分类号：TP391　　学校代码：10363
密　级：公开　　学　　号：2220910105

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目　基于自表示学习的子空间聚类算法研究

论 文 作 者　阚耀祖
校 内 导 师　卢桂馥 教授
校 外 导 师　段中华 高级工程师
专业学位类别　计算机技术
研究方向（领域）　智能系统与应用

论文提交日期：2025 年 5 月 9 日

分类号：TP391　　单位代码：10363
密 级：公开　　学　号：2220910105

题　目　基于自表示学习的子空间聚类算法研究

英文并列题目　Research on subspace clustering algorithm based on self-representation learning

论 文 作 者：阚耀祖
校 内 导 师：卢桂馥 教授
校 外 导 师：段中华 高级工程师

专业学位类别：计算机技术
研究方向（领域）：智能系统与应用

论文答辩日期：2025 年 5 月 10 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：

日期：2025 年 5 月 9 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在_____年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：

日期：2025 年 5 月 9 日

指导教师签名：

日期：2025 年 5 月 9 日

基于自表示学习的子空间聚类算法研究

摘 要

随着大数据时代的到来，数据的维度和复杂性显著增加，传统聚类方法在处理高维数据时面临巨大挑战。子空间聚类作为一种有效的无监督学习技术，能够从高维数据中提取低维结构信息，近年来受到广泛关注。然而，现有子空间聚类算法在处理复杂数据结构时仍存在局限性，如难以处理非线性相关的数据，计算效率较低等问题。因此，针对现有的子空间聚类算法存在的部分局限，本文设计了三种创新的子空间聚类算法，主要的研究内容和创新点如下：

(1)针对传统的子空间聚类算法存在的难以处理非线性相关的数据以及字典固定的问题，本文提出了一种基于自适应核字典的低秩表示子空间聚类方法（An adaptive kernel dictionary based low-rank representation method for subspace clustering，AKDLRR）。首先，AKDLRR 通过引入核技巧，将原始数据映射到一个新的特征空间，从而将非线性相关数据的聚类问题转化为线性特征空间中的子空间聚类问题；其次，该算法利用投影矩阵从特征空间中学习一种能够自适应更新的字典，以进一步提升聚类性能；最后，AKDLRR 在三次迭代之内能够收敛，并通过理论分析给出了相应的收敛性证明。在一些数据集上的实验表明，AKDLRR 不仅能够取得更好的聚类性能，并且计算速度较快。

(2)提出了一种具有闭式解的高效张量低秩表示算法（Efficient tensor low-rank representation with a closed form solution, ETLRR/CFS）。现有基于张量的子空间算法由于直接处理张量数据，具有较高的计算复杂度。为了解决这一问题，本文提出了 ETLRR/CFS 算法。具体来说，ETLRR/CFS 将张量核范数（TNN）和 Frobenius 范数整合到一

个统一的框架中，并推导出了一种高效的闭式解。在多个数据集上的实验结果表明，ETLRR/CFS 不仅比现有的基于张量的子空间聚类算法快得多，而且能够获得相似的聚类性能。

(3)提出了一种基于灵活图融合的多视图子空间聚类算法（Multi-view clustering using a flexible and optimal multi-graph fusion method, Engineering, MVC/FOMF）。现有的基于图的多视图子空间聚类算法存在图融合不够灵活以及聚类结构不被考虑的问题。为了解决这些问题，本文提出了一种基于灵活且最优的多图融合方法的多视图聚类方法（MVC/FOMF）。具体而言，MVC/FOMF 首先通过自表示方法获得每个视图的相似性图。其次，MVC/FOMF 将这些图融合为一个列和为 s（$0<s\leq1$）的共识图，并且可以通过调整 s 来寻找最佳的聚类性能。同时，MVC/FOMF 对共识图的拉普拉斯矩阵施加秩约束，以学习最佳的聚类结构。最后，MVC/FOMF 将这些步骤统一到一个框架中，并设计了基于交替乘子法的优化过程。实验表明，与最先进的算法相比，MVC/FOMF 在一些数据集上表现出非常优异的性能。

关键词：无监督学习，子空间聚类，张量核范数，核技巧，共识图

RESEARCH ON SUBSPACE CLUSTERING ALGORITHM BASED ON SELF-REPRESENTATION LEARING

ABSTRACT

With the advent of the big data era, the dimensionality and complexity of data have increased significantly, posing substantial challenges to traditional clustering methods in handling high-dimensional data. As an effective unsupervised learning technique, subspace clustering has attracted considerable attention in recent years due to its ability to extract low-dimensional structural information from high-dimensional data. However, existing subspace clustering algorithms still face limitations when dealing with complex data structures, such as difficulties in processing nonlinearly related data and relatively low computational efficiency. To address these challenges, this dissertation proposes three innovative subspace clustering algorithms. The main research contributions and innovations are as follows:

(1) To address the challenge of processing nonlinearly related data in traditional subspace clustering algorithms, we propose an Adaptive Kernel Dictionary-based Low-Rank Representation method for subspace clustering (AKDLRR). The proposed method consists of three key components. First, AKDLRR employs the kernel trick to map the original data into a new feature space, thereby transforming the clustering problem of nonlinear data into a subspace clustering problem in a linear feature space. Second, the algorithm utilizes a projection matrix to learn an adaptively updated dictionary from the feature space, which further enhances the clustering performance. Finally, AKDLRR achieves convergence within three iterations, and we provide theoretical analysis to prove its convergence properties. Experimental results on various datasets demonstrate that AKDLRR not only achieves superior clustering

performance but also maintains relatively fast computational speed.

(2) We propose an Efficient Tensor Low-Rank Representation algorithm with a Closed-Form Solution (ETLRR/CFS). Existing tensor-based subspace clustering algorithms often suffer from high computational complexity due to the direct processing of tensor data. To address this issue, we develop the ETLRR/CFS algorithm, which presents two main technical contributions. First, it integrates the Tensor Nuclear Norm (TNN) and Frobenius Norm into a unified framework. Second, and more importantly, it derives an efficient closed-form solution through rigorous mathematical derivation. Extensive experimental results on multiple datasets demonstrate that ETLRR/CFS not only achieves significant computational efficiency improvements compared to existing tensor-based subspace clustering algorithms but also maintains comparable clustering performance.

(3) We propose a novel Multi-view Clustering algorithm using a Flexible and Optimal Multi-graph Fusion method (MVC/FOMF). Existing graph-based multi-view subspace clustering algorithms face two main limitations: insufficient flexibility in graph fusion and inadequate consideration of clustering structures. To overcome these challenges, our MVC/FOMF algorithm introduces three key innovations. First, it employs a self-representation method to construct similarity graphs for each individual view. Second, it develops a flexible fusion mechanism that integrates these graphs into a consensus graph with column sums constrained to s (where $0 < s \leq 1$), where s can be adjusted to optimize clustering performance. Third, it incorporates a rank constraint on the Laplacian matrix of the consensus graph to facilitate optimal clustering structure learning. The proposed method unifies these components into a coherent framework and implements an efficient optimization procedure based on the Alternating Direction Method of Multipliers (ADMM). Comprehensive experimental results demonstrate that MVC/FOMF

significantly outperforms state-of-the-art algorithms on multiple datasets, showcasing its superior clustering capabilities.

KEY WORDS: Unsupervised learning, subspace clustering, tensor nuclear norm, kernel trick, consensus graph

目录

第1章 绪论

1.1 研究背景与意义

聚类分析是一种无监督学习方法，旨在将数据集中的样本划分为若干簇，使得簇内的样本相似度较高，而不同簇之间的样本相似度较低。其核心思想是通过数据的内在结构，自动发现数据中的潜在模式，而无需预先标注类别信息。聚类分析在数据挖掘、模式识别、机器学习等领域具有重要地位，是探索性数据分析的关键工具之一。为了实现这一目标，研究者提出了多种经典聚类算法。例如，K-means算法通过最小化样本与簇中心之间的欧氏距离来划分数据；层次聚类通过构建树状结构来反映数据的层次关系；密度聚类通过数据点密度分布发现任意形状的簇[1]等。这些算法在不同场景中展现了良好的性能，成为聚类分析的基础工具。

在实际应用中，聚类分析被广泛的应用于多个领域。例如，在生物信息学中[2]，聚类技术被用于基因表达数据分析，通过将具有相似表达模式的基因分组，帮助研究人员识别潜在的基因功能模块或疾病相关基因。在社交网络分析中，聚类算法可用于社区发现，通过将用户划分为不同的群体，揭示社交网络中的社区结构和用户行为模式。在市场营销中，聚类分析被用于客户细分[3]，通过将消费者划分为具有相似购买行为的群体，帮助企业制定精准的营销策略。

尽管传统聚类算法在许多应用中取得了显著成果，但随着各种电子产品和互联网的不断发展，各种类型的高维数据，如图像、网络文本、视频等，不断出现。对于机器学习和图像处理算法来说，高维数据由于噪声效应和冗余特征而增加计算复杂度并影响算法性能，这通常被称为维度灾难[4]。此外，许多传统方法（如K-means）假设数据分布是线性的，而实际数据往往具有复杂的非线性结构，这使得这些方法难以捕捉数据的真实分布。另一方面，传统聚类算法对噪声和异常值较为敏感，例如K-means容易受到离群点的影响，导致簇中心偏移，从而影响聚类结果的准确性。这些问题限制了传统聚类算法在复杂数据场景中的应用，也催生了更先进的聚类方法的研究。

为了克服传统聚类算法在处理高维数据时的局限性，研究者提出了子空间聚类方法（Subspace Clustering，SC）。传统聚类方法在高维空间中往往失效，

因为高维数据的稀疏性和噪声干扰使得样本之间的距离度量失去意义。此外，高维数据通常具有冗余特征，许多维度对聚类任务的贡献微乎其微。子空间聚类的提出正是为了解决这些问题，子空间聚类的核心思想是假设高维数据分布在多个低维子空间中。与使用原始数据聚类相比，子空间聚类具有以下优点：低维数据的处理效率更高，从而加快了算法的速度；将数据从原始空间转换到子空间后，可以滤除原始数据的噪声，提高聚类的精度。

自表示学习是一种基于数据自身表示数据的方法，其核心思想是每个样本可以表示为其余样本的线性组合。这种方法通过利用数据内部的线性关系，能够有效捕捉数据的全局结构，并揭示数据点之间的潜在联系。自表示学习与子空间聚类的结合，为高维数据的聚类问题提供了一种强有力的解决方案。子空间聚类假设数据分布在多个低维子空间中，而自表示学习通过学习数据的自表示矩阵，能够自然地揭示这些子空间结构。具体而言，自表示矩阵中的非零元素反映了数据点之间的相似性，从而为子空间聚类提供了重要的结构信息。

基于自表示学习的子空间聚类方法近年来得到了广泛研究。例如，稀疏子空间聚类（Sparse SC, SSC）[5]将每个数据点表示为其他点的稀疏线性组合。低秩表示（Low-Rank Representation ，LRR）[6]则通过低秩约束学习数据的全局结构，能够更好地处理噪声和异常值。此外，研究者们还提出了多种改进方法。例如，将图学习与子空间聚类算法进行结合，通过使用图正则化，以获取聚类结构更佳的相似图；而基于深度学习的方法则通过神经网络学习数据的非线性表示，进一步扩展了自表示学习的应用范围[7, 8]。

尽管上述基于自表示学习的子空间聚类方法在处理高维数据时展现了显著优势，但这些方法通常假设数据来自单一视图，即数据仅通过一种特征表示进行描述，也被称为单视图聚类。单视图聚类方法可能无法全面捕捉数据的复杂结构。例如，在图像聚类任务中，仅使用颜色特征可能无法区分纹理相似但颜色不同的图像，而仅使用纹理特征则可能忽略颜色信息的重要性。然而，在实际应用中，许多数据对象往往具有多种特征表示形式，这些来自多种渠道但描述同一物体的数据被称为多视图数据。例如，来自不同国家的媒体使用不同语言报道同一条新闻；对同一个人从不同角度进行拍摄等等。尽管多视图数据的表现形式不同，但本质上都是对同一事物的描述。相较于单视图数据，多视图数据往往包含了更丰富的信息，如果能够充分利用这些信息，往往能够取得更

佳的聚类性能。因此，如何有效利用多视图数据，成为进一步提升聚类性能的关键问题。这也为多视图聚类（Multi-view Clustering，MVC）的研究提供了重要的动机。

多视图聚类旨在充分利用多视图数据的一致性与多样性。具体而言，多视图聚类不仅能够提取每个视图特有的多样性信息，而且能够挖掘多视图数据的一致性信息，并且将这些多样性信息与一致性信息充分融合从而取得更佳的聚类性能。研究员们提出了许多基于自表示的多视图子空间聚类算法，这些算法通过将多视图与子空间学习相结合能够取得不错的性能。具体而言，基于自表示的多视图子空间聚类算法旨在从多个视图的数据中学习一个一致的自表示矩阵。通过融合来自不同视图的信息，基于多视图的子空间聚类算法往往能够取得更好的聚类性能。

随着电子设备的快速发展，许多不同类型的高维数据的不断涌现，子空间聚类算法能够有效的处理高维数据中。同时，自表示学习能够更好的揭示子空间的结构，因此如何将二者有效的结合起来具有重要的研究意义。

1.2 研究现状

1.2.1 基于单视图的子空间聚类方法

近些年，子空间聚类方法得到了广泛研究。其中，LRR和SSC是两种非常经典的单视图子空间算法。SSC和LRR假设输入数据中的任意数据点都可以由其他数据点的线性组合构成。SSC通过将L_1范数约束应用于其自表示系数矩阵来搜索原始数据中的稀疏结构，LRR源自鲁棒主成分分析（Robust principal component analysis，RPCA）[9]。RPCA假设数据由干净数据和噪声组成。与RPCA相比，LRR进一步将获得的干净数据划分为字典矩阵和自表示系数矩阵的乘积。事实上，SSC可能无法有效地捕获输入数据的全局结构，尤其是当数据严重损坏时。与SSC相比，LRR可以更有效地捕获数据的全局结构，因为LRR旨在发现隐藏在数据中的低秩结构。此后，研究者们还提出了多种基于自表示的子空间聚类算法的改进方法[10, 11]。例如，Liu等人提出了一种潜在LRR算法（Latent LRR for Subspace Segmentation and Feature Extraction，LatLRR）[12]，LatLRR创新性地将低秩表示扩展到多子空间情况，通过引入潜在表示矩阵，有效地解决了传统LRR在处理复杂数据时的局限性；Wen等人提出了一种自适应

图正则化的LRR方法（LRR with adaptive graph regularization，LRR_AGR）[13]，LRR_AGR将距离正则化和非负约束共同整合到LRR框架中，同时利用数据的全局和局部信息进行图学习。并且，引入一种新的秩约束用来学习具有清晰聚类结构的最优相似图。

尽管这些算法能够取得不错的聚类性能，但是，这些方法都是基于矩阵的。随着数据维数和样本数量的增加，这些方法需要先将数据压缩成矩阵。这一操作不仅破坏了数据的内在结构[14, 15]，而且压缩过程中会丢失一些结构信息[16, 17]。为了解决这些问题，人们开始研究基于张量的子空间聚类算法。Kilmer等人提出了一些张量运算相关定义如：张量积（tensor product，t-product）、张量奇异值分解（tensor singular value decomposition，t-SVD）和张量核范数（tensor nuclear norm，TNN）[18-20]。张量作为机器学习中不可或缺的数学工具，不仅能够捕获不同数据间的高阶相关信息，而且张量核范数更是可以有效的保留数据的内在结构。Pan等人提出了基于张量的LRR（Tensor LRR，TLRR）[21]，TLRR可以有效地还原张量数据的低秩结构，并对样本进行推断集群。同样，TLRR也使用了增广拉格朗日乘子法（Augmented Lagrange Multiplier，ALM）[14]来优化其算法。然而，TLRR需要多次迭代得到最优解，每次迭代需要计算t-product和t-SVD。不难看出，TLRR的计算复杂度高，计算量大，计算速度较慢。随后，涌现出了一些TLRR的改进算法，如增强TLRR（Enhanced TLRR，ETLRR）[22]和张量低秩稀疏表示（Tensor low-rank sparse representation for tensor subspace learning，TLRSR）[23]等。考虑到TLRR只考虑隐藏在数据中的拉普拉斯噪声，Du等人提出了ETLRR，ETLRR可以去除数据中的拉普拉斯噪声和高斯噪音，因此ETLRR可以更好的地捕获原始张量数据中的低秩结构，实验表明考虑两种噪音对数据的影响能够获得更好的聚类性能。TLRSR认为TLRR只能通过TNN捕获数据的全局结构，而对于数据的局部结构没有考虑。因此，为了同时捕捉数据中的全局结构和局部结构，TLRSR对系数张量同时施加稀疏约束和低秩约束。Cai等人提出基于一致张量低秩表示的张量子空间聚类算法（Tensor subspace clustering using consensus tensor low-rank representation，CTLRR）[24]，CTLRR通过对一致张量的拉普拉斯矩阵施加秩约束从而使一致张量具有精确的聚类结构。

综上所述，尽管这些基于单视图的子空间聚类算法能够取得不错的效果，但是仍然存在以下问题：1.传统基于自表示的单视图子空间聚类算法使用的字

典要么是包含噪音的原始数据，要么是通过额外预处理步骤得到的干净数据。2.传统基于自表示的单视图子空间聚类算法往往假设数据是线性相关的，但是在实际应用中，真实数据往往是非线性相关的。3.基于张量的子空间聚类算法往往需要花费很高的时间代价，因此难以处理大规模数据集。

1.2.2 基于多视图的子空间聚类方法

随着数据获取方式的多样化，许多实际应用中的数据往往具有多种特征表示形式，即多视图数据。例如，在图像聚类任务中，图像可以通过颜色、纹理、形状等多种特征进行描述。单视图聚类方法仅依赖于单一特征表示，可能无法全面捕捉数据的复杂结构。因此，如何有效利用多视图数据的信息，成为进一步提升聚类性能的关键问题。MVC旨在通过融合多个视图的信息，挖掘数据的一致性和互补性，从而更全面地捕捉数据的结构。与单视图聚类相比，多视图聚类能够利用不同视图之间的关联性，提升聚类结果的鲁棒性和准确性。

