学校代码 **10125**　　专业代码 **081203**

山西财经大学

硕 士 学 位 论 文

题目 基于对比学习的深度聚类算法研究

姓　　名 邢若苇

专　　业 计算机应用技术

研究方向 数据挖掘与商务智能

所属学院 信息学院

指导教师 冀素琴

二〇二三年 六月 十九日

University Code 10125 **Major Code** 081203

Shanxi University of Finance & Economics

Thesis for Master's Degree

Title Research on Deep Clustering Algorithm Based on Contrastive learning

Name Ruowei Xing

Major Computer Application Techonogy

Research Orientation Data Mining and Business Intelligence

School School of Information

Tutor Suqin Ji

June 19，2023

摘　要

聚类主要针对没有标签的数据集，按照某种特定的标准将数据进行分组，是无监督学习中的重要算法之一。深度聚类通过将神经网络与聚类算法相结合，同时优化特征空间与聚类结果，极大地提升了算法性能，深度聚类算法主要解决的问题是如何学习到产生更优结果的区分性表示。对比学习可以学到更高维度、更本质的代表性特征，在表示学习中得到了广泛关注。

由于对比学习的良好表现，出现了联合优化对比学习与深度聚类的算法，此类方法通常利用对比学习基本框架学习表示，基于该表示进行聚类。然而对比学习强调数据增强的重要性，通过使数据的不同增强在特征空间中尽可能一致对每个实例进行区分，而聚类目标是对实例进行分组。在将对比学习与深度聚类相结合时，若直接遵循对比学习的基本框架，会忽略聚类目标，此时学到的表示可能不是聚类的最佳表示，使得聚类性能受到限制。因此结合时需要考虑不同实例之间的相似性所构成的自然群组。本文针对以上问题，面向聚类中不同的数据关系提出两种算法，将对比学习与深度聚类相结合学习聚类友好型表示，以提升深度聚类的效果。主要研究内容包括以下两个方面：

（1）基于自步学习的实例级别对比聚类算法。该算法引入自步学习的思想，通过由简到难的方式对数据进行聚类。算法共分为两阶段，第一阶段通过考虑数据与簇之间的关系，对容易区分的数据进行了初步聚类，在得到的潜在空间内易分数据分布在相应簇中心周围，以此获得成对相似性关系。第二阶段使用对比学习进行训练，过程中难分的样本逐渐易分，各个簇内样本更加紧凑，不同簇间样本远离。对比学习中正负例由第一阶段获得的成对相似性来构造，从而进行实例级对比学习。该算法在常用数据集上的实验结果与多个算法结果相比都有不同程度的提升，说明该算法可以获得更好的聚类效果。

（2）基于图结构的实例-簇级别对比聚类算法。该算法通过考虑数据与数据间关系、簇与簇间关系来实现对比聚类，将常用的实例级别提升到簇级别。主要利用图结构来捕捉数据间的潜在关系，采用深度子空间聚类的基本思想，在自编码器中加入自表达层获取图结构，从中反应样本的邻域信息。通过得到的图结构来构造对比学习中的正负例，同时将特征矩阵的列作为数据的聚类预测，在列方

向进行簇级别对比学习。在实例级别与簇级别都融合了潜在的类别信息，从两方面进行训练同时进行特征学习与簇分配。该算法在三个数据集上进行了实验，与多个先进的聚类算法进行对比，验证了该算法的可行性与有效性。

本文提出两种深度聚类算法，通过考虑不同角度数据间的关系将对比学习更好地与深度聚类相结合，学习聚类友好型表示，以达到提升聚类效果的目的，具有重要的理论意义与实用价值。

关键词：对比学习；深度聚类；自步学习；成对相似性；图结构

ABSTRACT

Clustering is one of the important algorithms in unsupervised learning, which mainly aims at unlabeled datasets and groups data according to some specific standard. The deep clustering algorithm greatly improves the performance of the algorithm by combining the neural network with the clustering algorithm and optimizing the feature space and clustering results at the same time. The deep clustering algorithm mainly solves the problem of how to learn the differentiated representation to produce better results. Contrastive learning can learn higher-dimensional and more essential representative features, which has received widespread attention in representation learning.

Due to the good performance of contrastive learning, some algorithms jointly optimize contrastive learning and deep clustering. These methods use the basic framework of contrastive learning to learn the representation and cluster based on this representation. However, contrastive learning emphasizes the importance of data enhancement, distinguishes each instance by making different enhancements of the data as consistent as possible in the feature space, and the clustering goal is to group instances. When combining contrastive learning with deep clustering, if the basic framework of contrastive learning is followed directly, the clustering target will be ignored, and the learned representation may not the best representation of the cluster, which limits the clustering performance. Therefore, the natural groups formed by the similarities between different instances need to be considered when combining. In view of the above problems, this paper proposes two algorithms for different data relations in clustering, and combines contrastive learning with deep clustering to learn clustering friendly representation, so as to improve the performance of deep clustering. The research contents main include the following two aspects:

(1) An instance-level contrastive clustering algorithm based on self-paced learning. The algorithm introduces the idea of self-paced learning to cluster in a simple to difficult way. The algorithm consists of two phases. In the first phase, by considering

the relationship between data and clusters, the easily distinguished data is initially clustered, and the easily distinguished data in this potential space is distributed near the corresponding cluster center to obtain pairwise similarity relationship. In the second stage, contrastive learning is used for training, so that the indistinguishable samples are gradually easy to distinguish, making the samples within each cluster more compact and the samples between different clusters are far away. Positive and negative examples in contrastive learning are constructed from pairwise similarity results obtained in the first stage, then these examples to conduct instance-level contrastive learning. The experimental results of algorithm on common datasets are improved in different degrees compared with the results of other algorithms, indicating that algorithm can obtain better clustering effect.

(2) An instance-cluster level contrastive clustering algorithm based on graph structure. This algorithm achieves contrastive clustering by considering the relationship between data and data, and between clusters, and upgrades the common instance level to cluster level. Graph structure is used to capture the potential relationship between data. The basic idea of deep subspace clustering is used. The self-expression layer is added to auto-encoder to get the graph structure, from which the neighborhood information of the sample is reflected. Construct positive and negative examples in contrastive clustering by the graph relationship, and use the columns of the characteristic matrix as cluster predictions for data, and perform graph-based cluster level contrastive learning in column direction. Potential class information is fused at both the instance level and the cluster level, feature learning and cluster assignment are trained from two aspects simultaneously. The algorithm is experimented on three datasets and compared with several advanced clustering algorithms to verify the feasibility and validity of the algorithm.

This paper proposes two deep clustering algorithms, which combine contrastive learning with deep clustering better by considering the relationship between data from different perspectives, and learn clustering friendly representation to improve the clustering performance. It has important theoretical significance and use value.

Keywords: Contrastive learning; Deep clustering; Self-paced learning; Pairwise similarity; Graph structure

目 录

第 1 章 绪论

1.1 研究背景与意义

随着科技的不断发展，大数据时代产生的信息出现爆炸式增长，涌现出许多大批量数据。当数据达到一定的规模时，有价值的信息便被蕴藏在大量毫无规律且类型不一的低价值数据当中。由于数据的数量和复杂性在不断增加，数据分析成为一项艰巨的任务。面对具有多样性和复杂性的海量数据，如何快速从中挖掘出符合需求并对人们有价值的隐藏信息是当前需要解决的重要问题之一。

聚类是数据挖掘中的重要方法之一，其应用广泛并且具有重要地位。聚类是以无监督学习方式在数据集中寻找“自然分组”的技术，通过寻找各数据之间的潜在联系将它们划分为有意义的群组，同一群组内数据具有较高的相似性，不同群组之间具有较低的相似性。因此，聚类往往是数据挖掘任务中的一个重要预处理步骤，可以帮助分析和描述未知信息，是对样本进行标签化处理的常用方式。聚类被广泛用于商业研究、文本分析、生物医学、工业、信息安全等各个方面，通过聚类分析对事物特征进行共性研究，从而获取有价值的类簇信息。在文本分析中，聚类分析能对相关文档进行汇总，既方便对文档进行归类统计，又能提高文档检索时的准确性，给用户带来较好的使用体验[1]；在计算机视觉领域，对图像数据的初步处理和划分是必不可少的基础步骤，其中最为常见的是使用聚类算法，根据像素特征划分图像数据，使得同一类中图像具有高相似度的像素特征。

由于各行各业数据特点各不相同，使用聚类的目的不一，因此对于不同的数据集，需要根据其特点使用合适的聚类算法。传统的聚类算法已经得到了广泛的研究，提出了大量方法，大致可以分为基于划分、层次、密度和基于图的聚类算法等，这些方法速度快，适用范围广。但随着数据信息的爆炸式增长，对聚类算法的要求也逐渐增强。在高维空间中，点间距离的信息量变得更小，传统的距离度量在评估样本间相似度时容易出现误差，并且高维空间数据分布较为稀疏，簇类结构不清晰。因此将传统聚类算法直接应用在规模大、维度高、结构复杂的数据集时，通常会有较差的表现。

随着深度学习的发展，深度神经网络由于其固有的高度非线性转换特性使得

深层次大规模的特征提取成为了可能，因此出现了深度学习与聚类相结合的算法。深度聚类算法能够很好的提升聚类算法的性能，对深度聚类算法的研究在一定程度上可以解决传统聚类算法对高维数据处理的不足。深度聚类主要利用神经网络对数据进行表示学习，提取出数据的本质特征，利用该特征进行后续聚类。而聚类性能很大程度上取决于数据表示的质量，因此，如何利用深度学习去学习聚类友好表示，在统一框架下高效地进行聚类与表示学习，提高深度聚类算法的性能迅速成为研究者们关注的焦点。

近年来，对比学习由于良好的表示学习效果被广泛应用于各个领域，其着重通过数据增强学习实例的重要特征。因此已有工作将对比学习与深度聚类相结合学习表示并进行聚类任务，取得了显著的效果。但与对比学习相结合时，需要注意聚类的目标是对样本进行分组，而不是将每个样本区分开。若遵循对比学习的基本框架，将其直接与聚类将结合，使样本的不同增强尽可能一致从而学习样本的通用表示，这样仅对实例进行了区分而忽略了聚类目标，没有考虑不同实例之间的相似性所构成的自然群组，学习到的表示可能不是聚类的最佳表示，聚类性能会受到限制。因此，论文旨在研究如何更好地将对比学习与深度聚类相结合，学到聚类友好型表示，提升聚类性能。

针对以上问题，论文提出两种对比学习与聚类相结合的算法，超越单个样本，从不同方面考虑样本间的关系，纳入潜在的类别信息与聚类目标，提升深度聚类的性能，对推动深度聚类发展具有重要的意义。

1.2 国内外文献综述

由于数据表示的质量在很大程度上决定了聚类的性能，因此，如何利用神经网络来学习聚类友好表示，提高聚类性能成为无监督学习领域的研究热点。深度聚类是一系列采用深度神经网络学习聚类友好表示的聚类方法，重点在于学习面向聚类的表示。深度聚类多使用神经网络的“主分支”将输入转换为用于聚类的潜在表示，并将这些表示作为特定聚类方法的输入进行后续聚类，其动机是利用深度神经网络逼近非线性函数的能力保持联合优化中两个任务的优势。因此深度聚类损失函数通常包括网络损失与聚类损失，通过两种损失的联合训练使数据变得更具辨别力并形成集群。

从深度神经网络体系结构角度出发，本章将主要的深度聚类算法分为三大部分进行介绍：基于自编码器的深度聚类算法、基于生成式网络的深度聚类算法和基于孪生网络的深度聚类算法。孪生网络在深度聚类的使用中主要是与对比学习相结合，因此第三部分主要对基于对比学习的深度聚类算法做介绍。

1. 2. 1 基于自编码器的深度聚类算法

基于自编码器（Auto-Encoder，AE）的深度聚类算法通过自编码器直接学习特征表示，由于自编码器几乎可以与所有聚类算法相结合，因此基于自编码器的深度聚类算法是最常见、应用最为广泛的。自编码器存在各种变体，可将自编码器加入噪声形成去噪自编码器，也可将自编码器与卷积神经网络相结合形成卷积自编码器等。

2016 年 Xie[2]等人提出了聚类分析的深度嵌入方法 DEC，是深度聚类中最具有代表性的方法之一，该方法的提出开启了人们对深度聚类的深入研究。该方法提出使用自编码器学习数据的特征，使用提取后的特征进行计算得到软分配与对应的辅助目标分配，通过最小化软分配与辅助目标分配之间的 KL 散度训练网络得到最终的聚类结果。但 DEC 定义的损失在微调时可能会改变 AE 在预训练过程中学到的特征空间，从而导致非代表性的特征，对最终结果造成影响。针对该问题，Guo[3]等人对 DEC 进行改进以保护数据结构，为了保持数据生成分布的局部结构，采用了欠完备的自编码器，将聚类损失和自编码器的重建损失进行联合，优化聚类标签分配，并在保持局部结构的情况下学习适合聚类的特征。Li[4]等人对 DEC 模型进行改进使其应用于图像聚类，使用全卷积自编码器代替自编码器进行表示学习，然后提出了基于卷积自编码器的图像表示和聚类中心联合学习的统一聚类框架。

在深度聚类发展初始，由于传统聚类的良好表现，研究者试图将部分传统聚类方法与神经网络相结合来克服传统聚类算法的弊端以达到较好的聚类效果。Yang[5]等人将 k-means 与自编码器相结合提出了在学习 k-means 友好型表示的同时做聚类，联合优化重构损失与 k-means 损失，该方法实现简单，复杂度较低。Yang[6]等人提出一个递归框架将卷积自编码器与层次聚类相结合，核心思想是良好的表示有利于图像聚类，同时聚类结果反过来为表示学习提供监督信号。通过

将两个过程集成到一个模型中，使用加权三重态损失函数进行图像聚类与表示学习的联合更新，但其复杂度较高，消耗内存并难以优化。Ji[7]等人引入了一种新的自编码器结构以学习子空间聚类友好型非线性映射，将子空间聚类与神经网络相结合。主要思想为在编码器解码器之间加入了全连接的自表达层以模拟传统子空间聚类中的自表达特性，联合优化自表达损失与重构损失。Zhang[8]等人提出一种端到端的自监督子空间聚类网络，其中包括用于学习表示的卷积自编码器模块、用于子空间聚类的自表达模块与用于自监督的谱聚类模块，通过谱聚类模块的输出对表示学习模块与自表达模块实现双重自监督。Abavisani[9]等人在深度子空间聚类中使用了数据增强技术学习对于轻微变换的输入具有一致子空间的表示，基于此，该方法通过获得最高的平均轮廓系数，提供了一个简单有效的无监督程序找到有效的数据增强策略。

除与各种传统聚类算法结合外，研究人员提出许多新的深度聚类算法思路。Huang[10]等人通过使用有利于学习聚类友好型表示的两个约束训练自编码器，约束分别为：组稀疏性约束，作用在于学习非零群对应于簇的块对角表示；局部保持约束，旨在保持原始数据的局部结构特性。通过联合训练三个损失学到面向聚类的表示，之后采用 k-均值得到聚类结果。Chen[11]等人提出了基于深度学习的多流形聚类框架，同时优化重构损失与局部保持损失学习表示；其次，通过表示对簇中心的接近程度对表示进行惩罚，将两个目标集成到一个模型中获得簇类友好型表示，从而得到更精准的聚类结果。Shah[12]等人提出了深度连续聚类，使用了鲁棒连续聚类 RCC 具有明确连续目标且不需要设定簇个数的优势，将 RCC 与降维相结合做全局连续目标的优化，不需要预先设定集群数量，同时避免了以往算法中目标的离散重构。Hsu[13]等人提出一种可以处理大规模图像数据集的聚类算法，该方法使用 CNN 框架迭代的解决表示学习与聚类的问题，首先随机选取样本使用预训练好的模型提取特征作为簇中心，然后执行小批量 k-均值并更新参数，其中包括一种特征补偿方案，避免表示学习中两次连续迭代之间的特征不一致导致的误差。该方法是较早的可以处理百万级别图像聚类任务的深度聚类算法。

