基于狄利克雷变分自编码的深度嵌入聚类

李必嘉[1],吴昊旻[2]
1. 重庆师范大学 数学科学学院,重庆 401331
2. 重庆师范大学 重庆国家应用数学中心,重庆 401331

摘 要:目的 基于深度神经网络的聚类模型由于能从原始数据中学习到有效特征,在各种无监督应用中受到了广泛关注。针对现有的基于自编码的聚类模型没有生成能力,且通常以高斯分布作为先验,限制了对多模态特征的表达能力问题,提出一种深度嵌入聚类模型——DVADEC(Deep Embedded Clustering based on Dirichlet Variational Autoencoder),该模型将狄利克雷变分自编码器的表征学习能力和嵌入聚类的聚类能力结合到一个统一的模型中。**方法** 首先,在预训练阶段,利用狄利克雷分布的多模态特性,将其作为先验分布来指导隐变量的学习过程;然后,将训练好的权重加载到聚类模型中,并通过在隐藏空间中嵌入聚类层来进行类别分配;最后,通过交替优化目标函数来微调网络,以提升聚类结果。**结果** 实验结果显示:DVADEC 模型在 4 个基准数据集上展现出较好的聚类性能,其中在 MNIST 图像数据集上达到了 97.13%的准确率,在 REUTER-10k 文本数据集上达到了 80.1%的准确率。另外,可视化结果显示潜在特征具有明显的可分性,且根据特征生成的样本轮廓清晰、平滑多样。**结论** DVADEC 模型融合了生成能力和多模态特征的表达能力,并显著提高了特征提取和聚类性能,为数据挖掘和模式识别领域提供了新的思路和技术手段。

关键词:深度聚类;无监督学习;神经网络;狄利克雷分布;变分自编码
中图分类号:TP391.41;TP18 **文献标识码:**A doi:10.16055/j.issn.1672-058X.2025.0003.007

Deep Embedded Clustering Based on Dirichlet Variational Autoencoder

LI Bijia[1], WU Haomin[2]
1. School of Mathematical Science, Chongqing Normal University, Chongqing 401331, China
2. National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China

Abstract: Objective Deep neural network-based clustering models, capable of learning effective features from raw data, have received widespread attention in various unsupervised applications. Existing autoencoder-based clustering models lack generative ability and generally use Gaussian distribution as a prior, limiting the expression of multimodal features. This paper proposes a deeply embedded clustering model—DVADEC (Deep Embedded Clustering based on Dirichlet Variational Autoencoder), which integrates the representation learning capability of Dirichlet variational autoencoder and the clustering capability of embedded clustering into a unified model. **Methods** Firstly, during the pre-training phase, the multimodal nature of Dirichlet distribution is utilized as a prior distribution to guide the learning process of latent variables. Then, the trained weights are loaded into the clustering model, and class assignments are performed by embedding clustering layers in the latent space. Finally, the network is fine-tuned through alternating optimization of the

收稿日期:2023-03-05 **修回日期:**2023-05-21 **文章编号:**1672-058X(2025)03-0052-11
基金项目:重庆师范大学研究生科研创新项目(YKC23032).
作者简介:李必嘉(1993—),男,重庆綦江人,硕士研究生,从事机器学习与人工智能研究.
通信作者:吴昊旻(1999—),男,吉林省吉林市人,硕士研究生,从事机器学习与人工智能研究. Email:whm7892023@163.com.
引用格式:李必嘉,吴昊旻. 基于狄利克雷变分自编码的深度嵌入聚类[J]. 重庆工商大学学报(自然科学版),2025,42(3):52-62.
LI Bijia, WU Haomin. Deep embedded clustering based on Dirichlet variational autoencoder[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(3): 52-62.

objective function to enhance clustering results. **Results** Experimental results demonstrate that the DVADEC model exhibits good clustering performance on four benchmark datasets, achieving an accuracy of 97.13% on the MNIST image dataset and an accuracy of 80.1% on the REUTER-10k text dataset. Furthermore, visualization results demonstrate clear separability of latent features, and samples generated based on features exhibit distinct, smooth, and diverse contours. **Conclusion** The DVADEC model integrates generative capability and the ability to express multimodal features, significantly enhancing feature extraction and clustering performance. It provides new perspectives and technical means for the fields of data mining and pattern recognition.

Keywords: deep clustering; unsupervised learning; neural networks; Dirichlet distribution; variational autoencoder

1　引　言

聚类是人工智能领域中最广泛的研究主题之一,从文档分析[1-2]、区域科学[3]、图像检索[4-5]、图像分割[6-8],到网络分析[9-11]。在过去的几十年里,已经提出了许多聚类算法:例如 K-means[12],它首先随机选择 k 个聚类中心点,然后通过迭代的方式,根据每个数据点到聚类中心的距离将数据点分配到最近的簇中;层次聚类[13]是一种自底向上或自顶向下的聚类方法,将数据点逐渐合并为更大的簇或者将簇分解为更小的簇;DBSCAN(Density-Based Spatial Clustering of Applications with Noise)[14]是一种基于密度的聚类算法,能够自动发现任意形状的聚类,并对噪声数据进行有效的过滤;高斯混合模型(GMM)[15]是一种概率生成模型,通过假设数据点来自多个高斯分布的线性组合,将数据进行聚类。尽管进行了广泛的研究,但传统聚类方法在处理高维数据时的性能并不理想,主要原因可以归结于以下 3 点:首先,高维数据中存在维度灾难问题,导致样本稀疏性增加;其次,高维数据使得距离计算变得困难,常用的距离度量不再准确反映样本之间的相似性;另外,高维数据还容易引发维度的冗余和噪声,影响聚类结果的准确性和鲁棒性。为了解决这类问题,常见的方法是通过应用主成分分析(PCA)[16]或特征选择等降维技术将数据从高维特征空间转换为低维空间再进行聚类。然而,这些方案忽略了特征学习和聚类之间的相互关系。

近年来,随着深度学习在图像识别、自然语言处理和语音识别等领域取得的成功,聚类领域开始引入深度学习方法,从而产生了深度聚类。深度聚类将深度学习与聚类算法相结合,通过深度网络学习有意义的特征来进行聚类。特别地,自编码器(AE)和变分自编码器(VAE)[17]是在深度聚类中被广泛应用的神经网络,已被成功应用于很多深度聚类任务[18-19]。例如,Peng[20]和 Tian 等[21]利用 DNN 学习原始数据的低维表示或有用特征,然后传统聚类方法对低维特征进行聚类,在一定程度上提高了聚类性能。然而,这类方法无法保证学习到的低维特征适合所使用的聚类方法,聚类效果也不显著。相反,Xie 等[22]提出了深度嵌入聚类(DEC),将 DNN 的特征学习能力和聚类算法的聚类能力结合在一个统一的模型中,联合学习特征映射和聚类。但 DEC 中预训练结束后去掉了解码部分,可能导致空间扭曲,于是 Guo 等[23]提出了改进的深度嵌入聚类(IDEC),使得网络具有局部结构保持性。后来,Guo 等[24]又提出深度卷积嵌入聚类(DCEC),进一步提升了聚类性能。尽管 DEC 等相关工作取得了成功,但这类方法均是基于自编码网络作为特征提取器,学到的特征单一,限制了聚类效果的提升。

