

解模糊约束软子空间聚类

赵 洋, 潘兴广

贵州民族大学数据科学与信息工程学院, 贵州 贵阳

收稿日期: 2026年9月23日; 录用日期: 2026年10月16日; 发布日期: 2026年10月22日

摘 要

在现有的大多数软子空间聚类算法中都是基于簇内的信息开发的,忽略了簇间信息,造成信息严重丢失,算法性能受到较大程度的限制。本文在现有的软子空间聚类算法的目标函数上集成了簇内的信息以及簇间信息,同时为了提升算法的鲁棒性,我们采用解模糊化约束权重策略,结合模糊加权以及熵加权的思想构造了一个新的权重约束函数,据此构建了一个新的目标函数,并推导了DCSSC (Defuzzification Constraint Soft Subspace Clustering)算法。利用人工合成数据集以及UCI数据集进行实验,对该算法的性能评估。在与几种现有的软子空间聚类算法的对比实验中,DCSSC算法在性能和稳定性上均表现出优势。最后利用Friedman检验验证了新提出的算法的性能与现有的软子空间的性能存在显著性差异,并且这种差异具有统计意义。

关键词

软子空间聚类, 解模糊化, 权重约束, 特征加权

Defuzzification Constraint Soft Subspace Clustering

Yang Zhao, Xingguang Pan

School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang Guizhou

Received: September 23, 2026; accepted: October 16, 2026; published: October 22, 2026

Abstract

The development of most existing soft subspace clustering algorithms is based on information within clusters, while information between clusters is ignored, resulting in severe information loss and considerably limiting the performance of the algorithms. In this paper, information within clusters and information between clusters are integrated into the objective function of existing soft subspace clustering algorithms. Meanwhile, to improve the robustness of the algorithm, we adopt a defuzzification

constraint weight strategy and combine the ideas of fuzzy weighting and entropy weighting to construct a new weight constraint function. On this basis, a new objective function is constructed, and the DCSSC (Defuzzification Constraint Soft Subspace Clustering) algorithm is derived. Experiments were conducted using artificial synthetic datasets and UCI datasets to evaluate the performance of the algorithm. In comparative experiments with several existing soft subspace clustering algorithms, the DCSSC algorithm shows advantages in both performance and stability. Finally, the Friedman test is used to verify that the performance of the newly proposed algorithm is significantly different from that of existing soft subspace algorithms, and that this difference is statistically significant.

Keywords

Soft Subspace Clustering, Defuzzification, Weighting Constraint, Feature Weighting

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着人工智能与机器学习的发展, 聚类技术在对数据进行处理上有着重要的作用。聚类算法主要分为硬聚类或清晰聚类和软聚类也称模糊聚类。人们熟知的硬聚类算法有 K-means 算法。K-means 算法具有简单高效, 空间划分明确, 适应大规模数据的场景的优点。但是它的不足也十分明显, 由于 K-means 算法初始的聚类中心是随机选择的, 选择的初始聚类中心不同使得最后的聚类结果也会存在较大差异, 这就让 K-means 算法对于初始聚类中心的选择比较依赖, 所以导致算法的稳定性较差。模糊聚类在处理那些模糊的或不确定的数据时起着非常关键的作用。在过去的几十年里, 聚类算法广泛应用于, 模式识别和图像处理等领域[1]-[6]。

Kuo-Lung Wu 等人在 FS 算法的基础上对其目标函数进行修正, 在最小化簇内紧度的同时最大化簇间的分离度, 并推导出了 FCS 算法[7]。Lin Zhu 等人在 IFP-FCM 算法的基础进行改进, 将 IFP-FCM 算法目标函数中的隶属度约束函数替换为 $f(\mu_{ij}) = \sum_{j=1}^c \mu_{ij} (1 - \mu_{ij}^{m-1})$, 据此构建了新的目标函数, 并推导出了 GIFP-FCM 算法[8]。相较于 IFP-FCM 算法的隶属度模糊系数固定为 2, GIFP-FCM 算法的隶属度模糊系数则可以根据不同的数据集灵活改变。

尽管传统的模糊聚类算法可以在大多情况下表现出良好的聚类性能, 但是在面对高维空间聚类时, 传统的模糊聚类技术的性能就存在严重的不足[9]。因为传统的聚类技术主要是通过数据点之间距离衡量它们的相似程度, 然而在现实中的数据集中大多都是多特征的数据集, 可能会出现同一类中任意给定一对数据点可能在某些特征子集上的距离会比较远, 而另外一类的数据点在这些特征子集上的距离会比较近的情况。所以在许多现实中的多维度数据集中, 不同类的数据点通常与不同的特征子集相关, 即不同类的数据点存在于不同的特征子空间[10]。传统的聚类算法无法找到每个类别对应的特征子空间, 这也是传统聚类算法的局限性所在。

为了克服这个困难, 研究人员提出子空间聚类, 并且在近些年来成为研究热点。子空间聚类的主要思想是在同一个数据集中, 在不同的子空间去定位集群。子空间聚类分为两个主要的方法, 分别是“硬子空间聚类”和“软子空间聚类”。硬子空间聚类方法是为一数据集集中的不同类的集群去识别精确的

子空间。硬子空间聚类方法主要有两种的搜索子空间的方式, 分别是“自上而下”和“自下而上”的方法, 相应的硬子空间聚类算法都是基于这两个方法开发出来的。其中基于“自下而上”方法开发的硬子空间聚类算法包括: R. Agrawal 等人提出的 CLIQUE 算法[11]; C. H. Cheng 等人提出的 MAFIA 算法[12]; S. Goil, 等人提出的 ENCLUS 算法[13]。基于“自上而下”方法开发的硬子空间聚类算法包括: ORCLUS 算法[14], PROCLUS 算法[15], DOC 算法[16]。不同于硬子空间聚类能精确识别子空间, 在软子空间聚类过程中, 不同的特征被赋予不同的权重以此衡量不同的特征对特定的集群信息贡献。换句话说, 每个特征对不同集群的信息贡献是不同的。每个子空间可以通过聚类后的权重的值来识别。软子空间聚类算法在近年来成为研究热点, 许多软子空间聚类算法被研究人员开发出来。我们将在第二小节中详细介绍几种经典算法。本文内容安排如下:

- (1) 第二节介绍相关软子空间聚类算法及其原理, 分析其存在的不足, 并给出针对性的解决方案。
- (2) 第三节详细阐述本文提出的 DCSSC 算法。
- (3) 第四节为实验部分, 我们在人工数据集和 UCI 数据集上的实验, 验证 DCSSC 算法的优越性和有效性。
- (4) 最后通过 Friedman 检验分析得出这六种算法在 NMI, RI 指标上存在显著差异, 且这种差异具有统计学意义。

2. 相关工作

现有的大多数软子空间算法的目标函数通常包括加权距离项和正则化项, 旨在最小化数据点与所属簇中心的加权距离, 同时约束权重的分配。设 $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ 典型的形式如下:

$$J(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \text{wdist}(\mathbf{x}, \mathbf{v}) + \gamma P(\mathbf{w}_{jk}) \quad (1)$$