由于数据表示的质量对于后续的聚类有极大的影响，因此为了避免聚类只关注低级特征，而更好地从数据中获得具有代表性的高级信息，研究者选择在输入表示时加入噪声或者数据增强克服该问题。Dizaji[14]等人使用了加噪声的卷积自编

码器学习表示，利用带噪声数据提取的特征计算软分配，原始数据提取的特征计算辅助目标分配，重构损失中希望它们提取的特征尽可能相似以防止特征表示的损坏，并对以上两项使用相对熵损失函数，使用统一框架联合重构损失与相对熵损失。Hu[15]等人提出一种信息最大化自增强训练方法，最大化增强数据表示与原始数据表示的一致性以实现对数据表示施加预期不变性，与此同时学习概率分类器，获得将数据映射为离散表示的函数，最大化输入与预测的离散表示之间的互信息，使用数据增强对离散表示施加不变性训练网络。Guo[16]等人提出使用数据增强的深度嵌入聚类方法，将数据增强与 DEC 算法相结合提高泛化能力，该方法使用增强数据训练自编码器，同时通过最小化目标与编码器实际输出的 KL 散度实现聚类，其中目标使用非增强数据计算，模型输出使用增强数据。该方法与[14]较为相似，主体都是使用 DEC 的思路，同时施加增强数据或噪声数据对其进行相应的改进。

除此之外，有研究者考虑数据与数据间的成对信息，通过将成对信息融合进深度聚类中希望能从中获取数据间的相似信息，以达到更好的聚类效果。Sadeghi[17]等人提出了深度连续子空间聚类同时最小化数据点的加权重构损失与聚类损失，每次训练时都重点放在更可能属于相应集群的数据点上，即每次运行侧重于单独某一个簇，通过多次连续训练获得所有数据的最佳潜在空间并较好地形成聚类结果。Sadeghi[18]等人提出基于[17]的工作，捕捉成对数据关系中可用的有用信息。将[17]中工作作为基础，之后在网络中链接全连接层将数据映射到簇个数维数的空间内，以此作为数据的簇分配，通过[17]获取成对样本的相似度做监督信息对网络进行训练以得到更准确的聚类结果。Fogel[19]等人基于成对约束的深度嵌入算法，该方法脱离基于簇中心的方法，使用成对约束驱动聚类。使用相互最近邻分析提取成对约束，并将成对信息用作必须链接的约束，希望成对的数据点在潜在空间中输出相似的表示。

由于自编码器良好的重构性质可以与任何算法相结合，因此在深度聚类算法中最为常见、研究最多的是基于自编码器的算法。但是单纯使用自编码器提取聚类表示，学习到的特征没有足够的区分性，不能包含丰富的结构信息，会使算法的上限精度受到限制。因此此类深度聚类方法在较为复杂的数据集上表现一般。

1. 2. 2 基于生成式网络的深度聚类算法

随着生成式网络在许多领域的应用，引起研究者的广泛关注，已有工作将此类模型与深度聚类相结合，其中最常用的为变分自编码器(Variational Auto-Encoders，VAE)与生成对抗网络(Generative Adversarial Networks，GAN)。

（1）基于 VAE 的深度聚类算法

VAE 可以被视为 AE 的生成变体，它强制 AE 生成的潜在编码遵循预定义的分布，因此标准 AE 和 VAE 之间最显著的区别是 VAE 对潜在表示施加了概率先验分布。在 VAE 与聚类相结合的算法中，尽可能选择一个能够描述聚类结构的分布。提出的大多数算法都选择服从高斯分布，假设数据服从 k 个多元高斯分布组成的高斯混合分布，聚类的过程相当于推断数据点是从哪个分布生成的，认为从同一分布生成的数据属于同一簇，通过学习的高斯混合模型推断聚类分配[20]，但此类算法的复杂度较高。Jiang[21]等人提出的变分深度嵌入（Variational deep embedding,VaDE）主要运用了以上思想，其总体目标的第一项为重构损失，第二项为后验概率分布和先验分布的 KL 散度，最大化下界后可以直接从先验判断集群分配。Dilokthanakulet[22]等人使用与 VaDE 相近的思想，同样将最大化生成模型的对数似然函数转化为最大化证据下界，与此同时通过最小化信息约束使得模型在训练初期避免陷入局部解。Yang[23]等人将图嵌入与高斯混合模型相结合，获取数据结构的图形信息对高斯混合模型做补充，使网络能够学习到符合全局与局部结构约束的特征表示，该方法提出将样本视为图上的节点，最小化其后验分布之间的加权距离，并与对数似然最大化相结合同时优化表示学习与聚类。

（2）基于 GAN 的深度聚类算法

GAN 在生成网络与判别网络之间建立了一种对抗博弈，生成网络试图将样本从先验分布映射到数据空间，而判别网络试图计算输入是数据分布中真实样本而不是生成网络生成样本的概率。运用于聚类的主要思想为对变量的分布进行某种限定。通过将输入编码为连续特征和分类特征，以重构损失和对抗损失双重目标训练网络，分类特征即为聚类分配输出[24]，该类型的深度聚类算法相对来说难以收敛。Springenberg[25]等人提出分类生成对抗模型，将 GAN 的二分类识别器扩展到多个类，分类个数为簇的总个数，识别器的输出改为样本属于各个簇的概率识别器，对于生成器要求其生成属于某一个簇的样本而不是整个数据集。Chen[26]等

人提出最大化 GAN 噪声变量的固定子集和观测值之间的互信息，将识别器的输入噪声向量分解为两部分：不可压缩噪声和潜在编码，在对抗训练的同时最大化潜在编码与生成数据之间的互信息。当使用一个具有 k 个值的分类码和多个连续码对潜在编码进行建模时，它具有将数据点聚类为 k 个簇的功能。Mukherjee[27]等人通过从 one-hot 编码变量和连续潜在变量的混合中采样潜在变量，联合一种新的反向传播算法，将数据映射到潜在空间中与特定的聚类损失联合训练，在潜在空间实现聚类。

基于 VAE 与 GAN 的深度聚类算法能够从最终获得的聚类生成样本，具有很好的理论保证，但对于 VAE 和 GAN，由于难以优化，此类算法通常具有很高的计算复杂性。基于 GAN 的算法缺点与 GAN 相似，具有模式崩溃和收敛缓慢的问题。基于 GAN 的算法在一般 GAN 框架上施加了多类先验，它们比基于 VAE 的算法更灵活和多样化，其中一些旨在学习可解释的表示，并将聚类任务作为一个具体的案例。

1. 2. 3 基于孪生网络的深度聚类算法

由于在深度聚类中，孪生网络的应用大多数为与对比学习相结合，主要通过孪生网络学习原始样本或增强样本的表示，利用该表示进行对比学习，因此，该部分对基于对比学习的深度聚类算法进行介绍。对比学习通过学习数据的不同增强间的一致性从而学习无标签数据的一般特征，由于良好的学习数据特征能力被广泛应用。研究者们将其与深度聚类相结合希望通过对比学习学到较好的特征用于聚类。

Zhang[28]等人将对比学习与 DEC 相结合，采用经典的 SimCLR[29]对比学习学习特征表示，DEC 的损失作为聚类损失，联合聚类损失与对比损失训练网络。Gansbeke[30]等人提出一种两步骤方法，其中表示学习与聚类解耦。首先通过增强数据进行对比学习学到特征表示，基于特征相似度挖掘每个数据的近邻，之后将最近邻用作先验，训练模型将每个数据及其对应的邻居分为一类。Dang[31]等人在[30]的基础上在第二阶段进行改进，同时考虑局部与全局特征中存在的样本关系，分别从两个级别上寻找最近邻样本构建一致性损失与对比损失。Li[32]等人提出对比聚类，在 SimCLR 的思想下，利用对比学习促进聚类，该方法不单单将对比学

习作为上游任务，而是学习面向聚类的对比任务。将数据映射到簇个数维数的特征空间，在特征矩阵的列方向上采用对比学习，通过联合行方向与列方向的对比学习训练网络实现聚类。

Wang[33]等人考虑了实例间的相似性，实例间相似性不仅由相互吸引产生，还可以由对实例组的共同排斥产生。因此，该方法通过增强数据的对比学习去学习实例内的相似性，与此同时，通过实例与簇之间的对比学习去学习实例间的相似性，联合训练两类对比损失以完成聚类。Liu[34]等人提出使用图拉普拉斯滤波器对数据进行预处理，去除属性中的高频噪声；采用不同权重的网络对数据进行表示学习，基于不同的表示作为正例构造相应的对比学习，对于学到的良好表示采用 k−means 得到最终的聚类结果。Sharma[35]等人通过基于聚类的对比学习获得好的人脸特征，该方法直接先对数据进行简单的聚类操作，将结果中同一簇的数据作为正例，远离该簇的簇中数据作为负例进行对比学习在人脸中学到具有区分性的表示。

还有部分研究者将对比学习作为辅助解决聚类问题。Ma[36]等人针对现有方法通常忽略潜在区域中簇边界样本的问题，提出了局部归一化软对比聚类，利用了每个样本的局部数据之间的相似性邻域和全局不连通样本，以对比的方式利用样本对的积极性和消极性来区分不同簇。Lin[37]等人将对比学习用于多视图层次聚类，以对比方式跨多个视图进行表示，将表示嵌入到双曲空间中，并通过连续松弛分层聚类损失来优化双曲嵌入，从优化的双曲线嵌入中解码二叉聚类树。

在深度聚类与对比学习相结合的算法中，大多数方法往往只是简单将对比学习基本框架与深度聚类相结合。聚类要求对实例进行分组，而不是做实例区分。但对比学习基本框架只关注增强数据间的相似性，侧重于在实例层次上进行区分，忽略了实例间的相似性所构成的群组，没有考虑潜在的类别信息和聚类目标。因此，这样的特征学习是不稳定的，学习到的表示不是聚类的最佳表示，导致聚类性能受到限制。基于以上问题，论文提出了两种算法将对比学习与深度聚类更好的结合，提升深度聚类的性能。

1.3 研究内容与方法

对比学习强调两个增强样本之间的一致性，令同一数据的不同增强样本间的

相似度最大，与其他数据的相似度最小，从而学习数据的一般特征，在表示学习方面取得了较好的成果。深度聚类重点在于表示学习与聚类的联合优化，因此可以将对比学习与深度聚类统一到同一框架进行训练。但是在将两者结合时，大多数算法单纯使用对比学习做前期的表示学习，忽略了聚类任务的潜在信息，使用固定的损失函数，而不是专门面向聚类任务。针对以上问题，论文通过不同方式考虑数据间的关系，提出两种方法，超越单个样本信息，探索样本间关系，利用对比学习获得聚类友好型表示以提升深度聚类算法的效率。具体研究内容如下：

（1）基于自步学习（self-paced learning）的实例级别对比聚类算法

该算法引入了自步学习的思想，分为两阶段通过由简到难的方式对数据进行训练。在第一阶段得到的初始特征空间中将容易区分的数据进行初步聚类。第二阶段在训练的同时使难分的数据逐渐变得容易区分，最后聚类到合适的簇中。具体来说，第一阶段通过考虑数据与簇之间的关系，采用中心损失对神经网络进行训练，该过程中将易分数据进行了初步聚类，同时学习到数据的成对相似性为后续过程进行监督。为了避免学习到数据中的低级特征，在训练时使用了增强数据，最小化增强数据与原始数据之间的表示。第一阶段完成后，数据被映射到一个初始的潜在空间，在该空间内易分的数据分布在相应的簇中心附近。

在第二阶段对整体网络采用对比学习进行微调训练，对特征空间进行改进，其中根据第一阶段学到的数据间关系进行监督。根据数据间的成对相似性设置正负例，每个数据在指定阈值范围内的相似样本（不相似样本）构造正例对（负例对），进行实例级的对比学习，使得相似的成对样本尽可能聚集在一起，不相似的成对样本尽可能区分开来。在第二阶段的训练中，随着训练时间的增加，越来越多的难分样本逐渐变得容易区分，最后得到聚类结果。

（2）基于图结构的实例-簇级别对比聚类算法

该算法同时考虑数据与数据间、簇与簇之间的关系，利用该关系将对比学习与深度聚类结合。图结构被广泛应用于捕捉数据之间的潜在关系，因此选择图结构来体现数据间关系。论文采用深度子空间聚类网络基本思想，在自编码器中加入自表达层获取图结构，从中反应样本的邻域信息。与此同时，算法通过假设一个簇中的样本应该共享相似的特征表示和簇分配来体现簇与簇间的关系，将常用的实例级别提升到簇级别。具体来说，首先通过深度子空间聚类得到的自表达矩

阵构造相应的图结构，其中蕴含着潜在的类别信息。之后在表示学习方面，提出了基于图的实例级对比损失学习区分性的特征，其中的正负例根据图确定。按照标签表示的思想，当数据投影到维数等于聚类数的空间时，特征向量相应地表示其软标签。因此将特征矩阵的列作为数据的聚类预测，在列方向进行基于图的簇级别对比学习。从实例级别引申到簇级别，都融合了潜在的类别信息，从两方面进行训练以增加簇内样本的紧密度，减少簇间样本的紧密度，同时进行特征学习与集群分配。

1.4 主要工作和创新

（1）提出了基于自步学习的实例级别对比聚类算法。现有对比学习与深度聚类相结合的算法中，通常将对比学习独立于聚类学习表示。针对该问题，从聚类中数据与簇的关系出发，构造数据间成对相似性关系，通过融合聚类目标实现对比学习。同时引入自步学习的思想，通过由简到难的方式对数据进行训练，在第一阶段将容易区分的数据进行初步聚类。第二阶段在训练的同时使难分的数据逐渐变得容易区分，最后聚类到合适的簇中。

（2）提出了基于图结构的实例-簇级别对比聚类算法。为了避免对比学习与深度聚类相结合时过分依赖数据增强，对每个数据都进行区分，同时对原始数据利用不充分。本文通过考虑数据与数据、簇与簇间关系，融合类别信息学习聚类友好型表示。基于图结构从实例级别与簇级别同时进行对比学习，在双重对比学习的框架下达到聚类的目的。

1.5 论文的基本结构

论文分为五个章节进行论述，具体组织结构如下：

第一章为绪论。该章节首先介绍了论文的研究背景及意义。之后总结了主要的国内外研究论文，分别从基于自编码器、基于生成式网络、基于孪生网络三个方面介绍了深度聚类在国内外的研究现状，并对其进行了分析讨论。最后给出了本论文的主要研究内容与方法。

第二章为基础理论与预备知识。该章节主要介绍了论文涉及到的一些基础概

念以及后续使用到的理论知识，主要包括对比学习，深度子空间聚类以及自编码器等相关知识。

第三章为基于自步学习的实例级别对比聚类研究。该章对提出的算法进行详细介绍，描述了网络模型以及具体过程，利用成对相似性捕捉样本之间的关系，实现对比学习。在多个数据集上进行实验，与已提出的深度聚类方法进行比较，并进行论证分析，证明了其有效性。

第四章为基于图结构的实例-簇级别对比聚类研究。该章对基于图结构的算法进行详细介绍，描述了该方法的网络模型以及具体过程。利用图中关系构建对比学习，并拓展到簇级别进行双重对比聚类。在多个数据集上进行实验，与已提出的深度聚类方法进行比较，并进行论证分析，证明了其有效性。

第 2 章 基础理论与预备知识

本章主要介绍论文涉及到的一些基础概念以及后续使用到的理论知识，包括深度聚类相关概念、自编码器、深度子空间聚类以及对比学习的相关知识。

2. 1 深度聚类相关概念

聚类按照某个特定标准将数据集划分为不同的集群，将相似的数据划分到一起，使同一集群内数据相似度尽可能高，不同集群中数据的相似度尽可能低。

表示学习为了提高机器学习的准确率，将输入信息转换为有效的特征或者更一般性的表示。基本思路是从原始数据中学习出有效的特征，提高机器学习模型的最终性能，即找到对于原始数据更好地表达，以方便后续任务。