为了学到更丰富的特征,本文以变分自编码网络作为特征提取器,再结合一些合理的先验知识和假设来改进聚类性能,提出了一种基于狄利克雷变分自编码网络的聚类算法——DVADEC(Deep Embedded Clustering based on Dirichlet Variational Autoencoder),该算法利用先验分布对特征空间加以约束,有效提升了聚类性能;改进了狄利克雷变分自编码网络,使得模型具备生成能力且特征空间具有多模态表达能力;引入了软最大化拉普拉斯方法近似狄利克分布,解决了狄利克雷分布的重参数化。

2　相关工作

2.1　狄利克雷分布

狄利克雷分布(Dirichlet Distribution)又称多元贝塔分布(Multivariate Beta Distribution),是一种高维连续概率分布,它在实数域上以正单纯形作为支撑集,是贝塔分布在高维情况下的推广。通常可以将狄利克雷分布视为定义在[0,1]区间上的多个随机变量的联合概率分布,因此常被用作多项分布参数的先验。

假设随机变量 $\boldsymbol{y}=(y_1,y_2,\cdots,y_K)\in\boldsymbol{R}^K$ 服从狄利克雷分布,则概率密度函数 $P_{\mathrm{Dir}}(\boldsymbol{y}\mid\boldsymbol{\alpha})$ 具有以下表达式:

$$P_{\mathrm{Dir}}(\boldsymbol{y}\mid\boldsymbol{\alpha})=\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}y_i^{\alpha_i-1}\tag{1}$$

式(1)中,$y_i\geqslant 0$,且是独立同分布的,表示 K 维随机变量的各个分量,满足 $\sum_{i=1}^{K}y_i=1$;$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_K)$,$\alpha_i>0$,

表示狄利克雷分布的参数,其值也可以看作是各个分量的权重参数;$\Gamma(x)$ 表示伽马函数。除此之外,狄利克雷分布还具有以下性质:

(1) 均值: $E[y_i]=\dfrac{\alpha_i}{\sum_i \alpha_i}; i=1,\cdots,K$。

(2) 方差: $\mathrm{Var}[y_i]=\dfrac{\alpha_i\left(\left(\sum_i \alpha_i\right)-\alpha_i\right)}{\left(\sum_i \alpha_i\right)^2\left(\left(\sum_i \alpha_i\right)+1\right)}$; $i=1,\cdots,K$。

(3) 协方差矩阵:

$$\mathrm{Cov}(y_i,y_j)=\frac{-\alpha_i\alpha_j}{\left(\sum_i \alpha_i\right)^2\left(\left(\sum_i \alpha_i\right)+1\right)}; i,j=1,\cdots,K; i\neq j。$$

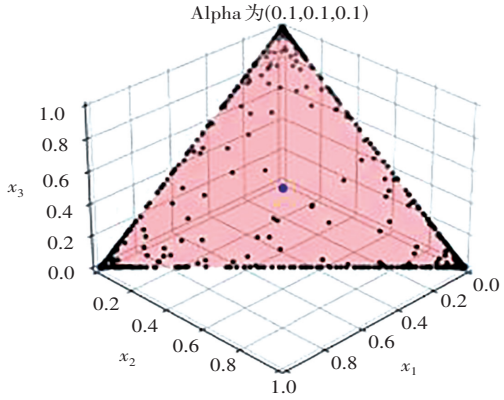
(4) 两个狄利克雷分布存在闭形式的 KL 散度(其值用 D_{KL} 表示):

$$D_{\mathrm{KL}}(q(x)\parallel p(x))=\log\Gamma\left(\sum \hat{\alpha}_i\right)-\log\Gamma\left(\sum \alpha_i\right)-\sum\log\Gamma(\hat{\alpha}_i)+\sum\log\Gamma(\alpha_i)+\sum(\hat{\alpha}_i-\alpha_i)\left(\psi(\hat{\alpha}_i)-\psi\left(\sum \hat{\alpha}_i\right)\right) \quad (2)$$

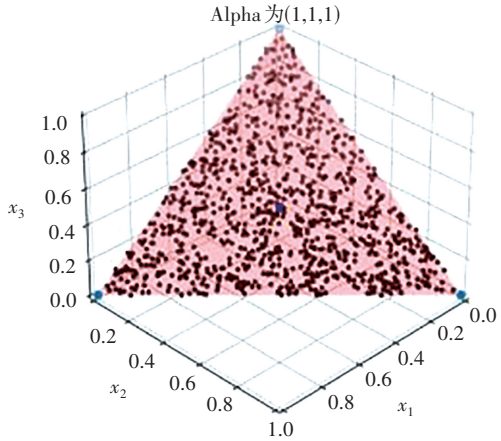
服从狄利克雷分布的 K 维随机变量的概率密度函数存在于 K 维空间上的 $K-1$ 维概率单纯形上,它的形状取决于参数 $\boldsymbol{\alpha}$。当 $\boldsymbol{\alpha}$ 的分量 α_i 都相等时,被称作对称狄利克雷分布,是狄利克雷分布的一个特例。当 $K=3$ 时,狄利克雷分布的数据点分布通常具有以下几种情况:

当 $\alpha_i<1$ 时,数据点聚集在单纯形的边缘,呈稀疏分布;当 $\alpha_i=1$ 时,数据点均匀分布在单纯形上,呈均匀分布;当 $\alpha_i>1$ 时,数据点变得更加集中在单纯形的中心,呈集中分布;当 α_i 不相等时,数据点变得更加集中在数值较大的一侧,呈随机分布。

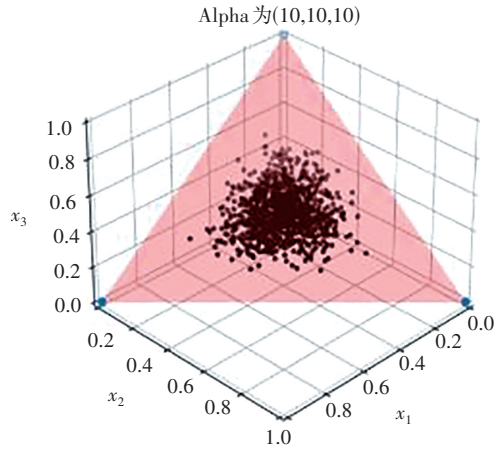
图 1 中,根据不同的 $\boldsymbol{\alpha}$ 值,在三维空间中绘制了从狄利克雷分布采样的 1 000 个点在单纯形上的分布情况,分别为图 1(a)稀疏分布、图 1(b)均匀分布、图 1(c)集中分布、图 1(d)随机分布。



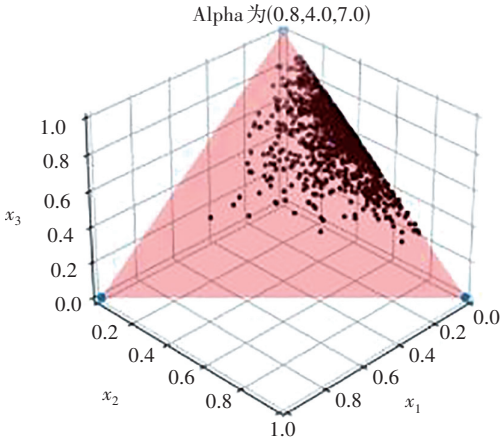
(a) 稀疏分布



(b) 均匀分布



(c) 集中分布



(d) 随机分布

图 1 不同参数值下狄利克雷分布的数据点分布

Fig. 1 Data point distribution of Dirichlet distribution under different parameter values

