 1 \end{cases} \tag{22}$$

其中, 若符合条件 (1) 或条件 (2), 则将进入边缘域, 符合条件 (3) 将进入核心域, 均不符合者将在该阶段不作划分, 等待至下一轮再作划分.

该阶段将遍历所有类簇的核心域样本, 对当前待划分对象 **x** 进行投票决策. 对于某一类簇, 若其核心域内的任意对象投出正票, 则视为该类簇参与对当前待划分对象的竞争, 划分对象便有机会进入该类簇的核心域. 当多个类簇同时对某一待划分对象竞争时, 将依据类簇 Z_k 与待划分对象的投票分数函数 $Sc(Z_k)$ 来确定样本对象的划分:

$$\begin{cases} Sc(Z_k) = \dfrac{1}{|Co(Z_k)|} \sum_{x_j \in Co(Z_k)} \dfrac{W_{\cdot j} - \min(W_{\cdot j})}{\max(W_{\cdot j}) - \min(W_{\cdot j})} \\ x_j \in \bigcup_{k=k_1}^{k_n} Co(Z_k) \end{cases} \tag{23}$$

其中, $W_{\cdot j}$ 为待划分对象 $\mathbf{x}_i$ 与核心域样本 $\mathbf{x}_j$ 的相似度, $|Co(Z_k)|$为类簇 Z_k 核心域样本数量, $\{k_1, \cdots, k_n\}$为参与竞争的多个类簇的编号. 待分类对象将被划入投票分数最高的类簇.

3.2.3 边缘域对象的重新划分

在本节我们将讨论如何对位于边缘域的对象进行重新划分. 挑选合适的边缘域样本是该部分的关键. 其中, 考虑边缘域中最有可能隶属于核心域的对象, 即相似度最大的对象, 是一种常见的选择方案. 但基于最大相似度的采样策略存在局限性, 随着边缘域样本的逐渐减少, 最大相似度的有效性会逐渐下降. 基于不确定性的采样是一种适用性广泛的采样策略, 通过优先划分不确定性最大的样本来减少整体模型的熵, 从而得到相对优的解. 本文提出一种基于相似度判别与不确定性判别交替采样的方法, 同时继续通过投票来确定每轮迭代中的采样模式.

考虑边缘域样本$\mathbf{x}_t \in \sum_{k=1}^{K} Fr(Z_k)$属于类簇的隶属概率为:

$$p(\mathbf{x}_t \in Co(Z_k)) = \frac{\frac{1}{|Co(Z_k)|}\sum_{\mathbf{x}_j \in Co(Z_k)} W_{\cdot j}}{\sum_{n=1}^{K}\frac{1}{|Co(Z_n)|}\sum_{\mathbf{x}_j \in Co(Z_k)} W_{\cdot j}} \tag{24}$$

通过排序求解式 (25) 得到隶属某一类簇概率最大的边缘域样本 $\mathbf{x}_s$.

$$\mathbf{x}_s = \arg\max_{\mathbf{x}\in\cup_{i=1}^{K} Fr(Z_i)} p(\mathbf{x} \in Co(Z_k)),\ k \in [1, K] \tag{25}$$

得到每个边缘域样本的隶属概率后, 通过不确定性熵计算公式计算出其不确定度:

$$H(\mathbf{x}) = -\frac{1}{K}\sum_{k=1}^{K}(p(\mathbf{x} \in Co(Z_k))\log_2 p(\mathbf{x} \in Co(Z_k))) \tag{26}$$

通过排序求解式 (27) 得到隶属某一类簇概率最大的边缘域样本 $\mathbf{x}_e$.

$$\mathbf{x}_e = \arg\max_{\mathbf{x}\in\cup_{i=1}^{K} Fr(Z_i)} H(\mathbf{x}) \tag{27}$$

至此, 当前拥有了两个候选对象 $\mathbf{X}_e$ 与 $\mathbf{X}_s$. 同时, 根据式 (24) 结果, 按照隶属类簇概率 p 的降序排列, 记录 $\mathbf{X}_e$ 与 $\mathbf{X}_s$ 隶属类簇的序列 E 与序列 S:

$$E = \{Z_{e_1}, Z_{e_2}, \cdots, Z_{e_n}\},\ S = \{Z_{s_1}, Z_{s_2}, \cdots, Z_{s_n}\} \tag{28}$$