深度聚类是使用深度神经网络同时学习特征表示和聚类分配的一系列方法。对表示学习与聚类进行联合优化，通过训练将数据转换为更有利于聚类表示的非线性映射。

2. 2 自编码器

自编码器[38]（Auto-Encoder，AE）是广泛使用的无监督表示学习方法之一。其基本思想是利用一层或多层神经网络对输入数据进行映射，映射后的向量是从数据中提取的重要特征，用于后续学习。为保证映射向量包含更为本质的特征，AE 通过该向量对输入进行重构，希望重构后的输出与输入尽可能相等。

基本自编码器模型是一个三层神经网络结构，分别为一个输入层，一个隐藏层与一个输出层，输入层与输出层的维数相同，令输入输出尽可能一致。自编码器主要由两大部分组成：编码器$f_{\varphi}(x)$与解码器$g_{\theta}(y)$，网络结构如图 2.1 所示，编码器将输入x通过网络转换为潜在表示y, 公式如下：

$$y = f_{\varphi}(x) \tag{2.1}$$

解码器将y通过网络进行重构，如公式 2.2 所示，尽可能得到原始输入x，以确保转换后的潜在表示y中包含有代表性的信息。

$$x' = g_\theta(y) = g_\theta\left(f_\varphi(x)\right) \tag{2.2}$$

使用编码器得到的特征向量对输入进行重构，其基本原理是通过最小化重构误差对网络进行训练，该网络损失函数如公式 2.3 所示：

$$L = ||x - x'||_2^2 = ||x - g_\theta(y)||_2^2 = ||x - g_\theta\left(f_\varphi(x)\right)||_2^2 \tag{2.3}$$

其中，φ是编码器的网络参数，θ是解码器的网络参数。

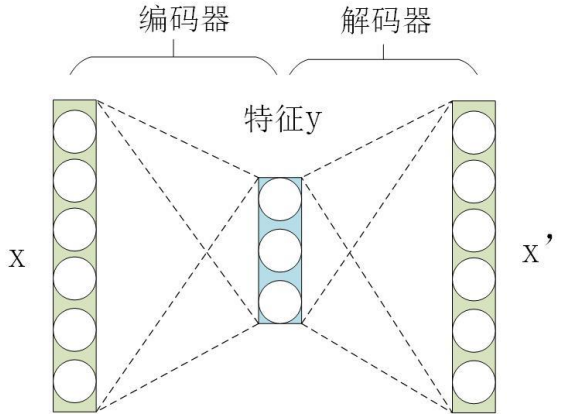


图 2.1 自编码器网络结构图

AE 一旦训练完成后，解码器部分将不再使用，此时编码器可以得到较好的特征表示，将编码器用于对数据进行映射。由于自编码器原理易于理解，操作简单并且可与大部分算法相结合，因此在深度聚类算法中被广泛使用。其中，出现了许多自编码器的变体以提高自编码器的性能：

（1）堆叠自编码器[38]：该编码器为最常见的自编码器之一，本质上是增加编码器与解码器的层数，使用更深层的网络结构，有利于学习到更优的表示。通常使用的自编码器都是堆叠自编码器。

（2）卷积自编码器[38]：原始自编码器的编码层与解码层都是简单的全连接层，可以将其进行变化，例如变换为卷积层、池化层等，构建成为卷积自编码器，有利于对图像数据进行特征提取。

（3）欠完备自编码器[38]：自编码器中编码器提取出的特征向量的维数可以任意选择。欠完备自编码器是对自编码器进行潜在的特征约束，使得编码器提取出的特征维数必须低于输入数据的维数，强制编码器学习原始空间中最显著的特征。欠完备自编码器主要用来进行数据降维，与常见的主成分分析、线性判别分析等降维算法相比，可以进行非线性降维，在复杂数据上的特征提取能力较好，

是利用神经网络进行降维的重要方法之一。因此，通常使用的自编码器默认均为欠完备自编码器。

（4）去噪自编码器[38]：为了避免过拟合并且提高网络的鲁棒性，可以向原始数据中添加适当噪声，使得输入数据是被某种形式的噪音破坏的样本$\tilde{x}$。此时，解码器必须从受到损坏的数据得到的特征向量中恢复原始数据，可以使编码器捕获数据更加显著的特征。其损失函数如下：

$$L = ||x - g_\theta\left(f_\varphi(\tilde{x})\right)||_2^2 \tag{2.4}$$

除此之外，也可对自编码器施以其他约束，例如稀疏自编码器通过增加稀疏性约束获得稀疏表示。

2. 3 深度子空间聚类

在图像处理等现实应用中，通常需要处理大量的高维数据，给数据分析与处理带来了一定的困难。复杂的高维数据难以用单一的子空间表示,一种较为合理的假设是认为高维数据分布于多个低维子空间的并集上，从而产生了子空间聚类算法，子空间聚类的目的是将数据划分到本质上所属的低维子空间中[39]。对于给定的一组数据，假设这组数据属于多个不同的线性子空间，子空间聚类是指将这组数据划分为多个类别，在理想情况下，每一类别对应一个子空间。

如今，已经提出了许多子空间聚类算法。这些方法大多可以归结为两个步骤：第一步，估计两两数据对之间的亲和度，以形成亲和度矩阵；第二步，使用亲和度矩阵进行归一化切割或谱聚类得到聚类结果。子空间聚类算法基本被分为三类：基于因式分解、基于高阶模型以及基于自表达的方法[7]。其中使用最为广泛的是基于自表达的方法，该方法使用同属于一个子空间中其他样本的线性组合来表示样本，并利用其中的系数矩阵来构建亲和度矩阵，其中起重要作用的为自表达特性。自表达特性指对于不同子空间中的N个数据$X = \{x_1, x_2, \dots, x_N\}$，可以将数据点表示为同一子空间中所有其他点的线性组合，即$x_j = \sum_{i \neq j} c_{ij} x_i$，若将所有的点$X$表示在同一矩阵中，那么自表达性可表示为：$X = XC$，其中$C$是自表达系数矩阵。已有学者证明，在子空间是独立的假设下，通过最小化C的某些范数，可以使C具有块对角结构，当点x_i和点x_j位于同一子空间时，$c_{ij} \neq 0$。通常利用矩阵C来构建谱聚类中的亲和度矩阵，优化问题表示为：

$$min\,||C||_p \quad s.t.\ X = XC, (diag(C) = 0) \tag{2.5}$$

公式(2.5)中的条件约束通常被放宽为正则化，如公式(2.6)所示：

$$min\,||C||_p + \frac{\lambda}{2}\,||X - XC||_p^2 \quad s.t.\ (diag(C) = 0) \tag{2.6}$$

其中$||\,.\,||_p$表示 p 范数。例如，稀疏子空间聚类[39]中使用 L1 范数来获得稀疏线性组合；低秩子空间聚类[40]中使用核范数来寻找低秩表示；低秩稀疏子空间聚类[41]使用 L1 范数和 L2 范数的混合或核范数来平衡线性组合系数的稀疏性和密集性；最小二乘回归(LSR)[42]和高效密集子空间聚类[43]中使用了 Frobenius 范数等。

但在实践中，数据通常位于非线性空间的情况较多，因此出现了结合神经网络和子空间聚类的算法，对表示与聚类进行共同优化，即深度子空间聚类。该方法主要利用了自编码器和自表达特性，将数据通过网络得到具有良好表示的特征向量，能够聚类具有复杂结构的数据点。深度子空间聚类主要思想是在编码器和解码器间加入全连接层作为自表达层，模拟传统子空间聚类中的自表达特性，新的自表达层提供了一种简单但有效的方法，通过反向传播学习自表达矩阵，从而获得所有数据点之间的亲和度。

具体来说，引入了新的损失函数公式如下：

$$L(\theta, C) = \frac{1}{2}||X - X'_\theta||_F^2 + \lambda_1\,||C||_p + \frac{\lambda_2}{2}\,||Z_{\theta_e} - Z_{\theta_e}C||_F^2 \quad s.t.\ (diag(C) = 0) \tag{2.7}$$

其中，θ表示自编码器的参数，θ_e为编码器的参数，Z_{θ_e}为编码器的输出，即提取的特征向量，X'_θ为重构后的数据。

深度子空间聚类将C认为是一个额外的网络层参数，使得可以利用反向传播对θ与C进行联合求解。自表达性中每个数据点可由一些点的加权线性组合得到，这样的线性表示正好对应于神经网络中一组没有非线性激活的线性神经元。因此，深度子空间聚类将每个数据点作为网络中的一个节点，使用一个完全连通的线性层来表示自表达层，自表达层的权值即为系数矩阵C，通过网络直接进行学习亲和度矩阵。将矩阵C也表示为网络参数θ_s，最终的损失函数为：

$$\tilde{L}(\tilde{\theta}) = \frac{1}{2}||X - X'_{\tilde{\theta}}||_F^2 + \lambda_1\,||\theta_s||_p + \frac{\lambda_2}{2}\,||Z_{\theta_e} - Z_{\theta_e}\theta_s||_F^2 \quad s.t.\ (diag(\theta_s) = 0) \tag{2.8}$$

其中，$Z_{\theta_e}\theta_s$表示编码器输出经过自表达层转换后的潜在表示，$\tilde{\theta}$表示网络中所有参数，共包括三部分：编码器参数、解码器参数与自表达层参数。

网络结构如图 2.2 所示。从网络结构可以看出，深度子空间聚类联合自编码器与自表达层，将自表达特性融入神经网络中。对网络进行训练后，可获得较好的矩阵C，利用C来构造亲和度矩阵，通常计算为$|C| + |C^T|$，获得亲和度矩阵后，就可以使用谱聚类算法对数据进行聚类，从而得到最终的聚类结果[39]。

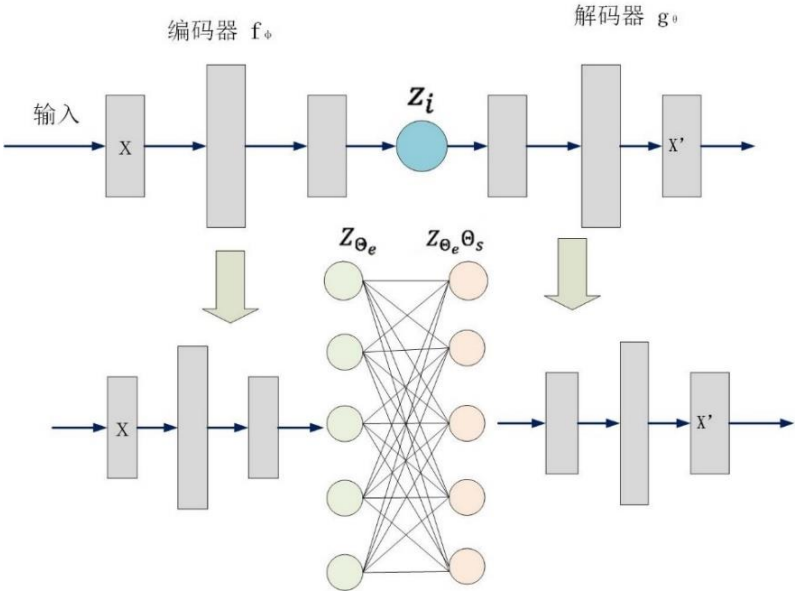


图 2.2 深度子空间聚类网络结构图

2. 4 对比学习

目前许多机器学习方法均为监督学习，需要使用大量有标签的数据进行训练。有监督学习通过标签训练得到的模型通常只能学到一些针对特定任务的知识，而不能学习到一种通用的知识，因此有监督学习学到的特征表示难以迁移到其他任务。而自监督学习能够很好地避免上面的问题，自监督学习属于无监督学习的一种，特点是无需对数据进行标注，可以从无标签数据中挖掘自身的监督信息，自动为数据产生标签，通过构造的标签信息学习数据的表示。自监督学习分为两种类型：生成式自监督学习与判别式自监督学习。其中 VAE 与 GAN 是典型的生成式自监督学习，此类方法难度与复杂度都较高；而对比学习是典型的判别式自监督学习。

对比学习在最近的表示学习中取得了非常好的性能，该方法用于在没有标签的情况下学习数据的一般特征。对比学习主要思想为将数据与语义相似的例子（正例）和语义不相似的例子（负例）进行对比，通过设计模型结构和对比损失，

使得在特征空间中相似的实例更接近，不相似的实例分布较远，从而学习数据的一般特征[29]。

针对正例负例的设置，最为常用的方法是将每个实例作为一个以特征向量表示的类，通过数据增强构造数据对。具体来说，同一个实例的两个不同增强视图构成正例对，每一实例与其余所有实例以及它们的增强数据构成负例对。如何定义正例对与负例对，构造能够遵循其基本原理的表示学习模型结构，是对比学习的关键问题。

SimCLR[29]是对比学习中最为典型的方法之一，使用增强数据学习数据的特征，体现了对比学习的基本框架。通过介绍该方法对对比学习框架进行说明，该方法效果较好并且结构简单清晰，易于理解，是基于对比学习的深度聚类算法中最常用的模型。

SimCLR 主要分为四部分，其流程图如图 2.3 所示。

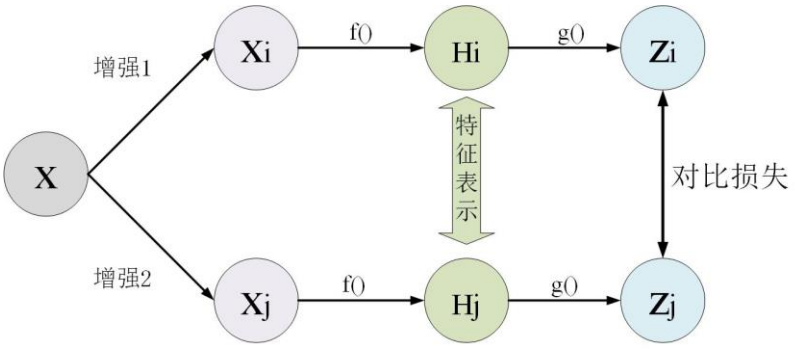


图 2.3 SimCLR 基本流程框架图

（1）构造正负例对。给定含有N个数据的数据集$X=\{x_1,x_2,\dots,x_N\}$，对于每个样本，从增强操作中任意选取两种增强方式，将原始数据进行不同的增强得到$2N$个数据，$\{x_{1i},x_{2i},\dots,x_{Ni},x_{1j},x_{2j},\dots,x_{Nj}\}$。对于其中的数据$x_{1i}$，可构成一个正例对$(x_{1i},x_{1j})$，$2N-2$ 个负例对$\{(x_{1i},x_{2i}),\dots,(x_{1i},x_{Ni}),(x_{1i},x_{2j}),\dots,(x_{1i},x_{Nj})\}$。通常使用的增强操作有图片的随机裁剪、获取灰度图、高斯模糊、旋转、剪裁、遮盖、加噪声等。

（2）特征提取编码器。得到正例与负例后，需要构造合适的网络架构，通过它将数据投影到某个表示空间内，并采取一定的方法，使得正例对之间的距离比较近，负例对之间的距离比较远。SimCLR 首先采用一个特征提取编码器$f(.)$，选取的是 ResNet，经过网络将所有数据及其增强转换为特征表示$H=\{h_{1i},h_{2i},\dots,h_{Ni},h_{1j},h_{2j},\dots,h_{Nj}\}$。

（3）非线性变换结构。数据在通过特征提取之后，经过一次非线性变换结构，该结构由 MLP 构成，进一步将特征表示映射成另一个空间里的向量$Z = \{z_{1i}, z_{2i}, \dots, z_{Ni}, z_{1j}, z_{2j}, \dots, z_{Nj}\}$。这样，经过$g(f(x))$两次非线性变换，将样本及其增强样本投影到新的特征空间，在该空间上进行对比学习。SimCLR 使用的具体网络结构为：全连接层−BN 层−ReLU 层−全连接层。通过实验证明，添加 MLP 层可以明显提高实验效果，因此，之后的对比学习模型中大多会在网络后加 MLP 层以提升性能。

（4）对比损失函数。损失函数的设置需要使表示空间内正例对距离更近，负例对的距离更远。该损失函数一般使用 infoNCE 损失，对于正例对(x_i, x_j)的损失函数定义公式如下：

$$l_{i,j} = -log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(z_i, z_k)/\tau)} \tag{2.9}$$

