分类号 C8/381　　密级
U D C　　编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕士学位论文

（专业学位）

论文题目 基于非负矩阵分解的
多视角函数型聚类算法研究与应用

研究生姓名：程莞莞
指导教师姓名、职称：高海燕、教授
学科、专业名称：统计学、应用统计硕士
研究方向：大数据分析

提交日期：2024 年 6 月 3 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：程莹莹 签字日期：2024年6月3日

导师签名：高海燕 签字日期：2024年6月3日

导师(校外)签名：_____ 签字日期：_____

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：程莹莹 签字日期：2024年6月3日

导师签名：高海燕 签字日期：2024年6月3日

导师(校外)签名：_____ 签字日期：_____

Research and Application of Multi-view Functional Clustering Algorithm Based on Non-Negative Matrix Factorization

Candidate: Wanwan Cheng

Supervisor: Haiyan Gao

摘 要

随着数据采集技术的进步，出现了具备无穷维和连续特征的函数型数据，由此展开了对函数型数据分析方法的探索，其中对函数型数据聚类分析方法的研究受到了广泛的关注。现有的多元函数型聚类方法大多采用先“融合”各一元函数型数据再进行聚类的策略，其难以挖掘各变量间的深层次信息，而机器学习领域中的多视角学习却有着出色的聚合性能，其聚类分析结果也更为全面。此外，非负矩阵分解由于其较强的可解释性及简单的模型求解方法，在聚类研究领域得到广泛应用。一些学者将多视角学习与非负矩阵分解结合起来，展开聚类研究。受此启发，本文在函数型数据分析的框架下，基于非负矩阵分解，将多视角学习与函数型聚类相结合，提出两种多视角函数型聚类算法，希望能够通过这两种算法有效地揭示函数型数据的内在结构和特征，为相关领域的研究带来新的启发与思考。文章的具体研究内容如下：

（1）针对包含噪声和异常值的函数型数据，构建基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法。该算法利用$l_{2,1}$范数，引入图正则化项，保持低秩数据矩阵的内在几何结构，提高算法性能。采用交替迭代方法对目标函数进行优化，给出模型的迭代更新求解算法及算法流程，证明了算法的收敛性并对其计算复杂度进行了探讨。在随机模拟数据集和 Growth 数据集上进行实验，表明该方法在提高聚类性能的同时具有鲁棒性。将其应用于对北京市空气质量监测站点的空间布局识别，结果表明该方法具有一定的现实意义。

（2）针对函数型数据高维且体量大的特点，构建鲁棒双正交多视角函数型聚类算法。采用$l_{2,1}$范数，引入图正则化，考虑数据的局部几何特征，集成多视角异构特征；同时对矩阵添加约束，利用表示矩阵和基矩阵的正交性提高算法的聚类性能。采用交替迭代方法对模型优化，给出算法流程，利用辅助函数法证明算法的收敛性。在随机模拟数据集、Growth 数据集以及 TIMIT 语音数据集上的实验表明，该方法能够有效提高聚类性能。同时，针对甘肃省行政区划气象数据的实际应用表明该方法具有良好的适用性。

关键词：函数型聚类 非负矩阵分解 多视角学习 鲁棒性 双正交

Abstract

With the progress of data collection technology, functional data with infinite-dimensional and continuous characteristics have appeared. This led to the exploration of functional data analysis methods, among which the research on functional data clustering analysis methods has received widespread attention. Most of the existing multivariate functional clustering methods adopt the strategy of "fusing" each monadic functional data before clustering, which is difficult to mine the deep information among variables. However, multi-perspective learning in the field of machine learning has excellent aggregation performance and its cluster analysis results are more comprehensive. In addition, non-negative matrix factorization is widely used in the field of clustering because of its strong interpretability and simple model solving methods. Some scholars combine multi-view learning with non-negative matrix factorization to carry out clustering research. Inspired by this, in the framework of functional data analysis, this thesis uses non-negative matrix factorization to combine multi-view learning with functional clustering, and proposes two multi-view functional clustering algorithms, they are expected to reveal the internal structure and characteristics of functional data effectively, and bring new inspiration and thinking to the research in related fields. The specific research content of this thesis as follows:

(1) A robust multi-view functional clustering algorithm based on graph regularized non-negative matrix decomposition is constructed for functional data with noise and outliers. This algorithm employed $l_{2,1}$ norm and introduced a graph Laplacian regularization terms to maintain the intrinsic geometric structure of the data set and improve the performance of the algorithm. Initially, an alternating iteration method was used to optimize the objective function, providing an iterative updating solution

algorithm and the algorithm flowchart. Subsequently, the convergence of the algorithm was proven, and its computational complexity was discussed. Experiments conducted on both randomly generated datasets and the Growth dataset demonstrated that this method improves clustering performance while exhibiting robustness. When applied to identify the spatial layout of air quality monitoring stations in Beijing, the results indicated that this method possesses certain practical significance.

(2) Aiming at the high dimensionality and large volume of functional data, a robust co-orthogonal constraint multi-view functional clustering algorithm is devised. The algorithm adopts the $l_{2,1}$ norm, the graph regularization is introduced, the local geometric characteristics of the data are considered, and the multi-view heterogeneous features are integrated. At the same time, constraints are added to the non-negative matrix, and the orthogonality of the representation matrix and the base matrix is used to improve the clustering performance of the algorithm. The alternating iterative method is used to optimize the model, the algorithm flow is given, and the auxiliary function method is used to prove the convergence of the algorithm. Experiments on the stochastic simulation dataset, the Growth dataset and the TIMIT speech dataset show that the proposed method can effectively improve the clustering performance. At the same time, the practical application of meteorological data for administrative divisions in Gansu Province shows that the method has good applicability.

Keywords: Functional clustering；Non-Negative Matrix Factorization；Multi-view learning；Robust；Co-orthogonal

目 录

1 绪论

1.1 研究背景和意义

1.1.1 研究背景

随着科学技术的发展，数据存储量和数据复杂度与日俱增，在医学、气象学、生物学、经济学等领域，出现了一种拥有无穷维、连续性特征的数据，该类数据被称为函数型数据(Functional Data，FD)[1]。如环境监测中的空气质量数据、医学中的心电图数据、金融市场的股票数据、电力系统的电力负荷数据等。由于函数型数据在时间维度上的无限维特性，这导致使用传统的统计分析工具对其进行建模时面临不小挑战，同时，鉴于函数型数据在现实生活中的广泛存在，对函数型数据分析(Functional Data Analysis，FDA)[2]方法的探索成为了一个新的研究方向。在进行函数型数据分析时通常使用泛函分析工具[3]，将数据看成一个函数，进而对函数进行研究。Ramsay(1991)[3]首次对函数型数据分析的方法(线性模型和主成分分析)进行了系统地介绍，并将方法应用于加拿大温度数据的实证研究；随后，Ramsay(1997)[2]全面地将函数型数据的基本思想、经典理论以及分析方法写于书中，为函数型数据的发展奠定了基础。

聚类分析在数据挖掘和人工智能领域被视为一项重要的研究方法。传统的聚类方法主要聚焦于有限维、等间隔取样的离散静态数据，即通过对样本点进行多次重复采样，直到样本点能够被准确分类为止，但是在实际应用中，由于函数型数据的无限维度、随机间隔及连续动态特征，其难以使用传统的聚类方法进行分析。因此为了有效地对函数型数据进行聚类，函数型聚类分析方法应运而生。在函数型聚类分析时大都需要将原始的离散数据点进行拟合，生成曲线，接着对拟合的曲线进行聚类分析。依据数据的不同形式，可以将函数型聚类算法分为两类：直接从离散的数据点出发进行聚类分析的原始数据法[4]；通过投影技术将函数映射到特定的空间中进行聚类的投影法[5]。

随着人工智能领域的快速发展，机器学习开始受到重视，成为一门独立的

学科。在机器学习中，多视角学习可以从不同的角度出发，对同一对象进行研究，充分提取各个视角的数据信息，比单视角获得的信息更全面、准确、可信[6, 7]，由此多位学者将多视角学习与聚类结合起来进行了多视角聚类研究[8]。在进行多视角聚类研究时，非负矩阵分解(Non-Negative Matrix Factorization，NMF)[9]具有良好的性能，它通过将高维矩阵分解为两个低秩矩阵，可用于处理高维数据，它不仅能够降低数据的维度，还能保持分解后数据的非负性。此外 NMF 在算法求解中不仅拥有更小的存储空间，而且得到的结果解释性更佳。因此，学者们将其与多视角聚类结合起来，在图像处理[10]、语音处理[11]、文本聚类[12]等领域展开了相关研究。

由于现有设备采集到的函数型数据大多存在噪声和异常值，且大都具有高维、数据量大的特性，而常规聚类算法在处理此类数据时存在局限性，难以充分考虑到数据的内在信息。因此，有必要探究新的函数型聚类算法，为函数型数据聚类分析的研究提供更好的理论基础。

1.1.2 研究目的及意义

本文围绕函数型数据展开研究，基于非负矩阵分解，旨在将多视角学习与函数型聚类结合起来，进行多视角函数型聚类算法研究。一方面，考虑到实际数据会存在噪声和异常值，构建鲁棒的多视角函数型聚类算法模型，从而减少噪声和异常值的影响。另一方面，考虑到数据高维且体量大的特性，在降维的基础上，加入流形学习、双正交约束，构建鲁棒双正交多视角函数型聚类算法模型，从而提升模型的聚类效果，以期解决社会生活领域中该类函数型数据的聚类问题。

1.2 国内外研究现状

针对函数型数据的相关聚类算法研究，近年来，国内外学者的研究成果相对较多，本部分对国内外研究现状及发展趋势进行文献梳理，从函数型数据聚类算法、非负矩阵分解方法、多视角聚类算法三个方面展开。

1.2.1 函数型数据聚类算法研究现状

在对函数型数据分析方法的探索中，聚类分析是一个至关重要的研究方向，可以依据函数型数据的特性将现有的函数型数据聚类算法划分为一元函数型聚类算法和多元函数型聚类算法。

(1)一元函数型聚类算法

现有的针对于一元函数型数据的聚类算法已趋于成熟，James 和 Sugar(2003)[13]提出一种针对函数型数据的聚类方法(fclust)，该方法主要用于处理稀疏数据；Rossi 等(2006)[14]提出了基于支持向量机的函数型数据聚类算法；Kayano 等(2010)[15]在对函数型数据聚类时，利用了正交高斯基函数，得到了良好的聚类效果；Yamamoto 和 Terada(2014)[16]基于子空间的聚类投影函数，提出函数型子空间聚类算法(FFKM)；王德青等(2015，2015，2016)[17-19]对基于自适应赋权的函数型聚类算法进行了深入研究；梁银双等(2017)[20]将函数型主成分得分与 K-Means 聚类算法进行有机结合，以此来深入分析京津冀地区的空气污染变化特征；黄恒君等(2019)[21]为了提高拟合效果和聚类效果，提出了基于距离的一步法(FCOF)，在实验中成功验证算法可以提高聚类精度；高海燕等(2020)[22]将非负矩阵分解应用于函数型聚类过程，提出 FNMF 算法，将其应用于北京市二氧化氮数据，获得了好的聚类结果；姚晓红等(2021)[23]为了解决带有少量标签信息的函数型数据聚类问题，引入约束非负矩阵分解的概念，进而提出非负半监督的函数型聚类方法；Zhong 等(2021)[24]采用 Box-Cox 变换对函数型数据分布特征进行转换，从而进行聚类分析；孟银凤等(2022)[25]提出了一种分裂转移式层次聚类算法，该算法不仅可以自发地发现内部层次关系从而对聚类个数进行设定，还可以有效降低曲线噪声；Wang 等(2023)[26]结合函数型数据分析和聚类回归，提出函数型聚类回归异质性学习模型，并在空气质量数据集上进行了实证研究。

(2)多元函数型聚类算法

在实际生活中，函数型数据通常是多元结构，如气象数据的指标涉及温度、湿度、风力等方面；空气质量数据通常涵盖有二氧化硫、臭氧、一氧化碳

等指标，通过对数据进行多指标分析，可以更为全面的了解气象或空气质量情况。上述数据若以观察视角的不同来看，可以称之为多视角函数型数据；以数据的形式来看，称之为多元函数型数据。目前对于此类数据的研究有：Singhal 和 Seborg(2010)[27]提出多元时间序列数据聚类算法，用于对多元数据进行聚类分析；任娟(2012)[28]提出了基于多指标面板数据的聚类方法，该方法有着较好的分类效果；Jacques 和 Preda(2014)[29]将 Funclust(2013)[30]扩展到多元函数进行聚类分析方法的研究与应用；随后 Schmutz 等(2020)[31]对其进行了拓展研究，提出了 funHDDC 算法；Ieva 等(2013)[32]将传统的 K-means 聚类方法应用于多元函数型数据，得到了更为全面的分析结果；Yamamoto 和 Hwang(2017)[33]提出了一种多元函数型聚类方法，该方法巧妙地结合子空间分割技术和函数型子空间聚类，旨在更精准地对多元函数型数据进行处理和分析；Misumi 等(2019)[34]提出多元非线性混合效应模型，用于聚类多个纵向数据，并对亚洲 17 年的台风数据进行了分析；姚晓红(2022)[35, 68]基于多视角学习，研究了若干个多元函数型聚类算法，并将其运用到实际生活中，得到了较好的实际分类效果。

综合上述多元函数型聚类算法可知，目前大多算法实质上是采用整合的手段进行分析，然而由于视角的不同，各变量所含有的信息具有互补性，同一变量下的信息具有一致性，这类算法大都没有充分挖掘数据的内部信息。