按照序列的顺序, 通过对每个核心域的对象进行问询投票, 来确定候选对象的选择. 引入一个 1×2 大小的投票统计矩阵 **reVote**, 其中 **reVote**{1, 1}部分为正票统计数, **reVote**{1, 2}部分为负票统计数, 对于候选对象 $\mathbf{x}$, 定义核心近邻 $RC(\mathbf{x}_j, rn)$ 为核心域 $\mathbf{A}_k$ 样本 $\mathbf{x}_j$ 在范围 rn 个近邻的集合, 投票问询的规则为:

$$\begin{cases} \text{if } (\mathbf{x} \in RC(\mathbf{x}_j, rn)), \text{ then } \mathbf{reVote}\{k,1\} = \mathbf{reVote}\{k,1\} + 1 \\ \text{if } (\mathbf{x} \notin RC(\mathbf{x}_j, rn)), \text{ then } \mathbf{reVote}\{k,2\} = \mathbf{reVote}\{k,2\} + 1 \end{cases} \tag{29}$$

其中, $\mathbf{x}_j \in \cup_{k=1}^{K} Co(Z_k)$. 对于某个类簇 Z_k, 候选对象 $\mathbf{x}$ 正票数量大于负票数量时, 即可与另一个候选对象进行票数比较, 否则将按照序列继续对下一个类簇进行投票问询. 当 $\mathbf{x}_e$ 与 $\mathbf{x}_s$ 获得各自的投票结果后, 基于二者的正票数量, 取票数较多者作为该次重划分的最终对象.

当 $\mathbf{x}_e$ 与 $\mathbf{x}_s$ 都被所有类簇否定时, 即:

$$\mathbf{reVote}\{k,1\} < \mathbf{reVote}\{k,2\}, 1 \leqslant k \leqslant K \tag{30}$$

当该情况出现时, 意味着当前边缘域内最大相似度指标的有效性已经完全丧失, 不再值得信任. 为了尽可能减少错误决策带来的影响, 将选择不确定性最大的候选样本 $\mathbf{X}_e$, 划入其隶属度最高的类簇 Z_{e_1} 中.

基于文献[19], 当边缘域的样本划入核心域后, 需要重新更新相似度矩阵. 将同类簇内的两个样本间的关系定义 ML, 非同类簇内的两个样本间的关系定义为 CL, 给出更新规则:

$$\begin{cases} \text{if } (x_i, x_j) \in ML, \text{ then } \mathbf{w}_{ij} = \mathbf{w}_{ji} = 1 \\ \text{if } (x_i, x_j) \in CL, \text{ then } \mathbf{w}_{ij} = \mathbf{w}_{ji} = 0 \end{cases} \tag{31}$$

4 实验

4.1 数据集与对比算法

为了验证本算法的性能, 我们采用 5 个真实数据集来验证, 分别是 Prokaryotic, 3-sources, Reuters, UCI Digits 与 Citeseer. 表 1 展示了这些数据的详细信息. 其中, 3-sources 与 Reuters 具备特征数量多, 样本数量少且数据稀疏的特点; 相反, UCI Digits 与 Prokaryotic 的样本数量较多但特征向量维度低; Citeseer 数据集拥有着数量相对近似的特征与样本数量. 通过对不同类型与特点的数据集进行实验测试, 可以更全面评估各算法的表现性能.

表 1 测试数据集

Data sets	Number of samples	Number of classes	Number of views
Prokaryotic	551	4	3
3-sources	169	6	3
Reuters	600	6	5
UCI Digits	2 000	10	3
Citeseer	3 312	2	6

Prokaryotic 数据集: 包含 551 种原核生物的数据集, 由数据文本与基因组[29]构成. 最大聚类样本数为 313 个, 最小聚类样本数为 35 个.

3-sources 数据集: 同时被 BCC, Reuters 和 Guardian 这 3 家报社刊登的新闻组成的数据集. 一共 169 个样本, 分属商业、政治、运动、健康、娱乐、科技 6 类领域.

Reuters 数据集: 由 5 种不同语言组成的文档功能数据集. 共分为 6 个类, 本文参照文献[3]选取标准, 选取了共 600 个数据样本.

