分类号 _____ 密级 _____
UDC _____ 编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕士学位论文

论文题目 线性投影的高维数据聚类算法研究

研 究 生 姓 名：吴义稳

指导教师姓名、职称：聂飞平 教授

学科、专业名称：管理科学与工程

研 究 方 向：信息管理与信息系统

提 交 日 期：2024年 5月 31 日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：吴义稳 签字日期：2024·5·30

导师签名：聂飞平 签字日期：2024.5.30

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入 CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：吴义稳 签字日期：2024·5·30

导师签名：聂飞平 签字日期：2024.5.30

A Study on Clustering Algorithms for High-dimensional Data with Linear Projection

Candidate : Wu Yiwen

Supervisor: Nie Feiping

摘 要

随着经济社会的发展，数据呈现爆炸性增长，这种增长主要体现在数据的数量和维度。高维数据的样本稀疏且包含大量的冗余特征和噪声信息，其有意义的类簇结构往往嵌入在低维子空间中。对高维数据进行降维是提高聚类算法性能的一个关键步骤。线性投影技术是常用的降维方法，其通过线性变换将高维数据变换到低维子空间中。传统的聚类算法在对高维数据进行聚类时主要存在以下两个问题：（1）对异常样本敏感，缺乏识别和处理异常样本的过程；（2）需要预先对高维数据进行降维处理，算法的时间复杂度通常较高，不适用于高维的大数据集。针对这些问题，本文提出了两个算法。

一、提出了带有实例惩罚的投影模糊 C 均值聚类算法（Projected Fuzzy c-means Clustering Algorithm with Instance Penalty，PCIP）。PCIP 算法将聚类任务和降维任务统一到一个目标函数中，迭代地学习线性投影矩阵和隶属度矩阵，同时完成降维和聚类。此外，PCIP 算法基于原始数据的分布构建实例惩罚矩阵，为每一个样本分配实例惩罚系数，降低了噪声样本对模型的影响。PCIP 算法还通过融合模糊 C 均值聚类（Fuzzy c-means Clustering，FCM）和主成分分析（Principal Component Analysis，PCA）来构造一个新颖的高维数据聚类模型。PCIP 的时间复杂度与样本数量线性相关，可以有效地处理大型数据集。为了验证 PCIP 算法的有效性，在 10 个图像数据集上与 7 个相关的对比算法进行了大量的聚类实验，实验结果证明了 PCIP 算法能够快速地学习投影矩阵和隶属度矩阵，并在投影子空间上获得良好的聚类效果。

二、提出了快速锚点图保持投影算法（Fast Anchor Graph Preserving Projections，FAGPP）。为了减少基于图的降维算法的时间复杂度，FAGPP 算法使用锚点图的学习代替邻接图的学习，同时利用原始数据空间的锚点图来指导投影空间中锚点图的学习。FAGPP 算法还融合了 PCA 模型使其不仅可以处理数据的聚类信息，还可以处理数据的全局信息。FAGPP 的时间复杂度为 $O(nmd)$，其中 n 表示样本的数量，m 表示锚的数量，d 表示数据的特征数。为了验证 FAGPP 算法的有效性，在 6 个图像数据集上与 5 个相关的对比算法进行了大量的聚类实验，实验结果证明 FAGPP 算法能快速地对数据进行降维，降维

的数据在投影子空间中具有较好的类簇结构，获得良好的聚类效果。

关键词: 降维　聚类　锚点　线性投影　实例惩罚

Abstract

With the development of economy and society, data shows explosive growth, and this growth is mainly reflected in the number and dimension of data. The samples of high-dimensional data are sparse and contain a lot of redundant features and noise information, and their meaningful class cluster structures are often embedded in low-dimensional subspaces. Dimensionality reduction of high dimensional data is a key step in improving the performance of clustering algorithms. The linear projection technique is a commonly used dimensionality reduction method that transforms the high-dimensional data into a low-dimensional subspace through a linear transformation. Traditional clustering algorithms have the following two main problems when clustering high-dimensional data: (1) they are sensitive to abnormal samples and lack the process of identifying and dealing with abnormal samples; (2) they need to reduce the high-dimensional data beforehand, and the time complexity of the algorithms is usually high, which is not applicable to high-dimensional large data sets. To address these problems, two algorithms are proposed in this paper.

(1) A Projected Fuzzy c-means Clustering Algorithm with Instance Penalty (PCIP) is proposed. The PCIP algorithm unifies the clustering task and the dimensionality reduction task into a single objective function, which iteratively learns the linear projection matrix and the membership

matrix, and performs dimensionality reduction and clustering simultaneously. In addition, the PCIP algorithm constructs an instance penalty matrix based on the distribution of the original data and assigns an instance penalty coefficient to each sample to reduce the influence of noisy samples on the model. The PCIP algorithm also reduces the influence of noisy samples on the model by merging Fuzzy c-means Clustering (FCM) and Principal Component Analysis (PCA) into a single objective function for clustering and dimensionality reduction. The PCIP algorithm also constructs a novel clustering model for high-dimensional data by fusing Fuzzy c-means Clustering (FCM) and Principal Component Analysis (PCA). The time complexity of the PCIP algorithm is linearly correlated with the number of samples, and it can efficiently handle large data sets. In order to verify the effectiveness of the PCIP algorithm, a large number of clustering experiments are conducted on 10 image datasets with 7 related comparison algorithms, the experimental results demonstrate that the PCIP algorithm is able to learn the projection matrix and the affiliation matrix quickly and obtain good clustering results on the projection space.

(2) Fast Anchor Graph Preserving Projections (FAGPP) algorithm is proposed. In order to reduce the time complexity of the graph-based dimensionality reduction algorithm, the FAGPP algorithm uses the learning of anchor graphs instead of neighbor graphs, and at the same

time, the anchor graphs in the original data space are used to guide the learning of the anchor graphs in the projection space. The FAGPP algorithm also incorporates the PCA model, making it capable of handling the clustering information and the global information of the data. The time complexity of FAGPP is $O(nmd)$, where n denotes the number of samples, m denotes the number of anchors, and d denotes the number of features of the data. In order to verify the effectiveness of the FAGPP algorithm, a large number of clustering experiments have been carried out on six image data sets with five related comparison algorithms, and the experimental results prove that the FAGPP algorithm can quickly downsize the data, and the downsized data has a better cluster structure in the projection space, and good clustering results are obtained.

Keywords: Dimensionality reduction; Clustering; Anchor; Linear projection; Instance penalty

目 录

1 绪论

1.1 研究背景

在当今互联网快速发展的时代，数据以超出想象的速度在增长，这种增长主要体现在数据样本的数量和维度，如基因数据、气象数据、图像数据和军事信息数据等[1,2]。在高维数据中，每个样本都是由多个特征组成，这些特征之间可能存在一定的关联性，也可能是冗余或不相关的。高维数据中冗余的特征信息不但影响模型的性能，而且徒增计算成本[3]。此外，高维数据中比较关键的一些特征往往决定一个样本的类别，例如医学领域中一些特殊基因的特征决定了某种特殊的疾病，而冗余的基因特征会影响医生对患者病情的诊断[4,5]。面对海量的高维数据，如何高效地剔除数据的冗余特征并从中挖掘出有价值的信息日益成为人们关注的焦点。

机器学习是人工智能中最受关注的研究领域之一，同时也是计算机科学中最活跃的研究分支之一。机器学习领域不断地涌现出新理论和新方法，这些新理论和方法不仅在计算机科学的众多领域中大放异彩，还能为人工智能技术和其他学科的交叉融合提供重要支撑，为新工科、新医科、新农科、新文科的建设助力[6,7]。数据挖掘是人工智能应用领域研究的热点，通过计算机技术和机器学习等方法从大量的原始数据中挖掘出关键的信息，用于预测、决策支持、优化等领域[8]。为了挖掘高维数据中潜在的有价值信息，需要针对数据的结构和不同任务设计不同的数据分析方法。常见的数据分析方法包含有监督学习、半监督学习和无监督学习。

聚类分析可以帮助揭示数据的内在结构，识别出数据中的自然分组，对于数据挖掘、模式识别和机器学习等领域至关重要[9]。聚类分析是一种挖掘数据潜在结构和关联特征信息的无监督学习方法，其依据数据的特征和分布特性将一组给定的无标签样本划分到若干个互不重叠的类簇中。聚类分析的目的是使得相同类簇中样本间的相似度尽可能高，而不相同类簇中样本间的相似度尽可能低。通过聚类分析观察每个子集的特征，进而充分挖掘数据的潜在信息和不同子集之间存在的关联。层次聚类[10]（Hierarchical Clustering，HC）是生物学

家和社会科学家最早使用的聚类方法，而聚类分析则成为统计多变量分析的一个重要分支[11]。常见的聚类算法有 K 均值聚类[12]（*k*-means）、模糊 C 均值聚类[13]（Fuzzy c-means Clustering，FCM）和密度峰值聚类[14]（Density Peaks Clustering，DPC）等。*k*-means 算法是一种硬聚类算法，其要求数据集中的每个样本严格属于某一特定的类簇。相反，FCM 算法是一种软聚类算法，其为每个样本和类簇中心之间赋予一个隶属度值，用于表示样本属于不同类簇的程度，隶属度值越大说明该样本越可能属于该类簇。DPC 算法是一种基于密度的聚类算法，DPC 算法同时考虑了样本的密度和样本间距离的关系，通过定义密度峰值来确定数据的类簇中心。

由于高维数据中欧式距离度量的失效以及数据样本过于稀疏等问题[15,16]，传统的聚类方法在高维数据中难以获得好的性能，这种现象称为“维度灾难”。如果直接对高维数据进行聚类，得到的类簇往往难以表达和解释，因为高维数据空间中的类簇不易被人理解和可视化[17,18]。高维数据聚类算法是数据科学技术的重要组成部分之一，研究高维数据聚类算法可以为数据科学技术提供新的思路和方法，促进数据科学技术的发展和应用。如何有效对高维数据进行聚类是需要及时解决的一个重要问题。

1.2 研究现状

高维数据包含大量的冗余特征和噪声信息，其有意义的类簇结构往往嵌入在低维子空间中，因此采用合适的降维方法对高维数据进行维数约简成为提高模型聚类效果的一个关键步骤。本文针对线性投影的高维数据聚类算法进行研究，重点对降维算法和高维数据聚类算法进行研究。

1.2.1 降维算法研究现状

降维作为处理大规模数据最有效、最直接的方法之一，近年来吸引了许多研究人员的注意。一些研究者提出通过减少高维数据的维度来缓解“维数灾难”的问题。常见的降维方法有特征选择（Feature Selection）和特征提取（Feature Extraction）。特征选择是从原始大量冗余的特征中选择最能反映类别统计的特征构成低维的特征子集，比如选择高光谱图像中的光谱纹理、颜色和形状等特

征[19,20]。特征选择通常包括个主要步骤：特征子集生成、特征子集评估、停止准则和结果验证[21]。其中，特征子集的生成和评估是最重要的，它们决定了特征选择的结果。根据搜索策略的不同，特征选择算法可分为穷举搜索[22]、启发式搜索和随机搜索。穷举搜索策略的核心是首先定义一个单调准则函数，然后从原始特征集开始，每次搜索消除一个特征，直到满足停止准则验证生成的最优特征子集。穷举搜索的策略可以保证给定评价准则下的最优特征子集，但是其计算效率较低。随机搜索策略通过随机生成特征子集，进行序列正向或反向的方法排列避免生成局部最优解。随机搜索策略在数据充足且循环次数足够大时可以得到近似最优解，但是该算法存在不确定的因素，难以实现模型的复现。启发式搜索策略兼顾获得最优特征子集和算法复杂度两个目标，其中顺序浮动选择[23]（Sequential Floating Selection）是启发式搜索中代表性的方法之一。特征提取是通过线性或非线性的变换将数据从原始特征空间映射到低维的空间中，并在这个低维的空间中尽可能多地保留原始数据的基本特征。特征选择和特征提取都是优化函数的过程，特征选择是选择一组满足标准函数的变量，其中包含最大的原始信息，而特征提取是搜索最优的数据转换来保留原始数据的特征信息。相较于特征选择，特征提取在降维过程中能够更好地保留数据的重要信息，提升数据的表征能力，从而更有效地提升模型的性能。

常见的特征选择算法一般分为两类，线性降维算法和非线性降维算法。线性降维算法可以将高维数据通过线性变换映射到低维空间，低维数据空间中样本的每个特征都是原始数据空间特征的线性组合。常见的线性降维算法有主成分分析[24]（Principal Component Analysis，PCA）、线性判别分析[25]（Linear Discriminant Analysis，LDA）和邻域保留嵌入[26](Neighborhood Preserving Embedding，NPE)等。其中，PCA 是一个经典的线性降维算法，在人脸识别[27,28]、金融管理[29]等领域得到了广泛的应用。PCA 通过构造一组正交基函数以获得数据分布的最大方差方向，尽量使投影到低维数据空间中的特征互不相关。然而 PCA 是建立在可投影在线性子空间中的假设之上，在处理非线性的数据时，PCA 的降维效果不佳。此外，PCA 还忽略了原始数据的局部几何结构特征，从原始数据中找到最紧凑的数据描述非常困难。LDA 是一个经典的线性降维算法，其基本思想将原始空间的数据投影到一条直线上，使得相同类别的样本在该直

线上的投影点尽可能接近，不同类别的样本在该直线上的投影点尽可能远。在处理现实场景中，面对大量的非线性数据，通过需要非线性的映射函数才能准确地找到原始数据的低维嵌入。非线性的降维方法考虑原始数据空间的非线性流形结构，具有处理一些复杂非线性数据的能力。一系列非线性降维算法被提出，包括核 PCA[30]（Kernel PCA, KPCA）、等轴测特征映射[31,32]（Isometric Feature Mapping, ISOMAP）和局部线性嵌入[33]（Locally Linear Embedding, LLE）。KPCA 算法保留了数据在高维空间中的非线性结构，其核心思想是利用核技巧（Kernel Trick）将原始特征空间中的数据映射到一个高维的核特征空间，然后在该高维空间中使用 PCA 算法来实现降维任务。ISOMAP 是一个经典的流形学习算法，它认为高维数据空间中两样本点的直线距离在低维嵌入流行上不可达，低维嵌入流行上两点之间的距离是“测地线”的距离。然而，这些非线性降维算法也有一些缺陷：（1）非线性降维算法的可解释性较低，只能获得训练数据的低维特征，无法提供高维数据和低维数据之间的明确特征映射；（2）非线性降维算法通常需要选择一个映射函数，但是如果映射函数过于复杂，有可能出现过拟合的问题；（3）计算过程过于复杂且耗时，不适用大规模数据集且难以处理样本外的数据。

机器学习中的许多问题可以通过图来表达，图不但可以反映两个样本之间的关系，还可以表示数据的一些统计或几何特征[34,35]。近年来，一些基于图的降维算法被提出。拉普拉斯特征映射（Laplacian Eigenmaps，LE）提出了一种由几何学驱动的用于表示高维数据的算法。LE 提供了一个计算上非常高效的方法，并具有保留数据局部关系的特点。在 LE 的基础上，局部保留投影算法[36]（Locality Preserving Projections，LPP）被提出。LPP 算法是一种基于图的线性降维算法。LPP 首先利用原始数据中样本的局部邻域信息构造邻接图，旨在低维空间中样本仍能保持这种局部关系。Yan 等人[37]认为大多数基于图的降维算法首先需要通过输入的数据或一些先验知识构建图，然后基于图的信息学习数据的低维表示，这使得模型的降维效果在很大程度上依赖于构建的图在多大程度上发现高维数据的潜在结构和分布。同理 LPP 预先构建邻接图的质量决定模型的性能。在高维空间中，由于“维数灾难”问题，LPP 利用样本间的欧式距离构造邻接图的质量难以得到保证。此外 LPP 的降维过程和邻接图的构建过程是

分开进行的，经常会出现构造出来的邻接图并不适合于随后的降维任务[38,39]。为了解决这一问题，研究人员提出了自适应学习图的降维算法，如图优化局部保持投影[40]（Graph Optimization Local Preserving Projection，GOLPP）和用于降维的联合图优化与投影学习[41]（Joint Graph Optimization and Projection Learning for Dimensionality Reduction，JGOPL）。GOLPP 和 JGOPL 融合了降维和图学习过程，在迭代优化目标函数的同时完成邻接图的学习，动态地捕获邻接图结构。

LPP 和 GOLPP 利用样本的局部邻居信息构建邻接矩阵，时间复杂度不低于 $O(n^2d)$，其中 n 表示数据集的样本数量，d 表示数据集的维度。显然 LPP 和 GOLPP 的时间复杂度过高，不适用于大的数据集。为了加速图的构造，研究者提出了锚点图[42-47]（Anchor Graph）方法。与 LPP 算法利用样本的局部信息构造邻接图的思想类似，局部锚点图保留投影[45]（Local Preserving Projection based on Anchor Graph, AGLPP）利用锚点与数据样本的局部关系快速地构建锚点图结构，期望在原始数据空间中构造的锚点图在低维空间中仍然能保留。AGLPP 利用锚点图的隶属度矩阵估计原始数据的隶属度矩阵，其时间复杂度为 $O(nmd)$，其中 m 表示构造的锚点数量。AGLPP 能高效地处理大数据集，但是其需要在进行降维任务之前完成锚点图的构建，锚点图的质量决定了算法降维的性能。

1.2.2 高维数据聚类算法研究现状

如何使用聚类算法挖掘高维数据中潜在的信息是一个研究热点。高维数据聚类是指对高维数据进行聚类分析，计算数据样本间的相似性并将其分为不同的类簇。因为“维数灾难”等问题，传统的聚类算法无法有效地处理高维数据[48]。此外由于低维空间的样本密度相较于高维空间大幅度提高，很容易计算样本间的欧式距离，因此需要将降维算法与聚类算法结合来处理高维数据聚类问题，从而提高数据处理的效率和准确度。

传统的高维数据聚类算法采用两阶段的策略，即先执行降维任务后执行聚类任务，例如首先利用 PCA 算法对原始数据进行降维，然后利用 *k*-means 算法

进行聚类学习[49]。谱聚类[50]（Spectral Clustering）算法如 ratio cut 算法[51]和 min-max cut 算法[52]，其本质上也是两阶段的高维数据聚类算法，其主要包含两步：（1）学习高维数据的低维特征表示；（2）对低维数据执行 *k*-means 算法。有学者认为这种两阶段的策略割裂了聚类任务和降维任务的联系，每一阶段优化不同的目标函数，难以保证在第一阶段降维后的数据仍然适合随后的聚类任务，因此需要融合这两个阶段来处理高维数据[53]。简化 *k* 均值聚类分析[54]（Reduced *k*-means Analysis，RKM）和因子 *k* 均值聚类分析[55]（Factorial *k*-means Analysis，FKM）串联了聚类任务与降维任务，在某些情况下取得较好的聚类效果，然而如果原始数据中大量存在与类簇结构无关的特征，这两种方法很可能无法准确地识别数据的类簇结构。为了解决这一问题，研究者提出了带有概率邻域的投影 FCM 算法[56]（Projected Fuzzy c-means with Probabilistic Neighbors，PFCM），PFCM 在降维的过程中利用数据的局部信息自适应地学习邻接图，该算法将聚类任务和降维任务统一到一个目标函数中，提高模型的聚类性能。然而，PFCM 需要计算数据样本间的相似度来构建邻接图，其时间复杂度不低于$O(n^2d)$，不适用于大的数据集。

现有的一些基于图的降维算法依赖欧式距离来构造邻接图，这些方法对原始数据中的噪声样本比较敏感，如局部投影模糊 C 均值聚类算法[57]（Projected Fuzzy c-means Clustering with Locality Preservation, LPFCM）和 LPP。为了有效地识别和处理数据中的噪声样本，研究者提出了利用图嵌入正则化的广义最小二乘法降维算法[58]（A Generalized Least-squares Approach Regularized with Graph Embedding for Dimensionality Reduction，GLSRGE）。GLSRGE 采用广义最小二乘方法考虑了全局数据分布，并为每个样本分配一个实例惩罚，实现了数据的局部结构和数据的全局结构保持平衡。KaUDDR[59]（Kernel Alignment Unsupervised Discriminative Dimensionality Reduction）将自适应图的学习和特征学习集成到一个联合学习框架中，对噪声样本具有鲁棒性，并能获得数据稳定的内在结构表示。JGOPL 采用基于$l_{21-norm}$的距离度量来求解目标函数，提高了算法处理异常样本的鲁棒性。为了更准确识别每个样本的局部邻居，缓解噪声样本和冗余特征的负面影响，Wang 等人[60]提出了局部自适应保持投影（Locality Adaptive Preserving Projections for Linear Dimensionality Reduction,

LAPP），LAPP 首先采用由粗到细的策略迭代获得最优的低维子空间，然后在最优子空间中自适应地确定样本的局部邻居。虽然 GLSRGE、JGOPL 和 LAPP 都采取一些措施处理噪声样本，但是这些算法没有考虑到数据的类簇信息，在对高维数据进行聚类时仍然采用两阶段的策略。

1.3 研究内容

高维数据聚类研究是数据挖掘和机器学习领域的重要研究课题，如何准确高效地获得高维数据的低维特征表示并与聚类模型融合是比较热门的研究方向。研究线性投影的高维数据聚类算法，开发出更加高效、准确的方法来处理和分析高维数据，解决“维度灾难”带来的问题，在各种实际应用中获得更好的性能和更深入的洞见。通过以上分析，可以发现众多学者针对高维数据聚类中存在的问题提出了许多算法。本文针对现阶段高维数据聚类算法存在的一些问题，如：算法的时间复杂度过高、算法对噪声样本敏感等问题，提出了一些解决思路。具体来说，本文主要研究内容包括以下两个方面：

（1）提出了带有实例惩罚的投影模糊 C 均值聚类算法（PCIP）。为了处理高维数据中的噪声样本，PCIP 首先基于原始数据的分布信息构建一个实例惩罚矩阵，为每一个样本分配相应的实例惩罚系数。异常分数越大的样本越可能成为异常样本，其惩罚系数就越小，从而降低噪声样本对模型的影响。其次，PCIP 算法将 FCM 算法和 PCA 算法融合在一起，构建一个统一的目标函数，使得模型在降维的过程中迭代地优化投影矩阵和隶属度矩阵。本文采用交替优化的方法对 PCIP 算法进行优化求解，并对 PCIP 算法进行了算法的时间复杂度分析和收敛性分析。在 10 个图像数据集上与 7 个相关的对比算法进行了大量的实验，包括不同维度分析实验、消融实验、运行时间实验、参数分析实验和收敛性分析实验，实验结果证明了 PCIP 算法的有效性。

（2）提出了快速锚点图保持投影算法（FAGPP）。FAGPP 算法首先利用原始数据空间中样本的信息构建锚点图结构，在迭代优化模型的同时学习锚点图来替换邻接图，期望在低维数据空间中仍然能保留这种锚点图结构。FAGPP 融合了类 FCM 的聚类模型和 PCA 算法，使其不但能学习数据的类簇结构还能捕获数据的全局信息。本文对 FAGPP 算法进行了算法的收敛性分析和时间复杂度

分析，其时间复杂度与样本的数量线性相关，可以高效处理的大的数据集。在 6 个图像数据集上与 5 个相关的对比算法进行了大量的实验，实验结果表明 FAGPP 算法能快速地学习锚点隶属度矩阵和投影矩阵，在低维子空间中获得良好的类簇结构。