其中，$sim(z_i, z_j)$为余弦相似度函数，计算z_i与z_j的相似性；$\mathbb{1}_{[k \neq i]}$为 0 或 1，当$k \neq i$时，结果为 1，相等时，结果为 0；τ是温度系数。从损失函数可以看出，该函数的分子部分为正例对的相似性，分母中包含所有的负例对。在训练过程中，希望分子越大越好，即正例对之间的相似性越来越高，特征空间中正例对间距离越来越近；分母越小越好，即负例对之间的相似性越来越低，特征空间中负例对间距离越来越远。

SimCLR 使用的是孪生网络，两支的网络都为步骤 2 与步骤 3 的结合，在获取增强数据后，两个分支表示两种不同增强方法后得到的数据，各自经过网络得到特征表示，之后两者通过对比损失函数进行训练。以上为 SimCLR 的大致流程，也可以看作是对比学习的基本框架。

2.5 本章小结

本章对论文涉及到的一些概念及知识进行了简单的介绍。首先对深度聚类方面的相关概念进行了简单的描述；之后对深度聚类中最为常见的自编码器模型进行了介绍，引申出几个对自编码器模型改进后的模型；并对论文后续用到的深度子空间聚类以及对比学习进行了介绍，奠定了后续工作的基础。

第 3 章 基于自步学习的实例级别对比聚类研究

现有对比学习与深度聚类相结合的算法中，通常使用对比学习做前期的表示学习，将增强数据作为正例，独立于聚类学习表示。针对该问题，本章提出了一种基于自步学习的对比聚类算法，其遵循认识事物的过程,按照从简单到复杂的方式进行学习。同时算法通过考虑数据与簇间的关系学习数据间成对相似性，面向聚类目标构造正负例进行对比学习，提高深度聚类算法的性能。将所提算法进行了多项实验，从各方面验证了其有效性。

3.1 算法思想与框架概述

对比学习的基本框架将每个实例进行区分，强调数据的不同增强视图之间的相似性，通过在特征空间内使两者尽可能接近，与其他样本都尽可能远来学习表示。而聚类是将实例进行分组,为了更好的将对比学习与深度聚类进行有效结合,获得包含更多聚类信息的数据表示，提升聚类效果。本章提出基于自步学习的实例级别对比聚类算法，通过考虑数据与簇间的关系得到成对相似性，使用成对相似性作为监督信息构造正负例实现对比聚类。

该模型引入自步学习[44]的思想，在每一步迭代中决定下一步学习样本，通过由简到难的方式对数据进行训练。算法将简单样本定义为对最小化损失函数贡献大的样本，复杂样本定义为对最小化损失函数贡献较小的样本，即简单样本为容易区分或有较大概率属于某一簇的样本，复杂样本为处于边缘区域不易分的样本。在两阶段的训练中，第一阶段学习成对相似性关系的同时，得到的初始特征空间中已经将简单样本进行了初步聚类，使其分布在相应的簇中心周围。基于初始潜在空间，在第二阶段使用对比学习训练的同时将复杂样本逐渐变得容易区分，最后聚类到合适的簇中，使得最终每个簇中数据更加紧凑，不同簇之间更加分离。

具体来说，在第一阶段，通过增强数据与中心损失预训练主网络，得到初始的潜在特征空间，使其能够学到有意义的特征表示，得到数据之间有效的成对相似性信息。在第二阶段，利用成对相似性构造对应的正负例，将相似与不相似的成对数据作为监督信息实现实例级别的对比聚类，以此对特征空间进行改进。通过训练使得相似数据聚集在一起，不相似的数据相互分离，得到聚类结果。

整体网络结构如图 3.1 所示。

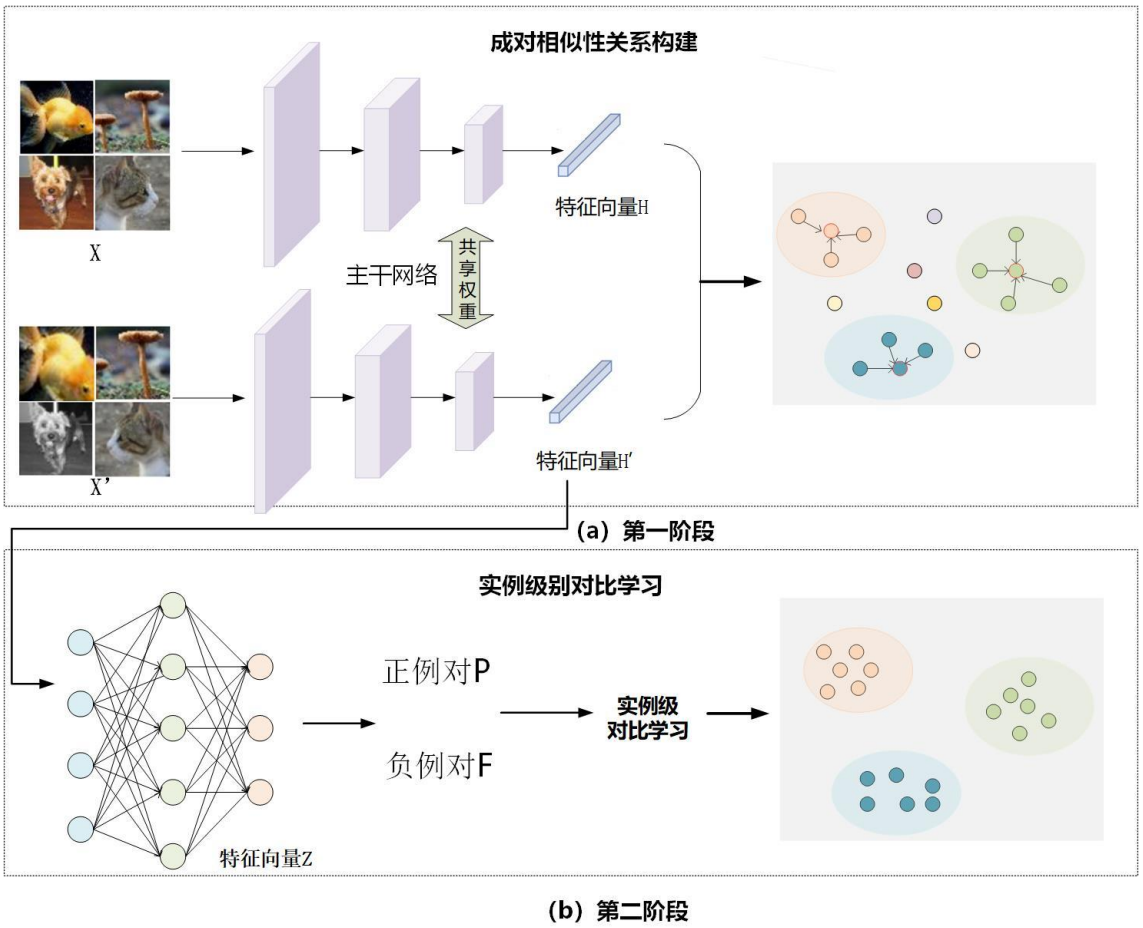


图 3.1 基于成对相似性的实例级别对比聚类算法网络框架

3. 2 基于自步学习的实例级别对比聚类算法

3. 2. 1 成对相似性关系构建

本章通过关注数据与簇之间的关系，获得数据与数据之间的内在联系，认为相似度高的数据有更大的概率位于同一个簇中，而相似度低的数据应该分布在不同的簇中。通过考虑数据点间的成对相似性关系为后续对比聚类提供有效的监督信息。因此，如何获得较好的成对相似性关系是重要问题之一，极大的影响着最终的结果。

成对相似性关系的构建为算法的第一阶段，该阶段相当于自步学习中对简单样本的学习，简单样本即软分配较高的样本，在不断的训练过程中对这些样本进行初始聚类，之后再逐步扩展到软分配相对较低的样本。位于同一簇中心周围的

数据对相似性较高，位于不同簇中心周围的数据相似性较低，以此得到数据间的成对相似性关系。

该阶段主要联合中心损失、原始数据X与增强数据X'间的损失对网络进行训练，得到特征表示以构建成对相似性关系。具体来说，对于每一个数据，首先使用数据增强T得到对应的数据$X' = T(X)$。之后将X与X'通过共享的深度神经网络$f_\theta(.)$得到特征向量H，利用损失函数对网络进行训练。成对相似性关系构建的整体框架结构如图 3.1(a)所示。

（1）构造增强损失

在对网络进行训练时，网络很容易学习到一些低级特征，例如颜色、纹理等，这样的特征可能会导致产生较差的结果。因此，为了尽量减少低级特征对结果的影响，算法引入了数据增强，最小化数据X与其对应增强数据X'得到的特征之间的相似度，以此进行训练以学到重要特征。具体损失函数公式如 3.1 所示：

$$L' = s\left(f_\theta(X), f_\theta\left(T(X)\right)\right) \tag{3.1}$$

其中相似度采用余弦相似度进行测量。

选择合适的数据增强方法对特征表示较为重要，算法按照 SimCLR 中使用的规则进行数据增强，其中包括五种增强方法：裁剪，将图像进行部分随机裁剪，并调整为原始大小；将图像进行水平翻转；将图像转换为灰度图；修改图像的亮度、对比度等；通过高斯函数模糊图像。对于给定的图像，每个增强方式以一定的概率独立使用[29]。

（2）构造中心损失

由于得到的成对相似性关系用于监督后续的对比聚类，因此该潜在空间中得到特征向量的数据分布应该接近于聚类期望的数据分布，即位于同一簇中心周围的样本在成对相似性关系中尽可能是相似的，位于不同簇中心周围的样本是不相似的。算法采用加权的中心损失进行训练，中心损失公式如下所示：

$$L'' = \sum_{k=1}^{K} \sum_{x_i \in X} p_{ik} ||f_\theta(x_i) - \mu_k||_2^2 \tag{3.2}$$

其中，K为簇个数，p_{ik}为权重，表示样本x_i属于簇cl_k的概率，该概率采用 t 分布计算，具体公式如(3.3)所示；$f_\theta(x_i)$为原始样本x_i经过网络后得到的特征向量；μ_k为簇cl_k的簇中心，计算公式如(3.4)所示。

$$p_{ik}=\frac{(1+||f_\theta(x_i)-\mu_k||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j=1}^{K}(1+||f_\theta(x_i)-\mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{3.3}$$

其中，α是 t-分布的自由度，通常取$\alpha=1$。

$$\mu_k=\frac{\sum_{x'_i\in X'}p_{ik}f_\theta(x'_i)}{\sum_{x'_i\in X'}p_{ik}} \tag{3.4}$$

计算簇中心时使用增强后的数据X'，希望原始数据获得的特征向量与增强数据所得到的簇中心尽可能接近，在使用增强数据的基础上使得模型进一步学习数据的不变特征，增加模型的鲁棒性。

在计算中心损失时，将软分配作为样本权值，可以看出，在每一次训练时，神经网络专注于对那些更可能属于每一簇的样本进行聚集。更具体地说，样本越接近簇中心，训练时有更大的影响，该样本在运行时对最小化损失函数的贡献就越大。在运行中，为了激励神经网络集中于易分的样本，将较高的权重分配给靠近簇中心的一系列样本。

（3）联合优化

该阶段总体损失函数如下所示：

$$L_1=\lambda L'+L'' \tag{3.5}$$

其中，λ为超参数，是两个损失函数的平衡参数。通过该损失函数对网络进行训练，在训练结束后，在潜在空间中易分数据应较为紧凑的围绕在它们相应的簇中心周围，通过靠近对应的簇中心，鼓励同一聚类的样本形成一个簇。此时得到了初始的特征向量，利用该特征向量获取数据间的成对相似度关系，用于指导第二阶段的对比学习，成对相似性关系计算公式如下：

$$Sim=H*H^T \tag{3.6}$$

第一阶段具体算法步骤如算法 3.1 所示。

算法 3.1 构建成对相似性关系矩阵

输入：原始数据集X，训练次数M_1，数据增强T，簇个数K；

输出：特征向量H，成对相似性关系矩阵Sim；

1. 初始化簇中心μ_k；
2. $X'=T(X)$；
3. for iter = 1 to M_1:
4. $H=f_\theta(X)$，$H'=f_\theta(X')$；

5. 使用公式(3.1)计算L′；
6. 使用公式(3.3)计算软分配 p_{ik}；
7. 使用公式(3.2)计算中心损失L″；
8. 通过公式(3.5)计算损失函数L_1来更新网络参数；
9. 使用公式(3.4)更新簇中心；
10. 使用公式(3.6)得到矩阵Sim；
11. End

3. 2. 2 实例级别对比学习

该部分为算法的第二阶段，可看做自步学习中对复杂样本的学习，在第一阶段形成簇中心周围较为紧凑的特征空间后，将处于边缘部分的难分数据通过网络的训练逐渐归于对应的簇中，形成最终的聚类结果。

对比学习的基本思想是最大化正例对的相似性，最小化负例对的相似性，使得在特征空间中相似的数据尽可能接近，不相似的数据尽可能远离，其中正例对、负例对的构建较为重要。算法根据第一阶段得出的数据间成对相似性关系构造正负例对，将对比学习与深度聚类相结合，将潜在的类别信息纳入聚类，得到适合于聚类任务的对比聚类。该阶段网络结构如图 3.2(b)所示。

（1）获取特征向量

算法在得到特征向量H，即初始潜在空间后，堆叠了非线性 MLP 与K维的 softmax 层, 记为$g_\eta()$，将特征映射到新的特征空间进行实例级别对比聚类。具体来说，此时 MLP 的输入为第一阶段训练后的特征向量，输出为改进后特征空间的特征向量Z，利用该向量进行对比学习，进行对比学习时采用第一阶段学到的成对相似性关系矩阵Sim做监督。

由于附加了一个K维的 softmax 层，使其每一维对应于一个簇，利用该层来获取簇分配的概率值，输出的结果可以直接表示为数据属于每个簇的概率，因此，最终的聚类结果直接由训练好后的网络输出得到。

（2）构造对比损失

由于该阶段的对比损失存在监督信息，将成对相似性关系中相似度较高的作为正例对，相似度较低的作为负例对。因此，对于每个数据来说，其正负例对数

量是不确定的，而 SimCLR 使用的对比损失函数正负对数量是固定的，常规的对比损失不再适用。算法采用有监督信息的对比损失函数，样本x_i的损失函数如下：

$$L^i = \frac{-1}{|P(i)|}\sum_{p\in P(i)} log\frac{\exp\left(s(z_i,z_p)/\tau\right)}{\sum_{f\in F(i)}\exp\left(s(z_i,z_f)/\tau\right)} \tag{3.7}$$

其中，τ是实例级温度参数。$P(i)$与$F(i)$由成对相似性关系得到，分别为数据x_i的正例集合、负例集合，$P(i)=\{p|Sim_{ip}\geq\xi,1\leq p\leq N\}$；$F(i)=\{f|Sim_{if}\leq\gamma,1\leq f\leq N\}$。$S(.)$为相似度计算，采用余弦距离测量，公式为：

$$s(z_i,z_j)=\frac{z_i{z_j}^T}{||z_i||*||z_j||} \tag{3.8}$$

该阶段总体损失函数为：

$$L_2=\frac{1}{N}\sum_{i\in N}L^i \tag{3.9}$$

从损失函数可以推断，只有相似度较高和相似度较低的样本在第二阶段训练过程中有贡献。具体来说，成对样本的相似度大于 ζ 或小于 γ，则它们对训练有贡献；若相似度在 ζ 和 γ 之间，即在模糊区域中，对当前训练时期没有贡献。因此，在对比聚类期间，最小化L_2会增强相似样本的相似性，使其越来越紧凑，削弱不相似样本之间的相似性，使其距离越来越远。随着网络训练的进行，越来越多的数据对会参与到训练过程，在训练结束时，只有非常少的数据留在模糊区域。希望最终向量尽可能接近 one-hot 向量，其中向量的最大元素的索引表示该数据的簇标签。在训练期间，主干网络的参数会通过反向传播进行微调。

