

融合线性插值和对抗性学习的深度子空间聚类

江雨燕^{1*}, 陶承凤¹, 李平²

(1. 安徽工业大学 管理科学与工程学院, 安徽 马鞍山 243032;

2. 南京邮电大学 计算机学院, 江苏 南京 210023)

摘要: 聚类算法的研究受到广泛关注, 但现有聚类算法通常无法对样本进行精准的聚类且不能有效分离个体聚簇, 从而导致聚类模型泛化性能弱等问题。针对于此, 提出了一种融合线性插值和对抗性学习的深度子空间聚类模型。该模型在编码器部分使用混合函数和来自均匀分布的 α 系数对两个样本的隐表示进行线性插值, 得到新的混合输出; 进而利用子空间聚类模型对混合输出和原样本隐变量进行聚类, 有效提升模型对难分类样本的聚类精度。同时, 将对抗性学习引入自编码器并在重构数据集上训练一个鉴别器, 用于预测 α 系数, 从而提升混合输出的数据质量。在3个公开数据集上进行实验, 采用准确率(ACC)和归一化互信息(NMI)对所提算法进行评估。在Extended Yale B、ORL、COIL20数据集上准确率分别达到了0.976 1、0.874 3、0.942 3, 与现有的一些算法相比, 所提算法的ACC和NMI均有较大的提升, 验证了模型在处理难分类样本时, 性能具有明显优于现有模型。

关键词: 聚类; 深度学习; 子空间聚类; 线性插值; 对抗性学习

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2023)03-0200-07

doi: 10.3969/j.issn.1673-629X.2023.03.030

Deep Subspace Clustering by Fusing Linear Interpolation and Adversarial Learning

JIANG Yu-yan^{1*}, TAO Cheng-feng¹, LI Ping²

(1. School of Management Science and Engineering, Anhui University of Technology, Ma'anshan 243032, China;

2. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: The research of clustering algorithms has received much attention, but the existing clustering algorithms usually can not accurately cluster samples and effectively separate individual clusters, which leads to the problems such as poor generalization performance of clustering models. In response to this, we propose a deep subspace clustering model by fusing the linear interpolation and adversarial learning. Specifically, this model uses a mixing function and a uniformly distributed α coefficient in the encoder part to obtain a new mixed output by linearly interpolating the latent representations of two samples; then uses the subspace clustering model to cluster the mixed output and the original sample latent variables, which significantly improves the clustering accuracy of the difficult-to-classify samples. At the same time, adversarial learning is introduced into the auto-encoder and a discriminator is trained on the reconstructed dataset for predicting α coefficients, so as to improving the data quality of the mixed output. Experiments are carried out on three public datasets, and the proposed algorithm is evaluated by accuracy (ACC) and normalized mutual information (NMI). On the Extended Yale B, ORL and COIL20 datasets, the accuracy rates are 0.976 1, 0.874 3 and 0.942 3 respectively. The proposed algorithm has a great improvement in ACC and NMI compared with some existing algorithms. It is verified that the model has significantly better performance than existing models when dealing with difficult-to-classify samples.

Key words: clustering; deep learning; subspace clustering; linear interpolation; adversarial learning

0 引言

目前, 聚类问题已涉及到许多实际应用程序, 成为计算机视觉的重要组成部分。例如, 图像分割^[1]、运动

分割^[2]和人脸聚类^[3]等。聚类作为无监督学习领域的一项关键技术, 其通过发现数据间的内在结构, 将相似的数据点划分到同一个簇中^[4]。传统的聚类方法可

收稿日期: 2022-02-28

修回日期: 2022-06-29

基金项目: 国家自然科学基金项目(62006126)

作者简介: 江雨燕(1966-), 女, 教授, 通讯作者, 研究方向为机器学习、智能计算; 陶承凤(1999-), 女, 硕士研究生, 研究方向为机器学习、智能计算。

以分为划分式聚类算法、基于密度的聚类算法、基于图的聚类算法、基于层次的聚类算法、基于高斯混合模型的聚类算法。典型的 K-means 聚类^[5-6]通过确定簇心,并计算各个数据点到簇心的距离,来划分数据样本类别;谱聚类^[7]把所有数据看作空间中的点,样本点之间用带权重的边相连、建立图论关联,通过切图来达到聚类的目的。尽管以上传统方法各有擅长的聚类情形,但是当面对高维复杂数据时,不能有效地进行处理。

针对高维复杂数据,文献[8]提出了子空间聚类算法,该算法是基于无监督的方式将样本集合分成不同的簇,使同簇内的样本尽可能相似,不同簇的样本尽可能相异。具有代表性的算法有:稀疏子空间聚类(Sparse Subspaces Clustering, SSC)^[9-10]和低秩子空间聚类(Low-Rank Subspaces Clustering, LRSC)^[11]等。然而子空间聚类算法仅对线性数据结构具有良好的处理效果,而不能有效地处理非线性问题。