1.4 论文组织结构

本文共有五个章节，第三章和第四章为主要研究内容。针对现阶段聚类算法处理高维数据时存在的一些问题提出了相应的解决方案。下面对论文的组织结构进行简单介绍。

第一章：绪论。首先介绍本文的研究背景，阐述研究高维数据聚类算法的重要意义。其次，总结现有算法的研究进展，包括降维算法和高维数据聚类算法的研究现状。最后概括地介绍本文的研究内容和论文的组织结构。

第二章：预备知识。首先介绍论文所使用的符号及说明。然后简单地介绍与本文相关的一些算法包括 K 均值聚类、模糊 C 均值聚类、局部保持投影、孤立森林、主成分分析和基于平衡k-means 的分层k-means 算法。

第三章：提出带有实例惩罚的投影模糊 C 均值聚类算法（PCIP），首先介绍实例惩罚矩阵的概念，详细描述了模型的构建和求解过程。其次对 PCIP 算法进行了收敛性分析和时间复杂度分析。在 10 个图像数据集上与 7 个相关的对比算法进行聚类实验来证明 PCIP 算法的有效性。

第四章：提出快速锚点图保持投影算法（FAGPP），首先介绍锚点的概念和模型的构建思路和求解过程。其次对模型进行了收敛性分析和时间复杂度分析。在 6 个图像数据集上与 5 个相关的对比算法进行实验包括不同维度的实验分析、参数分析实验、收敛性分析实验、时间运行实验等来证明 FAGPP 算法的有效性。

第五章：分析和总结了全文的研究工作，展望未来的研究方向和发展趋势。

2 预备知识

本章主要介绍与本文所提算法相关的一些算法包括聚类算法、降维算法和异常检测算法，并对本文所使用的一些符号进行简要说明。

2.1 符号及说明

本文所使用的符号和相关说明如表 2.1 所示。

表 2.1 本文使用的符号说明

符号	说明
$X=[x_1, x_2, \ldots, x_n] \in R^{d\times n}$	X 是一个数据集包含 n 个样本和 d 个特征 x_i 是数据集 X 中第 i 个样本
$P=[p_{\bullet 1}, p_{\bullet 1}, \ldots, p_{\bullet n}] \in R^{c\times n}$	P 是一个隶属度矩阵， $p_{\bullet i}$ 表示样本 x_i 隶属不同类簇的集合， p_{ki} 表示样本 x_i 属于类簇 k 的隶属度
$W \in R^{d\times \tilde{d}}$	W 是一个投影矩阵， $\tilde{d}$ 是降维后的维度
$V=[v_1, v_2, \ldots, v_c] \in R^{d\times c}$	V 是原始空间的类簇中心， v_k 表示原始空间中第 k 个类簇中心
$Z=[z_1, z_2, \ldots, z_c] \in R^{\tilde{d}\times c}$	Z 是投影子空间的类簇中心， z_k 表示投影子空间中第 k 个类簇中心。
$F \in R^{n\times n}$	F 表示实例惩罚矩阵， f_{ii} 是 F 的对角矩阵，表示样本 x_i 的实例惩罚系数
$O=[o_1, o_2, \ldots, o_m] \in R^{d\times m}$	O 表示原始空间的锚点， o_k 表示原始空间中对 k 个锚点
$\tilde{O}=[\tilde{o}_1, \tilde{o}_2, \ldots, \tilde{o}_m] \in R^{\tilde{d}\times m}$	$\tilde{O}$ 表示投影空间的锚点， $\tilde{o}_k$ 表示投影空间中对 k 个锚点
$J=[j_{\bullet 1}, j_{\bullet 1}, \ldots, j_{\bullet n}] \in R^{m\times n}$	J 是原始空间的锚点隶属度矩阵， j_{ki} 表示样本 x_i 属于锚点 o_k 的隶属度。
$H=[h_{\bullet 1}, h_{\bullet 1}, \ldots, h_{\bullet n}] \in R^{m\times n}$	H 是投影空间的锚点隶属度矩阵， h_{ki} 表示投影空间样本 $W^T x_i$ 属于投影空间锚点 $\tilde{o}_k$ 的隶属度。

2.2K 均值聚类（*k*-means）

k-means 聚类算法是一种经典的无监督学习算法，因其简单高效被广泛地应用于许多场景中[61,62]。假设数据集 $X=[x_1,x_2,\ldots,x_n]$ 有 n 个样本，*k*-means 算法的目标是将所有样本划分成互不重叠的 k 个类簇 G_1，G_2，…，G_k。令 $V=\{v_1,v_2,\ldots,v_k\}$，其中 v_j 是类簇 G_j 的第 j 个类簇中心。令 $\chi=[\chi_{ij}]_{n\times k}$，其中 χ_{ij} 是一个二元变量，其中 $\chi_{ij}\in\{0,1\}$。当样本 x_i 属于类簇 G_j 时，$\chi_{ij}=1$，反之 $\chi_{ij}=0$。在算法的每次迭代过程中，*k*-means 计算每个样本距离所有类簇中心的欧式距离并对距离进行排序，将该样本划分到距离最近的类簇中心。*k*-means 的目标函数如下：

$$\begin{aligned} &J(\chi,V)=\sum_{i=1}^{n}\sum_{j=1}^{k}\chi_{ij}\left\|x_i-v_j\right\|_2^2 \\ &s.t.\ \chi_{ij}=\begin{cases}1, & \left\|x_i-v_j\right\|_2^2=\min\limits_{j\in[1,k]}\left\|x_i-v_j\right\|_2^2\\ 0, & otherwise\end{cases}\end{aligned} \tag{2-1}$$

公式(2-1)中 $\left\|x_i-v_j\right\|_2^2$ 表示样本 x_i 与类簇中心 v_j 之间的欧式距离。在每次迭代中需要更新类簇的中心来优化目标函数 $J(\chi,V)$，第 k 个类簇中心 v_k 计算如下：

$$v_k=\frac{\sum_{i=1}^{n}\chi_{ik}x_i}{\sum_{i=1}^{n}\chi_{ik}} \tag{2-2}$$

2.3 主成分分析（PCA）

PCA[24]算法是一个经典的数据降维技术，在科学和工程领域得到广泛的应用。其通过线性变换将高维数据映射到低维空间中，以发现数据中的主要结构和关系。PCA 在最小二乘法的框架下寻求一个投影矩阵 $W\in R^{d\times\tilde{d}}$，通过最小化投影数据点与原始数据点之间的重构误差，从而寻求数据的全局低维子空间。因为 PCA 是一种线性变换的算法，无法捕捉到非线性的复杂数据结构。PCA 的

目标函数如下：

$$\begin{aligned} &\min \sum_{i=1}^{n} \left\| x_i - WW^T x_i \right\|_2^2 \\ &s.t.\ W^T W = I \end{aligned} \tag{2-3}$$

经过简单的计算，公式(2-3)可以如下目标函数表示：

$$\begin{aligned} &\max Tr(W^T S_t W) \\ &s.t. W^T W = I \end{aligned} \tag{2-4}$$

公式(2-4)中 S_t 表示数据集 X 的协方差矩阵，显然投影矩阵 W 可以由协方差矩阵 S_t 中最大的前 $\tilde{d}$ 个特征值对应的特征向量组成。

2.4 局部保持投影（LPP）

LPP[63]是一种针对非线性拉普拉斯特征图的线性流形学习技术。LPP 利用原始数据空间中的样本局部邻居信息构建邻接图和计算投影矩阵 W，使用 W 将原始空间的数据映射到低维子空间，期望在原始空间构建的邻接图在投影空间中得到保留。LPP 与 PCA 都是线性降维算法，PCA 通过最小化数据的重构误差来寻求全局最优投影子空间。然而，LPP 通过在投影空间中重建样本与其邻域之间的关系来保持原始数据空间的数据结构。LPP 的目标函数定义如下：

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij} \left\| y_i - y_j \right\|_2^2 = \min \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij} \left\| W^T x_i - W^T x_j \right\|_2^2 \tag{2-5}$$

公式(2-5)中 y_i 和 y_j 分别是样本 x_i 和样本 x_j 在投影空间的样本。$S = [S_{ij}] \in R^{n \times n}$ 表示邻接矩阵，S_{ij} 是原始数据空间中样本 x_i 和样本 x_j 之间的权重，两个样本之间的距离越小，其权重就越大。公式(2-5)可以化简为：

$$\begin{aligned} &\min \left(\sum_{i=1}^{n} W^T x_i D_{ii} x_i^T W - \sum_{i=1}^{n}\sum_{j=1}^{n} W^T x_i S_{ij} x_j^T W\right) \\ &= \min \left(W^T X (D - S) X^T W\right) \\ &= \min \left(W^T X L X^T W\right) \\ &s.t. W^T X D X^T W = I \end{aligned} \tag{2-6}$$

公式(2-6)中 $D_{ii}=\sum_{j=1}^{n}S_{ij}$ 是一个度矩阵， D_{ii} 越大说明样本 y_i 越重要。 $L=D-S$ 是一个拉普拉斯矩阵。

2.5 孤立森林（iForest）

孤立森林[64]（Isolation Forest，iForest）是一个快速的异常点检测算法，孤立表示样本因某种差异而相互分离。iForest 不同于传统的基于距离和基于密度的异常检测算法，它通过构建一定数量的孤立树（Isolation Tree，iTree）来孤立数据中的异常样本。iForest 认为相较于正常样本，异常样本的数量上较少且特征值差异较大。iForest 的构建主要包含 3 个步骤：（1）通过不放回采样的方式采样得到 t 个不同的子集，其中每个子集的大小为 ψ；（2）利用每个子集中样本的信息上构建 iTree；（3）构建由 t 棵 iTree 共同组成的 iForest。

iTree 从本质上看是一个二叉树，其中每个节点包含 0 个或者 2 个子节点。iTree 通过随机选择特征属性和该属性下的分裂节点递归地分裂数据。iTree 包含两种节点外部节点（External-node，eNode）和内部节点（Internal-node，iNode）。假设样本 x_i 是 iTree 中的一个节点，当 x_i 不能被分裂时，定义 x_i 为 eNode。显然当 iTree 的高度达到高度限制或者 x_i 是独立节点时，则称 x_i 为 eNode。相反地，当样本 x_i 所在的分区可以继续分裂时， x_i 定义为内部节点 iNode。

算法 1[64]描述了 iTree 构建的过程，iTree 采用随机分裂的方式对数据进行分割，即随机选择数据的一个属性，再从这个属性中随机选择一个分裂点进行分割。为了更加清晰地描述 iTree，随机生成 20 个样本，每个样本包含 2 个特征，分别为特征 x 和特征 y。样本的分布如图 2.1 所示。iTree 中节点的分裂过程如图 2.2 所示，图 2.3 表示 iTree 特定的树结构。其中 $p_1, p_2, \ldots, p_{11}$ 表示 iTree 使用 11 个不同的分裂点先后对数据进行分割。红色的实线表示算法选择特征 x 将数据分裂成左右两个部分，绿色的实线表示算法选择特征 y 将数据分裂成上下两个部分。

算法 1: *iTree(X,ch,hl)*

输入： 数据集 X，当前树的高度 ch，树的高度限制 hl

输出： ***iTree***

1. **If** $ch \le hl \quad and \quad |X| > 1$
2. 令 Q 为 X 的一个属性集合
3. 从 Q 中随机选择一个属性 q
4. 从 X 的第 q 列随机选择一个分裂节点 p，其中 $p \in (min(X(:,q)), max(X(:,q)))$
5. 将 X 分裂成两部分，$X_{left} = X(X(:,q) < p), \quad X_{right} = X(X(:,q) \ge p)$
6. **Return** ***internal - node*** $\begin{pmatrix} Left = iTree(X_{left}, ch+1, hl), \\ Right = iTree(X_{right}, ch+1, hl), \\ SplitAtt = q, SplitValue = p \end{pmatrix}$
7. **Else**
8. **Return** ***external - node***$(size = |X|)$
9. **End if**

从图 2.2 和图 2.3 可以看出：

（1）iTree 一共进行了 11 次节点分裂过程，iTree 的高度为 5。

（2）iTree 第一次分裂节点为 p_1。将样本分为两部分，左边部分为 $\{x_2, x_5, x_7, x_{13}, x_{15}, x_{16}, x_{20}\}$，右边部分 $\{x_1, x_3, x_4, x_6, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{14}, x_{17}, x_{18}, x_{19}\}$。

（3）当树的高度为 3 时，样本 x_{16} 为一个外部节点，当树的高度为 4 时，样本 x_{14} 为一个 eNode。

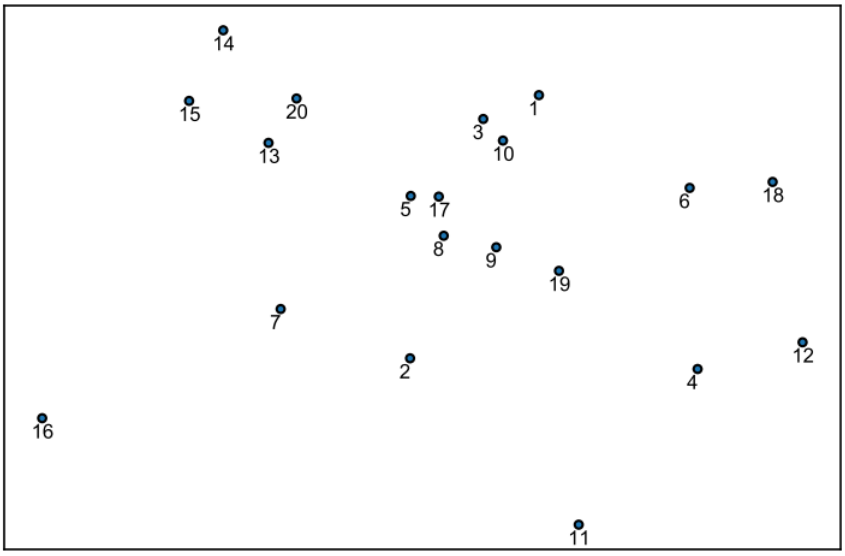


图 2.1 样本的原始分布图

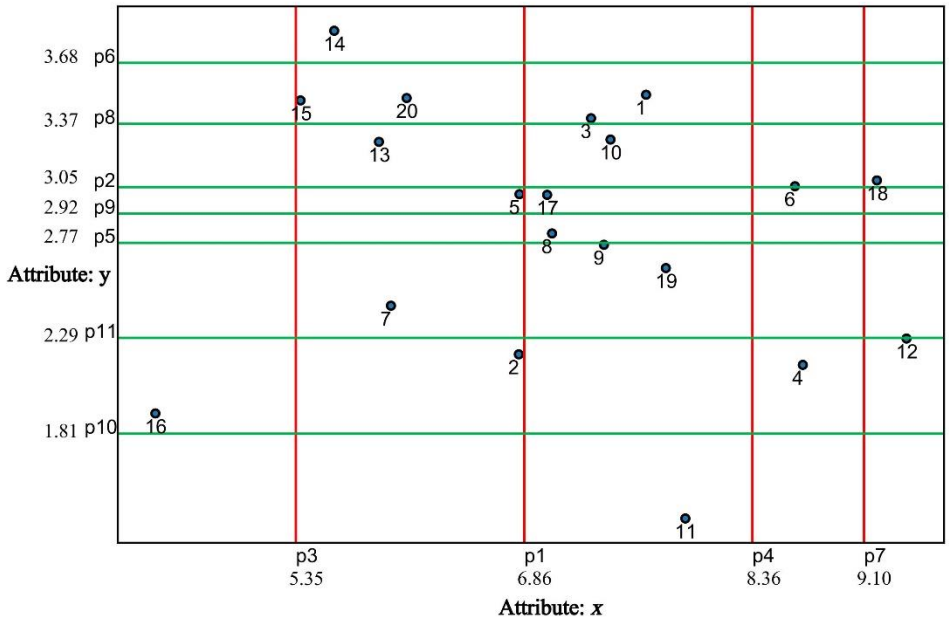


图 2.2 iTree 中节点的分裂过程

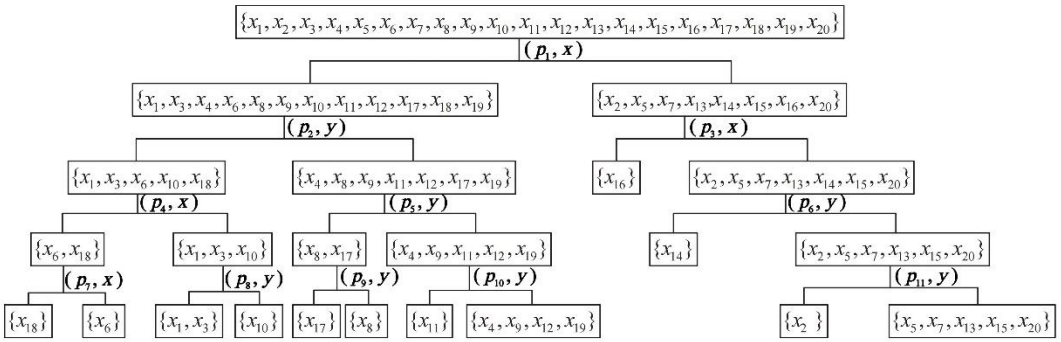


图 2.3 iTree 的树结构

iForest 由多棵 iTree 构成，其构建过程见算法 2[64]。其中 $sample(X,\psi)$ 表示将数据 X 通过随机采样的方式构造大小为 ψ 的子集。$Forest(i)$ 表示第 i 棵 iTree。

算法 2: *iForest(X,t,φ)*

输入： 数据集 X，iTree 的数量 t，采样子集的大小 φ

输出： ***Forest***

1. 初始化 ***Forest*** $=\varnothing$
2. 计算每棵 iTree 的限制高度 $hl=\log_2\varphi$
3. **For** $r=1:t$
4. $\quad$ ***Forest***$(r)=$***iTree***$($***sample***$(X,\varphi),0,hl)$
5. $\quad$ ***Forest*** $=$ ***Forest*** $\bigcup$ ***Forest***(i)
6. **End for**
7. **Return** ***Forest***

iForest 采用异常分数来判断一个样本是否为异常样本，具体来说基于 iTree 中节点的路径长度来定义异常分数。在 iTree 中，当样本 x_i 为 eNode 时，其路径长度为该节点距根节点的长度。节点的路径长度计算见算法 3[64]。鉴于 iTree 计算 eNode 的路径长度 $h(x_i)$ 与二叉查找树（Binary Search Tree，BST）[65-67]中不成功搜索的长度 $c(n)$ 相同。iForest 引入 $c(n)$ 对 $h(x_i)$ 进行归一化。样本 x_i 的异常分数 $g(x_i)$ 计算如下：

$$g(x_i) = e^{-(E(h(x_i))/c(n))} \tag{2-7}$$

在公式(2-7)中， n 为 x_i 所在子集的样本数量， $E(h(x_i))$ 为 x_i 在不同 iTree 中的平均路径长度， $c(n) = 2H(n-1) - 2(n-1)/n$ 。对于一个样本而言，其节点路径长度越小，异常分数就越大，该样本越可能为异常样本。

算法 3: ***Pathlength(x,iTree,ch)***

输入： 一个样本 x ，树的当前高度 ch ，TR 表示一棵 *iTree*

输出： 样本 x 的节点路径长度

```
1.  If TR 是一个 eNode
2.      Return ch + c(TR.size)
3.  End if
4.  令 a = TR.SplitAtt ， b = TR.SplitValue
5.  If x(:, a) ≥ b
6.      Return Pathlength(x, TR.Right, ch + 1)
7.  Else
8.      Return Pathlength(x, TR.Left, ch + 1)
9.  End if
```

2.6 模糊 C 均值聚类（FCM）

传统的 *k*-means 算法在对数据进行聚类时，仅允许样本属于一个类簇，对异常样本数据较为敏感。为克服这一缺陷，Bezdek 等人提出了著名的模糊 C 均值聚类[13](Fuzzy c-means Clustering, FCM)。与传统的 *k*-means 算法不同，FCM 允许一个样本属于多个类簇，而不是一个类簇。令 $V = [v_1, v_2, \ldots, v_c] \in R^{d \times c}$ 表示数据的类簇中心集合， $v_k \in R^{d \times 1} (k = 1, 2, 3, \cdots, c)$ 表示第 k 个类簇中心。FCM 算法的目标函数如下：

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{k=1}^{c} p_{ki}^{\alpha}\left\|x_i - v_k\right\|_2^2 \\ &s.t.\ P \geq 0, (p_{\bullet i})^T \mathbf{1}_c = 1 \end{aligned} \tag{2-8}$$

公式(2-8)中 $P=[p_{ki}]\in R^{c\times n}$ 是一个隶属度矩阵， p_{ki} 是 P 的一个元素，表示样本 x_i 属于类簇中心 v_k 的隶属度。 $p_{\bullet i}$ 表示矩阵 P 的第 i 列，约束 $(p_{\bullet i})^T \mathbf{1}_c = 1$ 表示对于每一个样本 x_i，其隶属于不同的类簇的隶属度之和为 1。约束 $P\geq 0$ 表示矩阵 P 是一个非负矩阵。模糊指数 α 影响算法目标函数的凹凸性和算法的收敛性，决定着 p_{ki}^{α} 的值，控制着类簇中心的分布和 FCM 算法的性能。在算法的迭代过程中，通过不断地优化目标函数以寻求最优的隶属度矩阵 P 和类簇中心 V 。

2.7 基于平衡*k*-means 的分层*k*-means（BKHK）

LPP 算法利用样本的局部邻居信息构建邻接图，时间复杂度不低于 $O(n^2 d)$，不适用于大的数据集。一些研究者尝试通过构建锚点图来保留数据的局部信息，常用的构造锚点图的策略有随机选择生成锚点和 *k*-means 生成锚点。随机生成锚点算法的通用性很差，而 *k*-means 生成锚点需要大量的时间成本而且不适用于多类簇的数据。基于平衡*k*-means 的分层*k*-means[68]（Balanced K-means based Hierarchical K-means，BKHK）有效地解决以上缺陷。BKHK 采用平衡二叉树结构来生成锚点，其时间复杂度为 $O(nd\log(m)t)$ 。令 $O=[o_1,o_2,\ldots,o_m]\in R^{\bar{d}\times m}$ 为原始数据空间的锚点， o_k 表示原始空间中对 k 个锚点。BKHK 的目标函数如下：

$$\begin{aligned} &\min \sum_{k=1}^{m} d(x_i,o_k) j_{ki} + \gamma \sum_{k=1}^{m} (j_{ki})^2 \\ &s.t.\ \ j_{\bullet i}^{T} 1_m = 1,\ J \geq 0 \end{aligned} \tag{2-9}$$

公式(2-9)中 $J=[j_{\bullet 1}, j_{\bullet 1},\ldots, j_{\bullet n}]\in R^{m\times n}$ 是原始数据空间的锚点隶属度矩阵， j_{ki} 表示样本 x_i 与锚点 o_k 之间的隶属度。 $d(x_i,o_k)$ 指的是样本 x_i 与锚点 o_k 之间的欧式距离， $d(x_i,o_k)=\left\|x_i-o_k\right\|_2^2$ 。对于样本 x_i ，令 $d(x_i)$ 为样本 x_i 与所有锚点之间的距离集合，并对 $d(x_i)$ 按递增顺序排序。因此，在 r 个元素为非零值的约束下，优

化问题(2-9)的解为：

$$j_{ki}=\begin{cases}\dfrac{d(x_i,o_{r+1})-d(x_i,o_k)}{\sum_{t=1}^{r}(d(x_i,o_{r+1})-d(x_i,o_t))}, & k\le r\\ 0, \quad otherwise\end{cases} \tag{2-10}$$