该阶段伪代码如算法 3.2 所示。

算法 3.2 实例级对比学习

输入：特征向量H，成对相似度矩阵Sim，网络g_η，训练次数M_2；

输出：网络参数η；特征向量Z；

1. for iter = 1 to M_2:
2. 计算：$Z=g_\eta(H)$;
3. 根据矩阵Sim得到正负对集合：$P(i)$、$F(i)$;
4. 使用公式(3.7)-(3.9)计算实例级对比损失L_2;
5. 更新网络参数以最小化L_2；
6. End

3. 2. 3 算法整体流程

给定一个聚类问题，对于包含N个数据的数据集$X = \{x_1, x_2, \dots, x_N\}$,将其分为$K$个不相交的簇，$K$为预先定义的参数。整体框架中主要包含两部分网络，分别为第一阶段的主干网络$f_\theta(.)$与第二阶段的 MLP+softmax 层$g_\eta(.)$，θ与η分别为网络的参数。为了在训练中提取到较好的决定性特征作为后续的监督信息，首先对数据集X采用特定的数据增强方式T，得到增强数据集$X' = \{x'_1, x'_2, \dots, x'_N\}$，此时共有 $2N$个数据。第一阶段主要将 $2N$个数据经过主干网络得到初始潜在空间中数据的特征向量$H = \{h_1, h_2, \dots, h_N | h_i = f_\theta(x_i)\}$，$H' = \{h'_1, h'_2, \dots, h'_N | h'_i = f_\theta(x'_i)\}$。通过损失函数进行训练，使得在初始潜在空间中形成有利于后续聚类的分布，有较大概率属于某个簇的一系列数据都位于簇中心周围，得到有效的成对相似性来体现数据间的关系，其中使用μ_k来表示第k个簇的簇中心。

第二阶段中，对于训练后得到的原始数据的特征向量H，将其经过网络$g_\eta(.)$得到最终的特征向量$Z = \{z_1, z_2, \dots, z_N | z_i = g_\eta(h_i)\}$，$g_\eta(.)$将$H$映射为$K$维向量，$z_i^k$表示为数据$x_i$属于第$k$个簇的概率。基于第一阶段得到的成对相似性关系构建对比损失对特征空间进行改进，使得相似数据聚集在一起，不相似的数据相互分离，最终得到聚类结果$Cl = \{cl_1, cl_2, \dots, cl_K\}$。

整体算法如算法 3.3 所示。

算法 3.3 基于自步学习的实例级别对比聚类算法

输入：数据集X，网络f_θ、g_η，训练次数M_1、M_2，簇个数K；

输出：簇分配；

1. 使用算法 1 得到θ、H以及Sim；　　//第一阶段
2. H作为输入使用算法 2 得到参数η以及特征向量Z；　　//第二阶段

//最终聚类分配结果：

3. for x in X:
4. 　　通过$h = f_\theta(x)$提取特征；
5. 　　计算$z = g_\eta(h)$，$cl = argmax(z)$得到簇分配
6. End

3.3 实验及其分析

本节内容通过实验验证所提算法的有效性。首先对实验所用数据集、对比算法以及实现细节进行简单介绍。之后进行多项实验，包括与多个算法结果的对比实验，消融实验以及部分参数实验，并对实验结果进行了分析。

3.3.1 数据集及比较算法

（1）实验数据集

论文在四个广泛使用的数据集上验证了所提出方法的有效性。考虑到聚类任务的无监督性质，在使用时连接训练集和测试集。组合训练数据集和测试数据集是聚类研究领域的常见做法。使用的数据集包括：MNIST、CIFAR-10、CIFAR-100、STL-10，具体信息如表 3.1 所示。

表 3.1 数据集详细信息表

数据集	数量	簇个数	大小
MNIST	70000	10	28*28
CIFAR-10	60000	10	32*32*3
CIFAR-100	60000	20	32*32*3
STL-10	13000	10	96*96*3

（2）对比算法

论文将所提算法与多种聚类算法进行了比较，包括传统的和先进的基于深度学习的聚类方法。传统的聚类方法有 k 均值[45]、谱聚类（SC）[46]、凝聚聚类[47]和局部保持非负矩阵分解（LPMF）[48]。对于基于表示的聚类方法，如[2]中所述，使用一些无监督学习方法，包括 AE[2]、SAE[49]、DAE[50]、DeCNN[51]、SWWAE[52]、AEVB[53]和 GAN[25]，以学习图像的特征表示，并使用 K-means 对图像进行聚类作为后续处理；其它的深度聚类算法有深度嵌入聚类（DEC）[2]、深度聚类网络（DCN）[5]、DKM[54]。

3.3.2 评价指标

对于不同算法的实验结果，使用两个标准度量来评估聚类性能，包括聚类精度（ACC）[55]和归一化互信息（NMI）[56]。ACC 找到了真实和预测聚类标签之间

的最佳映射。NMI 发现同一数据点的两个不同标签之间相似性的标准化度量。计算公式分别如下：

$$ACC = max_m \frac{\sum_{i=1}^{N} 1\{l_i = map(cl_i)\}}{N} \tag{3.9}$$

$$NMI = \frac{I(l;cl)}{\max(H(l),H(cl))} \tag{3.10}$$

其中l_i和cl_i分别表示数据点x_i的真实标签和预测标签。map(.)表示数据点的预测标签和真实标签之间的最佳映射。$I = (l; cl)$表示所有数据点的真实标签$l = \{l_1, l_2, \dots, l_N\}$和预测聚类分配$Cl = \{cl_1, cl_2, \dots, cl_N\}$之间的互信息。H(.)表示熵函数。ACC 和 NMI 范围均在区间[0, 1]内，得分越高表示聚类性能越高。

3. 3. 3 网络结构及参数设置

本章采用 python3.8 来完成所有实验，使用的网络框架为 pytorch。对于不同数据集，本章选择采用不同的主干网络进行训练。对于 MNIST 灰度数据集，使用卷积自编码器，自编码器首先需要进行预训练，之后按算法过程进行训练。对于 RGB 数据集，首先采用 ResNet 来提取抽象特征。然后，将提取的特征提供给自编码器，将 RGB 数据集的 AE 架构设置为 2048−500−500−1000−d，其中所有层都使用 ReLU 激活函数[2]。

MLP 是一个完全连接的网络，它将 AE 得到的特征向量作为输入，并生成 K 维向量。MLP 的架构为 128−128−128−K。对于 MLP 的所有层中，除使用 softmax 函数的最后一层之外，都使用了 BN 层和 ReLU 激活函数。

对于潜在空间中特征向量h的维数，选取灰度图像的维数为 10，RGB 图像的维数为 20。为了初始化神经网络的参数，即自编码器中θ_e，θ_d和簇中心μ，选择仅通过最小化样本重建损失来执行端到端训练，以此训练自动编码器，然后使用 k−means 算法应用于训练后的潜在空间，得到初始化的簇中心。

对于所有数据集，在算法的第一阶段中，λ、M_1、T_1分别设置为 0. 7、100 和 2。第二阶段中超参数 ζ、γ、τ、T_2和M_2分别设置为 0. 8、0. 2、0.5、5 和 50。其中T_1/T_2指每T_1/T_2轮进行一次簇中心的更新。本章中利用 Adam 优化器更新自编码器和 MLP 的权重，其学习率分别设置为 10^{-5} 和 10^{-3}。

3.3.4 实验结果与分析

（1）对比实验及分析

该部分将所提算法在四个数据集上与其他多个算法进行了比较，表 3.2 与表 3.3 分别展示了论文所提算法与对比算法的 NMI 值与 ACC 值的结果，表中各个数据集的最优结果均使用黑体进行表示。可以看出，在灰度图像与 RGB 图像数据集中，本章所提算法的 NMI 值与 ACC 值都显著优于其他算法。

通过表中数据可进一步观察，基于表示的聚类方法（例如 AE、AEVB 等）的性能优于传统方法（例如 K−means、SC 等）。这表明聚类方法对性能的影响相对较小，而数据的表示更加重要，这显示了表示学习在聚类中起着较为重要的作用。因此，论文从数据的表示方面去提升聚类效果是可行并且有效的。从表中可以看出，本章所提算法在所有数据集上相较其他算法均效果更好。对于灰度图像 MNIST，由于其相对简单容易区分，可看出各算法在该数据集上表现没有较大差异；而在更为复杂的 RGB 图像数据集 CIFAR−10、CIFAR−100、STL−10 上，所提算法具有更为明显的优势，效果提升更多，这验证了所提算法对于大规模图像数据集的有效性。综上所述，本章所提算法在表示学习方面的改进具有较好的效果，验证了其有效性。

表 3.2 各方法聚类结果的 NMI 值

	MNIST	CIFAR-10	CIFAR-100	STL-10
k-means	0.500	0.087	0.084	0.125
SC	0.663	0.103	0.090	0.098
AC	0.609	0.105	0.098	0.239
LPMF	0.452	0.081	0.079	0.096
AE	0.725	0.239	0.100	0.249
SAE	0.756	0.247	0.109	0.252
DAE	0.756	0.251	0.111	0.224
DeCNN	0.757	0.239	0.092	0.227
SWWAE	0.736	0.233	0.103	0.196
GAN	0.764	0.265	0.120	0.210
AEVB	0.736	0.245	0.108	0.200
DEC	0.772	0.257	0.136	0.276
DCN	0.810	0.246	0.125	0.241
DKM	0.815	0.261	0.123	0.291
算法 3.3	**0.861**	**0.621**	**0.358**	**0.610**

表 3.3 各方法聚类结果的 ACC 值

	MNIST	CIFAR-10	CIFAR-100	STL-10
k-means	0.572	0.228	0.129	0.192
SC	0.696	0.247	0.136	0.159
AC	0.695	0.227	0.138	0.332
LPMF	0.471	0.191	0.118	0.180
AE	0.812	0.313	0.165	0.303
SAE	0.827	0.297	0.157	0.320
DAE	0.832	0.297	0.151	0.302
DeCNN	0.818	0.282	0.133	0.298
SWWAE	0.825	0.284	0.147	0.270
GAN	0.828	0.315	0.151	0.281
AEVB	0.831	0.291	0.152	0.298
DEC	0.843	0.301	0.185	0.359
DCN	0.830	0.305	0.201	0.338
DKM	0.840	0.353	0.181	0.326
算法 3.3	**0.875**	**0.719**	**0.382**	**0.702**

（2）消融实验

消融实验共分为两部分，第一部分验证第一阶段中加入增强损失的有效性；第二部分与只使用增强数据的对比学习做比较，验证所提算法的有效性。

a.有关增强损失的消融实验

验证增强损失的影响实验中，在第一阶段单独使用中心损失对潜在空间进行训练，与中心损失、增强损失联合训练的结果进行比较。表 3.4 展示了在数据集 MNIST、CIFAR-10 上的实验结果。从表中可以看出使用增强损失可以提升聚类算法的性能，说明第一阶段中增强损失的使用有助于进一步学习到数据的本质特征，避免学到低级特征，验证了增强损失的有效性。

表 3.4 增强数据的影响实验

	MNIST		CIFAR-10	
	NMI	ACC	NMI	ACC
中心损失	0.770	0.829	0.583	0.667
中心损失+增强损失	0.861	0.875	0.621	0.719

b.与对比学习基本框架的比较实验

为验证所提算法的有效性，该部分与基于对比学习基本框架的深度聚类算法进行比较，将对比学习基本框架直接与聚类相结合，只使用原始数据与增强数据进行对比学习。SimCLR 为典型的对比学习算法，K-means 与谱聚类（SC）为经典的传统聚类算法，因此选择 SimCLR 分别与 SC、K-means 相结合，通过

SimCLR 学习数据的重要特征，之后经过聚类算法得到最终结果。表 3.5 展示了三种算法在数据集 CIFAR-10、CIFAR-100 上的结果对比，其中使用 ACC 评价指标。从表中可以看出所提算法均优于其他两种算法，说明单独使用增强数据进行对比学习学到的特征可能不是聚类的最佳表示，从而影响了最终的聚类性能。证明了所提算法使用成对相似性作为监督构造正负例实现对比学习是有效的。

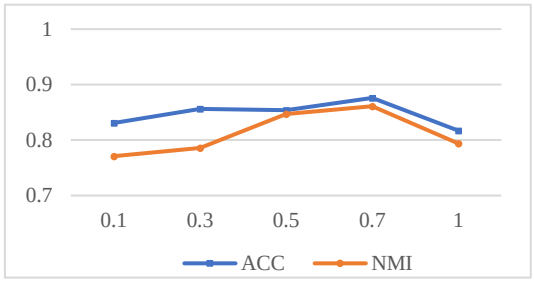
表 3.5 与对比学习基本框架比较

	CIFAR-10	CIFAR-100
SimCLR+SC	0.660	0.292
SimCLR+K-means	0.628	0.380
算法 3.3	0.719	0.382

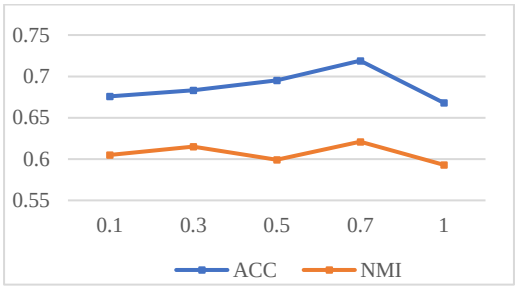
（3）超参数实验

该部分研究了不同超参数对实验结果的影响。

图 3.2 展示了数据集 MNIST、CIFAR-10 中λ不同取值对应的聚类结果，λ为第一阶段损失函数中的权衡参数，可以通过该值研究损失函数中增强损失的重要性。从图 3.2(a)(b)中可以看出，在两个数据集中整体趋势大致相同，都在 0.7 处取得最高值。当取值为 0.1 时，增强损失在损失函数中作用较小，从图中可以看出，两个数据集中实验结果均相对较低。当λ取值增加时，聚类结果有不同程度的提升，但当取值增加到 1 时，聚类结果反而有所下降。其中当取值为 0.7 时聚类性能较好，说明了增强损失对于提升聚类结果是有效的，因此在所有数据集上均采用 0.7 进行实验。



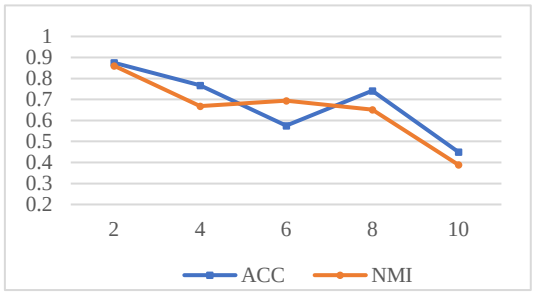
(a) λ对于数据集 MNIST 的影响



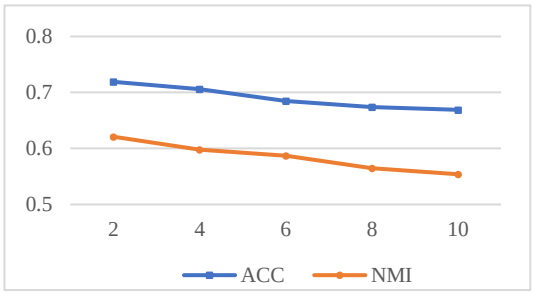
(b) λ对于数据集 CIFAR-10 的影响

图 3.2 λ的不同取值对实验结果的影响

图 3.3 展示了数据集 MNIST、CIFAR-10 中T_1不同取值对结果的影响。T_1表示了第一阶段中更新簇中心的频率，每T_1轮进行一次更新。从图中可以看出，两个数据集结果变动幅度不同，但随着T_1的增加，聚类结果整体都呈下降趋势。较小的T_1值可以获得更好的聚类性能，因此选取T_1值为 2 进行实验。