UCI Digits 数据集: 取自荷兰公共服务地上 0–9 手写数字组成的数据集. 从 10 个类别中选取了 200 个样本, 由 3 个视图组成. 分别为 76 个字符形状的傅里叶系数、216 个轮廓相关的特征和 Karhunen-Love 系数.

Citeseer 数据集: 各类科学文章组成的数据集. 由两个视图组成, 分别是文本信息与文章引用关系.

基于上述数据集分别与如下对比算法进行比较.

(1) 基于协同正则化的多视图谱聚类 (co-regularized multi-view spectral clustering, Co-Reg)[30], 该算法通过两种正则化方法得到视角.

(2) 基于鲁棒的多视图谱聚类 (robust multi-view spectral clustering, RMSC)[31], 该算法通过构建低秩转移概率矩阵作为马尔科夫链的关键输入, 降低了各个视图数据中噪点对聚类性能的影响.

(3) 凸稀疏多视图谱聚类 (convex sparse multi-view spectral clustering, CSMSC)[32]. 该算法利用稀疏性对谱聚类进行了优化.

(4) 基于低秩稀疏约束的自权重多视角子空间聚类 (self-weighted multi-view subspace clustering with low-rank sparse constraint, SWMSC)[2], 一种新的基于低秩约束的自权重多视角聚类算法.

(5) 多视图稀疏子空间聚类 (centroid multi-view low-rank sparse subspace clustering, MLSSC)[3], 该算法通过低秩与稀疏约束, 构建共享各个视图的一致性矩阵来学习联合子空间表示的子空间聚类.

其中, 为了保证与本文正则化方案的一致性, 算法 1 与算法 4 均基于质心进行正则化. 通过对比原始算法, 传统算法与近几年最新的算法, 可以更加直观地体现出本文算法的性能表现.

4.2 评价指标

文本采用两个常见的指标来评价算法的性能, 分别是精确度 (accuracy, *ACC*) 与归一化互信息 (normalized mutual information, *NMI*).

ACC 表示聚类的准确度. N 表示样本数, P_i 是第 i 个样本聚类产生的标签, 其真实标签是 y_i. 当且仅当 $x=y$ 时, $\delta(x,y)$ 等于 1, 否则等于 0.

$$ACC = \frac{1}{N}\sum_{i}^{N}\delta(y_i, map(P_i)) \tag{32}$$

NMI 表示两个聚类成果的相近程度.

$$NMI(Y,C) = \frac{2\times I(Y;C)}{H(Y)+H(C)} \tag{33}$$

其中, Y 表示数据真实的类别, C 表示聚类结果, $H(\cdot)$ 表示交叉熵.

4.3 参数设置

多视图低秩稀疏子空间聚类中, 低秩参数 β_1, 稀疏参数 β_2, 视图间一致性参数 λ 与惩罚系数 μ 对于不同的数据集, 需要做出相应调整. 其中, 其针对 UCI Digits, 3-sources, Reuters 这 3 个数据集的参数调整在文献[5]中给出, 分别设定为: {0.9, 0.1, 0.7, 10}, {0.1, 0.9, 0.7, 10^2}, {0.5, 0.5, 0.3, 10^4}.

在三支决策中, 初始核心域系数 ω, 核心近邻系数 θ_1, 边界近邻系数 θ_2 以及最大迭代次数 Q 依据数据集 Prokaryotic, 3-sources, Reuters 设定为: {15, 0.8, 1.6, 15}, {30, 2.2, 2.6, 20}, {8.8, 2, 10, 100}.

初始核心域系数 ω 影响着初始聚类空间与原低秩稀疏聚类算法结果的相似度. 最大迭代次数 Q 在一定程度上影响着核心域样本与边缘域样本在算法结束时的数量. 核心近邻系数 θ_1 控制进入核心域的严苛程度, 当其较大时, 会提高算法的运行时间. 边缘近邻系数 θ_2 影响着算法过程中边缘域的数量, 在一定程度上影响着算法结果的波动程度.

4.4 实验结果与分析

对每一组数据的测试, 本文的算法 (TW-MLSSC) 与多视图稀疏子空间聚类算法都将参数调整至表现性能最好, 运行 10 次获得评价指标, 并计算各个评价指标的平均值以及方差.