1.2.2 非负矩阵分解方法研究现状

Lee and Seung(1999)[9]在《自然》杂志上首次提出了非负矩阵分解(Nonnegative Matrix Factorization，NMF)的相关概念，该方法旨在将原始矩阵分解为两个非负矩阵[①]乘积的形式，这两个非负矩阵分别被称为基矩阵和系数矩阵，通过对基矩阵和系数矩阵增加非负约束，可以保证 NMF 分解结果的可解释性。该方法拥有易于实现、占用存储空间小等优点。目前 NMF 已成功应用于机器学习[36, 37]、模式识别[38, 39]和数据挖掘[40, 41]中。

近年来，NMF 受到了广泛的关注，而针对其效率和性能的提升诸多学者展

[①] 非负矩阵中所有的元素都是非负实数，即都是大于等于零的实数。

开了多种 NMF 变体聚类算法。Cai 等(2011)[42]基于 NMF，巧妙地融入了流形学习技术，提出了图非负矩阵分解算法(GNMF)，通过寻找数据的内在信息进而提高聚类性能；Li 等(2017)[43]提出了图正则化非负矩阵分解(GNLMF)，该方法不仅能够实施图正则化，还能够在处理过程中有效地提取原始数据的低秩结构，从而更全面地揭示数据的内在特征；Yang 等(2019)[44]基于半监督非负矩阵分解，提出了带有双正交约束的半监督正则化深度非负矩阵分解(SGDNMF)，利用矩阵的双正交性提高了聚类性能；Liang 等(2020)[45]在此基础上，提出具有双正交约束的 NMF 算法，通过基矩阵和表示矩阵的正交性提升聚类性能；Li 等(2021)[46]基于 NMF，提出了一种半监督的双图正则化 NMF 与双正交约束(SDGNMF-BO)，实现了更好的局部表示；陈献等(2021)[47]对非负矩阵引入核学习方法和正则项约束，提出有向图聚类算法，提高了聚类质量；李向利等(2022)[48]针对传统模糊 C-均值算法存在的问题，基于非负矩阵分解，提出了一种修正模糊聚类算法，有效提高了大规模计算时的聚类效果；随后，他基于集成聚类，提出了层次预处理的非负矩阵分解加权集成聚类算法，相较于其它集成聚类算法，提高了聚类性能[49]；黄路路等(2023)[50]在研究非负矩阵分解时，选择l_p范数重新定义损失函数，对系数p采取不同取值从而实现更优的聚类效果。

由此可见，非负矩阵分解在聚类领域拥有着重要地位，相关的聚类算法研究有着良好的聚类性能，因此基于非负矩阵分解展开相关聚类算法研究是一个新的研究方向，值得进一步展开相关研究。

1.2.3 多视角聚类算法

多视角学习[7]采用联合训练的方式优化所有函数，利用数据之间的共识性和互补性充分挖掘数据间的信息特征，最终提高学习效果。根据应用场合的不同分为协同训练[51]、多核学习[52]和子空间学习[53]。而由于聚类算法在机器学习中的重要性，许多学者将多视角学习与聚类结合在一起展开多视角聚类[54]研究。

多视角聚类算法是在单视角聚类算法的基础上发展而来的，它可以有效地

融合并利用多个视角的信息，准确地识别出数据间的相似性和差异性，获得比单一视角聚类更准确和稳健的聚类结果。Liu 等(2016)[55]采用矩阵诱导正则化的方法，用来减少数据的冗余，增强核的多样性，进而提高聚类性能；Tang 等(2019)[56]针对相似性矩阵如何表达数据的内在几何结构与数据间的邻域问题，提出了基于联合潜在表示和相似性学习的多视角聚类方法(LALMVC)，该算法具有良好的聚类性能；Wang 等(2020)[57]提出一种基于图的多视角聚类算法(GMC)，该算法考虑了不同视角的权重问题；林燕铭等(2022)[58]提出流行正则引导的自适应加权多视角子空间聚类，算法采用核范数和自适应学习获取公式表示，以此实现聚类；王丽娟等(2023)[59]采用二部图协同聚类，将图学习、谱聚类和特征嵌入学习整合在一起进行优化，提出基于一致性图权重自适应多视角谱聚类算法，并验证了方法的有效性。

由于基于 NMF 的聚类算法在单视角数据中有着良好的性能，许多学者将 NMF 应用于多视角聚类。Xie 等(2016)[60]将 NMF 与基于高斯混合模型的谱聚类结合在一起，提出一种多视角聚类算法，并将其应用于图像检索领域；刘正等(2016)[61]基于特征加权和非负矩阵分解，同时考虑特征权重和数据高维性，进行多视角聚类算法研究；宗林林等(2017)[62]充分利用多视角信息聚类并融合多流形，有效提高了算法的聚类效果；Mekthanavanh 等(2019)[63]基于 NMF，结合图正则化，对具有大规模不完整视角的社交网络视频数据进行在线多视角聚类；李骜等(2022)[64]面向视角非对齐数据将多视角非负矩阵分解与二部图结合起来，进一步提高模型的学习能力，获得了较优的性能；郝敬琪等(2023)[65]将均方残差思想应用于 NMF，提出的多视图聚类算法可以有效提高聚类精度；杜虹燕等(2023)[66]采用非负矩阵分解提取全局信息，引入亲和矩阵提取共识信息，提出的算法具有良好的聚类性能；Wang 等(2023)[67]采用希尔伯特-施密特独立性准则衡量视图之间的相关性，提出具有双重希尔伯特-施密特独立性准则约束的多视图非负矩阵分解聚类方法，并验证了该方法的有效性。

上述相关研究表明，将非负矩阵分解与多视角学习结合起来的聚类算法能够对数据进行有效聚类，该类算法在聚类时基于多视角学习的互补性原则和一致性原则，以及非负矩阵分解所得结果更好的直观性和可解释性，展现出优越

的效率和聚类性能。由此可见，利用非负矩阵分解和多视角学习展开相关聚类研究具有广阔的前景。

1.3 研究思路及内容安排

本文在相关文献的基础上，受基于非负矩阵分解的函数型聚类算法(FNMF,2020)[22]及基于多视角学习的多元函数型聚类算法(MFNMF,2022)[35, 68]的启发，在函数型数据分析的框架下，基于 NMF 将多视角学习与函数型聚类相结合，提出两种多视角函数型聚类算法，希望为相关领域的研究带来新的启发。

首先，由于函数型数据中存在噪声和异常值，而 NMF 对于噪声和异常值不敏感，引入鲁棒损失函数，提高模型的鲁棒性，并在模型中加入图正则化项，考虑数据的邻域信息，提高聚类性能，由此提出基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法(第 3 章)。其次，基于函数型数据高维且体量大的特性，对基矩阵和表示矩阵施加双正交约束，提出基于非负矩阵分解的鲁棒双正交多视角函数型聚类算法(第 4 章)，希望在处理高维大规模数据时可以得到更好的聚类效果。

论文的主要研究思路如图 1.1 所示。

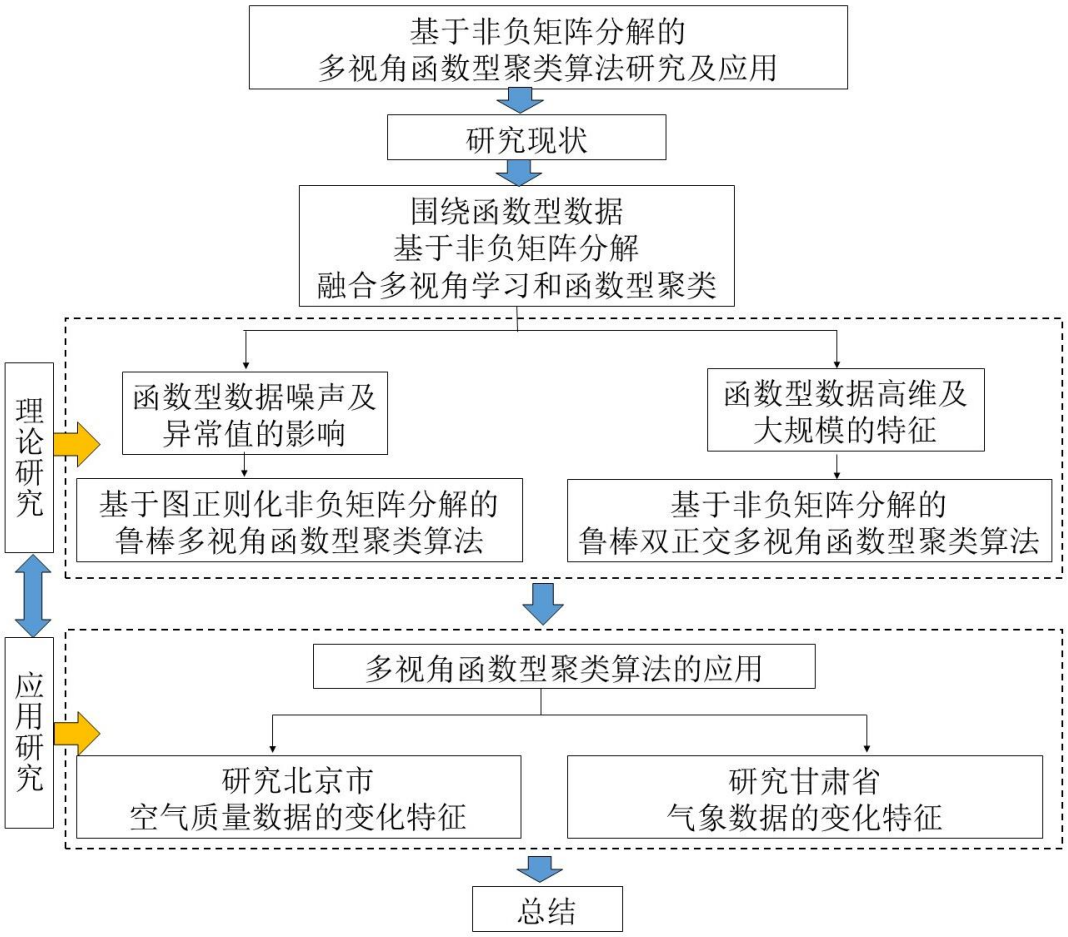


图 1.1　主要研究思路

论文的具体研究内容安排如下：

第 1 章是绪论。旨在介绍论文的研究背景、目的及意义；针对函数型数据聚类算法、非负矩阵分解以及多视角聚类算法的研究现状进行详细的文献梳理；给出论文的主要研究思路及具体内容安排；介绍论文的创新点。

第 2 章是预备知识。介绍了非负矩阵分解、基于非负矩阵分解的函数型聚类算法及多元函数型聚类算法的相关知识，同时给出后文证明算法收敛性时采用的辅助函数法以及本文评估聚类算法效果时采用的聚类评价指标。

第 3 章构建一种基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法。针对带有噪声和异常值的数据，采用$l_{2,1}$范数，引入图拉普拉斯正则化项，在保持算法鲁棒性的同时提高聚类精度。首先，给出该模型的目标函数；其次，给出算法的迭代更新求解过程、证明算法的收敛性并讨论计算复杂度；然后，通过模拟分析验证算法的性能；最后，对北京市空气质量监测站点进行聚

类应用，证明该方法的现实意义，表明其可解决实际生活中的问题。

第 4 章构建一种鲁棒双正交多视角函数型聚类算法。针对函数型数据高维且体量大的特点，利用表示矩阵和基矩阵的正交性在达到最优降维的同时提高聚类性能。首先，给出该模型的目标函数；其次，给出算法的迭代更新求解过程并证明算法的收敛性；然后，通过在数据集上的模拟分析验证了该算法的良好性能；最后，将其应用于甘肃省行政区划气象数据，证明该方法的实用性。

第 5 章是结论与展望。该章节对前面的章节内容进行深入分析，得出相应的结论，并对这些结论进行详细的阐述，同时，结合当前相关研究领域的发展趋势，对未来的研究方向进行展望，提出可能的研究方向和问题，为相关领域的研究提供一定的参考。

1.4 研究创新点

本文围绕函数型数据出发，基于非负矩阵分解，将多视角学习同函数型聚类结合起来，提出两种多视角函数型聚类算法。本文的创新点主要有：

（1）考虑到获取的数据往往带有噪声和异常值，提出基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法。该算法基于非负矩阵分解，采用$l_{2,1}$范数代替F范数，以此提高模型的鲁棒性，同时引入图拉普拉斯正则化项，保持数据内在的几何结构，提高算法性能，用于处理实际生活中带有噪声和异常值的函数型数据。

（2）考虑到函数型数据高维且体量大的特点，构造基于非负矩阵分解的鲁棒双正交多视角函数型聚类算法。由于对非负矩阵添加约束条件可以得到所期望的降维表示，因此基于非负矩阵分解，对表示矩阵和基矩阵施加双正交约束，在达到最优降维的同时获取准确的结果，用以处理高维且体量大的函数型数据。

2 预备知识

2.1 非负矩阵分解

非负矩阵分解[9]是一种矩阵分解方法，它通过基向量的线性组合来表示原始数据矩阵中的样本或特征，同时采用稀疏矩阵为每个样本或特征分配相应的标签。这种分解方式在保持数据非负性的同时，还能够有效降低数据维度，提升聚类的准确性。

给定非负数据矩阵$\boldsymbol{X}=(\boldsymbol{x}_1,\cdots,\boldsymbol{x_n})\in\mathbb{R}^{d\times n}$，NMF 旨在将原始数据矩阵$\boldsymbol{X}$转化为两个低秩非负矩阵的乘积，为了评估 NMF 算法性能，采用重构误差的平方作为损失函数来衡量，即

$$\min_{\boldsymbol{U}\geq 0,\boldsymbol{V}\geq 0}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}_v^T\|_F^2=\sum_{j=1}^{n}\|\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}_v^T\right)_j\|_2^2 \tag{2.1}$$