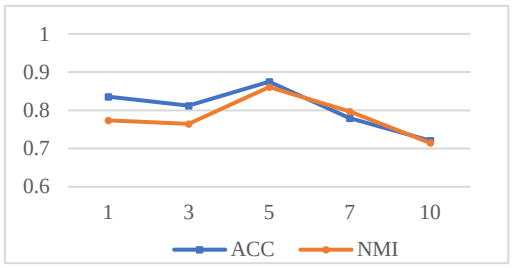
(a) T_1对于数据集 MNIST 的影响



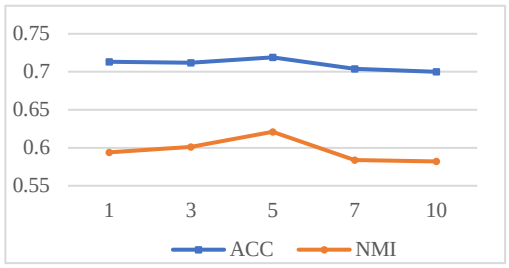
(b) T_1对于数据集 CIFAR-10 的影响

图 3.3 T_1的不同取值对实验结果的影响

图 3.4 展示了在数据集 MNIST、CIFAR-10 中T_2不同取值对实验结果的影响。T_2为第二阶段中更新簇中心的频率。图中可以看出，数据整体波动较小，尤其是数据集 CIFAR-10。当T_2取值为 10 时，评价指标值有所降低。说明在第二阶段中，T_2的取值对于聚类结果的影响相比第一阶段较小。可能是由于在第一阶段中已经形成了相对稳定的簇中心，相比第一阶段不需要频繁的对簇中心进行修改，但修改间隔也不应太大。图中可以得出T_2取值为 5 时聚类结果较好。



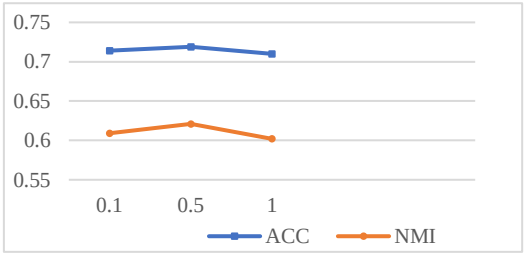
(a) T_2对于数据集 MNIST 的影响



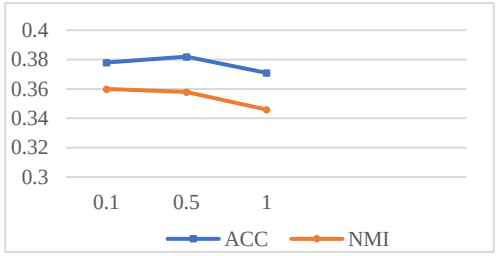
(b) T_2对于数据集 CIFAR-10 的影响

图 3.4 T_2的不同取值对实验结果的影响

图 3.5 展示了数据集 CIFAR-10、CIFAR-100 中温度系数对结果的影响，选取通常使用的 0.1、0.5、1 进行实验。从图中可以看出，CIFAR-10 中τ的改变对于结果影响相对较小。但总体来讲，温度系数太大或太小所得到的结果都有不同程度的降低，因此本章选取τ为 0.5 进行实验。



(a) τ对于数据集 CIFAR-10 的影响



(b) τ对于数据集 CIFAR-100 的影响

图 3.5 τ的不同取值对实验结果的影响

3.4 本章小结

该章节在将对比学习与深度聚类相结合时，避免对比学习只强调样本与增强数据之间的相似性，通过考虑数据与簇之间的关系实现对比聚类。该章节提出了基于自步学习的对比聚类算法，整体采用了自步学习的框架，共分为两阶段，第一阶段主要学习数据间的成对相似性关系，同时在获得的初始潜在空间中，将易分的数据分布于对应簇中心的周围。第二阶段，基于学到的成对相似性关系构造正负例进行对比学习进行训练，训练中使难分数据逐渐归于对应的簇，形成簇内较为紧凑，簇间远离的聚类结果。

第 4 章 基于图结构的实例-簇级别对比聚类研究

由于对比学习良好的特征学习效果，可以将其与深度聚类相结合，但在结合时应考虑类别信息和聚类实际目标，避免过分依赖于数据增强，单纯通过单个实例与其增强数据进行学习。本章超越单个实例，通过考虑数据与数据、簇与簇之间的关系进行对比聚类。使用不同于第三章的方式考虑数据间关系，通过图结构发现数据间是如何相关的，将其纳入对比学习。同时从实例级别扩展到簇级别，从两个方向同时进行对比学习，使对比聚类朝向最终的聚类目标，提高算法的性能。通过实验验证了该算法的有效性。

4.1 算法思想与框架概述

为学习到较好的特征表示，论文选择将对比学习应用于深度聚类，有研究者直接使用对比学习中的增强样本，遵循对比学习基本框架，只假设同一数据的增强样本在特征空间中应该是相似的，忽略聚类任务的本质，过分依赖于数据增强。但由于不同数据集的复杂性，通常采用的几种数据增强方式是否适合于所有的数据集是不确定的，并且大量利用数据增强会导致原始数据的利用不充分，因此过分强调数据增强可能会影响最终的结果。本章算法充分利用原始数据，通过考虑数据与数据、簇与簇间的关系构造对比学习中的正负例以学习聚类友好型表示，将其与深度聚类相结合。图作为重要的数据结构之一，被广泛应用于社交网络、推荐系统等方面，该算法超越单个实例，通过图结构发现数据中的局部结构关系并将其集成到对比学习中确定实例的分组。以聚类为导向从实例级别与簇级别分别进行对比学习，提高聚类性能，得到良好的聚类效果。

该算法主要采用深度子空间聚类网络思想，在自编码器中加入自表达层进行图结构的学习，之后连接两个不同的对比头——实例级别对比头与簇级别对比头，由簇级别对比头得到最终聚类结果。算法假设一个簇中的样本应该共享相似的特征表示和簇分配，将常用的实例级别提升到簇级别，从两个角度进行双重对比学习，同时学习聚类友好型表示与聚类。

算法首先通过自表达层学习到的自表达矩阵构造相应的图，其中蕴含着数据之间的关联信息。一方面，通过图构造正负例实现实例级别对比学习，希望正例

之间的特征表示尽可能相似，负例之间的特征表示尽可能不同。另一方面，按照数据及其邻域具有相似的簇分配，将特征矩阵的每一列作为簇分布，基于图中最近邻关系进行簇级别的对比学习，以实现双重对比学习。该方法在进行实例分组的同时从簇的角度也进行对比学习，都加入了潜在的类别信息，从两方面进行训练以减少簇内方差，增加簇间方差，在获得聚类友好型表示的同时实现聚类。

算法整体网络结构见图 4.1。

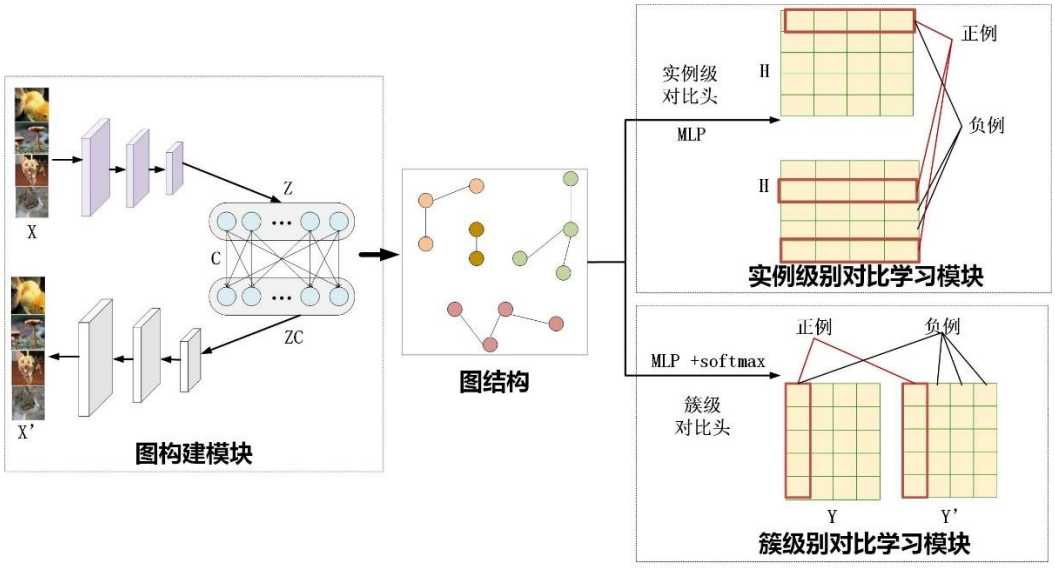


图 4.1 基于图结构的实例-簇级别对比聚类网络框架

4. 2 基于图结构的实例-簇级别对比聚类算法

4. 2. 1 图构建模块

图是最常见的数据结构类型之一，由顶点和连接顶点的边构成，顶点表示对象，而边可以表示两个对象之间的特定关系。本章考虑使用图的性质来体现出数据间的关系，决定数据间应该相互聚集或相互分离。

深度子空间聚类认为数据处于多个低维子空间的并集中，其目的是将所有数据正确划分到对应的低维子空间，每一个子空间表示一个类。具体来说，主要通过神经网络学习原始数据的低维特征表示，计算亲和度矩阵,然后利用谱聚类获得最终聚类结果。谱聚类是从图论中演化出来的算法，主要思想是使用图来描述数据间的关系，通过对所有数据点组成的图进行切图，从而达到聚类的目的。因此，深度子空间聚类的重点在于构造好的亲和度矩阵来表示图，该矩阵极大地影

响着最终聚类的结果。利用深度子空间聚类构造图，其中起作用的关键部分为它的自表达特性，该特性指出子空间中的一个数据点可以表示为该空间中其他数据点的线性组合。算法使用深度子空间聚类中自表达特性构造图结构，利用其中蕴含的数据关系实现后续的对比聚类。

图构建模块的主要网络为对称特征提取网络 AE 与自表达层，在编码器与解码器之间加入无偏置的全连接层学习自表达特性，AE 采用由多个卷积层组成的卷积自编码器。首先使用 AE 对原始数据X进行特征提取$Z = f_\theta(X)$，f_θ表示编码器及其参数。为了保证学习到的表示Z包含来自原始数据X的重要特征，将特征Z输入到编码器的对称网络解码器中，重构数据X'，希望重构后的数据与原始数据尽可能相似以保证学到有意义的信息。使用的编码器-解码器损失函数是数据间的重构误差，公式如下所示：

$$L_{re} = \sum_{i=1}^{N} ||x_i - x_i'||_2^2 = \sum_{i=1}^{N} ||x_i - g_\varphi(f_\theta(x_i))||_2^2 \tag{4.1}$$

其中g_φ表示解码器及其参数，$x_i' = g_\varphi(f_\theta(x_i))$，$N$是数据集大小。

在对数据进行特征提取后，通过该特征构建图结构，使用自表达模块学习自表达系数，主要通过无偏置的全连接层来实现自表达中的线性组合。对于学到的特征表示z_i，自表达层可以计算出由其他节点$z_{j,j\neq i}$的线性组合来近似z_i，该部分损失函数如下，通过最小化损失来优化C：

$$L_{self} = ||C||_l + \lambda||Z - ZC||_F^2 \tag{4.2}$$

其中，$Z = \{z_1, z_2, \dots, z_N\}$是提取好的特征向量；$C$为描述自表达特性的自表达系数矩阵，表示组合系数，在网络中为全连接层的参数；λ为权衡参数。

本章使用自表达系数矩阵C建立图结构，得到一个有向图$\tilde{G}$（X, $\tilde{E}$），图的顶点X是N个数据点，当数据点x_j是x_i的表示向量之一时边$(x_i, x_j) \in \tilde{E}$存在，即$C_{ij} \neq 0$。自表达矩阵中的值作为两数据间的权重，可以看出C是图$\tilde{G}$的邻接矩阵。$\tilde{G}$是一个有向图，因此C是不对称的，为方便后续计算，构造一个新的无向加权图$G(X, E)$，其中$A_{ij} = C_{ij} + C_{ji}$。A为图G的邻接矩阵，是有效的相似度表示，与C的本质含义相同。之后利用图G中蕴含的数据间关系构造后续的对比聚类。

该部分总体损失函数为：

$$L_{gra} = L_{re} + L_{self} \tag{4.3}$$

具体算法见算法 4.1。

算法 4.1 图构建算法

输入：数据集X；迭代次数M；

输出：矩阵 A；

1. 预训练 AE 进行参数初始化θ，φ；
2. 初始化自表达矩阵C；
3. for iter = 1 to M:
4. $Z = f_\theta(X)$，$X' = g_\varphi(Z)$；
5. 使用公式(4.1)计算重构误差L_{re}；
6. 使用公式(4.2)计算自表达损失L_{self}；
7. 使用公式(4.3)计算损失L_{gra}并更新网络参数；
8. 获得更新后的自表达矩阵C；
9. 利用C获得矩阵A；// X作为点，A作为边的权重表示图$G(X,E)$。
10. End

4.2.2 实例级别对比学习模块

自表达矩阵的含义就是在子空间中的数据可以用同一子空间中所有其他点的线性组合来表示，而深度子空间聚类认为同一子空间中的数据应该属于同一类别。因此利用自表达矩阵构造的图结构能体现数据间关系，存在边并且边具有较大权重的两个数据有更大的可能属于同一个簇中。为计算方便，将自表达矩阵进行简单变换，新的矩阵仍具有以上特性。因此算法在使用对比学习进行聚类时，按照以上规则作为基本思想构造正负例对。

实例级别的对比头部分加入一个投影头，将特征矩阵映射到新的空间。根据自编码器 AE 提取到的特征Z，通过非线性变换结构，得到最终特征H。对于之后的实例级别对比学习使用特征H进行训练。

主干网络得到的$G(X,E)$是一个无向加权图，其中X为数据点集合，E是边的集合，该图可以由第一部分得到的邻接矩阵A表示。邻接矩阵A为对称矩阵，从A可以看出，其中当$A_{ij} > 0$时，数据点x_i与x_j存在边，数据可以相互表示；$A_{ij} = 0$时，数据点x_i与x_j不相连，不可以相互表示。因此，对于每一个数据x_i，取A中第

i行不为 0 的数据点构成正例对，其余作为负例对，数据x_i的正例集合为：$P(i)=\{p|A_{ip}>0,1\le p\le N\}$;负例集合为：$F(i)=\{f|X-P(i),1\le f\le N\}$。已有工作表示，更多的负例会提高对比学习的效果[29]，因此对正负例个数进行如下限制：若得到的图G中数据点x_i所连边较为稀疏，即A_{ij}中大于 0 的数据点个数较少，则直接按上式得到正例对与负例对；若数据点所连边较多，正例个数大于负例个数，则对此类数据点选择k近邻，即权重值较大的k个数据点作为正例对。对于数据x_i，正例对为A_{ij}中值最大的k个数据点，其余都为负例对。通过此规则进行实例级对比学习，使得图中相连或权重较大的边对应的数据尽可能拉近，同时与其他数据拉远。

对于数据x_i的对比损失函数为：

$$L^i=\frac{-1}{|P(i)|}\sum_{p\in P(i)}log\frac{\exp\left(s(h_i,h_p)/\tau_1\right)}{\sum_{f\in F(i)}\exp\left(s(h_i,h_f)/\tau_1\right)} \tag{4.4}$$

其中$|P(i)|$为正例对个数，τ_1为实例级别的温度系数，相似度$s(a,b)$使用余弦相似度测量：

$$s(a,b)=\frac{ab^T}{||a||*||b||} \tag{4.5}$$

考虑数据集中所有数据，实例级对比损失如下：

$$L_{ins}=\frac{1}{N}\sum_{i\in N}L^i \tag{4.6}$$

4. 2. 3 簇级别对比学习模块

算法通过假设一个簇中的样本共享相似的特征表示和簇分配，将常用的实例级别提升到簇级别。通过对簇信息进行对比学习，使得算法能够学习到更好的簇分配，更适合于聚类任务。从簇角度而言，算法从簇的样本分布中构建正负例进行簇级别的对比学习，并根据簇级别对比头得到最终的聚类分配。

算法基于一个思想：当数据投影到维数等于聚类数的空间时，特征向量相应地表示其软分配，此时特征矩阵的列可以作为数据的聚类预测。与实例级对比头相似，簇类级别的对比头也加入一个投影头，包括非线性变换结构与 softmax 层，其中非线性变换结构与实例级别对比头共享相同参数。将自编码器 AE 提取到的特征矩阵Z，通过簇级别投影头将其映射到K维空间，得到K维特征Y。簇类级别的对比聚类使用特征Y进行训练。