通过公式(2-10)可以计算得到一个稀疏的锚点隶属度矩阵 J，将数据信息、锚点信息和锚点隶属度矩阵融合在一起构建锚点的图结构。

3 带有实例惩罚的投影模糊 C 均值聚类算法

本章将详细介绍带有实例惩罚的投影模糊 C 均值聚类算法（PCIP）。为了识别数据中的异常样本，PCIP 算法首先基于 iForest 构造实例惩罚矩阵，并为每个样本分配实例惩罚系数。其次，PCIP 算法融合了 FCM 算法与 PCA 算法，构造一个统一的模型可以同时完成聚类任务和降维任务。本章还对 PCIP 算法进行了详细的时间复杂度分析和算法的收敛性分析。

3.1 实例惩罚矩阵

传统的高维聚类算法缺乏处理噪声样本的过程，对数据中的噪声样本较为敏感。iForest 利用数据的分布信息为每个样本赋予异常分数来识别数据中的噪声样本。受 iForest 思想的启发，本文考虑为每个样本分配一个实例惩罚系数来识别和处理数据中的噪声样本，减轻噪声样本对模型的影响。实例惩罚矩阵的定义如下：

定义 3.1 实例惩罚矩阵

$$F = Diag(G^{-1}) \tag{3-1}$$

公式(3-1)中异常分数矩阵 $G=[g(x_1),g(x_1),g(x_1),\cdots,g(x_n)]$，$g(x_i)$ 表示样本 x_i 的异常分数，可以由公式(2-7)计算得到。函数 $Diag(\bullet)$ 是一个可以将矩阵变为对角矩阵的函数。

显然 $F=\begin{bmatrix} f_{11} & & & \\ & f_{22} & & \\ & & \cdots & \\ & & & f_{nn} \end{bmatrix} \in R^{n\times n}$，$f_{ii}$ 是 F 的一个对角元素，它表示样本 x_i 的实例惩罚系数。如果样本 x_i 在孤立树中的路径长度越短，表示样本 x_i 越容易被孤立出来，其异常分数 $g(x_i)$ 越大，其实例惩罚系数 f_{ii} 越小。对于一个样本而言，小的实例惩罚系数表示该样本对模型的影响比较小。通过对数据集 X 每个样本分配不同的实例惩罚系数，降低异常样本对模型的影响，从而提高模型的聚类性能。算法 4 详细地描述了实例惩罚矩阵的计算过程。

算法 4: 计算实例惩罚矩阵

输入： 数据集 X，iTree 的数量 t，采样子集的大小 φ

输出： 实例惩罚矩阵 F

1. 利用算法 2 构建孤立森林 ***iForest(X,t,φ)***
2. **For** $i=1:size(X)$
3. 初始化 $E(h(x_i))=0$， $count=0$
4. **For** $r=1:t$
5. $h(x_i)=\boldsymbol{Pathlength}(x_i,\boldsymbol{Forest}(r),0)$
6. $E(h(x_i))=E(h(x_i))+h(x_i)$
7. $count=count+1$
8. **End for**
9. $E(h(x_i))=\dfrac{E(h(x_i))}{count}$
10. **End for**
11. 根据公式(2-7)计算 G
12. 根据公式(3-1)计算实例惩罚矩阵 F
13. **Return** F

3.2 模型

原始数据中的异常样本会严重影响模型的聚类结果，如何识别和处理这些异常样本从而提高模型的鲁棒性是一个研究热点。本文首先基于原始的数据分布构建实例惩罚矩阵，为每个样本分配一个实例惩罚系数，并将其与 FCM 算法相结合来降低异常样本对模型的影响。因此，本文首先设计了如下模型：

$$\begin{aligned}&\min_{P,W,Z}\sum_{i=1}^{n}f_{ii}\sum_{k=1}^{c}p_{ki}^{\alpha}\left\|W^{T}x_i-z_k\right\|_2^2\\&s.t.W^{T}W=I,(p_{\cdot i})^{T}\boldsymbol{1}_c=1,P\geq 0\end{aligned}\tag{3-2}$$

模型(3-2)中 f_{ii} 表示样本 x_i 的实例惩罚系数，α 为模糊指数，c 指的是数据类簇个数，W 是投影矩阵，z_k 表示投影空间中的第 k 个类簇中心。

为了捕获数据的全局结构，在保留数据的主要信息的同时减少噪声信息和冗余数据的影响。本文还将 PCA 算法嵌入到模型(3-2)中，因此本文最终模型如下：

$$\begin{aligned}&\min_{P,Z,W}\sum_{i=1}^{n}f_{ii}\sum_{k=1}^{c}p_{ki}^{\alpha}\left\|W^{T}x_i-z_k\right\|_2^2-\lambda Tr(W^{T}S_tW)\\&s.t.\ W^{T}W=I,(p_{\cdot i})^{T}\boldsymbol{1}_c=1,P\geq 0\end{aligned}\tag{3-3}$$

模型(3-3)中 S_t 表示数据集 X 的协方差矩阵，λ 是一个权衡参数可以调节模型中 $Tr(W^T S_t W)$ 的重要性。具体来说 λ 越大，模型(3-3)中的 $Tr(W^T S_t W)$ 越大。

3.2.1 模型优化

本文采用一种交替优化的算法求解模型(3-3)，在算法每次迭代的过程中，首先固定隶属度矩阵 P，求解投影矩阵 W 和投影空间类簇中心 Z；其次固定 W 和 Z，求解 P。模型(3-3)的优化分两步进行。

（1）固定 P，优化 W 和 Z

当 P 固定时，令 $f_{ii} p_{ki}^{\alpha} = u_{ki}$，因为模型(3-3)对簇心 z_k 没有约束，首先对 z_k 求导并令其偏导数为 0。可以得到：

$$\begin{aligned} &\frac{\partial(\sum_{i=1}^{n}\sum_{k=1}^{c} u_{ki} \left\| W^T x_i - z_k \right\|_2^2 - \lambda Tr(W^T X X^T W))}{\partial z_k} \\ &= \sum_{i=1}^{n} u_{ki}(-2W^T x_i + 2z_k) = 0 \end{aligned} \tag{3-4}$$

根据公式(3-4)，可以得到：

$$z_k = \frac{\sum_{i=1}^{n} u_{ki} W^T x_i}{\sum_{i=1}^{n} u_{ki}} = W^T \frac{\sum_{i=1}^{n} u_{ki} x_i}{\sum_{i=1}^{n} u_{ki}} \tag{3-5}$$

令 $v_k = \dfrac{\sum_{i=1}^{n} u_{ki} x_i}{\sum_{i=1}^{n} u_{ki}}$，则 $z_k = W^T v_k$。可以看出 $V = [v_1, v_2, \ldots, v_c] \in R^{d \times c}$ 为原始数据的类簇中心。基于此，模型(3-5)的第一部分可以表示为：

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{k=1}^{c} u_{ki} \left\| W^T x_i - W^T v_k \right\|_2^2 \\ &s.t.\ W^T W = I, p_{\cdot i}^{\ T} \mathbf{1}_c = 1, P \geq 0 \end{aligned} \tag{3-6}$$

为了方便计算，将公式(3-6)转换为：

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\left\|W^Tx_i-W^Tv_k\right\|_2^2\\
&=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}(W^Tx_i-W^Tv_k)^T(W^Tx_i-W^Tv_k)\\
&=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\left(x_i^TW-v_k^TW\right)(W^Tx_i-W^Tv_k)\\
&=\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\left(x_i^TWW^Tx_i-v_k^TWW^Tx_i-x_i^TWW^Tv_k+v_k^TWW^Tv_k\right)\\
&=\sum_{i=1}^{n}\left(A_{ii}^xx_iWW^Tx_i^{\mathrm{T}}\right)-2\sum_{i=1}^{n}\sum_{k=1}^{c}(u_{ki}v_k^TWW^Tx_i)+\sum_{k=1}^{c}(A_{kk}^yv_k^TWW^Tv_k)\\
&=Tr(W^T(XA^xX^T-2VUX^T+VA^yV^T)W)
\end{aligned}\tag{3-7}$$

其中 $A^x\in R^{n\times n}$ 和 $A^y\in R^{c\times c}$ 为两个不同的对角矩阵。A^x 的第 i 个对角元素为 $A_{ii}^x=\sum_{k=1}^{c}u_{ki}$。$A^y$ 的第 k 个对角元素为 $A_{kk}^y=\sum_{i=1}^{n}u_{ki}$。

令 $M=XA^xX^T-2VUX^T+VA^yV^T$，$Q=(M+M^T)/2$ 则：

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{k=1}^{c}u_{ki}\left\|W^Tx_i-W^Tv_k\right\|_2^2\\
&=Tr(W^TMW)=Tr(W^T\left(\frac{M+M^T}{2}\right)W)\\
&=Tr(W^TQW)
\end{aligned}\tag{3-8}$$

根据公式(3-8)，模型(3-3)转变为：

$$\begin{aligned}
&\min Tr(W^T\mathrm{Q}W)-Tr(W^T\lambda XX^TW)\\
&=\min Tr(W^T\mathrm{Q}W)-Tr(W^T\lambda S_tW)\\
&=\min Tr(W^T(Q-\lambda S_t)W)
\end{aligned}\tag{3-9}$$

其中 S_t 为 X 的协方差。在 $W^TW=I$ 的约束下，通过求解

$$(M-\lambda S_t)W=\beta W\tag{3-10}$$

得到投影矩阵 W 为 β 中 $\tilde{d}$ 个最小特征值对应的特征向量。

（2）固定 W 和 Z，优化 P

令 $q_{ki}=f_{ii}\left\|W^Tx_i-z_k\right\|_2^2$，则模型(3-3)的第一部分可以表示为：

$$\begin{aligned}
&\min\sum_{i=1}^{n}\sum_{k=1}^{c}p_{ki}^{\alpha}q_{ki}\\
&s,t.\ p_{\cdot i}^{\ T}1_c=1,P\geq 0
\end{aligned}\tag{3-11}$$

根据拉格朗日乘数法，可以得到：

$$J(p_{\cdot i},\alpha)=\sum_{k=1}^{c}q_{ki}{p_{ki}}^{\alpha}-\gamma(\sum_{k=1}^{c}p_{ki}-1) \tag{3-12}$$

令 ${p_{ki}}^{*}$ 表示 p_{ki} 的最优解，γ^{*} 表示 γ 的最优解。分别对 ${p_{ki}}^{*}$ 和 γ^{*} 求偏导，可以得到：

$$\begin{cases}\dfrac{\partial J(p_{\cdot i},\alpha)}{\partial {p_{ki}}^{*}}=\alpha q_{ki}({p_{ki}}^{*})^{\alpha-1}-\gamma^{*}=0\\ \dfrac{\partial J(p_{\cdot i},\alpha)}{\partial \gamma^{*}}=-(\sum_{k=1}^{c}{p_{ki}}^{*}-1)=0\end{cases} \tag{3-13}$$

通过求解公式(3-13)，可以得到 p_{ki} 的最优解如下：

$${p_{ki}}^{*}=\frac{{q_{ki}}^{\frac{1}{1-\alpha}}}{\sum_{j=1}^{c}{q_{ji}}^{\frac{1}{1-\alpha}}} \tag{3-14}$$

将公式(3-14)代入公式(3-11)可以得到：

$$\begin{aligned}\min\sum_{i=1}^{n}\sum_{k=1}^{c}p_{ki}^{\alpha}q_{ki}&=\min\sum_{i=1}^{n}\sum_{k=1}^{c}\left(\frac{{q_{ki}}^{\frac{1}{1-\alpha}}}{\sum_{j=1}^{c}{q_{ji}}^{\frac{1}{1-\alpha}}}\right)^{\alpha}q_{ki}\\&=\min\sum_{i=1}^{n}\sum_{k=1}^{c}\frac{{q_{ki}}^{\frac{1}{1-\alpha}}}{\sum_{j=1}^{c}{q_{ji}}^{\frac{\alpha}{1-\alpha}}}\end{aligned} \tag{3-15}$$

从公式(3-15)中可以看出随着 q_{ki} 的减少，$\sum_{i=1}^{n}\sum_{k=1}^{c}p_{ki}^{\alpha}q_{ki}$ 逐渐降低。

3.2.2 算法描述

本文的 3.2.1 节利用交替优化的方法求解投影矩阵 W，类簇中心 Z 和隶属度矩阵 P。PCIP 算法是一个迭代的算法，在每次迭代的过程中通过优化投影矩阵 W 和隶属度矩阵 P，进而降低目标函数的值。当目标函数值收敛时，PCIP 算法获得最优解。PCIP 算法的伪代码如下：

算法 5: PCIP 算法

输入： 数据集 $\boldsymbol{X}$

参数： 模糊指数 $\boldsymbol{\alpha}$，权衡参数 $\boldsymbol{\lambda}$，投影空间维度 $\tilde{d}$，构建 *iTree* 的数量 $\boldsymbol{t}$，抽样子集的大小 φ

输出： 投影矩阵 $\boldsymbol{W}$ 和隶属度矩阵 $\boldsymbol{P}$

1. 利用约束 $P \geq 0$ 和 $(p_{\cdot i})^T \mathbf{1}_c = 1$ 初始化隶属度矩阵 $\boldsymbol{P}$
2. 计算 $\boldsymbol{X}$ 的协方差矩阵 S_t
3. 利用算法 4 计算数据的实例惩罚矩阵 $\boldsymbol{F} = \boldsymbol{CIPM}(X, t, \varphi)$
4. **When not convergence**
5. 计算矩阵 $\boldsymbol{U} = \boldsymbol{PF}$
6. **For** each $v_k \in \boldsymbol{V}$
7. $v_k = \sum_{i=1}^{n} u_{ki} x_i / \sum_{i=1}^{n} u_{ki}$
8. **End For**
9. 计算 $\boldsymbol{M} = XA^x X^T - 2VUX^T + VA^y V^T$
10. 计算 $\boldsymbol{Q} = (M + M^T)/2$
11. 利用公式(3-10)更新 $\boldsymbol{W}$
12. 计算 $z_k = W^T v_k$, $q_{ki} = f_{ii} \left\| W^T x_i - z_k \right\|_2^2$
13. 利用公式(3-14)计算 p_{ki}，更新 $\boldsymbol{P}$
14. 计算模型(3-3)的目标函数值
15. **Until convergence**
16. **Return** $\boldsymbol{W}$ 和 $\boldsymbol{P}$

3.3 收敛性分析

算法的收敛性分析在算法设计和优化中起着重要的作用，提供了对算法性能和正确性的验证。在本节将对 PCIP 算法的收敛性进行证明，采用交替优化的方法对 PCIP 算法的目标函数进行求解。模型(3-3)的目标函数为：

$$\begin{aligned} &J(Z,P,W) = \min \sum_{i=1}^{n} f_{ii} \sum_{k=1}^{c} p_{ki}^{\alpha} \left\| W^T x_i - z_k \right\|_2^2 - \lambda Tr(W^T X X^T W) \\ &s,t. W^T W = I, p_{\cdot i}^{\ T} \mathbf{1}_c = 1, P \geq 0 \end{aligned} \tag{3-16}$$

显然，公式(3-16)中目标函数的第一部分 $\sum_{i=1}^{n} f_{ii} \sum_{k=1}^{c} p_{ki}^{\alpha} \left\| W^T x_i - z_k \right\|_2^2 \geq 0$。令 δ_i 为 XX^T 的第 i 个最小的特征值，可以得到 $-\lambda Tr(W^T X X^T W) \geq -\lambda \sum_{i=1}^{\tilde{d}} \delta_i$。因此，可以

得到 $J(P,W,Z) \geq -\lambda\sum_{i=1}^{\tilde{d}}\delta_i$。可以看出，目标函数值有一个下界 $-\lambda\sum_{i=1}^{\tilde{d}}\delta_i$。

假设模型(3-3)在第 i 次迭代后，其最优解为 $W^{(i)}$、$Z^{(i)}$ 和 $P^{(i)}$，公式(3-16)最优值为 $J(Z^{(i)},P^{(i)},W^{(i)})$。模型(3-3)采用交替优化的方法对目标函数进行求解，在每次迭代有两个步骤：（1）固定 $P^{(i)}$，优化 $W^{(i)}$ 和 $Z^{(i)}$；（2）固定 $W^{(i)}$ 和 $Z^{(i)}$，优化 $P^{(i)}$。

在第一步，根据公式(3-4)，$Z^{(i)}$ 的最优解可以用偏导数来求解。因此可以得到：

$$J(Z^{(i+1)},P^{(i)},W^{(i)}) \leq J(Z^{(i)},P^{(i)},W^{(i)}) \tag{3-17}$$

在约束 $W^TW=I$ 的情况下，可以得到：

$$J(Z^{(i+1)},P^{(i)},W^{(i+1)}) \leq J(Z^{(i+1)},P^{(i)},W^{(i)}) \tag{3-18}$$

结合公式(3-17)和公式(3-18)，可以得到：

$$J(Z^{(i+1)},P^{(i)},W^{(i+1)}) \leq J(Z^{(i)},P^{(i)},W^{(i)}) \tag{3-19}$$

在第二步，通过拉格朗日数乘法求解 $P^{(i)}$，可以得到：

$$J(Z^{(i+1)},P^{(i+1)},W^{(i+1)}) \leq J(Z^{(i+1)},P^{(i)},W^{(i+1)}) \tag{3-20}$$

结合公式(3-19)和公式(3-20)，最终得到：

$$J(Z^{(i+1)},P^{(i+1)},W^{(i+1)}) \leq J(Z^{(i)},P^{(i)},W^{(i)}) \tag{3-21}$$

由不等式(3-21)可以看出，目标函数值在每次迭代中减少。此外，得到目标函数值有一个下界 $-\lambda\sum_{i=1}^{\tilde{d}}\delta_i$。因此目标函数在迭代过程中将会收敛并获得局部最优值。

3.4 时间复杂度分析

假设数据集 X 有 n 个样本，有 d 个特征，有 c 个类簇。令 t 为构建 iTree 的棵数，φ 为每个采样子集的大小，则构建 iForest 的时间复杂度为 $O(t\varphi\log\varphi)$。

其次，构建实例惩罚矩阵 F 的时间复杂为 $O(nt\log\varphi)$。显然，PCIP 算法是一个迭代的算法。在算法的每次迭代中，需要更新计算隶属度矩阵 P，投影矩阵 W 和类簇中心 Z，时间复杂度分别为 $O(ncd), O(nd\tilde{d})$ 和 $O(ncd)$，其中 $\tilde{d}$ 为投影空间的维度。因此，PCIP 算法总的时间复杂度为 $O(t\varphi\log\varphi+nt\log\varphi+nd\tilde{d}+ncd)$。当数据集的样本数量较大时，即在 $n\gg c$， $n\gg t$， $n\gg\psi$ 和 $n\gg d$ 的情况下，PCIP 算法的时间复杂度与样本的数量 n 线性相关。因此，PCIP 算法适用于高维的大数据集。

3.5 实验

为了验证所提出的 PCIP 算法的有效性，本章节选择了 7 个相关的算法作为对比算法，包括 AGLPP、GOLPP、JGOPL、LAPP、LPFCM、LPP 和 NPE。在 3.5.1 节中，本文将详细描述用于实验的 10 个图像数据集。3.5.2 节将简要介绍实验所使用的对比算法和 PCIP 算法的实验参数。本章的所有实验都是在一台处理器为 Intel(R) Core(TM) i9-10850K CPU、机带 RAM 为 32GB、MATLAB 2019b 和操作系统为 Windows10 的计算机上进行的。

3.5.1 数据集

本章节介绍了实验所用的 10 个图像数据集，包括 Yale、ORL、umist、FERET32x32、COIL20、AR、MSRA25、Palm、UPS、MINIST2k2k。表 3.1 展示了数据集的详细信息，包括样本量、特征、类簇、图片尺寸和英文缩略语。为了更加清晰地描述数据集，本文还从不同的数据集中随机选择了两个不同的类别，在每个类别中随机选择了 7 张图像。具体的图像如图 3.1 所示。此外，本文还对每个数据集进行了简单地介绍：

Yale[69]数据集是一个人脸图像数据集，包含 15 个人的 165 张图片。每个人在不同的灯光下有不同的姿势和表情，每张图的像素大小为 32 × 32。ORL[70]人脸数据集包含 40 个不同个体的 400 张图像。每张图片的大小为 32 × 32 像素。

umist[1]人脸数据集共包含 20 个不同个体的 574 张图像。每张图片的大小为 112 × 92 像素。FERET32x32[2]数据集是 FERET[71]的一个子集，包含 200 个不同的个体，拥有 1400 张不同光照下的灰度人脸图像。COIL20[72]数据集包含 20 个不同的个体，每个人旋转 360°并以 5°间隔拍摄。每个对象有 72 张图片，每张图片有 32 × 32 像素。AR[73]人脸数据集包含了 120 个不同个体的 1680 张图像。每个人有 14 张不同光线和表情的图像。每张图片的大小为 32 × 32 像素。MSRA25[74]人脸数据集包含 12 个不同个体的 1799 张照片。每个人有 113~186 张图片，每张图片有 16 × 16 像素。Palm[3]数据集是包含 2000 张灰度图像和 100 个不同个体的图像数据集，每张图片有 16 × 16 像素。UPS[4]是一个数字图像数据集，包含 2007 张样本图像，每张图像的像素为 16 × 16。MINIST2k2k[5]是包含 10 种手写数字(0~9)的数字图像数据集，包含 4000 张样本图像，每幅图像有 28 × 28 像素。

表 3.1 数据集信息

序号	数据集	样本量	特征数	类簇数	图片尺寸	简写
1	Yale	165	1024	15	32 × 32	YAL
2	ORL	400	1024	40	32 × 32	ORL
3	umist	574	10304	20	112 × 92	UMI
4	FERET32x32	1400	1024	200	32 × 32	FER
5	COIL20	1440	1024	20	32 × 32	COI
6	AR	1680	1024	120	32 × 32	AR
7	MSRA25	1799	256	12	16 × 16	MSR
8	Palm	2000	256	100	16 × 16	PAL
9	UPS	2007	256	100	16 × 16	UPS
10	MINIST2k2k	4000	784	10	28 × 28	MIN

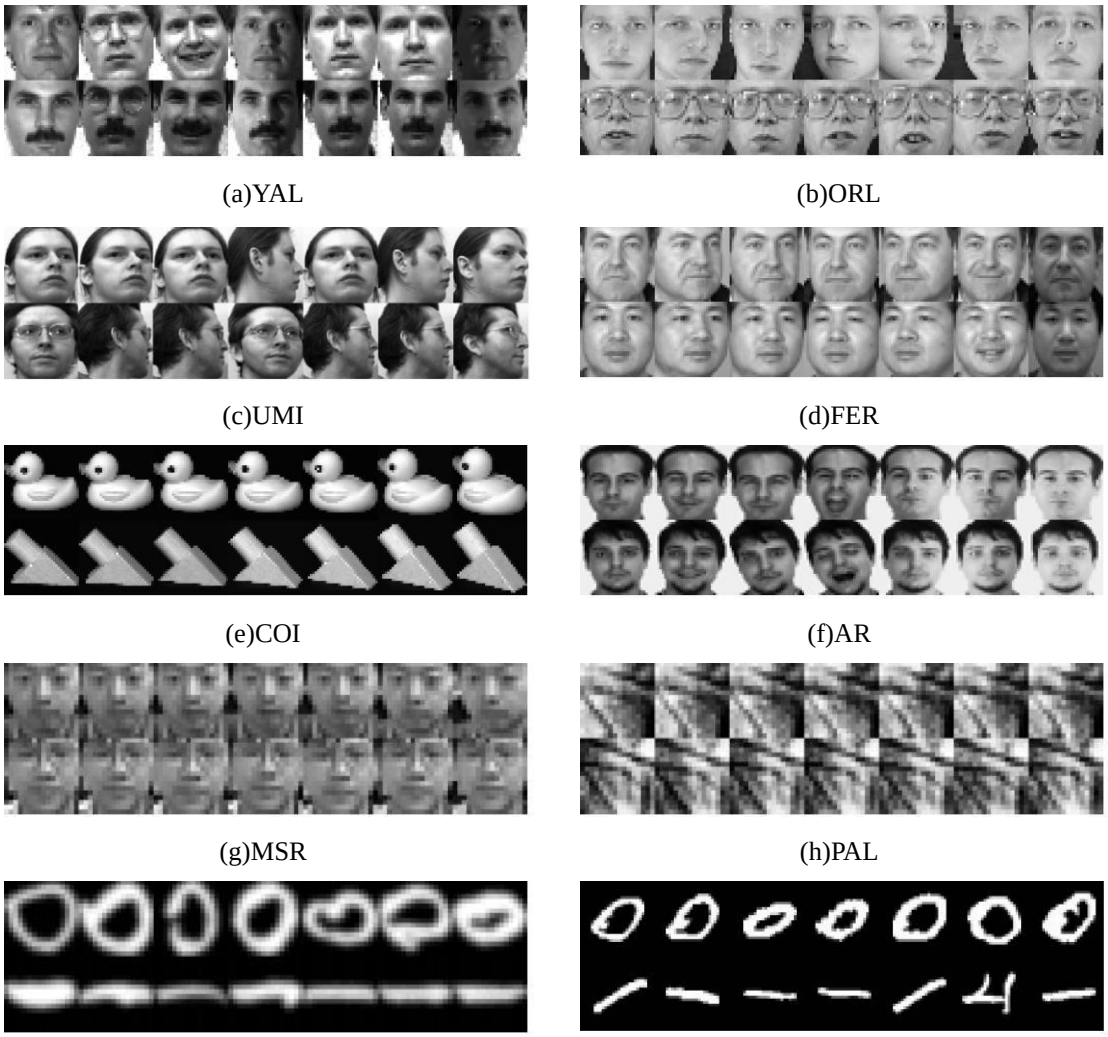
[1] http://images.ee.umist.ac.uk/danny/database.html.