其中，基矩阵$\boldsymbol{U}\in\mathbb{R}^{d\times k}$，系数矩阵$\boldsymbol{V}\in\mathbb{R}^{n\times k}$。

根据乘法更新规则对式(2.1)进行求解，可得：

$$\begin{aligned}\boldsymbol{U}_{ij}&\longleftarrow\boldsymbol{U}_{ij}\frac{(\boldsymbol{X}\boldsymbol{V})_{ij}}{(\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{V})_{ij}}\\ \boldsymbol{V}_{ij}&\longleftarrow\boldsymbol{V}_{ij}\frac{(\boldsymbol{U}^T\boldsymbol{X})_{ij}}{(\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V}^T)_{ij}}\end{aligned} \tag{2.2}$$

2.2 基于非负矩阵分解的函数型聚类

函数型聚类分析方法主要分为原始数据法[4]和投影法[5]两类，由于现实观察到的数据是离散的，所以为完成对函数型数据的聚类分析，从离散数据点到曲线的数据拟合步骤不可或缺。在对函数型聚类算法进行梳理时，可以根据拟合过程与聚类过程之间是否独立，将其分为一步法和多步法。高海燕等(2020)[22]在投影法的限定下，依据 NMF 的聚类特性，将曲线拟合过程和聚类过程同时进行，提出了基于距离的函数型聚类一步法(FNMF)。

FNMF 模型的框架如下

$$\min_{\boldsymbol{U},\boldsymbol{V}}\|\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^T\|_F^2+\alpha\|\boldsymbol{D}_d\boldsymbol{U}\|_F^2 \tag{2.3}$$

其中，α代表正则化参数，$\boldsymbol{D}_d$代表d阶差分矩阵，$\boldsymbol{\Phi}$是 B-样条基矩阵，$\|\boldsymbol{D}_d\boldsymbol{U}\|_F^2$

作为惩罚项用来防止过拟合。

在 FNMF 中，对于给定的原始数据矩阵$\boldsymbol{Y}$，其可用$\boldsymbol{\Phi U V}^T$来近似表示。所以原来对矩阵$\boldsymbol{Y}$的聚类问题就转化为了对矩阵$\boldsymbol{V}$的聚类。

利用乘法更新规则求解式(2.3)有以下更新公式：

$$\begin{aligned} \boldsymbol{V}_{ij} &\longleftarrow \boldsymbol{V}_{ij}\sqrt{\frac{(\boldsymbol{Y}^T\boldsymbol{\Phi U})_{ij}^{+}+(\boldsymbol{V\Lambda}^{-})_{ij}}{(\boldsymbol{Y}^T\boldsymbol{\Phi U})_{ij}^{-}+(\boldsymbol{V\Lambda}^{+})_{ij}}} \\ \boldsymbol{U}_{ij} &\longleftarrow \boldsymbol{U}_{ij}\left(\left(\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\alpha\boldsymbol{D}_d^T\boldsymbol{D}_d\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{Y}\right)\boldsymbol{V}\right)_{ij} \end{aligned} \tag{2.4}$$

其中，$\boldsymbol{\Lambda}=\boldsymbol{V}^T\boldsymbol{Y}^T\boldsymbol{\Phi U}$。由于$\boldsymbol{\Lambda}$、$\boldsymbol{U}$无非负约束，这就导致$\boldsymbol{V\Lambda}$和$\boldsymbol{Y}^T\boldsymbol{\Phi U}$都不是非负矩阵。为了解决这一问题，采取分解的方式进行处理，令$\boldsymbol{\Lambda}^{+}=\frac{1}{2}\left(|\boldsymbol{\Lambda}|+\boldsymbol{\Lambda}\right)$表示正数绝对值，令$\boldsymbol{\Lambda}^{-}=\frac{1}{2}\left(|\boldsymbol{\Lambda}|-\boldsymbol{\Lambda}\right)$表示负数绝对值，类似地可定义$\left(\boldsymbol{Y}^T\boldsymbol{\Phi U}\right)^{+}$和$\left(\boldsymbol{Y}^T\boldsymbol{\Phi U}\right)^{-}$确保分解后的矩阵满足非负性。

2.3 基于多视角学习的多元函数型聚类

为了改进多元函数型聚类算法性能，姚晓红等(2022)[35, 68]在多视角学习的框架下，提出多元函数型聚类模型(Multivariate Functional Non-negative Matrix Factorization，MFNMF)。该模型在处理多元函数型数据时，不仅将生成过程和聚类过程进行了统一处理，还兼顾了各一元函数型数据的互补信息和共同信息，协调数据聚类结果的一致性，拥有较好的聚类效果。

MFNMF 模型的框架如下

$$\begin{aligned} &\min\sum_{v=1}^{n_v}\left(\|\boldsymbol{Y}_v-\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\|_F^2+\alpha_v\|\boldsymbol{V}_v-\boldsymbol{V}^{\star}\|_F^2+\beta_v\|\boldsymbol{U}_v\|_F^2\right) \\ &\text{s.t.}\boldsymbol{V}_v\geq 0,\boldsymbol{V}_v^T\boldsymbol{V}_v=\boldsymbol{I},\boldsymbol{V}^{\star}\geq 0 \end{aligned} \tag{2.5}$$

其中，调节参数α_v既调整各变量之间的相对权重，又调整重构误差与差异项$\|\boldsymbol{V}_v-\boldsymbol{V}^{\star}\|_F^2$之间的相对误差；为了防治过拟合现象的出现，引入$\beta_v\|\boldsymbol{U}_v\|_F^2$作为惩罚项，在这一过程中，调节参数$\beta_v$发挥着关键作用，用来调整各视角之间的相对权重，确保各视角在聚类过程中的贡献度得以平衡。

对式(2.5)进行求解，可得：

$$\begin{aligned} \boldsymbol{V}_v &\longleftarrow \boldsymbol{V}_v \frac{\left(\boldsymbol{Y}_v^T \boldsymbol{\Phi}_v \boldsymbol{U}_v\right)^+ + \alpha \boldsymbol{V}^\star + \left(\boldsymbol{V}_v \boldsymbol{\Gamma}_v^-\right)}{\left(\boldsymbol{Y}_v^T \boldsymbol{\Phi}_v \boldsymbol{U}_v\right)^- + \left(\boldsymbol{V}_v \boldsymbol{\Gamma}_v^+\right)} \\ \boldsymbol{U}_v &= \left(\left(\boldsymbol{\Phi}_v^T \boldsymbol{\Phi}_v + \beta_v \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}_v^T \boldsymbol{Y}_v\right) \boldsymbol{V}_v \\ \boldsymbol{V}^\star &= \frac{\sum_{v=1}^{n_v} \alpha_v \boldsymbol{V}_v}{\sum_{v=1}^{n_v} \alpha_v} \end{aligned} \tag{2.6}$$

其中，$\boldsymbol{\Gamma}_v = \boldsymbol{V}_v^T \boldsymbol{Y}_v^T \boldsymbol{\Phi}_v \boldsymbol{U}_v + \alpha \boldsymbol{V}_v^T \boldsymbol{V}^\star$。

由于Γ_v、$\boldsymbol{U}_v$缺乏非负约束，不能保证矩阵$\boldsymbol{V}_v\boldsymbol{\Gamma}_v$和$\boldsymbol{Y}_v^T\boldsymbol{\Phi}_v\boldsymbol{U}_v$的非负性。为了解决该问题，对矩阵进行分解，令$\boldsymbol{\Gamma}_v^+ = \frac{1}{2}\left(|\boldsymbol{\Gamma}_v| + \boldsymbol{\Gamma}_v\right)$表示$\boldsymbol{\Gamma}_v$的正部，$\boldsymbol{\Gamma}_v^- = \frac{1}{2}\left(|\boldsymbol{\Gamma}_v| - \boldsymbol{\Gamma}_v\right)$表示$\boldsymbol{\Gamma}_v$的负部，相应地可定义$\left(\boldsymbol{Y}_v^T\boldsymbol{\Phi}_v\boldsymbol{U}_v\right)^+$和$\left(\boldsymbol{Y}_v^T\boldsymbol{\Phi}_v\boldsymbol{U}_v\right)^-$确保分解后的矩阵满足非负性。

2.4 辅助函数法

辅助函数法被用于解决连续无约束优化问题[23, 35, 69]，该方法巧妙地构造了一个复合函数(即辅助函数)，用来引导目标函数跳出当前局部极小点所在的邻域，从而探索更低的目标函数值所在的邻域。需要借助辅助函数法证明算法的收敛性，具体来说，需要证明当算法接近某个局部极小点时，它能够在有限次循环迭代中，跳出该邻域寻求最优解。

接下来阐述辅助函数的定义及相关引理，随后采用辅助函数法，对提出的两个算法的收敛性进行证明，为后续应用研究奠定基础。

定义 2.1 针对下面条件

$$G(x, x^t) \geq F(x)，\ G(x, x) = F(x)$$

若$G\left(x, x^t\right)$满足，则称拉格朗日函数$F(x)$的辅助函数为$G\left(x, x^t\right)$。

引理 2.1 若$F(x)$的辅助函数是$G\left(x, x^t\right)$，则在更新规则

$$x^{t+1} = \arg\min_x G(x, x^t) \tag{2.7}$$

下，$F(x)$下是非增的。

证明 由定义 2.1 可得，接下来需要证明

$$F(x^{t+1}) \leq G(x^{t+1}, x^t) \leq G(x^t, x^t) = F(x^t)$$

显然，$G\left(x, x^t\right)$的局部极小值为 x^t 时，有$F(x^{t+1}) = F(x^t)$。若$F(x)$可导，且

在 x^t 的小邻域内连续，即$\bigtriangledown F(x^t)=0$，则通过对式(2.7)进行迭代更新求解，可得到收敛于$F(x)$局部极小值$x_{\min}=\arg\min_x F(x)$的序列

$$F(x_{\min})\leq\cdots\leq F(x^{t+1})\leq F(x^t)\leq\cdots\leq F(x^2)\leq F(x^1)\leq F(x^0)$$

即引理 2.1 得证。

2.5 聚类评价指标

本文基于相关多视角函数型聚类算法的研究[22, 23, 35, 68]，选取 3 个常用指标来评估模型的聚类性能。下面对聚类评价指标进行介绍。

（1）聚类纯度

纯度(Purity, PUR)衡量了聚类算法将数据点正确分类到类别的程度。

$$\mathrm{PUR}=\frac{1}{N}\sum_k\max_j|c_k\cap l_j| \tag{2.8}$$

其中，N是样本总数，c_k是聚类结果中第k个类别的样本集合，l_j是真实类别中第j个类别的样本集合，$|c_k\cap l_j|$表示聚类结果中第k个类别与真实类别中第j个类别的交集大小。

该评价指标计算了每个聚类中最常见的真实类别，并将其平均值作为整体纯度，数值在 0 到 1 之间，数值越高表示聚类结果余越纯净，即聚类算法更好地将数据点分配到正确的类别中。

（2）聚类精度

聚类精度(Accuracy, ACC)是衡量了聚类结果中被正确分类的样本数量。

$$\mathrm{ACC}=\frac{\sum_{i=1}^{K}\max_{j=1}^{K}|c_i\cap l_j|}{N} \tag{2.9}$$

其中，K样本总数，c_i是聚类结果中第i个簇的样本集合，l_j是真实类别中第j个类别的样本集合，$|c_i\cap l_j|$表示聚类结果中第i个簇与真实类别中第j个类别的交集大小。

该指标通过计算聚类结果中与真实类别的一对一映射来评估算法的效果，其计算方法为正确分类的样本数量占总样本数量的比例，取值范围在 0 到 1 之间，数值越高表示聚类算法将数据点分配到正确类别的准确性越高。

（3）兰德指数

兰德指数(Rand Index, RI)衡量了聚类结果中实例之间的一致性程度，即两

个样本在真实类别和聚类结果中的归属情况是否一致。

$$\mathrm{RI}=\frac{a+b}{C_2^N} \tag{2.10}$$

其中，对数据进行聚类分析后，若数据的聚类结果和真实类别都一样，则将样本对数计为a；若都在不同类别中，则将其样本对数计为b；将总的样本对数计为C_2^N。

兰德指数的取值范围在-1 到 1 之间，若值为 1，则归属情况完全一致，值为 0 意味着随机分类，而-1 表示完全不一致。

3 基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法

为了缓解函数型数据中噪声和异常值的问题，本章基于非负矩阵分解，将多视角学习和函数型聚类结合在一起，提出一种基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法(Robust Graph-regularized for Multi-View Functional Clustering Algorithm，RMNMFFC)，该算法不仅对噪声和异常值具有鲁棒性，还考虑了数据的局部几何特征，能够有效提高算法性能。

3.1 问题概述

现实生活中获取到的函数型数据，由于设备的原因大都存在噪声和异常值，而标准的 NMF 不具有鲁棒性，会导致聚类结果不准确。研究证明，重新定义损失函数可以有效提高模型的鲁棒性，如 Guan 等(2019)[70]提出了横断柯西损失，采用该损失函数提高了模型的鲁棒性；Peng 等(2020)[71]采用基于信息熵的损失函数来抑制数据中的非高斯噪声，并有效提升了算法在聚类任务中的有效性和鲁棒性；高海燕等(2023)[72]提出了鲁棒自适应对称非负矩阵分解聚类算法，其采用$l_{2,1}$范数提高算法的鲁棒性。