其中, m 表示模糊指数(一般情况 $m > 1$, 控制隶属度的模糊性); τ 表示权重指数(控制权重分布的稀疏性); γ 表示正则化系数, $R(\mathbf{w}_{jk})$ 表示惩罚项。基于以上目标函数的一般形式, 研究人员构造许多新型的目标函数, 并推导出性能优异且应用广泛的软子空间聚类算法。本节我们介绍了五种较为经典的软子空间聚类算法。

首先, 我们对 G. J. Gan, J. H. Wu 等人提出的 FSC (Fuzzy Subspace Clustering) [17] 算法进行简单的介绍。该算法的目标函数如下所示:

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}, \mathbf{W}) &= \varepsilon \sum_{j=1}^c \sum_{k=1}^d w_{jk}^\tau + \sum_{j=1}^c \sum_{i=1}^n u_{ij} \sum_{k=1}^d w_{jk}^\tau (x_{ik} - v_{jk})^2 \\ &= \sum_{j=1}^c \sum_{k=1}^d w_{jk}^\tau \left(\sum_{i=1}^N u_{ij} (x_{ik} - v_{jk})^2 + \varepsilon \right) \\ \text{s.t. } u_{ij} &\in \{0, 1\}, \sum_{j=1}^c u_{ij} = 1, 0 < \sum_{i=1}^N u_{ij} < N, 0 < w_{jk} < 1, \sum_{k=1}^d w_{jk} = 1 \end{aligned} \quad (2)$$

这里 $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c]$, $\mathbf{U} = [u_{ij}]_{C \times N}$ 分别表示每个集群的聚类中心矩阵和硬划分矩阵; C, D, N 分别表示数据集中集群的数量, 数据集特征的个数, 以及数据点的数量; 每一个集群分配的权重向量是不同的, w_{jk} 表示分配给第 j 个集群的第 k 个维度的权重。该算法是根据各维度对每个集群的信息贡献的多少分配不同的权重, 在目标函数中的 ε 表示的是一个很小的数, 它的作用是避免在聚类过程中因某个维度的离散度为零造成计算变得困难。

与 FSC 算法控制权重的分配不同, L. Jing 等人提出了 EWKM (Entropy Weighting K-means) 算法, 该

算法是利用信息熵控制权重的分配[18], 它的目标函数如下所示:

$$J(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{j=1}^c \sum_{i=1}^n u_{ij} \sum_{k=1}^d w_{jk} (x_{ik} - v_{jk})^2 + \gamma \sum_{j=1}^c \sum_{k=1}^d w_{jk} \log w_{jk} \quad (3)$$

$$\text{s.t. } u_{ij} \in \{0, 1\}, \sum_{j=1}^c u_{ij} = 1, 0 < \sum_{j=1}^c u_{ij} < N, 0 \leq w_{ij} \leq 1, \sum_{k=1}^d w_{jk} = 1$$

该算法通过引入信息熵, 对于那些在识别类别过程中贡献大的特征维度赋予更大的权重值, 而贡献较小的特征维度则被赋予更小的权重, 利用信息熵控制权重分配的算法除了 EWKM 算法外, Zhou, Jin 等人认为加权模糊聚类是一个最大熵推理问题, 旨在寻找全局规律性, 并从现有的数据中获得最平滑的重构, 据此提出了 EWFCM (Maximum Entropy Regularized Weighted Fuzzy C-means) 算法[19], 该算法结合了不相似度量和引入特征熵加权正则化项, 提出了一个新的目标函数, 如下所示:

$$J(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk} (x_{ik} - v_{jk})^2 + \gamma^{-1} \sum_{j=1}^c \sum_{k=1}^d w_{jk} \ln w_{jk} \quad (4)$$

$$\text{s.t. } \sum_{j=1}^c u_{ij} = 1, 0 \leq u_{ij} \leq 1, \sum_{k=1}^d w_{jk} = 1, 0 \leq w_{jk} \leq 1$$

其中 m 为模糊指数, 参数 $\gamma > 0$ 。

除了 EWKM 算法和 EWFCM 算法考虑了信息熵以外, LAC (Local Adaptive Clustering) 算法[20]同样考虑熵。LAC 算法的目标函数如下所示:

$$J(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{j=1}^c \sum_{k=1}^d w_{jk} X_{jk} + \gamma \sum_{j=1}^c \sum_{k=1}^d w_{jk} \ln w_{jk} \quad (5)$$

$$\text{s.t. } u_{ij} \in \{0, 1\}, \sum_{i=1}^n u_{ij} = 1, 0 < \sum_{i=1}^n u_{ij} < N \text{ and } \sum_{k=1}^d w_{jk} = 1$$

其中 $X_{jk} = \frac{\sum_{i=1}^n u_{ij} (x_{ik} - v_{jk})^2}{\sum_{i=1}^n u_{ij}}$, 这里 $\gamma > 0$ 是该算法中的一个参数, 它控制着权重之间的相对差异。

以上介绍的四个算法主要是通过模糊加权以及熵加权两种方式分配权重, 除了这两种分配权重的方式我们接下来介绍的算法也称为 LAC (Local Adaptive Clustering) [21] 算法, 该算法不同于前面提到的基于熵的 LAC 算法, 为避免混淆我们将这里介绍的 LAC 算法暂时命名为 LAC1, 前面提到的 LAC 算法暂时命名为 LAC2 算法。LAC1 算法的目标函数如下所示:

$$J(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \sum_{j=1}^c \sum_{k=1}^d w_{jk} \exp(-X_{jk}) \quad (6)$$

$$\text{s.t. } u_{ij} \in \{0, 1\}, \sum_{j=1}^c u_{ij} = 1, 0 < \sum_{i=1}^n u_{ij} < n \text{ and } \sum_{k=1}^d w_{jk}^2 = 1$$

其中 $X_{jk} = \frac{\sum_{i=1}^n u_{ij} (x_{ik} - v_{jk})^2}{\sum_{i=1}^n u_{ij}}$, 与常规算法的优化目标不同, 该算法的优化目标是最大化目标函数。

上述的软子空间聚类算法被开发并且得到了广泛的应用, 但是这些算法的性能也有值得改进的地方。大多算法都是基于簇内的紧密程度开发的, 忽略了一些重要的信息, 如: 聚类之间的分离度, 使得算法的性能不能得到有效的提升。出于对这个问题的考虑, 我们结合了簇内的紧密性以及簇间的分离度, 开

发了一种改进模糊加权子空间聚类算法。

3. DCSSC 算法

算法描述

第 2 节中介绍的软子空间聚类算法可以看出现有的大多数软子空间聚类算法都是基于簇内信息开发得到的, 一些重要的区分信息并未得到有效利用。其中, 簇间的信息对聚类性能有着重要影响, Wu 等人利用簇内紧度与簇间分离度开发了一种新的模糊聚类方法。含有 K 个簇的数据集的模糊簇内散点矩阵和模糊簇间散点矩阵[7]的表示如下:

$$\mathbf{S}_{FW} = \sum_{j=1}^c \sum_{i=1}^n \mu_{ij}^m (x_i - v_j)(x_i - v_j)^t \quad (7)$$

$$\mathbf{S}_{FB} = \sum_{j=1}^c \sum_{i=1}^n \mu_{ij}^m (v_j - v_0)(v_j - v_0)^t \quad (8)$$