矩阵Y从行的角度可看作如下形式：

$$Y = \begin{bmatrix} y_1 \\ \cdots \\ y_N \end{bmatrix}_{N*K} \qquad Y' = \begin{bmatrix} y'_1 \\ \cdots \\ y'_N \end{bmatrix}_{N*K}$$

Y为 N 个数据通过 softmax 得到的结果特征矩阵，Y'为Y中每一数据对应最近邻的特征矩阵，即y'_i表示在图中与数据x_i相连并且权重最大边的数据点通过 softmax 得到的向量。具体来说，为邻接矩阵A中第i行值最大的数据点对应的向量。其中，y_i为软分配，y_{ij}表示样本x_i属于簇j的分配概率。由于每一个样本只属于一个簇，因此在理想情况下，Y的每一行都为 one-hot 向量。此时，Y的第i列可以看作是第i个簇的表示，并且所有列应该彼此不同。Y'中各含义与Y同理。从列的角度看，Y与Y'可写作如下形式，为每一个簇的数据预测：

$$Y = [cl_1, \ldots, cl_K]_{N*K}, \qquad Y' = [cl'_1, \ldots, cl'_K]_{N*K}$$

从列角度可以看出哪些数据被分配给了簇i，其中cl_i为簇i的表示。

按照以上形式，矩阵 Y和Y'应具有相同簇分配。按照以上规则构造正负例，在簇级别，将cl_i与cl'_i作为正簇对，与其余簇表示都作为负簇对。对于每个簇，存在$K-1$对负簇对。具体损失函数如下，利用该损失函数将不同簇区分开来：

$$L_{clu} = -\frac{1}{K}\sum_{i\in K} \log\left(\frac{\exp\left(s(cl_i, cl'_i)/\tau_2\right)}{\sum_{j\in K}\exp\left(s(cl_i, cl'_j)/\tau_2\right)}\right) \tag{4.7}$$

其中，相似度计算同实例级别一样采用余弦相似度，τ_2为簇级别的温度系数。

实例-簇级别对比学习具体算法见算法 4.2。

算法 4.2 实例-簇级别对比学习

输入：特征Z；邻接矩阵A；迭代次数M；实例级对比头q_{in}；簇级对比头q_{cl}；

输出：各网络参数；

1. for iter = 1 to M:
2. $H = q_{in}(Z)$; $Y = q_{cl}(Z)$
   //实例级别
3. 由矩阵A得出正例对$P(.)$和负例对$F(.)$；
4. 根据公式(4.4)-(4.6)计算实例级对比损失L_{ins}；
   //簇级别
5. 由矩阵A得出特征矩阵Y'；
6. 根据Y与Y'构造正簇对和负簇对；

7. 根据公式(4.7)计算簇级别对比损失L_{clu}；
8. 最小化对比损失$L_{ins}+L_{clu}$更新参数
9. End

4. 2. 4 算法整体流程

给定一个有N个数据的数据集$X=\{x_1,x_2,\dots,x_N\}$，将其输入主干网络得到特征Z与自表达矩阵C，根据C构造出对应的图G，使用邻接矩阵A来表示图G。特征Z通过实例级对比头和簇级对比头分别得到特征H和K维特征Y，根据矩阵A与既定规则来设置正负例对，特征H根据正负例对进行相应的实例级别对比损失，特征Y根据正负簇对进行簇级别对比损失。通过构造的深度子空间聚类损失和双重对比损失进行端到端的训练，整体损失函数如下所示:

$$L=L_{gra}+L_{ins}+L_{clu} \tag{4.8}$$

最终通过簇级对比头得到 K 个簇结果$Cl=\{cl_1,cl_2,\dots,cl_k\}$。

总体算法伪代码见算法 4.3。

算法 4.3 基于图结构的实例-簇级别对比聚类

输入：数据集X；迭代次数M；簇个数K；

输出：聚类结果Cl。

1. 网络参数初始化；
2. for iter = 1 to M:
3. 根据算法 4.1 得到图 G 以及其邻接矩阵 A；
4. 根据算法 4.1 计算出损失函数L_{gra}；
5. 根据算法 4.2 计算出时双重对比损失$L_{ins}+L_{clu}$；
6. 由公式(4.8)通过最小化损失函数L更新参数；
7. End
8. For x in X:
9. 提取特征$z=f_\theta(x)$;
10. 通过 argmax $p_{cl}(z)$计算簇分配;
11. End

4.3 实验及其分析

该部分首先介绍了实验使用的数据集、比较算法、评价指标以及参数设置等实验相关细节。之后为了验证所提算法的有效性，在三个数据集上进行了实验，与多个算法的结果进行了对比，还包括对比头的消融实验与部分参数实验，并对实验结果进行了分析。

4.3.1 数据集及比较算法

（1）实验数据集

论文在三个广泛使用的基准数据集上进行实验比较。以下介绍了这些数据集的特点，表 4.1 显示了数据集具体信息。

CIFAR-10/100:图像大小为 32×32×3。在实验中，CIFAR10 与 CIFAR−100 数据集分别考虑了 10 个类和 20 个超级类，所有 60000 张图像被联合用于聚类。

STL−10:STL−10 是一个图像识别数据集，包含 10 个类别，每个类别包含 1300 张图像，图像大小为 96×96×3。

表 4.1 各数据集详情信息

数据集	数量	簇个数	大小
CIFAR−10	60000	10	32*32*3
CIFAR−100	60000	20	32*32*3
STL−10	13000	10	96*96*3

（2）比较算法

论文在图像基准上评估了所提出的算法，并将其与多个代表性的先进聚类方法进行了比较，包括传统的和基于深度学习的方法，其中包括谱聚类（SC）[46]、凝聚聚类（AC）[47]、NMF[48]、自动编码器（AE）[2]、去噪自动编码器（DAE）[50]、DCGAN[57]、去卷积网络（DeCNN）[51]、变分自动编码（VAE）[21]、深度嵌入聚类（DEC）[2]、联合无监督学习（JULE）[6]、深度自适应图像聚类（DAC）[58]、不变信息聚类[59]、DDC[60]、深度综合相关挖掘（DCCM）[61]、分区置信度最大化（PICA）[62]和深度鲁棒聚类（DRC）[63]。

4. 3. 2 评价指标

本章使用三种常用的度量来评估聚类性能，包括归一化互信息（NMI）、聚类精度（ACC）和调整后的兰德指数（ARI）[64]，数值越大表示聚类性能越高。其中 NMI 与 ACC 指标介绍见第三章实验部分。

ARI 是 RI 指数的推广，它反映两种划分的重叠程度。其计算公式如下：

$$\mathrm{ARI} = \frac{\mathrm{RI}-\mathrm{E}[\mathrm{RI}]}{\max(RI)-E[RI]} \tag{4.9}$$

E[RI]是兰德指数 RI 的期望， $\mathrm{RI} = \frac{a+b}{C_n^2}$，其中$a$为在实际情况与聚类结果中都被划分到同一簇中的样本对数量。b为在实际情况与聚类结果中都被划分到不同簇中的样本对数量，共同来统计实际情况与聚类结果中标签一致的情况，分母表示数据集可以形成的总对数。ARI 取值范围为[-1,1], 值越大聚类结果与真实情况越一致。

4. 3. 3 网络结构及参数设置

本章采用 python3.8 来完成所有实验，使用的网络框架为 pytorch。在本章算法框架中，主要采用卷积自编码器，并在编码器与解码器之间加入一层无偏置的全连接层作为主要的网络架构，其中的激活函数均使用 ReLU 函数。首先通过重构损失对自编码器训练进行初始化，能够提取到较好的数据特征，方便后续操作，其中采用 Adam 优化器进行训练。对于实例级对比头与簇级对比头，均采用两层全连接层加一层 ReLU 函数的结构，其中簇级别对比头在最后加入一层 softmax 获得数据的簇分配结果，最终聚类结果直接经过主网络自编码器与簇级对比头得到。

对于算法中各参数的设置，实例级对比头中，行空间的维数设置为 128 以保留图像的更多信息。在所有实验中，实例级别对比损失中温度参数设置为τ_1=0.5。对于簇级对比头，列空间的维度设置为簇的数量，并且设置温度参数τ_2=1.0用于所有数据集，实验中权衡参数λ设置为 1。对于图的构造，由于实验中对大部分数据来说相连边较多，因此选取所有数据的k近邻来构造图结构，其中设置k=5。在训练中采用学习率为 0. 0001 的 Adam 优化器来同时优化两个对比头和主干网络，训练模型 150 轮。

4.3.4 实验结果与分析

（1）对比实验

该部分将所提算法与多个算法在数据集上的结果相比较，表 4.2、表 4.3、表 4.4 分别展示了论文所提算法与各对比算法的 NMI 值、ARI 值与 ACC 值的结果。其他方法的结果直接取自不同论文的实验结果。

基于这些结果，首先可以看到，基于深度学习的方法比传统的聚类方法获得了更好的结果，准确度远高于传统的聚类算法。例如，在 CIFAR−10 数据集上，大多数基于深度学习的聚类方法的准确度远高于 0.3，而这些经典方法（包括 SC、AC 和 NMF）的准确度低于 0.2。其次，基于对比学习的方法，如 PICA、DRC，相比其他深度聚类算法具有更优的效果，可以学习更具区别性的特征表示。在三种不同的评估指标下，本章所提算法在大多数基准上明显优于其他方法。在 ARI 评价指标中，所提算法在 CIFAR−100 数据集上与 DRC 相差 0.001，在其余结果上，与先进的 PICA 和 DRC 方法相比，都有明显的进步，显示了所提算法可以更好的学习聚类友好型特征。上述结果可以很好地证明本章提出的方法的有效性。

表 4.2 各算法间 NMI 值比较

	CIFAR-10	CIFAR-100	STL-10
SC	0.103	0.090	0.098
AC	0.105	0.098	0.239
NMF	0.081	0.079	0.096
AE	0.239	0.100	0.249
DAE	0.251	0.111	0.224
DCGAN	0.265	0.120	0.210
DeCNN	0.240	0.092	0.227
VAE	0.245	0.108	0.200
JULE	0.192	0.103	0.182
DEC	0.257	0.136	0.276
DAC	0.396	0.185	0.366
DDC	0.424	−	0.371
DCCM	0.496	0.285	0.376
PICA	0.591	0.310	0.611
DRC	0.621	0.356	0.644
算法 4.3	**0.661**	**0.357**	**0.663**

表 4.3　各算法间 ARI 值比较

	CIFAR-10	CIFAR-100	STL-10
SC	0.085	0.022	0.048
AC	0.065	0.034	0.140
NMF	0.034	0.026	0.046
AE	0.169	0.048	0.161
DAE	0.163	0.046	0.152
DCGAN	0.176	0.045	0.139
DeCNN	0.174	0.038	0.162
VAE	0.167	0.040	0.146
JULE	0.138	0.033	0.164
DEC	0.161	0.050	0.186
DAC	0.306	0.088	0.257
DDC	0.329	-	0.267
DCCM	0.408	0.173	0.262
PICA	0.512	0.171	0.531
DRC	0.547	**0.208**	0.569
算法 4.3	**0.602**	0.207	**0.627**

表 4.4　各算法间 ACC 值比较

	CIFAR-10	CIFAR-100	STL-10
SC	0.247	0.136	0.159
AC	0.228	0.138	0.332
NMF	0.190	0.118	0.180
AE	0.314	0.165	0.303
DAE	0.297	0.151	0.302
DCGAN	0.315	0.151	0.298
DeCNN	0.282	0.133	0.299
VAE	0.291	0.152	0.282
JULE	0.272	0.137	0.277
DEC	0.301	0.185	0.359
DAC	0.522	0.238	0.470
IIC	0.617	0.257	0.310
DDC	0.524	-	0.489
DCCM	0.623	0.327	0.482
PICA	0.696	0.337	0.713
DRC	0.727	0.367	0.747
算法 4.3	**0.764**	**0.393**	**0.775**

（2）消融实验

为了研究实例级别对比头与簇级别对比头对聚类结果的影响，该部分通过移除其中某一个对比头进行消融实验，在数据集 CIFAR-10 与 STL-10 上进行实验。

当移除簇级别对比头时，不能直接获得最终的簇分配，因此在实例级别训练完成后，在特征空间内使用 k-means 获得最终聚类结果进行比较。表 4.5 展示了三种情况下的聚类结果，从表中可以看出两种对比头进行联合训练时的结果要优于使用单一对比头的结果。证明了簇级别对比头的有效性，以及在所提算法中两种对比头的共同使用可以对聚类性能进行一定的改进。

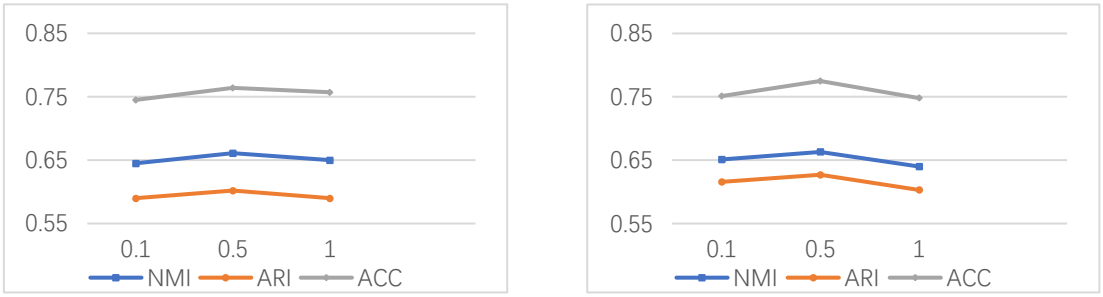
表 4.5 对比头的影响比较

	对比头	NMI	ARI	ACC
CIFAR-10	实例级别	0.648	0.595	0.758
	簇级别	0.655	0.598	0.761
	实例级别+簇级别	0.661	0.602	0.764
STL-10	实例级别	0.615	0.602	0.731
	簇级别	0.653	0.619	0.762
	实例级别+簇级别	0.663	0.627	0.775

（3）超参数实验

该部分为算法中超参数对聚类结果的影响实验，展示了在数据集 CIFAR-10 与 STL-10 上的实验结果。

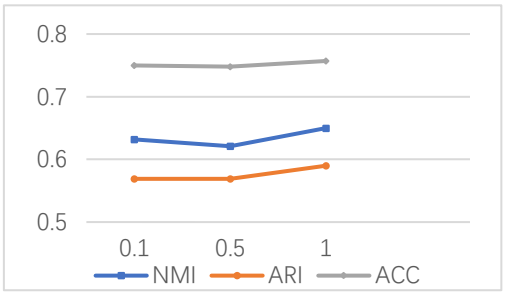
图 4.2 展示了实例级对比学习中温度参数对实验结果的影响，分别选取参数为 0.1、0.5、1 时的结果进行比较。从图中可以得出取值较大或较小都会在一定程度上造成聚类结果的下降，当取值为 0.5 时聚类结果达到最优。



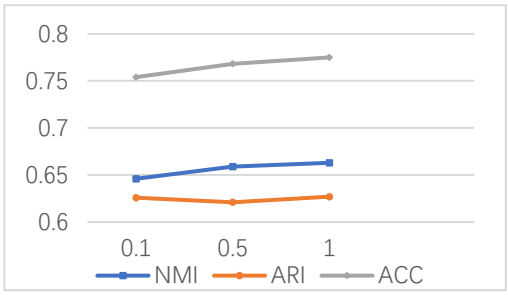
(a) τ_1对数据集 CIFAR-10 的影响　　(b) τ_1对数据集 STL-10 的影响

图 4.2 τ_1的不同取值对结果的影响

图 4.3 展示了簇级对比学习中温度参数对实验结果的影响。图中可以看出，与实例级别中温度参数的影响不同，随着取值的增大，整体呈上升趋势。其中，对数据集 STL-10 的影响相对较小。当取值为 1 时聚类结果较好，因此选取$\tau_2 = 1$进行本章实验。