近年来，涌现出了许多多视图聚类算法。目前，多视图聚类算法大致可以分为[25]：基于深度的MVC算法；基于多核的MVC算法；基于图模型的MVC算法和基于子空间的MVC算法等。基于深度的MVC算法[26]通过深度学习技术（如自编码器、生成对抗网络等）自动学习多视图数据的非线性表示，能够有效捕捉数据的高维特征和复杂结构。基于多核的MVC算法通过融合多个核函数来捕捉不同视图之间的非线性关系，能够有效利用多视图数据的互补性和一致性。基于图模型的MVC算法[27]通过构建多个视图的相似性图，并融合这些图来学习一致的聚类结构。基于子空间的MVC方法由于其优秀的聚类性能，成为了多视图聚类领域中的研究热点[28-31]。Gao等人[32]提出了多视图子空间聚类方法（Multi-view Subspace Clustering，MSC），MSC使用所有视图来获得共识聚类指标，并保证不同视图之间的一致性。Maria等人[33]提出了一种MSC方法，称为多视图低秩稀疏子空间聚类（Multi-view low-rank sparse subspace clustering，MLRSSC），MLRSSC通过联合学习多个视图的子空间表示，构建一个共享的亲和矩阵。同时，MLRSSC引入低秩和稀疏约束来构建亲和矩阵，确保其既能捕捉数据的全局结构，又能保持局部结构的稀疏性。Zhang等人[34]提出了潜在多视图子空间聚类（Latent Multi-view Subspace Clustering, LMSC），与大多数现有算法不同，LMSC利用所有视图来寻找潜在表示并重构潜在矩阵。Zhou等人[35]提出了双重共享-特定多视图子空间聚类（Dual Shared-Specific Multiview Subspace Clustering,

DSS-MSC），该方法同时学习多个视图之间共享信息的相关性，并利用视图特定信息来描述每个视图的特定属性。Tang等人[36]设计了多样性正则化方法（Learning a Joint Affinity Graph for Multiview Subspace Clustering），以学习每个视图的最优权重，从而抑制冗余并增强不同视图之间的多样性。Zheng等人[37]提出了特征拼接多视图子空间聚类（Feature Concatenation Multi-view Subspace Clustering, FCMSC），该方法使用L_{21}范数两次重构数据，并对相似矩阵施加低秩约束。Zhao等人[38]提出了一种用于多视图聚类的亲和矩阵学习方法（Clean and Robust Affinity Matrix Learning for Multi-view Clustering, CRAA），该方法通过构建所有视图的表示来学习共享结构，并通过鲁棒主成分分析获得更优质的相似矩阵。Yao等人[39]使用了双重表示结构的尺度单纯形表示（Double Structure Scaled Simplex Representation, DSSSR），可以减少噪声数据的负面影响并拼接多视图数据。

为了充分挖掘来自多个视图的数据之间的高阶相关信息，研究员们考虑将基于矩阵的多视图聚类算法向张量进行拓展。近年来，基于张量的多视图聚类算法得到了广泛的关注[40-42]。例如，Du等人[43]提出了一种基于相似性矩阵和低秩张量分解的多视图子空间聚类算法（Multi-view subspace clustering based on affinity matrix and low-rank tensor decomposition，ATMSC）。ATMSC不仅构建了更具意义的相似性矩阵，还充分利用了多视图互补信息并探索了高阶相关性。Liu等人[44]提出了一种新的基于低秩张量的不完全多视图子空间聚类方法（Incomplete Multi-view subspace clustering with Low-Rank Tensor，IMSCLT）。IMSCLT采用带有低秩张量约束的子空间表示，以同时挖掘数据点之间的视图特定关系和跨视图关系，并捕捉多视图的高阶相关性。此外，IMSCLT通过近似视图特定和共同子空间表示的内积，为多视图学习任务学习一个判别性相似图。Wang等人[45]提出了一种低秩和稀疏张量表示（Low-rank and sparse tensor representation，LRSTR）方法。LRSTR通过自表示张量学习相似性矩阵，并保留视图维度的相似性信息，用于多视图子空间聚类。具体来说，LRSTR通过对自表示张量施加张量核范数和张量稀疏约束来表征视图之间的关系。

综上所述，基于多视图的子空间聚类方法通过融合多个视图的信息，能够更全面地捕捉数据的结构，提升聚类性能。

1.3 主要研究内容与创新点

本课题对近年来单视图子空间聚类以及多视图子空间聚类的国内外研究现状进行了详细的梳理，了解了目前子空间聚类的发展历程。针对基于自表示的子空间聚类方法进行了深入地研究，通过分析现有算法的不足，提出了有效的改进算法。本文的研究内容框架图如图 1-1 所示。

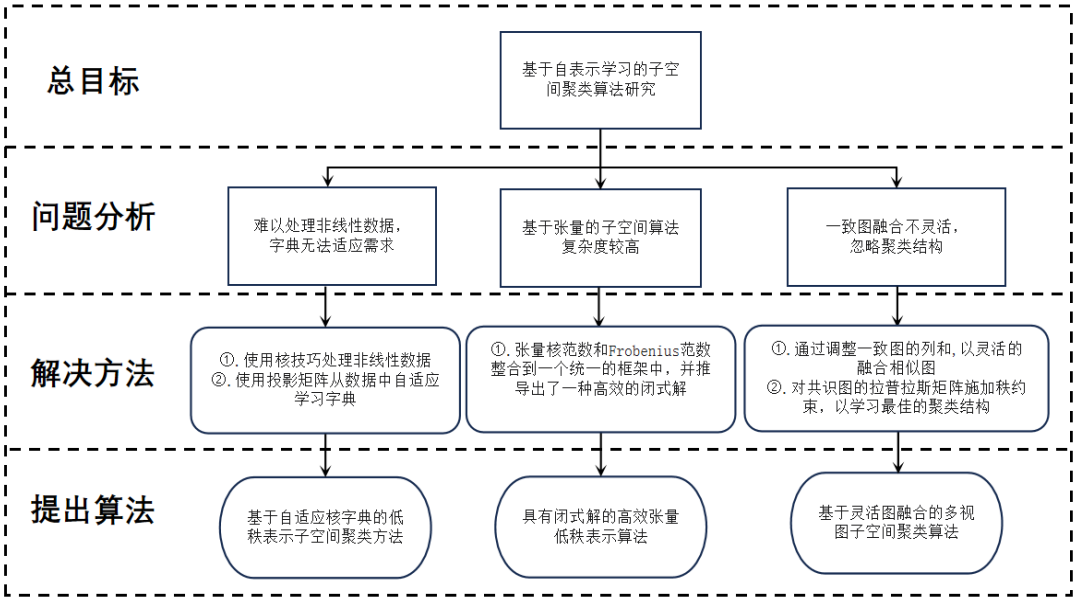


图 1-1 论文研究思路框架图

Figure 1-1 The frame diagram of the research idea of this paper

主要研究工作和创新点总结如下：

（1）传统的子空间聚类算法默认数据点之间是线性相关的，因此难以处理实际应用中非线性相关的数据。传统基于自表示的子空间聚类算法使用的字典多是固定的，固定的字典难以适应迭代需求。为此，本文提出了基于自适应核字典的低秩表示子空间聚类方法（An adaptive kernel dictionary based low-rank representation method for subspace clustering，AKDLRR）。首先，AKDLRR通过引入核技巧，将原始数据映射到一个新的特征空间，从而将非线性数据的聚类问题转化为线性特征空间中的子空间聚类问题；其次，该算法利用投影矩阵从特征空间中学习一种能够自适应更新的字典，以进一步提升聚类性能；最后，AKDLRR在三次迭代之内能够收敛，并通过理论分析给出了相应的收敛性证明。在一些数据集上的实验表明，AKDLRR不仅能够取得更好的聚类性能，并且计算速度较快。

（2）传统的基于张量的子空间聚类算法的计算开销普遍较高，难以处理大规模数据。为了解决这一问题，本文提出了一种具有闭式解的高效张量低秩表示算法（Efficient tensor low-rank representation with a closed form solution, ETLRR/CFS）。现有基于张量的子空间算法由于直接处理张量数据，具有较高

的计算复杂度。为了解决这一问题，本文提出了ETLRR/CFS算法。具体来说，ETLRR/CFS将TNN和Frobenius范数整合到一个统一的框架中，并推导出了一种高效的闭式解。在多个数据集上的实验结果表明，ETLRR/CFS不仅比现有的基于张量的子空间聚类算法快得多，而且能够获得相似的聚类性能。

（3）传统多视图子空间聚类算法没有考虑聚类结构，导致聚类性能不佳。传统多视图子空间聚类算法在融合来自多个视图的相似图时，将共识图的列和严格设置为1，导致最终获得的共识图聚类效果较差。为此，本文提出了一种基于灵活图融合的多视图子空间聚类算法（Multi-view clustering using a flexible and optimal multi-graph fusion method, Engineering, MVC/FOMF）。具体而言，MVC/FOMF首先通过自表达方法获得每个视图的相似性图。其次，MVC/FOMF将这些图融合为一个列和为 S （$0 < S \leq 1$）的共识图，并且可以通过调整 S 来寻找最佳的聚类性能。同时，MVC/FOMF对共识图的拉普拉斯矩阵施加秩约束，以学习最佳的聚类结构。最后，MVC/FOMF将这些步骤统一到一个框架中，并设计了基于交替乘子法的优化过程。实验表明，MVC/FOMF能够取得最佳的聚类性能，同时计算时间也很快。

算法之间的关系总结如下：

AKDLRR和ETLRR/CFS是基于单视图的子空间聚类算法，MVC/FOMF是基于多视图的子空间聚类算法。因此从AKDLRR和ETLRR/CFS到MVC/FOMF是单视图子空间聚类向多视图子空间聚类的拓展性工作。而AKDLRR和ETLRR/CFS旨在解决传统单视图子空间聚类算法中存在的不同问题，二者之间是并列关系。

1.4 论文的结构安排

本文分为六个章节，其中第3至第5章为论文主体。

第1章是绪论，主要介绍子空间聚类的背景和发展现状以及本文工作的研究重点和组织架构。

第2章为相关知识介绍，主要介绍了子空间聚类的相关理论，常用技术和聚类评价指标。

第3章提出一种基于自适应核字典的低秩表示子空间聚类方法（AKDLRR），并详细介绍了AKDLRR的提出过程、求解及优化过程，收敛性分析，并设计了充分的实验证明所提方法的有效性。

第4章提出了一种具有闭式解的高效张量低秩表示算法（ETLRR/CFS），并详细介绍了ETLRR/CFS 的提出过程、求解、复杂度分析、参数分析，并在公开数据集上验证了ETLRR/CFS 优越的聚类性能。

第5章中提出了一种基于灵活图融合的多视图子空间聚类算法（MVC/FOMF），并详细介绍了MVC/FOMF的提出过程、求解及优化过程、复杂度分析、运行时间比较，大量的实验结果证明了所提出方法的高效性。

第6章是本文的总结与展望。

第2章 相关知识介绍

2.1子空间聚类算法

SSC和LRR是两种经典的子空间聚类算法。SSC假设所有数据点都可以由其他数据点的线性组合构成。具体来说，SSC通过使用稀疏约束在输入的原始数据中搜索稀疏结构：

$$\min_{Z,E} \|Z\|_1 + \|E\|_1 \ \ s.t.\ X = XZ + E \ and \ diag(Z) = 0 \tag{2.1}$$

其中 X 表示原始数据，Z 表示自表示系数矩阵，E 表示原始数据中的噪音。$\|\bullet\|_1$ 表示L_1范数约束。

RPCA假设数据由干净数据和噪声组成。为了从原始数据中获得干净的低秩数据，在RPCA中提出了秩最小化问题：

$$\min_{L,E} \ rank(L) + \lambda \|E\|_{\ell} \ \ ss.t.\ \ X = L + E \tag{2.2}$$

其中 ℓ 表示不同的范数如L_1范数，L_{21}范数等，L 表示干净数据。与RPCA不同，LRR进一步将干净数据细分为低秩结构矩阵和字典：

$$\min_{Z,E} \ rank(Z) + \lambda \|E\|_{\ell} \ \ ss.t.\ \ X = AZ + E \tag{2.3}$$

其中 A 表示字典。此外，由于求解 $rank(Z)$ 是一个NP问题，因此LRR使用核范数运算来约束 Z ：

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1} \ \ s.t.\ X = AZ + E \tag{2.4}$$

其中 $\|\bullet\|_*$ 表示核范数约束。由于上述算法都是基于矩阵设计的，因此在处理张量数据时，需要对数据进行压缩。这无疑会丢失原始数据中存在的结构信息。为了能够直接处理张量数据，pan等人提出了LRR的张量版本[21]，即TLRR：

$$\min_{\mathcal{Z},\mathcal{E}} \|\mathcal{Z}\|_* + \lambda \|\mathcal{E}\|_1 \ \ s.t.\ \mathcal{X} = \mathcal{A} * \mathcal{Z} + \mathcal{E} \tag{2.5}$$

其中 $\mathcal{X}$ 表示原始数据，$\mathcal{Z}$ 表示自表示系数张量，$\mathcal{A}$ 表示字典，$\mathcal{E}$ 表示噪音张量。

2.2 张量学习

2.2.1 基本概念

张量学习（Tensor Learning）是机器学习和数据分析领域的一个重要分支，主要研究如何利用张量（Tensor）这一数学工具来处理高维数据。张量可以看作是向量和矩阵的高维扩展，能够更自然地表示多维数据结构。

2.2.2 张量操作

在本文中常用到的为张量操作，因此下面会介绍关于张量操作的知识。

定义2-1（块循环矩阵）：给定张量$\mathcal{A} \in R^{n_1 * n_2 * n_3}$，它的块循环矩阵定义为：

$$bcirc(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \cdots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \cdots & \mathcal{A}^{(1)} \end{bmatrix} \in R^{n_1 n_3 \times n_2 n_3} \tag{2.6}$$

定义2-2（展开和折叠运算）：设张量$\mathcal{A} \in R^{n_1 * n_2 * n_3}$，展开和折叠运算定义为：

$$unfold(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(n_3)} \end{bmatrix} \tag{2.7}$$

$$fold(unfold(\mathcal{A})) = \mathcal{A} \tag{2.8}$$

定义2-3（张量积）：给定张量$\mathcal{B} \in \mathbf{R}^{n_1 \times l \times n_3}$，和另一个张量$\mathcal{C} \in \mathbf{R}^{l \times n_2 \times n_3}$，两个张量的t-product可定义为：

$$\mathcal{B} * \mathcal{C} = \text{fold}(\text{bcirc}(\mathcal{B}) \cdot \text{unfold}(\mathcal{C})) \tag{2.9}$$

定义2-4（正交张量）：如果张量$\mathcal{A} \in R^{n_1 * n_2 * n_3}$是正交张量，那么：

$$\mathcal{A} * \mathcal{A}^T = \mathcal{A}^T * \mathcal{A} = \mathcal{I} \tag{2.10}$$

其中$\mathcal{I}$是单位张量，第一个前向切片是一个$n_1 \times n_1$的单位矩阵，其他前向切片都是0。

定义2-5（张量奇异值分解）：给定一个张量$\mathcal{A} \in R^{n_1 * n_2 * n_3}$，它的张量奇异值分解（t-SVD）可以描述为:

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^* \tag{2.11}$$

其中$\mathcal{U} \in R^{n_1 \times n_1 \times n_3}$和$\mathcal{V} \in R^{n_2 \times n_2 \times n_3}$都是正交张量，$\mathcal{S} \in R^{n_1 \times n_2 \times n_3}$是一个f-对角张量。

定义2-6（张量核范数）：设定张量$\mathcal{C} \in R^{n_1 * n_2 * n_3}$，其张量核范数可以定义为

$$\|\mathcal{C}\|_{TNN} = \sum_{i=1}^{\min(n_1, n_2)} \sum_{k=1}^{n_3} \left| S_f(i,i,k) \right| \tag{2.12}$$

其中S_f可以通过快速傅里叶变换（FFT）获得。

2.3 增广拉格朗日乘子法

增广拉格朗日乘子法（Augmented Lagrangian Method，ALM）是一种用于求解约束优化问题的数值优化方法。它通过将原始的约束优化问题转化为无约束优化问题，并引入拉格朗日乘子和罚函数来逐步逼近原问题的最优解。

考虑以下约束优化问题：

$$\begin{cases} \min\ f(x) \\ s.t.\ h(x) = 0 \end{cases} \tag{2.13}$$

其中$h(x)$是约束条件，$f(x)$是目标函数。其增广拉格朗日函数定义为：

$$L(x, \lambda) = f(x) + \lambda^T (h(x)) + \frac{\mu}{2} \|h(x)\|_F^2 \tag{2.14}$$

其中λ是拉格朗日乘子，μ是一个惩罚参数。

具体求解过程中，增广拉格朗日乘子法采用交替迭代的方式进行。首先，固定拉格朗日乘子λ的值，通过求解无约束优化问题$\min L(x, \lambda)$来更新变量x。然后，在更新x后，通过更新拉格朗日乘子λ，使约束条件更好地满足。这个过程 迭代执行，直到达到收敛条件为止。

增广拉格朗日乘子法是一种强大的约束优化方法，通过结合拉格朗日乘子法和罚函数法，能够高效地求解复杂的约束优化问题。

2.4 聚类评价指标

在实际应用中，常常需要综合考虑多个评价指标来进行聚类结果的综合评估。在本章中选取了三个常见的聚类评价指标进行详细介绍。

准确率（Accuracy，ACC）是一种用于评估聚类算法性能的指标。ACC是衡量聚类结果与真实标签一致性的指标，表示聚类结果中正确分类的样本比例。

$$\mathrm{ACC}=\frac{\sum_{i=1}^{n}\delta(s_i,map(r_i))}{n} \tag{2.15}$$

其中 r_i， s_i 分别表示数据 x_i 所对应的获得的标签和真实标签， $map(r_i)$ 将聚类标签映射到真实标签的最佳匹配（通常使用匈牙利算法）， n 为数据总的个数，δ 表示函数如下

$$\delta(x,y)=\begin{cases}1, & \text{if } x=y\\ 0, & \text{otherwise}\end{cases} \tag{2.16}$$