从表 2 与表 3 可以看出, 在 3-sources 与 Reuters 这类特征向量多且样本数据较为稀疏的数据集上, 本文算法的性能相较于原算法有所提升. 但在 UCI Digits 与 Prokaryotic 这些特征数量相对样本数量较少的数据集上, 性能表现反而有所下降; 相较于 RSCM 与 CSMSC 算法, 本算法总体表现出了更好的性能. 与目前较新的多视图聚类算法相比, 在部分数据集上有着更优异的表现.

从整体结果进行分析, 本文设计的基于投票的三支决策聚类在特征向量、样本数量稀疏的实验测试数据集上, 对原算法会有较大的正向提升; 但当特征向量数量相对样本数量较少时, 提升效果并不理想. 这一结果也符合实际情况: 三支决策聚类的思想是在缺乏足够信息的情况下, 延迟决策的进行, 来避免高风险决策的发生. 若延迟决策后依旧无法有足够的信息支撑, 算法的实际效果便会大打折扣, 在实验中的具体表现为实验数据集样本特征数量相对样本数量较少的情况. 反之, 在样本特征向量较多时, 延迟决策所带来的收益便会相对明显, 这是因为有了足够且有效的信息为聚类提供决策支持.

表 2 对比算法在不同数据集上的 *ACC*

数据集	Prokaryotic	3-sources	Reuters	UCI Digits	Citeseer
Co-Reg	0.459 (0.010)	0.505 (0.032)	0.362 (0.017)	0.754 (0.067)	0.461 (0.107)
RMSC	0.447 (0.027)	0.477 (0.033)	0.361 (0.019)	0.742 (0.070)	0.451 (0.013)
CSMSC	0.462 (0.026)	0.482 (0.026)	0.365 (0.005)	0.775 (0.045)	0.477 (0.014)
SWMSC	0.593 (0.019)	**0.702 (0.027)**	0.453 (0.003)	0.831 (0.046)	0.614 (0.008)
MLSSC	**0.605 (0.026)**	0.654 (0.042)	0.432 (0.010)	**0.835 (0.047)**	0.593 (0.021)
TW-MLSSC	0.586 (0.024)	0.687 (0.048)	**0.457 (0.019)**	0.802 (0.058)	**0.623 (0.011)**

表 3 对比算法在不同数据集上的 *NMI*

数据集	Prokaryotic	3-sources	Reuters	UCI Digits	Citeseer
Co-Reg	0.296 (0.018)	0.514 (0.026)	0.291 (0.014)	0.783 (0.033)	0.170 (0.004)
RMSC	**0.315 (0.041)**	0.517 (0.024)	0.297 (0.018)	0.778 (0.040)	0.151 (0.001)
CSMSC	0.269 (0.022)	0.518 (0.026)	0.295 (0.020)	0.819 (0.019)	0.213 (0.002)
SWMSC	0.253 (0.013)	0.574 (0.027)	0.273 (0.012)	0.812 (0.024)	0.254 (0.017)
MLSSC	0.297 (0.015)	0.579 (0.066)	0.376 (0.017)	**0.854 (0.023)**	0.226 (0.021)
TW-MLSSC	0.246 (0.012)	**0.651 (0.030)**	**0.378 (0.014)**	0.791 (0.018)	**0.273 (0.011)**

图 2 和图 3 给出了不同边缘近邻参数 θ_2 参数设定下 3-sources 中核心近邻参数 θ_1 对 3-sources 算法性能的影响, Y 轴为本文算法与二支决策的多视图稀疏子空间算法在评价指标上的差值. 可以看出, 本文算法性能对参数 θ_1 与 θ_2 的设置较为敏感. 算法不同参数的设置下, 在某些局部区间均出现较好算法表现. 同时, 也在某些区间表现出截然不同的算法性能, 说明了不恰当的参数设置会对算法的性能带来负提升.