[2] https://www.nist.gov/itl/products-and-services/color-feret-database.

[3] https://www.gwern.net/Crops.

[4] https://www.kaggle.com/datasets/bistaumanga/usps-dataset.

[5] http://yann.lecun.com/exdb/mnist.



(a)YAL (b)ORL

(c)UMI (d)FER

(e)COI (f)AR

(g)MSR (h)PAL

(i)UPS (j)MIN

图 3.1 图像数据集的样本图像

3.5.2 评价指标

本文采用聚类精度[75]（Cluster Accuracy，Accuracy）和归一化互信息[76,77]（Normalized Mutual Information，NMI）两个评价指标评估了 PCIP 算法的聚类性能，并在 10 个图像数据集上进行了实验。

Accuracy 是评估聚类算法精确度的一个重要指标，其值为样本分配到正确簇的个数占所有样本个数的比例。假设数据集的真实标签集为 $l=\{l_1,l_2,\ldots,l_n\}$，通过聚类算法获得伪标签集为 $L=\{L_1,L_2,\ldots,L_n\}$，其中 l_i 表示样本 x_i 的真实标签，L_i 表示聚类算法赋予样本 x_i 的标签。Accuracy 的计算如下：

$$Accuracy=\frac{\sum_{i=1}^{n}\delta(map(L_i),l_i)}{n} \tag{3-22}$$

其中 $map(\bullet)$ 是一个映射函数用于将每个类簇标签匹配映射到真实标签。$\delta(x,y)$ 是一个指示函数，其计算如下：

$$\delta(x,y)=\begin{cases}1, if\ \ x=y\\ 0, otherwise\end{cases} \tag{3-23}$$

Accuracy 的取值范围为[0,1]，值越接近 1，说明算法的聚类性能越好。

互信息(Mutual Information，MI)[78]是一种对称度量，它量化了两个变量之间共享的统计信息。假设数据集 X 具有 v 个类簇 $C=\{C_1,C_2,\ldots,C_v\}$，聚类得到的 r 个类簇 $T=\{T_1,T_2,\ldots,T_r\}$。归一化互信息(NMI)是一种归一化的 MI，用来度量真实类簇集 C 和伪类簇集 T 之间的相似性。从数据集 X 中随机选取一个数据样本 x_i，$|C_s|$ 表示样本属于类簇 C_s 的个数，则样本 x_i 属于类簇 C_s ($1\le s\le v$)的概率为：

$$P(C_s)=\frac{|C_s|}{n} \tag{3-24}$$

根据公式(3-24)，C 和 T 之间的 NMI 计算如下：

$$NMI(C,T)=\frac{\sum_{k=1}^{v}\sum_{j=1}^{r}P(C_k\cap T_j)\log\frac{P(C_k\cap T_j)}{P(C_k)P(T_j)}}{\sqrt{\sum_{k=1}^{v}P(C_k)\log(P(C_k)\sum_{j=1}^{r}P(T_j)\log(P(T_j))}} \tag{3-25}$$

式(3-25)中 $P(C_k\cap T_j)$ 表示数据样本同时属于类簇 C_k 和 T_j 的概率，NMI 的取值范围为[0,1]，该值越接近 1，表示算法的聚类性能越好。

3.5.3 实验参数设置

本节将简单介绍对比算法和实验的参数设置。AGLPP 是一种基于锚点图的降维算法，该算法不同于基于构建邻接图的算法，其利用锚点与样本的关系构建锚点图，更高效地捕获数据的结构。LPP 利用原始样本的局部邻居信息构建

邻接矩阵和邻接图，期望在原始空间构建的邻接图在投影空间中得到保留。LPFCM 将 FCM 与 LPP 相结合，在降维过程完成聚类。它不仅考虑了样本之间的相似性，还考虑了样本与相邻样本之间的相似性。NPE 通过考虑样本的局部邻域的线性重构系数，从数据中提取关键的信息，这使得原始空间的局部线性结构在低维空间中得以保留。GOLPP 通过在目标函数中加入熵正则化来保持图的均匀性，考虑了降维过程中的数据信息并动态地优化邻接图结构，减轻了算法对原始数据空间中 k 近邻准则的依赖。JGOPL 在优化邻接图的过程中引入局部约束，保持原始数据的局部结构。此外，为了提高模型对数据或异常值变化的鲁棒性，JGOPL 对损失函数采用了基于 l_{21} 范数的距离度量。LAPP 采用自适应局部和全局加权方法，可以有效地处理高维数据的降维。

实验包括三个主要步骤：（1）由于原始数据的协方差存在大量的零空间，为了提高算法的实验效率并保持实验一致性，首先使用 PCA 将所有图像数据集投影到 100 维的子空间中[79]；（2）在此基础上，利用每个算法将原始数据降维至 $\tilde{d}$ 维，其中 $\tilde{d} \in [10,100]$，步长为 10；（3）利用 k-means 算法对降维后的数据进行聚类。由于 PCIP 算法和 LPFCM 算法在降维过程中完成了聚类任务，因此不需要使用 k-means 算法进行聚类。由于 LPP、NPE、LPFCM 和 LAPP 都是通过构造邻接图来保留数据的局部结构，为了保证对比算法的公平性，本文统一设置参数 $k = 12$ 来构造邻接图[58]。在 PCIP 中，参数 α 是用来控制数据类簇中心的模糊指数，实验设置模糊指数 $\alpha \in \{1.1,1.2,\cdots,1.9,2\}$。参数 λ 是用来调整模型 (3-3) 中的第二项 $Tr(W^T S_t W)$ 的重要程度，通过设置不同的权衡参数 $\lambda \in \{10^{-5},10^{-4},10^{-3},10^{-2},10^{-1},1,10^{1},10^{2},10^{3},10^{4}\}$。此外，根据文献[64]实验通过设定 iTrees 的数目 t = 100 来构建 iForest。对于抽样的样本子集大小 φ，如果 $n > \varphi$，设定 $\varphi = 256$，否则设定 $\varphi = n$。所有算法的实验参数如表 3.2 所示。

表 3.2 实验参数设置

算法	参数设置
LPP	带宽 $\sigma=\{2,20,200,2000,20000,200000,2000000\}$
GOLPP	权衡因子 η=1，迭代终止阈值 $\varepsilon=0.001$
AGLPP	平滑参数 $\sigma=\{0.005,0.05,0.5,1.5\}$ 锚点 $m\in\{0.1n,0.11n,\cdots,0.19n,0.2n\}$
JGOPL	权衡因子 $\alpha,\beta\in\{0.00001,0.0001,0.01,0.1,1\}$
LAPP	带宽 $\sigma=\{2,20,200,2000,20000,200000,2000000\}$ 迭代终止阈值 $\varepsilon=0.001$
LPFCM	权衡因子 $\gamma=\{0.0001,0.001,0.01,0.1,1,10,100,1000\}$ 权衡因子 $\mu=\{0.01,0.1,1,10,100,1000,10000\}$
PCIP	模糊指数 $\alpha=\{1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2\}$ 权衡因子 $\lambda\in\{10^{-5},10^{-4},10^{-3},10^{-2},10^{-1},1,10^{1},10^{2},10^{3},10^{4}\}$

为了进一步评估实验数据，本文在 0.1 显著性水平上进行 Wilcoxon 符号秩检验[80]。符号“+”表示 PCIP 明显优于对比算法。符号“-”表示 PCIP 明显低于对比算法。符号“~”表示 PCIP 与对比算法没有显著差异。

3.5.4 不同维度实验分析

本节采用 Accuracy 和 NMI 作为聚类评价指标在 10 个图像数据集上进行实验。AGLPP、LPP、NPE、GOLPP、LAPP 和 JGOPL 算法在对数据进行降维后执行 100 次 *k*-means 算法，记录实验结果的标准差和平均值。选择最好的聚类结果所在的维度作为最优的维度。鉴于 LPFCM 算法和 PCIP 算法在降维的过程中同时完成了聚类任务，所以这两个算法不需要降维后再使用 *k*-means 进行聚类，只需要记录降维完成后的类簇结构。本文采用平均值±标准差（最佳维度）的形式描述实验的结果如表 3.3 和表 3.4 所示，表中粗体字体表示最好的聚类性能指标数据。此外，为了研究各个算法在不同维度下的降维效果，本文记录了所有算法在不同维度的最高的 Accuracy，实验结果如图 3.2 所示。

表 3.3 算法在 10 个基准数据集上的 Accuracy

Accuracy	AGLPP	GOLPP	JGOPL	LAPP	LPFCM	LPP	NPE	PCIP
YAL	0.9520 (100) ±0.023	0.1839 (10) ±0.013	0.3761 (70) ±0.021	0.4209 (10) ±0.025	0.5030 (100)	0.4203 (10) ±0.018	0.4355 (10) ±0.03	**0.5333 (20)**
ORL	0.4380 (70) ±0.019	0.3366 (100) ±0.027	0.5069 (100) ±0.025	0.6137 (30) ±0.023	0.5500 (100)	0.6227 (30) ±0.03	0.6111 (20) ±0.027	**0.6825 (90)**
UMI	0.4421 (60) ±0.023	0.2341 (100) ±0.022	0.3227 (90) ±0.019	0.4483 (10) ±0.019	0.4913 (90)	0.4393 (10) ±0.024	0.3075 (40) ±0.019	**0.5122 (90)**
FER	0.2473 (40) ±0.004	0.2621 (100) ±0.009	0.2689 (90) ±0.005	0.2801 (70) ±0.009	0.3257 (70)	0.2799 (70) ±0.007	0.2953 (60) ±0.011	**0.3414 (70)**
COI	0.6780 (60) ±0.023	0.2453 (100) ±0.031	0.5858 (90) ±0.022	0.7186 (10) ±0.027	0.6771 (10)	0.7224 (10) ±0.033	0.5799 (10) ±0.036	**0.7458 (80)**
AR	0.3049 (60) ±0.006	0.5826 (100) ±0.022	0.6450 (70) ±0.026	0.6465 (60) ±0.012	0.7482 (90)	0.6505 (60) ±0.021	0.6562 (40) ±0.026	**0.7851 (90)**
MSR	0.5481 (90) ±0.037	0.3068 (100) ±0.041	0.4971 (90) ±0.043	0.6632 (10) ±0.057	0.7132 (90)	0.6635 (10) ±0.05	0.5263 (30) ±0.054	**0.7221 (90)**
PAL	0.6226 (90) ±0.019	0.6535 (100) ±0.037	0.7108 (100) ±0.034	0.7933 (30) ±0.024	0.8480 (90)	0.7888 (30) ±0.022	0.7623 (40) ±0.02	**0.9025 (90)**
UPS	0.6698 (30) ±0.033	0.3119 (100) ±0.038	0.5098 (30) ±0.019	0.6902 (10) ±0.034	0.6891 (100)	0.6907 (10) ±0.039	0.6681 (10) ±0.027	**0.6912 (30)**
MIN	0.5413 (40) ±0.024	0.3039 (100) ±0.023	0.4801 (60) ±0.027	0.5128 (10) ±0.016	0.5648 (100)	0.5176 (10) ±0.017	0.4936 (10) ±0.022	**0.5685 (50)**
平均值	0.4887	0.3509	0.4903	0.5788	0.6110	0.5796	0.5336	**0.6485**
Wilcoxon	+	+	+	~	~	~	+	**N/A**

从表 3.3、表 3.4 和图 3.2 的实验结果中可以得到以下结论：

（1）PCIP 算法在 10 个数据集上获得了最高的平均 Accuracy 和平均 NMI。PCIP 的平均 Accuracy 比排在第二位的 LPFCM 算法高 3.75%，PCIP 的平均 NMI 比 LPFCM 算法高 2.75%。此外，表 3.3 中 PCIP 算法在 10 个数据集的 Accuracy 全部排名第一，表 3.4 中 PCIP 算法在 7 个数据集上的 NMI 排名第一。实验结果表明，PCIP 算法能够很好地学习高维图像数据集上的投影矩阵和隶属度矩阵，并在投影子空间上获得良好的聚类结构。

（2）在表 3.4 中，PCIP 算法在 MSR 和 UPS 数据集上的 NMI 略低于 LAPP，在 MIN 数据集上的 NMI 略低于 LPFCM。可能是因为样本赋予的实例惩罚系数

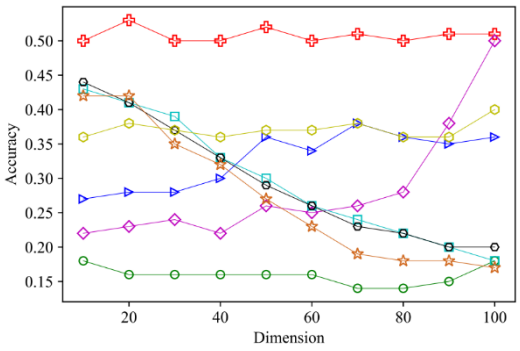
有些许偏差。但在剩余的 7 个数据集上，PCIP 的 NMI 均高于对比算法，这也证明了 PCIP 算法的有效性。

表 3.4 算法在 10 个基准数据集上的 NMI

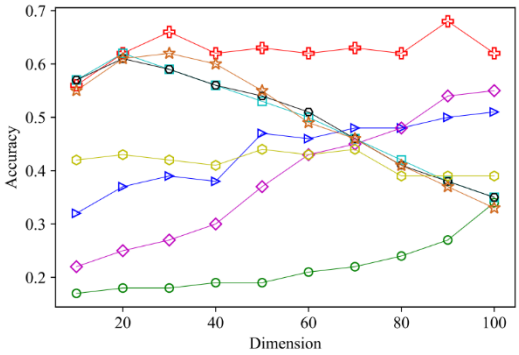
NMI	AGLPP	GOLPP	JGOPL	LAPP	LPFCM	LPP	NPE	PCIP
YAL	0.4424 (100) ±0.019	0.1614 (10) ±0.016	0.4287 (70) ±0.021	0.4816 (10) ±0.019	0.541 (100)	0.4806 (10) ±0.018	0.484 (10) ±0.021	**0.5904 (90)**
ORL	0.6565 (70) ±0.01	0.4669 (100) ±0.029	0.7004 (100) ±0.016(	0.7845 (20) ±0.014	0.7508 (100)	0.7833 (20) ±0.009	0.7887 (20) ±0.012	**0.8356 (90)**
UMI	0.6221 (60) ±0.015	0.2651 (100) ±0.021	0.3967 (90) ±0.029	0.6671 (10) ±0.007	0.6755 (100)	0.6669 (10) ±0.012	0.4084 (40) ±0.02	**0.6858 (90)**
FER	0.659 (50) ±0.003	0.6005 (10) ±0.005	0.6656 (90) ±0.004	0.6504 (50) ±0.003	0.6844 (80)	0.6495 (50) ±0.005	0.6654 (30) ±0.005	**0.7020 (70)**
COI	0.7689 (90) ±0.01	0.2995 (100) ±0.04	0.6811 (100) ±0.029	0.7987 (10) ±0.012	0.7658 (100)	0.8011 (10) ±0.017	0.7222 (10) ±0.016	**0.8188 (70)**
AR	0.6391 (60) ±0.003	0.8184 (100) ±0.013	0.8630 (100) ±0.01	0.8677 (60) ±0.005	0.9123 (100)	0.8689 (60) ±0.008	0.8757 (50) ±0.01	**0.9126 (90)**
MSR	0.5930 (50) ±0.02	0.2866 (100) ±0.066	0.5402 (90) ±0.03	**0.7825 (10)** ±0.043	0.7634 (90)	0.7706 (10) ±0.033	0.6045 (30) ±0.043	0.7782 (90)
PAL	0.8283 (70) ±0.007	0.8654 (100) ±0.022	0.9041 (100) ±0.014	0.9385 (50) ±0.009	0.9520 (90)	0.9389 (50) ±0.009	0.9303 (40) ±0.006	**0.9661 (90)**
UPS	0.6264 (60) ±0.016	0.1991 (100) ±0.053	0.4433 (30) ±0.009	**0.6686 (10)** ±0.017	0.6217 (100)	0.6682 (10) ±0.022	0.6367 (10) ±0.016	0.6680 (80)
MIN	0.4984 (100) ±0.014	0.2159 (100) ±0.025	0.3606 (80) ±0.008	0.4688 (10) ±0.012	**0.4817 (100)**	0.4660 (10) ±0.01	0.4284 (10) ±0.01	0.4707 (60)
平均 NMI	0.6333	0.4179	0.5984	0.7108	0.7153	0.7094	0.6544	**0.7428**
Wilcoxon	+	+	+	~	~	~	+	**N/A**

（3）表 3.3 和表 3.4 中的 Wilcoxon 统计检验的结果表明，所提的 PCIP 算法与 LPP、LAPP 和 LPFCM 算法没有显著差异，显著优于 AGLPP、GOLPP、JGOPL 和 NPE 算法。

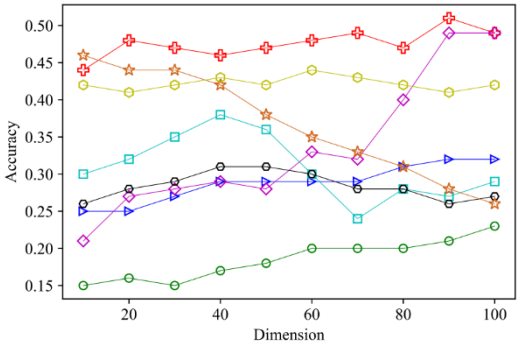
（4）从图 3.2 中可以看出，在 YAL、ORL、UMI、FER、COI、MSR、UPS 和 MIN 数据集上，PCIP 算法在不同维度的 Accuracy 显著高于对比算法。在 AR 和 PAL 数据集上，PCIP 算法在较低的维度上的 Accuracy 低于对比算法，但是从整体上看，PCIP 算法获得最高的 Accuracy。



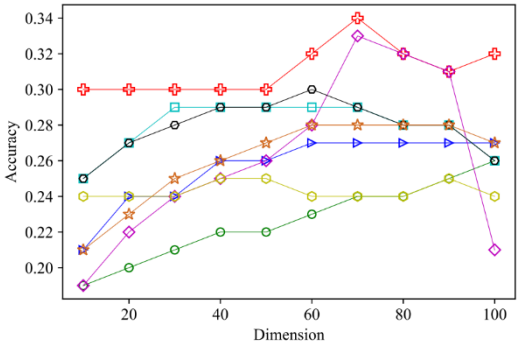
(a)YAL



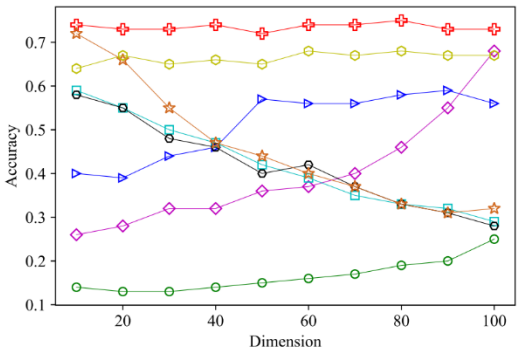
(b)ORL



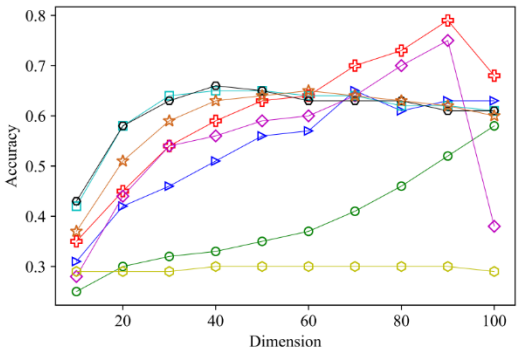
(c)UMI



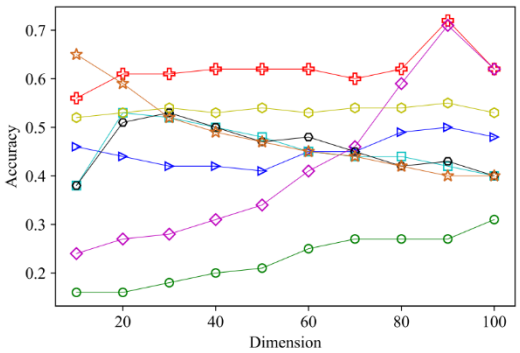
(d)FER



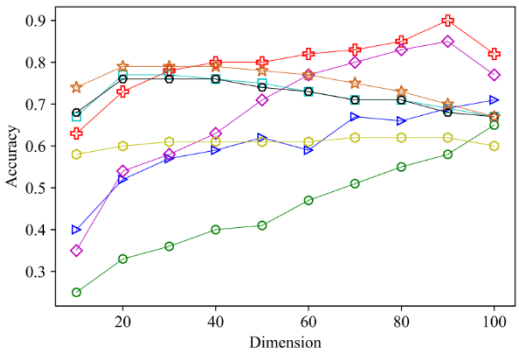
(e)COI



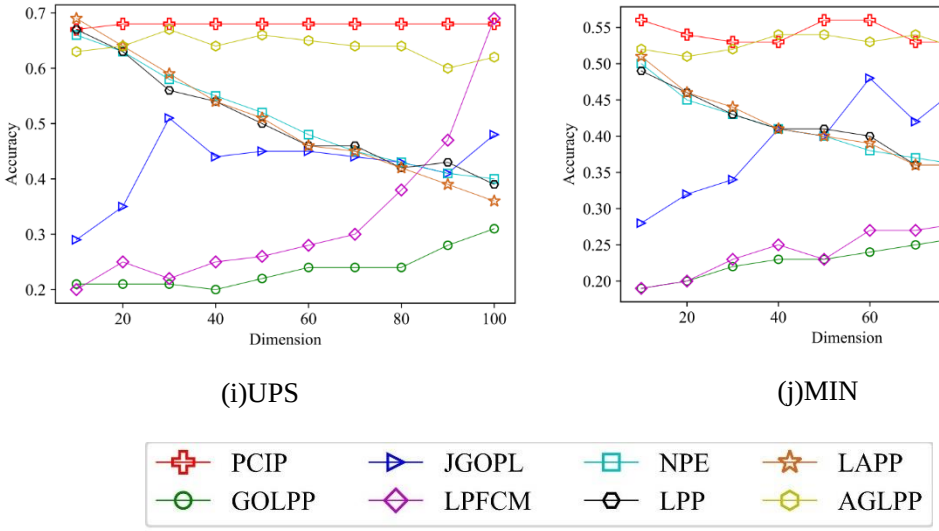
(f)AR



(g)MSR



(h)PAL



(i)UPS (j)MIN

图 3.2 每个算法在不同维度上的 Accuracy

3.5.5 消融实验

从模型(3-3)中可以看出，模型主要包含两个部分，第一部分构建实例惩罚矩阵 F 并与 FCM 算法相结合，第二部分用参数 λ 来调节模型第二项 $Tr(W^T S_t W)$ 的重要程度。为了研究每一部分对模型的影响，本章节进行了消融实验。消融实验包含两个部分：（1）令实例惩罚矩阵 F 为单位矩阵从而消除实例惩罚系数对模型的影响，将其命名为 Ablation1；（2）令 λ 为一个极小值从而消除 $Tr(W^T S_t W)$ 对模型的影响，将其命名为 Ablation2。本节将 Ablation1、Ablation2 和 PCIP 算法在 10 个数据集上进行实验，记录实验结果的 Accuracy 和最优投影子空间的维度。消融实验的结果如表 3.5。