此外，由于 NMF 在对数据进行聚类时会忽视数据的全局结果，无法保持相邻点的数据相关性。近年来，有些研究者提出了基于流形学习的算法，该算法基于样本的相似性，通过保护数据的几何结构，进而提升 NMF 的性能。Cai 等(2011)[42]结合 NMF 和流形学习技术，提出了图非负矩阵分解算法(GNMF)；Li 等(2017)[43]提出名为图正则化非负矩阵分解(GNLMF)的方法，该方法不仅实现了数据的图正则化，还能够在处理过程中提取原始数据的低秩结构，进而全面地揭示数据的内在特征；余沁茹等(2022)[73]提出了一种自适应图正则化的 NMF 算法，该算法不仅引入了低秩约束，而且还在图构建时采用自适应方式求解相似度矩阵，具备良好的聚类性能。

基于上述启发，本章围绕函数型数据，基于非负矩阵分解，结合多视角学习，提出一种鲁棒图正则化多视角函数型聚类方法(Robust Graph-regularized for Multi-View Functional Clustering Algorithm，RMNMFFC)。该聚类算法采用$l_{2,1}$范数，对噪声和异常值具有鲁棒性；引入图正则化，充分考虑数据的局部几何特

征，集成多视角异构特征。

3.2 算法模型

假设$\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_{n_v}\}$表示所有$n_v$个视角的数据，$\boldsymbol{Y}_v \in \mathbb{R}^{d_v \times n}$表示第$v$个视角，其中$v = 1, 2, 3, \cdots, n_v$。对于每个视角的原始数据$\boldsymbol{Y}_v$是不能直接观测到的，其函数拟合为$\boldsymbol{Y}_v \approx \boldsymbol{\Phi}_v \boldsymbol{A}_v$，其中$\boldsymbol{\Phi}_v \in \mathbb{R}^{d_v \times r}$是基矩阵，$\boldsymbol{A}_v \in \mathbb{R}^{r \times n}$为完全决定曲线之间差异的系数矩阵。进一步利用 NMF 将系数矩阵$\boldsymbol{A}_v$分解为两个低秩矩阵的乘积，即$\boldsymbol{Y}_v \approx \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T$。其中$\boldsymbol{U}_v \in \mathbb{R}^{r \times K}$为第$v$个视角的基矩阵，$\boldsymbol{V}_v^T \in \mathbb{R}^{K \times n}$为第$v$个视角的聚类指示矩阵。

采用结构化稀疏性$l_{2,1}$范数，降低模型对噪声和异常值的影响，以确保鲁棒性；引入局部流形正则化，在矩阵分解时，利用流形的局部不变性，确保样本的几何结构在从高维空间$\boldsymbol{Y}_v$分解到低维空间$\boldsymbol{V}_v$中仍能继续保持。RMNMFFC 算法的目标函数可以表示为

$$\begin{aligned} &\min_{\boldsymbol{U}_v, \boldsymbol{V}_v} \sum_{v=1}^{n_v} \left\{ \frac{1}{2} \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T\|_{2,1} + \frac{\lambda}{2} \mathrm{tr}(\boldsymbol{V}_v^T \boldsymbol{L}_v \boldsymbol{V}_v) + \alpha_v \|\boldsymbol{U}_v\|_F^2 \right\} \\ &\text{s.t. } \boldsymbol{U}_v \geq 0, \ \boldsymbol{V}_v \geq 0 \end{aligned} \tag{3.1}$$

其中，λ为正则化参数，α_v为调节参数。$\boldsymbol{L}_v = \boldsymbol{D}_v - \boldsymbol{W}_v$为拉普拉斯矩阵，$\boldsymbol{W}_v = (W_{vij})$为相似矩阵，$\boldsymbol{D}_v = (D_{vij})$为度矩阵，且$D_{vjj} = \Sigma_i W_{vij}$，

$$W_{vij} = \begin{cases} 1 & \boldsymbol{y}_{vi} \in N_k(\boldsymbol{y}_{vj}) \ \text{ or } \ \boldsymbol{y}_{vj} \in N_k(\boldsymbol{y}_{vi}) \\ 0 & \text{otherwise} \end{cases}$$

其中，$N_k(\boldsymbol{y}_{vj})$表示第v个视角的第j个样本的k近邻，也就是这一组k个样本数据最接近样本$\boldsymbol{y}_{vj}$。在这组数据中，将任意两个样本数据之间的距离定义为

$$d(\boldsymbol{y}_{vi}, \boldsymbol{y}_{vj}) = \sqrt{\sum_{t=1}^{m} (y_{vti} - y_{vtj})^2}$$

正则化项$\mathcal{R} = \frac{\lambda}{2} \mathrm{tr}(\boldsymbol{V}_v^T \boldsymbol{L}_v \boldsymbol{V}_v)$可以在低维表示$\boldsymbol{v}_i$和$\boldsymbol{v}_j$中保留$\boldsymbol{y}_j$和$\boldsymbol{y}_j$有相似特征的局部信息，从而大大提高其表示能力和聚类性能。

3.3 求解算法

3.3.1 优化求解

由于$l_{2,1}$范数是非光滑的，因此可以将优化问题式(3.1)分割为几个子问题，然后分别采用乘性更新迭代方法对模型进行更新求解。为书写方便，可以记

$$\begin{aligned}\boldsymbol{H}_v &= \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\|_{2,1} \\ &= \mathrm{tr}\left[(\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T)^T\boldsymbol{G}_v(\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T)\right] \\ &= \mathrm{tr}\left[\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v - 2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T + \boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\right]\end{aligned} \tag{3.2}$$

其中，$\boldsymbol{G}_v = \mathrm{diag}\left\{g_{v11}, g_{v22}, \cdots, g_{vd_vd_v}\right\} \in \mathbb{R}^{d_v \times d_v}$是对应于第$v$个视角的对角矩阵，对角线上的第$i$项定义为：

$$g_{vii} = \frac{1}{\|\boldsymbol{e}_v^i\|}, \quad \forall \ \ i = 1, 2, \ldots, d_v \tag{3.3}$$

其中，$\boldsymbol{e}_v^i = (e_{vi1}, e_{vi2}, \cdots, e_{vin})^T$是以下矩阵$\boldsymbol{E}_v$的第$i$行形成的列向量

$$\boldsymbol{E}_v = \boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T = \left(\boldsymbol{e}_v^1, \boldsymbol{e}_v^2, \cdots, \boldsymbol{e}_v^{d_v}\right)^T \tag{3.4}$$

(1)更新$\boldsymbol{U}_v$和$\boldsymbol{V}_v$。固定$\boldsymbol{G}_v$，关于$\boldsymbol{U}_v$和$\boldsymbol{V}_v$的子问题为

$$\begin{aligned}&\min_{\boldsymbol{U}_v, \boldsymbol{V}_v} \frac{1}{2}\left\{\boldsymbol{H}_v + \lambda\mathrm{tr}(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v) + 2\alpha_v\|\boldsymbol{U}_v\|_F^2\right\} \\ &\text{s.t.} \ \ \boldsymbol{U}_v \geq 0, \ \boldsymbol{V}_v \geq 0\end{aligned} \tag{3.5}$$

其拉格朗日函数为：

$$\begin{aligned}\mathcal{L}_1 =& \frac{1}{2}\left\{\boldsymbol{H}_v + \lambda\mathrm{tr}(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v) + 2\alpha_v\|\boldsymbol{U}_v\|_F^2\right\} \\ &- \mathrm{tr}(\boldsymbol{\Lambda}_v\boldsymbol{U}_v^T) - \mathrm{tr}(\boldsymbol{\Gamma}_v\boldsymbol{V}_v^T)\end{aligned} \tag{3.6}$$

其中，$\boldsymbol{\Lambda}_v$，$\boldsymbol{\Gamma}_v$分别为约束条件$\boldsymbol{U}_v \geq 0$和$\boldsymbol{V}_v \geq 0$的拉格朗日乘子矩阵。注意到

$$\frac{\partial \boldsymbol{H}_v}{\partial \boldsymbol{U}_v} = -2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v + 2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v \tag{3.7}$$

$$\frac{\partial \boldsymbol{H}_v}{\partial \boldsymbol{V}_v} = -2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v + 2\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v \tag{3.8}$$

$\mathcal{L}_1$对$\boldsymbol{U}_v$求偏导，令$\frac{\partial \mathcal{L}_1}{\partial \boldsymbol{U}_v} = 0$，即

$$\frac{\partial \mathcal{L}_1}{\partial \boldsymbol{U}_v} = -\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v + \boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v + \alpha_v\boldsymbol{U}_v - \boldsymbol{\Lambda}_v = 0$$

可解得

$$\boldsymbol{\Lambda}_v = \boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v + \alpha_v\boldsymbol{U}_v - \boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v \tag{3.9}$$

因为$\boldsymbol{U}_v \geq 0$，即满足非负约束，因此 KKT 条件$\boldsymbol{\Lambda}_v \odot \boldsymbol{U}_v = 0$成立，即有

$$\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha_v\boldsymbol{U}_v-\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v\right)_{ij}\odot\boldsymbol{U}_{vij}=0$$

其中，$\odot$为 Hadamard 积。因此，$\boldsymbol{U}_v$的更新规则为

$$\boldsymbol{U}_{vij}\longleftarrow\boldsymbol{U}_{vij}\frac{(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v)_{ij}}{(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha_v\boldsymbol{U}_v)_{ij}} \tag{3.10}$$

$\mathcal{L}_1$关于$\boldsymbol{V}_v$求偏导，并令$\frac{\partial\mathcal{L}_1}{\partial\boldsymbol{V}_v}=0$，得到

$$\frac{\partial\mathcal{L}_1}{\partial\boldsymbol{V}_v}=-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{L}\boldsymbol{V}_v-\boldsymbol{\Gamma}_v=0$$

有

$$\boldsymbol{\Gamma}_v=\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{L}\boldsymbol{V}_v-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v \tag{3.11}$$

非负约束$\boldsymbol{V}_v\geq0$使得 KKT 条件$\boldsymbol{\Gamma}_v\odot\boldsymbol{V}_v=0$成立，即满足

$$\left(\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{L}\boldsymbol{V}_v-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\right)_{ij}\odot\boldsymbol{V}_{vij}=0$$

因此，$\boldsymbol{V}_v$的更新规则为

$$\boldsymbol{V}_{vij}\longleftarrow\boldsymbol{V}_{vij}\frac{\left(\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{W}_v\boldsymbol{V}_v\right)_{ij}}{(\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v)_{ij}+\lambda\boldsymbol{D}_v\boldsymbol{V}_v} \tag{3.12}$$

（2）更新$\boldsymbol{G}_v$。固定$\boldsymbol{U}_v$和$\boldsymbol{V}_v$。利用式(3.3)和式(3.4)更新$\boldsymbol{G}_v$。

3.3.2 算法步骤

通过 3.3.1 依次交替迭代更新$\boldsymbol{U}_v$、$\boldsymbol{V}_v$和$\boldsymbol{G}_v$，可以求解非凸优化问题式(3.1)，该算法的具体步骤如下所示。

算法 3.1 RMNMFFC 算法

输入：原始数据矩阵$\boldsymbol{Y}_v$，基底矩阵$\boldsymbol{\Phi}_v$，惩罚参数λ、调节参数α_v和类别数K

过程：

1：首先构造拉普拉斯矩阵$\boldsymbol{L}_v$

2：其次初始化矩阵$\boldsymbol{U}_v^0$和$\boldsymbol{V}_v^0$，同时令$\boldsymbol{G}_v=\boldsymbol{I}_{d_n}$

3：for $t=1,2,\cdots$,最大更新迭代次数

4： for $v=1,2,\cdots,n_v$

5： 固定$\boldsymbol{V}_v^{t-1}\boldsymbol{V}_v^{t-1}$和$\boldsymbol{G}_v^{t-1}$，根据式(3.10)更新$\boldsymbol{U}_v^t$;

6： 固定$\boldsymbol{U}_v^t$和$\boldsymbol{G}_v^{t-1}$，根据式(3.12)更新$\boldsymbol{V}_v^t$;

7： 固定$\boldsymbol{U}_v^t$和$\boldsymbol{V}_v^t$，根据式(3.3)和式(3.4)更新$\boldsymbol{G}_v^t$;

8： end for

9： if 式(3.1)收敛

10： break

11： end if

12：end for

输出：$\boldsymbol{U}_v^t$、$\boldsymbol{V}_v^t$和$\boldsymbol{G}_v^t$，类别划分$C=\{C_1,C_2\cdots,C_K\}$

3.3.3 算法收敛性证明

采用第二章节的辅助函数法证明算法 3.1 的收敛性。

对于算法 3.1，有下列结论成立。

定理 3.1 (1)固定$\boldsymbol{G}_v$，则有

$$\begin{aligned}&\min_{\boldsymbol{U}_v,\boldsymbol{V}_v}\frac{1}{2}\left\{\boldsymbol{H}_v+\lambda\mathrm{tr}(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v)+2\alpha_v\|\boldsymbol{U}_v\|_F^2\right\}\\&\text{s.t. }\ \boldsymbol{U}_v\geq 0,\ \boldsymbol{V}_v\geq 0\end{aligned}$$

的最优解为$\boldsymbol{U}_v$的更新规则(式 3.10)；(2)固定$\boldsymbol{G}_v$，在$\boldsymbol{U}_v$的更新规则(式(3.10))下，目标函数(式(3.1))是非增的。

事实上，固定$\boldsymbol{G}_v$，则

$$\begin{aligned}&\min_{\boldsymbol{U}_v,\boldsymbol{V}_v}\frac{1}{2}\left\{\boldsymbol{H}_v+\lambda\mathrm{tr}(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v)+2\alpha_v\|\boldsymbol{U}_v\|_F^2\right\}\\&\text{s.t. }\ \boldsymbol{U}_v\geq 0,\ \boldsymbol{V}_v\geq 0\end{aligned}$$