近年来,随着深度学习的进展,提出了一种基于深度神经网络来解决非线性问题的方法(深度聚类方法),该方法可以看作是通过自编码器预训练获取低维的隐表示+无监督学习微调的一个过程。现有的深度聚类算法包括:深度 k-means 聚类^[12]、深度密度聚类^[13]、深度子空间聚类^[14-15]等。其中,深度子空间聚类网络(Deep Subspace Clustering Net, DSC)将自编码器与子空间聚类相结合具有深远的影响,该算法以无监督的方式,通过编码器非线性地将输入数据映射到一个特征空间,在此空间中,把每个数据样本利用自表达特性表示为同一子空间中其他样本的线性组合,以形成近邻矩阵。

尽管现有深度聚类算法在一定程度上解决了复杂结构数据聚类问题,但是,此类方法有以下局限性:(1)通常无法有效处理难分类样本,对临界簇间的数据样本不能明确分类;(2)模型没有改变隐表示的学习位置,仅使数据点与其指定的簇中心的距离最小化,使现有簇之间的距离最大化,不能有效分离个体聚簇。针对深度聚类算法的缺点,该文在深度子空间聚类模型的基础上进行改进,提出了融合线性插值和对抗性学习的深度子空间聚类(Deep Subspace Clustering by Fusing Linear Interpolation and Adversarial learning, DSC-IA)。

本模型为学习更好的低维隐表示以处理难分类样本且有效分离个体聚簇,引入了插值训练方法和对抗性学习,该方法在编码器部分使用混合函数和来自均匀分布的 α 系数以形成混合输出。新混合输出数据通过解码器重构数据集,为了确保重构输出与易分类样本更加类似,在混合重构集上训练一个鉴别器

$D_u(x)$,用于预测混合函数的 α 系数。通过对 DSC 方法的改进,本模型中对抗自编码器的目标是通过预测 α 系数始终为 0 来欺骗鉴别器,同时,“混合重构”损失项也应增加到自编码器的损失函数中。总之,该文主要研究在无监督场景下的深度子空间聚类模型中如何加入插值训练方法和对抗性学习来提升聚类精度。

1 相关工作

1.1 子空间聚类

子空间聚类算法是基于无监督的方式将样本集合分成不同的簇,使同簇内的样本尽可能相似,不同簇的样本尽可能相异。求解子空间聚类问题的方法包括两个步骤:首先,构造近邻矩阵,通过测量每一对数据点的亲和性来完成;然后,对近邻矩阵进行归一化切割^[16]或谱聚类^[7]。但是此算法主要是依赖于线性方法,无法处理复杂结构的高维数据,因此,近年提出的深度子空间聚类模型逐渐受到关注和研究,并被广泛应用于人脸识别、自然图像的聚类与分割等场景。

1.2 深度子空间聚类

深度子空间聚类模型如图 1 所示,使用阴影圆表示数据向量、阴影方块表示卷积或反卷积后的通道,不强制设置相应的编码器层和解码器层的权重相同。

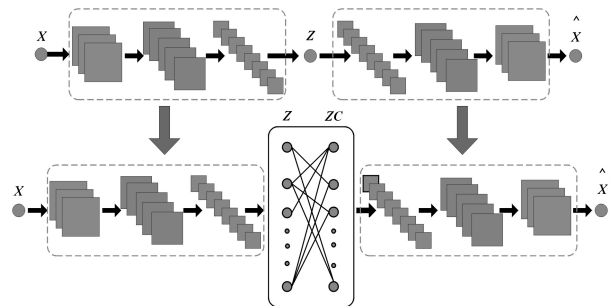


图 1 深度子空间聚类

该模型主要利用自编码器自表达层的特性,具体地,给定从多个线性子空间 $\{S_i\}_{i=1,2,\dots,k}$ 中提取的数据点 $\{Z_i\}_{i=1,2,\dots,N}(Z_i \in R^{d \times m})$,可以将子空间中的一个点表示为同一子空间中其他点的线性组合,这个属性被称为自表达性^[14]。如果将所有的点 z_i 堆叠到数据矩阵 Z 的列中,自表达属性可以简单地表示为一个等式,即 $Z = ZC$,其中 $C(C \in R^{m \times m})$ 是自表达系数矩阵。在子空间是独立的假设下,通过最小化 C 的某些范数, C 保证具有块对角结构,即 $c_{ij} = 0$ (如果点 z_i 和点 z_j 位于同一个子空间中)。因此,可以利用矩阵 C 来构造谱聚类的近邻矩阵。在数学上,这个想法被形式化为优化问题:

$$\min \|C\|_p \quad \text{s. t. } Z = ZC, (\text{diag}(C) = 0) \quad (1)$$

其中, $\|\cdot\|_p$ 表示一个任意的矩阵范数,而 C 上的可选对角约束防止了稀疏性诱导范数的平凡解,如 L_1 范

数。现在已经提出了 C 的各种规范,例如,稀疏子空间聚类(SSC)^[9]提出的 L_1 范数;低秩表示(LRR)^[17]和低秩子空间聚类(LRSC)^[11]中的核范数,以及最小二乘回归(LSR)^[18]和有效密集子空间群(EDSC)^[19]中的弗罗比乌斯范数。为了更好地解释数据损坏的原因,(1)中的等式约束通常作为一个正则化项,即:

$$\begin{aligned} \min \quad & \|C\|_p + \frac{\lambda}{2} \|Z - ZC\|_F^2 \\ \text{s. t.} \quad & (\text{diag}(C) = 0) \end{aligned} \quad (2)$$