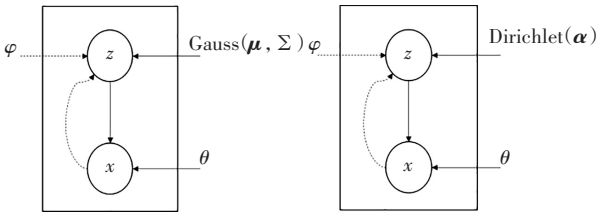
2.2 狄利克雷变分自编码

狄利克雷变分自编码(Dirichlet Variational Autoencoder, DirVAE)是 Joo 等[25]在 2020 年提出的一种基于狄利克雷分布为先验分布的变分自编码器。在传统的变分自编码器(VAE)中,隐空间被假设为多元高斯分布,这种分布仅能适应连续型数据。然而,对于

离散型数据或多模态数据,高斯分布并不适用。为了解决这个问题,DirVAE 提出了狄利克雷分布作为隐空间的先验分布,它不仅可以更好地学习隐空间的分布,还解决了高斯 VAE 出现的模型崩溃问题。

2.2.1　网络结构

DirVAE 的网络结构由编码网络和解码网络两部分组成。编码(推断)网络将输入数据映射到一个隐空间,并输出每个潜在维度对应的狄利克雷参数。解码(生成)网络根据隐变量重建输入数据,并计算重构误差。图 2 描述了 VAE 和 DirVAE 的图模型。



(a) VAE　　(b) DirVAE

图 2　两种 VAE 图模型

Fig. 2　Two VAE graph models

图 2 中,φ、θ 分别表示网络参数,z 表示隐空间的变量,分别服从参数为 $\boldsymbol{\mu}$、Σ 的高斯分布和参数为 $\boldsymbol{\alpha}$ 的狄利克雷分布。

2.2.2　训练过程

DirVAE 的训练过程与 VAE 类似,主要包括前向传播、损失计算和反向传播。其中值得注意的是损失函数的计算有所差异,涉及不同的证据下界(Evidence Lower Bound,ELBO)。DirVAE 的优化策略本质为最大似然法则,即最大化概率 $\log p(x)$。

$$\begin{aligned}\log p(x) &= \int_z q(z\mid x)\log p(x)\,\mathrm{d}z = \\ &\int_z q(z\mid x)\log\frac{p(x,z)\,q(z\mid x)}{q(z\mid x)p(z\mid x)}\mathrm{d}z = \\ &\int_z q(z\mid x)\log\frac{p(x,z)}{q(z\mid x)}\mathrm{d}z + D_{\mathrm{KL}}(q(z\mid x)\,\|\,p(z\mid x))\end{aligned} \tag{3}$$

式(3)中,第一项即为 ELBO,第二项为真实后验和近似后验的 KL 散度。由于后验分布 $p(z|x)$ 的计算难以处理,对此 VAE 中引入了变分推理的思想,用分布 $q_\varphi(z|x)$ 近似后验分布 $p(z|x)$,将推理问题转化为优化问题,同样的处理被运用到了 DirVAE 中。又因为 KL 散度值大于零,所以 DirVAE 的目标函数可以重写为

$$\log p(x) = L(\varphi,\theta;x) + D_{\mathrm{KL}}(q(z|x)\,\|\,p(z|x)) \geqslant L(\varphi,\theta;x) = q_\varphi(z|x)\log p_\theta(x|z) - D_{\mathrm{KL}}(q_\varphi(z|x)\,\|\,p_\theta(z)) \tag{4}$$

式(4)中,ELBO 包含重构项和正则化项两部分。重构项用于衡量解码器网络在重构输入数据时的性能,可以使用交叉熵损失函数进行计算。正则化项是 KL 散度,用于衡量潜在空间的分布与狄利克雷先验的差异。因为狄利克雷分布可以看作是服从伽马分布的随机变量归一化后的联合分布,所以先验分布可以用多元伽马分布代替。这样,使得正则化项不仅具有闭形式的表达,而且可以通过逆累积密度函数对伽马分布重参数化。式(5)是两个多元伽马分布 Q 和 P 的 KL 散度:

$$D_{\mathrm{KL}}(Q\,\|\,P) = \sum_i(\log\Gamma(\alpha_i) - \log\Gamma(\hat{\alpha}_i) + (\hat{\alpha}_i - \alpha_i)\psi(\hat{\alpha}_i)) \tag{5}$$

其中,$\psi(\hat{\alpha}_i)$ 表示对数伽马函数的导数,α_k、$\hat{\alpha}_k$ 分别表示为两个多元伽马分布的参数。

2.3　嵌入聚类

嵌入聚类是由 Xie 等[22]在 2016 年提出并应用于 DEC 中的一种聚类方法。这种方法的核心思想是将预训练自编码网络的编码部分与嵌入的聚类层相结合,再进行聚类分析。在微调过程中,使用式(6)中的聚类损失函数 L_c 作为目标函数进行优化。

$$L_c = D_{\mathrm{KL}}(P\,\|\,Q) = \sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}} \tag{6}$$

式(6)中,Q 表示原始分布,P 表示目标分布,q_{ij} 表示嵌入空间中点 z_i 的软标签,是由嵌入点 z_i 和聚类中心 u_j 根据式(7)的学生-T 分布[26]所得:

$$q_{ij} = \frac{(1 + \|z_i - u_j\|^2/a)^{-\frac{a+1}{2}}}{\sum_j(1 + \|z_i - u_j\|^2/a)^{-\frac{a+1}{2}}} \tag{7}$$

其中,a 是自由度,由于在无监督环境下交叉验证,通常设 $a=1$;z_i 表示嵌入空间中第 i 个样本点;u_j 表示第 j 个聚类中心;q_{ij} 表示第 i 个样本点属于第 j 个类的概率;并且目标分布 p_{ij} 是根据式(8)所得:

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_j\left(q_{ij}^2/\sum_i q_{ij}\right)} \tag{8}$$

3　聚类模型

在本节中,将结合 DirVAE 和嵌入聚类算法进行模型建立。具体来说,通过利用狄利克雷分布作为

DVADEC 模型中隐变量的先验分布这一优势,使得网络能够更好地对离散数据或多模态数据进行建模,从而捕捉到有利于聚类的特征。然后通过嵌入在隐空间的聚类层增强数据点的置信度,进一步提升聚类结果的准确性。接下来,将从目标函数、网络结构和算法过程 3 个方面介绍该模型。

3.1 目标函数

DVADEC 模型的目标函数主要包含两个部分,具体表达式如下:

$$L=L_{\text{Dirvae}}+\gamma L_{\text{c}} \tag{9}$$

其中,L_{Dirvae} 表示 DirVAE 作为特征提取器的网络损失,其表达式如式(10)所示;L_{c} 表示聚类损失,其表达式如式(6)所示;$\gamma>0$,表示用来控制嵌入空间的扭曲程度系数。