其中, $\mathbf{S}_{FW}$ 表示模糊簇内散点矩阵, $\mathbf{S}_{FB}$ 表示模糊簇间散点矩阵。相应模糊簇内紧凑度 J_C 和模糊簇间分离度 J_S 可表示为如下形式:

$$J_C = \sum_{j=1}^c \sum_{i=1}^n \mu_{ij}^m \|x_i - v_j\|^2 = \text{tr}(\mathbf{S}_{FW}) \quad (9)$$

$$J_S = \sum_{j=1}^c \sum_{i=1}^n \mu_{ij}^m \|v_j - v_0\|^2 = \text{tr}(\mathbf{S}_{FB}) \quad (10)$$

类似地可以将其拓展到模糊加权簇内紧度 J_{C_fw} 以及模糊加权簇间分离度 J_{S_fw} , 表示形式如下:

$$J_{C_fw} = \sum_{j=1}^c \left(\sum_{i=1}^n \mu_{ij}^m \right) \sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 \quad (11)$$

$$J_{S_fw} = \sum_{j=1}^c \left(\sum_{i=1}^n \mu_{ij}^m \right) \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2 \quad (12)$$

为了降低维度灾难造成算法性能降低的影响, 我们采用去模糊化约束策略并构造了一个权重约束函数 $f(w_{jk}) = \sum_{k=1}^d w_{jk} (1 - w_{jk}^{r-1})$, 据此构建出新的目标函数:

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}, \mathbf{W}) &= J_{C_fw} - \eta J_{S_fw} + \gamma \sum_{j=1}^c \sum_{k=1}^d w_{jk} (1 - w_{jk}^r) \\ &= \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 + \gamma \sum_{j=1}^c \sum_{k=1}^d w_{jk} (1 - w_{jk}^{r-1}) \\ &\quad - \eta \sum_{j=1}^c \left(\sum_{i=1}^n u_{ij}^m \right) \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2 \\ \text{s.t. } 0 &\leq u_{ij} \leq 1, \sum_{j=1}^c u_{ij} = 1, 0 \leq w_{jk} \leq 1, \sum_{k=1}^d w_{jk} = 1 \end{aligned} \quad (13)$$

这里 $\mathbf{U} = (u_1, u_2, \dots, u_n)$ 表示模糊划分矩阵, $\mathbf{V} = (v_1, v_2, \dots, v_c)$ 表示聚类中心矩阵, $\mathbf{W} = (w_1, w_2, \dots, w_c)$ 表示权重矩阵。 c , n , d 分别表示聚类数, 样本量, 维度的数量, x_{ik} 表示第 i 个样本的第 k 个特征; v_{jk} 表示第 j 个簇中心的第 k 个特征; w_{jk} 表示第 j 个类别的第 k 个特征; u_{ij} 表示第 i 个样本属于第 j 个类别的隶属度。据此推导出 DCSSC (Defuzzification Constraint Soft Subspace Clustering) 算法。我们提出的解模糊化约束软子空间聚类算法由于引入了权重约束函数的缘故, 使得特征的权重更加集中。对那些信息贡

献较大的维度赋予更大的权重, 那些信息贡献较小的维度赋予较小的权重, 不仅极大地利用有用的信息, 更是降低了冗余特征造成的干扰。

定理 1: 当 $\mathbf{V} = (v_1, v_2, \dots, v_C)$ 和 $\mathbf{W} = (w_1, w_2, \dots, w_C)$ 固定时, 若 $\mathbf{U}$ 满足以下条件, 目标函数达到最小值:

$$u_{ij} = \frac{(d_{ij})^{\frac{1}{1-m}}}{\sum_{j=1}^C (d_{ij})^{\frac{1}{1-m}}} \quad (14)$$

其中, $d_{ij} = \sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 - \eta \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2$ 。为保证 d_{ij} 是非负, 需要对 η 作出一些限制。即

$$\sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 > \eta \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2。因此可以制定自适应规则为 $\eta = \min \left\{ \min_j \left(\frac{\sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2}{\sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2} \right), \eta \right\}$,$$

$\forall i=1, 2, 3, \dots, N$ 。如果 η 大于 $\min_j \left(\frac{\sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2}{\sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2} \right)$, 则 η 由后者代替; 反之, 则 η 的值不变。这样总能

保证 d_{ij} 非负。

证明: 根据 Lagrange 乘子法, 式(14)的有约束优化问题转化下面的无约束优化问题

$$\begin{aligned} L(\mathbf{U}, \mathbf{V}, \mathbf{W}, \boldsymbol{\lambda}_1) = & \sum_{j=1}^C \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 + \gamma \sum_{j=1}^C \sum_{k=1}^d w_{jk} (1 - w_{jk}^{r-1}) \\ & - \eta \sum_{j=1}^C \left(\sum_{i=1}^n u_{ij}^m \right) \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2 - \sum_{i=1}^n \lambda_{1i} \left(\sum_{j=1}^C u_{ij} - 1 \right) \end{aligned} \quad (15)$$

其中, $\boldsymbol{\lambda}_1 = (\lambda_{11}, \lambda_{12}, \dots, \lambda_{1n})$ 表示模糊划分矩阵 $\mathbf{U} = (u_1, u_2, \dots, u_N)$ 的拉格朗日乘子向量, 对 $\mathbf{U}$ 求偏导, 并且

令 $\frac{\partial L}{\partial u_{ij}} = 0$, 得到如下的式子:

$$\frac{\partial L}{\partial u_{ij}} = m u_{ij}^{m-1} \left(\sum_{k=1}^d w_{jk}^r (x_{ik} - v_{jk})^2 - \eta \sum_{k=1}^d w_{jk}^r (v_{jk} - v_{0k})^2 \right) - \lambda_{1i} = 0 \quad (16)$$

又

$$\sum_{j=1}^C u_{ij} = 1 \quad (17)$$

联立式(16)和式(17)即得: $u_{ij} = \frac{(d_{ij})^{\frac{1}{1-m}}}{\sum_{j=1}^C (d_{ij})^{\frac{1}{1-m}}}$ 证毕。

定理 2: 表明了当 $\mathbf{U} = (u_1, u_2, \dots, u_N)$ 和 $\mathbf{W} = (w_1, w_2, \dots, w_C)$ 固定时, 式(18)为目标函数的达到最小值的必要条件

$$v_{jk} = \frac{\sum_{i=1}^n u_{ij}^m (x_{jk} - \eta v_{0k})}{\sum_{i=1}^n u_{ij}^m (1 - \eta)} \quad (18)$$

证明: 将 $\mathbf{U}=(u_1, u_2, \dots, u_N)$ 和 $\mathbf{W}=(w_1, w_2, \dots, w_C)$ 固定, 可转换为如下的优化问题:

$$L(\mathbf{V}) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk}^\tau (x_{ik} - v_{jk})^2 - \eta \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk}^\tau (v_{jk} - v_{0k})^2 \quad (19)$$