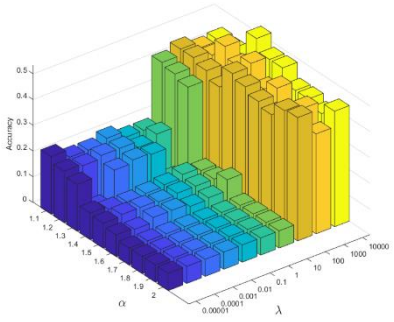
从表 3.5 中可以看出：所提算法 PCIP 在 10 个数据集中都排名第一，且获得了最高的平均 Accuracy。PCIP 的平均 Accuracy 相较于 Ablation1 增加了 1.91%。这说明了 PCIP 算法构造的实例惩罚矩阵可以有效发现并处理异常样本。同样，PCIP 的平均 Accuracy 相较于 Ablation2 增加了 3.52%。这说明 PCIP 结合 PCA 算法能充分考虑了数据的全局信息，在低维子空间获得好的类簇结构。从实验结果上看，融合实例惩罚矩阵、FCM 和 PCA 的模型在处理高维图像数据集上是行之有效的。

表 3.5 消融实验

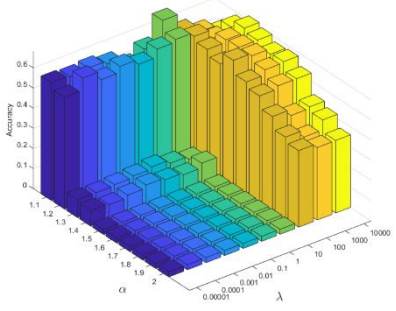
Accuracy	Ablation1	Ablation2	PCIP
YAL	0.5212(40)	0.4848(100)	**0.5333(20)**
ORL	0.6450(40)	0.6150(90)	**0.6825(90)**
UMI	0.5000(90)	0.4547(100)	**0.5122(90)**
FER	0.3257(70)	0.3343(70)	**0.3414(70)**
COI	0.7444(10)	0.7146(100)	**0.7458(80)**
AR	0.7565(90)	0.7054(90)	**0.7851(90)**
MSR	0.6815(90)	0.7098(90)	**0.7221(90)**
PAL	0.8930(90)	0.8850(90)	**0.9025(90)**
UPS	0.6781(100)	0.6781(100)	**0.6912(30)**
MIN	0.5470(10)	0.5512(100)	**0.5685(50)**
平均 Accuracy	0.6294	0.6133	**0.6485**

3.5.6 参数实验

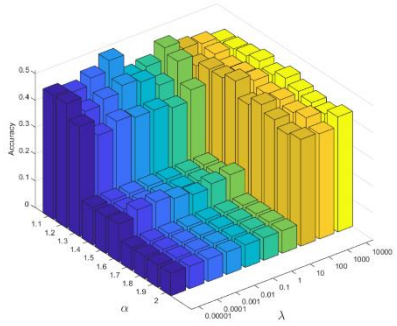
PCIP 算法包含两个超参数，参数α和参数λ。参数α是模糊指数，它用来控制样本到不同聚类中心的隶属度。参数λ是一个权衡因子，它用来调节$Tr(W^T S_t W)$的重要性。记录 PCIP 算法获得最高 Accuracy 时的投影子空间维度，在该维度下记录在不同参数下 PCIP 的聚类精度。图 3.3 的三维图展示了不同的参数α和参数λ在 10 个图像数据集上的 Accuracy。



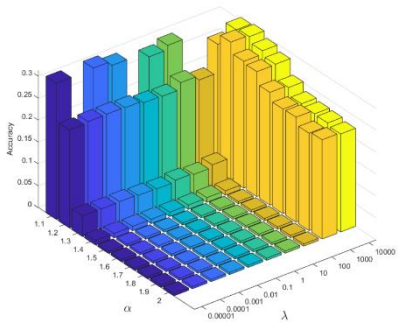
(a)YAL



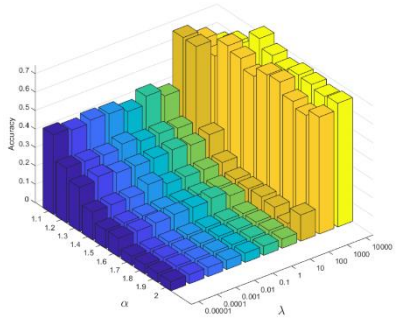
(b)ORL



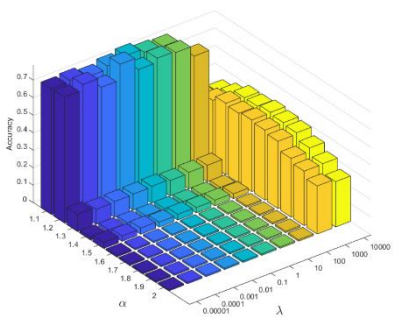
(c)UMI



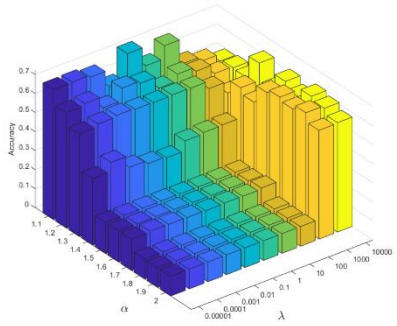
(d)FER



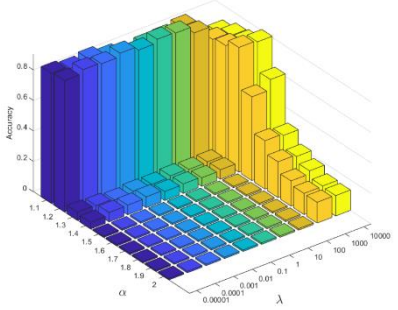
(e)COI



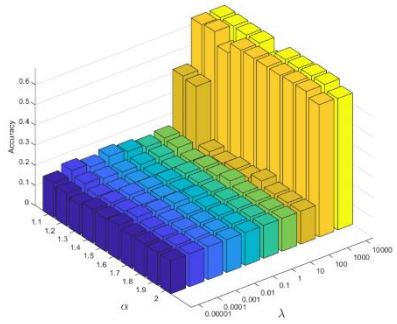
(f)AR



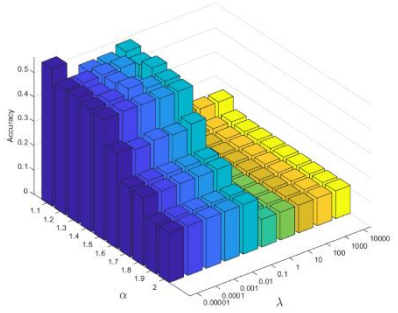
(g)MSR



(h)PAL



(i)UPS



(j)MIN

图 3.3 参数 α 和参数 λ 对 PCIP 算法的影响

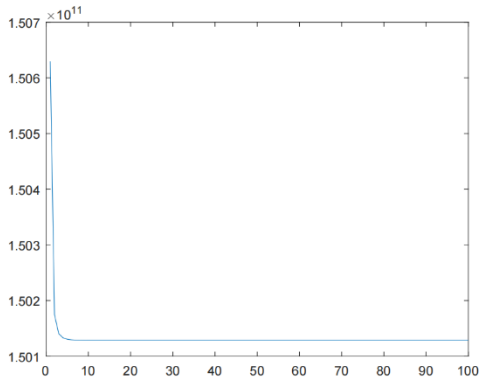
从图 3.3 中可以得到：

（1）在 YAL、ORL、UMI、FER、COI、AR、MSR、PAL 和 MIN 数据集上，当$1.1\le\alpha\le1.2$时，PCIP 获得较高的 Accuracy，因为当模糊指数α较小时，样本对于某一个类的隶属度明显高于其他类，聚类结果比较明确。当$\alpha\ge1.3$时，PCIP 算法的 Accuracy 随着α的增加显著下降，因为当α较大时，样本被分配到每个类的隶属度相对平均，聚类结果较为模糊。所以模糊指数α的推荐取值范围为$[1.1,1.2]$。

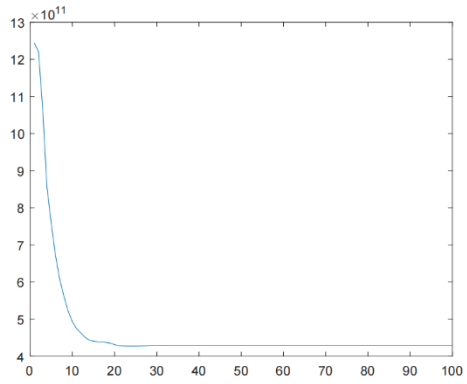
（2）在 YAL、COL 和 UPS 数据集上，当$\lambda\ge1000$时，PCIP 的 Accuracy 随着λ的增加显著增加。在 MSR、PAL、UNI、FER、ORL 数据集上，随着λ的增加显著增加，Accuracy 没有显著变化。在 AR 和 MIN 数据集上，PCIP 的 Accuracy 随着λ的增加显著减少。因此，权衡因子γ的合适取值取决于具体的数据集。

3.5.7 收敛性实验

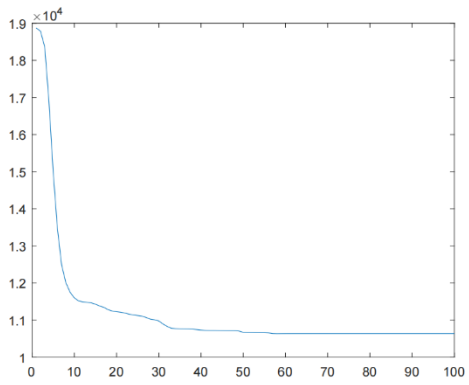
本文已经在第 3.3 节对 PCIP 算法的收敛性进行了理论分析。本节将对 PCIP 算法的收敛性进行实验，以证明理论分析的准确。本文选择 PCIP 获得最高聚类精度时的参数 α、λ和最优维数，记录了 100 次迭代的目标函数值，实验结果为如图 3.4 所示。



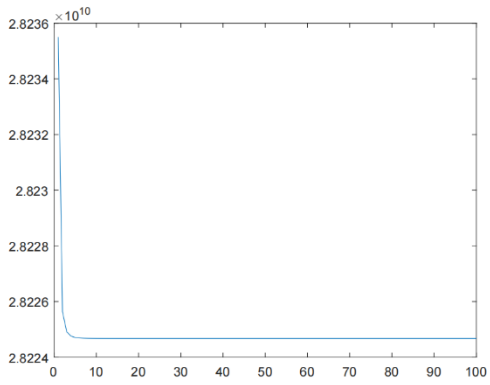
(a)YAL



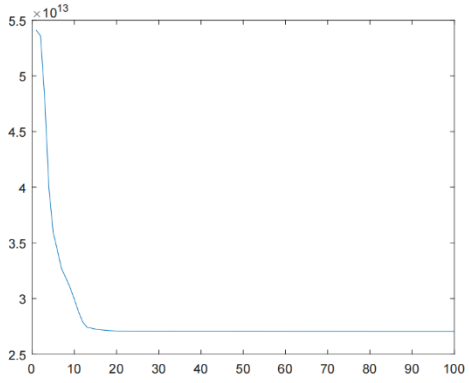
(b)ORL



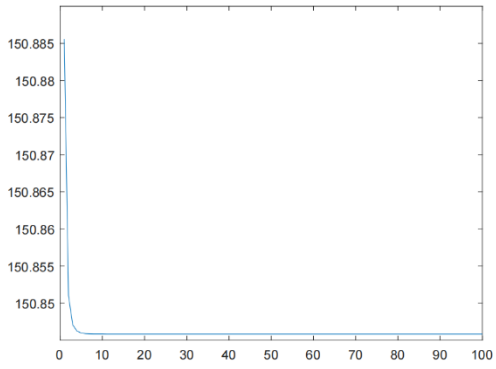
(c)UMI



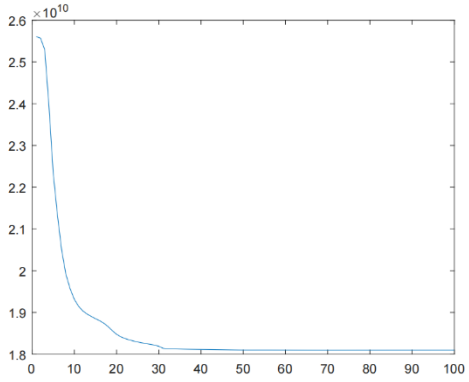
(d)FER



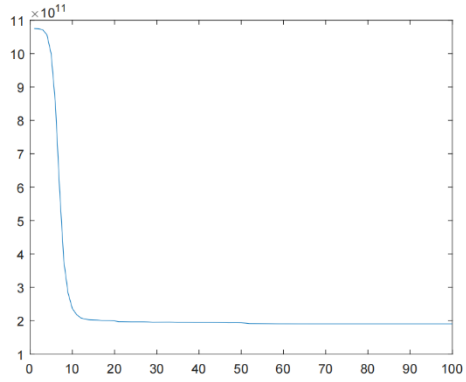
(e)COI



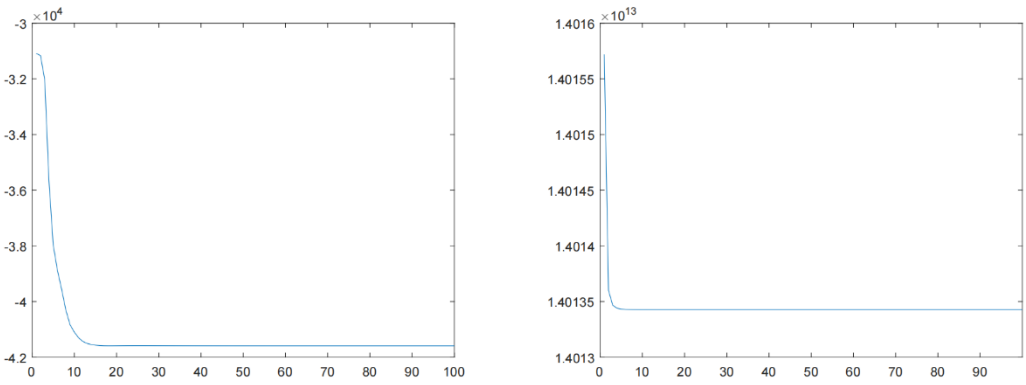
(f)AR



(g)MAR



(h)PAL



(i)UPS (j)MIN

图 3.4 PCIP 算法在 10 个数据集上的收敛曲线

从图 3.4 可以看出，PCIP 算法的目标函数值随着迭代次数的增加不断减少直至收敛，实验结果与理论分析保持一致。此外，PCIP 算法在 YAL、ORL、FER、COI、AR、PAL、UPS 和 MIN 数据集上经过 20 次迭代后收敛，PCIP 算法收敛速度很快。

3.5.8 运行时间实验

算法的时间复杂度是衡量算法效率和性能的重要指标。在 3.4 节中，本文对 PCIP 算法进行了算法的时间复杂度分析。为了进一步证明分析的准确性，本文进行了算法的运行时间实验。具体来说，首先将低维空间的维度 $\tilde{d}$ 设为 50，计算每个算法在该维数下的平均运行时间。由于 PCIP、JGOPL、GOLPP、AGLPP、LPFCM 和 LAPP 都是迭代的算法，为了保证实验的公平性，只记录了这些算法一次迭代的运行时间。运行时实验结果如表 3.6 所示，括号中的数字表示相应算法的运行时排名。排序越小表示算法运行地越快。

表 3.6 每种算法在不同数据集上的平均运行时间(毫秒)

Time	AGLPP	GOLPP	JGOPL	LAPP	LPFCM	LPP	NPE	PCIP
YAL	47.82(5)	128.12(7)	**3.12(1)**	15.98(4)	137.39(8)	64.73(6)	15.62(3)	10.03(2)
ORL	40.72(3)	109.38(7)	**15.41(1)**	50.94(5)	560.35(8)	98.21(6)	46.88(4)	34.52(2)
UMI	**18.94(1)**	185.94(7)	87.75(6)	69.02(5)	227.73(8)	40.18(2)	46.88(3)	62.22(4)
FER	**83.33(1)**	375.00(6)	203.38(2)	256.63(3)	8439.73(8)	276.79(4)	437.5(7)	290.18(5)
COI	**76.23(1)**	667.19(7)	250.88(4)	257.88(5)	2610.46(8)	227.68(3)	359.38(6)	148.68(2)
AR	**92.33(1)**	485.94(5)	1618.94(7)	338.82(4)	1695.37(8)	323.66(3)	703.12(6)	266.44(2)
MSR	**111.4(1)**	742.19(7)	532.47(6)	382.23(5)	6422.82(8)	343.75(3)	359.38(4)	160.91(2)
PAL	**133.5(1)**	1001.56(7)	468.95(4)	466.16(3)	3223.97(8)	488.84(5)	750.0(6)	348.7(2)
UPS	**125.9(1)**	528.12(5)	2769.84(8)	479.62(4)	2260.04(7)	473.21(3)	750.0(6)	237.1(2)
MIN	**404.7(1)**	1614.0(3)	6664.2(7)	2081.(4)	43248.94(8)	2263.39(5)	4578.12(6)	814.1(2)
平均排名	1.6	6.1	4.6	4.2	7.9	4	5.1	2.5

从表 3.6 中可以看出，在大多数的数据集上，AGLPP 算法运行时间最短，因为 AGLPP 只需要考虑锚点与样本之间的关系，节省了大量的计算时间，其时间复杂度为$O(nmd)$。JGOPL 在 YAL 和 ORL 数据集上排名第一，因为其采用l_{21}范数构建邻接图，当数据的样本量较少时，JGOPL 能快速地优化矩阵。在 Yale、ORL、UMI、FER 等较小的数据集上，由于 PCIP 算法基于 iForest 构造实例惩罚矩阵，在构建 iForest 时需要构建大量的 iTree，增加了算法的运行时间，因此 PCIP 算法在小的数据集上运行速度不具有优势。但针对 MIN、UPS 和 PAL 等高维的大数据集，PCIP 算法相较于大部分对比算法运行速度更快，因为 PCIP 的时间复杂度为$O(t\varphi\log\varphi+nt\log\varphi+nd\tilde{d}+ncd)$，其时间复杂度与样本数量$n$近似线性相关。当数据集的样本量越大，PCIP 算法的运行速度相比对比算法越快。总的来说，运行实验的结果与 3.4 节的算法时间复杂度分析一致。

3.6 本章小结

针对高维数据聚类中存在的一些问题，本章提出了一种带有实例惩罚的投影模糊 C 均值聚类算法（PCIP）。该算法融合了聚类任务和降维任务，在迭代优化目标函数的同时学习隶属度矩阵和投影矩阵，在投影子空间中获得良好的聚类结构。为了有效识别和处理数据中的异常样本，减轻噪声样本对模型的影响，PCIP 基于数据的分布构造实例惩罚矩阵，为每个样本分配实例惩罚系数。此外，本文还对 PCIP 算法的收敛性进行了证明并分析了算法的时间复杂度。PCIP 算法的时间复杂度与样本的数量 n 近似线性相关，能够有效地处理大规模

数据集。为了验证所提算法的有效性，本文在 10 个图像数据集上进行了大量实验，包括不同维度实验、消融实验、参数实验、收敛性实验等。实验结果验证了所提出的 PCIP 算法的优越性。

4 快速锚点图保持投影算法

现有的基于图的降维算法普遍存在两个缺陷：（1）通过构建邻接图来保留数据的局部信息，时间复杂度不低于$O(n^2d)$无法适用于大的数据集，其中 n 表示数据的样本量，d 表示数据的维度；（2）没有考虑原始数据空间的类簇信息，导致降维后有价值的信息弱化甚至丢失。为了解决以上问题，本章将详细介绍所提的快速锚点图保持投影算法（FAGPP）。

4.1 模型

LPP 利用原始数据的局部信息构建邻接图和邻接矩阵，期望在投影空间仍然能保留原始空间中样本的局部信息。与 LPP 算法的思想类似，FAGPP 利用原始数据的类簇信息构建锚点图和锚点隶属度矩阵，期望在低维空间中仍然保留原始数据的类簇信息。为了保持原始数据中所构造的锚点图信息，在 FAGPP 模型中引入$\|J-H\|_F^2$项，其中 J 为原始数据空间中构建的锚点与样本之间的隶属度矩阵，H 为在投影空间中学到的锚点与投影空间中样本之间的隶属度矩阵。下面的问题是如何在低维子空间快速地中学习锚点和锚点隶属度矩阵。鉴于 FCM 模型可以同时学习类簇中心和隶属度矩阵。受到这个想法的启发，定义了以下模型：

$$\begin{aligned}&\min \sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i-\tilde{o}_k\right\|_2^2\\&s.t.\, W^TW=I, h_{\cdot i}{}^T\mathbf{1}_m=1, H\geq 0\end{aligned} \tag{4-1}$$

公式(4-1)是一个类似 FCM 模型，其中$\tilde{o}_k$表示投影数据空间的锚点，h_{ik}表示投影空间中样本W^Tx_i与锚点$\tilde{o}_k$之间的隶属度关系。

为了在投影空间中保留原始数据空间的锚点图结构，提出了如下模型：

$$\begin{aligned}&\min \|J-H\|_F^2+\gamma\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i-\tilde{o}_k\right\|_2^2\\&s.t.\, W^TW=I, h_{\cdot i}{}^T\mathbf{1}_m=1, H\geq 0\end{aligned} \tag{4-2}$$