ACC的取值范围为0到1，值越接近1表示聚类结果与真实标签越相似，值越接近0表示聚类结果与真实标签越不相似。

标准化互信息（Normalized Mutual Information ，NMI）：标准化互信息衡量聚类结果与真实标签之间的信息共享程度，是一种基于信息论的指标，范围在[0, 1]之间。NMI越接近1表示聚类结果与真实标签的一致性越好。

$$NMI(\Omega,C)=\frac{I(\Omega,C)}{(H(\Omega)+H(C))/2} \tag{2.17}$$

其中, I 表示互信息(Mutual Information)， H 表示熵。互信息 $I(\Omega,C)$ 表示给定类簇信息 C 的前提条件下,类别信息 Ω 的增加量。

纯度（Purity，PUR）：纯度衡量每个聚类簇中最多数样本所属的真实类别的比例，反映聚类结果的“纯净度”。

$$Purity(\Omega,C)=\frac{1}{N}\sum_{k=1}^{K}\max_{j}\left|w_k\cap c_j\right| \tag{2.18}$$

其中，N 表示样本总数，K 表示聚类簇的数量，Ω 表示簇划分，c_j 表示第 j 个真实类别，w_k 表示第 k 个聚类簇，$\left|w_k \cap c_j\right|$ 表示第 k 个聚类簇中属于第 j 个真实类别的样本数。

2.5 本章小结

本章主要对子空间聚类算法的研究基础进行了概述。首先介绍了几种经典的子空间聚类算法；然后介绍了张量相关的知识；随后介绍了本文求解目标函数使用的增广拉格朗日乘子法，最后为后续章节实验部分用到的聚类评价标准ACC、NMI、 PUR 提供了理论定义。

第3章 基于自适应核字典的低秩表示子空间聚类算法

3.1 引言

近年来，子空间聚类作为一种重要的聚类技术，在机器学习、图像处理、计算机视觉等领域得到了广泛应用。子空间聚类的设计目的是将数据点划分到其各自的子空间中，并且能够有效去除原始数据中的噪声。研究人员提出了许多子空间聚类方法，其中SSC和LRR是两种经典的子空间聚类方法。

SSC和LRR假设输入数据中的所有点都可以表示为数据中其他点的线性组合。SSC通过对系数矩阵施加L1范数约束来寻找原始数据中的稀疏结构，而LRR则是从RPCA中衍生出来的。RPCA假设数据由干净数据和噪声组成。与RPCA相比，LRR进一步将获得的干净数据分解为字典矩阵和自表达矩阵的乘积。事实上，SSC可能无法有效捕捉输入数据的全局结构，尤其是在数据严重损坏的情况下。与SSC相比，LRR能够更有效地捕捉数据的全局结构，因为它旨在发现数据中隐藏的低秩结构。通常，LRR假设原始数据由干净数据和噪声组成，其中干净数据由去噪后的数据（作为字典）乘以低秩系数矩阵构成。

近年来，许多基于LRR的算法被提出，例如：基于自适应图正则化的低秩表示方法（LRR_AGR）[13]。具体来说，LRR_AGR通过将LRR、非负约束和距离正则化整合到一个统一的框架中，LRR_AGR可以利用全局和局部数据信息进行图学习。目前，大多数基于LRR的算法使用原始数据或去噪后的数据作为字典，这使得大多数基于LRR的算法的字典是固定的，即字典无法有效满足迭代的需求，从而导致聚类性能较差。

此外，大多数子空间聚类方法假设数据是线性独立的。因此，这些方法只能处理线性子空间。在实际应用中，数据点通常不是线性相关的，而是从非线性子空间中提取的。通常，有两种方法可以解决涉及原始数据的非线性相关问题。第一种方法是流形学习方法[46]，它假设高维数据嵌入在低维空间中。第二种方法是所谓的核技术。通过假设非线性数据点通常位于高维环境中，原始数据可以通过核映射被映射到高维特征空间，使得映射后的特征位于多个线性子空间的并集中。本文提出的算法采用了第二种方法。事实上，核技术已经在聚类算法中得到了广

泛的应用[25, 47-49]。例如，核稀疏子空间聚类（Kernel sparse subspace clustering，KSSC）[50]和鲁棒核低秩表示（Robust Kernel Low-Rank Representation，RKLRR）[51]分别通过核技术将SSC和LRR扩展到非线性场景。核映射用于将原始数据映射到新的特征空间，这意味着非线性数据的聚类问题被转化为线性特征空间中的子空间聚类问题。实验表明，这些基于核的算法能够比原始算法实现更好的聚类性能。

本文提出了一种基于自适应核字典的低秩表示方法，称为AKDLRR。图3-1展示了AKDLRR的流程图。AKDLRR通过使用核技术解决了大多数子空间聚类方法无法有效处理非线性关系的问题。同时，通过使用自适应字典，AKDLRR缓解了之前开发的子空间聚类方法中因固定字典而导致的聚类性能较差的问题。具体来说，AKDLRR致力于在核空间中学习一个投影矩阵，然后用于生成一个在每次迭代中更新的自适应字典。此外，由于迭代过程中的子问题具有闭式解，AKDLRR的计算效率相对较高，并且对AKDLRR收敛性能的理论分析表明，在某些条件下，它最多可以在三次迭代内收敛。具体来说，本章工作的贡献如下：

（1）本文提出了一种基于自适应核字典的低秩表示子空间聚类算法，称为AKDLRR。AKDLRR采用核技巧来解决传统算法无法处理非线性相关数据的问题。此外，本章提供了一个定理来证明表示定理在我们提出的模型中仍然成立；即核投影矩阵可以用映射到核空间的数据矩阵与一个适当维度的矩阵的乘积来替代。这一贡献为后续目标函数的推导奠定了理论基础。

（2）AKDLRR尝试在核空间中学习一个投影矩阵，以生成一个能够自动适应每次迭代的字典。在传统算法中，通常使用固定字典，但这些字典在求解过程中无法适应每次迭代，导致聚类性能较低。

（3）本章为AKDLRR每次迭代中的子问题提供了闭式解，从而提高了计算速度。最后，对所提出的AKDLRR方法的收敛性能进行了理论分析，结果表明在某些条件下，AKDLRR最多可以在三次迭代内收敛。

（4）实验表明，AKDLRR不仅比其他算法具有更好的聚类性能，而且速度更快。

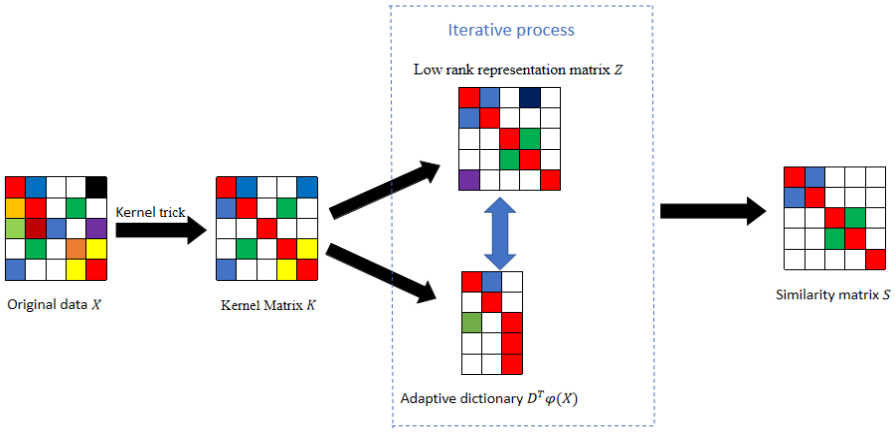


图 3-1 AKDLRR的流程图

Figure 3-1 Flowchart of AKDLRR

3.2 基于自适应核字典的低秩表示子空间聚类算法

3.2.1 AKDLRR 模型搭建

LRR及其变体算法仅适用于数据点之间存在线性相关性的情况。然而，这些方法在实际应用中无法有效处理非线性数据。为此，本章提出AKDLRR，旨在解决现有LRR方法及其变体的局限性。此外，AKDLRR通过设置自适应字典，能够缓解基于LRR算法中使用固定字典导致的聚类性能不佳的问题。通常，基于LRR的算法描述如下：

$$\min_{Z,E}\|Z\|_*+\frac{\lambda}{2}\|E\|_\ell \quad s.t.\ X=AZ+E \tag{3.1}$$

为了更好的解决公式(3.1)，AKDLRR将重构误差加入到目标函数中：

$$\min_{Z}\|Z\|_*+\frac{\lambda}{2}\|X-AZ\|_F^2 \tag{3.2}$$

为了探索原始数据中包含的非线性信息，AKDLRR在公式(3.2)中使用了核技术。随后，得到以下结果：

$$\min_{Z}\|Z\|_*+\frac{\lambda}{2}\|\varphi(X)-\varphi(A)Z\|_F^2 \tag{3.3}$$

其中 $\varphi(A)$ 表示将 A 映射到希尔伯特空间. 为了消除字典学习中任意尺度的因素，可以对字典施加正交性约束：

$$\min_{Z}\|Z\|_{*}+\frac{\lambda}{2}\|\varphi(X)-\varphi(A)Z\|_{F}^{2}\ \ s.t.\ \varphi(A)\varphi(A)^{T}=I_{d} \tag{3.4}$$

其中 d 表示数据经过降维后的维数。AKDLRR考虑使用 $\varphi(P)^{T}\varphi(X)$ 代替 $\varphi(X)$ 和 $\varphi(A)$, 其中 P 是一个低维投影矩阵， $\varphi(P)$ 表示将 P 映射到希尔伯特空间. 随后, 公式(3.4)可以被改写为:

$$\min_{Z}\|Z\|_{*}+\frac{\lambda}{2}\left\|\varphi(P)^{T}\varphi(X)-\varphi(P)^{T}\varphi(X)Z\right\|_{F}^{2}\ \ s.t.\ \varphi(P)^{T}\varphi(X)\varphi(X)^{T}\varphi(P)=I_{d} \tag{3.5}$$

在求解公式(3.5)之前，我们首先给出以下定理1。

定理1：针对问题 $\min_{\varphi(P)}\|\varphi(X)-\varphi(P)X\|_{F}^{2}$，存在一个具有如下形式的最优解 $\varphi(P)^{*}$：

$$\varphi(P)^{*}=\varphi(X)D \tag{3.6}$$

其中 $D\in R^{n\times d}$ 。

证明：利用标准的正交分解过程，可以将 $\varphi(P)^{*}$ 写成：

$$\varphi(P)^{*}=\varphi(P)+\varphi(P)_{\perp} \tag{3.7}$$

其中 $\varphi(P)$ 和 $\varphi(P)_{\perp}$ 均为矩阵， $\varphi(P)$ 的列向量属于 $\varphi(X)$ 的列空间，而 $\varphi(P)_{\perp}$ 的列向量则位于该列空间的正交补空间中。具体而言， $\varphi(P)=\varphi(X)D$ 且 $\varphi(X)^{T}\varphi(P)_{\perp}=0$ 。随后，可以得到：

$$\left\|\varphi(X)-\varphi(P)^{*}X\right\|_{F}^{2}=\left\|\varphi(X)-(\varphi(P)+\varphi(P)_{\perp})X\right\|_{F}^{2} \tag{3.8}$$

$$=\|\varphi(X)(I-DX)-\varphi(P)_{\perp}X\|_{F}^{2} \tag{3.9}$$

$$=tr([\varphi(X)(I-DX)-\varphi(P)_{\perp}X]^{T}[\varphi(X)(I-DX)-\varphi(P)_{\perp}X]) \tag{3.10}$$

$$=tr[(I-DX)^{T}\varphi(X)^{T}\varphi(X)(I-DX)]+tr[X^{T}\varphi(P)_{\perp}^{T}\varphi(P)_{\perp}X] \tag{3.11}$$

由于 $\varphi(P)_{\perp}^{T}\varphi(P)_{\perp}$ 是一个正半定矩阵，则公式(3.11)的第二项可以表示为：

$$tr[X^{T}\varphi(P)_{\perp}^{T}\varphi(P)_{\perp}X]=\sum_{i=1}^{n}x_{i}^{T}\varphi(P)_{\perp}^{T}\varphi(P)_{\perp}x_{i}\geq 0 \tag{3.12}$$

由公式(3.12)可知，第二项只会增加成本函数，因此为了得到最优解，我们需要消去第二项。具体而言，$\varphi(P)_{\perp}=0$且$\varphi(P)^{*}=\varphi(P)=\varphi(X)D$为最优解。

□

随后，使用定理1，可以将$\varphi(P)$替换成$\varphi(X)D$：

$$\begin{aligned}&\min_{Z}\|Z\|_{*}+\frac{\lambda}{2}\left\|D^{T}\varphi(X)^{T}\varphi(X)-D^{T}\varphi(X)^{T}\varphi(X)Z\right\|_{F}^{2}\\&s.t.\ D^{T}\varphi(X)^{T}\varphi(X)\varphi(X)^{T}\varphi(X)D=I_{d}\end{aligned}\tag{3.13}$$

由于映射到希尔伯特空间的维数很高，目标函数很难直接优化。为了方便求解，计算Gram矩阵$K=\varphi(X)^{T}\varphi(X)$作为代替，得到最终目标函数如下：

$$\min_{Z,D}\|Z\|_{*}+\frac{\lambda}{2}\left\|D^{T}K-D^{T}KZ\right\|_{F}^{2}\ \ s.t.\ D^{T}KK^{T}D=I_{d}\tag{3.14}$$

3.2.2 AKDLRR模型求解

为了解决公式(3.14)，AKDLRR迭代的更新它的变量。具体而言，每次更新一个变量并固定其他变量。首先，AKDLRR更新Z并固定其余变量。因此，公式(3.14)可以被转换为另一个问题：

$$\min_{Z}\|Z\|_{*}+\frac{\lambda}{2}\left\|D^{T}K-D^{T}KZ\right\|_{F}^{2}\tag{3.15}$$

公式(3.15)可以通过定理2求解。

定理2：设$\mathrm{A}=U\Lambda V^{T}$为矩阵$A$的SVD。则问题如下：

$$\min_{C}\|C\|_{*}+\frac{\tau}{2}\|A-AC\|_{F}^{2}\tag{3.16}$$

的最优解为$C=V_{1}(I-\frac{1}{\tau}\Lambda_{1}^{-2})V_{1}^{T}$。其中$U=[U_{1},U_{2}]$，$\Lambda=diag(r_{1},r_{2},...,r_{n})$和$V=[V_{1},V_{2}]$都是根据集合$\{k:r_{k}>1/\sqrt{\tau}\}$和$\{k:r_{k}\le 1/\sqrt{\tau}\}$划分的。

最后，AKDLRR更新变量D并固定其他变量。因此，公式(3.14)可以被转换为另一个问题：

$$\min_{D} \left\| D^T K - D^T K Z \right\|_F^2 \ \ s.t.\ D^T K K^T D = I_d \tag{3.17}$$

设 $R = K - KZ$ ，可以得到：

$$\min_{D} \left\| D^T R \right\|_F^2 \ \ s.t.\ D^T K K^T D = I_d \tag{3.18}$$

随后，将公式(3.18)转换成如下等价问题：

$$\min_{D} tr(R^T D D^T R) \ s.t.\ D^T K K^T D = I_d \tag{3.19}$$

其中 $tr(\bullet)$ 表示矩阵求迹。采用拉格朗日乘子法求解公式(3.19)，得到：

$$L(D) = tr(R^T D D^T R) - \mu \cdot tr(D^T K K^T D - I_d) \tag{3.20}$$

其中参数 $\mu > 0$ 。令 $\frac{\partial L(D)}{\partial D} = 0$ ，得到：

$$R R^T D = \mu K K^T D \tag{3.21}$$

该问题可转化为广义特征值问题，通过计算 d 个最小特征向量得到最优解。因此，公式(3.21)的最优解是由 $R R^T D = \mu K K^T D$ 的前 d 个特征向量组成对应于 d 个最小特征值组成的： $D = [u_1, u_2, ..., u_d]$ 。最后，在算法1中展示了AKDLRR的流程。

算法 1: AKDLRR

输入: 原始数据 $X \in R^{c \times n}$, 参数 $\lambda > 1$

初始化: 通过 $K = \varphi(X)^T \varphi(X)$ 计算 K ， $D = I_n$.