这里的 L_{Dirvae} 形式上同式(4)是一致的,但其中的正则化项有所差异。在 DirVAE 中,正则化项是关于两个多元伽马分布的 KL 散度,这里通过对狄利克雷分布的研究,发现两个狄利克雷分布之间的 KL 散度存在闭形式的解,故尝试用式(2)替换式(4)中的正则化项。但是狄利克雷分布没有高斯分布那样具有可重参数化函数,这是一个显著的局限性。为了解决这个问题,决定采用 Laplace 方法对先验狄利克雷分布进行近似,将其转换为逻辑高斯分布的形式,这样可以有效地解决狄利克雷分布的不可重参数化问题。主要借鉴 Mackay[27] 理论,即可以通过软最大拉普拉斯(Softmax Laplace)近似来将狄利克雷分布近似为多元高斯分布。由此,狄利克雷分布的参数 $\boldsymbol{\alpha}$ 和高斯分布的参数 $\boldsymbol{\mu}$、Σ 存在以下关系:

$$\mu_i=\log\alpha_i-\frac{1}{K}\sum_k\log\alpha_k$$

$$\sum\nolimits_i=\frac{1}{\alpha_i}\left(1-\frac{2}{K}\right)+\frac{1}{K^2}\sum_k\frac{1}{\alpha_k}$$

其中,K 表示 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\mu}$ 的维度,Σ 表示多元高斯分布的方差且假设为一个对角矩阵。通过这样的处理,正则化项被定义为两个多元高斯分布的 KL 散度。最终,重新表达 Laplace 近似之后的狄利克雷变分自编码目标函数,具体表达式如式(10)所示:

$$L_{\text{Dirvae}}=\sum_{i=1}^{N}\left[E_{q(z|x,\varphi)}[\log(p(x\mid z,\theta))]-\left(\frac{1}{2}\left\{\log\frac{\left|\sum\nolimits_1\right|}{\left|\sum\nolimits_0\right|}+\operatorname{tr}\left(\sum\nolimits_1^{-1}\sum\nolimits_0\right)+(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)-K\right\}\right)\right] \tag{10}$$

式(10)中,第一项表示重建误差,φ、θ 分别为推断网络和生成网络参数;第二项表示两个多元高斯分布的 KL 散度,$\operatorname{tr}(\sum_1^{-1}\sum_0)$ 表示矩阵的迹,$(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}$ 表示对应变量的转置。以下对两个多元高斯分布的 KL 散度进行详细推导。

假设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_K)$ 为服从多元高斯分布的随机向量,且有 $p_0(\boldsymbol{x})$、$p_1(\boldsymbol{x})$ 两个多元高斯分布:

$$p_0(\boldsymbol{x})=\frac{1}{(2\pi)^{\frac{K}{2}}\left|\sum_0\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\sum\nolimits_0^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right) \tag{11}$$

$$p_1(\boldsymbol{x})=\frac{1}{(2\pi)^{\frac{K}{2}}\left|\sum_1\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right) \tag{12}$$

将式(11)、式(12)代入 KL 散度计算公式,有

$$D_{\text{KL}}(p_0\parallel p_1)=E_{p_0}[\log(p_0)\parallel\log(p_1)]=E_{p_0}\log(p_0)+E_{p_0}\log(-p_1) \tag{13}$$

第一项计算结果为

$$E_{p_0}\log(p_0)=-\frac{K}{2}(\log 2\pi+1)-\frac{1}{2}\log\left|\sum\nolimits_0\right| \tag{14}$$

接下来,主要计算第二项。

$$E_{p_0}\log(-p_1)=E_{p_0}\left[\frac{K}{2}\log 2\pi+\frac{1}{2}\log\left|\sum\nolimits_1\right|+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]=\frac{K}{2}\log 2\pi+\frac{1}{2}\log\left|\sum\nolimits_1\right|+\frac{1}{2}E_{p_0}\left[(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right] \tag{15}$$

下面利用迹的恒等式进行推导。

$$E_{p_0}\left[(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]=E_{p_0}\left[\operatorname{tr}\left((\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right)\right]=E_{p_0}\left[\operatorname{tr}\left(\sum\nolimits_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\right)\right]=\operatorname{tr}\left(\sum\nolimits_1^{-1}E_{p_0}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}-\boldsymbol{\mu}_1\boldsymbol{x}^{\mathrm{T}}-\boldsymbol{x}\boldsymbol{\mu}_1^{\mathrm{T}}+\boldsymbol{\mu}_1\boldsymbol{\mu}_1^{\mathrm{T}}]\right)=$$

$$\begin{aligned}&\mathrm{tr}\left(\sum\nolimits_1^{-1}\left[\sum\nolimits_0+\boldsymbol{\mu}_0\boldsymbol{\mu}_0^{\mathrm{T}}-\boldsymbol{\mu}_1\boldsymbol{\mu}_0^{\mathrm{T}}-\boldsymbol{\mu}_0\boldsymbol{\mu}_1^{\mathrm{T}}+\boldsymbol{\mu}_1\boldsymbol{\mu}_1^{\mathrm{T}}\right]\right)=\\&\mathrm{tr}\left(\sum\nolimits_1^{-1}\sum\nolimits_0+\sum\nolimits_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\right)=\\&\mathrm{tr}\left(\sum\nolimits_1^{-1}\sum\nolimits_0\right)+(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)\end{aligned}\tag{16}$$

将式(16)代入式(15)得到第二项结果:

$$\begin{aligned}&E_{p_0}\log(-p_1)=\frac{K}{2}\log 2\pi+\frac{1}{2}\log\left|\sum\nolimits_1\right|+\\&\frac{1}{2}\left[\mathrm{tr}\left(\sum\nolimits_1^{-1}\sum\nolimits_1\right)+(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)\right]\end{aligned}\tag{17}$$

最后,将式(17)代入式(14)可得两个多元高斯分布的KL散度:

$$\begin{aligned}&D_{\mathrm{KL}}(p_0\parallel p_1)=\\&\frac{1}{2}\left\{\log\frac{\left|\sum\nolimits_1\right|}{\left|\sum\nolimits_0\right|}+\mathrm{tr}\left(\sum\nolimits_1^{-1}\sum\nolimits_0\right)+\right.\\&\left.(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\sum\nolimits_1^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)-K\right\}\end{aligned}\tag{18}$$

3.2　网络结构

在整个网络构建过程中,根据 IDEC 搭建了类似的网络,这样处理有两个好处。首先,保留的重构项能够保证生成数据的局部结构,防止嵌入空间被扭曲。其次,通过在隐空间中嵌入聚类层,可以进一步提高软标签的置信度。网络结构主要包含 3 部分:编码(推断)网络、解码(生成)网络和聚类层,DVADEC 模型网络结构如图 3 所示。