公式(4-2)中 J 为原始数据空间的锚点隶属度矩阵，$\|J-H\|_F^2$表示原始数据空间

构建的锚点图结构在投影空间中得到保留，γ是一个权衡因子，用于调节模型第二项$\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^{T}x_{i}-\tilde{o}_{k}\right\|_{2}^{2}$的重要性。

此外，为了处理原始数据的全局信息，FAGPP 还融合了 PCA 学习模型。因此，基于以上分析，最后，提出如下 FAGPP 模型：

$$\begin{aligned}&\min\left\|J-H\right\|_{F}^{2}+\gamma\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^{T}x_{i}-\tilde{o}_{k}\right\|_{2}^{2}-\lambda Tr(W^{T}S_{t}W)\\&s.t.W^{T}W=I,h_{\cdot i}{}^{T}\mathbf{1}_{m}=1,H\geq 0\end{aligned}\tag{4-3}$$

式(4-3)中λ是一个权衡因子，用于调节模型第三项$Tr(W^{T}S_{t}W)$的重要性。

4.1.1 模型优化

本文采用一种交替优化的算法求解模型(4-3)，在算法每次迭代的过程中，首先固定隶属度矩阵H，求解投影矩阵W和投影空间锚点$\tilde{O}$；其次固定W和$\tilde{O}$，求解H。模型(4-3)的优化分两步进行。

（1）固定H，优化W和$\tilde{O}$

当锚点隶属度矩阵H固定时，模型(4-3)可以转化为：

$$\begin{aligned}&\min\gamma\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^{T}x_{i}-\tilde{o}_{k}\right\|_{2}^{2}-\lambda Tr(W^{T}S_{t}W)\\&s.t.W^{T}W=I,h_{\cdot i}{}^{T}\mathbf{1}_{m}=1,H\geq 0\end{aligned}\tag{4-4}$$

因为模型(4-3)中投影空间中第 k 个锚点$\tilde{o}_{k}$没有约束，首先对$\tilde{o}_{k}$求导并令其导数为0。可以得到：

$$\begin{aligned}&\frac{\partial(\gamma\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^{T}x_{i}-\tilde{o}_{k}\right\|_{2}^{2}-\lambda Tr(W^{T}S_{t}W))}{\partial\tilde{o}_{k}}\\&=\gamma(\sum_{i=1}^{n}2h_{ki}W^{T}x_{i}-2h_{ki}\tilde{o}_{k})=0\end{aligned}\tag{4-5}$$

根据公式(4-5)可以得到：

$$\tilde{o}_k = \frac{\sum_{i=1}^{n} h_{ki} W^T x_i}{\sum_{i=1}^{n} h_{ki}} = W^T \frac{\sum_{i=1}^{n} h_{ki} x_i}{\sum_{i=1}^{n} h_{ki}} \tag{4-6}$$

令 $o_k = \frac{\sum_{i=1}^{n} h_{ki} x_i}{\sum_{i=1}^{n} h_{ki}}$，则 $\tilde{o}_k = W^T o_k$。显然 $O = [o_1, o_2, \ldots, o_k] \in R^{d \times k}$ 为原始数据的锚点。

由公式(4-6)可以得到了一个有趣的结论：低维数据空间中的第 k 个锚点 $\tilde{o}_k$ 是锚点 O_k 通过投影矩阵 W 在原空间中的线性变换。优化问题(4-4)可以表示为：

$$\begin{aligned} &\min \gamma \sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} \left\| W^T x_i - W^T o_k \right\|_2^2 - \lambda Tr(W^T S_t W) \\ &s.t. W^T W = I, h_{\bullet i}^T \mathbf{1}_m = 1, H \geq 0 \end{aligned} \tag{4-7}$$

下面对公式(4-7)进行求解，优化方式与公式(3-7)类似。令 $L^k \in R^{n \times n}$，显然 $L^k \in R^{m \times m}$ 是一个对角矩阵，其第 i 个对角元素为 $l_{ii}^k = \sum_{k=1}^{m} h_{ki}$。令 L^i 是另一个对角矩阵，其第 k 个对角元素为 $l_{kk}^i = \sum_{i=1}^{n} h_{ki}$。令 $R = W^T X$，$Z = W^T O$，$r_{\bullet i}$ 为矩阵 R 的第 i 列元素，$z_{\bullet k}$ 为矩阵 Z 的第 k 列元素，可以得到：

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} \left\| r_{\bullet i} - z_{\bullet k} \right\|_2^2 = \sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} (r_{\bullet i} - z_{\bullet k})^T (r_{\bullet i} - z_{\bullet k}) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} (r_{\bullet i}^T r_{\bullet i} - 2 z_{\bullet k}^T r_{\bullet i} + z_{\bullet k}^T z_{\bullet k}) \\ &= \sum_{i=1}^{n} l_{ii}^k r_{\bullet i}^T r_{\bullet i} + \sum_{k=1}^{m} l_{kk}^I z_{\bullet k}^T z_{\bullet k} - 2 \sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} z_{\bullet k}^T r_{\bullet i} \end{aligned} \tag{4-8}$$

公式(4-8)中由 3 个独立的项构成，可以由迹的形式表达：

$$\begin{aligned} &\sum_{i=1}^{n} l_{ii}^k r_{\bullet i}^T r_{\bullet i} = Tr(L^k R^T R) = Tr(R L^k R^T) \\ &\sum_{k=1}^{m} l_{kk}^I z_{\bullet k}^T z_{\bullet k} = Tr(L^i Z^T Z) = Tr(Z L^i Z^T) \\ &\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} z_{\bullet k}^T r_{\bullet i} = \sum_{i=1}^{n} (Z e_i)^T \sum_{k=1}^{m} h_{ki} R_k = Tr(RHZ^T) \end{aligned} \tag{4-9}$$

公式(4-9)中 e_i 表示第 i 个正则向量。令 $B = XL^k X^T - 2XH\tilde{O}^T + \tilde{O}L^i\tilde{O}^T$，$\tilde{B} = (B + B^T)/2$，将可以得到：

$$\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i - W^T o_k\right\|_2^2 = Tr(W^T \tilde{B} W) \tag{4-10}$$

根据公式(4-10)，模型(4-4)可以转化为：

$$\begin{aligned}
&\min \gamma\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i - W^T o_k\right\|_2^2 - \lambda Tr(W^T S_t W)\\
&s.t.\ W^T W = I,\ h_{\cdot i}^{\ T}\mathbf{1}_m = 1,\ H \ge 0\\
&= \min \gamma Tr(W^T \tilde{B} W) - \lambda Tr(W^T S_t W)\\
&s.t.\ W^T W = I,\ h_{\cdot i}^{\ T}\mathbf{1}_m = 1,\ H \ge 0\\
&= \min Tr(W^T(\gamma\tilde{B} - \lambda S_t)W)\\
&s.t.\ W^T W = I,\ h_{\cdot i}^{\ T}\mathbf{1}_m = 1,\ H \ge 0
\end{aligned} \tag{4-11}$$

其中 S_t 为 X 的协方差。在 $W^T W = I$ 的约束下，投影矩阵 W 为 $\gamma\tilde{B} - \lambda S_t$ 中 $\tilde{d}$ 个最小特征值对应的特征向量。

（2）固定 W 和 $\tilde{O}$，优化 H

当投影矩阵 W 和锚点 $\tilde{O}$ 固定时，模型(4-3)转化为以下模型：

$$\begin{aligned}
&\min \left\|J - H\right\|_F^2 + \gamma\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i - \tilde{o}_k\right\|_2^2\\
&s.t.\ h_{\cdot i}^{\ T}\mathbf{1}_m = 1,\ H \ge 0
\end{aligned} \tag{4-12}$$

将公式(4-12)进一步简化为以下最小化问题：

$$\begin{aligned}
&\min \left\|J - H\right\|_F^2 + \gamma\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki} d_{ki}\\
&s.t.\ h_{\cdot i}^{\ T}\mathbf{1}_m = 1,\ H \ge 0
\end{aligned} \tag{4-13}$$

公式(4-13)中 $d_{ki} = \left\|W^T x_i - \tilde{o}_k\right\|_2^2$ 表示投影空间中样本 $W^T x_i$ 与锚点 $\tilde{o}_k$ 的欧式距离。

下面对公式(4-13)做一些数学变换：

$$\begin{aligned}&\min_{\boldsymbol{H}}\|J-H\|_F^2+\gamma\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}d_{ki}\\&=\min_{\boldsymbol{H}}\sum_{i=1}^{n}\sum_{k=1}^{m}({h_{ki}}^2-2h_{ki}j_{ki}+\gamma h_{ki}d_{ki})\\&=\min_{\boldsymbol{H}}\sum_{i=1}^{n}\sum_{k=1}^{m}({h_{ki}}^2-h_{ki}(2j_{ki}-\gamma d_{ki}))\\&=\min_{\boldsymbol{H}}\sum_{i=1}^{n}\sum_{k=1}^{m}(h_{ki}-(j_{ki}-\gamma d_{ki}/2))^2\end{aligned}\tag{4-14}$$

由于模型对每个$h_{\cdot i}$上都有约束，因此将优化问题(4-14)划分为 n 个子问题，则第 i 个子问题为：

$$\min_{\boldsymbol{H}_i}\sum_{i=1}^{n}\sum_{k=1}^{m}(h_{ki}-(j_{ki}-\gamma d_{ki}/2))^2\tag{4-15}$$

因此，问题(4-12)中求解优化问题的关键步骤是求解以下问题：

$$\begin{aligned}&\min\frac{1}{2}\|h_{\cdot i}-\tau_{\cdot i}\|_2^2\\&s.t.\ {h_{\cdot i}}^T\boldsymbol{1}_m=1,\ h_{\cdot i}\geq 0\end{aligned}\tag{4-16}$$

其中$\tau_{\cdot i}=((j_{1i}-\gamma d_{1i}/2),\ldots,(j_{mi}-\gamma d_{mi}/2))$。让$\tau_{ki}$为$\tau_{\cdot i}$中的第 k 个元素，将问题(4-16)中优化问题的拉格朗日函数定义为：

$$\frac{1}{2}\|h_{\cdot i}-\tau_{\cdot i}\|_2^2-\alpha_i({h_{\cdot i}}^T\boldsymbol{1}_m-1)-\beta_i^T h_{\cdot i}\tag{4-17}$$

公式(4-17)中α_i和β_i是两个拉格朗日乘数。令问题(4-17)中的最优解为$h_{\cdot i}^*$，α_i和β_i的最优解分别为α_i^*和β_i^*。根据 KKT 条件，可以得到：

$$\begin{cases}h_{ki}^*-\tau_{ki}-\alpha_i^*-\beta_i^*=0\\h_{ki}^*\geq 0\\{h_{\cdot i}}^T\boldsymbol{1}_m=1\\\beta_{ki}^*\geq 0\\h_{ki}^*\beta_{ki}^*=0\end{cases}\tag{4-18}$$

其中h_{ki}^*是$h_{\cdot i}$的第 k 个标量元素。根据约束${h_{\cdot i}}^T\boldsymbol{1}_m=1$，可以得到$\alpha_i^*=\frac{1-{\boldsymbol{1}_m}^T\tau_{\cdot i}-{\boldsymbol{1}_m}^T\beta_i^*}{m}$，因此$h_{\cdot i}=(\tau_{\cdot i}-\frac{{\boldsymbol{1}_m}^T}{m}\tau_{\cdot i}\boldsymbol{1}_m+\frac{1}{m}\boldsymbol{1}_m-\frac{{\boldsymbol{1}_m}^T\beta_i^*}{m}\boldsymbol{1}_m)+\beta_i^*$。令

$\bar{\beta}_i^* = \frac{\mathbf{1}_m^T \beta_i^*}{m}$，$\xi_i = \tau_{\cdot i} - \frac{\mathbf{1}_m^T}{m}\tau_{\cdot i}\mathbf{1}_m + \frac{1}{m}\mathbf{1}_m$，可以得到 $h_{\bullet i} = \xi_i + \beta_i^* - \bar{\beta}_i^*$。对于任意 k，可以得到 $h_{ki} = \xi_{ki} + \beta_{ki}^* - \bar{\beta}_i^*$。

由于 $\xi_{ki} + \beta_{ki}^* - \bar{\beta}_i^* = (\xi_{ki} - \bar{\beta}_i^*)_+$，其中 $x_+ = max(x,0)$，因此 $h_{ki} = (\xi_{ki} - \bar{\beta}_i^*)_+$。在约束为 $h_{\cdot i}^T \mathbf{1}_m = 1$ 的情况下，$\sum_{k=1}^{m}(\xi_{ki} - \bar{\beta}_i^*)_+ - 1 = 0$。定义函数：

$$f(\bar{\beta}_i^*) = \sum_{k=1}^{m}(\xi_{ki} - \bar{\beta}_i^*)_+ - 1 \tag{4-19}$$

通过求解上述函数 $f(\bar{\beta}_i^*) = 0$，可以获得最优解 $\bar{\beta}_i^*$。此外，$\bar{\beta}_i^* \geq 0$ 和 $f(\bar{\beta}_i^*)$ 是一个逐块线性凸函数，该函数除在 ξ_{ki} 点外都是可微的，可以用该点的左导数代替导数。因此，可以利用牛顿法有效地求出 $f(\bar{\beta}_i^*) = 0$ 的解。$\bar{\beta}_i^*$ 可以用下面的公式来更新：

$$\bar{\beta}_i^*(t+1) = \bar{\beta}_i^*(t) - \frac{f(\bar{\beta}_i^*(t))}{f'(\bar{\beta}_i^*(t))} \tag{4-20}$$

4.1.2 算法描述

FAGPP 算法基于数据的类簇信息使用学习的锚点图替换传统的基于图的降维算法学习的邻接图，加速了图的构建。利用原始数据的信息构建锚点隶属度矩阵来约束低维数据空间的锚点隶属度矩阵，期望在原始数据中构建的锚点图结构在投影空间中得到保留。FAGPP 算法是一个迭代的算法，在每次迭代的过程中通过不断优化投影矩阵 W、锚点隶属度矩阵 H 和投影锚点 $\tilde{O}$，不断降低目标函数的值。FAGPP 算法的描述如下：

算法 6: FAGPP 算法

输入： 数据集 X，参数 γ 和 λ，锚点数 m 和投影维度数 $\tilde{d}$

输出： 投影矩阵 W 和锚点隶属度矩阵 H

1. 计算数据 X 的协方差矩阵 S_t
2. 利用 BKHK 算法计算原始数据空间的锚点隶属度矩阵 J
3. 初始化锚点隶属度矩阵 $H: H \leftarrow J$
4. **When not convergence**
5. **For** each $\tilde{o}_k \in \tilde{O}$
6. $\tilde{o}_k = W^T(\sum_{i=1}^{n} h_{ki}x_i / \sum_{i=1}^{n} h_{ki}) = W^T O$
7. **end**
8. 计算对角矩阵 L^k，其第 i 个对角元素为 $l_{ii}^k = \sum_{k=1}^{m} h_{ki}$
9. 计算对角矩阵 L^i，其第 k 个对角元素为 $l_{kk}^i = \sum_{i=1}^{n} h_{ki}$
10. 计算 $B = XL^kX^T - 2XH\tilde{O}^T + \tilde{O}L^i\tilde{O}^T$，$\tilde{B} = (B + B^T)/2$
11. 利用公式(4-11)计算投影矩阵 W
12. 计算 $d_{ki} = \left\| W^T x_i - \tilde{o}_k \right\|_2^2$
13. 计算 $h_{ki} = j_{ki} - \gamma d_{ki}/2$
14. 更新 H,判断目标函数是否收敛
15. **Until convergence**
16. **Return** $\boldsymbol{W}$ 和 $\boldsymbol{H}$

为了更加清晰地描述 FAGPP 算法，下面有一张图来刻画原始数据空间的锚点图结构和投影空间的锚点图结构。

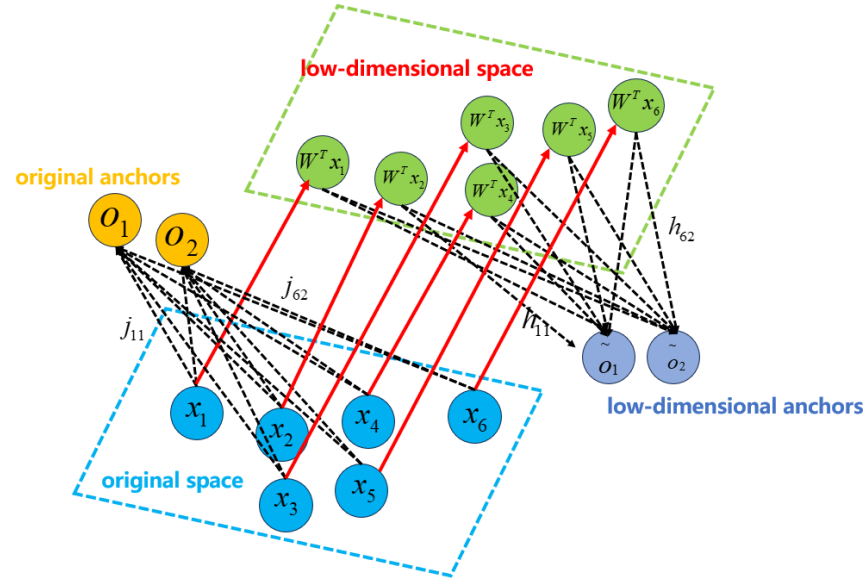


图 4.1 锚点图说明

图 4.1 中样本 x_1、x_2、x_3、x_4、x_5 和 x_6 为原始数据空间中的样本，样本 $W^T x_1$、$W^T x_2$、$W^T x_3$、$W^T x_4$、$W^T x_5$ 和 $W^T x_6$ 为投影空间中的样本，o_1 和 o_2 是原始数据空间中的锚点，$\tilde{o}_1$ 和 $\tilde{o}_2$ 是投影数据空间中的锚点。j_{11} 表示样本 x_1 到锚点 o_1 之间的隶属度，h_{62} 表示投影空间样本 $W^T x_6$ 到锚点 $\tilde{o}_2$ 的隶属度。$\min\|J-H\|_F^2$ 表示利用原始数据空间中的样本与锚点之间的关系，构建的锚点隶属度矩阵 J 来约束投影空间中构建的锚点隶属度矩阵 H，以期望在原始数据空间构建的锚点图结构在投影空间中得到保留。

4.2 收敛性分析

算法 6 采用迭代的方法不断降低目标函数的值，模型(4-3)的目标函数如下：

$$\begin{aligned} &J(W,O,H)=\min\|J-H\|_F^2+\gamma\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^T x_i-\tilde{o}_k\right\|_2^2-\lambda Tr(W^T S_t W) \\ &s.t.\ W^T W=I,\ h_{\cdot i}^{\ T}\mathbf{1}_m=1,\ H\geq 0 \end{aligned} \tag{4-21}$$

公式(4-21)中由 3 个独立项构成，其中第一项 $\|J-H\|_F^2\geq 0$，第二项 $\sum_{i=1}^{n}\sum_{k=1}^{m}h_{ki}\left\|W^T x_i-\tilde{o}_k\right\|_2^2\geq 0$。设 δ_i 为 S_t 的第 i 个最小特征值，在 $W^T W=I$ 约束下，可以得到 $-\lambda Tr(W^T S_t W)\geq -\lambda\sum_{i=1}^{\tilde{d}}\delta_i$。因此，目标函数 $J(W,O,H)\geq -\lambda\sum_{i=1}^{\tilde{d}}\delta_i$，显然，模型(4-3)的目标函数有一个下界 $-\lambda\sum_{i=1}^{\tilde{d}}\delta_i$。

假设模型(4-3)在第 i 次迭代后，其最优解为 $W^{(i)}$、$O^{(i)}$ 和 $H^{(i)}$，则公式(4-21)最优值为 $J(O^{(i)},H^{(i)},W^{(i)})$。模型(4-3)采用交替优化的方法对目标函数进行求解，在每次迭代有两个步骤：（1）固定 $H^{(i)}$，优化 $W^{(i)}$ 和 $O^{(i)}$；（2）固定 $W^{(i)}$ 和 $O^{(i)}$，优化 $H^{(i)}$。

在算法每次迭代的第一步，通过公式(4-5)对 O 求偏导获得最优值，因此可以得到以下不等式：

$$J(O^{(i+1)},H^{(i)},W^{(i)}) \le J(O^{(i)},H^{(i)},W^{(i)}) \tag{4-22}$$

在$W^TW=I$的正交约束下，通过求W的最优值，可以得到：

$$J(O^{(i+1)},H^{(i)},W^{(i+1)}) \le J(O^{(i+1)},H^{(i)},W^{(i)}) \tag{4-23}$$

结合不等式(4-22)和不等式(4-23)，可以得到：

$$J(O^{(i+1)},H^{(i)},W^{(i+1)}) \le J(O^{(i)},H^{(i)},W^{(i)}) \tag{4-24}$$

在算法每次迭代的第二步，在固定W和O的情况下优化求解H，可以得到不等式：

$$J(O^{(i+1)},H^{(i+1)},W^{(i+1)}) \le J(O^{(i+1)},H^{(i)},W^{(i+1)}) \tag{4-25}$$

将不等式(4-24)和不等式(4-25)结合可以得到：

$$J(O^{(i+1)},H^{(i+1)},W^{(i+1)}) \le J(O^{(i)},H^{(i)},W^{(i)}) \tag{4-26}$$

由不等式(4-26)可知，算法 6 在每次迭代中都会减小目标函数(4-21)的值。此外，由于目标函数具有下界$-\lambda\sum_{i=1}^{\tilde{d}}\delta_i$，因此算法 6 收敛于局部最优。

4.3 时间复杂度分析

假设数据集X是有包含 n 个样本，每个样本包含d个特征。FAGPP 算法首先利用 BKHK 算法计算原始数据空间中的隶属度矩阵J，其时间复杂度为$O(nd\zeta\log m)$，其中 m 为锚点数，ζ是一个常量。此后通过交替优化的方法迭代地优化模型，从而获得最优解。每次迭代主要包括以下三个部分：（1）计算锚点O的时间复杂度为$O(nmd)$；（2）在优化求解投影矩阵W的过程中，需要计算B，其时间复杂度为$O(nd\tilde{d})$；（3）优化求解锚点隶属度矩阵H的时间复杂度为$O(nmd)$。总的来说，FAGPP 算法总的时间复杂度为$O(nd\zeta\log m+t(nd\tilde{d}+nmd))$，其中 t 表示算法的迭代次数。当数据集的样本数量较大时，即在$n\gg\tilde{d}$，$n\gg m$和$n\gg d$的情况下，FAGPP 算法的时间复杂度与样本数量n线性相关。

4.4 实验

为了验证所提出的 FAGPP 算法的有效性，在 6 个基准数据集上进行了大量的实验。本章节选择了 5 个相关的算法作为对比算法，包括 LPP、NPE、AGLPP、LAPP 和 KaUDDR。本章的所有实验都是在一台处理器为 Intel(R) Core(TM) i9-10850K CPU、机带 RAM 为 32GB、MATLAB 2019b 和操作系统为 Windows10 的计算机上进行的。

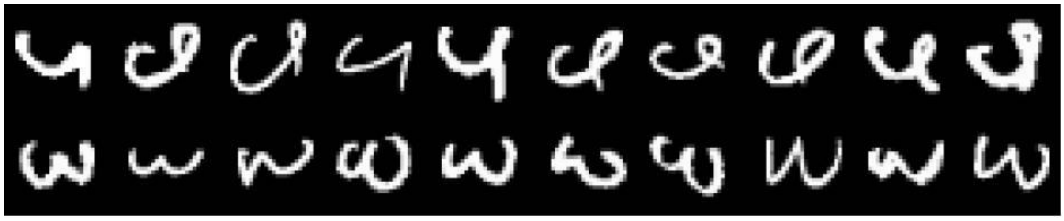
4.4.1 数据集

选择 6 个图像数据集进行实验，包括 AR、MINIST2k2k、Mnist05、PIE、umist 和 YaleB。表 4.1 展示了数据集的详细信息，包括样本量、特征、类簇、图片尺寸和英文缩写。