对 v_{jk} 求导, 并且令 $\frac{\partial L}{\partial v_{jk}} = 0$, 可以得到下面的式子:

$$\frac{\partial L}{\partial v_{jk}} = w_{jk}^\tau \sum_{i=1}^n u_{ij}^m \left[(x_{ik} - v_{jk})^2 - \eta (v_{jk} - v_{0k})^2 \right] = 0 \quad (20)$$

$$\text{经化简即得 } v_{jk} = \frac{\sum_{i=1}^n u_{ij}^m (x_{ik} - \eta v_{0k})}{\sum_{i=1}^n u_{ij}^m (1 - \eta)} \text{ 证毕。}$$

定理 3: 当 $\mathbf{U}=(u_1, u_2, \dots, u_N)$ 和 $\mathbf{V}=(v_1, v_2, \dots, v_C)$ 固定时, 如果 w_{jk} 满足以下条件则等式达到局部最小值:

$$w_{jk} = \frac{(\delta_{jk} - \gamma)^{\frac{1}{1-\tau}}}{\sum_{k'=1}^D (\delta_{jk'} - \gamma)^{\frac{1}{1-\tau}}} \quad (21)$$

其中, $\delta_{jk} = \sum_{i=1}^n u_{ij}^m \left[(x_{ik} - v_{jk})^2 - \eta (v_{jk} - v_{0k})^2 \right]$ 当 $\mathbf{U}$ 和 $\mathbf{V}$ 固定时, 式(22)为等式(14)达到局部最小值的必要条件。这里需要注意的是, 需适当设置参数 γ 的值, 如果 γ 的值过大, 则 δ_{jk} 可能会出现负值, 这与约束条件中 $0 \leq w_{jk} \leq 1$ 相违背, 为了确保得到权重有意义, 因此我们在设置参数时需要对参数的设置进行约束, 确保 δ_{jk} 非负, 避免最后算出来的权重没有意义, 利用等式(23)的规则对参数 γ 进行约束:

$$\gamma = \min \left\{ \min_k \sum_{i=1}^n u_{ij}^m \left[(x_{ik} - v_{jk})^2 - \eta (v_{jk} - v_{0k})^2 \right], \gamma \right\} \quad (22)$$

如果我们设置的参数 γ 大于 $\min_k \sum_{i=1}^n u_{ij}^m \left[(x_{ik} - v_{jk})^2 - \eta (v_{jk} - v_{0k})^2 \right]$, 则参数 γ 由后者代替; 反之 γ 不变, 这样保证 w_{jk} 总是意义的, 并且 w_{jk} 能够被有效地用来衡量各维度对每个集群信息的贡献。

证明: 将 $\mathbf{U}$ 和 $\mathbf{V}$ 和固定, 通过引入拉格朗日乘子, 将式(14)的有约束优化问题转化为无约束问题:

$$\begin{aligned} L(\mathbf{W}, \lambda_2) = & \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \sum_{k=1}^d w_{jk}^\tau (x_{ik} - v_{jk})^2 + \gamma \sum_{j=1}^c \sum_{k=1}^d w_{jk} (1 - w_{jk}^{\tau-1}) \\ & - \eta \sum_{j=1}^c \left(\sum_{i=1}^n u_{ij}^m \right) \sum_{k=1}^d w_{jk}^\tau (v_{jk} - v_{0k})^2 - \sum_{j=1}^c \lambda_{2j} \left(\sum_{k=1}^d w_{jk} - 1 \right) \end{aligned} \quad (23)$$

对 w_{jk} 求偏导, 并且令 $\frac{\partial L}{\partial w_{jk}} = 0$, 可以得到如下的式子:

$$\frac{\partial L}{\partial w_{jk}} = \tau w_{jk}^m \left[\sum_{i=1}^n u_{ij}^m \left((x_{ik} - v_{jk})^2 - \eta (v_{jk} - v_{0k})^2 \right) - \gamma \right] + \gamma - \lambda_{2j} = 0 \quad (24)$$

又

$$\sum_{k=1}^d w_{jk} = 1 \quad (25)$$

联立式(24)和式(25)即得 $w_{jk} = \frac{(\delta_{jk} - \gamma)^{\frac{1}{1-\tau}}}{\sum_{k'=1}^D (\delta_{jk'} - \gamma)^{\frac{1}{1-\tau}}}$ 证毕。

DCSSC 算法伪代码如表 1 所示:

Table 1. The defuzzification constraint soft subspace clustering (DCSSC) algorithm

表 1. 解模糊约束软子空间聚类算法

Algorithm DCSSC
Input: 数据集, 集群个数 C, 参数 $\gamma > 0$ 和 $\eta > 0$, 隶属度的模糊指数 $m > 1$, 权重的模糊指数 $\tau > 1$, 阈值 $\varepsilon > 0$, 迭代次数的最大值 M。随机初始化聚类中心 $\mathbf{V}^{(0)}$ 和初始化权重矩阵 $\mathbf{W}^{(0)}$ 为 $w_k = \frac{1}{d}$ 和设置 $t = 0$。 Repeat: 根据(14)式计算模糊划分矩阵 $\mathbf{U}^{(t)}$; 根据(15)式计算聚类中心矩阵 $\mathbf{V}^{(t)}$; 根据(16)式计算权重矩阵 $\mathbf{W}^{(t)}$; $t = t + 1$。 Until: $\ J(t+1) - J(t)\ < \varepsilon$ or $t = M$

4. 实验结果与分析

为了评估所提出的子空间聚类算法的性能, 我们在人工数据集以及真实数据上进行大量实验, 将解模糊化约束软子空间聚类算法与以上回顾的 6 种算法进行对比, 以此来评估我们提出的算法的性能。首先我们描述了用以评估算法性能的指标。然后比较了各算法在人工数据集, 以及在 UCI 机器学习数据库中的 Australian 数据集, Balance_scale 数据集, Wine 数据集, Zoo 数据集, Heart 数据集, Diabetes 数据集, ThreeOf9 数据集, Led7 数据集, Teaching_Assistant_Evaluation 数据集, Sirtuin6 等数据集上的表现。本文中展示所有的实验均在配置为 3.40 GH 的 CPU 和 RAM 为 16.0 G 的计算机上使用 Matlab 编程实现。

4.1. 评价标准

本文我们引进了 NMI (Normalized Mutual Information) [22] 以及 Rand index (RI) [23] 这两个评价标准, 用来衡量我们新提出的解模糊化约束软子空间聚类算法在人工数据集以及真实数据集上的性能。NMI 和 RI 指标的定义如下所示:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}}$$

其中 X 和 Y 分别表示真实标签和聚类结果; $I(X, Y) = \sum_{i=1}^c \sum_{j=1}^c \left(N_{ij} \log \left(\frac{N \cdot N_{ij}}{N_i \cdot N_j} \right) \right)$ 表示 X 和 Y 的互信息;