是一个带约束的最优化问题。在上述算法(即式(3.10))的求解过程中可以证明定理 3.1 的结论(1)成立。

根据引理 2.1，需要构建一个合适的辅助函数来证明定理 3.1 的结论(2)，即证明目标函数(式(3.1))在特定的更新规则(式(3.10)和式(3.12))下是非增的。因此，接下来首先给出辅助函数的构造方式，进而证明定理 3.1 中的结论(2)。

命题 3.1 设定函数F的一阶导函数为F'，则

$$\begin{aligned}G\left(U_v,U_{vij}^t\right)=&F_{U_{vij}}\left(U_{vij}^t\right)+F_{U_{vij}}\left(U_{v_{ij}}^t\right)\left(U_v-U_{vij}^t\right)\\&+\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}\left(U_v-U_{vij}^t\right)^2}{U_{v_{ij}}^t}\end{aligned}\tag{3.13}$$

可以看作是函数$F_{U_{vij}}(U_v)$的辅助函数，且满足条件$G(U_v,U_v)=F_{U_{vij}}(U_v)$及$G(U_v,U_{vij}^t)\geq F_{U_{vij}}(U_v)$。其中，$F_{U_{vij}}(U_v)$指的是目标函数(式(3.1))中有关$U_{vij}$的部分。

证明 显然，可以知道$G(U_v,U_v)=F_{U_{vij}}(U_v)$。根据辅助函数的定义，还需证明$G(U_v,U_{vij}^t)\geq F_{U_{vij}}(U_v)$。因此，将式(3.13)与$F_{U_{vij}}(U_v)$在$U_v=U_{vij}^t$处的泰勒展开式

$$F_{U_{vij}}(U_v)=F_{U_{vij}}(U_{vij}^t)+F'_{U_{vij}}(U_{vij}^t)(U_v-U_{vij}^t)+\frac{1}{2}F''_{U_{vij}}(U_{vij}^t)(U_v-U_{vij}^t)^2$$

进行比较，那么就只需要证明

$$\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}+\alpha_v\boldsymbol{U}_v}{{U_v}_{ij}^t}\geq\frac{1}{2}F''_{U_{vij}}=\frac{1}{2}\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\right)_{ii}\left(\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{jj}+\frac{1}{2}\alpha_v$$

成立即可。因为

$$\begin{aligned}&\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}+(\alpha_v\boldsymbol{U}_v)_{ij}\\=&\sum_{k=1}^{K}\left(\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\right)_{ik}\left(\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{kj}+(\alpha_v\boldsymbol{U}_v)_{ij}\right)\\\geq&\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\right)_{ij}\left(\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{jj}+\alpha_v\boldsymbol{U}_{vij}\\\geq&\boldsymbol{U}_{vij}^t\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\right)_{ii}\left(\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{jj}+\alpha_v\boldsymbol{U}_{vij}^t\end{aligned}$$

所以

$$\begin{aligned}\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}}{{U_v}_{ij}^t}&\geq\boldsymbol{U}_{vij}^t\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\right)_{ii}\left(\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{jj}+\alpha_v\boldsymbol{U}_{vij}^t\\&=\frac{1}{2}F''_{U_{vij}}\end{aligned}$$

因此可得函数$F_{U_{vij}}(U_v)$的辅助函数就是$G(U_v,U_{vij}^t)$。即式(3.13)正是函数$F_{U_{vij}}(U_v)$的辅助函数。

接下来对定理 3.1 中的结论(2)进行证明，具体步骤如下：

将式(3.13)中的$G(U_v,U_{vij}^t)$代入式(2.7)，可得

$${U_v}_{ij}^{t+1}=\arg\min_{U_v}G(U_v,U_{vij}^t)=\boldsymbol{U}_{vij}\frac{(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v)_{ij}}{(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha_v\boldsymbol{U}_v)_{ij}}$$

因为$G(U_v,U_{vij}^t)$是函数$F_{U_{vij}}(U_v)$的辅助函数，所以根据引理 2.1 可得：函数$F_{U_{vij}}(U_v)$在更新规则(式(3.10))下是非增的。因此，定理 3.1 的结论(2)得证。

根据相同的方法，同理可证$F_{V_{vij}}(V_v)$在更新规则(式(3.12))下是非增的。由此，证明了算法 3.1 的收敛性。

3.3.4 计算复杂度分析

进一步讨论 RMNMFFC 算法的计算复杂度，各符号表示如下：t是迭代次数，K是类别数，n是观测数，n_v是视角数，m是特征数量。其中 B-样条基底矩阵$\boldsymbol{\Phi}$为带状矩阵，p为 B-样条基数量。

RMNMFFC 算法的计算复杂度涵盖了$\boldsymbol{U}_v$、$\boldsymbol{V}_v$和$\boldsymbol{G}_v$三个方面：

(1) 首先对$\boldsymbol{U}_v$进行计算复杂度分析。对于$\boldsymbol{U}_v\,(v=1,2,\cdots,n_v)$的更新(式(3.10))，根据矩阵乘法规则，可以得到其计算复杂度为$O(nlK^2+K)$，则更新n_v

个变量总的计算复杂度为$O(n_v(nlK^2+K))$。

(2) 接下来，进一步分析$\boldsymbol{V}_v$的计算复杂度。在$\boldsymbol{V}_v\,(v=1,2,\cdots,n_v)$的更新步骤中(式(3.12))，依据矩阵乘法的运算规则，可以推导出其计算复杂度为$O(nmpK+m^2K)$，考虑到需要更新n_v个变量，因此更新n_v个变量总的计算复杂度为$O(n_v(nmpK+m^2K))$。

(3) 最后计算$\boldsymbol{G}_v$。通过式(3.3)和式(3.4)，可以实现对于$\boldsymbol{G}_v$的更新，经分析可以得到对于$\boldsymbol{G}_v$，更新n_v个变量的计算复杂度为$O(n_v(nmpK))$。

综上所述，RMNMFFC 算法在经历t次迭代后，其整体的计算复杂度为$O(tn_v(nmpK+m^2K))$。

3.4 模拟分析

通过对随机模拟数据集和 Growth 数据集进行聚类分析，用以证明算法 3.1 的性能，实验选取纯度(Purity)、聚类精度(Accuracy)和兰德指数(RI)进行聚类效果评价，实验代码采用 R 语言，实验的计算机环境为：Intel 酷睿 i5-12500H 2.50GHz，内存 4GB，Windows10 64 位操作系统。

3.4.1 数据集

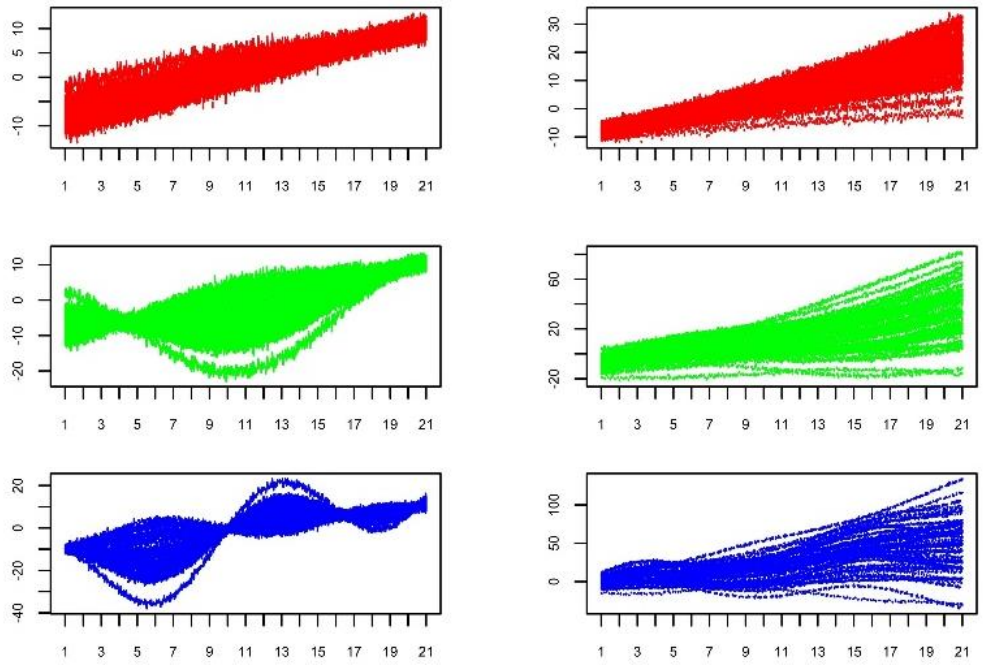
（1）随机模拟数据集

借鉴 Jacques 和 Preda(2014)[29]的方法成功模拟生成了两组包含三个不同类别的函数型数据集。在构建数据集时，分别选用三角函数和多项式函数进行线性组合，确保数据的多样性和准确性，关于模拟数据的具体图示，见图 3.1。此外，数据的生成模型及参数设置如下：

$$X_1(t)=-\frac{21}{2}+t+kU_1cos\left(k\frac{t}{10}\right)+kU_1sin\left(k+\frac{t}{10}\right)+\epsilon(t)$$

$$X_2(t)=-\frac{21}{2}+t+kU_1sin\left(k\frac{t}{10}\right)+kU_2cos\left(k+\frac{t}{10}\right)+kU_3\left(\left(\frac{t}{10}\right)^2+\frac{t}{10}+1\right)+\epsilon(t)$$

其中，U_i为随机变量，且$U_i\sim N(1,1)$，$i=1,2,3$；$\varepsilon(t)$是白噪声，服从$N(0,1)$分布；选取 1、3、5 对k赋值，表示每组数据包含 3 个类别，每个类别随机生成 50 条曲线；$t\in$[1,21]，选择 1001 个等距离散点，即取$t=${1,1.02,1.04,……,21}。

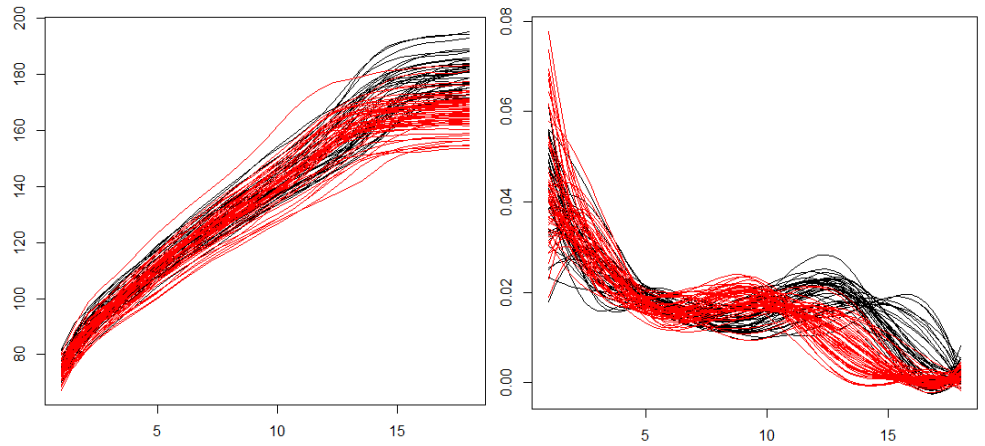


(a)$X_1(t)$　　(b)$X_2(t)$

图 3.1 随机模拟数据

（2）Growth 数据集

Growth 数据集[②]记录了 54 名女孩和 39 名男孩的身高数据，其将 1 岁到 18 岁划分为 31 个阶段。数据曲线图如图 3.2 所示。



(a) 原始曲线　　(b) 增长率曲线

图 3.2 Growth 数据

[②] Growth 数据来源于 Berkeley Growth Study data in fda： Functional Data Analysis(https://rdrr.io/cran/fda/man/growth.html)，其通过对身高增长曲线进行聚类来反映性别的差异。

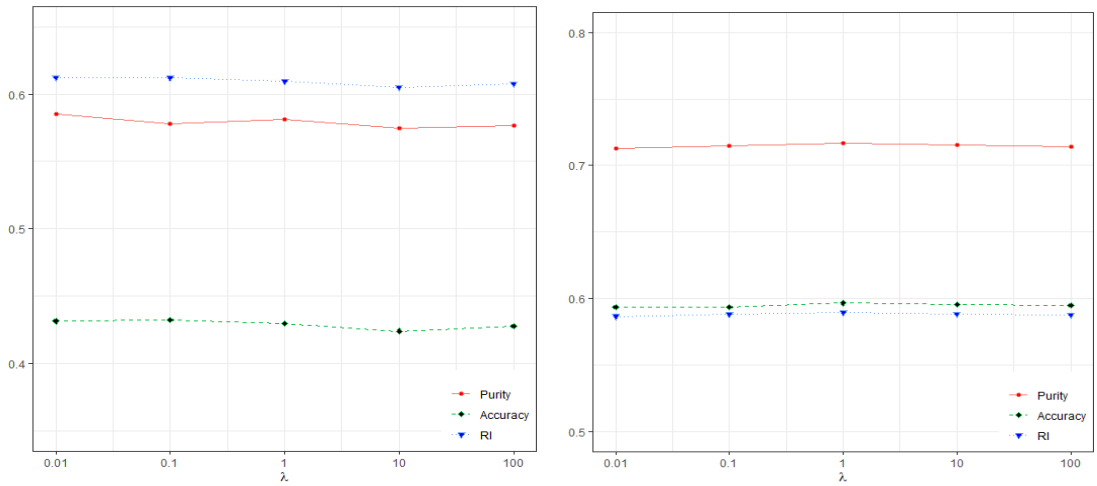
3.4.2 参数设置与对比方法

算法的相关参数设定如下：①对于图 3.1 的随机模拟数据集，设定视角数$n_v=2$，类别数$K=3$；②对于图 3.2 的 Growth 数据集，取视角数$n_v=2$，聚类类别数$K=2$；③假定在数据处理的过程中每个视角具有相同的权重，即取调节参数$\alpha_v=\frac{1}{2}$；④在拟合曲线时，采用 B-样条作为基底，通过对 B-样条基底数量的调整，用来对曲线的平滑程度进行调节；⑤迭代次数设为 200 次。