表 4.1 数据集信息

序号	数据集	样本量	特征数	类簇数	图片尺寸	简写
1	YaleB	2414	1024	38	32 × 32	YLB
2	umist	574	10304	20	112 × 92	UMI
3	AR	1680	1024	120	32 × 32	AR
4	PIE	11560	1024	68	32 × 32	PIE
5	Mnist05	3456	784	10	28 × 28	MST
6	MINIST2k2k	4000	784	10	28 × 28	MIN

本文在 3.5.1 节中已经详细介绍了 Yale、umist、AR 和 MINIST2k2k 数据集，Yale 数据集是 YaleB 数据集的一个子集。下面将详细 PIE 和 Mnist05 数据集。PIE 包含 11560 张 68 个人的灰色人脸图像，每个图像的尺寸为 32 × 32 像素。Mnist05 包含 3495 张手写数字图像，每张图像的像素为 28 × 28。从不同的数据集中随机选择了两个不同的类别，每个类别中随机选择了 10 张图像。具体的图像如图 4.2 所示。



(a)MST

(b)PIE

图 4.2 MST 和 PIE 图像数据集的样本图像

4.4.2 实验参数设置

本节将简单介绍实验的对比算法和实验的具体参数设置。在 3.5.3 节中已经对 LPP、NPE、AGLPP 和 LAPP 算法做了简单的介绍。KaUDDR 将自适应图学习和特征学习集成到一个联合学习框架中，对噪声样本具有鲁棒性，并能获得数据稳定的内在结构表示。KaUDDR 还定义了数据核和相似度指标核，使得其可以通过测量数据核和相似度指标核之间的一致性来有效地捕获数据的潜在几何结构。

为了消除原始数据协方差矩阵的零空间，提高计算效率，本文首先采用 PCA 算法作为预处理步骤。由于所有的对比算法都涉及到矩阵特征值的计算，为了实验的运行效率和一致性，使用 PCA 将数据的维数统一降维至 100 维。FAGPP、KaUDDR、AGLPP 和 LAPP 的迭代次数为 100 次。对于所有算法，本文设置维度 $\tilde{d} \in [10,100]$，搜索步长设置为 10。此外，AGLPP、LPP、NPE 和 LAPP 算法实验参数设置同 3.5.2 节保持一致。KaUDDR 算法和 FAGPP 算法的实验参数设置如下：（1）KaUDDR：调节因子 $\lambda \in \{n, 2n, 3n, 4n, 5n\}$，其中 n 为数据集的样本数量。构建邻接矩阵的参数 $k = 12$；（2）FAGPP：锚点的参数 $m \in \{63, 127, 255\}$，权衡因子 $\lambda, \gamma \in \{e-3, e-2, 1, e+2, e+3\}$。

4.4.3 不同维度实验分析

采用聚类精度（Accuracy）和归一化互信息（NMI）两个评价指标评估所提出的 FAGPP 算法的聚类性能，并在 6 个图像数据集上进行了实验。Accuracy 和 NMI 已经在 3.5.2 节进行了详细的介绍。首先利用所有算法在不同数据集上获得投影数据，再执行 100 次 *k*-means 算法，记录实验结果的标准差和平均值。本文选择最高的评价指标所在的维度作为最优的维度，采用平均值±标准差(最佳维度)的形式描述实验的结果如表 4.2 和表 4.3 所示，表中粗体字体表示最好的聚类性能指标数据。在 90%的置信水平上进行了 Wilcoxon 秩和检验。其中符号“+”、“-”分别表示 FAGPP 明显优于、劣于对比算法。符号“~”表示 FAGPP 算法与对比算法没有显著差异。此外，为了研究各个算法在不同维度下的降维效果，本文记录了所有算法在不同维度的最高的 Accuracy，实验结果如图 4.3 所示。

表 4.2 算法在 6 个基准数据集上的 Accuracy

Accuracy	AGLPP	KaUDDR	LAPP	NPE	LPP	FAGPP
UMI	0.4421(60) ±0.023	0.4321(20) ±0.026	0.4483(10) ±0.019	0.3075(40) ±0.019	0.4393(10) ±0.024	**0.5004(90)** ±0.035
AR	0.3049(60) ±0.006	0.3757(80) ±0.012	0.6465(60) ±0.012	0.6562(40) ±0.026	0.6505(60) ±0.021	**0.6602(90)** ±0.023
YLB	0.0917(10) ±0.001	0.1732(10) ±0.009	0.3829(90) ±0.025	0.3804(70) ±0.03	0.3735(100) ±0.018	**0.488(40)** ±0.022
PIE	0.0671(100) ±0.001	0.0843(100) ±0.004	0.2763(100) ±0.021	0.2656(100) ±0.017	0.2828(100) ±0.017	**0.327(80)** ±0.012
MIN	0.5413(40) ±0.024	0.5283(100) ±0.033	0.5054(10) ±0.02	0.4936(10) ±0.022	0.5176(10) ±0.017	**0.5453(70)** ±0.032
MST	0.3752(70) ±0.02	0.5558(90) ±0.036	0.5218(10) ±0.016	0.5056(10) ±0.028	0.5461(50) ±0.041	**0.5664(60)** ±0.038
平均值	0.3037	0.3582	0.4635	0.4348	0.4683	**0.5146**
平均方差	0.013	0.020	0.019	0.024	0.023	0.027
Wilcoxon	+	+	~	+	~	**N/A**

从表 4.2、表 4.3 和图 4.3 的实验结果中可以得到以下结论：

（1）与对比算法相比，FAGPP 算法在所有 6 个图像数据集上都取得了最佳的聚类精度。FAGPP 算法相较于 AGLPP、KaUDDR、LAPP、NPE 和 LPP 算法，其平均 Accuracy 分别高了 21.09%、15.64%、5.11%、7.98%和 4.63%，其平均 NMI 分别高了 20.9%、13.47%、2.77%、8.18%和 2.82%。实验结果表明，

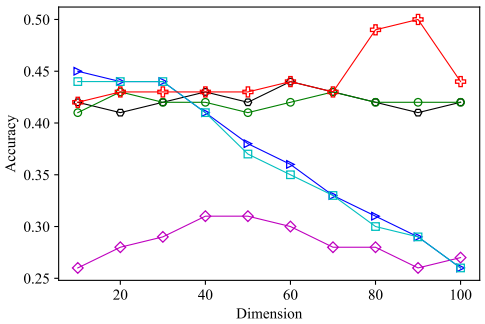
与对比算法相比，本文提出的 FAGPP 算法能够在保持原始数据中锚点图结构信息的前提下学习最优投影矩阵，构造的锚点图结构有助于降维。

（2）由表 4.2 和表 4.3 的统计检验结果表明，FAGPP 算法显著优于 AGLPP、KaUDDR 和 NPE 算法，与 LAPP 和 LPP 算法没有显著差异。实验结果证明了 FAGPP 算法的有效性。

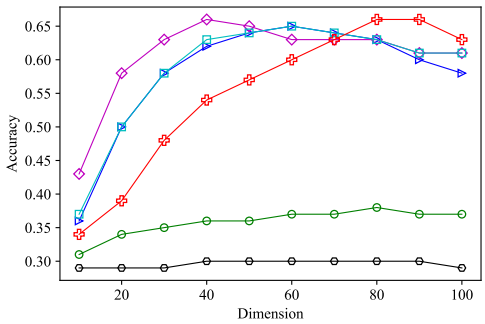
表 4.3 算法在 6 个基准数据集上的 NMI

NMI	AGLPP	KaUDDR	LAPP	NPE	LPP	FAGPP
UMI	0.6211(60) ±0.015	0.6482(20) ±0.022	0.6671(10) ±0.007	0.4084(40) ±0.02	0.6669(10) ±0.012	**0.6719(90)** ±0.03
AR	0.6391(60) ±0.003	0.6959(80) ±0.006	0.8677(60) ±0.005	0.8757(50) ±0.01	0.8689(70) ±0.008	**0.8835(90)** ±0.008
YLB	0.133(10) ±0.001	0.271(10) ±0.008	0.4966(90) ±0.019	0.5051(70) ±0.015	0.4896(70) ±0.013	**0.5689(80)** ±0.013
PIE	0.1486(100) ±0.003	0.1851(80) ±0.003	0.4104(100) ±0.016	0.4012(100) ±0.017	0.4129(100) ±0.015	**0.4654(80)** ±0.008
MIN	**0.4984(100)** ±0.014	0.4585(40) ±0.018	0.4665(10) ±0.014	0.4284(10) ±0.01	0.466(10) ±0.01	0.4742(70) ±0.018
MST	0.2725(70) ±0.009	0.4998(90) ±0.017	0.4922(10) ±0.014	0.4575(10) ±0.013	0.4935(50) ±0.02	**0.5028(50)** ±0.018
平均值	0.3855	0.4598	0.5668	0.5127	0.5663	**0.5945**
平均方差	0.0075	0.0123	0.0125	0.0142	0.0130	0.0158
Wilcoxon	+	+	~	+	~	**N/A**

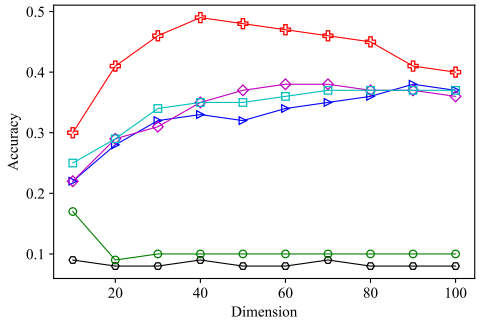
（3）由图 4.3 可以看出，总体而言，在 6 个基准图像数据集上，FAGPP 的聚类精度明显优于比较算法。在获得最优性能后，FAGPP 的分类精度随维数的增加变化不大，实验证明了 FAGPP 算法的鲁棒性。



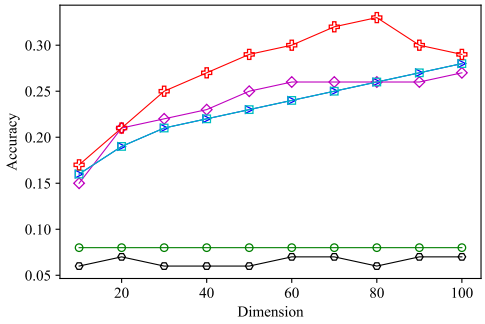
(a)UMI



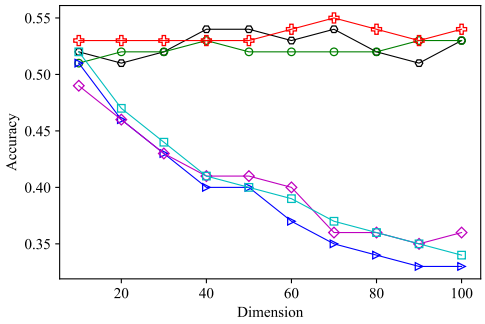
(b)AR



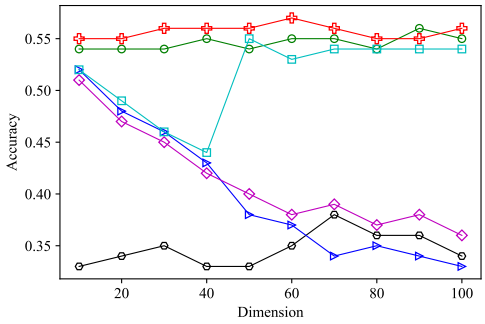
(c)YLB



(d)PIE



(e)MIN



(f)MST

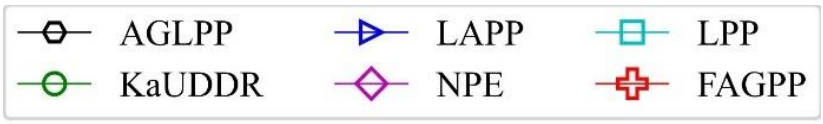
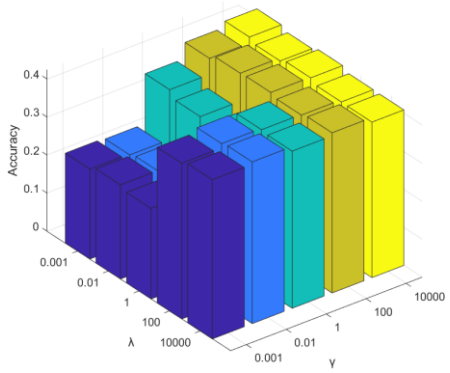


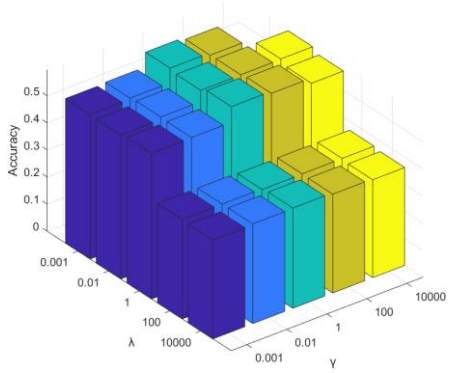
图 4.3 算法在不同维度上的 Accuracy

4.4.4 参数敏感性分析

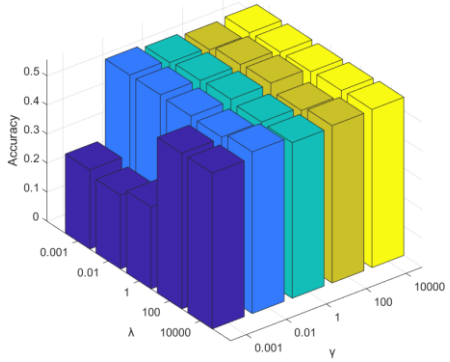
FAGPP 算法包含两个参数，锚点数 m，权衡因子 γ 和 λ。权衡因子 γ 表示模型(4-3)中第二项 $\sum_{i=1}^{n}\sum_{k=1}^{m} h_{ki}\left\|W^T x_i - \tilde{o}_k\right\|_2^2$ 的重要程度，权衡因子 λ 表示模型(4-3)中第三项 $Tr(W^T S_t W)$ 的重要程度。为了探究两个权衡因子对模型的敏感性，本实验通过固定锚点数来找出权衡因子 γ 和 λ 对模型的影响。在参数敏感性分析实验中，固定锚点数 $m = 63$，选择参数 $\lambda,\gamma \in \{e-3, e-2, 1, e+2, e+3\}$。FAGPP 算法对 λ 和 γ 不同值的聚类精度如图 4.4 所示。



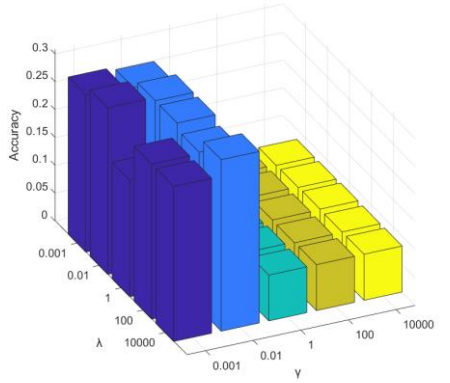
(a)UMI



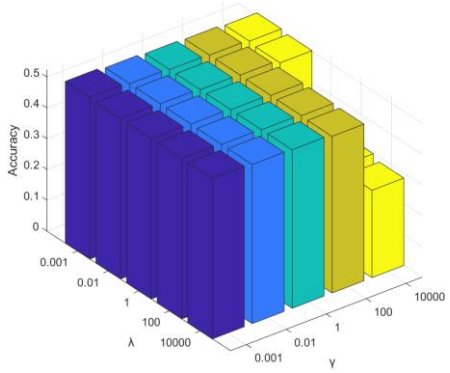
(b)AR



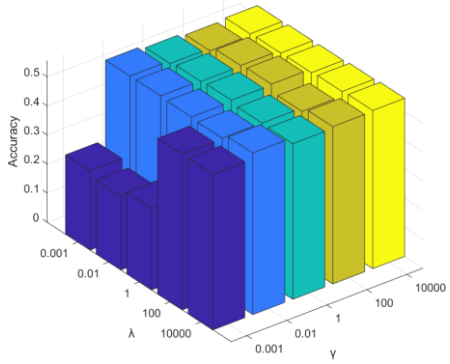
(c)YLB



(d)PIE



(e)MIN



(f)MST

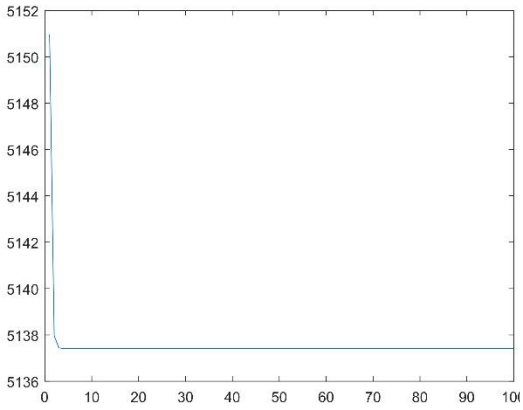
图 4.4 参数 λ 和参数 γ 对 FAGPP 算法聚类精度的影响

从图 4.4 中可以发现，FAGPP 算法在 YLB 和 PIE 数据集上对参数 λ 和参数 γ 比较敏感，随着权衡因子 λ 和 γ 的增加，FAGPP 算法的聚类精度大幅度下降。此外有，UMI 和 MST 数据集上，权衡因子 λ 越大，算法的聚类精度越高。相反，

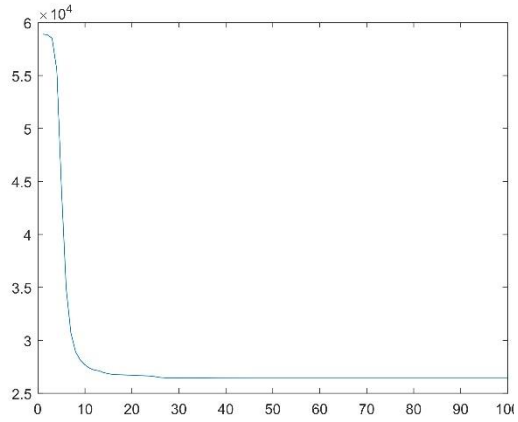
在 MIN 和 AR 数据集上，较小的γ能获得较大的聚类精度。

4.4.5 收敛性实验

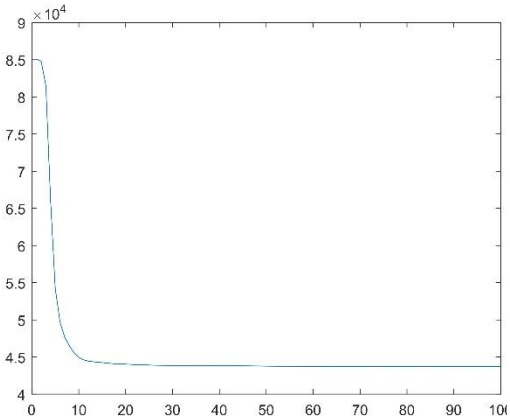
定理 2 从理论上证明了 FAGPP 算法的收敛性。为了证明理论分析的准确，本节将从实验的角度证明所提算法的收敛性。具体来说，通过固定参数来统计不同迭代次数的目标函数值，在六个图像数据集进行了收敛实验。首先固定 FAGPP 算法获得最优性能时的维度$\tilde{d}$，设置权衡因子$\gamma = 0.1$，权衡因子$\lambda = 0.01$和锚点数$m = 63$。图 4.5 中展示了算法在不同数据集上的收敛曲线。



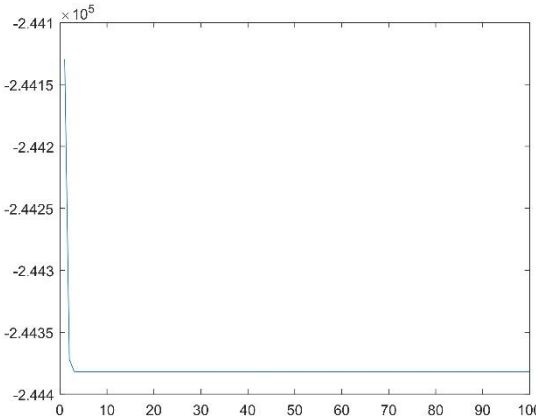
(a)UMI



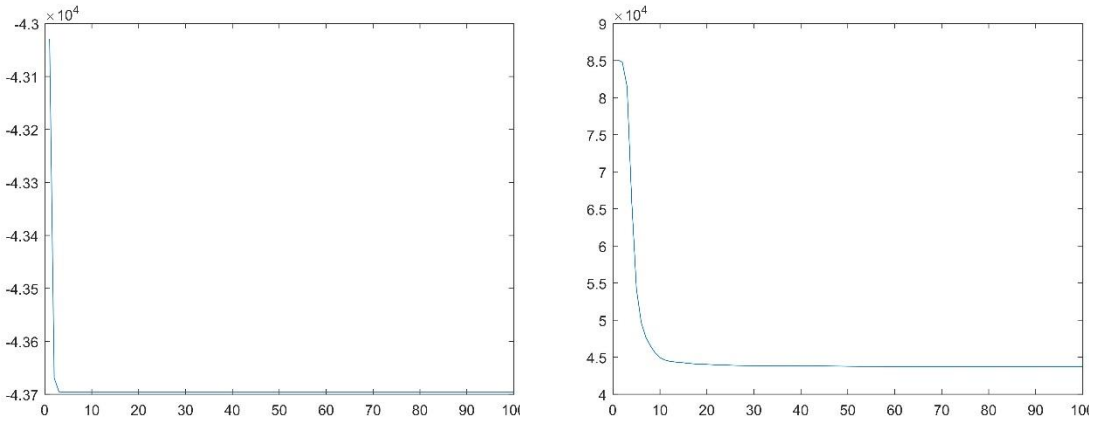
(b)AR



(c)YLB



(d)PIE



(e)MIN (f)MST

图 4.5 FAGPP 算法 6 个数据集上的收敛曲线

从图 4.5 中可以看出：

（1）随着迭代次数的增加，目标函数的值逐渐减小最终收敛。实验结果与 4.2 节中算法的收敛性分析一致，证明了 FAGPP 算法的收敛性。

（2）当迭代次数达到 20 次时，FAGPP 算法在大多数数据集的收敛曲线趋于稳定目标函数实现收敛。实验的结果表明 FAGPP 算法的收敛速度非常快，具有广泛的应用前景。

4.4.6 运行时间实验

时间复杂度是衡量算法效率的一个重要指标，本文在 4.3 节对 FAGPP 算法的时间复杂度进行了理论分析。在本章节将对所有算法在 UMI、AR、YLB、PIE、MIN 和 MST 数据集上进行了运行时间的实验。实验结果如表 4.4 所示。括号中的数字表示相应算法的运行时间排序。数字越小，表示算法的运行速度越快。另外，由于 FAGPP、KaUDDR 和 LAPP 都是迭代的算法，为了保证实验的公平性，只记录了这三种算法一次迭代的运行时间。

表 4.4 每种算法在不同数据集上的平均运行时间(毫秒)