2 DSC-IA 模型

深度子空间聚类模型与其他传统的聚类模型相比

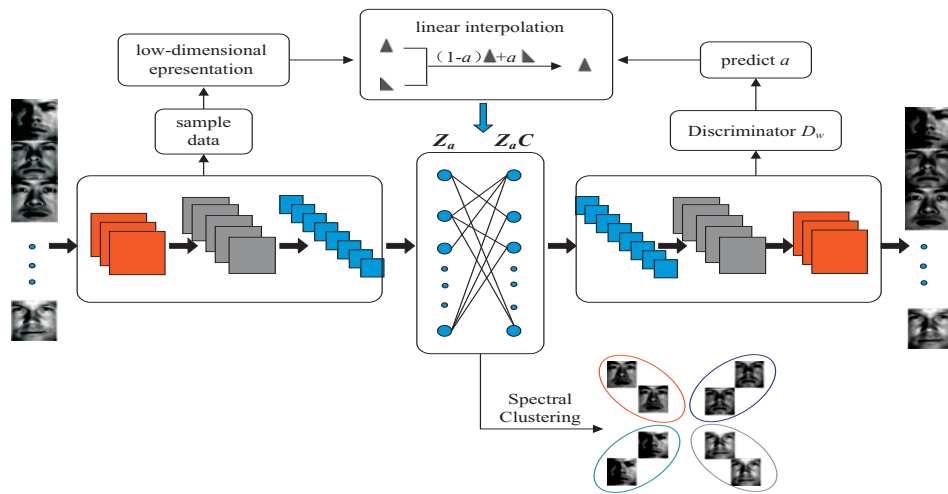


图 2 融合线性插值和对抗性学习的深度子空间聚类

2.1 DSC-IA 的目标函数

首先,设定一个对抗自编码器模型 $F(\cdot)$, 其由编码器部分 $f_e(x)$ 和解码器部分 $g_d(f_e(x))$ 组成。对于给定一对输入 $\{x^{(i)}, x^{(j)}\} \in X (X \in R^{d \times m})$, 将它们编码为低维隐表示 $z^{(i)} = f_e(x^{(i)})$ 和 $z^{(j)} = f_e(x^{(j)})$, 其中 $\{z^{(i)}, z^{(j)}\} \in Z (Z \in R^{l \times m})$ 。然后,使用混合函数和混合系数 α 将它们混合,得到 k 个混合数据:

$$z_\alpha^{(i)} = (1 - \alpha)f_e(x^{(i)}) + \alpha f_e(x^{(j)}) \quad (3)$$

其中, $\alpha \in [0, 1]$, 如果将 k 个混合样本数据和原样本数据都堆叠到数据矩阵 Z_α 的列中,即 $z_\alpha^{(i)} \in Z_\alpha, Z \in Z_\alpha (Z_\alpha \in R^{l \times (m+k)})$ 。那么,自表达层属性可以简单地表示为一个等式离子,即 $Z_\alpha = Z_\alpha C$, 其中 $C (C \in R^{(m+k) \times (m+k)})$ 是自表示系数矩阵。为了实现自表达能力,本模型最小化以下损失:

$$L_1 = \lambda_1 \|C\|_p + \frac{\lambda_2}{2} \|Z_\alpha - Z_\alpha C\|_F^2 \quad (4)$$

最后, $z_\alpha^{(i)}$ 通过解码器 $g_d(z_\alpha^{(i)})$ 运行,得到重构的 $\hat{x}_\alpha^{(i)}$, 为了训练此对抗自编码器,本模型最小化以下重构损失:

$$L_2 = \|x - g_d(f_e(x))\|^2 \quad (5)$$

其中, e 和 d 表示对抗自编码器的学习参数。

较,聚类精度已经有了很大的提高,但是在对于难分类样本、能否有效分离个体聚簇问题上还存在一定的不足。针对上述两个问题,该文提出了 DSC-IA 模型(如图 2 所示),该图展示了一个具有三层卷积编码器层、一个自表达层和三层反卷积解码器层的对抗自编码器。首先,将样本数据编码为低维隐表示;然后,对其进行线性插值,用 \blacktriangle 和 \blacktriangleright 表示两个难分类样本,使用混合函数和混合系数进行混合表示,新数据进入自表达层;最后,通过解码器得到重构数据。特别地,在解码器部分训练鉴别器用于预测混合系数,使得 Z_α 的隐表示更加适合于空间聚类。在训练期间,通过预训练对整个网络进行初始化微调。