本次实验的对比算法有 FFKM(Yamamoto 等，2014)[16]、Funclust(Jacques 等，2013)[30]和 funHDDC(Julien 等，2014)[29]。

3.4.3 参数敏感性分析

考虑超参数λ对聚类性能的影响，分别取$\lambda=\{0.01,0.1,1,10,100\}$，在随机模拟数据集和 Growth 数据集上分别进行聚类，聚类的评价指标结果见图 3.3：



(a) 随机模拟数据 (b) Growth 数据

图 3.3 超参数λ与聚类性能的关系

从图 3.3 可以看出，在随机模拟数据集和 Growth 数据集上的实验结果表明：聚类结果对于超参数λ的变化不敏感，特别是在 Growth 数据上，其影响几乎可忽略不计。这充分说明了 RMNMFFC 对超参数λ具有出色的鲁棒性。

3.4.4 聚类实验结果

在随机模拟数据以及 Growth 数据上，利用上述相关参数设定，对比 FFKM、Funclust 和 funHDDC 与 RMNMFFC 的聚类性能。聚类评价结果如表 3.1 和表 3.2 所示。

需要说明 funHDDC 聚类方法通过在特定函数子空间内对各组进行建模，其包含 6 个子模型(详见 Schmutz 等(2020)[31]的描述)，该方法的结果是对 6 个子模型的聚类结果取最优。表 3.1 和表 3.2 中粗体表示比较结果为优。

表 3.1 随机模拟数据集的聚类评价结果(均值±标准差)

评价指标 / 聚类算法	聚类纯度 (Purity)	聚类精度 (Accuracy)	兰德指数 (RI)
FFKM	0.33 ± 0.0080	0.32 ± 0.0014	0.55 ± 0.0071
Funclust	0.49 ± 0.0531	0.36 ± 0.0433	0.50 ± 0.0736
funHDDC	0.58 ± 0.0507	0.42 ± 0.0435	0.61 ± 0.0471
RMNMFFC	**0.62 ± 0.0939**	**0.46 ± 0.0863**	**0.63 ± 0.0729**

表 3.2 Growth 数据集的聚类评价结果(均值±标准差)

评价指标 / 聚类算法	聚类纯度 (Purity)	聚类精度 (Accuracy)	兰德指数 (RI)
FFKM	0.66 ± 0.0023	0.53 ± 0.0015	0.56 ± 0.0032
Funclust	0.63 ± 0.0379	0.55 ± 0.0370	0.55 ± 0.0487
funHDDC	0.60 ± 0.0547	0.52 ± 0.0893	0.53 ± 0.0788
RMNMFFC	**0.72 ± 0.0105**	**0.60 ± 0.0084**	**0.59 ± 0.0087**

表 3.1 和表 3.2 的结果表明，在纯三个聚类评价指标上，RMNMFFC 算法的结果均优于 FFKM、Funclust 和 funHDDC。因此，本文提出的 RMNMFFC 算法有助于提高聚类性能。

为了能更细致地描述 RMNMFFC 算法在 Growth 数据集上良好的聚类性能，下面用箱线图来进一步描述四种聚类算法在 Growth 数据集上的评价结果。

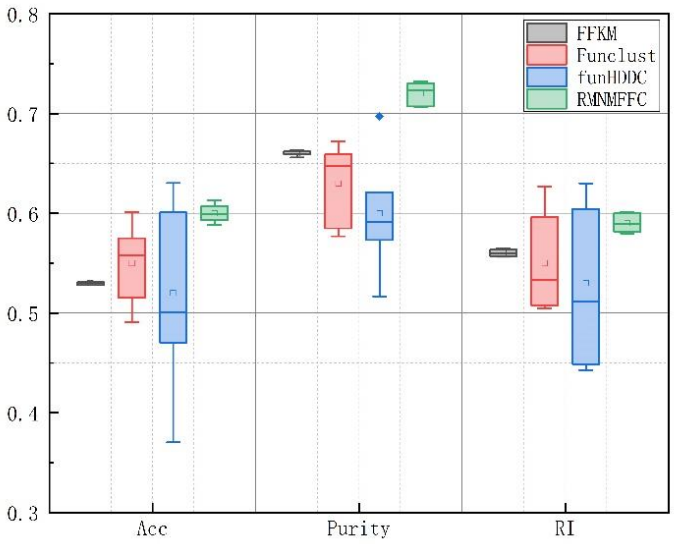


图 3.4 Growth 数据聚类评价结果图示

图 3.4 中绿色箱线代表 RMNMFFC 算法的聚类评价结果，从图中可以清晰地看出对于三种聚类评价指标，RMNMFFC 算法相较于其它三种算法拥有良好的聚类性能。

3.5 实例分析——以北京市空气质量监测站点聚类为例

将 RMNMFFC 算法应用于北京市污染物小时浓度数据，旨在通过聚类分析对进行空气质量监测站点进行划分，从而揭示站点的布局特征，进而说明该算法在实际应用中的价值。

根据北京市环境监测中心(http://www.bjmemc.com.cn/)公布的数据，北京市共有 35 个空气质量监测站点，被分为四种类型，站点的具体类别及经纬度信息如表 3.3 所示。

表 3.3 北京市空气质量监测站点经纬度信息

站点类别	编号	站点名称(经纬度坐标)	编号	站点名称(经纬度坐标)	编号	站点名称(经纬度坐标)
	1	东四 (116.42, 39.93)	9	香山(植物园) (116.21, 40.00)	17	顺义新城 (116.66, 40.13)
	2	天坛 (116.41, 39.89)	10	丰台花园 (116.28, 39.86)	18	昌平镇 (116.23, 40.22)

续表 3.3

站点类别	编号	站点名称(经纬度坐标)	编号	站点名称(经纬度坐标)	编号	站点名称(经纬度坐标)
城市环境评价点	3	西城官园 (116.34, 39.93)	11	云岗 (116.15, 39.82)	19	双峪(门头沟) (116.11, 39.94)
	4	万寿西宫 (116.35, 39.88)	12	古城 (116.18, 39.91)	20	平谷镇 (117.10, 40.14)
	5	奥体中心 (116.40, 39.98)	13	良乡(房山) (116.14, 39.74)	21	怀柔镇 (116.63, 40.33)
	6	农展馆 (116.46, 39.94)	14	黄村(大兴) (116.40, 39.72)	22	密云镇 (116.83, 40.37)
	7	海淀万柳 (116.29, 39.99)	15	亦庄 (116.51, 39.80)	23	夏都(延庆) (115.97, 40.45)
	8	北部新区 (116.17, 40.09)	16	通州北苑 (116.66 39.89)		
区域背景传输点	24	京西北(八达岭) (115.99, 40.37)	26	京东(东高村) (117.12, 40.10)	28	京南(榆垡) (116.30, 39.52)
	25	京东北 (116.91, 40.50)	27	京东南 (116.78,39.71)	29	京西南 (116.00, 39.58)
交通污染控制点	30	前门 (116.40, 39.90)	32	西直门 (116.35, 39.95)	34	东四环 (116.48, 39.94)
	31	永定门 (116.39, 39.88)	33	南三环 (116.37, 39.86)		
城市清洁对照点	35	定陵 (116.22，40.29)				

从互联网收集北京市 35 个站点 2018 年 1 月 1 日至 2018 年 12 月 31 日的 6 种大气污染物小时浓度数据，包括一氧化碳(CO)、二氧化氮(SO_2)、二氧化氮(NO_2)、细微颗粒物(PM_{10}和 $PM_{2.5}$)和臭氧(O_3)。在进行数据预处理时，首先进行异常值处理，确保数据的准确性和可靠性；随后进行缺失值插补，保证数据的完整性；最后转化为日平均数据，并进行标准化。

数据处理之后，以北京市大气污染物的类别数为视角数，对 34 个监测站点

(除“城市清洁对照点”(定陵))利用 RMNMFFC 算法进行聚类分析，采用 B-样条作为基底对曲线进行拟合；设置 200 次迭代，100 次聚类；假设各变量调节参数相等，取$\alpha_v = \frac{1}{6}(v = 1,2,\cdots\cdots, 6)$；以监测站点的类别作为标签，取聚类数$K = 3$用检验聚类效果。

表 3.4 详细展示了利用 RMNMFFC 算法对 34 个监测站点所监测到的 6 种污染物浓度数据进行聚类分析的结果。

表 3.4 RMNMFFC 算法聚类结果

类别	聚类结果
第一类	1 东四 2 天坛 3 西城官园 4 万寿西宫 5 奥体中心 6 农展馆 7 海淀万柳 8 北部新区 10 丰台花园 11 云岗 12 古城 13 良乡 14 黄村 15 亦庄 16 通州北苑 17 顺义新城 19 双峪 20 平谷镇 26 京东
第二类	9 香山 18 昌平镇 21 怀柔镇 22 密云镇 23 夏都 24 京西北 25 京东北
第三类	27 京东南 28 京南 29 京西南 30 前门 31 永定门 32 西直门 33 南三环 34 东四环

进一步采用 ArcGIS 10.2 展示聚类结果的空间布局，如图 3.5 所示。

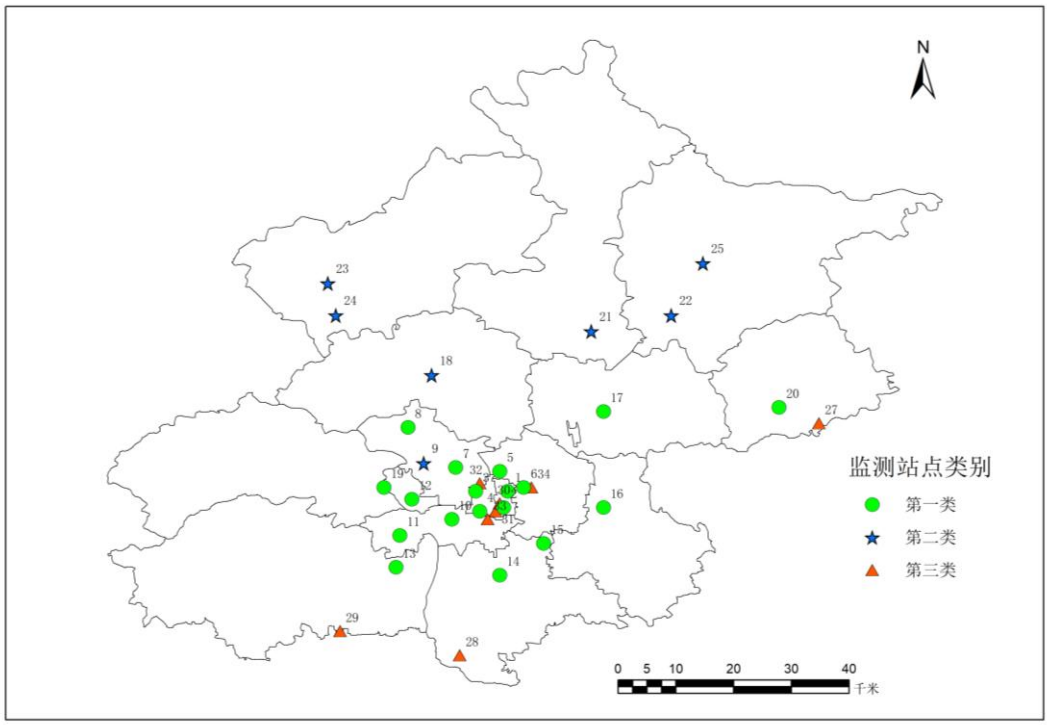


图 3.5 RMNMFFC 算法聚类结果空间分布图

图 3.5 表示基于 RMNMFFC 算法聚类结果生成的北京市空气质量监测站点空间分布图，将其与北京市现有监测站点类型划分(表 3.3)进行对比分析，可以看出：表 3.3 中的城市环境评价点(1-23)大部分位于聚类结果的第一类(除 9 号“香山”、18 号“昌平镇”、21 号“怀柔镇”、22 号“密云镇”、23 号“夏都”等 5 个站点)，分布较为分散，位于北京市的各个区域；聚类结果的第二类包含了两个区域背景传输站点，其分布在北京市的外围，处于东部以及南部的郊区位置；表 3.3 中所有交通污染控制点(30-34)都位于聚类结果的第三类中，通过图 3.5，可以发现这些监测站点都位于北京市的中心城区，这里人口多，密度大，交通流量也大，相应地在一个类别之中。因此，结合北京市现有空气质量监测站点的类型，可以发现：图中的第一类、第二类和第三类分别代表城市环境评价点、区域背景传输点以及交通污染控制点。实例应用结果显示，RMNMFFC 在识别空气质量监测站点的空间布局方面显示出卓越的适用性，可以有效揭示站点间的空间分布规律，对于优化站点布局具有重要意义。

3.6 本章小结

本章提出的 RMNMFFC 算法，是一种结合了图正则化和非负矩阵分解思想的多视角函数型聚类算法。该算法使用结构化稀疏$l_{2,1}$范数来定义损失函数，具有较强的鲁棒性；在降维过程中，利用图正则化从样本的相似性上维持数据与特征的原始空间结构，进而提升算法性能。在实验数据集上的分析证明了算法在具有鲁棒性的同时可以提高聚类性能；在北京市空气质量数据上的实例应用也表明该算法在解决实际问题中的应用价值。