Time	AGLPP	KaUDDR	LAPP	NPE	LPP	FAGPP
UMI	18.94(2)	225.47(6)	69.02(5)	46.88(4)	40.18(3)	**14.01(1)**
AR	92.33(2)	1967.15(6)	338.82(4)	703.12(5)	323.66(3)	**87.32(1)**
YLB	72.92(2)	14430.66(6)	748.85(3)	1781.2(5)	781.25(4)	**61.6(1)**
PIE	163.1(2)	1198494.(6)	9042.19(5)	34937(3)	40906.2(4)	**157.3(1)**
MIN	404.7(2)	97451.26(6)	2090.83(3)	4578.12(5)	2263.39(4)	**303.15(1)**
MST	72.92(2)	58552.78(6)	10789.32(5)	3500(4)	1643.47(3)	**65.64(1)**
平均排名	2	6.0	4.2	4.3	3.5	1

从表 4.4 中可以看出，相较于对比算法，FAGPP 在所有 6 个不同数据集上都具有最短的运行时间，FAGPP 算法的运行速度平均排名第一。这是因为 FAGPP 算法利用原始数据的类簇信息构建锚点隶属度矩阵和锚点图结构，其总的时间复杂度为 $O(nd\zeta \log m + t(nd\tilde{d} + nmd))$，算法的时间复杂度与样本数量 n 近似成线性相关，其中 n 为数据集的样本数量，d 为数据集的特征数，$\tilde{d}$ 为投影数据空间的维度，m 为锚点数。AGLPP 算法的平均排名第二，因为虽然 AGLPP 也是通过构建了锚点图来保持数据的局部信息，但是 AGLPP 采用聚类方法获得一组聚类中心作为虚拟锚点，消耗了一定的计算量。总的来说，实验结果与理论分析一致。

4.5 本章小结

本章提出了一种快速锚图保投影降维算法（FAGPP），该算法在优化目标函数的过程中同时学习投影矩阵和锚点隶属度矩阵。FAGPP 充分利用数据的类簇信息学习锚点图结构，一方面构建的锚点图用于捕获原始数据的分布信息；另一方面，降维过程中学习到的锚点图可以约束保留预先构造的锚图的结构信息。此外，为了处理原始数据的全局信息，FAGPP 算法嵌入了 PCA 算法。因此，FAGPP 算法在捕获原始数据类簇信息的同时，也捕获了数据的全局信息。FAGPP 算法的时间复杂度为 $O(nmd)$，与样本数量成线性关系，使得该方法适用于高维大数据集。为了证明所提算法的有效性，在 6 个图像数据集上与 5 个相关的对比算法进行了大量的聚类实验，实验结果证明了 FAGPP 算法良好的聚类效果。

5 总结与展望

5.1 本文工作总结

随着经济社会的发展，数据以超出想象的速度在增长，面对海量的高维数据，如何高效地剔除数据的冗余特征并从中挖掘出有价值的信息具有重要的研究价值。由于高维数据中的样本过于稀疏、噪声样本多等问题，传统的聚类方法在高维数据中难以获得好的性能。本文对现有降维算法和高维聚类算法的研究现状进行了详细地描述，现有的高维数据聚类算法普遍存在的两个问题：（1）对异常样本敏感，缺乏识别和处理异常样本的过程；（2）需要预先对高维数据进行降维处理，算法的时间复杂度通常较高，不适用于高维的大数据集。针对这些问题，本文提出了两个新颖的高维数据聚类模型。第一个模型是带有实例惩罚的投影模糊 C 均值聚类算法（PCIP），该算法首先基于原始数据的信息构建实例惩罚矩阵，为每个样本分配实例惩罚系数，从而减少异常样本对模型的影响。PCIP 算法还将聚类任务和降维任务统一到一个目标函数中，迭代优化投影矩阵和隶属度矩阵。本文还从理论上证明了 PCIP 算法的收敛性分析和并对其进行了时间复杂度分析，PCIP 的时间复杂度于样本的数量 n 线性相关。为了证明所提算法的有效性，选择聚类精度（Accuracy）和归一化互信息（NMI）作为聚类评价指标，在 10 个图像数据集上与 7 个相关的对比算法进行了大量的实验，实验结果证明了 PCIP 算法能有效处理高维的大数据集，在投影空间中获得好的聚类效果。

第二个模型是快速锚点图保持投影算法（FAGPP）。现有的一些基于图的降维算法如 LPP 和 LPFCM 利用原始数据的信息预先构建邻接矩阵和邻接图，其具有两个问题：（1）构建邻接图的时间复杂度不低于 $O(n^2d)$，不适用于高维的大数据集；（2）在对数据进行降维之前需要提前构建邻接图，图的质量一般决定了模型的效果，难以保证构建的图结构适用于随后的降维任务。为了解决以上问题，FAGPP 算法首先使用锚点图的学习代替邻接图的学习，减少基于图的降维算法的时间复杂度。FAGPP 算法还融合了 PCA 模型使其不仅可以处理数据的聚类信息，还可以处理数据的全局信息。此外，FAGPP 构建了一个类似

FCM 模型，在降维的过程中迭代地优化投影矩阵和锚点隶属度矩阵。本文还从理论上证明了 FAGPP 算法的收敛性并分析了算法的时间复杂度，FAGPP 算法的时间复杂度与样本的数量线性相关，能够高效地处理高维的大数据集。在 6 个图像数据集上与 5 个相关的对比算法进行实验，实验结果证明了 FAGPP 算法的聚类效果。

5.2 后续工作展望

针对现有高维聚类算法存在的问题，本文提出了两个新颖的模型 PCIP 和 FAGPP，这两个模型不仅能高效地处理高维的大数据集，还能在投影空间中获得良好的类簇结构，相较于传统的聚类算法提升了聚类精度，具有良好的应用前景。然而这两个模型还存在以下问题：（1）为了识别噪声样本，PCIP 算法需要提前构建实例惩罚矩阵，实例惩罚矩阵的质量决定了模型的性能；（2）FAGPP 算法也需要提前指定锚点的数量，模型对锚点数比较敏感。针对第一个问题，考虑将实例惩罚矩阵的学习嵌入模型优化中，利用投影空间样本间的信息和类簇信息自适应学习实例惩罚矩阵，使得算法能更加高效和准确地识别异常样本，提升模型的聚类性能。针对第二个问题，未来的研究将围绕如何在降维的过程中利用数据的信息自适应地确定锚点数量，进而学习动态的锚点图，以期望构建的锚点图结构能最大保留原始数据分布的信息。

参考文献

[1] Ayesha S, Hanif M K, Talib R.Overview and comparative study of dimensionality reduction techniques for high dimensional data[J].Information Fusion,2020, 59: 44-58.

[2] Agrawal R, Gehrke J, Gunopulos D, Raghavan P.Automatic subspace clustering of high dimensional data[J].Data Mining and knowledge discovery,2005, 11: 5-33.

[3] Thrun M C, Ultsch A.Using projection-based clustering to find distance-and density-based clusters in high-dimensional data[J].Journal of Classification,2021, 38 (2): 280-312.

[4] Raymer M L, Punch W F, Goodman E D, Kuhn L A, Jain A K.Dimensionality reduction using genetic algorithms[J].IEEE transactions on evolutionary computation,2000, 4 (2): 164-171.

[5] Engel D, Hüttenberger L, Hamann B. A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization[C]//Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011,2012.

[6] 张蕾, 崔勇, 刘静, 江勇, 吴建平.机器学习在网络空间安全研究中的应用[J].计算机学报,2018, 41 (09): 1943-1975.

[7] Zhu X, Huang Z, Yang Y, Shen H T, Xu C, Luo J.Self-taught dimensionality reduction on the high-dimensional small-sized data[J].Pattern Recognition,2013, 46 (1): 215-229.

[8] Liao S-H, Chu P-H, Hsiao P-Y.Data mining techniques and applications–A decade review from 2000 to 2011[J].Expert systems with applications,2012, 39 (12): 11303-11311.

[9] Xu D, Tian Y.A comprehensive survey of clustering algorithms[J].Annals of data science,2015, 2: 165-193.

[10] Nielsen F, Nielsen F.Hierarchical clustering[J].Introduction to HPC with MPI for Data Science,2016: 195-211.

[11] Oyelade J, Isewon I, Oladipupo O, Emebo O, Omogbadegun Z, Aromolaran O, Uwoghiren E, Olaniyan D, Olawole O. Data clustering: Algorithms and its applications[C]//2019 19th International Conference on Computational Science and Its Applications (ICCSA),2019: 71-81.

[12] Hartigan J A, Wong M A.Algorithm AS 136: A k-means clustering algorithm[J].Journal of the royal statistical society. series c (applied statistics),1979, 28 (1): 100-108.

[13] Bezdek J C, Ehrlich R, Full W.FCM: The fuzzy c-means clustering algorithm[J].Computers & geosciences,1984, 10 (2-3): 191-203.

[14] Rodriguez A, Laio A.Clustering by fast search and find of density peaks[J].science,2014, 344 (6191): 1492-1496.

[15] Indyk P, Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]//Proceedings of the thirtieth annual ACM symposium on Theory of computing,1998: 604-613.

[16] Wang J, Wu Y, Li S, Nie F.A self-training algorithm based on the two-stage data editing method with mass-based dissimilarity[J].Neural Networks,2023, 168: 431-449.

[17] 万静, 吴凡, 何云斌, 李松.新的降维标准下的高维数据聚类算法[J].计算机科学与探索,2020, 14 (01): 96-107.

[18] 胡洁.高维数据特征降维研究综述[J].计算机应用研究,2008 (09): 2601-2606.

[19] Huang L, Chen C, Li W, Du Q.Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors[J].Remote Sensing,2016, 8 (6): 483.

[20] Oliva A, Torralba A.Modeling the shape of the scene: A holistic representation of the spatial envelope[J].International journal of computer vision,2001, 42: 145-175.

[21] Alelyani S, Tang J, Liu H.Feature selection for clustering: A review[J].Data Clustering,2018: 29-60.

[22] Dash M, Liu H.Consistency-based search in feature selection[J].Artificial intelligence,2003, 151 (1-2): 155-176.

[23] Pudil P, Novovičová J, Kittler J.Floating search methods in feature

selection[J].Pattern recognition letters,1994, 15 (11): 1119-1125.

[24] Wold S, Esbensen K, Geladi P.Principal component analysis[J].Chemometrics and intelligent laboratory systems,1987, 2 (1-3): 37-52.

[25] Xanthopoulos P, Pardalos P M, Trafalis T B, Xanthopoulos P, Pardalos P M, Trafalis T B.Linear discriminant analysis[J].Robust data mining,2013: 27-33.

[26] He X, Cai D, Yan S, Zhang H-J. Neighborhood preserving embedding[C]//Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1,2005: 1208-1213.

[27] Hammouche R, Attia A, Akhrouf S, Akhtar Z.Gabor filter bank with deep autoencoder based face recognition system[J].Expert Systems with Applications,2022, 197: 116743.

[28] He X, Yan S, Hu Y, Niyogi P, Zhang H-J.Face recognition using laplacianfaces[J].IEEE transactions on pattern analysis and machine intelligence,2005, 27 (3): 328-340.

[29] Basak S, Kar S, Saha S, Khaidem L, Dey S R.Predicting the direction of stock market prices using tree-based classifiers[J].The North American Journal of Economics and Finance,2019, 47: 552-567.

[30] Rosipal R, Girolami M, Trejo L J, Cichocki A.Kernel PCA for feature extraction and de-noising in nonlinear regression[J].Neural Computing & Applications,2001, 10: 231-243.

[31] Zhang Y, Li B, Wang Z, Wang W, Wang L.Fault diagnosis of rotating machine by isometric feature mapping[J].Journal of Mechanical Science and Technology,2013, 27: 3215-3221.

[32] Moradzadeh A, Pourhossein K, Mohammadi-Ivatloo B, Mohammadi F.Locating inter-turn faults in transformer windings using isometric feature mapping of frequency response traces[J].IEEE Transactions on Industrial Informatics,2020, 17 (10): 6962-6970.

[33] Belkin M, Niyogi P.Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering[J].Advances in Neural Information Processing Systems,2001, 14 (6): 585-591.

[34] 殷俊, 金忠.图最优化线性鉴别投影及其在图像识别中的应用[J].模式识别与人工智能,2011, 24 (05): 658-664.

[35] 徐剑豪, 胡文军, 王哲昀, 胡天杰.最近邻子空间保持的特征提取方法[J].计算机应用与软件,2024, 41 (02): 293-299.

[36] He X.Locality preserving projections[J].Advances in Neural Information Processing Systems,2003, 16 (1): 186-197.

[37] Yan S, Xu D, Zhang B, Zhang H-J, Yang Q, Lin S.Graph embedding and extensions: A general framework for dimensionality reduction[J].IEEE transactions on pattern analysis and machine intelligence,2006, 29 (1): 40-51.

[38] 王继奎, 杨正国, 易纪海, 刘学文, 王会勇, 聂飞平.稀疏约束的嵌入式模糊均值聚类算法[J].复旦学报(自然科学版),2020, 59 (06): 725-733.

[39] Yang Z, Wang J, Li Q, Yi J, Liu X, Nie F.Graph optimization for unsupervised dimensionality reduction with probabilistic neighbors[J].Applied Intelligence,2023, 53 (2): 2348-2361.

[40] Zhang L, Qiao L, Chen S.Graph-optimized locality preserving projections[J].Pattern Recognition,2010, 43 (6): 1993-2002.

[41] Yi Y, Wang J, Zhou W, Fang Y, Kong J, Lu Y.Joint graph optimization and projection learning for dimensionality reduction[J].Pattern Recognition,2019, 92: 258-273.

[42] Kang Z, Zhou W, Zhao Z, Shao J, Han M, Xu Z. Large-scale multi-view subspace clustering in linear time[C]//Proceedings of the AAAI conference on artificial intelligence,2020: 4412-4419.

[43] Liu W, He J, Chang S-F. Large graph construction for scalable semi-supervised learning[C]//Proceedings of the 27th international conference on machine learning (ICML-10),2010: 679-686.

[44] Lu X, Feng S.Structure diversity-induced anchor graph fusion for multi-view clustering[J].ACM Transactions on Knowledge Discovery from Data,2023, 17 (2): 1-18.

[45] Jiang R, Fu W, Wen L, Hao S, Hong R.Dimensionality reduction on anchorgraph with an efficient locality preserving projection[J].Neurocomputing,2016, 187:

109-118.

[46] Zhang R, Zhang Y, Lu C, Li X.Unsupervised graph embedding via adaptive graph learning[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2022, 45 (4): 5329-5336.

[47] Chen Y, Lai Z, Ding Y, Lin K, Wong W K. Deep supervised hashing with anchor graph[C]//Proceedings of the IEEE/CVF international conference on computer vision,2019: 9796-9804.

[48] 朱建勇, 李兆祥, 徐彬, 杨辉, 聂飞平.基于图嵌入的正交局部保持投影无监督特征选择[J].计算机科学,2023, 50 (S2): 552-560.

[49] 王继奎, 杨正国, 刘学文, 易纪海, 李冰, 聂飞平.一种基于极大熵的快速无监督线性降维方法[J].软件学报,2023, 34 (04): 1779-1795.

[50] Mousazadeh S, Cohen I.Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering[J].IEEE Transactions on Audio Speech and Language Processing,2013, 21 (6): 1261-1271.

[51] Hagen L, Kahng A B.New spectral methods for ratio cut partitioning and clustering[J].IEEE transactions on computer-aided design of integrated circuits and systems,1992, 11 (9): 1074-1085.

[52] Nie F, Ding C, Luo D, Huang H. Improved minmax cut graph clustering with nonnegative relaxation[C]//Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II 21,2010: 451-466.

[53] Van De Velden M, D'enza A I, Yamamoto M.Special feature: dimension reduction and cluster analysis[J].Behaviormetrika,2019, 46: 239-241.

[54] De Soete G, Carroll J D: K-means clustering in a low-dimensional Euclidean space,New approaches in classification and data analysis: Springer,1994: 212-219.

[55] Vichi M, Kiers H A.Factorial k-means analysis for two-way data[J].Computational Statistics & Data Analysis,2001, 37 (1): 49-64.

[56] Wang J, Yang Z, Liu X, Li B, Yi J, Nie F.Projected fuzzy C-means with probabilistic neighbors[J].Information Sciences,2022, 607: 553-571.

[57] Zhou J, Pedrycz W, Yue X, Gao C, Lai Z, Wan J.Projected fuzzy C-means

clustering with locality preservation[J].Pattern Recognition,2021, 113: 107748.

[58] Shen X-J, Liu S-X, Bao B-K, Pan C-H, Zha Z-J, Fan J.A generalized least-squares approach regularized with graph embedding for dimensionality reduction[J].Pattern Recognition,2020, 98: 107023.

[59] Gao Y, Luo S, Pan J, Wang Z, Gao P.Kernel alignment unsupervised discriminative dimensionality reduction[J].Neurocomputing,2021, 453: 181-194.

[60] Wang A, Zhao S, Liu J, Yang J, Liu L, Chen G.Locality adaptive preserving projections for linear dimensionality reduction[J].Expert Systems with Applications,2020, 151: 113352.

[61] Meng Y, Liang J, Cao F, He Y.A new distance with derivative information for functional k-means clustering algorithm[J].Information Sciences,2018, 463: 166-185.

[62] Zhu J, Jiang Z, Evangelidis G D, Zhang C, Pang S, Li Z.Efficient registration of multi-view point sets by K-means clustering[J].Information Sciences,2019, 488: 205-218.

[63] He X, Niyogi P.Locality preserving projections[J].Advances in neural information processing systems,2003, 16.

[64] Liu F T, Ting K M, Zhou Z-H. Isolation forest[C]//2008 eighth ieee international conference on data mining,2008: 413-422.

[65] Sleator D D, Tarjan R E.Self-adjusting binary search trees[J].Journal of the ACM (JACM),1985, 32 (3): 652-686.

[66] Bell J, Gupta G.An evaluation of self‐adjusting binary search tree techniques[J].Software: Practice and Experience,1993, 23 (4): 369-382.

[67] Bronson N G, Casper J, Chafi H, Olukotun K.A practical concurrent binary search tree[J].ACM Sigplan Notices,2010, 45 (5): 257-268.

[68] Nie F, Zhu W, Li X.Unsupervised large graph embedding based on balanced and hierarchical k-means[J].IEEE Transactions on Knowledge and Data Engineering,2020, 34 (4): 2008-2019.

[69] Yang J, Zhang D, Frangi A F, Yang J-Y.Two-dimensional PCA: a new approach to appearance-based face representation and recognition[J].IEEE transactions on

pattern analysis and machine intelligence,2004, 26 (1): 131-137.

[70] Xiong H, Swamy M, Ahmad M O.Two-dimensional FLD for face recognition[J].Pattern Recognition,2005, 38 (7): 1121-1124.

[71] Phillips P J, Wechsler H, Huang J, Rauss P J.The FERET database and evaluation procedure for face-recognition algorithms[J].Image and vision computing,1998, 16 (5): 295-306.

[72] Jiang B, Ding C, Luo B, Tang J. Graph-Laplacian PCA: Closed-form solution and robustness[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2013: 3492-3498.

[73] Chan T-H, Jia K, Gao S, Lu J, Zeng Z, Ma Y.PCANet: A simple deep learning baseline for image classification?[J].IEEE transactions on image processing,2015, 24 (12): 5017-5032.

[74] Wang J, Wang L, Nie F, Li X.A novel formulation of trace ratio linear discriminant analysis[J].IEEE Transactions on Neural Networks and Learning Systems,2021, 33 (10): 5568-5578.

[75] Dudoit S, Fridlyand J.Bagging to improve the accuracy of a clustering procedure[J].Bioinformatics,2003, 19 (9): 1090-1099.

[76] Liese F, Vajda I.On divergences and informations in statistics and information theory[J].IEEE Transactions on Information Theory,2006, 52 (10): 4394-4412.

[77] Estévez P A, Tesmer M, Perez C A, Zurada J M.Normalized mutual information feature selection[J].IEEE Transactions on neural networks,2009, 20 (2): 189-201.

[78] Cover T M, Thomas J A.Information theory and statistics[J].Elements of information theory,1991, 1 (1): 279-335.

[79] Nie F, Wang X, Huang H. Clustering and projected clustering with adaptive neighbors[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining,2014: 977-986.

[80] Zimmerman D W, Zumbo B D.Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks[J].The Journal of Experimental Educational,1993: 75-86.

致谢

路漫漫其修远兮，吾将上下而求索。回想三年前，我怀着忐忑的心来到兰州财经大学，立誓要学有所成，不负韶华。三年时光转瞬即逝，在这里我不但收获了专业素养的提高，同时带给我的是性格的磨砺和品性的塑造。能走到今天，我向所有指导过我、关心过我和批评过我的老师、同学和亲人表达由衷的感谢！

首先感谢我的指导老师聂飞平教授和王继奎教授。初来实验室懵懵懂懂，老师给予了我细心的指导。聂飞平教授和王继奎教授不仅在专业和学术上开阔了我的视野，而且在为人处事上对我进行教导。在每周的组会上，王老师对我大论文和小论文的选题、开题、构思和论文撰写等各个环节都提出了珍贵的意见，从而使我能够顺利的完成大论文和小论文的课题研究。另外，聂老师和王老师的治学严谨，学术的渊博，谦虚严谨的态度，脚踏实地的作风和刚正不阿的品行也给带我来了人生启迪。因此，我对聂飞平老师和王继奎老师表达最诚挚的敬意和最由衷的感谢！

感谢我的父亲吴俊峰和母亲周宜侠的养育之恩，你们一直坚定不移尊重、支持我的选择，并给予我无微不至的照顾。你们日以继夜，不辞辛苦的工作换来了我现在的一切！

感谢兰州财经大学 LIST 团队的杨正国老师、易纪海老师、武根强老师、尚庆生老师和赵瑞娟老师给予的指导和帮助。感谢实验室刘学文师兄、李冰师姐、段会雨、张翠红、黄雪艳、赵薇、吉成竹、李茜苒、刘飞飞、刘飘飘和李梦瑶同学的无私帮助！感谢我的室友陈贵富、赵金雨、韩运龙和孙梦泽，从陌生到熟悉，回到寝室后总会有无限的欢乐，从不觉得无趣。

最后，我要感谢百忙之中抽出时间来评审我的论文的各位专家教授以及答辩委员会的老师们，感谢你们对本文的指导与宝贵意见！

攻读硕士学位期间发表的论文及科研情况

A．作者在攻读学位期间的发表的论文

[1] Jikui W, **Yiwen W**, Bing L, Zhenguo Y, Feiping N. Fast anchor graph preserving projections[J].Pattern Recognition, 2024, 146.（中科院分区计算机科学 1 区）

[2] Jikui W, **Yiwen W**, Shaobo L, Feiping N. A self-training algorithm based on the two-stage data editing method with mass-based dissimilarity[J].Neural networks : the official journal of the International Neural Network Society, 2023, 168: 431-449.（中科院分区计算机科学 1 区）

[3] **Yiwen W**, Huiyu D, Jikui W. Identification model of poverty-prone population and analysis of poverty-causing factors[C]//International Conference on Intelligent Systems, Communications, and Computer Networks (ISCCN 2022). SPIE, 2022, 12332: 465-469.

B．作者在攻读学位期间参与的科研项目

[1] 甘肃省自然科学基金项目“高维数据的鲁棒半监督多标签学习算法研究”(项目编号 22JR5RA554)，参与

[2] 甘肃省高等学校创新基金项目“类簇结构保持的快速无监督线性降维算法研究”(项目编号 2022A-092)，参与

[3] 公共大数据国家重点实验室开放课题项目“图优化降维和聚类融合学习模型与算法研究”（项目编号 GZU-PBD2021-101），参与