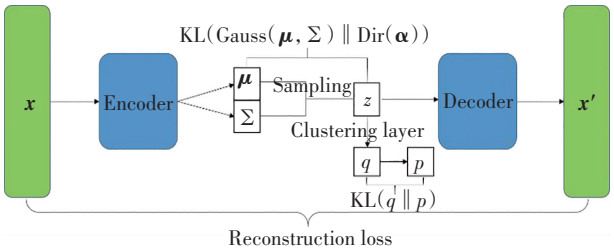


图 3　DVADEC 网络结构图

Fig. 3　Network architecture diagram of DVADEC

Encoder: $\boldsymbol{\mu}$、$\Sigma=f_\varphi(x)$,f_φ 表示一个非线性映射,x 是输入变量,输出是近似后验分布 $q_\varphi(z|\boldsymbol{x})$ 的参数。

Decoder: $\boldsymbol{x}'=g_\theta(z)$,$g_\theta$ 表示一个非线性映射,z 是输入变量,输出是由生成网络生成的样本。

Clustering layer:隐变量 z 和聚类中心 u,输入式(7)计算软标签 q,再根据式(8)更新 p。

3.3　算法步骤

DVADEC 模型算法

输入　X:输入数据;K:聚类的数量;T:目标分布更新间隙;δ:停止迭代阈值;MaxIter:最大迭代次数。

输出　W:网络权重;u:聚类的中心;y_{pred}:聚类的标签。

初始化网络权重 W 和聚类中心 u;

批量载入数据,预训练 DirVAE 网络,并保存网络权重 W;

将权重 W 载入 DVADEC 模型,将所有输入模型,并保存输出的潜在特征和重建数据;

用 K-means 对潜在特征聚类,得到标签 y_{pred},并赋值 $y_{\mathrm{pred\_last}}=y_{\mathrm{pred}}$;

循环迭代 iter $\in\{1,2,\cdots,\mathrm{MaxIter}\}$;

若 iter%T==0:

批量载入数据到 DVADEC 模型,计算嵌入点 $\{z_i=f_\varphi(x_i)\}_{i=1}^n$;

通过式(7)更新软分配 q、式(8)更新目标分布 p,根据式(9)进行模型微调;

根据 $y_{\mathrm{pred}}=\mathrm{argmax}(q)$ 得到新的标签;

若 $(y_{\mathrm{pred}}\neq y_{\mathrm{pred\_last}})/n<\delta$,即停止迭代;

下一批次数据输入到 DVADEC 网络;

更新网络权重 W 和聚类中心 u。

4　仿真实验与结果分析

4.1　数据集

为了验证 DVADEC 模型的聚类效果,选取 4 个公开的基准数据集进行试验(表 1)。

表 1　数据集统计

Table 1　Dataset statistics

数据集	样本数	维　度	类别数
MNIST	70 000	1 * 28 * 28	10
F-MNIST	70 000	1 * 28 * 28	10
USPS	9 298	256	10
REUTERS-10K	10 000	2 000	4

MNIST 是一个包含手写数字灰度图像的数据集,共有 10 个类别,每个图像尺寸为 28×28 像素。

F-MNIST(Fashion-MNIST)是一个由 10 个不同种类的时尚服饰和配件灰度图像组成的数据集。每个图像尺寸为 28×28 像素,分别表示 1 个时尚服饰或配件。

USPS 数据集是一个包含美国邮政服务(USPS)的手写数字灰度图像数据集。每个图像都表示一个手写

的数字,范围从0到9共10类。每个图像的尺寸为16×16像素。

REUTERS,原始路透社(Reuters)数据集,其中,大约有810 000篇英文新闻报道被标记了一个类别树。从中随机抽取其中10 000篇文档,这里命名为REUTERS-10k。

4.2 聚类性能评估指标

为了评估聚类性能,采用以下3种评估指标。

(1) 聚类精确度(Clustering Accuracy,CA),该值越高,聚类性能越好。用A_c表示CA值,其计算公式如下:

$$A_c=\max_{m\in M}\frac{\sum_{i=1}^{N}1\{l_i=m(c_i)\}}{N}$$

其中,N表示样本的总数量,l_i表示原始数据的真实标签,c_i是从模型获取到的聚类分配,M是在聚类分配和标签之间的所有可能的一对一映射的集合。

(2) 归一化互信息(Normalized Mutual Information,NMI),是一种用来量化两组数据的相似性或共享信息量的度量方式,取值范围在[0,1]。用I_{NM}表示NMI值,其计算公式如下:

$$I_{NM}(X,Y)=\frac{I_M(X,Y)}{H(X)+H(Y)}$$

X、Y为两组随机变量;$I_M(X,Y)$表示X和Y的互信息;$H(X)$和$H(Y)$分别是X、Y的信息熵。

(3) 调整兰德指数(Adjusted Rand Index,ARI),是对RI度量的一种调整,用于衡量两个数据集聚类结果之间相似性的度量方法,取值范围在[-1,1],ARI越大,表示聚类效果越好。用I_{AR}表示ARI值,其具体计算公式如下:

$$I_{AR}=\frac{2(T_P\times T_N-F_P\times F_N)}{F_P{}^2+F_N{}^2+2T_P\times T_N+(T_P+T_N)\times(F_P+F_N)}$$

T_P表示将相似样本划分到同一簇,T_N表示将不相似样本划分到不同簇,这两种属于正确决策;F_P表示将不相似样本划分到同一簇,F_N表示将相似样本划分到不同簇。

4.3 实验设置和步骤

实验使用Ubuntu20.04.2LTS操作系统,PyTorch深度学习开发框架,用Python作为开发语言;CPU为Intel(R) Core(TM) i5-12400F,GPU为NVIDIA GeForce GTX 1660 SUPER。根据图3所示的网络结构配置网络,其中编码网络和解码网络分别是input-512-256-128-10和10-128-256-512-output。具体地,在训练过程中选用Adam优化器[28]和Xavier初始化器进行训练。学习率$l_r=0.001$,迭代次数$E_p=200$,批量样本大小$B_s=256$,聚类损失权重系数$\gamma=0.0001$,先验分布参数$\alpha=0.1$,参数K设置为数据集的类别个数。除了输入、输出和嵌入层外,所有的内部层都使用ReLU非线性激活函数。

4.4 实验设计与分析

4.4.1 模型聚类性能对比实验

鉴于该聚类模型是在IEDC的基础上改进的,所以为了实验的公正性,主要同与之相关的嵌入聚类方法进行比较。同时为了实验的完整性,还引入了传统的聚类算法K-means和GMM。采用CA、NMI和ARI 3项评估指标,与其他聚类方法进行比较,表2展示了DVADEC模型在各个数据集上的聚类优势。

根据表2中展示的实验结果,可以得出以下结论:第一,相较于传统聚类法方法(K-means,GMM等),基于深度网络学习潜在特征,然后再聚类的方法在性能上有更显著的提升;第二,DVADEC模型在图像和文本数据集上的聚类精度明显高于其他两种嵌入聚类算法。具体地,在MNIST和F-MNIST数据集上的精度分别达到了97⁺%和70⁺%,相较于IDEC算法,精度提高了10⁺%。由此可见,DirVAE网络对潜在特征提取是非常有效的。