1.判断收敛

2. 通过定理2求解问题(3.15)更新 Z ;

3. 通过求解问题(3.17)更新 D ;

4.结束

5. 计算相似矩阵 $S = \frac{|Z| + |Z^T|}{2}$

输出: S

3.2.3 AKDLRR 算法收敛性分析

AKDLRR的每个子问题都具有闭式解。在一定条件下，两个子问题的计算结果将重复迭代直至达到收敛。以下定理3给出了AKDLRR的收敛条件。

定理3：当$\lambda>1$且$d=k$时，AKDLRR将会在三次迭代内收敛，其中k由定理2决定。

证明：设定$g(K)=\|Z\|_*+\frac{\lambda}{2}\|D^TK-D^TKZ\|_F^2$且$D_0=I_n$。在第一轮迭代中，最优值$g(K)$可以通过下述公式计算：

$$g(K)=\sum_{i\in V_1}(1-\frac{1}{2\lambda}r_i^{-2})+\frac{\lambda}{2}\sum_{i\in V_2}r_i^2 \tag{3.22}$$

其中$[U_t,S_t,V_t]=SVD(K)$，$diag(S_t)=[r_1,r_2,...,r_m]$，$V_1$表示$V_t$最左边的$k$列，$V_2$表示$V_t$剩余列。

在第二次迭代中，通过解决下述问题更新D_2：

$$R_1R_1^TD_1=\mu KK^TD_1 \tag{3.23}$$

其中$R_1=K-KZ_1$，D_1由前d个特征向量$u_1,u_2,..,u_d$组成。在目标函数中，有$D^TKK^TD=I_d$的约束。这表明D_1^TK是一个正交矩阵。众所周知，在奇异值分解中，正交矩阵的奇异值均为1。根据正交矩阵奇异值分解的性质，可以得到$[U_t,S_t,V_t]=SVD(D_1^TK)$，其中$S_t$是一个主对角线元素为1其余元素为零的矩阵。此外，$S_t^{'}$由$S_t$的前$d$个对角元素组成。如果$d=k$，那么$S_t^{'}$就是一个大小为$k\times k$的单位矩阵。

如果$\lambda>1$，那么$\frac{1}{\sqrt{\lambda}}<1$。此外，由于$D_1^TK$是一个正交矩阵，其奇异值$diag(S_t^{'})=[r_1,r_2,...,r_m]$恒为1。因此，所有的$r_k>\frac{1}{\sqrt{\lambda}}$。这意味着，$V_1$由$V_t$中的所有列组成，而$V_2$是一个空集（此处的$V_1$和$V_2$来自定理2）。因此，更$Z$时，公式(3.22)可以被转换为：

$$g(K)=\sum_{i\in V}(1-\frac{1}{2\lambda}r_i^{-2}) \tag{3.24}$$

在随后的迭代中，由于$d=k$，因此$S_t^{'}$一直是一个单位矩阵。这意味着$g(K)$将不会随着迭代而变化。因此$g(K)$将会在$t=3$后保持不变，t代表迭代次数。

□

3.2.4 AKDLRR 算法复杂度

根据算法1，AKDLRR的主要复杂度来源于SVD和特征值分解过程。在算法1初始化时，需要计算Gram矩阵，其复杂度为$O(cn^2)$。当更新变量Z时，需要对矩阵D进行SVD，其复杂度为$O(dn^2)$。当更新变量D时，需要进行特征值分解，其复杂度为$O(n^3)$。因此，AKDLRR的算法复杂度为$O(t(dn^2+n^3)+cn^2)$。其中，t代表迭代次数。根据定理3，AKDLRR的迭代次数不会超过3.

3.3 实验及分析

3.3.1 实验设置

实验使用的环境是MATLAB 2019b，所使用的计算机包含一个3.70 GHz 的 i9-10900H CPU 和 64 GB内存。

数据集

ExtendYaleB: ExtendYaleB 数据集包含 28 个人的图像，每人 30 张，共计 840 张图像。该数据集的挑战在于每个人物图像都是在强光条件下拍摄的，光照变化较大。

ORL: ORL 数据集包含40个人的400张图像，图像之间的主要差异在于人物的表情变化。

FRDUE: FRDUE 数据集相对较大，包含3040张图像，分为152个类别。该数据集要求算法能够同时高效地区分多个类别。由于样本数量和类别数量都较大，这对测试算法提出了较高的挑战。

COIL20: COIL20 是一个用于物体识别的数据集，包括20种物体，每种物体从不同角度拍摄了72张图像，共计1440个样本。这些物体包括各种日常物品和工具，如杯子、笔记本电脑和花瓶等。

Scene15: Scene15 数据集用于场景分类，包含 15 种不同场景，每个场景大约有300 张图像，共计 4485 个样本。这些场景包括森林、海滩、高速公路等。

CCV: CCV 数据集包含 6773 张图像，用于图像分类和检索任务。这些图像来自不同的场景，属于多个类别，如动物、自然景观和建筑物等。

表3-1数据集的统计信息

Table 3-1 Statistics of data sets

数据集	样本数	特征数	类簇数
ExtendYaleB	840	4800	28
ORL	400	1024	40
FRDUE(100)	1980	550	99
FRDUE(all)	3040	550	152
COIL20	1440	1024	20
Scene15	4485	1180	15
CCV	6773	20	20

对比算法

在实验中，选取了如下5 种单视图聚类方法与AKDLRR 综合比较。

SSC[5]：SSC通过对系数矩阵施加稀疏约束，在原始数据中寻找稀疏结构。

LRR[6]：LRR是从RPCA中衍生出来的。与RPCA相比，LRR进一步从干净数据中分离出低秩结构以进行聚类。

LRR/CFS[52]：LRR/CFS将低秩约束和Frobenius范数统一到一个框架中。实验结果表明，LRR/CFS不仅速度快，而且与许多其他基于LRR的算法性能相当。

AOPL_ROOT[53]：AOPL_ROOT将高阶邻近性引入结构化图学习，并结合了自适应邻近性学习。

ACMK_SC[54]：ACMK_SC将基础结果提取和共识学习整合到一个框架中，使得这两个任务能够相互促进，从而获得良好的聚类性能。

3.3.2 对比实验结果与分析

在实验中，使用聚类准确率（ACC）和归一化互信息（NMI）作为评价指标。评价指标的范围为[0-100]，如果值越大，表示聚类效果越好。

表 3-2 展示了AKDLRR与对比算法在多个数据集上的聚类性能。表3-3展示了AKDLRR与对比算法在多个数据集上的运行时间。最佳结果以粗体标记。可以得出以下几点结论：

（1）不难看出，AKDLRR在所有数据集上的聚类性能均优于其他算法。具体而言，AKDLRR在ExtendYaleB数据集上的性能比其他算法至少高出10%。在ORL、FRDUE（100）、FRDUE（all）、Scene15、CCV和COIL20数据集上，AKDLRR的性能比排名第二的方法高出约1%至5%。尽管AOPL_ROOT和ACMK_SC都采用了自适应策略，但它们处理非线性相关数据的能力较弱，且这些方法未应用自适应字典策略。因此，从表3可以看出，AKDLRR的聚类性能优于这两种算法。

（2）AKDLRR只在ExtendYaleB数据集上是最快的。这是因为LRR/CFS可以通过闭式解一步获得结果。尽管AKDLRR最多只需要三次迭代即可收敛，但其与LRR/CFS之间仍存在一定差距。然而，尽管LRR/CFS是最快的方法，其聚类性能并不理想。将AKDLRR与其他算法进行比较，可以清楚地看到，AKDLRR在COIL20、ORL和FRDUE数据集上是速度最快的技术。尽管AKDLRR在三次迭代内收敛，但其复杂度仍为$O(n^3)$，因此在CCV和Scene15这两个大数据集上的计算时间较长。

（3）表3和表4表明，AKDLRR不仅具有最优的聚类性能，而且计算速度也非常快。

表 3-2 不同算法的聚类性能

Table 3-2 Clustering performance of different algorithms

数据集	指标	SSC	LRR	LRR/CFS	AOPL_ROOT	ACMK_SC	AKDLRR
ExtendYaleB	ACC	68.0	70.7	64.6	71.8	57.1	**84.3**
	NMI	83.7	78.7	71.9	85.9	71.5	**90.1**
	PUR	73.7	73.1	66.9	75.8	59.0	**86.0**
ORL	ACC	68.4	65.9	65.0	63.0	59.6	**71.6**
	NMI	83.6	80.5	80.0	77.1	76.1	**84.2**
	PUR	83.7	69.3	68.4	69.0	62.8	**75.0**
FRDUE (100)	ACC	77.6	77.9	87.9	89.5	80.3	**95.4**
	NMI	90.2	91.5	95.8	97.9	94.3	**98.8**
	PUR	83.8	82.9	90.4	93.9	83.4	**96.4**
FRDUE (all)	ACC	74.1	70.1	85.4	90.6	83.6	**93.1**

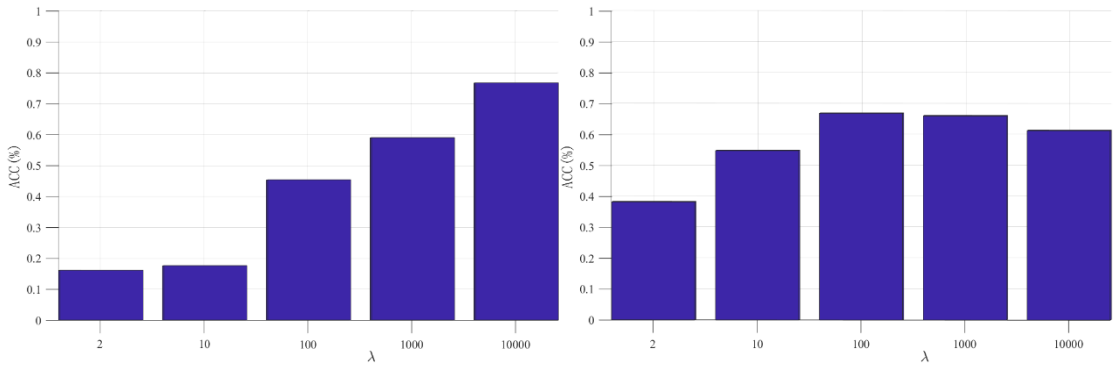
	NMI	87.9	90.5	95.2	97.9	95.4	**98.2**
	PUR	81.3	72.3	88.2	94.1	85.3	**94.4**
COIL20	ACC	77.5	66.2	67.9	84.2	47.9	**88.9**
	NMI	91.9	75.1	74.9	92.7	59.9	**92.8**
	PUR	84.8	69.1	70.5	88.6	50.6	**89.4**
Scene15	ACC	17.1	26.4	35.3	26.7	36.7	**37.2**
	NMI	8.2	23.1	36.7	27.7	38.8	**39.0**
	PUR	18.7	30.1	38.1	29.1	40.3	**40.9**
CCV	ACC	21.7	17.6	20.5	10.66	20.5	**23.9**
	NMI	18.6	16.1	17.9	3.8	18.0	**20.5**
	PUR	24.3	20.7	23.7	10.78	23.8	**26.7**

表 3-3 不同算法的运行时间

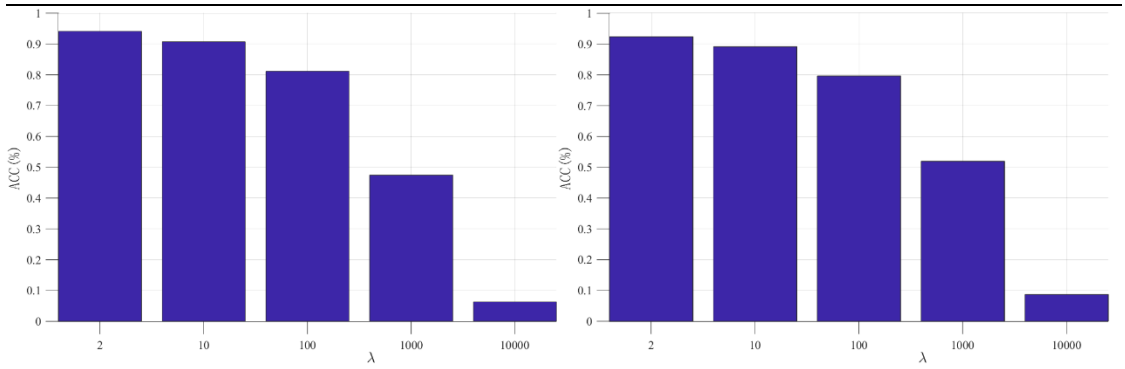
Table 3-3 Runtime of different algorithms

数据集	**SSC**	**LRR**	**LRR/CFS**	**AOPL_ROOT**	**ACMK_SC**	**AKDLRR**
ExtendYaleB	134.406	73.585	0.746	4.45	443.98	**0.339**
ORL	7.647	9.100	**0.071**	1.05	66.82	0.108
FRDUE (100)	44.223	271.171	**0.154**	46.13	629.64	5.424
FRDUE (all)	91.670	1337.40	**0.260**	160.56	1876.53	18.176
COIL20	36.051	109.28	**0.278**	36.314	413.48	2.380
Scene15	252.78	4637.02	**1.226**	937.08	2899.60	1416.75
CCV	356.78	9384.45	**1.280**	4184.31	4756.82	6832.45

3.3.3 参数敏感度分析



(a) ExtendYaleB (b) ORL



(c) FRDUE(100)　　(d) FRDUE(all)

图 3-2. 四个数据集上不同λ值对应的 AKDLRR 的 ACC 值

Figure 3-2. ACC values of AKDLRR corresponding to different values in the four data sets

在图3-2中，展示了不同λ值下，AKDLRR实现的ACC聚类性能。从图3-2不难看出，AKDLRR对其参数不敏感，在大多数情况下可以保持比竞争方法更好的聚类性能。对于AKDLRR方法中的参数，首先在[2,10,100,1000,10000]内搜索，然后在当前参数附近寻找最佳参数。

3.4 本章小结

本章提出了一种基于自适应核字典的低秩表示方法AKDLRR。AKDLRR解决了大多数子空间聚类方法无法通过核技术有效处理非线性关系的问题。同时，AKDLRR使用自适应字典来缓解固定字典导致的聚类性能不佳的问题。在五个数据集上将AKDLRR与其他几种算法进行了比较。从实验结果可以看出，AKDLRR具有最优的聚类性能，并且其计算速度也非常快。

第4章 具有闭式解的高效张量低秩表示算法

4.1 引言

如今，随着各种电子产品和互联网的不断发展，各种类型的高维数据，如图像、网络文本、视频等，不断涌现。对于机器学习和图像处理算法而言，高维数据会由于噪声效应和冗余特征而增加计算复杂度并影响算法性能，这种现象通常被称为“维度灾难”。事实上，这些高维数据通常位于低维结构中。例如，人脸图像虽然受到不同光照和遮挡的影响，但同一个人的脸部图像本质上位于同一个子空间中。因此，如何找到数据的低维子空间成为解决问题的关键。

为了解决上述问题，通常需要找到数据的低维子空间，然后将数据划分到各自的子空间中，这一过程通常称为SC。子空间聚类不仅可以从原始数据中去除噪声，还能更有效地处理低维数据。子空间聚类已应用于运动分割[55, 56]、场景聚类[57]等领域[58]。在所有子空间聚类方法中，有两种代表性方法，即SSC和LRR。SSC源于稀疏表示，而LRR则从RPCA演变而来。SSC和LRR都是子空间聚类中非常重要且具有代表性的方法。它们的主要区别在于，SSC对系数表示施加L1范数约束，而LRR对系数表示施加低秩约束。SSC和LRR都能有效恢复数据的多个子空间结构。此外，LRR还发展了几种变体，通过引入额外的正则化项来寻找低秩结构。在低秩表示系数矩阵上，使用诸如Ncut等聚类工具来完成最终的聚类。即使数据受到各种噪声的污染，这些方法仍然能够取得良好的效果。通常，LRR使用ALM来求解。

然而，这些方法都是基于矩阵的。随着数据维度的不断增加和数据规模的扩大，这些方法需要将数据压缩成矩阵形式。这种操作破坏了数据的内在结构，并且在压缩过程中会丢失一些信息。此外，这些方法的计算复杂度较高，计算效率也不尽如人意。为了解决这些问题，研究员们开始研究基于张量的子空间聚类算法[40, 59]。Kilmer等人[20]提出了一些与张量操作相关的定义，如t-积、t-SVD和TNN。在此基础上，Lu等人将RPCA扩展为张量鲁棒主成分分析（TRPCA）[15]。实验结果表明，TRPCA具有良好的性能。随后，TLRR被提出，它是LRR的泛化形式。TLRR能够有效恢复张量数据的低秩结构并推导样本聚类。同样，TLRR也使用ALM来优化其算法。然而，TLRR需要多次迭代才能获得解，并且每次迭代都需要计算t-积和t-SVD。不难看出，TLRR的计算复杂度较高，计算速度较慢。此外，还

有一些TLRR的改进算法，例如ETLRR和TLRSR等。ETLRR不仅考虑了数据中的拉普拉斯噪声，还考虑了高斯噪声，能够更好地恢复数据的低秩结构。TLRSR则认为TLRR仅考虑了数据的全局结构，为了捕捉数据的局部结构，TLRSR在TLRR的基础上对系数张量施加了稀疏约束。尽管这些算法能够取得比TLRR更好的结果，但它们的计算复杂度仍然非常高。

为了解决TLRR及其改进算法的问题，本章提出了一种具有闭式解的高效张量低秩表示方法（ETLRR/CFS），以提高TLRR的计算效率。与TLRR及其一些改进方法完全不同，ETLRR/CFS的解是闭式解。也就是说，它不需要通过迭代过程来求解ETLRR/CFS，而只需一步即可获得其解。因此，其计算速度非常快，计算复杂度也非常低。具体来说，ETLRR/CFS将TNN和Frobenius范数整合到一个统一的框架中，并给出了其闭式解。实验结果表明，ETLRR/CFS不仅比TLRR及其改进方法快得多，而且能够获得相似的聚类性能。

本章的主要贡献如下：

（1）本章提出了一种新的基于张量的子空间聚类方法，即ETLRR/CFS。

（2）ETLRR/CFS的理论推导表明，它具有闭式解。也就是说，它只需一步即可获得最终解，这大大提高了其计算效率。

（3）在六个数据集上的实验结果验证了ETLRR/CFS具有高效性和良好的性能。

4.2 具有闭式解的高效张量低秩表示算法

4.2.1 ETLRR/CFS 模型搭建

TLRR可以实现良好的聚类性能，但需要多次迭代。而且在迭代过程中需要计算大量的t-product和t-SVD，因此TLRR的计算复杂度高，计算速度慢。因此，本章提出了ETLRR/CFS方法，可以大大提高计算效率。它只需要一步就可以得到它的解，从而大大提高了计算效率。此外，ETLRR/CFS的聚类性能可与最先进的方法相媲美。

假设 $\boldsymbol{\mathcal{X}} \in R^{n_1 \times n_2 \times n_3}$ 是原始数据，通用的寻找原始数据的低秩结构的公式：

$$\min_{A,C} \|\boldsymbol{\mathcal{Z}}\|_{TNN} + \frac{\lambda}{2}\|\boldsymbol{\mathcal{E}}\|_{\ell} \quad s.t. \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Z}} + \boldsymbol{\mathcal{E}} \tag{4.1}$$