4 基于非负矩阵分解的鲁棒双正交多视角函数型聚类算法

针对函数型数据高维且大规模的特性，本章基于非负矩阵分解，提出了鲁棒双正交多视角函数型聚类算法(Robust Multi-View Functional Clustering Algorithm with Co-orthogonal constraints，RMNMFFC-CC)。该算法结合非负矩阵分解和多视角学习，采用$l_{2,1}$范数，对噪声和异常值具有鲁棒性；引入图正则化，考虑数据的局部几何特征，集成多视角异构特征；利用表示矩阵和基矩阵的正交性，在达到最优降维的同时提高聚类性能。

4.1 问题概述

随着信息化时代的发展，收集到数据的维度与体量正在逐渐扩大，呈现出爆炸式的增长趋势。为了处理这类数据，基于降维的方法引起了人们的关注，如主成分分析(PCA)[74]、独立分量分析(ICA)[75]等。在这些技术中，非负矩阵分解(NMF)由于具有处理高维数据的能力，在多视角聚类中有着多种应用。

虽然 NMF 可以处理高维数据，但其聚类性能容易受到所分解的基矩阵以及表示矩阵的内向量正交性的影响。为了学习所需要的降维表示，一个自然的方案就是在传统的 NMF 中加入约束。而相关研究证明，在 NMF 中加入约束可以得到自己更为准确的结果。Ding 等(2006)[76]给出了 NMF、单正交 NMF、双正交 NMF 的推导，并给出双正交约束的确可以提高聚类效果；Wang 等(2016)[77]在字典学习中证明了合适的基矩阵可以更好的表示聚类；Wang 等(2018)[78]基于 NMF，进行多视角数据聚类研究，他采用正交性进而衡量每个视图内部信息的多样性；Yang 等(2020)[44]基于半监督非负矩阵分解，提出了带有双正交约束的半监督正则化深度非负矩阵分解(SGNMF)，由于矩阵的双正交性，提高了聚类性能；Liang 等(2020)[45]在此基础上，提出了一种具有双正交约束的 NMF 算法(NMF-CC)，该算法利用基矩阵和表示矩阵的正交性进行降维，从而提高聚类性能；Li 等(2021)[46]基于 NMF，提出了一种半监督的双图正则化 NMF 与双正交约束(SDGNMF-BO)，实现了更好的局部表示。

基于上述启发，本章基于非负矩阵分解，围绕函数型数据，提出了一种新的多视角函数型聚类算法，即鲁棒双正交多视角函数型聚类算法(Robust Multi-

View Functional Clustering Algorithm with co-orthogonal constraints，RMNMFFC-CC)。该算法结合多视角学习，采用$l_{2,1}$范数定义损失函数，具有鲁棒性；引入图正则化，考虑数据的局部几何特征，集成多视角异构特征；同时采用表示矩阵和基矩阵的正交性在达到最优降维的同时提高聚类性能。

4.2 算法模型

假设$\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_{n_v}\}$表示所有$n_v$个视角的数据，$\boldsymbol{Y}_v \in \mathbb{R}^{d_v \times n}$表示第$v$个视角，其中$v = 1, 2, 3, \cdots, n_v$。对于每个视角的原始数据$\boldsymbol{Y}_v$是不能直接观测到的，因此，采用矩阵分解的方法将其函数拟合为$\boldsymbol{Y}_v \approx \boldsymbol{\Phi}_v \boldsymbol{A}_v$，其中$\boldsymbol{\Phi}_v \in \mathbb{R}^{d_v \times r}$是基矩阵，$\boldsymbol{A}_v \in \mathbb{R}^{r \times n}$为完全决定曲线之间差异的系数矩阵。进一步利用 NMF 将系数矩阵$\boldsymbol{A}_v$分解为两个低秩矩阵的乘积，即$\boldsymbol{A}_v \approx \boldsymbol{U}_v \boldsymbol{V}_v^T$。

综上，可以将原始数据矩阵$\boldsymbol{Y}_v$用下式表示

$$\boldsymbol{Y}_v \approx \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T$$

其中，$\boldsymbol{U}_v \in \mathbb{R}^{r \times K}$为第$v$个视角的基矩阵，$\boldsymbol{V}_v^T \in \mathbb{R}^{K \times n}$为第$v$个视角的聚类指示矩阵。

采用结构化稀疏性$l_{2,1}$范数，降低模型对噪声和异常值的影响，以确保模型的鲁棒性；引入局部流形正则化，利用流形的局部不变性将高维空间$\boldsymbol{Y}_v$中样本的几何结构在矩阵分解后仍能保持在低维空间$\boldsymbol{V}_v$中；对矩阵$\boldsymbol{U}_v$和矩阵$\boldsymbol{V}_v$施加双正交约束，保证算法在对高维大规模数据优化求解时仍能获得更好的聚类性能。

鲁棒双正交多视角函数型聚类算法的目标函数可以表示为

$$\begin{aligned} \min_{\boldsymbol{U}_v, \boldsymbol{V}_v} \ & \sum_{v=1}^{n_v} \Big\{ \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T\|_{2,1} + \lambda \operatorname{tr}\left(\boldsymbol{V}_v^T \boldsymbol{L}_v \boldsymbol{V}_v\right) \\ & \quad + \alpha \left(\mathbf{1}^T \boldsymbol{U}_v^T \boldsymbol{U}_v \mathbf{1} - \operatorname{tr}\left(\boldsymbol{U}_v^T \boldsymbol{U}_v\right)\right) + \frac{\mu}{2} \|\boldsymbol{V}_v \boldsymbol{V}_v^T - \boldsymbol{I}\|_F^2 \Big\} \\ \text{s.t.} \ & \boldsymbol{U}_v \geq 0, \boldsymbol{V}_v \geq 0, \forall v \end{aligned} \tag{4.1}$$

其中，λ为正则化参数，α, μ为调节参数。$\mathbf{1}$表示元素全部为 1 的列向量。$\boldsymbol{L}_v = \hat{\boldsymbol{W}}_v - \boldsymbol{W}_v$为拉普拉斯矩阵，$\boldsymbol{W}_v = (W_{vij})$为相似矩阵，$\hat{\boldsymbol{W}}_v$为度矩阵，两者的表示为

$$W_{vij} = \begin{cases} 1 & \boldsymbol{y}_{vi} \in N_k(\boldsymbol{y}_{vj}) \ \text{ or } \ \boldsymbol{y}_{vj} \in N_k(\boldsymbol{y}_{vi}) \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\boldsymbol{W}}_v = \operatorname{diag}\left([\sum_{i=1}^{n} w_{vi1}, \cdots, ([\sum_{i=1}^{n} w_{vin}]\right)$$

在构建最邻近图时，采用 0-1 加权策略，$N_k(\boldsymbol{y}_{vj})$表示第v个视角的第j个样本的k近邻，包含了与样本$\boldsymbol{y}_{vj}$距离最近的k个样本数据。任意两个样本数据之间的距离定义为

$$d\left(\boldsymbol{y}_{vi}, \boldsymbol{y}_{vj}\right) = \sqrt{\sum_{t=1}^{m}(y_{vti} - y_{vtj})^2}$$

式(4.1)中，第一项采用$l_{2,1}$范数，确保模型的鲁棒性；第二项为正则化项，可以在低维表示$\boldsymbol{v}_i$和$\boldsymbol{v}_j$中保留$\boldsymbol{y}_i$和$\boldsymbol{y}_j$有相似特征的局部信息，从而大大提高其表示能力和聚类性能；第三项为对第v个视角的基矩阵$\boldsymbol{U}_v$施加的正交约束，对基矩阵的列向量进行学习，从而使其对应的表示矩阵拥有更好的聚类性能；第四项为对第v个视角的聚类指示矩阵$\boldsymbol{V}_v$的正交约束，用来衡量每个视图间的多样性。

4.3 求解算法

4.3.1 优化求解

为了优化模型(4.1)，采用交替更新技术，即在优化一个变量的同时固定其他变量。由于变量矩阵对于不同的视角是独立的，我们着重推导与第v个视角相关的更新公式$\boldsymbol{U}_v, \boldsymbol{V}_v$。模型(4.1)的损失函数可以简化为

$$\begin{aligned} L_v =& \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T\|_{2,1} + \lambda \operatorname{tr}\left(\boldsymbol{V}_v^T \boldsymbol{L}_v \boldsymbol{V}_v\right) \\ &+ \alpha\left(\boldsymbol{1}^T \boldsymbol{U}_v^T \boldsymbol{U}_v \boldsymbol{1} - \operatorname{tr}\left(\boldsymbol{U}_v^T \boldsymbol{U}_v\right)\right) + \frac{\mu}{2}\|\boldsymbol{V}_v \boldsymbol{V}_v^T - \boldsymbol{I}\|_F^2 \end{aligned} \tag{4.2}$$

由于$l_{2,1}$范数是非光滑的，可将此优化问题式(2)分割为几个子问题，然后分别采用乘性更新迭代方法更新求解模型。

令$L_v = (L_1)_v + \lambda (L_2)_v + \alpha (L_3)_v + \frac{\mu}{2}(L_4)_v$，其中

$$\begin{aligned} (L_1)_v &= \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v \boldsymbol{U}_v \boldsymbol{V}_v^T\|_{2,1} \\ (L_2)_v &= \operatorname{tr}\left(\boldsymbol{V}_v^T \boldsymbol{L}_v \boldsymbol{V}_v\right) \\ (L_3)_v &= \left(\boldsymbol{1}^T \boldsymbol{U}_v^T \boldsymbol{U}_v \boldsymbol{1} - \operatorname{tr}\left(\boldsymbol{U}_v^T \boldsymbol{U}_v\right)\right) \\ (L_4)_v &= \|\boldsymbol{V}_v \boldsymbol{V}_v^T - \boldsymbol{I}\|_F^2 \end{aligned} \tag{4.3}$$

为书写方便起见，记

$$\begin{aligned}(L_1)_v &= \|\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\|_{2,1} \\ &= \mathrm{tr}\left[(\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T)^T\boldsymbol{G}_v(\boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T)\right] \\ &= \mathrm{tr}\left[\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v - 2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T + \boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\right]\end{aligned}$$

其中，$\boldsymbol{G}_v = \mathrm{diag}\left\{g_{v11}, g_{v22}, \cdots, g_{vd_vd_v}\right\} \in \mathbb{R}^{d_v \times d_v}$是对应于第$v$个视角的对角矩阵，对角线上的第$i$项定义为：

$$g_{vii} = \frac{1}{\|\boldsymbol{e}_v^i\|}, \quad \forall \ \ i = 1, 2, \ldots, d_v \tag{4.4}$$

其中，$\boldsymbol{e}_v^i = (e_{vi1}, e_{vi2}, \cdots, e_{vin})^T$是以下矩阵$\boldsymbol{E}_v$的第$i$行形成的列向量

$$\boldsymbol{E}_v = \boldsymbol{Y}_v - \boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T = \left(\boldsymbol{e}_v^1, \boldsymbol{e}_v^2, \cdots, \boldsymbol{e}_v^{d_v}\right)^T \tag{4.5}$$

记式(4.2)的拉格朗日函数为$\mathcal{L}_v$，则有

$$\mathcal{L}_v = L_v - \boldsymbol{\Lambda}_v\boldsymbol{U}_v^T - \boldsymbol{\Gamma}_v\boldsymbol{V}_v^T$$

满足 KKT 条件

$$\begin{aligned}\boldsymbol{\Lambda}_v \odot \boldsymbol{U}_v = 0 \\ \boldsymbol{\Gamma}_v \odot \boldsymbol{V}_v = 0\end{aligned} \tag{4.6}$$

其中，$\odot$为 Hadamard 积。在此基础上，依次对变量矩阵$\boldsymbol{U}_v, \boldsymbol{V}_v$进行更新求解，具体求解的更新规则如下。

（1）固定$\boldsymbol{G}_v, \boldsymbol{V}_v$，求$\boldsymbol{U}_v$。

分别对L_v的子问题进行求导，通过传统的矩阵运算，可以得到

$$\begin{aligned}\frac{\partial (L_1)_v}{\partial \boldsymbol{U}_v} =& \frac{\partial \mathrm{tr}\left[\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v - 2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T + \boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\right]}{\partial \boldsymbol{U}_v} \\ =& -2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v + 2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\end{aligned}$$

因为$(L_2)_v$以及$(L_4)_v$是独立于$\boldsymbol{U}_v$的，所以有

$$\frac{\partial (L_2)_v}{\partial \boldsymbol{U}_v} = 0$$

$$\begin{aligned}\frac{\partial (L_3)_v}{\partial \boldsymbol{U}_v} =& \frac{\partial \left(\boldsymbol{1}^T\boldsymbol{U}_v^T\boldsymbol{U}_v\boldsymbol{1} - \mathrm{tr}\left(\boldsymbol{U}_v^T\boldsymbol{U}_v\right)\right)}{\partial \boldsymbol{U}_V} \\ =& \frac{\partial \mathrm{tr}\left[\boldsymbol{1}^T\boldsymbol{U}_v^T\boldsymbol{U}_v\boldsymbol{1} - \boldsymbol{U}_v^T\boldsymbol{U}_v\right]}{\partial \boldsymbol{U}_V} \\ =& \frac{\partial \mathrm{tr}\left[\boldsymbol{1}\boldsymbol{1}^T\boldsymbol{U}_v^T\boldsymbol{U}_v - \boldsymbol{U}_v^T\boldsymbol{U}_v\right]}{\partial \boldsymbol{U}_V} \\ =& 2\boldsymbol{U}_v^T\boldsymbol{1}\boldsymbol{1}^T - 2\boldsymbol{U}_v\end{aligned}$$

$$\frac{\partial (L_4)_v}{\partial \boldsymbol{U}_v} = 0$$

所以可得

$$\begin{aligned}\frac{\partial \mathcal{L}_v}{\partial \boldsymbol{U}_v} =&\frac{\partial (L_1)_v}{\partial \boldsymbol{U}_v}+\lambda\frac{\partial (L_2)_v}{\partial \boldsymbol{U}_v}+\alpha\frac{\partial (L_3)_v}{\partial \boldsymbol{U}_v}+\frac{\mu}{2}\frac{\partial (L_4)_v}{\partial \boldsymbol{U}_v}-\boldsymbol{\Lambda}_v\\ =&-2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v+2\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+2\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T-2\alpha\boldsymbol{U}_v-\boldsymbol{\Lambda}_v\\ =&2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T-\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v-\alpha\boldsymbol{U}_v\right)-\boldsymbol{\Lambda}_v\end{aligned} \tag{4.7}$$