为了确保重构的 $\hat{x}_\alpha^{(i)} = g_d(z_\alpha^{(i)})$ 与原始的和难分类的样本更类似,本模型在实重构和混合重构集上训练一个鉴别器 $D_w(x)$, 预测用于混合函数的 α 系数:

$$\hat{\alpha} = \begin{cases} D_w(z_\alpha^{(i)}) & \{z_\alpha^{(1)}, z_\alpha^{(2)}, \dots, z_\alpha^{(k)}\} \in Z_\alpha \\ D_w(z^{(i)}) & \{z^{(1)}, z^{(2)}, \dots, z^{(m)}\} \in Z \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

其中,输出 $\hat{\alpha}$ 是一个标量值。然后将鉴别器 D_w 训练以最小化:

$$L_3 = \|D_w(\hat{x}_\alpha) - \alpha\|^2 + \|D_w(\gamma x + (1 - \gamma)\hat{x})\|^2 \quad (7)$$

其中, γ 是一个标量超参数,第一项用来预测 α 系数,第二项用来提高训练稳定性。

为了在簇之间产生一个具有低维的特征空间结构,并能够在它们之间进行线性插值,混合表示 z_α 的重构 \hat{x}_α 必须看起来像来自同一数据集的真实数据点。由于在原始数据集中没有这些数据信息,本模型认为给定的两个输入 $\{x^{(i)}, x^{(j)}\} \in X (X \in R^{d \times m})$ 的构造 \hat{x} 看起来必须像其中一个输入数据点,如下:

$$\hat{x} = \begin{cases} x^{(i)} & \alpha \in [0, 0.5] \\ x^{(j)} & \alpha \in (0.5, 1] \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

因此,在 DSC-IA 中,对抗自编码器的部分目标是产生实际的混合重构,使它看起来像来自用于混合操作的数据点之一。为了在无监督的子空间聚类背景下说明 DSC-IA,该文对从数据集中输入的 m 个数据点 $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\} \in X (X \in R^{d \times m})$ 的样本批次处理,并尽量减少以下损失:

$$L = \underbrace{\lambda_1 \|\mathbf{C}\|_p}_{\text{最小化C的范数}} + \underbrace{\frac{\lambda_2}{2} \|\mathbf{Z}_\alpha - \mathbf{Z}_\alpha \mathbf{C}\|_F^2}_{\text{自表达损失}} + \underbrace{\sum_{i=1}^n \|x^{(i)} - g_d(f_e(x^{(i)}))\|^2}_{\text{重构损失}} + \underbrace{\sum_{i=1}^k \lambda_3 \|D_w(\hat{x}_\alpha^{(i)}) - \alpha\|^2 + \|D_w(\gamma x + (1-\gamma)\hat{x})\|^2}_{\text{混合重构重建法}} + \underbrace{\sum_{i=1}^k \|x^{(i)} - \hat{x}_\alpha^{(i)}\|^2}_{\text{创建混合重构的重建}}$$

s. t. $(\text{diag}(\mathbf{C}) = 0)$ (9)

其中, $\hat{x}_\alpha^{(i)}$ 表示第 i 个混合表示 $z_\alpha^{(i)}$ 的重构数据 $\hat{x}_\alpha^{(i)} = g_d(z_\alpha^{(i)})$, 且 $z_\alpha^{(i)} = (1-\alpha)f_e(x^{(i)}) + \alpha f_e(x^{(j)})$, $j \neq i$ 。参数符号说明如表 1 所示。

表 1 参数符号说明

符号	含义
$x^{(i)}, x^{(j)}$	表示一对输入数据, $\{x^{(i)}, x^{(j)}\} \in X (X \in R^{d \times m})$
$f_e(x^{(i)})$	编码器部分
$g_d(f_e(x^{(i)}))$	解码器部分
e, d	表示对抗自编码器的学习参数
$z^{(i)} = f_e(x^{(i)})$	低维隐表示, $\{z^{(i)}, z^{(j)}\} \in Z (Z \in R^{l \times m})$
$z^{(j)} = f_e(x^{(j)})$	
$z_\alpha^{(i)}$	表示混合数据, $z_\alpha^{(i)} \in Z_\alpha (Z_\alpha \in R^{l \times (m+k)})$
α	标量值, $\alpha \in [0, 1]$
\mathbf{C}	表示系数矩阵, $\mathbf{C} \in R^{(m+k) \times (m+k)}$
$\hat{x}_\alpha^{(i)}$	表示第 i 个混合表示 $z_\alpha^{(i)}$ 的重构数据
$D_w(\hat{x}_\alpha)$	鉴别器,用于预测混合函数 α 系数
$\lambda_1, \lambda_2, \lambda_3, \gamma$	表示标量超参数

2.2 DSC-IA 的算法流程

由于从零开始直接训练一个具有数百万个参数的网络很困难,该文仍然采用深度子空间聚类使用的预训练+微调策略。这在避免平凡的全零解的同时,最小化式(9)中的损失函数。

如图 1 所示,首先,对所有数据进行了深度自编码器的预训练,而没有自表达层。然后,使用训练好的参数来初始化网络的编码器层和解码器层。最后,进行谱聚类得到聚类结果。