$H(X) = \sum_{i=1}^c N_i \log \left(\frac{N_i}{N} \right)$ 和 $H(Y) = \sum_{j=1}^c N_j \log \left(\frac{N_j}{N} \right)$, 分别表示真实标签和聚类结果的熵; N_{ij} 为聚类结果与真实的分类结果一致的数据对的个数; N_i 是第 i 个集群的数据点的个数; N_j 是经过聚类后划分到第 j 个集群的数据点的个数; N 为整个数据集中数据点的个数。

$$RI = \frac{a+b}{a+b+c+d}$$

其中, a 表示在真实标签和聚类结果中同属一类的样本对数; b 表示在真实标签和聚类结果中均不同类的样本对数; c 在真实标签中同类但聚类结果中不同类的样本对数表示; d 表示在真实标签中不同类但聚类结果中同类的样本对数; 无论是 NMI 指标还是 RI 指标, 它们的取值范围在 $[0, 1]$ 这个区间, 并且如果某个聚类算法的在这两个指标上的值越接近 1, 说明该算法的性能越好; 利用这两个指标的这一特性, 我们将通过对比改进的模糊加权子空间聚类以及上述回顾的六种子空间聚类算法在不同数据集上的这两个指标的值, 来评判各算法的性能。

4.2. 参数设置

与大多数模糊聚类算法一样, 参数的设置合理与否在很大程度上影响着 DCSSC 算法的聚类性能。在这一部分, 我们讨论了参数的设置。传统的模糊聚类算法关于模糊指数的设置都有介绍[24] [25]。通过网络搜索策略寻找各算法相对最优的参数组合, 确保在对比时的公平性。在我们提出的 DCSSC 算法中, 模糊指数的设置, 采用一种简单并且有效的理论方法进行设置[25], 若 $\min(n-1, d) \geq 3$ 时, 则

$m \leq \frac{\min(n-1, d)}{\min(n-1, d) - 2}$ 。为了方便计算 m 可设置为 $\frac{\min(n-1, d)}{\min(n-1, d) - 2}$ 。由于其余的参数设置缺乏有效的理论

方法, 因此, DCSSC 算法的超参数是经过大量实验对比后确定的较优参数组合。DCSSC 算法以及其余对比算法参数的设置如表 2 所示:

Table 2. Parameter settings

表 2. 参数设置

算法	参数设置
DCSSC	$m = \frac{\min(n-1, d)}{\min(n-1, d) - 2}$
	$\tau = \{1.5, 2, 2.5\}$
	$\gamma = \{1, 10, 20, 30, 40, 50\}$
	$\eta = \{0.001, 0.002, 0.1, 0.2, 0.5, 0.9\}$
FSC	$\varepsilon = \{0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	$\tau = \{1, 2, 5, 10, 25, 50\}$
EWKM	$\gamma = \{1, 5, 10, 15, 50, 100, 1000\}$
LAC1	$\gamma = \{5, 10, 25, 60, 70\}$
LAC2	$\gamma = \{5, 10, 25, 60, 65\}$
EWFCM	$\gamma = \{1, 15, 35, 100, 1000, 10000\}$

4.3. 各算法在不同数据集下的实验对比

为系统评的评估新提出的方法解决现有挑战的有效性, 本研究采用 K-means++初始化方法合成数据集上进行验证[26]。该数据集通过可控参数设置, 能够有效模拟真实场景中的关键挑战, 为方法性能评估提供可靠基准。我们通过计算机模拟生成 12 维, 包含三个类别, 每个类别均有 100 个样本的数据集, 并且该数据集服从高斯分布, 不同类别的均值 μ 和标准差 σ 各不相同。其中, 第一个类别服从均值为 -50, 标准差为 50 的高斯分布; 第二个类别服从均值为 150, 标准差为 5 的高斯分布; 第三个类别则是服从均

值为 100，标准差为 150 的高斯分布。如图 1 所示：

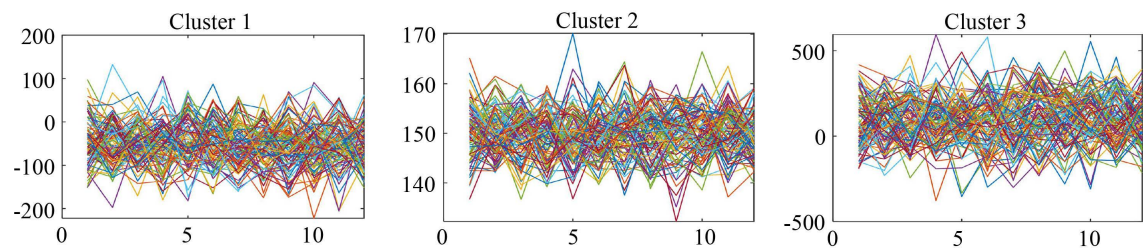


Figure 1. The distribution of three clusters in different subspaces
图 1. 三簇数据在不同子空间的分布

图 1 是我们生成的人工数据集的分布情况。横轴表示的是数据的特征维度，纵轴表示每个特征的值，图中的每一条折线表示的是一个样本。三个子图分别表示三个服从不同均值和标准差的高斯分布的集群。如第一个子图表示的是服从均值为-50，标准差为 50 的高斯分布的集群；第二个子图表示的是服从均值为 150，标准差为 5 的高斯分布的集群；第三个子图则表示的是服从均值为 100，标准差为 150 的高斯分布的集群。将本文提出的 DCSSC 算法与 FSC 算法、EWKM 算法、LAC1 算法、LAC2 算法，以及 EWFCM 算法在人工数据集上进行聚类实验，以 NMI 与 RI 的均值和标准差作为评价标准，对我们提出的 DCSSC 算法与另外 5 种算法的性能进行比较。表 3 是各算法在聚类过程中经过十次运行得到 NMI、RI 均值最佳的结果以及对应的标准差：

Table 3. The mean values and variances of NMI and RI for each algorithm on the synthetic dataset
表 3. 各算法在人工的数据集上 NMI 和 RI 的平均值及方差

		FSC	EWKM	LAC1	LAC2	EWFCM	DCSSC
人工数据集	NMI (Mean)	0.6963	0.6882	0.6867	0.6851	0.7050	0.7424
	NMI (Std)	0.0121	0.0223	0.0215	0.0222	0.0068	0.0000
	RI (Mean)	0.8313	0.8257	0.8243	0.8223	0.7991	0.8709
	RI (Std)	0.0106	0.0280	0.0123	0.0143	0.0052	0.0000
	ACC	0.8133	0.8500	0.8467	0.8233	0.7867	0.8867

表 3 统计的实验结果显示，在人工数据集上，我们从 NMI、RI 和精度三个角度评估了 DCSSC 的性能。实验结果显示，DCSSC 的 NMI 均值为 0.7424，标准差为 0.0000 而五种对比算法中表现最优者的 NMI 均值仅为 0.7050，最低标准差为 0.0121；RI 指标上，DCSSC 的均值(0.8709)和标准差(0.0000)同样优于对比算法中的最优值(均值 0.8313，标准差 0.0106)。此外，在聚类精度方面，DCSSC 达到 0.8867，高于 FSC、EWKM、LAC1、LAC2 和 EWFCM 等算法的结果。综合上述结果，DCSSC 在人工数据集上的聚类性能与稳定性均优于其他五种对比算法。除了在人工数据集上进行实验来比较每各算法的性能外，我们在实验中选择了十组 UCI 机器学习数据库里面的数据这六种算法进行比较，这十组数据的信息如表 4 所示。