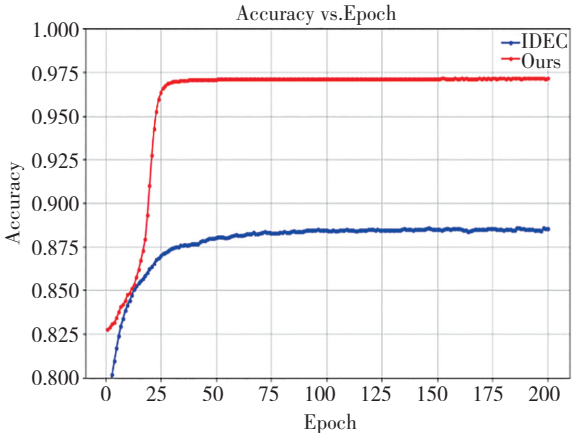
表2 不同聚类方法在不同数据集上的聚类精度对比

Table 2 Comparison of clustering accuracy of different clustering methods on different datasets

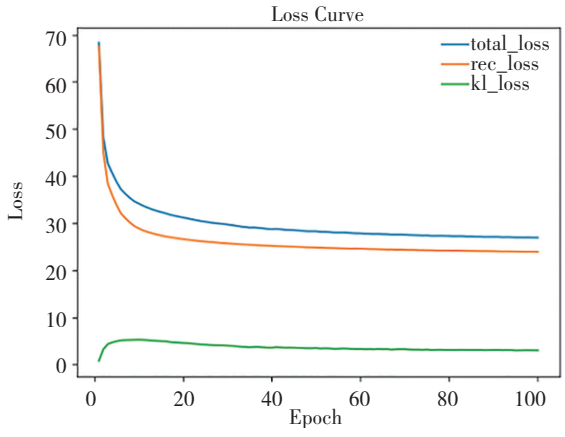
Method	MNIST-full			MNIST-test			F-MNIST			USPS			REUTERS-10K		
	A_c	I_{NM}	I_{AR}	A_c	I_{NM}	I_{AR}	A_c	I_{NM}	I_{AR}	A_c	I_{NM}	I_{AR}	A_c	I_{NM}	I_{AR}
K-means	0.532	0.500	0.372	0.533	0.499	0.387	0.474	0.512	0.370	0.657	0.620	0.533	0.515	0.309	0.299
GMM	0.540	0.509	0.386	0.537	0.501	0.390	0.523	0.409	0.353	—	—	—	0.535	0.337	0.308
DEC	0.863	0.834	0.751	0.840	0.830	0.753	0.595	0.510	0.403	0.758	0.769	0.688	0.756	0.481	0.480
IDEC	0.874	0.859	0.773	0.881	0.867	0.773	0.609	0.513	0.410	0.759	0.777	0.695	0.530	0.209	—
DVADEC	0.971	0.926	0.938	0.973	0.931	0.940	0.702	0.683	0.578	0.862	0.854	0.834	0.801	0.537	0.625

此外,为了更全面地了解模型在训练过程中的表现,这里使用 MNIST 数据集对精度和损失进行记录,以便能够观察它们在训练过程中的变化情况。

图 4(a)展示了 IDEC 和 DVADEC 两种聚类算法在 MNIST 数据集上训练过程的精度变化情况;图 4 (b)展示了 DVADEC 模型在 MNIST 数据集上预训练的损失函数。根据观察结果,可以得出以下 3 点:第一,DVADEC 模型相较于 IDEC 模型,其收敛速度更快;第二,该聚类方法精度远高于 IDEC;第三,损失函数在前 20 个 E_p 的收敛更快,同时重构损失和 KL 散度趋于稳定。



(a) Accuracy vs Epoch



(b) Loss Curve

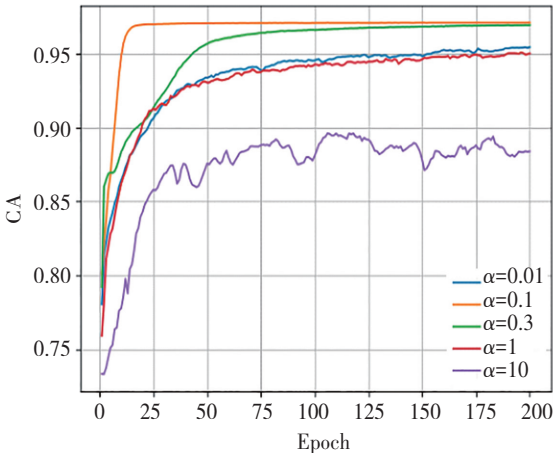
图 4　MNIST 数据集上训练的精度和损失

Fig. 4　Accuracy and loss during training on MNIST dataset

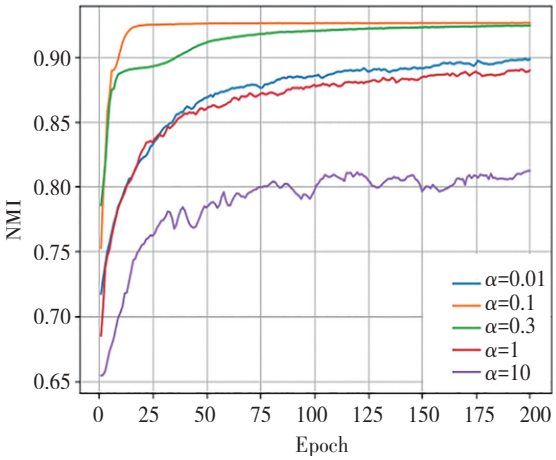
4.4.2　先验分布参数对 CA 和 NMI 的影响

在本实验中,为了更好地分析和评估模型的性能,并作出相应地调整和改进,用 MNIST 数据集进行训练,通过设置不同的先验分布参数来分析对 CA 和 NMI 的影响,α 分别取值为 0.01、0.1、0.3、1 与 10,实验结果如图 5 所示。

图 5 的实验结果可见:特征空间中先验分布的参数值对 CA 和 NMI 有显著影响。首先,当先验参数值过小时,聚类精度较低,其原因可能是被误分类的点无法在训练过程中拉回到其所属的类别,同时精度值的波动也相对较大,可见整个模型的性能不稳定;其次,先验参数值过大时,虽然收敛相对平缓,但聚类效果并不理想,这是因为参数值过大会使得嵌入空间中的分布呈现为图 1 (c)所示的集中分布,不利于聚类。最终实验结果显示,将先验参数值设置在 0.1~0.3 之间能够获得较好的聚类效果。



(a) CA vs Epoch



(b) NMI vs Epoch

图 5　不同先验值对 CA 和 NMI 影响

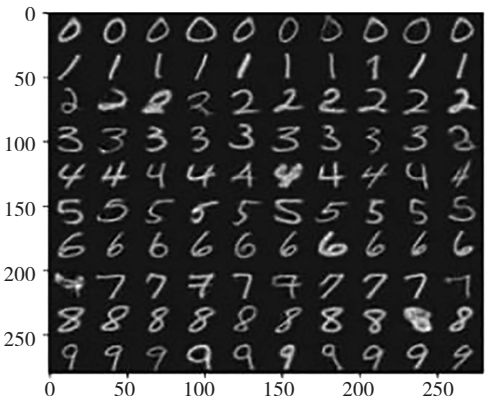
Fig. 5　Effects of different prior values on CA and NMI