具体优化流程如算法 1 所示:

算法 1: DSC-IA

输入: 原始数据 X , 参数 $\lambda_1, \lambda_2, \lambda_3$ 。

输出: 聚类结果。

预训练:

1. 经过编码器层计算潜在表示 $z^{(i)} = f_e(x^{(i)})$;

2. 使用 α 系数将两个样本 $z^{(i)}, z^{(j)}$ 混合, 得到新的潜在表示

$Z_\alpha = \{z_\alpha^{(1)}, z_\alpha^{(2)}, \dots, z_\alpha^{(m)}\}$;

3. 经过解码器层计算混合重构 $\hat{x}_\alpha = g_d(z_\alpha^{(i)})$ 和原重构 $\hat{x} = g_d(z^{(i)})$;

4. 通过鉴别器 $D_w(\hat{x})$ 运行两个重构集 \hat{x} 和 \hat{x}_α 。

微调阶段:

1. 根据鉴别器在式(7)中的损失函数,通过梯度下降法来更新该鉴别器;

2. 根据其在式(9)中定义的损失函数,通过梯度下降法来更新自编码器。

初始化: 使用训练后的参数来初始化网络的编码器层和解码器层。

通过自表达层 $\mathbf{Z}_\alpha - \mathbf{Z}_\alpha \mathbf{C}$ 构建近邻矩阵 \mathbf{C} ;

将 \mathbf{C} 应用谱聚类方法求出聚类结果。

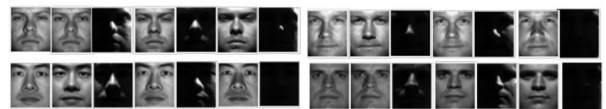
3 实验及结果分析

3.1 数据集

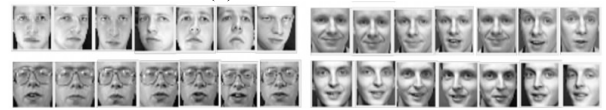
为了验证所提算法的有效性,使用几种较为常见的聚类算法在三种数据集(包括: Extended Yale B^[20]、ORL^[21]、COIL20^[22])进行实验测试。在同一数据集进行 10 次独立实验,取平均值作为最终实验结果。实验环境为 Microsoft Windows 10, 处理器为英特尔酷睿 i7, 内存容量 8 GB, 显卡配置为 NVIDIA GeForce 920M。

3.1.1 Extended Yale B 数据集

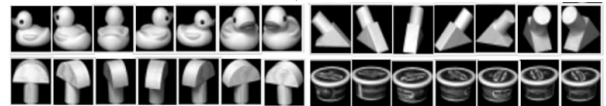
Extended Yale B 数据集是一个流行的子空间聚类基准。它由 38 名受试者组成,每名受试者在不同姿势和光照条件下均获得 64 张面部图像,并将图像的大小调整为 42×42 , 该数据集的样本图像见图 3(a), 网络参数设置见表 2。



(a) Extended Yale B



(b) ORL



(c) COIL20

图 3 样本图像

表 2 Extended Yale B 数据集的网络参数设置

layers	encoder-1	encoder-2	encoder-3	self-expressive	decoder-1	decoder-2	decoder-3
kernel size	5×5	3×3	3×3	-	3×3	3×3	5×5
channels	10	20	30	-	30	20	10
parameters	260	1 820	5 430	5 914 624	5 420	1 810	251

3.1.2 ORL 数据集

ORL 数据集有 40 名受试者,每名受试者有 10 个样本,共由 400 张人脸图像组成。对于每名受试者,这些图像是在不同的光照条件下拍摄的,并且带有不同

的面部表情(睁眼/闭眼、微笑/不微笑)、面部饰品(有/无眼镜),将这些图像的大小调整为 32×32,该数据集的样本图像见图 3(b),网络参数设置见表 3。

表 3 ORL 数据集的网络参数设置

layers	encoder-1	encoder-2	encoder-3	self-expressive	decoder-1	decoder-2	decoder-3
kernel size	5×5	3×3	3×3	-	3×3	3×3	5×5
channels	5	3	3	-	3	3	5
parameters	130	138	84	160 000	84	140	126

3.1.3 COIL20 数据集

COIL 数据集是由灰度图像样本组成,COIL20 包含 20 多个受试对象,共由 1 440 张图像组成,受试对象的类型多种多样,如鸭和车型等;此数据集在黑色背景的转盘上,每个物体每隔 5 度拍摄 72 张图像,即同一对象以不同的角度捕获不同样本。在实验中,图像的大小被调整为 32×32,该数据集的样本图像见图 3(c),网络参数设置见表 4。

表 4 COIL20 数据集的网络参数设置

layers	encoder-1	self-expressive	decoder-1
kernel size	3×3	-	3×3
channels	15	-	15
parameters	150	2 073 600	136