从表 5 的实验结果来看，DCSSC 算法在十个 UCI 数据集上的整体聚类效果优于其他五种对比算法。在 Wine 数据集上，DCSSC 算法的 NMI 均值为 0.8615，高于 EWKM 算法(0.8357)，标准差为 0.0000，低于 LAC1(0.0178)，证明其在该数据集上准确性与稳定性均占优。在 Zoo 数据集上，DCSSC 的 NMI 均值(0.8369)亦优于 LAC1(0.8158)。前九组数据集中，DCSSC 的 NMI 均值均领先，且标准差大多为最小。在 Sirtuin6 数据集上，DCSSC 的 NMI 均值略低于 EWKM，但其标准差(0.0000)为所有算法中最小，稳定性具有优势。

Table 4. UCI dataset
表 4. UCI 数据集

Dataset	<i>N</i>	<i>D</i>	<i>C</i>
Australian	690	14	2
Balance_scale	625	4	3
Wine	178	13	3
Zoo	101	16	7
Heart	270	13	2
Diabetes	768	8	2
ThreeOf9	512	9	2
Led7	3200	7	10
Teaching_Assistant_Evaluation (TAE)	151	5	3
Sirtuin6	100	6	2

Table 5. Mean and standard deviation of NMI values
表 5. NMI 的平均值和标准差

数据集		FSC	EWKM	LAC1	LAC2	EWFCM	DCSSC
Australian	Mean	0.2795	0.2755	0.3007	0.2867	0.3273	0.4238
	Std	0.1388	0.1736	0.1826	0.1720	0.0029	0.0049
Balance_scale	Mean	0.1234	0.1243	0.1289	0.1279	0.1487	0.1718
	Std	0.0245	0.0223	0.0307	0.0253	0.0741	0.1001
Wine	Mean	0.7899	0.8357	0.8330	0.8366	0.8336	0.8615
	Std	0.0254	0.0179	0.0178	0.0222	0.0000	0
Zoo	Mean	0.6301	0.8025	0.8158	0.8053	0.8082	0.8369
	Std	0.0874	0.0552	0.0497	0.0502	0.0251	0.0561
Heart	Mean	0.2204	0.2406	0.2592	0.2466	0.2647	0.273
	Std	0.1658	0.0885	0.0557	0.0758	0.0000	0.0006
Diabetes	Mean	0.0491	0.0587	0.0518	0.0532	0.0649	0.0674
	Std	0.0308	0.0109	0.0004	0.0067	0.0012	0.0004
ThreeOf9	Mean	0.0468	0.0522	0.0460	0.0496	0.0470	0.0714
	Std	0.0285	0.0444	0.0297	0.0449	0.0459	0.0271
Led7	Mean	0.3096	0.4553	0.4904	0.4908	0.5605	0.5622
	Std	0.0522	0.0368	0.0273	0.0277	0.0222	0.0125
TAE	Mean	0.0899	0.0889	0.0905	0.0972	0.0670	0.1215
	Std	0.0159	0.0211	0.0190	0.0218	0.0141	0.0024
Sirtuin6	Mean	0.2349	0.2956	0.2781	0.2789	0.2436	0.2918
	Std	0.0526	0.0188	0.0142	0.0134	0.0289	0

Table 6. Mean and standard deviation of RI values
表 6. RI 的平均值和标准差

数据集		FSC	EWKM	LAC1	LAC2	EWFCM	DCSSC
Australian	Mean	0.6806	0.6657	0.6619	0.6727	0.7027	0.7348
	Std	0.0724	0.0999	0.1096	0.0990	0.0012	0.0563
Balance_scale	Mean	0.5898	0.5930	0.5932	0.5959	0.6038	0.6142
	Std	0.0140	0.0103	0.0203	0.0127	0.0370	0.0492
Wine	Mean	0.9089	0.9333	0.9332	0.9329	0.9331	0.9474
	Std	0.0146	0.0085	0.0090	0.0098	0.0000	0.0000
Zoo	Mean	0.7653	0.9046	0.9155	0.9063	0.8928	0.9169
	Std	0.0930	0.0430	0.0382	0.0343	0.0266	0.0446
Heart	Mean	0.6406	0.6529	0.6658	0.6554	0.6700	0.6744
	Std	0.0873	0.0596	0.0360	0.0474	0.0000	0.0000
Diabetes	Mean	0.5351	0.5553	0.5558	0.5562	0.5550	0.5575
	Std	0.0209	0.0007	0.0000	0.0014	0.0000	0.0003
ThreeOf9	Mean	0.5284	0.5343	0.5265	0.5325	0.5292	0.5406
	Std	0.0168	0.0294	0.0263	0.0298	0.0310	0.0425
Led7	Mean	0.7020	0.8579	0.8723	0.8753	0.9029	0.9035
	Std	0.0900	0.0189	0.0131	0.0148	0.0109	0.0075
TAE	Mean	0.5112	0.5609	0.5592	0.5598	0.5652	0.6067
	Std	0.0455	0.0412	0.0392	0.0379	0.0177	0.0014
Sirtuin6	Mean	0.6389	0.6862	0.6673	0.6679	0.6460	0.6768
	Std	0.0340	0.0121	0.0098	0.0092	0.0226	0.0000

根据表 6 统计的实验结果，我们提出得 DCSSC 算法在 RI 这个指标上的表现同样优于其他的五种算法。如：在 Australian 数据集上和 Wine 数据集上 DCSSC 算法得到的 RI 的均值分别为 0.7348 和 0.9474，在这两组数据上 DCSSC 算法得到的 RI 的均值比另外的五种算法得到的 RI 的均值更接近于 1，所以 Australian 和 Wine 这两组数据集上我们提出的 DCSSC 算法的性能要优于另外的五种算法。在表 5 中除了这两组数据集，DCSSC 算法在前九组数据集上得到的 RI 的均值都比另外的物种算法得到的 RI 的均值更接近于 1。在 Sirtuin6 数据集上 DCSSC 算法得到的 RI 的均值同样比 EWKM 算法得到的 RI 的均值略低，但是 DCSSC 算法对应 RI 的方差比 EWKM 算法对应 RI 的方差要低，说明 DCSSC 算法更稳定，结果更有说服力。

4.4. 超参数敏感性分析

聚类算法的超参数对算法的性能有着一定的影响。为了了解所提出的 DCSSC 算法性能对参数变化的敏感程度，评估参数在一定范围内波动时算法输出的稳定性，在这一小节中我们随机选择四组 UCI 数据集进行实验，以 RI 均值作为基准。在固定 τ 的值的的前提下，变化 γ 和 η 的值观察算法的输出情况，实验结果如下所示。