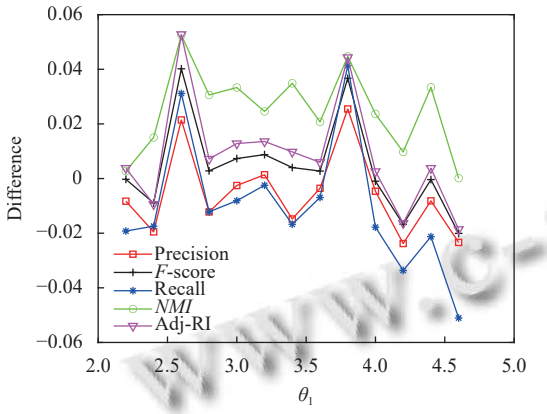

图 2 3-sources 核心近邻参数 θ_1 对算法性能的影响 {ω=30, Q=20, θ_2=2.2}

5 结论与展望

为处理高维多视图数据, 本文采用一种低秩表示的稀疏子空间聚类算法. 稀疏子空间聚类保证了在其子空间内每个聚类视角用尽可能少的数据点来线性表示, 低秩表示用来减少数据冗余, 降低噪声的干扰问题. 此外, 本文采用三支决策思想, 设计了一种基于投票的三支决策方法, 将其应用于多视图低秩稀疏子空间聚类. 从实验结果来看, 本文提出的算法在一些特定数据集上有效提高了聚类的性能, 证明了“延迟决策”这一三支决策的主要思想应用在聚类领域的可行性与积极作用.

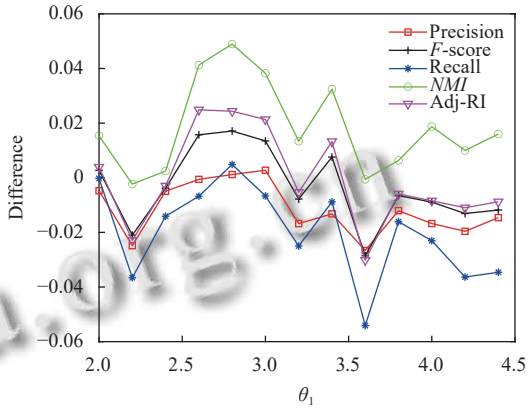

图 3 3-sources 核心近邻参数 θ_1 对算法性能的影响 {ω=30, Q=20, θ_2 =2}

该算法也存在着缺陷, 例如在特征向量较少的数据集中表现较差, 甚至造成了性能的负提升; 参数选取的区间不稳定等. 在未来的工作中, 我们将侧重于提高算法的效率, 同时借鉴其他多视图聚类算法的思想, 对现有的算法模型进行优化. 此外, 尝试探究近邻系数对于三支决策聚类更深层次的影响与其背后的运行机制.