其中 $\boldsymbol{\mathcal{Z}} \in R^{n_2 \times n_2 \times n_3}$ 表示 $\boldsymbol{\mathcal{X}}$ 的低秩结构，$\|\cdot\|_l$ 表示各种噪声的正则化策略。例如，在TLRR中，L_1范数用于测量拉普拉斯噪声，而在ETLRR中，L_1范数和L_2范数同时

用于测量拉普拉斯噪声和高斯噪声。然而，虽然两种方法都能获得很好的实验结果，但计算速度相对较低。因此，为了能够得到封闭形式的解，ETLRR/CFS使用Frobenius范数而不是TLRR中的L_1范数。随后，公式(4.1)可以被改写成：

$$\min_{A,C}\|\mathcal{Z}\|_{TNN}+\frac{\lambda}{2}\|\mathcal{E}\|_F^2 \ \ s.t.\mathcal{X}=\mathcal{X}\mathcal{Z}+\mathcal{E} \tag{4.2}$$

传统算法通常将原始数据直接作为字典进行处理。然而，考虑到原始数据普遍存在噪声污染的问题，本研究采用数据分解的方法，将原始数据分离为纯净数据和噪声两个组成部分，即 $\mathcal{X}=\mathcal{L}+\mathcal{E}$，其中 $\mathcal{L}$ 是纯净数据张量，$\mathcal{E}$ 是噪声张量。随后，公式(4.2)可以被重写为：

$$\min_{\mathcal{A},\mathcal{C}}\|\mathcal{Z}\|_{TNN}+\frac{\tau}{2}\|\mathcal{E}\|_F^2 \ \ s.t.\mathcal{L}=\mathcal{L}*\mathcal{Z},\ \mathcal{X}=\mathcal{L}+\mathcal{E} \tag{4.3}$$

最后，可以将公式(4.3)中的等式约束放宽，得到最终的目标函数：

$$\min_{\mathcal{A},\mathcal{C}}\|\mathcal{Z}\|_{TNN}+\frac{\tau}{2}\|\mathcal{L}-\mathcal{L}*\mathcal{Z}\|_F^2+\frac{\alpha}{2}\|\mathcal{X}-\mathcal{L}\|_F^2 \tag{4.4}$$

4.2.2 ETLRR/CFS 求解

上述目标函数可以使用如下定理3求解。

定理3：对于给定张量 $\mathcal{L}$，其t-SVD可表示为 $\mathcal{X}=\mathcal{U}*\mathcal{S}*\mathcal{V}^T$。则公式(4.4)的最优解为：

$$\begin{aligned}\mathcal{L}^*&=\mathcal{U}*\mathcal{D}*\mathcal{V}^T\\ \mathcal{Z}^*&=\mathcal{V}*\mathcal{T}*\mathcal{V}^T\end{aligned} \tag{4.5}$$

其中 $\mathcal{D}$ 是一个大小为 $n_1\times n_2\times n_3$ 的f-对角张量，其对角元素满足：

$$\mathcal{S}(i,i,j)=\varphi\left(\mathcal{D}(i,i,j)\right)=\begin{cases}\mathcal{D}(i,i,j)+\dfrac{1}{\alpha\tau\mathcal{D}^3(i,i,j)}, & \mathcal{D}(i,i,j)>\dfrac{1}{\sqrt{\tau}}\\ \mathcal{D}(i,i,j)+\dfrac{\tau}{\alpha\mathcal{D}(i,i,j)}\mathcal{S}(i,i,j), & \mathcal{D}(i,i,j)\le\dfrac{1}{\sqrt{\tau}}\end{cases} \tag{4.6}$$

$\mathcal{T}$ 同样是一个大小为 $n_1\times n_2\times n_3$ 的f-对角张量，其对角元素满足：

$$\mathcal{T}(i,i,j)=\begin{cases}1-\dfrac{1}{\tau\mathcal{D}^2(i,i,j)}, \mathcal{D}(i,i,j)>\dfrac{1}{\sqrt{\tau}}\\ 0, \qquad\quad \mathcal{D}(i,i,j)\le\dfrac{1}{\sqrt{\tau}}\end{cases} \tag{4.7}$$

在证明定理2之前，首先给出定理3。

定理4：设定 $D=U\Sigma V^T$ 是给定矩阵 D 的SVD。那么公式

$$\min_{A,C}\|Z\|_*+\frac{\tau}{2}\|L-LZ\|_F^2+\frac{\alpha}{2}\|D-Z\|_F^2 \tag{4.8}$$

的最优解为：

$$\begin{aligned}&\mathrm{A}=U\Lambda V^T\\ &C=V_1(I-\frac{1}{\tau}\Lambda_1^{-2})V_1^T\end{aligned} \tag{4.9}$$

其中，$\Lambda=diag(\lambda_1,...,\lambda_n)$ 中的每个元素均可表示为 $\Sigma=diag(\sigma_1,...,\sigma_n)$ 中对应元素的某种函数关系：

$$\sigma=\psi(\lambda)=\begin{cases}\lambda+\frac{1}{\alpha\tau}\lambda^{-3} & \text{if } \lambda>1/\sqrt{\tau}\\ \lambda+\frac{\tau}{\alpha}\lambda & \text{if } \lambda\le1/\sqrt{\tau}\end{cases} \tag{4.10}$$

矩阵 $U=[U_1\ U_2]$，$\Lambda=diag(\Lambda_1,\Lambda_2)$，$V=[V_1\ V_2]$ 可以根据集合 $I_1=\{i:\lambda_i>1/\sqrt{\tau}\}$ 和 $I_2=\{i:\lambda_i\le1/\sqrt{\tau}\}$ 划分。

下面给出定理3的证明过程。

证明：在傅里叶域中，公式(4.4)可以被改写为：

$$\begin{aligned}&\arg\min_{\mathcal{A},\mathcal{C}}\|\mathcal{Z}\|_{TNN}+\frac{\tau}{2}\|\mathcal{L}-\mathcal{L}*\mathcal{Z}\|_F^2+\frac{\alpha}{2}\|\mathcal{X}-\mathcal{L}\|_F^2\\ &=\arg\min_{\bar{A},\bar{C}}\frac{1}{n_3}\left(\|\bar{Z}\|_*+\frac{\tau}{2}\|\bar{L}-\bar{L}*\bar{Z}\|_F^2+\frac{\alpha}{2}\|\bar{X}-\bar{L}\|_F^2\right)\\ &=\arg\min_{\bar{A}^{(i)},\bar{C}^{(i)}}\frac{1}{n_3}\sum_{j=1}^{n_3}\left(\|\bar{Z}^{(j)}\|_*+\frac{\tau}{2}\|\bar{L}^{(j)}-\bar{L}^{(j)}*\bar{Z}^{(j)}\|_F^2+\frac{\alpha}{2}\|\bar{X}^{(j)}-\bar{L}^{(j)}\|_F^2\right)\end{aligned} \tag{4.11}$$

设定 $\overline{\mathrm{X}}^{(i)}$ 的SVD为 $\bar{X}^{(j)}=\bar{U}^{(j)}\bar{S}^{(j)}\bar{V}^{(j)T}$，随后可以使用定理4解决公式(4.11):

$$\begin{aligned}&\bar{A}^{(j)}=\bar{U}^{(j)}\bar{D}^{(j)}\bar{V}^{(j)T}\\ &\bar{C}^{(j)}=\bar{V}_1^{(j)}\bar{T}^{(j)}\left(\bar{V}_1^{(j)}\right)^T\end{aligned} \tag{4.12}$$

其中 $\overline{D}^{(j)} = diag(\bar{\lambda}_1^{(j)}..., \bar{\lambda}_n^{(j)})$ 中的每个元素均可表示为 $\bar{S}^{(j)} = diag(\bar{\sigma}_1^{(j)}..., \bar{\sigma}_n^{(j)})$ 中对应元素的某种函数关系：

$$\bar{\sigma}^{(j)} = \varphi(\bar{\lambda}^{(j)}) = \begin{cases} \bar{\lambda}^{(j)} + \dfrac{1}{\alpha\tau}(\bar{\lambda}^{(j)})^{-3}, & \bar{\lambda}^{(j)} > \dfrac{1}{\sqrt{\tau}} \\ \bar{\lambda}^{(j)} + \dfrac{\tau}{\alpha}\bar{\lambda}^{(j)}, & \bar{\lambda}^{(j)} \le \dfrac{1}{\sqrt{\tau}} \end{cases} \tag{4.13}$$

同时，$\bar{U}^{(j)} = \left[\bar{U}_1^{(j)} \quad \bar{U}_2^{(j)}\right]$，$\bar{\Sigma}^{(j)} = diag\left(\bar{D}_1^{(j)} \quad \bar{D}_2^{(j)}\right)$ 和 $\bar{V}^{(j)} = \left[\bar{V}_1^{(j)} \quad \bar{V}_2^{(j)}\right]$ 都可以根据集合 $I_1 = \left\{i : \bar{\lambda}_i > \sqrt{\tau}\right\}$ 和 $I_2 = \left\{i : \bar{\lambda}_i \le \sqrt{\tau}\right\}$ 划分。

设定

$$\bar{\mathcal{D}}(i,i,j) = \bar{\lambda}_i^{(j)} \tag{4.14}$$

和

$$\bar{T}^{(j)} = diag\left(I - \frac{1}{\tau}\left(\bar{D}_1^{(j)}\right)^{-2} \quad \mathbf{0}\right) \tag{4.15}$$

其中 0 是一个和 $\bar{\Lambda}_2^{(i)}$ 具有相同维度的全零矩阵。随后，得到 $\mathcal{T} = ifft\left(\bar{\mathcal{T}}, \ [\,], \ 3\right)$，其中 $\bar{\mathcal{T}}$ 是一个额叶切片由 $\bar{T}^{(i)}$ 组成的张量。最后，可以得到公式(4.4)的最优解：

$$\begin{aligned} \mathcal{C} &= \mathcal{V} * \mathcal{T} * \mathcal{V}^T \\ \mathcal{A} &= \mathcal{U} * \mathcal{D} * \mathcal{V}^T \end{aligned} \tag{4.16}$$

□

下面给出算法2求解ETLRR/CFS。

算法 2: ETLRR/CFS

输入: 原始数据 $\mathcal{X} \in R^{n_1 \times n_2 \times n_3}$，参数 α 和 τ

初始化: $\mathcal{A} = \mathcal{C} = 0$

1.计算 $\bar{\mathcal{X}} = fft(\mathcal{X}, [\,], 3)$

2.Perform t-SVD of $\bar{\mathcal{X}}$, i.e., $\bar{\mathcal{X}} = \bar{\mathcal{U}} * \bar{\mathcal{S}} * \bar{\mathcal{V}}^T$

3.通过公式(4.13)计算 $\bar{\mathcal{D}}$;

4.通过公式(4.15)计算 $\bar{\mathcal{T}}$;

5.计算: $\bar{\mathcal{A}} = \bar{\mathcal{U}} * \bar{\mathcal{D}} * \bar{\mathcal{V}}^T$;

6.计算: $\overline{\mathcal{C}} = \overline{\mathcal{V}} * \overline{\mathcal{T}} * \overline{\mathcal{V}}^T$;

7.计算: $\mathcal{A} = ifft(\overline{\mathcal{A}}, [\ ], 3)$

8.计算: $\mathcal{C} = ifft(\overline{\mathcal{C}}, [\ ], 3)$

输出: $\mathcal{A}, \mathcal{C}$

通过算法2，ETLRR/CFS可以得到低秩结构 $\mathcal{Z}$ 。注意原始数据张量 $\mathcal{X}$ 的额叶切片 $\mathcal{X}^{(j)}$ 的自表示系数为 $\mathcal{Z}^{(j)}$ 。因此，ETLRR/CFS可以用额叶切片构造相似矩阵 S ：

$$S = \frac{1}{2n_3} \sum_{j=1}^{n_3} (\left| \mathcal{Z}^{(j)^*} \right| + \left| \mathcal{Z}^{(j)} \right|) \tag{4.17}$$

随后，ETLRR/CFS使用算法3得到最终的聚类结果。

算法 3: 使用ETLRR/CFS聚类

输入: 原始数据 $\mathcal{X} \in R^{n_1 \times n_2 \times n_3}$, 簇的数量 c

输出: 聚类结果

1.使用算法2获得 $\mathcal{A}, \mathcal{C}$.

2.通过公式(4.17)构建相似矩阵 S .

3. 通过Ncut将样本分成 c 个类.

4.2.3 ETLRR/CFS 算法复杂度分析

如算法2所述，本算法的计算复杂度主要由傅里叶变换和t-SVD两部分决定。基于此，ETLRR/CFS的复杂度分析将着重考察这两个关键环节的计算代价。具体而言，设定原始数据 $\mathcal{X} \in R^{n_1 \times n_2 \times n_3}$ ，计算其傅里叶变换的复杂度为 $O(n_1 n_2 n_3 \log(n_1 n_2 n_3))$ ，计算其t-SVD的复杂度为 $O(n_3 n_1^2 n_2)$ 。因此，ETLRR/CFS算法的复杂度为 $O(n_1 n_2 n_3 \log(n_1 n_2 n_3) + n_3 n_1^2 n_2)$ 。

4.3 实验及分析

4.3.1 实验设置

本节将ETLRR/CFS与几种不同的方法在不同数据集上进行比较，以展示ETLRR/CFS的优越性能。所有实验均在Matlab2019b环境下进行，使用的计算机配置为i7-8750H处理器、2.20GHzCPU和16GB 内存。在实验中，选择了五个基准数据集（ORL, YaleB, ExtendYaleB, FRDUE, Yale），其信息汇总在表 4-1 中。

表4-1 数据集的统计数据

Table 4-1 Statistics of the dataset

数据集	样本数	簇	尺寸
ExtendYaleB	840	30	80×60
ORl	400	10	32×32
Yale	165	11	32×32
YaleB	650	65	50×50
FRDUE (100)	1980	20	25×22
FRDUE (all)	3040	20	25×22

在实验中，使用 LRR, SSC, TLRR(S-TLRR和R-TLRR), ETLRR,TLRSR （LRR与SSC介绍见第三章实验部分）与ETLRR/CFS进行比较。

S-TLRR 和 R-TLRR[21]: TLRR通过使用张量操作将LRR向张量进行拓展。当使用原始数据作为字典时，称为 S-TLRR；当使用通过 TRPCA 预处理的数据作为字典时，称为 R-TLRR。

ETLRR[22]: ETLRR 比 TLRR 更多地考虑了高斯噪声对数据的影响。通过处理更多的噪声，可以获得更准确的结果。

TLRSR[23]:TLRSR 在 TLRR 的模型基础上，对系数表示张量添加了稀疏表示约束。这确保了系数表示张量能够捕捉全局和局部结构。因此，TLRSR 能够更好地恢复数据的低秩结构，并实现更好的聚类性能。

4.3.2 对比实验结果与分析

为了量化不同聚类算法的性能，使用ACC,NMI和PUR作为评价指标。上述评价指标的范围为[0-1]。值越高，聚类效果越好。

表4-2展示了所有对比算法在不同数据集上的性能，最佳性能以粗体标出。可以看出：1.基于矩阵的低秩表示方法（如SSC、LRR）的聚类性能通常不如基于张量的低秩表示方法（如TLRR、ETLRR、TLRSR）。这表明，如果将张量数据压缩为矩阵，张量数据的张量结构会被破坏，并且在压缩过程中会丢失一些信息；2.总体而言，ETLRR/CFS在所有数据集上都取得了良好的聚类性能。例如，在FRDUE和FRDUE100数据集上，ETLRR/CFS表现最佳。在其他数据集上（如ExtendYaleB、ORL等），ETLRR/CFS虽然不是最优的，但接近最优水平。

表4-2展示了各算法在不同数据集上的运行时间。通过比较ETLRR/CFS与TLRR、ETLRR和TLRSR在所有数据集上的表现，可以看出ETLRR/CFS是最快的，比其他方法快数倍甚至数十倍。从表3和表4中可以看出，ETLRR/CFS不仅比TLRR及其改进方法快得多，而且能够获得相似的聚类性能。