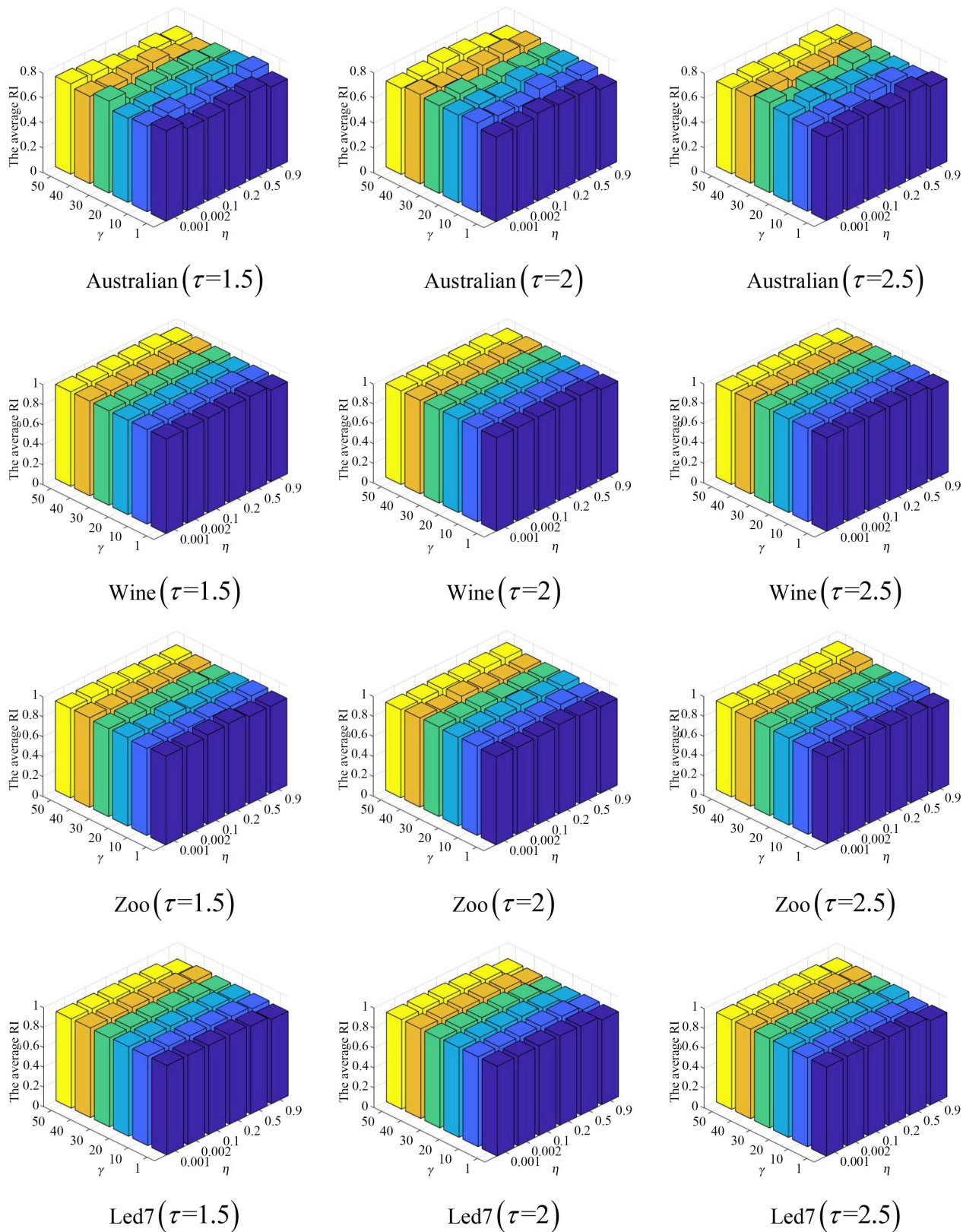


Figure 2. The hyperparameter sensitivity of the DCSSC algorithm is validated on four datasets

图 2. 在四个数据集上验证 DCSSC 算法的超参数敏感性

根据图 2 中的实验结果, 在 τ 的值固定的情况下, DCSSC 算法的输出情况随着 γ 的变化没有明显的变动; 同时随着 η 增加有一个下降的趋势, 但是这一趋势并不明显。从整体上来看 γ 和在一定范围内变动变化对 DCSSC 算法的性能没有显著的影响。说明 DCSS 算法对于 γ 和 η 的变化不敏感, 这也证明了算法具有较好的稳定性, 同时降低了调参的成本, 使得算法能快速适配各种复杂的数据场景中。

5. 统计分析

在这一小节中我们引进了 Friedman 检验, 用于检验六种算法之间的差异是否 η 具有统计意义。Friedman 检验的原假设为各算法之间的聚类性能相同, 如果拒绝原假设, 说明这六种算法之间的聚类性能存在显著性差异[27]。这里我们将显著性水平设置为 0.05。检验结果如表 7, 表 8 所示:

Table 7. Friedman test based on the NMI metric
表 7. 基于 NMI 指标的 Friedman 检验

算法	平均秩	p 值	原假设
FSC	5.6000	1.23018×10 ⁻⁵	拒绝
EWKM	4.0000		
LAC1	3.8000		
LAC2	3.3000		
EWFCM	3.2000		
DCSSC	1.1000		

Table 8. Friedman test based on the RI metric
表 8. 基于 RI 指标的 Friedman 检验

算法	平均秩	P 值	原假设
FSC	5.6000	1.51355×10 ⁻⁵	拒绝
EWKM	3.6000		
LAC1	4.0000		
LAC2	3.4000		
EWFCM	3.3000		
DCSSC	1.1000		

表 7, 表 8 分别展示了基于 NMI 指标, RI 指标, DBI 指标的 Friedman 检验结果, 其中平均秩这一项可以看作是各算法在同一数据上表现的排名, 平均秩的值越低, 说明算法的性能越好。从平均秩上来看 DCSSC 算法在 NMI 指标上的平均秩为 1.1000, 要低于另外五种算法的平均秩这说明 DCSSC 算法在 NMI 这个评价指标上要优于另外的五种算法; 在 RI 这个指标上 DCSSC 算法的平均秩为 1.1000 同样低于另外五种算法, 也就是说在 RI 这个会标上 DCSSC 算法的性能同样优于其余的五种算法。综合三个表的平均秩来看, DCSSC 算法的性能是这六种算法中相对较好。

6. 结论

与现有的软子空间聚类算法不同, 本文不仅考虑了聚类内的信息以及内间的信息, 还构造了新的权重约束函数提出了一种新目标函数, 开发了 DCSSC 算法并研究了该算法的性质。该算法试图通过最小化集群内样本的距离, 以及最大集群之间的距离, 同时通过我们构造的权重约束函数控制权重的分配。对

那些在识别集群的过程中贡献较大的特征维度赋予更大的权重, 而对于那些在识别集群过程中贡献小的特征维度赋予较小的权重。这种聚类过程就避免了通过稀疏特征维度识别聚类的问题。

我们在合成的人工数据集上使用 ACC, NMI 指标和 RI 指标的均值和标准差, 来对比 DCSSC 算法, FSC 算法, EWKM 算法, LAC1 算法, LAC2 算法以及 EWFCM 算法这六种算法的聚类性能, 对比实验的结果来看, DCSSC 算法无论是从聚类精度, 还是在 NMI, RI 上的表现效果都要好于另外的五种算法, 同时我们还在 UCI 数据库中的真实数据集上进行对比实验, 从实验的结果表明显然是 DCSSC 算法的性能更优越, 并且算法的稳定性更好。最后引入了 Friedman 检验这一非参数检验, 检验的结果显示, 我们提出的 DCSSC 算法与另外的五种算法之间存在显著差异, 并认为它们之间的差异具有统计意义。