3.2 评价指标

该文使用了两种流行的聚类指标(即精度(ACC)和归一化互信息(NMI))来评估所有方法的聚类性能,ACC 和 NMI 的值越大,聚类性能就越好。

ACC 被定义为:

$$ACC = \frac{\sum_{i=1}^n 1(y_i = m(c_i))}{n} \quad (10)$$

其中, y_i 是标签, c_i 是模型的聚类分配, $m(\cdot)$ 是聚类分配和标签之间的映射函数, $1(\cdot)$ 是指示器函数返回 1 或 0。

NMI 定义为:

$$NMI(Y, C) = \frac{I(Y, C)}{\frac{1}{2}[H(Y) + H(C)]} \quad (11)$$

NMI 计算互信息得分 $I(Y, C)$, 然后根据熵函数 H 的数量对其进行归一化,以解释聚类的总数^[23]。

3.3 结果分析

将该方法与以下几种方法在上述数据集上进行了

比较,包括:稀疏子空间聚类(Sparse Subspace Clustering, SSC)^[9]、核稀疏子空间聚类(Kernel Sparse Subspace Clustering, KSSC)^[24]、低秩表示法(Low Rank Representation, LRR)^[17]、SSC 与预训练的卷积自编码器功能相结合的方法(AE+SSC)^[25]、深度子空间聚类(Deep Subspace Clustering, DSC)^[14-15]。

稀疏子空间聚类(SSC):通过将每个数据点表示为来自同一子空间的其他数据点的稀疏线性组合构造子空间的数据近邻矩阵。

核稀疏子空间聚类(KSSC):通过核技巧将 SSC 扩展到非线性流形。

低秩表示法(LRR):在所有可以表示数据样本基的线性组合的候选对象中寻求最低的线性表示。

AE+SSC:结合 SSC 和自编码器的优点。

深度子空间聚类(DSC- l_1):基于 l_1 正则化的构架。

深度子空间聚类(DSC- l_2):基于 l_2 正则化的构架。

上述几种算法均由作者提供的代码进行实验,为确保实验公平性,具体参数根据论文设置为最优,记录最优参数下结果进行对比,对比结果如表 5 所示。

由表 5 可知,所提算法相比一些子空间聚类算法在公开标准数据集上可以得出良好的聚类效果。在 Extended Yale B 数据集上 ACC 相比 DSC- l_1 、DSC- l_2 算法高 0.94%、0.28%;在 ORL 数据集上,ACC 相比 DSC- l_1 、DSC- l_2 算法高 1.68%、1.43%;在 COIL-20 数据集上,ACC 相比 DSC- l_1 、DSC- l_2 算法高 1.18%、0.55%。这说明在编码器部分使用混合函数和系数 α 将一对输入进行线性插值,得到的新混合输出可以在自表达层学习更好的特征表示,使得预训练过程更加缜密,提高了模型的泛化能力;并且在重构数据集上引入鉴别器,使重构的数据与难分类的样本更易分类,有

效分离个体聚簇。

表5 不同算法在三个数据集上的实验结果对比

Method	Extended Yale B		ORL		COIL20	
	ACC	NMI	ACC	NMI	ACC	NMI
SSC	0.724 9	0.779 6	0.675 0	0.845 9	0.851 4	0.889 2
KSSC	0.722 5	0.735 9	0.657 5	0.807 0	0.753 5	0.824 3
LRR	0.651 9	0.863 6	0.617 5	0.860 3	0.689 9	0.874 7
AE+SSC	0.746 7	0.783 3	0.732 5	0.855 5	0.779 2	0.899 0
DSC- λ_1	0.966 7	0.968 7	0.857 5	0.902 3	0.930 5	0.935 3
DSC- λ_2	0.973 3	0.970 3	0.860 0	0.903 4	0.936 8	0.940 8
DSC-IA	0.976 1	0.974 5	0.874 3	0.911 7	0.942 3	0.946 9

在所提算法中,平衡参数 λ_1 、 λ_2 和 λ_3 的取值对聚类结果的影响较大。为讨论3个参数对所提算法的影响,实验采用固定其二、改变其一来观察ACC和NMI的变化。实验结果见图4~图6。在Extended Yale B数据集上,ACC和NMI在 λ_1 和 λ_2 取值区间中起伏较大,在 λ_3 取值为 $\{6, 6.5, 7\}$ 时变化相对较小、基本平稳,最佳值设置为 $\lambda_1 = 1.00$ 、 $\lambda_2 = 10.00$ 、 $\lambda_3 = 6.30$;在ORL数据集上,ACC和NMI均在 λ_1 小于0.1、 λ_2 小于0.5、 λ_3 大于1的取值范围内聚类性能显著,最佳值