参考文献

1 Chao GQ, Sun SL, Bi JB. A survey on multiview clustering.

IEEE Transactions on Artificial Intelligence, 2021, 2(2): 146–168. [doi: 10.1109/tai.2021.3065894]
2 夏菁, 丁世飞. 基于低秩稀疏约束的自权重多视角子空间聚类. 南京大学学报 (自然科学), 2020, 56(6): 862–869. [doi: 10.13232/j.cnki.jnju.2020.06.008]
3 Brbić M, Kopriva I. Multi-view low-rank sparse subspace clustering. Pattern Recognition, 2018, 73: 247–258. [doi: 10.1016/j.patcog.2017.08.024]
4 Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Vancouver: MIT Press, 2001. 849–856.
5 Shi JB, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888–905. [doi: 10.1109/34.868688]
6 Wang YX, Xu H, Leng CL. Provable subspace clustering: When LRR meets SSC. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 64–72.
7 Liu GC, Lin ZC, Yu Y. Robust subspace segmentation by low-rank representation. Proceedings of the 27th International Conference on International Conference on Machine Learning. Haifa: Omnipress, 2010. 663–670.
8 Vidal R, Favaro P. Low rank subspace clustering (LRSC). Pattern Recognition Letters, 2014, 43: 47–61. [doi: 10.1016/j.patrec.2013.08.006]
9 Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(11): 2765–2781. [doi: 10.1109/TPAMI.2013.57]
10 Tian L, Du Q. Parallel multi-view low-rank and sparse subspace clustering for unsupervised hyperspectral image classification. Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. Honolulu: IEEE, 2018. 618–621. [doi: 10.23919/APSIPA.2018.8659796]
11 王丽娟, 丁世飞, 夏菁. 基于多样性的多视图低秩稀疏子空间聚类算法. 智能系统学报, 2023, 18(2): 399–407. [doi: 10.11992/tis.202110026]
12 Yao YY. Three-way decision: An interpretation of rules in rough set theory. Rough Sets and Knowledge Technology. Berlin: Springer, 2009. 642–649.
13 Yu H, Jiao P, Yao YY, *et al*. Detecting and refining overlapping regions in complex networks with three-way decisions. Information Sciences, 2016, 373: 21–41. [doi: 10.1016/j.ins.2016.08.087]
14 Yu H, Zhang C, Wang GY. A tree-based incremental overlapping clustering method using the three-way decision theory. Knowledge-based Systems, 2016, 91: 189–203. [doi: 10.1016/j.knosys.2015.05.028]
15 Yu H. A framework of three-way cluster analysis. Proceedings of the 2017 International Joint Conference on Rough Sets. Olsztyn: Springer, 2017. 300–312. [doi: 10.1007/978-3-319-60840-2_22]
16 于洪, 毛传凯. 基于 K-means 的自动三支决策聚类方法. 计算机应用, 2016, 36(8): 2061–2065, 2091. [doi: 10.11772/j.issn.1001-9081.2016.08.2061]
17 夏月月, 张以文. 一种融合三支决策理论的改进 K-means 算法. 小型微型计算机系统, 2020, 41(4): 724–731. [doi: 10.3969/j.issn.1000-1220.2020.04.009]
18 徐天杰, 王平心, 杨习贝. 基于人工蜂群的三支 K-means 聚类算法. 计算机科学, 2023, 50(6): 116–121. [doi: 10.11896/jsjkx.220800150]
19 Yu H, Wang XC, Wang GY, *et al*. An active three-way clustering method via low-rank matrices for multi-view data. Information Sciences, 2020, 507: 823–839. [doi: 10.1016/j.ins.2018.03.009]
20 Yu H, Zhou QF. A cluster ensemble framework based on three-way decisions. Proceedings of the 8th International Conference on Rough Sets and Knowledge Technology. Halifax: Springer, 2013. 302–312.
21 Yao YY. Three-way decisions and cognitive computing. Cognitive Computation, 2016, 8(4): 543–554. [doi: 10.1007/s12559-016-9397-5]
22 Yu H, Zhang C, Hu F. An incremental clustering approach based on three-way decisions. Proceedings of the 9th International Conference on Rough Sets and Current Trends in Computing. Granada: Springer, 2014. 152–159.
23 Boyd S, Parikh N, Chu E, *et al*. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, 2011, 3(1): 1–122. [doi: 10.1561/2200000016]
24 Hong MY, Luo ZQ. On the linear convergence of the alternating direction method of multipliers. Mathematical Programming, 2017, 162(1): 165–199. [doi: 10.1007/s10107-016-1034-2]
25 Cai JF, Candès EJ, Shen ZW. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 2010, 20(4): 1956–1982. [doi: 10.1137/080738970]
26 Donoho DL. De-noising by soft-thresholding. IEEE

Transactions on Information Theory, 1995, 41(3): 613–627. [doi: 10.1109/18.382009]

27 Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics, 2004, 57(11): 1413–1457. [doi: 10.1002/cpa.20042]

28 胡凌超, 于洪. 一种基于投票的三支决策聚类集成方法. 计算机应用, 2016, 37(8): 1741–1745.

29 Brbić M, Piškorec M, Vidulin V, *et al*. The landscape of microbial phenotypic traits and associated genes. Nucleic Acids Research, 2016, 44(21): 10074–10090. [doi: 10.1093/nar/gkw964]

30 Kumar A, Rai P, Daumé H. Co-regularized multi-view spectral clustering. Proceedings of the 24th International Conference on Neural Information Processing Systems. Granada: Curran Associates Inc., 2011. 1413–1421.

31 Xia RK, Pan Y, Du L, *et al*. Robust multi-view spectral clustering via low-rank and sparse decomposition. Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec: AAAI Press, 2014. 2149–2155.

32 Lu CY, Yan SC, Lin ZC. Convex sparse spectral clustering: Single-view to multi-view. IEEE Transactions on Image Processing, 2016, 25(6): 2833–2843. [doi: 10.1109/TIP.2016.2553459]

(校对责编: 牛欣悦)