表4-2 聚类性能的实验结果

Table 4-2 Experimental results of clustering performance

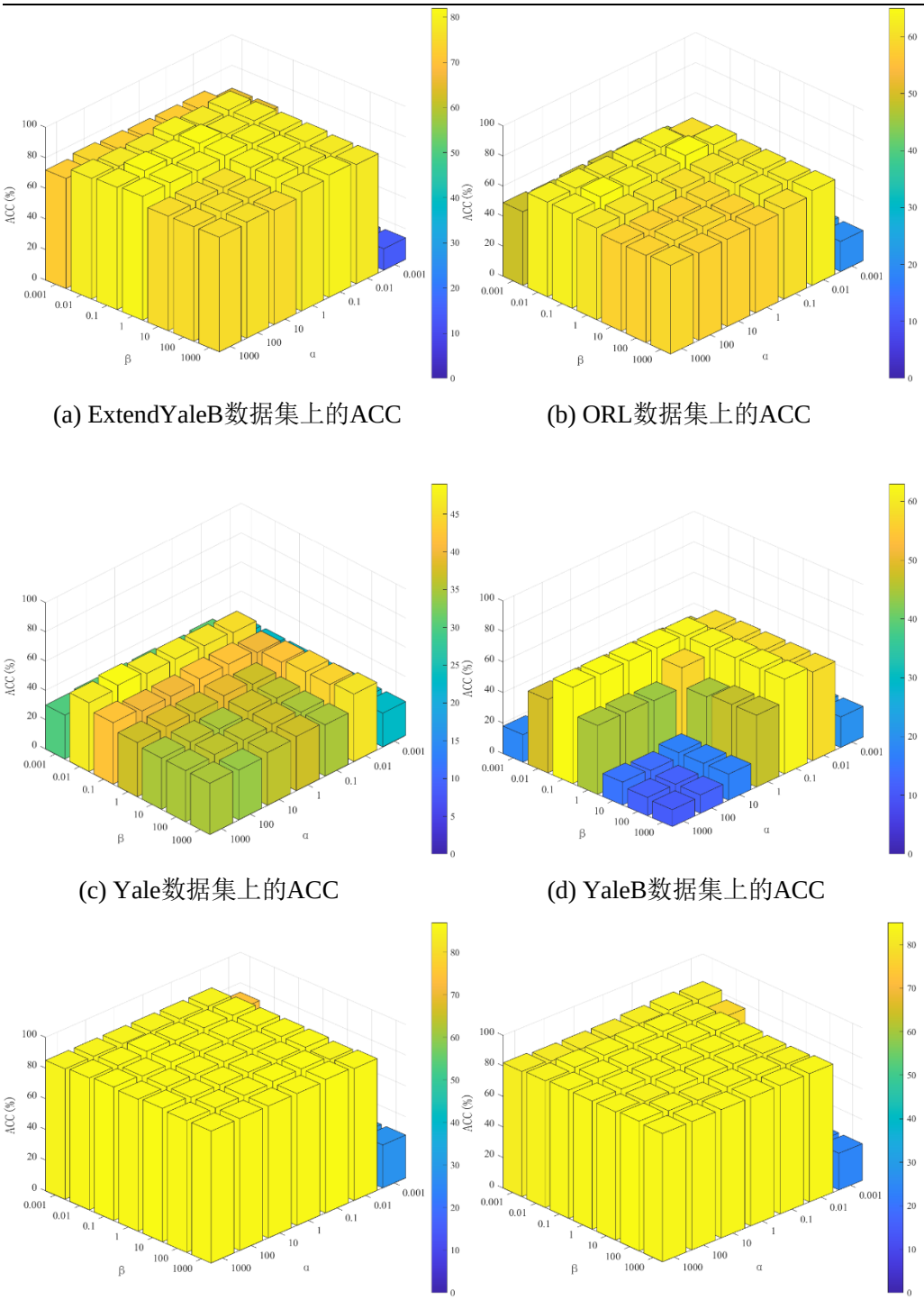
数据集	指标	SSC	LRR	S-TLRR	R-TLRR	ETLRR	TLRSR	ETLRR/CFS
ExtendYaleB	ACC	0.680	0.707	0.829	0.839	**0.841**	0.832	0.817
	NMI	0.837	0.787	0.889	**0.908**	0.904	0.900	0.879
	PUR	0.737	0.731	0.843	0.860	**0.861**	0.853	0.831
ORL	ACC	**0.684**	0.659	0.603	0.558	0.619	0.620	0.655
	NMI	**0.836**	0.805	0.748	0.719	0.777	0.775	0.791
	PUR	**0.837**	0.693	0.632	0.591	0.649	0.651	0.684
Yale	ACC	**0.631**	0.523	0.377	0.369	0.484	0.524	0.520
	NMI	0.559	**0.572**	0.436	0.435	0.537	0.559	0.564
	PUR	**0.632**	0.546	0.409	0.394	0.514	0.549	0.545
YaleB	ACC	**0.761**	0.443	0.478	0.613	0.725	0.740	0.719
	NMI	0.683	0.380	0.452	0.599	0.673	**0.684**	0.669
	PUR	**0.761**	0.470	0.485	0.613	0.729	0.741	0.720
FRDUE (100)	ACC	0.776	0.779	0.867	0.849	0.866	**0.870**	**0.870**
	NMI	0.902	0.915	0.956	0.949	0.957	**0.961**	0.958
	PUR	0.838	0.829	0.894	0.879	0.893	**0.899**	0.897
FRDUE (all)	ACC	0.741	0.701	0.843	0.830	0.839	0.845	**0.847**
	NMI	0.879	0.905	0.950	0.944	0.951	**0.955**	0.952
	PUR	0.813	0.723	0.875	0.862	0.870	**0.876**	0.877

表4-3 算法运行时间结果

Table 4-3 Algorithm running time results

数据集	SSC	LRR	S-TLRR	R-TLRR	ETLRR	TLRSR	ETLRR/CFS
ExtendYaleB	205.361	75.947	272.847	311.053	189.495	966.737	**4.671**
ORL	88.569	21.490	25.967	30.754	19.026	92.454	**0.343**
Yale	61.681	6.240	13.716	16.264	35.281	47.869	**0.069**
YaleB	61.747	41.943	35.338	37.107	30.750	109.934	**0.222**
FRDUE (100)	63.702	433.336	49.397	60.106	41.042	371.229	**5.908**
FRDUE (all)	145.037	421.191	84.706	106.693	73.843	383.626	**14.807**

4.3.3 参数敏感度分析



(a) ExtendYaleB数据集上的ACC　　(b) ORL数据集上的ACC

(c) Yale数据集上的ACC　　(d) YaleB数据集上的ACC

(e) FRDUE (100)数据集上的ACC　　(f) FRDUE (all)数据集上的ACC

图 4-1 所有数据集上不同参数值对应的 ACC

Figure 4-1 ACC corresponding to different parameter values on all data sets

从图4-1不难看出，ETLRR/CFS对参数并不是非常敏感。随着参数的变化，ETLRR/CFS的聚类性能在大部分情况下保持平衡。

4.4 本章小结

本章提出了一种ETLRR/CFS方法。过去提出的大多数低秩张量表示方法需要大量的计算时间。尽管这些方法能够取得良好的结果，但由于需要多次迭代求解，并且在每次迭代中都需要计算t-积和t-SVD，因此计算速度较慢。为此，本章提出了一种具有闭式解的低秩张量表示算法。ETLRR/CFS不仅能够达到与之前算法相当的聚类性能，还可以通过一步计算得到结果，这意味着其计算速度非常快。在六个数据集上将ETLRR/CFS与其他几种方法进行了比较。实验结果表明，ETLRR/CFS具有高效性和有效性。

第5章 基于灵活图融合的多视图子空间聚类算法

5.1 引言

在大数据时代，数据扮演着非常重要的角色。分析这些数据并挖掘其潜在的有价值信息是聚类分析的主要任务。过去，研究人员提出了许多聚类方法，例如K-means和谱聚类，这些方法取得了显著的成果。然而，这些传统方法在处理高维数据时仍存在不足。近年来，子空间聚类因其能够更好地处理高维数据而成为一个热门研究方向。

子空间聚类基于所谓的子空间表示假设，即高维数据空间可以划分为多个子空间的并集，而同一类数据样本可以被划分到同一个子空间中。这是一种从高维空间中寻找低维空间的方法。为了避免寻找重构字典矩阵，一些基于自表达子空间模型的聚类算法被提出。在自表达子空间模型中，任何数据样本都可以通过同一类别的其他样本线性表示。

2009年，Elhamifar等人提出的SSC是最早的自表示子空间聚类算法之一。它通过L1范数约束图矩阵，使其尽可能稀疏，这在一定程度上能够确保在数据受损时正常恢复子空间。不久之后，Liu等人提出了LRR，它通过核范数约束重构的图矩阵，使其尽可能低秩。这种方法揭示了数据的相关性结构，并通过低秩约束恢复子空间结构。由于SSC和LRR缺乏解析解，计算它们非常耗时。Lu等人提出的最小二乘回归（LSR）[55]使用F范数约束图矩阵，由于使用解析解,因此复杂度较低。最近，Xu等人提出了尺度单纯形表示（SSR）[60]，它可以对图矩阵的列和进行尺度约束。

随后，许多改进的单视图算法被提出，并取得了良好的效果。然而，上述算法可能仅在单视图数据上表现良好，而在多视图数据上表现较差。与单视图数据相比，多视图数据具有互补性和一致性的优势，对聚类分析非常有帮助。因此，多视图聚类通常比单视图聚类更具优势。

Gao等人[32]提出了MSC，MSC使用所有视图来获得共识聚类指标，并保证不同视图之间的一致性。Maria等人[33]提出了MLRSSC，MLRSSC通过联合学习多个视图的子空间表示，构建一个共享的亲和矩阵。同时，MLRSSC引入低秩和稀疏约束来构建亲和矩阵，确保其既能捕捉数据的全局结构，又能保持局

部结构的稀疏性。Zhang等人提出了LMSC [34]，与大多数现有算法不同，LMSC利用所有视图来寻找潜在表示并重构潜在矩阵。Zheng等人提出了FCMSC[37]，该方法使用L_{21}范数两次重构数据，并对相似矩阵施加低秩约束。Zhao等人提出了CRAA[38]，CRAA通过构建所有视图的表示来学习共享结构，并通过鲁棒主成分分析获得更优质的相似矩阵。Yao等人提出了DSSSR[39]，DSSSR可以减少噪声数据的负面影响并拼接多视图数据。

尽管上述多视图算法取得了显著成果，但它们仍存在一些不足。首先，共识图的列和被限制为1，这在实际应用中不够灵活，无法有效处理数据点被噪声或异常值污染的情况。这种约束可能导致同一类别中被噪声污染的点被错误地划分到不同类别中，从而影响聚类性能。其次，共识图的聚类结构通常未被充分考虑，这些因素可能导致聚类效果不佳。

为此，本章提出了一种基于灵活且最优多图融合方法的多视图聚类算法（MVC/FOMF）。首先，MVC/FOMF通过自表示学习获得每个视图的相似图。其次，MVC/FOMF将这些图融合为一个共识图，并将其列和约束为S（$0 < S \leq 1$），以提高其灵活性。通过调整S，MVC/FOMF可以灵活调整共识图的生成和判别特性，以找到最佳的聚类性能。第三，MVC/FOMF对共识图的拉普拉斯矩阵施加秩约束，以学习最佳的聚类结构。最后，所有这些步骤被统一到一个框架中，并设计了基于交替乘子法的优化过程。更重要的是，MVC/FOMF复杂度低于许多代表性算法。实验表明，MVC/FOMF表现出了非常令人鼓舞的性能。总的来说，本章的主要贡献如下：

（1）为了解决共识图灵活性不足和聚类结构非最优的问题，本章提出了一种基于灵活且最优多图融合方法的新型多视图聚类算法。

（2）在MVC/FOMF中，可以通过调整S（$0 < S \leq 1$）使共识图更加灵活，并对共识图的拉普拉斯矩阵施加秩约束，以学习最优的聚类结构。

（3）为MVC/FOMF设计了一种高效的优化算法。与最先进的算法相比，MVC/FOMF在某些数据集上表现出非常令人鼓舞的性能。

5.2 基于灵活图融合的多视图子空间聚类算法

5.2.1 MVC/FOMF 模型搭建

一般来说，大多数多视图聚类算法属于以下模型：

$$\begin{aligned}&\min_{Z^v}\sum_{v=1}^{m}\mathcal{L}(X^v,X^vZ^v)+\Omega(Z^v)\\&s.t.\ \ 0\le Z^v\le 1,\ \ \mathbf{1}^TZ^v=\mathbf{1}^T\end{aligned}\tag{5.1}$$

其中$\mathcal{L}(X^v,X^vZ^v)$和$\Omega(Z^v)$分别表示重构损失项与对Z^v施加的正则化项。通过融合多个图Z^v得到一个共识矩阵，MVC/FOMF可以从公式(5.1)中得到具体的目标函数，将其改写为：

$$\begin{aligned}&\min_{Z^v,Z^*}\sum_{v=1}^{m}\|X^v-X^vZ^v\|_F^2+\alpha\|Z^v\|_F^2+\beta\|Z^v-Z^*\|_F^2\\&\text{s.t. }0\le Z^v\le 1,1^TZ^v=1^T\end{aligned}\tag{5.2}$$

其中Z^*表示共识图矩阵。但是，如果Z^v的列和受非负约束约束，则列和为1。这些约束可能会限制其自表示能力，并在一定程度上削弱聚类性能。因此，MVC/FOMF将Z上的约束转移到共识图Z^*上，公式(5.2)可以改写为：

$$\begin{aligned}&\min_{Z^v,Z^*}\sum_{v=1}^{m}\|X^v-X^vZ^v\|_F^2+\alpha\|Z^v\|_F^2+\beta\|Z^v-Z^*\|_F^2\\&\text{s.t. }0\le Z^*\le 1,1^TZ^*=1^T\end{aligned}\tag{5.3}$$

然而，如果Z^*的列和被严格约束为1，这就限制了共识图的灵活性。为了使共识图更加灵活，MVC/FOMF对公式(5.3)中的约束进行修改，可以将其重新表述为：

$$\begin{aligned}&\min_{Z^v,Z^*}\|X^v-X^vZ^v\|_F^2+\alpha\|Z^v\|_F^2+\beta\|Z^v-Z^*\|_F^2\\&\text{s.t.}\ \ \ 0\le Z^*\le 1,\mathbf{1}^TZ^*=s\mathbf{1}^T,0<s\le 1\end{aligned}\tag{5.4}$$

其中正则化参数$\alpha>0$，$\beta>0$。

定理5：相似图中连通分量的个数等于对应拉普拉斯矩阵的特征值0的个数。

如定理5所述，理想的聚类结构应使相似图矩阵中连接分量的数量等于聚类的数量。使共识图符合此结构有助于后续的聚类。随后，可以使用如下约束：

$$rank(L)=n-k\tag{5.5}$$

其中 $L=D-S\in R^{n\times n}$ 是拉普拉斯矩阵。此处， $S=(Z^*+(Z^*)^T)/2$ 是邻接矩阵，度矩阵 $D\in R^{n\times n}$ 定义为一个对角矩阵，其第 i 个对角元素为 $\sum_j(z_{ij}^*+z_{ji}^*)/2$ 。如果直接将公式(5.5)加入到公式(5.4)中，则难以直接求解。得益于Ky Fan的定理，可以将其改写为：

$$\min_P Tr(P^TLP)\quad \text{s.t.}P^TP=I \tag{5.6}$$

其中指示矩阵 $P\in R^{n\times k}$ 是由 L 的前 k 个最小特征向量组成的。最后，我们将公式(5.6)和公式(5.4)融合进一个框架之中，得到MVC/FOMF的目标函数：

$$\begin{aligned}&\min_{Z^v,Z^*,P}\|X^v-X^vZ^v\|_F^2+\alpha\|Z^v\|_F^2+\beta\|Z^v-Z^*\|_F^2+\gamma Tr(P^TLP)\\&\text{s.t.}\quad 0\le Z^*\le 1,\mathbf{1}^TZ^*=s\mathbf{1}^T,0<s\le 1,P^TP=I\end{aligned} \tag{5.7}$$

MVC/FOMF的输出是共识图 Z^* 和聚类指标矩阵 P ， Z^* 和 P 都可以用于计算聚类结果。本文选择指标矩阵 P 进行K-means聚类，得到最终的标签。

5.2.2 MVC/FOMF 模型求解

本节设计了一种用于MVC/FOMF优化的交替乘法器。

Z^v-子问题：当 Z^* 和 P 都不变时，目标函数可以被转换为如下问题：

$$\min_{Z^v}\sum_{v=1}^{m}\|X^v-X^vZ^v\|_F^2+\alpha\|Z^v\|_F^2+\beta\|Z^v-Z^*\|_F^2 \tag{5.8}$$

通过对公式(5.8)求导并令其等于零，可以得到结果：

$$Z^v=((X^v)^TX^v+(\alpha+\beta)I)^-((X^v)^TX^v+\beta Z^*) \tag{5.9}$$

Z^*-子问题：当 Z^v 和 P 都不变时，目标函数可以被转换为如下问题：

$$\begin{aligned}&\min_{Z^*}\gamma Tr(P^TLP)+\beta\sum_{v=1}^{m}\|Z^v-Z^*\|_F^2\\&\text{s.t.}\quad \mathbf{1}^TZ^*=s\mathbf{1}^T,Z^*\ge 0\end{aligned} \tag{5.10}$$

为了公式(5.10)更好求解，我们设定 L 中的 D 为常数矩阵。此外，可以采用一种两步近似法来优化 Z^* 。

在第一步中，首先将公式(5.10)改写为：

$$\min_{Z^*} -\gamma Tr(P^T[Z^* + (Z^*)^T]/2P) + \beta\sum_{v=1}^{m} \| Z^v - Z^* \|_F^2 \tag{5.11}$$

对公式(5.11)求导，令结果为零。则可得 Z^*：

$$\tilde{Z} = (\gamma PP^T + 2\beta\sum_{v=1}^{m} Z^v)/(2m\beta) \tag{5.12}$$

在第二步中，将 $\tilde{Z}$ 投影到一个约束空间中，从而可以得到 Z^* 的近似解。对于每一列，求解以下问题：

$$\min_{Z_{:,i}^* \geq 0, \mathbf{1}^T Z_{:,i}^* = s} \| Z_{:,i}^* - \tilde{Z}_{:,i} \|_2^2 \tag{5.13}$$

为了求解公式(5.13)，引入拉格朗日乘子法，将其转化为以下形式：

$$\mathcal{L}(Z_{:,i}^*, \eta, \xi) = \frac{1}{2} \| Z_{:,i}^* - \tilde{Z}_{:,i} \|^2 + \eta(\mathbf{1}^T Z_{:,i}^* - s) - Z_{:,i}^* \xi^T \tag{5.14}$$

其中 η 和 ξ 是拉格朗日乘子，Z^* 的第 i 列优化可以通过对公式(5.14)求导并将每个元素的结果设为零来实现：

$$\begin{aligned} &\frac{\partial \mathcal{L}(Z_{:,i}^*, \eta, \xi)}{\partial Z_{j,i}^*} = Z_{j,i}^* - \tilde{Z}_{j,i} + \eta - \xi_j = 0 \\ &Z_{j,i}^* = \tilde{Z}_{j,i} - \eta + \xi_j \end{aligned} \tag{5.15}$$

根据公式(5.15)，$Z_{j,i}^* \geq 0, \mathbf{1}^T Z_{:,i}^* = s$ 和Karush–Kuhn–Tucker条件，可以得到：

$$\sum_{j=1}^{n} Z_{j,i}^* = \sum_{j=1}^{\rho} Z_{j,i}^* = \sum_{j=1}^{\rho} \tilde{Z}_{j,i} - \eta = s \tag{5.16}$$

其中 $0 < \rho \leq n$ 是非负元素的个数。为了提高计算效率，我们将 $\tilde{Z}_{:,i}$ 排序为 μ，其中 $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$。然后，可以用下面的问题来求解 ρ：

$$\rho(s, \mu) = \max\{ j \in [n] : \mu_j - \frac{1}{j}(\sum_{r=1}^{j} \mu_r - s) > 0 \} \tag{5.17}$$

因此，μ 的最优解可以通过公式(5.17)和公式(5.16)得到

$$\eta = \frac{1}{\rho}(\sum_{j=1}^{\rho} \tilde{Z}_{j,i} - s) \tag{5.18}$$

最后，根据公式(5.18)，公式(5.17)和公式(5.16)，可以得到 $Z^*_{:,i}$ 的每个元素的最优解：

$$Z^*_{j,i} = \max(\tilde{Z}_{j,i} - \eta, 0) \tag{5.19}$$

P-子问题：当 Z^v 和 Z^* 都不变时，目标函数可以被转换为如下问题：

$$\begin{aligned} &\min_P Tr(P^T L P) \\ &\text{s.t. } P^T P = I \end{aligned} \tag{5.20}$$

由于 P 是一个正交矩阵，并且受到谱聚类解的启发， P 的解可以转化为 $L = U\sum U^T$ 的特征值分解问题。因此， P 的列向量是对应前 k 个最小特征值的特征向量。

具体来说，MVC/FOMF的优化可以用算法1来概括，具体算法如下：

算法 4: MVC/FOMF

输入: 原始数据 X^v , 参数 $k, \alpha, \beta, \gamma, s$

初始化: $Z^* = 0$， $P = 0$.