设置为 $\lambda_1 = 10^{-3}$ 、 $\lambda_2 = 0.1$ 、 $\lambda_3 = 1.1$;在COIL20数据集上,在 λ_1 取值为 $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ 、 λ_2 取值为 $\{0.5, 1, 1.5, 2\}$ 、 λ_3 取值为 $\{0.1, 0.5, 1\}$ 时,ACC值的变化相对较小,NMI值的变化基本平稳,最佳值设置为 $\lambda_1 = 1.00$ 、 $\lambda_2 = 1.5$ 、 $\lambda_3 = 0.4$ 。从图4~图6可知,噪声对不同的数据集影响程度不同,因此,平衡参数的最佳值也不相同。但在合适的取值区间上,提出的融合线性插值和对抗性学习的深度子空间聚类算法体现出了较好的稳定性。

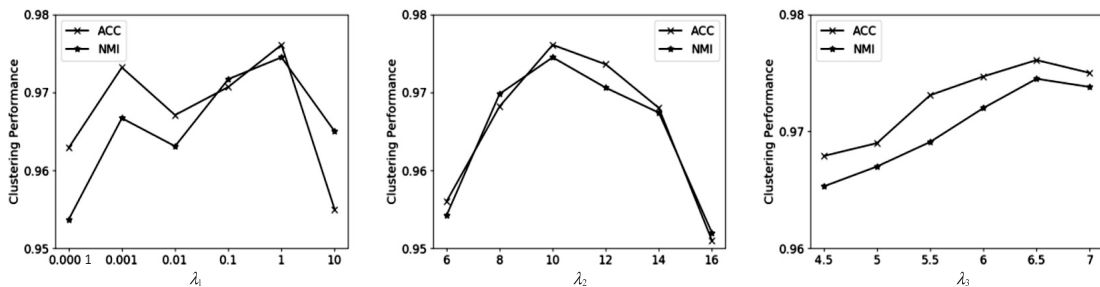


图4 Extended Yale B数据集中不同 λ_1 、 λ_2 、 λ_3 上的ACC和NMI变化曲线

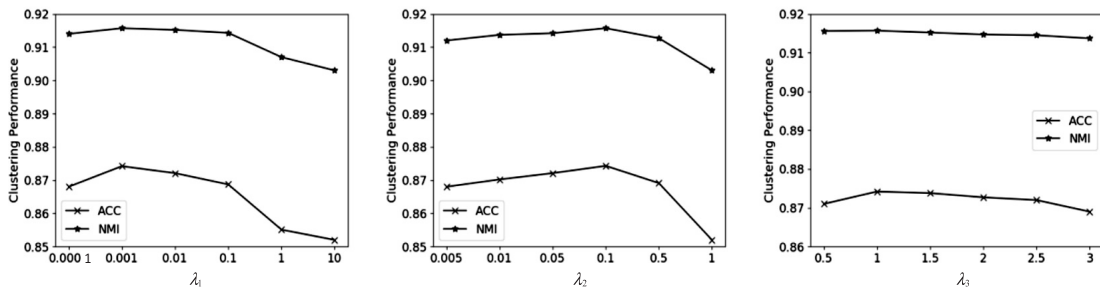


图5 ORL数据集中不同 λ_1 、 λ_2 、 λ_3 上的ACC和NMI变化曲线

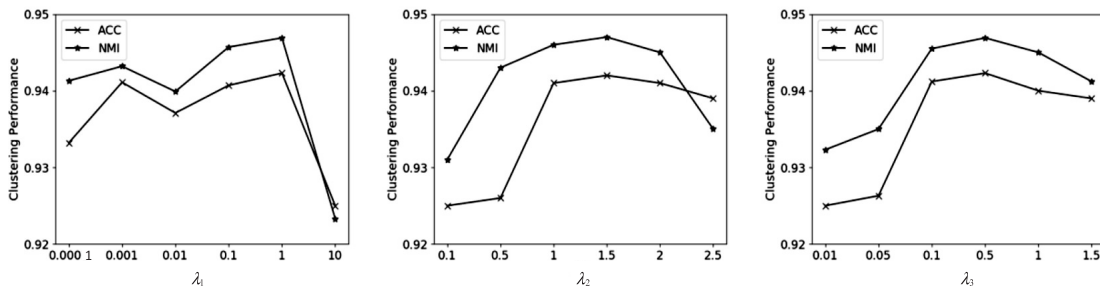


图6 COIL20数据集中不同 λ_1 、 λ_2 、 λ_3 上的ACC和NMI变化曲线

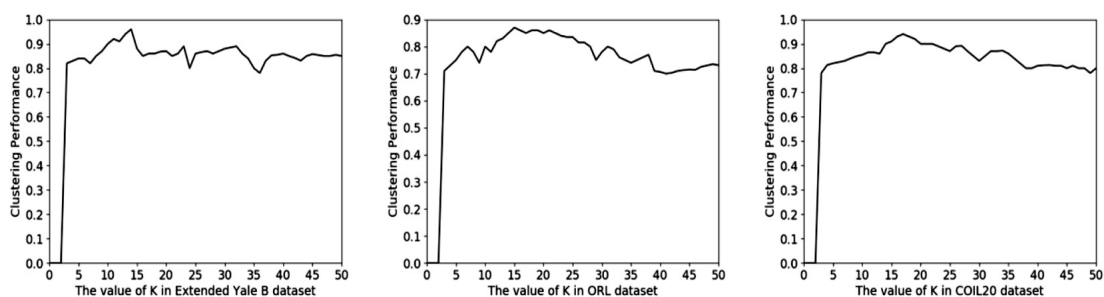


图 7 在 Extended Yale B、ORL、COIL20 数据集上不同 k 值聚类结果

此外,邻接矩阵的学习方法与深度子空间聚类相同,由于近邻数不同,很大程度上影响了聚类准确度,因此,为了考虑 k 值变化对算法的影响,通过将 k 区间设置为(2,50)来显示不同聚类结果。通过图 7 可以清晰地看出 k 在不同数据集上最优值。在 Extended Yale B 数据集上,当 $k = 14$ 时,聚类准确率达到最优;在 ORL 数据集上,当 $k = 15$ 时,聚类准确率达到最优;在 COIL20 数据集上,当 $k = 17$ 时,聚类准确率达到最优。由图 7 可知,在三个数据集上 k 值均在(10,30)区间比其他区间的聚类效果好,这说明所提算法在簇数较多时聚类精度较高,同时验证了算法对分离个体簇具有有效性。

4 结束语

该文提出了一种融合线性插值和对抗性学习的深度子空间聚类方法(DSC-IA),该方法有以下创新点:其一,在编码器部分使用混合函数和 α 系数线性插值两个隐表示;其二,通过对抗性学习,在新的混合重构集上训练一个鉴别器来预测用于混合函数的 α 系数。在三个常用基准数据集上的实验结果证明了 DSC-IA 相对于其他算法的优越性,并验证了本模型设计的优势。算法在聚类精度方面,比其他的子空间聚类解决方案有显著改进。

参考文献:

- [1] YANG A Y, WRIGHT J, MA Y, et al. Unsupervised segmentation of natural images via lossy data compression[J]. *Computer Vision & Image Understanding*, 2008, 110(2): 212-225.
- [2] KUROSAWA N, KANATANI K. Motion segmentation by subspace separation and model selection[C]//*Proceedings eighth IEEE international conference on computer vision*. Vancouver; IEEE, 2001; 586-591.
- [3] HO J, YANG M H, LIM J, et al. Clustering appearances of objects under varying illumination conditions[C]//*IEEE computer society conference on computer vision & pattern recognition*. New York; IEEE, 2006.