4.4.3　模型的生成能力与可视化

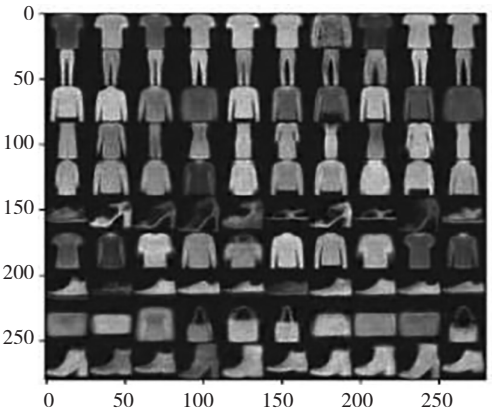
该实验旨在通过视觉呈现来更直观地展示模型的生成能力。为此,实验中对比了使用 IDEC 模型重构的样本与使用 DVADEC 模型生成的样本。通过比较这两种生成样本,可以更好地评估和理解模型的生成能力。

IDEC 算法通过编码网络提取有效特征,然后使用这些特征进行聚类,并最终通过重构来生成与输入数据相似的样本。相比之下,DVADEC 模型首先通过网络学习适合聚类的概率分布参数,然后根据这个分布参数进行低维特征的随机采样;接着,通过聚类层对这些采样到的低维特征进行聚类;最后,从相应的类别中随机采样低维特征,再将其转化为高维特征,从而生成与输入数据相似的样本。两种方法得到的样本如图 6 所示。

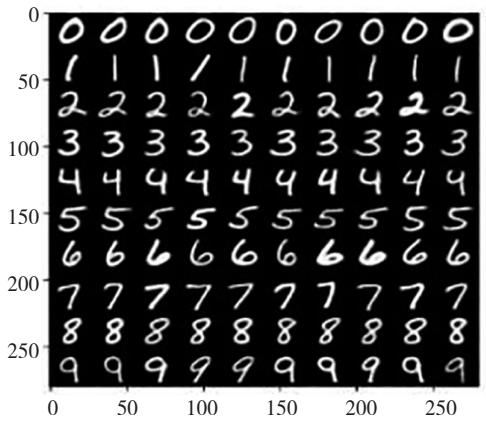
图 6 展示了 IDEC 算法在 MNIST 和 F-MNIST 数据集上训练得到的重构样本,图 6(c)和图 6(d)展示了 DVADEC 模型在这两个数据集上训练得到的生成样本。通过对比可以观察到:相较于 IDEC 的重构样本,DVADEC 生成的样本具有更清晰平滑的特点。此外,图 6(a)还存在明显的误分类现象,如数字 7 和 9,这意味着这两个簇之间的特征差异不明显,即学到的特征区分度较低。相反,图 6(c)生成的样本展现出明显的类间差异,且每个簇中样本相似度很高,这说明 DirVAE 网络所提取的特征更有助于聚类。



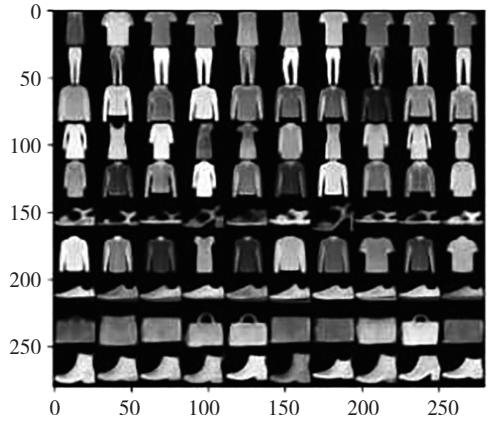
(a) Mnist samples(IDEC)



(b) F-mnist samples(IDEC)



(c) Mnist samples



(d) F-mnist samples

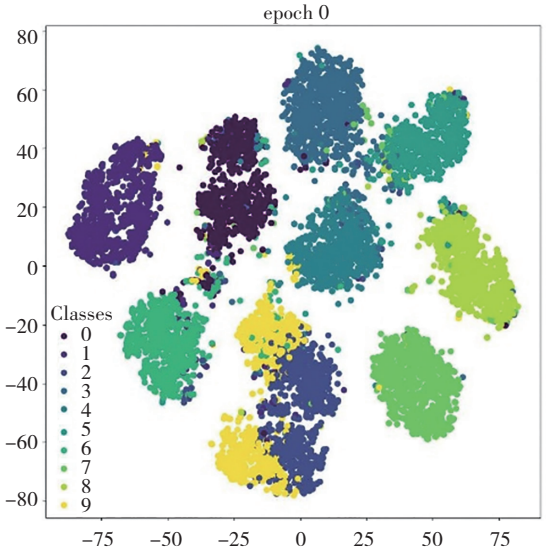
图 6 MNIST 和 F-MNIST 数据集的生成样本

Fig. 6 Generated samples of MNIST and F-MNIST datasets

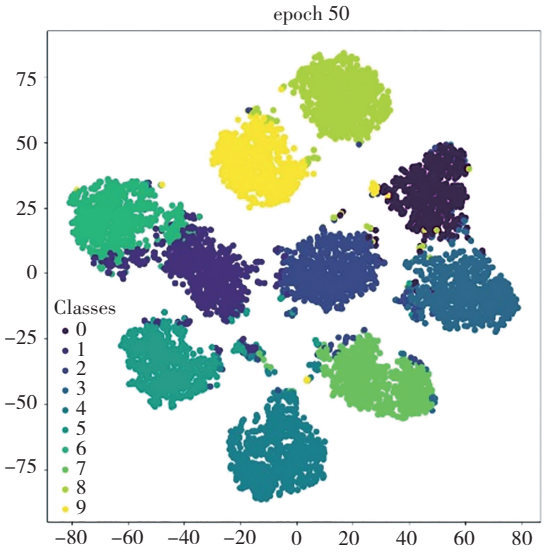
为了进一步观察 DVADEC 模型在隐空间中获得的聚类特征的分离程度,本实验使用 T-SNE[26] 算法对经过聚类的 MNIST 测试集的隐空间进行可视化。将原本的 10 维隐变量降到 2 维,并在 2 维空间中展示可视化结果,可视化效果如图 7 所示。

图 7 分别展示了在不同训练迭代次数下的 DVADEC 模型隐空间聚类的准确性。当 $E_P = 0$ 时,准确度为 82%;当 $E_P = 50$ 时,准确度为 87%;当 $E_P = 100$ 时,准确度为 91%;当 $E_P = 200$ 时,准确度达到 97%以上。

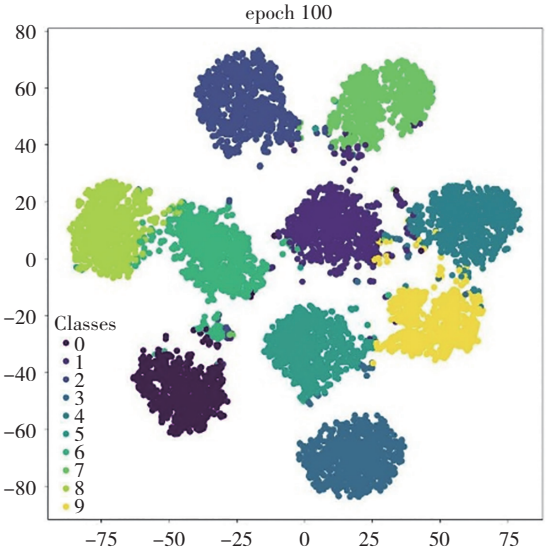
每种颜色代表一个聚类簇,以 MNIST 数据集为例,每个聚类簇表示 0—9 中的一个数字。根据图 7 的可视化结果可以得出以下结论:首先,隐空间中的聚类簇数量与实际类别数量一致,说明该模型提取的特征具有明显的分离性;其次,随着训练迭代次数的增加,聚类的准确性也提高,表明提取到的特征更具可分性。