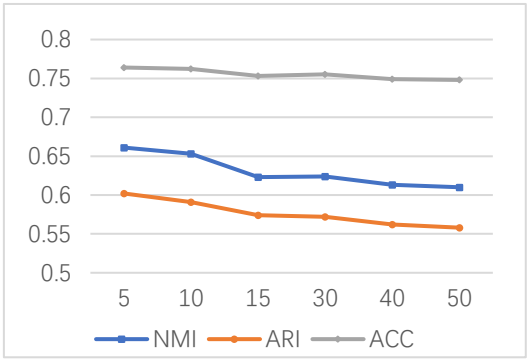
(a) τ_2对数据集 CIFAR-10 的影响



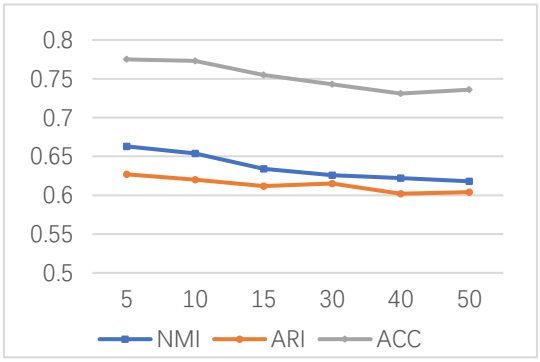
(b) τ_2对数据集 STL-10 的影响

图 4.3 τ_2的不同取值对结果的影响

图 4.4 展示了近邻个数对实验结果的影响。从图中可以看出，随着近邻个数的增加，聚类结果整体呈下降趋势，当取值大于 15 后，下降幅度有所减小。可能是由于近邻个数太多时有较大概率将相似度较低或不属于同一簇的数据作为正例，从而影响了实验结果，近邻取值较小时可以在一定程度上保证这些数据属于同一簇中。在取值为 5 时效果较好，因此选取 k = 5 进行实验。



(a) k 对数据集 CIFAR-10 的影响



(b) k 对数据集 STL-10 的影响

图 4.4 k 的不同取值对结果的影响

4. 4 本章小结

该章节将对比学习与深度聚类相结合，避免单纯通过单个实例与其增强数据进行学习，过分依赖数据增强。算法超越单个实例，结合时考虑了类别信息和聚类目标，通过数据与数据、簇与簇之间的关系进行对比聚类。通过图结构发现数据间的相关关系并基于此实现对比学习。图结构的发现主要通过深度子空间聚类基本思想，得到图结构后，从实例级别扩展到簇级别，从两个方向同时进行对比学习，使对比聚类朝向最终的聚类目标，提高算法的性能，最终结果通过簇级对比头获得。通过实验验证了所提算法的有效性。

第 5 章 总结与展望

5.1 总结

面对大量毫无规律且类型混杂的数据，快速从中挖掘出有价值的信息和规律是关注的重要问题之一。聚类以无监督方式将数据进行分组，是数据挖掘中重要的方法，被广泛应用于各方面的研究。深度聚类利用神经网络对数据进行表示学习，因此聚类性能在很大程度上取决于数据表示的质量。已有工作将对比学习与深度聚类算法相结合，但此类方法遵循对比学习基本框架，只假设样本与其增强样本在特征空间中是相似的，单纯使用对比学习做前期的表示学习，忽略了聚类任务的潜在信息，使用固定的损失函数，而不是专门面向聚类任务。本文在将对比学习应用于深度聚类以学习良好表示的基础上，提取有效的、聚类友好型表示，高效进行聚类与表示学习，提高深度聚类的性能。论文通过不同方式考虑数据与簇间、数据与数据间、簇与簇间的关系，提出两种算法，超越了单个样本信息，探索样本间关系，利用对比学习获得聚类友好型表示以提升深度聚类算法的效率。所提算法总结如下：

（1）基于自步学习的实例级别对比聚类算法

该算法引入了自步学习的思想，分为两阶段通过由简到难的方式对数据进行训练。在第一阶段得到的初始特征空间中将容易区分的数据进行初步聚类。第一阶段通过考虑数据与簇之间的关系，采用中心损失对神经网络进行训练，该过程中将易分数据进行了初步聚类，同时在训练中学习到数据的成对相似性为后续过程进行监督。第一阶段完成后，数据被映射到一个初始的潜在空间，在该空间内易分的数据分布在相应的簇中心附近。

第二阶段在训练的同时使难分的数据逐渐变得容易区分，最后聚类到合适的簇中。具体来说根据第一阶段学到的数据间关系设定正负例，从而采用对比学习进行微调训练，对特征空间进行改进。在第二阶段的训练中，随着训练时间的增加，越来越多的难分样本逐渐变得容易区分，最后得到聚类结果。

（2）基于图结构的实例-簇级别对比聚类算法

该算法同时考虑数据与数据间、簇与簇间的关系，将对比学习与深度聚类结

合。算法通过在自编码器中加入自表达层获取图结构，从中反应样本的邻域信息。在表示学习方面，提出了基于图的实例级对比损失学习区分性的特征，其中的正负例根据图结构确定。与此同时，将常用的实例级别提升到簇级别。按照标签表示的思想，当数据投影到维数等于聚类数的空间时，特征向量相应地表示其软分配。因此将特征矩阵的列作为数据的聚类预测，在列方向进行基于图的簇级别对比学习。从实例级别引申到簇级别，两方面都融合了潜在的类别信息，从两方面进行训练以增加簇内样本的紧密度，减少簇间样本的紧密度，同时进行特征学习与集群分配。

针对提出的两种算法，本文分别在多个数据集上进行了实验，与传统聚类算法、基于表示学习的聚类算法、先进的深度聚类算法相比较，都有不同程度的提升，证明了所提算法的有效性，提升了聚类效果，具有重要的理论意义与使用价值。

5.2 展望

本文针对深度聚类算法进行研究，通过对比学习更好地与深度聚类相结合来提升聚类算法的效果，还有以下几方面需要继续深入研究：

（1）在基于自步学习的对比聚类算法中，为方便采用了两阶段的训练，首先获得数据间关系，再基于此构造正负例进行后续对比学习。该算法没有采用端到端的方式进行训练，而现在较多的深度聚类算法开始采用端到端方式对表示学习与聚类进行联合优化，部分已有工作表示采用端到端方式可能会提高算法效果。因此，如何使用端到端更好的将两者进行结合是后续可以优化的方向之一。

（2）在采用图结构表示数据间关系时，主要采用深度子空间聚类的思想进行图结构的构建，而深度子空间聚类中自表达层的构造与样本大小密切相关，因此构建时所需计算成本相对较高。可利用一些方法对深度子空间聚类中自表达层的构建进行优化，也可思考利用其他更为简单方便的方法构建图，是论文中可继续研究的方面。

参考文献

[1] 曹妥恰. 基于图结构的聚类算法研究[D], 北京交通大学硕士学位论文, 2021.

[2] Xie J, Girshick R, Farhadi A.Unsupervised deep embedding for clustering analysis[C].International Conference on Machine Learning, 2016:478-487.

[3] Guo X, Gao L, Liu X, et al. Improved deep embedded clustering with local structure preservation[C].International Joint Conference on Artificial Intelligence, 2017:1753-1759.

[4] Li F, Qiao H,Zhang B. Discriminatively boosted image clustering with fully convolutional auto-encoders[J].Pattern Recognit,2018,(83):161-173.

[5] Yang B, Fu X, Nicholas D, et al. Towards k-means-friendly spaces:simultaneous deep learning and clustering[C]. International Conference on Machine Learning,2017:3861–3870.

[6] Yang J, Parikh D, Batra D. Joint unsupervised learning of deep representations and image clusters[J].Pattern Recognit, 2016:5147–5156.

[7] Ji P, Zhang T, Li H, et al. Deep subspace clustering networks[J]. Advances in Neural Information Processing Systems, 2017:23–32.

[8] Zhang J, Li C, You C,et al. Self-Supervised convolutional subspace clustering network[C]. Computer Vision and Pattern Recognition, 2019:5468-5477.

[9] Mahdi A, Alireza N, Dimitris N,et al. Deep subspace clustering with data augmentation[C]. Advances in Neural Information Processing Systems ,2020: 10360-10370.

[10] Huang P, Huang Y, Wang W,et al. Deep embedding network for clustering[J]. Pattern Recognit, 2014 :1532–1537.

[11] Chen D, Lv J, Yi Z. Unsupervised multi-manifold clustering by learning deep representation[C]. AAAI Conference on Artificial Intelligence, 2017:385–391.

[12] Shah S, Koltun V. Deep continuous clustering[Z]. ArXiv preprint, 2018.

[13] Hsu C, Lin C. CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data[J]. IEEE Transactions on Multimedia,

2018,20(2):421–429.

[14] Dizaji K, Herandi A, Deng C,et al. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization[C]. IEEE International Conference on Computer Vision, 2017: 5747–5756.

[15] Hu W , Miyato T , Tokui S. Learning discrete representations via information maximizing self augmented training[C].International Conference on Machine Learning. 2017,70: 1558-1567.

[16] Guo X , Zhu E , Liu X. Deep embedded clustering with data augmentation[C]. Asian Conference on Machine Learning, 2018, 95:550-565.

[17] Sadeghi M, Armanfard N. Deep successive subspace learning for data clustering [C]. International Joint Conference on Neural Networks, 2021:1-8.

[18] Sadeghi M, Armanfard N. Deep clustering with self-supervision using pairwise data similarities[Z]. ArXiv preprint,2021.

[19] Fogel S, Averbuch-Elor H, Cohen-Or D,et al. Clustering-driven deep embedding with pairwise constraints[J]. IEEE Computer Graphics and Applications, 2019, 39(4):16-27.

[20] 邓祥, 俞璐. 深度聚类算法综述 [J]. 通信技术, 2021, 54(8): 1807-1814.

[21] Jiang Z, Zheng Y, Tan H. Variational deep embedding:an unsupervised and generative approach to clustering[C]. International Joint Conference on Artificial Intelligence,2017:1965-1972.

[22] Dilokthanakulet N, Mediano P, Garnelo M, et al. Deep unsupervised clustering with gaussian mixture variational autoencoders[C]. International Conference on Learning Representations,2017.

[23] Yang L, Cheung N, Li J,et al. Deep clustering by gaussian mixture variational autoencoders with graph embedding[C]. International Conference on Computer Vision, 2019:6439-6448.

[24] 姬强,孙艳丰,胡永利等.深度聚类算法研究综述[J].北京工业大学学报, 2021, 47(08):912-924.

[25] Springenberg J. Unsupervised and semi-supervised learning with categorical generative adversarial networks[C]. International Conference on Learning Representations,2016.

[26] Chen X, Duan Y, Houthooft R, et al. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets[C]. Advances in Neural Information Processing Systems,2016:2172–2180.

[27] Mukherjee S, Asnani H, Lin E,et al. ClusterGAN: Latent space clustering in generative adversarial networks[C]. the AAAI Conference on Artificial Intelligence, 2019,33(01):4610-4617.

[28] Zhang D, Nan F, Wei X. Supporting clustering with contrastive learning[C]. North American Chapter of the Association for Computational Linguistics,2021:5419-5430.

[29] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]. International Conference on Machine Learning, 2020:1597–1607.

[30] Gansbeke W,Vandenhende S,Georgoulis S,et al. SCAN: Learning to classify images without labels[C]. European Conference on Computer Vision, 2020:268-285.

[31] Dang Z, Cheng D, Xu Y. Nearest neighbor matching for deep clustering[C]. Computer Vision and Pattern Recognition, 2021:13693-13702.

[32] Li Y, Hu P, Liu Z,et al. Contrastive clustering[C]. Association for the Advance of Artificial Intelligence, 2021:8547-8555.

[33] Wang X, Liu Z, Yu S. Unsupervised feature learning by cross-level instance-group discrimination[C]. Computer Vision and Pattern Recognition, 2021:12581-12590.

[34] Liu Y, Yang X, Zhou S, et al. Simple contrastive graph clustering [Z], ArXiv preprint, 2022.

[35] Sharma V, Tapaswi M, Sarfraz M, et al. Clustering based contrastive learning for improving face representations[C]. Automatic Face and Gesture Recognition, 2020:109-116.

[36] Ma X， Kim W. Locally normalized soft contrastive clustering for compact clusters[C]. International Joint Conference on Artificial Intelligence, 2022:3288-3295.

[37] Lin F, Bai B, Bai K, et al. Contrastive multi-view hyperbolic hierarchical clustering[C]. International Joint Conference on Artificial Intelligence,2022:3250-3256.

[38] Min E, Guo X, Liu Q, et al. A survey of clustering with deep learning: from the perspective of network architecture[J].IEEE Access, 2018,6:39501-39514.

[39] Elhamifar E, Vidal R. Sparse subspace clustering[C]. Computer Vision and Pattern Recognition, 2008:2790–2797.

[40] Liu G, Lin Z, Yan S,et al. Robust recovery of subspace structures by low-rank representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013,35(1):171–184.

[41] Favaro P, Vidal R, Ravichandran A. A closed form solution to robust subspace estimation and clustering[C]. Computer Vision and Pattern Recognition, 2011:1801–1807.

[42] Lu C, Min H, Zhao Z,et al. Robust and efficient subspace segmentation via least squares regression[C]. European Conference on Computer Vision, 2012:347–360.

[43] Ji P, Salzmann M, Li H. Efficient dense subspace clustering[C]. Workshop on Applications of Computer Vision, 2014:461–468.

[44] Lee Y, Grauman K. Learning the easy things first: self-paced visual category discovery[C], Computer Vision and Pattern Recognition, 2011:1721-1728.

[45] Wang J, Song J, Xu X, et al. Optimized cartesian k-means[J]. IEEE Transactions on Knowledge and Data Engineering. 2015,27(1):180–192.

[46] Zelnik-Manor L, Perona P. Self-tuning spectral clustering[C]. Neural Information Processing Systems, 2004:1601–1608.

[47] Gowda K, Krishna G. Agglomerative clustering using the concept of mutual nearest neighbourhood[J]. Pattern Recognition, 1978,10(2):105–11.

[48] Cai D, He X, Wang X,et al. Locality preserving nonnegative matrix factorization[C]. International Joint Conference on Artificial Intelligence, 2009:1010–1015.

[49] Ng A. Sparse autoencoder[Z]. CS294A Lecture notes, 2011,72:1–19.

[50] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010,11:3371–3408.

[51] Zeiler M, Krishnan D, Taylor G,et al. Deconvolutional networks[C]. Computer Vision and Pattern Recognition, 2010:2528–2535.

[52] Zhao J, Mathieu M, Goroshin R,et al. Stacked what-where auto-encoders[C]. International Conference on Learning Representations, 2016.

[53] Kingma D, Welling M. Auto-encoding variational bayes[C]. International Conference on Learning Representations,2014.

[54] Maziar M, Thibaut T, Eric G. Deep k-means: Jointly clustering with k-means and learning representations[J]. Pattern Recognition, 2020,138:185–192.

[55] Yang Y, Xu D, Nie F, et al. Image clustering using local discriminant models and global integration[J]. IEEE Transactions on Image Processing, 2010,19(10): 2761–2773.

[56] Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization[C]. Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, 2003:267–273.

[57] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[C]. Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, 2018:31-38.

[58] Chang J, Wang L, Meng G, et al. Deep adaptive image clustering[C].IEEE International Conference on Computer Vision, 2017: 5879–5887,.

[59] Xu J, Henriques J, Andrea V. Invariant information clustering for unsupervised image classification and segmentation[C]. IEEE International Conference on Computer Vision, 2019: 9865–9874.

[60] Chang J, Guo Y, Wang L, et al.Deep discriminative clustering analysis[Z]. ArXiv preprint,2019 .

[61] Wu J, Long K, Wang F, et al. Deep comprehensive correlation mining for image clustering[C]. IEEE International Conference on Computer Vision, 2019:8150–8159.

[62] Huang J, Gong S, Zhu X. Deep semantic clustering by partition confidence maximisation[C]. Computer Vision and Pattern Recognition, 2020: 8849–8858.

[63] Zhong H, Chen C, Jin Z, et al. Deep robust clustering by contrastive learning[Z]. ArXiv preprint, 2020.

[64] Vinh N, Epps J, Bailey B. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance [J]. Journal of Machine Learning Research, 2010, 11(11):2837–2854.