- [4] 章永来,周耀鉴. 聚类算法综述[J]. *计算机应用*, 2019, 39(7): 1869-1882.
- [5] WONG J A H A. Algorithm AS 136: a K-Means clustering algorithm[J]. *Journal of The Royal Statistical Society*, 1979, 28(1): 100-108.
- [6] 吕红伟,王士同. 基于 RPCA 对高维数据子空间聚类的预测方法[J]. *计算机工程与科学*, 2017, 39(3): 553-561.
- [7] 白璐,赵鑫,孔钰婷,等. 谱聚类算法研究综述[J]. *计算机工程与应用*, 2021, 57(14): 15-26.
- [8] VIDAL R. Subspace clustering[J]. *IEEE Signal Processing Magazine*, 2011, 28(2): 52-68.
- [9] ELHAMIFAR E, VIDAL R. Sparse subspace clustering[C]//*IEEE conference on computer vision & pattern recognition*. Miami; IEEE, 2009: 2790-2797.
- [10] 傅德胜,周辰. 基于密度的改进 K 均值算法及实现[J]. *计算机应用*, 2011, 31(2): 432-434.
- [11] VIDAL R, FAVARO P. Low rank subspace clustering (LR-SC)[J]. *Pattern Recognition Letters*, 2014, 43(1): 47-61.
- [12] KANUNGO T, MOUNT D M, NETANYAHU N S, et al. An efficient k-means clustering algorithm: analysis and implementation[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2002, 24(7): 881-892.
- [13] REN Y, WANG N, LI M, et al. Deep density-based image clustering[J]. *Knowledge - Based Systems*, 2020, 197: 105841.
- [14] JI P, ZHANG T, LI H, et al. Deep subspace clustering networks[J]. *arXiv: 1709.02508*, 2017.
- [15] 何锦蓉. 深度子空间聚类算法研究[D]. 北京: 中国矿业大学, 2020.
- [16] SHI J, MALIK J. Normalized cuts and image segmentation[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2000, 22(8): 888-905.
- [17] LIU G, YAN S. Latent low-rank representation for subspace segmentation and feature extraction[C]//*International conference on computer vision*. Barcelona; IEEE, 2011.
- [18] LU C Y, MIN H, ZHAO Z Q, et al. Robust and efficient subspace segmentation via least squares regression[C]//*European conference on computer vision*. Hersonisos; Springer, 2012: 347-360.
- [19] PAN J, SALZMANN M, LI H. Efficient dense subspace clustering[C]//*IEEE winter conference on applications of computer vision*. Columbus; IEEE, 2014: 461-468.