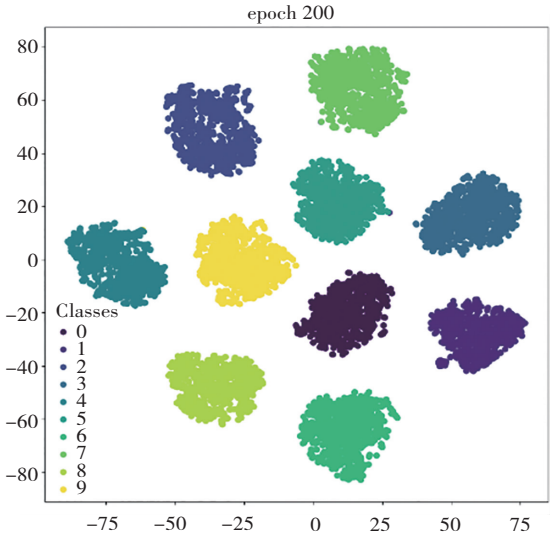
(a) $E_P=0$



(b) $E_P=50$



(c) $E_P=100$



(d) $E_P=200$

图 7　MNIST 数据集嵌入特征的可视化

Fig. 7　Visualisation of embedded features in MNIST datasets

5　结论与展望

本研究提出一种将狄利克雷变分自编码与嵌入聚类相结合的深度聚类方法。在这种方法中,狄利克雷变分自编码被用作预训练网络和特征提取器,并且在隐空间中嵌入了聚类层,用于执行聚类任务。实验结果显示:这种聚类模型显著增强了捕捉有效特征的能力,并在基准数据集上取得了良好的聚类效果。由此可见,该方法对于聚类研究具有重要意义。

此外,DVADEC 模型存在一个显著的局限性,即隐空间先验分布的重参数化问题。为了解决这个问题,未来的研究将围绕分布的重参数化或近似展开。希望能够实现采样狄利克雷分布,通过将狄利克雷分布直接作为先验来训练网络,进一步提高聚类准确性。同时,与最新的聚类算法相比,本研究还有许多改进的空间,例如使用数据增强或卷积网络预处理等技术对输入的原始数据进行处理,以进一步提高隐空间中特征的质量。这些改进将有助于提升聚类算法的性能。

参考文献(References):

[1] CUI X, GAO J, POTOK T E. A flocking based algorithm for document clustering analysis[J]. Journal of Systems Architecture, 2006, 52(8-9): 505-515.

[2] CARULLO M, BINAGHI E, GALLO I. An online document clustering technique for short web contents[J]. Pattern Recognition Letters, 2009, 30(10): 870-876.

[3] CUSHING B, POOT J. Crossing boundaries and borders:

Regional science advances in migration modelling[J]. Papers in Regional Science, 2004, 83(1): 317-338.

[4] CHEN Y, WANG J Z, KROVETZ R. CLUE: Cluster-based retrieval of images by unsupervised learning[J]. IEEE Transactions on Image Processing, 2005, 14(8): 1187-1201.

[5] GOLDBERGER J, GORDON S, GREENSPAN H. Unsupervised image-set clustering using an information theoretic framework[J]. IEEE Transactions on Image Processing, 2006, 15(2): 449-458.

[6] XIE P, XING E. Integrating image clustering and codebook learning[C]//Proceedings of the 29th AAAI Conference on Artificial Intelligence. New York: ACM, 2015: 1903-1909.

[7] GAO Y, LI X, DONG M, et al. An enhanced artificial bee colony optimizer and its application to multi-level threshold image segmentation[J]. Journal of Central South University, 2018, 25(1): 107-120.

[8] SHI J, MALIK J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[9] XU Z, YAN F, QI Y. Bayesian nonparametric models for multiway data analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(2): 475-487.

[10] BORGATTI S P, MEHRA A, BRASS D J, et al. Network analysis in the social sciences[J]. Science, 2009, 323(5916): 892-895.

[11] XU Z, LIU B, ZHE S, et al. Variational random function model for network modeling[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 30(1): 318-324.

[12] HARTIGAN J A, WONG M A. Algorithm AS 136: A k-means clustering algorithm[J]. Applied Statistics, 1979, 28(1): 100-108.

[13] MURTAGH F. A survey of recent advances in hierarchical clustering algorithms[J]. The Computer Journal, 1983, 26(4): 354-359.

[14] SCHUBERT E, SANDER J, ESTER M, et al. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN[J]. ACM Transactions on Database Systems, 2017, 42(3): 1-21.

[15] HAN M, LIU J, SUN Y. A background modeling algorithm based on improved adaptive mixture Gaussian[J]. Journal of Computers, 2013, 8(9): 2239-2244.

[16] MACKIEWICZ A, RATAJCZAK W. Principal components analysis (PCA)[J]. Computers & Geosciences, 1993, 19(3): 303-342.

[17] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. Stat, 2014, 1050(1): 1-14.

[18] FLEURET F, Fast binary feature selection with conditional mutual information[J]. Journal of Machine Learning Research, 2004, 5(4941): 1531-1555.

[19] XU Y, HUANG Q, WANG W, et al. Fully deep neural networks incorporating unsupervised feature learning for audio Tagging[J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2016, 25(6): 1230-1241.

[20] PENG X, XIAO S, FENG J, et al. Deep subspace clustering with sparsity prior[C]//Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: ACM, 2016: 1925-1931.

[21] TIAN F, GAO B, CUI Q, et al. Learning deep representations for graph clustering[C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. New York: ACM, 2014: 1293-1299.

[22] XIE J, GIRSHICK R, FARHADI A, et al. Unsupervised deep embedding for clustering analysis[C]//Proceedings of the 33rd International Conference on Machine Learning. New York: ACM, 2016: 478-487.

[23] GUO X, GAO L, Liu X, et al. Improved deep embedded clustering with local structure preservation[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. New York: ACM, 2017: 1753-1759.

[24] GUO X, LIU X, ZHU E, et al. Deep clustering with convolutional autoencoders[C]//International Conference on Neural Information Processing. Cham: Springer International Publishing, 2017: 373-382.

[25] JOO W, LEE W, PARK S, et al. Dirichlet variational autoencoder[J]. Pattern Recognition, 2020, 107(1): 1-10.

[26] VANDER M L, HINTON G. Visualizing data using t-sne[J]. Journal of Machine Learning Research, 2008, 9(11): 2579-2605.

[27] MACKAY D J C. Choice of basis for Laplace approximation[J]. Machine Learning, 1998, 33(1): 77-86.

[28] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. ICLR, 2015, 1(1): 1-15.

责任编辑:李翠薇