1.判断收敛

2. 通过求解问题(5.9)更新 Z^v;

3. 通过求解问题(5.19)更新 Z^*;

4. 通过求解问题(5.20)更新 P

5.结束

输出: Z^* 和 P

5.2.3 MVC/FOMF 算法复杂度

如算法4所示，MVC/FOMF的主要计算复杂度可分为三个部分。更新 Z^v 的复杂度为 $\mathcal{O}(d^v \times n^2 + n^3)$ ，其中 d^v 是 X^v 的维度， $\mathcal{O}(d^v \times n^2)$ 是计算 $X^v(X^v)^T$ 的复杂度。更新 Z^* 的复杂度为 $\mathcal{O}(n^2 \log n)$ ，更新 P 的复杂度为 $\mathcal{O}(k^3)$ 。由于 $X^v(X^v)^T$ 只需计算一次，因此MVC/FOMF的总计算复杂度为

$\mathcal{O}(md_v n^2 + tmn^3 + tn^2 \log n + tk^3)$，其中$t$是MVC/FOMF的迭代次数。此外，在实验部分展示了算法在部分数据集上的实际运行时间。

5.3 实验及分析

5.3.1 实验设置

本节将提出的 MVC/FOMF 算法与一些代表性算法在多个数据集上进行比较，以展示其聚类性能。其次，本节比较了所有对比算法在部分数据集上的时间消耗。所有代码均在配备 Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 的台式计算机上使用 Matlab 2019b 运行。

数据集说明

实验选择了6个图像数据集和1个新闻数据集来验证算法的可靠性，包括MSRCV1、Coil20、ORL、YaleB、EyaleB10、BBCsport和Handwritten数据集。表5-1则详细列出了这些数据集的特征（维度、视图、类别和样本数量）。

MSRCV1：包含240张图像和8个对象类别，具有粗略的像素级标签。我们选择了其中的210张图像和7个类别进行实验。

Coil20：包含20个物体，每个物体水平旋转360度，每5度拍摄一张图像，因此每个物体有72张图像。

ORL：包含400张人脸图像，涉及40个人，每人10张图像。这些图像是在不同光照条件和面部表情下拍摄的。

YaleB：包含10个人的5850张多姿态、多光照图像。我们为每个人选择了65张图像。

EyaleB10：包含28个人在9种姿态和64种光照条件下的16128张图像。我们选择了其中的640张图像和10个人进行实验。

BBCsport：包含2004至2005年间BBC Sport网站上644篇体育新闻，涵盖5个主题。

Handwritten：该数据集由来自UCI机器学习库的0到9数字组成，共包含2,000个数据点。

表5-1 七个数据集的统计信息

Table 5-1 Statistics of the seven datasets

数据集	样本	视图	簇	*d*1	*d*2	*d*3	*d*4	*d*5	*d*6
Coil20	1440	4	20	1024	944	4096	576	–	–
BBCsport	544	2	5	3183	3203	-	-	-	-
ORL	400	4	40	512	59	864	254	–	–
MSRCV1	210	6	7	1302	48	512	100	256	210
EyaleB10	640	3	10	1024	1239	256	–	–	–
YaleB	650	3	10	2500	3304	6750	–	–	–
Handwritten	2000	6	10	216	76	64	6	240	47

对比算法

SSR[60]：是一种单视图聚类算法，实验中将多个视图连接成一个新矩阵作为SSR的输入。

SEANC (Spectral Embedded Adaptive Neighbors Clustering)[61]：在自适应图的基础上，SEANC对拉普拉斯矩阵施加秩约束。

RGC (Robust graph learning from noisy data)[62]：RGC应用RPCA对数据矩阵进行去噪，然后使用去噪后的数据学习相似度矩阵。

FCMSC[37]：FCMSC使用L21范数和核范数约束目标函数，输入数据也是通过特征连接获得的。

LMSC[34]：与连接输入矩阵不同，它使用多个视图学习共享表示矩阵。

CBF-MSC (Constrained bilinear factorization multi-view subspace clustering)[63]：CBF-MSC通过重构每个视图获得多个相似度矩阵，并寻找共享的相似度矩阵。

MCLES (Relaxed multi-view clustering in latent embedding space)[64]：MCLES同时学习多视图数据的潜在嵌入空间和可靠亲和矩阵，并通过K-means获得聚类标签。

NESC (Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding)[65]：非负嵌入NESC直接学习一致的聚类结果。

GFSC (Multi-graph fusion for multi-view spectral clustering)[66]：GFSC通过多个非负图学习共识图，并自适应调整每个视图的权重。

SLMVGC (Sample-level multi-view graph clustering)[67]：SLMVGC通过学习数据的拓扑结构探索隐含的数据流形。同时，SLMVGC探索多个视图的交集以捕捉视图间的一致性信息。

5.3.2 对比实验结果与分析

本节将MVC/FOMF与上述算法在7个数据集上进行比较，所有结果（NMI、ACC和Purity）如下列表所示。

从下列表中可以看出，与现有的一些算法相比，MVC/FOMF在这些数据集上取得了非常令人鼓舞的性能，特别是Coil20、YaleB、EyaleB10和MSRCV1。具体来说，MVC/FOMF在Coil20数据集上完美划分了1440个样本，MVC/FOMF的ACC指数平均提高了18.14%。对于YaleB和yalb10，这两个数据集是在不同光照或遮光条件下获得的，这使得许多现有算法的聚类效果不理想，但MVC/FOMF可以很好地识别这些不同遮光的图片，MVC/FOMF在EyeleB10和YaleB上的ACC分别比其他现有算法高37.24%和41.55%。对于MSRCV1，MVC/FOMF相对于其他现有算法在ACC指标上平均提高了13%以上。对于BBCsport和ORL，MVC/FOMF在这两个数据集上的聚类性能可能没有太大的提高，但是MVC/FOMF在BBCsport和ORL上的ACC平均分别超过其他现有算法的5.91%和5.05%。总的来说，MVC/FOMF在一些数据集上取得了很好的性能。

表5-2 MSRCV1数据集上的聚类结果

Table 5-2 Clustering results on the MSRCV1 dataset

方法	NMI	ACC	Purity
SSR	0.77 ± 0.01	0.87 ± 0.00	0.87 ± 0.00
SEANC	0.68 ± 0.00	0.74 ± 0.00	0.74 ± 0.00
RGC	0.87 ± 0.00	0.92 ± 0.00	0.92 ± 0.00
LMSC	0.75 ± 0.03	0.85 ± 0.05	0.85 ± 0.04
FCMSC	0.72 ± 0.01	0.82 ± 0.01	0.82 ± 0.00
CBF-MSC	0.69 ± 0.00	0.82 ± 0.00	0.67 ± 0.00
MCLES	0.79 ± 0.01	0.88 ± 0.01	0.88 ± 0.01
NESC	0.75 ± 0.00	0.77 ± 0.00	0.82 ± 0.00
GFSC	0.92 ± 0.01	0.96 ± 0.00	0.96 ± 0.00
SLMVGC	0.78 ± 0.00	0.87 ± 0.00	0.87 ± 0.00
MVC/FOMF	**0.96 ± 0.00**	**0.98 ± 0.00**	**0.98 ± 0.00**

表 5-3 Coil20 数据集上的聚类结果

Table 5-3 Clustering results on the Coil20 dataset

方法	NMI	ACC	Purity
SSR	0.99 ± 0.00	0.98 ± 0.00	0.98 ± 0.00
SEANC	0.93 ± 0.00	0.86 ± 0.00	0.88 ± 0.00

RGC	0.95 ± 0.00	0.85 ± 0.00	0.89 ± 0.00
LMSC	0.84 ± 0.02	0.78 ± 0.02	0.78 ± 0.02
FCMSC	0.87 ± 0.01	0.81 ± 0.02	0.81 ± 0.02
CBF-MSC	0.86 ± 0.00	0.81 ± 0.00	0.76 ± 0.00
MCLES	0.74 ± 0.02	0.71 ± 0.03	0.71 ± 0.03
NESC	0.91 ± 0.00	0.78 ± 0.00	0.84 ± 0.00
GFSC	0.85 ± 0.01	0.80 ± 0.01	0.80 ± 0.00
SLMVGC	0.92 ± 0.01	0.85 ± 0.01	0.87 ± 0.01
MVC/FOMF	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**

表 5-4 EyaleB数据集上的聚类结果

Table 5-4 Clustering results on EyaleB dataset

方法	NMI	ACC	Purity
SSR	0.67 ± 0.00	0.61 ± 0.00	0.6290 ± 0.0029
SEANC	0.54 ± 0.00	0.55 ± 0.00	0.5453 ± 0.0000
RGC	0.54 ± 0.00	0.52 ± 0.01	0.5196 ± 0.0051
LMSC	0.62 ± 0.03	0.64 ± 0.03	0.6544 ± 0.0285
FCMSC	0.44 ± 0.01	0.48 ± 0.02	0.4666 ± 0.0158
CBF-MSC	0.51 ± 0.00	0.55 ± 0.00	0.3588 ± 0.0034
MCLES	0.48 ± 0.01	0.48 ± 0.00	0.4822 ± 0.0012
NESC	0.61 ± 0.00	0.63 ± 0.00	0.6297 ± 0.0000
GFSC	0.71 ± 0.00	0.70 ± 0.00	0.7063 ± 0.0021
SLMVGC	0.77 ± 0.01	0.78 ± 0.00	0.7819 ± 0.0039
MVC/FOMF	**0.98 ± 0.00**	**0.99 ± 0.00**	**0.9891 ± 0.0000**

表 5-5 YaleB数据集上的聚类结果

Table 5-5 Clustering results on the YaleB dataset

Method	NMI	ACC	Purity
SSR	0.54 ± 0.01	0.64 ± 0.02	0.64 ± 0.01
SEANC	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
RGC	0.63 ± 0.01	0.62 ± 0.00	0.62 ± 0.00
LMSC	0.46 ± 0.04	0.49 ± 0.03	0.46 ± 0.03
FCMSC	0.63 ± 0.00	0.79 ± 0.00	0.79 ± 0.00
CBF-MSC	0.55 ± 0.00	0.55 ± 0.00	0.35 ± 0.00
MCLES	0.51 ± 0.00	0.51 ± 0.00	0.52 ± 0.00
NESC	0.53 ± 0.00	0.52 ± 0.00	0.54± 0.00

GFSC	0.47 ± 0.00	0.47 ± 0.00	0.47 ± 0.00
SLMVGC	0.47 ± 0.00	0.46 ± 0.00	0.47 ± 0.00
MVC/FOMF	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**

表 5-6 BBCsport数据集上的聚类结果

Table 5-6 Clustering results on the BBCsport dataset

方法	NMI	ACC	Purity
SSR	0.91 ± 0.00	0.97 ± 0.00	0.97 ± 0.00
SEANC	0.84 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
RGC	0.92 ± 0.00	0.98 ± 0.00	0.98 ± 0.00
LMSC	0.82 ± 0.03	0.93 ± 0.01	0.93 ± 0.01
FCMSC	0.88 ± 0.00	0.96 ± 0.00	0.96 ± 0.00
CBF-MSC	0.89 ± 0.00	0.97 ± 0.00	0.94 ± 0.00
MCLES	0.81 ± 0.01	0.88 ± 0.00	0.88 ± 0.00
NESC	0.67 ± 0.00	0.85 ± 0.00	0.99 ± 0.00
GFSC	0.83 ± 0.00	0.94 ± 0.00	0.94 ± 0.00
SLMVGC	0.63 ± 0.04	0.81 ± 0.10	0.83 ± 0.05
MVC/FOMF	**0.97 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**

表 5- 7 ORL数据集上的聚类结果

Table 5-7 Clustering results on ORL dataset

方法	NMI	ACC	Purity
SSR	0.94 ± 0.01	0.86 ± 0.01	0.89 ± 0.01
SEANC	0.96 ± 0.00	0.84 ± 0.00	0.88 ± 0.00
RGC	0.95 ± 0.00	0.87 ± 0.01	0.90 ± 0.01
LMSC	0.94 ± 0.01	0.86 ± 0.02	0.88 ± 0.01
FCMSC	0.95 ± 0.01	0.88 ± 0.02	0.90 ± 0.01
CBF-MSC	0.94 ± 0.01	0.87 ± 0.02	0.80 ± 0.03
MCLES	0.96 ± 0.00	0.88 ± 0.01	0.90 ± 0.01
NESC	0.92 ± 0.00	0.83 ± 0.00	0.87 ± 0.00
GFSC	0.95 ± 0.01	0.87 ± 0.02	0.89 ± 0.01

SLMVGC	0.92 ± 0.00	0.84 ± 0.00	0.86 ± 0.00
MVC/FOMF	**0.97 ± 0.00**	**0.91 ± 0.00**	**0.94 ± 0.00**

表 5-8 Handwritten数据集上的聚类结果

Table 5-8 Clustering results on the Handwritten dataset

方法	NMI	ACC	Purity
SSR	0.93 ± 0.00	0.97 ± 0.00	0.97 ± 0.00
SEANC	0.72 ± 0.00	0.65 ± 0.00	0.69 ± 0.00
RGC	0.80 ± 0.00	0.77 ± 0.00	0.80 ± 0.00
LMSC	0.74 ± 0.00	0.68 ± 0.00	0.62 ± 0.00
FCMSC	0.74 ± 0.00	0.85 ± 0.00	0.87 ± 0.00
CBF-MSC	0.78 ± 0.00	0.88 ± 0.00	0.78 ± 0.00
MCLES	0.67 ± 0.00	0.68 ± 0.00	0.68 ± 0.00
NESC	0.87 ± 0.00	0.86 ± 0.00	0.86 ± 0.00
GFSC	0.83 ± 0.03	0.86 ± 0.06	0.90 ± 0.02
SLMVGC	0.90 ± 0.00	0.89 ± 0.00	0.89 ± 0.00
MVC/FOMF	**0.95 ± 0.00**	**0.98 ± 0.00**	**0.98 ± 0.00**

5.3.3 运行时间与复杂度比较

表 5-9 不同算法的运行时间

Table 5-9 Running time of different algorithms

方法	MSRCV1	YaleB	EyaleB10	ORL	BBCsport	Coil20	Handwritten
SSR	0.2261	3.9366	1.5208	0.6856	1.7080	14.3253	3.5963
SEANC	0.0817	1.3830	0.6210	0.2249	16.6466	5.7280	0.8338
RGC	2.3967	50.1253	1.2490	4.5021	21.0454	110.8623	55.5375
LMSC	3.4390	38.7992	20.1493	7.3242	20.3496	139.2890	263.6012
FCMSC	5.5091	41.5227	26.3362	7.2335	23.0589	400.7559	128.1421
CBF-MSC	1.7966	71.4850	19.3579	7.3783	27.6869	378.8421	12.8152
MCLES	4.5153	297.5465	706.6104	49.0748	99.0254	–	–
NESC	1.7966	1.4305	1.2490	1.2921	25.7467	12.6139	4.6277
GFSC	2.0590	10.2776	7.3266	4.2223	5.0301	49.4634	35.1316
SLMVGC	0.2195	0.8345	0.8117	0.5529	0.7082	3.4632	43.3943

MVC/FOMF	**0.7298**	**5.4979**	**5.6554**	**2.1686**	**3.9247**	**45.6228**	**52.4755**

本节记录了上述所有算法的运行时间，如表5-9所示，记录了每个算法在7个数据集上运行的平均时间。可以看出MVC/FOMF在保持高性能的同时具有较低的时间复杂度。具体来说，SSR、SLMVGC、SEANC和NESC等算法比MVC/FOMF花费的时间更少，但MVC/FOMF的性能要好得多。其他算法如RGC、LMSC、FCMSC、CBF-MSC、MCLES和GFSC不仅耗时，而且聚类性能也不如MVC/FOMF。（“-”表示内存不足或运行时间过长，未获得聚类结果）

5.4 本章小结

本章提出了一种基于灵活且最优多的图融合方法的新型多视图聚类算法（MVC/FOMF）。MVC/FOMF旨在生成一个共识图，其列和可以约束为标量s，从而使其更加灵活。此外，共识图的拉普拉斯矩阵施加秩约束，以学习最优的聚类结构。与最先进的算法相比，MVC/FOMF在某些数据集上表现出非常令人鼓舞的性能。