参考文献

- [1] 王士同. 神经模糊系统及其应用[M]. 北京: 北京航空航天大学出版社, 1998.
- [2] Chen, Y., Wang, Z. and Bai, X. (2023) Fuzzy Sparse Subspace Clustering for Infrared Image Segmentation. *IEEE Transactions on Image Processing*, **32**, 2132-2146. <https://doi.org/10.1109/tip.2023.3263102>
- [3] Duan, L., Xue, W., Huang, J. and Zheng, X. (2024) Joint Sample Position Based Noise Filtering and Mean Shift Clustering for Imbalanced Classification Learning. *Tsinghua Science and Technology*, **29**, 216-231. <https://doi.org/10.26599/tst.2023.9010006>
- [4] Qi, J., Jia, Y., Liu, H. and Hou, J. (2025) Structural-Spectral Graph Convolution with Evidential Edge Learning for Hyperspectral Image Clustering. *IEEE Transactions on Image Processing*, **34**, 7919-7929. <https://doi.org/10.1109/tip.2025.3636098>
- [5] Bian, Z., Yu, L., Qu, J., Deng, Z. and Wang, S. (2025) An Ensemble Clustering Method via Learning the CA Matrix with Fuzzy Neighbors. *Information Fusion*, **120**, Article 103105. <https://doi.org/10.1016/j.inffus.2025.103105>
- [6] Zhang, W., Deng, Z., Ding, W., Zhou, J., Zhang, T., Choi, K., et al. (2026) Enhanced One-Step Incomplete Multiview Fuzzy Clustering with Dual Representation Learning. *IEEE Transactions on Fuzzy Systems*, **34**, 966-980. <https://doi.org/10.1109/tfuzz.2026.3651747>
- [7] Wu, K., Yu, J. and Yang, M. (2005) A Novel Fuzzy Clustering Algorithm Based on a Fuzzy Scatter Matrix with Optimality Tests. *Pattern Recognition Letters*, **26**, 639-652. <https://doi.org/10.1016/j.patrec.2004.09.016>
- [8] Zhu, L., Chung, F.-L. and Wang, S. (2009) Generalized Fuzzy C-Means Clustering Algorithm with Improved Fuzzy Partitions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **39**, 578-591. <https://doi.org/10.1109/tsmcb.2008.2004818>
- [9] Cao, Y. and Wu, J. (2002) Projective ART for Clustering Data Sets in High Dimensional Spaces. *Neural Networks*, **15**, 105-120. [https://doi.org/10.1016/s0893-6080\(01\)00108-3](https://doi.org/10.1016/s0893-6080(01)00108-3)
- [10] Deng, Z., Choi, K., Chung, F. and Wang, S. (2010) Enhanced Soft Subspace Clustering Integrating Within-Cluster and Between-Cluster Information. *Pattern Recognition*, **43**, 767-781. <https://doi.org/10.1016/j.patcog.2009.09.010>
- [11] Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998) Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, Seattle, 1-4 June 1998, 94-105. <https://doi.org/10.1145/276304.276314>
- [12] Cheng, C., Fu, A.W. and Zhang, Y. (1999) Entropy-Based Subspace Clustering for Mining Numerical Data. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, 15-18 August 1999, 84-93. <https://doi.org/10.1145/312129.312199>
- [13] Goil, S., Nagesh, H. and Choudhary, A. (1999) Mafia: Efficient and Scalable Subspace Clustering for Very Large Data Sets. Technical Report CPDC-TR-9906-010, Northwest University.
- [14] Aggarwal, C.C. and Yu, P.S. (2000) Finding Generalized Projected Clusters in High Dimensional Spaces. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, 15-18 May 2000, 70-81. <https://doi.org/10.1145/342009.335383>
- [15] Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C. and Park, J.S. (1999) Fast Algorithms for Projected Clustering. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, 31 May-3 June 1999, 61-72. <https://doi.org/10.1145/304182.304188>
- [16] Woo, K., Lee, J., Kim, M. and Lee, Y. (2004) FINDIT: A Fast and Intelligent Subspace Clustering Algorithm Using Dimension Voting. *Information and Software Technology*, **46**, 255-271. <https://doi.org/10.1016/j.infsof.2003.07.003>
- [17] Gan, G. and Wu, J. (2008) A Convergence Theorem for the Fuzzy Subspace Clustering (FSC) Algorithm. *Pattern*

-
- Recognition*, **41**, 1939-1947. <https://doi.org/10.1016/j.patcog.2007.11.011>
- [18] Jing, L., Ng, M.K., Xu, J. and Huang, J.Z. (2005) Subspace Clustering of Text Documents with Feature Weighting K-Means Algorithm. In: Ho, T.B., Cheung, D. and Liu, H. Eds., *Lecture Notes in Computer Science*, Springer, 802-812. https://doi.org/10.1007/11430919_94
 - [19] Zhou, J., Chen, L., Chen, C.L.P., Zhang, Y. and Li, H. (2016) Fuzzy Clustering with the Entropy of Attribute Weights. *Neurocomputing*, **198**, 125-134. <https://doi.org/10.1016/j.neucom.2015.09.127>
 - [20] Domeniconi, C., Papadopoulos, D., Gunopulos, D. and Ma, S. (2004) Subspace Clustering of High Dimensional Data. *Proceedings of the 2004 SIAM International Conference on Data Mining*, 517-521. <https://doi.org/10.1137/1.9781611972740.58>
 - [21] Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M. and Papadopoulos, D. (2007) Locally Adaptive Metrics for Clustering High Dimensional Data. *Data Mining and Knowledge Discovery*, **14**, 63-97. <https://doi.org/10.1007/s10618-006-0060-8>
 - [22] Strehl, A. and J. Ghosh. (2002) Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, **3**, 583-617.
 - [23] Wang, J., Wang, S., Chung, F. and Deng, Z. (2013) Fuzzy Partition Based Soft Subspace Clustering and Its Applications in High Dimensional Data. *Information Sciences*, **246**, 133-154. <https://doi.org/10.1016/j.ins.2013.05.029>
 - [24] Yu, J. and Yang, M.-S. (2005) Optimality Test for Generalized FCM and Its Application to Parameter Selection. *IEEE Transactions on Fuzzy Systems*, **13**, 164-176. <https://doi.org/10.1109/tfuzz.2004.836065>
 - [25] Yu, J., Cheng, Q. and Huang, H. (2004) Analysis of the Weighting Exponent in the FCM. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, **34**, 634-639. <https://doi.org/10.1109/tsmcb.2003.810951>
 - [26] Li, Y., Dong, M. and Hua, J. (2008) Localized Feature Selection for Clustering. *Pattern Recognition Letters*, **29**, 10-18. <https://doi.org/10.1016/j.patrec.2007.08.012>
 - [27] Xiao, J., Lu, J. and Li, X. (2017) Davies Bouldin Index Based Hierarchical Initialization K-Means. *Intelligent Data Analysis*, **21**, 1327-1338. <https://doi.org/10.3233/ida-163129>