第6章 总结与展望

6.1 工作总结

随着大数据时代的到来，数据的维度和复杂性显著增加，传统的聚类方法在处理高维数据时面临巨大挑战。子空间聚类作为一种有效的无监督学习技术，能够从高维数据中提取低维结构信息，近年来受到了广泛关注。然而，现有的子空间聚类算法在处理复杂数据结构时仍存在诸多局限性，如难以处理非线性相关的数据、计算效率较低等问题。针对这些问题，本文提出了三种创新的子空间聚类算法，分别从自适应核字典、张量低秩表示和多视图图融合的角度进行了深入研究，取得了显著的成果。

首先，在第三章中提出了一种基于自适应核字典的低秩表示子空间聚类方法（AKDLRR）。该方法通过引入核技巧，将原始数据映射到新的特征空间，从而将非线性数据的聚类问题转化为线性特征空间中的子空间聚类问题。同时，AKDLRR利用投影矩阵从特征空间中学习一种能够自适应更新的字典，进一步提升了聚类性能。实验结果表明，AKDLRR不仅能够取得更好的聚类性能，并且计算速度较快，尤其是在处理非线性数据时表现出色。

其次，在第四章中提出了一种具有闭式解的高效张量低秩表示算法（ETLRR/CFS）。现有的基于张量的子空间聚类算法由于直接处理张量数据，具有较高的计算复杂度。为了解决这一问题，ETLRR/CFS将TNN和Frobenius范数整合到一个统一的框架中，并推导出了一种高效的闭式解。实验结果表明，ETLRR/CFS不仅比现有的基于张量的子空间聚类算法快得多，而且能够获得相似的聚类性能，尤其是在处理大规模数据时表现出色。

最后，在第五章中提出了一种基于灵活图融合的多视图子空间聚类算法（MVC/FOMF）。现有的基于图的多视图子空间聚类算法存在图融合不够灵活以及聚类结构不被考虑的问题。为了解决这些问题，MVC/FOMF首先通过自表示方法获得每个视图的相似性图，并将这些图融合为一个列和为S（$0 < S \leq 1$）的共识图。通过调整S，MVC/FOMF能够灵活地寻找最佳的聚类性能。同时，MVC/FOMF对共识图的拉普拉斯矩阵施加秩约束，以学习最佳的聚类结构。实

验表明，与最先进的算法相比，MVC/FOMF在一些数据集上表现出非常优异的性能，尤其是在多视图数据融合方面具有显著优势。

总的来说，本文提出的三种算法分别在非线性数据处理、张量低秩表示和多视图图融合方面取得了重要进展，显著提升了子空间聚类的性能和效率。

6.2 未来展望

尽管本文提出的三种子空间聚类算法在多个方面取得了显著的成果，但在实际应用中仍存在一些局限性和挑战。未来的研究工作可以从以下几个方面展开：

（1）深度学习与子空间聚类的结合

近年来，深度学习技术在数据表示和特征提取方面取得了显著进展。未来的研究可以探索如何将深度学习技术与子空间聚类相结合，利用深度神经网络学习数据的非线性表示，进一步提升子空间聚类的性能。例如，可以尝试使用自编码器或生成对抗网络来学习数据的低维表示，并结合子空间聚类算法进行研究。

（2）缺失数据的处理

在实际应用中，多视图数据往往存在缺失或不完整的情况。本文的算法在处理完整数据时表现良好，但在处理缺失数据时仍有改进空间。未来的研究可以探索如何在不完整的多视图数据中进行有效的子空间聚类，设计能够自动补全缺失数据的算法，并结合子空间聚类技术进行联合优化。

（3）算法效率的进一步提升

虽然本文第三章和第四章提出的算法在计算效率上已经有所改进，但在处理超大规模数据集时，仍然面临计算复杂度较高的问题。未来的研究可以进一步优化算法的计算过程，探索更高效的优化方法，以降低算法的时间复杂度和空间复杂度，使其能够更好地应用于大规模数据处理场景。

综上所述，未来的研究工作将继续围绕算法效率、深度学习结合和缺失数据处理等方面展开，以期在子空间聚类领域取得更加深入和广泛的研究成果。

参考文献

[1] 朱轶凡, 罗程阳, 马瑞遥, et al. 基于密度的多度量空间数据聚类算法 [J]. 软件学报, 2025, 36(2): 851-73.

[2] WEI Z G, CHEN X, ZHANG X D, et al. Comparison of Methods for Biological Sequence Clustering [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2023, 20(5): 2874-88.

[3] SAKINA N, ARUN A P, P R, et al. Optimizing Customer Segmentation: A Comparative Analysis of Clustering Algorithms Using Evaluation Metrics; proceedings of the 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), F 7-9 Nov. 2024, 2024 [C].

[4] ELHAMIFAR E, VIDAL R J I T O P A, INTELLIGENCE M. Sparse subspace clustering: Algorithm, theory, and applications [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(11): 2765-81.

[5] VIDAL E E R. Sparse subspace clustering; proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol, F, 2009 [C].

[6] LIU G, LIN Z, YU Y. Robust subspace segmentation by low-rank representation; proceedings of the Proceedings of the 27th international conference on machine learning (ICML-10), F, 2010 [C].

[7] YU X, JIANG Y, CHAO G, et al. Deep Contrastive Multi-View Subspace Clustering With Representation and Cluster Interactive Learning [J]. IEEE Transactions on Knowledge and Data Engineering, 2025, 37(1): 188-99.

[8] ZHAO L, MA Y, CHEN S, et al. Deep Double Self-Expressive Subspace Clustering; proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), F 4-10 June 2023, 2023 [C].

[9] CANDèS E J, LI X, MA Y, et al. Robust principal component analysis? [J]. Journal of the ACM, 2011, 58(3): 1-37.

[10] BRBIC M, KOPRIVA I. L0 -Motivated Low-Rank Sparse Subspace Clustering [J]. IEEE Trans Cybern, 2020, 50(4): 1711-25.
[11] DU H, MA L, LI G, et al. Low-rank graph preserving discriminative dictionary learning for image recognition [J]. Knowledge-Based Systems, 2020, 187.
[12] LIU G, YAN S. Latent Low-Rank Representation for subspace segmentation and feature extraction; proceedings of the 2011 International Conference on Computer Vision, F 6-13 Nov. 2011, 2011 [C].
[13] HE W, CHEN J X, ZHANG W J S C. Low-rank representation with graph regularization for subspace clustering [J]. Soft computing, 2017, 21: 1569-81.
[14] BERTSEKAS D P. Constrained optimization and Lagrange multiplier methods [M]. Academic press, 2014.
[15] LU C, FENG J, CHEN Y, et al. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization; proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, F, 2016 [C].
[16] YANG J, LUO L, QIAN J, et al. Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes [J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(1): 156-71.
[17] LU Y, LAI Z, LI X, et al. Low-rank 2-D neighborhood preserving projection for enhanced robust image representation [J]. IEEE transactions on cybernetics, 2018, 49(5): 1859-72.
[18] KILMER M E, MARTIN C D J L A, APPLICATIONS I. Factorization strategies for third-order tensors [J]. Linear Algebra and its Applications, 2011, 435(3): 641-58.
[19] KILMER M E, BRAMAN K, HAO N, et al. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging [J]. SIAM Journal on Matrix Analysis and Applications, 2013, 34(1): 148-72.

[20] KERNFELD E, KILMER M, AERON S J L A, et al. Tensor-tensor products with invertible linear transforms [J]. Linear Algebra and its Applications, 2015, 485: 545-70.
[21] ZHOU P, LU C, FENG J, et al. Tensor low-rank representation for data recovery and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 43(5): 1718-32.
[22] DU S, LIU B, SHAN G, et al. Enhanced tensor low-rank representation for clustering and denoising [J]. Knowledge-Based Systems, 2022, 243: 108468.
[23] DU S, SHI Y, SHAN G, et al. Tensor low-rank sparse representation for tensor subspace learning [J]. Knowledge-Based Systems, 2021, 440: 351-64.
[24] CAI B, LU G-F. Tensor subspace clustering using consensus tensor low-rank representation [J]. Information Sciences, 2022, 609: 46-59.
[25] ZHANG T, LIU X, GONG L, et al. Late Fusion Multiple Kernel Clustering With Local Kernel Alignment Maximization [J]. IEEE Transactions on Multimedia, 2023, 25: 993-1007.
[26] 邴睿, 袁冠, 孟凡荣, et al. 多视图对比增强的异质图结构学习方法 [J]. 软件学报, 2023, 34(10): 4477-500.
[27] 赵兴旺, 王淑君, 刘晓琳, et al. 基于二部图的联合谱嵌入多视图聚类算法 [J]. 软件学报, 2024, 35(9): 4408-24.
[28] ZHANG P, LIU X, XIONG J, et al. Consensus One-step Multi-view Subspace Clustering (Extended abstract); proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE), F 3-7 April 2023, 2023 [C].
[29] TAO X, JIE L, WENSHEN Q, et al. Multi-View Subspace Clustering Via Simultaneously Cycle Projection and Similarity Learning; proceedings of the 2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), F 14-16 Dec. 2024, 2024 [C].
[30] 纪霞, 施明远, 周芃, et al. 自适应相似图联合优化的多视图聚类 [J]. 计算机学报, 2024, 47(2).

[31] 曹容玮, 祝继华, 郝问裕, et al. 双加权多视角子空间聚类算法 [J]. 软件学报, 2022, 33(2): 585-97.
[32] GAO H, NIE F, LI X, et al. Multi-view Subspace Clustering; proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), F 7-13 Dec. 2015, 2015 [C].
[33] BRBIĆ M, KOPRIVA I. Multi-view low-rank sparse subspace clustering [J]. Pattern Recognition, 2018, 73: 247-58.
[34] ZHANG C, HU Q, FU H, et al. Latent Multi-view Subspace Clustering; proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), F 21-26 July 2017, 2017 [C].
[35] ZHOU T, ZHANG C, PENG X, et al. Dual Shared-Specific Multiview Subspace Clustering [J]. IEEE Transactions on Cybernetics, 2020, 50(8): 3517-30.
[36] TANG C, ZHU X, LIU X, et al. Learning a Joint Affinity Graph for Multiview Subspace Clustering [J]. IEEE Transactions on Multimedia, 2019, 21(7): 1724-36.
[37] ZHENG Q, ZHU J, LI Z, et al. Feature concatenation multi-view subspace clustering [J]. Neurocomputing, 2020, 379: 89-102.
[38] ZHAO J-B, LU G-F. Clean and robust affinity matrix learning for multi-view clustering [J]. Applied Intelligence, 2022, 52(14): 15899-915.
[39] YAO L, LU G-F. Double structure scaled simplex representation for multi-view subspace clustering [J]. Neural Networks, 2022, 151: 168-77.
[40] LIU Y, CHEN J, LU Y, et al. Adaptively Topological Tensor Network for Multi-View Subspace Clustering [J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(11): 5562-75.
[41] HAO W, PANG S, YANG B, et al. Tensor-based multi-view clustering with consistency exploration and diversity regularization [J]. Knowledge-Based Systems, 2022, 252.
[42] CHEN Y, XIAO X, PENG C, et al. Low-Rank Tensor Graph Learning for Multi-View Subspace Clustering [J]. IEEE

Transactions on Circuits and Systems for Video Technology, 2022, 32(1): 92-104.
[43] YU Y, DU S, ZHANG K, et al. Multi-View Subspace Clustering Based on Affinity Matrix and Low-rank Tensor Decomposition; proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), F 15-17 April 2022, 2022 [C].
[44] LIU J, TENG S, ZHANG W, et al. Incomplete Multi-View Subspace Clustering with Low-Rank Tensor; proceedings of the ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), F 6-11 June 2021, 2021 [C].
[45] WANG S, CHEN Y, CE Y, et al. Low-Rank And Sparse Tensor Representation For Multi-View Subspace Clustering; proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), F 19-22 Sept. 2021, 2021 [C].
[46] BELKIN M, NIYOGI P, SINDHWANI V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples [J]. Journal of Machine Learning Research, 2006, 2399-2434.
[47] LIU X, DOU Y, YIN J, et al. Multiple kernel k-means clustering with matrix-induced regularization [Z]. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, Arizona; AAAI Press. 2016: 1888–94
[48] CHEN Y, DU L, ZHOU P, et al. Multiple kernel clustering with local kernel reconstruction and global heat diffusion [J]. Information Fusion, 2024, 105.
[49] SU R, GUO Y, WU C, et al. Kernel correlation-dissimilarity for Multiple Kernel k-Means clustering [J]. Pattern Recognition, 2024, 150: 110307.
[50] PATEL V M, VIDAL R. Kernel sparse subspace clustering [J]. IEEE International Conference on Image Processing (ICIP), 2014: 2849-53.

[51] XIAO S, TAN M, XU D, et al. Robust Kernel Low-Rank Representation [J]. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 2016, 27(11): 2268-81.
[52] FAVARO P, VIDAL R, RAVICHANDRAN A. A closed form solution to robust subspace estimation and clustering; proceedings of the CVPR 2011, F, 2011 [C]. IEEE.
[53] WU D, CHANG W, LU J, et al. Adaptive-order proximity learning for graph-based clustering [J]. Pattern Recognition, 2022, 126.
[54] ZHOU P, DU L, LI X. Adaptive Consensus Clustering for Multiple K-Means Via Base Results Refining [J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(10): 10251-64.
[55] LU C-Y, MIN H, ZHAO Z-Q, et al. Robust and efficient subspace segmentation via least squares regression; proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12, F, 2012 [C]. Springer.
[56] HU H, LIN Z, FENG J, et al. Smooth representation clustering; proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, F, 2014 [C].
[57] GREENE D, CUNNINGHAM P. Practical solutions to the problem of diagonal dominance in kernel document clustering; proceedings of the Proceedings of the 23rd international conference on Machine learning, F, 2006 [C].
[58] MA Y, DERKSEN H, HONG W, et al. Segmentation of multivariate mixed data via lossy data coding and compression [J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(9): 1546-62.
[59] ZHANG X, ZHAO S, WANG J, et al. Purity-Preserving Kernel Tensor Low-Rank Learning for Robust Subspace Clustering [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2024, 34(3): 1900-13.

[60] XU J, YU M, SHAO L, et al. Scaled Simplex Representation for Subspace Clustering [J]. IEEE Transactions on Cybernetics, 2021, 51(3): 1493-505.
[61] WANG Q, QIN Z, NIE F, et al. Spectral Embedded Adaptive Neighbors Clustering [J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(4): 1265-71.
[62] KANG Z, PAN H, HOI S C H, et al. Robust Graph Learning From Noisy Data [J]. IEEE Transactions on Cybernetics, 2020, 50(5): 1833-43.
[63] ZHENG Q, ZHU J, TIAN Z, et al. Constrained bilinear factorization multi-view subspace clustering [J]. Knowledge-Based Systems, 2020, 194: 105514.
[64] CHEN M-S, HUANG L, WANG C-D, et al. Relaxed multi-view clustering in latent embedding space [J]. Information Fusion, 2021, 68: 8-21.
[65] HU Z, NIE F, WANG R, et al. Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding [J]. Information Fusion, 2020, 55: 251-9.
[66] KANG Z, SHI G, HUANG S, et al. Multi-graph fusion for multi-view spectral clustering [J]. Knowledge-Based Systems, 2020, 189: 105102.
[67] TAN Y, LIU Y, HUANG S, et al. Sample-level Multi-view Graph Clustering; proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), F 17-24 June 2023, 2023 [C].

攻读学位期间获得的成果

[1] YaoZu Kan, Gui-Fu Lu, Yangfan Du, An adaptive kernel dictionary based low-rank representation method for subspace clustering, Neural Networks, 178 (10) (2024) (106434) 1-8 (第一作者, SCI, 中科院1 区)

[2] YaoZu Kan, Gui-Fu Lu, Liang Yao, Bing Cai, JinBiao Zhao, Multi-view clustering using a flexible and optimal multi-graph fusion method, Engineering Applications of Artificial Intelligence, 128 (2) (2024) (107452), 1-12. (第一作者, SCI, 中科院2 区)

[3] YaoZu Kan, Gui-Fu Lu, Yangfan Du, Guangyan Ji, Efficient tensor low-rank representation with a closed form solution, Asian Conference on Pattern Recognition(ACPR2023), Kitakyushu, Japan, November 5–8, 2023, Lecture Notes in Computer Science, vol 14407, 326-339 (第一作者, EI)

[4] YaoZu Kan., Gui-Fu Lu.,Guangyan Ji,Yangfan Du, Tensor low-rank representation combined with consistency and diversity exploration. International Journal of Machine Learning and Cybernetics,15, 5173–5184 (2024). (第一作者, SCI, 中科院 3 区)

致 谢

时光飞逝如电，读书三载，转眼间已要划上句号，回顾求学生涯，感恩感谢遇见的每一个人。

首先，感谢我的导师卢桂馥老师，卢老师治学严谨，学识渊博。入学后，在卢老师的悉心指导下走上了科研道路，初写论文，词句不通，逻辑不顺，卢老师耐心指导我修改每一句话，甚至于标点符号。论文被拒时，卢老师会安慰我不要灰心，论文接收时，卢老师会真诚祝贺我。每一篇论文的背后都离不开卢老师的辛勤教导。卢老师待人诚恳，平易近人，也会经常关心学生的生活。幸遇良师，受益匪浅，谆谆教诲，受用一生。

感谢PRiSE组的王勇老师，唐肝翌老师，李钧老师，范莉莉老师与其他老师，感谢他（她）们在我攻读硕士 期间对我的关心与帮助。

感谢712 和715实验室的师兄师姐，师弟师妹和同门好友们，同窗共读情难忘，友谊如水长伴行。

感谢我的家人，一路走来无条件支持和信任我，给予我前进的勇气和动力，无论何时，你们都是我最坚强的后盾。

行文至此，已到终章，纵有不舍，终会离别。