令$\frac{\partial \mathcal{L}_v}{\boldsymbol{U}_v}=0$，可得

$$\boldsymbol{\Lambda}_v=2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T-\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v-\alpha\boldsymbol{U}_v\right) \tag{4.8}$$

结合式(4.6)和式(4.8)，有

$$\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T-\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v-\alpha\boldsymbol{U}_v\right)\odot\boldsymbol{\Lambda}_v=0$$

从而得到$\boldsymbol{U}_v$的更新公式为

$$\boldsymbol{U}_{vij}\leftarrow\boldsymbol{U}_{vij}\frac{\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v+\alpha\boldsymbol{U}_v}{\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T} \tag{4.9}$$

（2）固定$\boldsymbol{G}_v,\boldsymbol{U}_v$，求$\boldsymbol{V}_v$。

分别对L_v的子问题进行求导，通过传统的矩阵运算，可以得到

$$\begin{aligned}\frac{\partial (L_1)_v}{\partial \boldsymbol{V}_v} =&\frac{\partial \mathrm{tr}\left[\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v-2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T+\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\right]}{\partial \boldsymbol{U}_v}\\ =&-2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+2\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\end{aligned}$$

$$\begin{aligned}\frac{\partial (L_2)_v}{\partial \boldsymbol{V}_v} =&\frac{\partial \mathrm{tr}\left(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v\right)}{\boldsymbol{V}_v}\\ =&2\boldsymbol{V}_v\boldsymbol{L}_v\\ =&2\boldsymbol{V}_v\left(\hat{\boldsymbol{W}}_v-\boldsymbol{W}_v\right)\end{aligned}$$

因为$(L_3)_v$是独立于$\boldsymbol{V}_v$的，所以有

$$\frac{\partial (L_3)_v}{\partial \boldsymbol{V}_v}=0$$

$$\begin{aligned}\frac{\partial (L_4)_v}{\partial \boldsymbol{V}_v} =&\frac{\partial\|\boldsymbol{V}_v\left(\boldsymbol{V}_v^T\right)-\boldsymbol{I}\|_F^2}{\partial \boldsymbol{V}_V}\\ =&\frac{\partial \mathrm{tr}\left[\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\boldsymbol{V}_v^T-2\boldsymbol{V}_v\boldsymbol{V}_v^T\right]}{\partial \boldsymbol{V}_V}\\ =&4\left(\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v-\boldsymbol{V}_v\right)\end{aligned}$$

所以可得

$$\begin{aligned}\frac{\partial \mathcal{L}_v}{\partial \boldsymbol{V}_v} =&\frac{\partial (L_1)_v}{\partial \boldsymbol{V}_v}+\lambda\frac{\partial (L_2)_v}{\partial \boldsymbol{V}_v}+\alpha\frac{\partial (L_3)_v}{\partial \boldsymbol{V}_v}+\frac{\mu}{2}\frac{\partial (L_4)_v}{\partial \boldsymbol{V}_v}-\boldsymbol{\Gamma}_v\\ =&-2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+2\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+2\lambda\boldsymbol{V}_v\left(\hat{\boldsymbol{W}}_v-\boldsymbol{W}_v\right)\\ &+2\mu\left(\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v-\boldsymbol{V}_v\right)-\boldsymbol{\Gamma}_v\\ =&-2\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+2\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+2\lambda\boldsymbol{V}_v\hat{\boldsymbol{W}}_v\\ &-2\lambda\boldsymbol{V}_v\boldsymbol{W}_v+2\mu\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v-2\mu\boldsymbol{V}_v-\boldsymbol{\Gamma}_v\end{aligned} \tag{4.10}$$

$$
\begin{aligned}
=&2\Big(\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{V}_v\hat{\boldsymbol{W}}_v+\mu\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\\
&-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v-\lambda\boldsymbol{V}_v\boldsymbol{W}_v-\mu\boldsymbol{V}_v\Big)-\boldsymbol{\Gamma}_v
\end{aligned}
$$

令$\frac{\partial\mathcal{L}_v}{\boldsymbol{V}_v}=0$，可得

$$
\begin{aligned}
\boldsymbol{\Gamma}_v=&2\Big(\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{V}_v\hat{\boldsymbol{W}}_v+\mu\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\\
&-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v-\lambda\boldsymbol{V}_v\boldsymbol{W}_v-\mu\boldsymbol{V}_v\Big)
\end{aligned}
\tag{4.11}
$$

结合式(4.6)和式(4.11)，有

$$
\Big(\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{V}_v\hat{\boldsymbol{W}}_v+\mu\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v-\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v-\lambda\boldsymbol{V}_v\boldsymbol{W}_v-\mu\boldsymbol{V}_v\Big)\odot\boldsymbol{\Gamma}_v=0
$$

从而得到$\boldsymbol{V}_v$的更新公式为

$$
\boldsymbol{V}_{vij}\leftarrow\boldsymbol{V}_{vij}\frac{\boldsymbol{Y}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{V}_v\boldsymbol{W}_v+\mu\boldsymbol{V}_v}{\boldsymbol{V}_v\boldsymbol{U}_v^T\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v+\lambda\boldsymbol{V}_v\hat{\boldsymbol{W}}_v+\mu\boldsymbol{V}_v\boldsymbol{V}_v^T\boldsymbol{V}_v}
\tag{4.12}
$$

（3）固定$\boldsymbol{U}_v,\boldsymbol{V}_v$，更新$\boldsymbol{G}_v$。

利用式(4.4)和式(4.5)对$\boldsymbol{G}_v$进行更新。

4.3.2 算法步骤

根据 4.3.1 的更新过程，依次交替迭代对$\boldsymbol{U}_v,\boldsymbol{V}_v$和$\boldsymbol{G}_v$进行更新，即可求解式(4.1)的优化问题，实现鲁棒双正交多视角函数型聚类算法(RMNMFFC-CC)，算法的具体步骤如下所示。

算法 4.1 RMNMFFC-CC 算法

输入：数据矩阵$\boldsymbol{Y}_v$，基底矩阵$\boldsymbol{\Phi}_v$，参数λ,α,μ和类别数K

过程：

1：首先构造拉普拉斯矩阵$\boldsymbol{L}_v$

2：其次对矩阵$\boldsymbol{U}_v^0$，$\boldsymbol{V}_v^0$进行初始化，同时令$\boldsymbol{G}_v=\boldsymbol{I}_{d_n}$

3：for $t=1,2,\cdots$最大更新迭代次数

4：　for $v=1,2,\cdots,n_v$

5：　　固定$\boldsymbol{V}_v^{t-1}$和$\boldsymbol{G}_v^{t-1}$，根据式(4.9)更新$\boldsymbol{U}_v^t$;

6：　　固定$\boldsymbol{U}_v^t$和$\boldsymbol{G}_v^{t-1}$，根据式(4.12)更新$\boldsymbol{V}_v^t$;

7：　　固定$\boldsymbol{U}_v^t$和$\boldsymbol{V}_v^t$，根据式(4.4)和式(4.5)更新$\boldsymbol{G}_v^t$;

8：　end for

9：　if 式(4.1)收敛

10：　　break

11：　end if

12：end for

输出：$\boldsymbol{U}_v^t$，$\boldsymbol{V}_v^t$和$\boldsymbol{G}_v^t$，类别划分$\mathcal{C}=\{C_1,C_2,\ldots,C_K\}$

4.3.3 算法收敛性证明

采用辅助函数法证明算法 4.1 的收敛性。

对于算法 4.1，有下列结论成立。

定理 4.1 (1)固定$\boldsymbol{G}_v$，$\boldsymbol{U}_v$的更新规则(式(4.9))是

$$\begin{aligned}\min_{\boldsymbol{U}_v,\boldsymbol{V}_v}\ & \sum_{v=1}^{n_v}\Big\{\|\boldsymbol{Y}_v-\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\|_{2,1}+\lambda\mathrm{tr}\left(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v\right)\\ &+\alpha\left(\mathbf{1}^T\boldsymbol{U}_v^T\boldsymbol{U}_v\mathbf{1}-\mathrm{tr}\left(\boldsymbol{U}_v^T\boldsymbol{U}_v\right)\right)+\frac{\mu}{2}\|\boldsymbol{V}_v\boldsymbol{V}_v^T-\boldsymbol{I}\|_F^2\Big\}\\ \text{s.t.}\ & \boldsymbol{U}_v\geq 0,\boldsymbol{V}_v\geq 0,\forall v\end{aligned}$$

的最优解；(2)固定$\boldsymbol{G}_v$，目标函数(式(4.1))在$\boldsymbol{U}_v$的更新规则(式(4.9))下是非增的。

显然，固定$\boldsymbol{G}_v$，

$$\begin{aligned}\min_{\boldsymbol{U}_v,\boldsymbol{V}_v}\ & \sum_{v=1}^{n_v}\Big\{\|\boldsymbol{Y}_v-\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\|_{2,1}+\lambda\mathrm{tr}\left(\boldsymbol{V}_v^T\boldsymbol{L}_v\boldsymbol{V}_v\right)\\ &+\alpha\left(\mathbf{1}^T\boldsymbol{U}_v^T\boldsymbol{U}_v\mathbf{1}-\mathrm{tr}\left(\boldsymbol{U}_v^T\boldsymbol{U}_v\right)\right)+\frac{\mu}{2}\|\boldsymbol{V}_v\boldsymbol{V}_v^T-\boldsymbol{I}\|_F^2\Big\}\\ \text{s.t.}\ & \boldsymbol{U}_v\geq 0,\boldsymbol{V}_v\geq 0,\forall v\end{aligned}$$

是一个带约束的最优化问题。在上述算法求解过程中(即式(4.9)的求解过程)可以证明定理 2.1 的结论(1)成立。

采用辅助函数法对定理 2.1 中的结论(2)进行证明。由引理 2.1 可知，需要寻找恰当的辅助函数，从而证明目标函数(式(4.1))在更新规则(式(4.9)和式(4.12))下是非增的。下面首先构造辅助函数，然后通过辅助函数法证明定理 2.1 的结论(2)。

命题 4.1 令F'是函数F的一阶导函数，则

$$\begin{aligned}G\left(u,u_{vij}^t\right)=&F_{ij}\left(u_{vij}^t\right)+F_{ij}'\left(u_{v_{ij}}^t\right)\left(u-u_{vij}^t\right)\\ &+\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T\right)_{ij}\left(u_v-u_{vij}^t\right)^2}{u_{v_{ij}}^t}\end{aligned}\tag{4.16}$$

是函数$F_{ij}(u)$的辅助函数，且满足$G(u,u)=F_{ij}(u)$且$G(u,u_{vij}^t)\geq F_{ij}(u)$。其中$F_{ij}(u)$是目标函数(式(4.1))中关于$\boldsymbol{U}_v$的部分，$t$指当前的迭代次数。

证明 显然$G(u,u)=F_{ij}(u)$，根据辅助函数的定义，还需证明$G(u,u_{vij}^t)\geq F_{ij}(u)$。式(4.16)与$F_{ij}(u)$在$u=u_{vij}^t$处的泰勒展开式

$$F_{ij}(u)=F_{ij}(u^t)+F_{ij}^{'}(u^t)(u-u_{vij}^t)+\frac{1}{2}F_{ij}^{''}(u^t)(u_v-u_{vij}^t)^2$$

进行比较，只需证明

$$\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T\right)_{ij}\left(u-u_{vij}^t\right)^2}{u_{vij}^t}\geq\frac{1}{2}F_{ij}^{''}(u)\left(u-u_{vij}^t\right)^2 \quad (4.17)$$

对于$F_{ij}(u)$的一阶导和二阶导很容易求得

$$\begin{aligned}F_{ij}^{'}(u)=&2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T-\alpha\boldsymbol{U}_v\right)_{ij}\\F_{ij}^{''}(u)=&2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}+2\alpha\left(\left(\mathbf{1}\mathbf{1}^T\right)_{ij}-1\right)\\=&2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}\end{aligned}$$

则式(4.17)可以转化为

$$\begin{aligned}\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v\mathbf{1}\mathbf{1}^T\right)_{ij}}{u_{v_{ij}}^t}\geq&\frac{1}{2}F_{ij}^{''}(u)\\=&\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}\end{aligned} \quad (4.18)$$

对式(4.18)基于代数操作求解，有

$$\begin{aligned}\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}=&\sum_l^k u_{vij}^t\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}\\\geq&u_{vij}^t\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v\right)_{ij}\end{aligned}$$

由于式(4.18)中$\left(\boldsymbol{U}_v\mathbf{1}\mathbf{1}^T\right)_{ij}\geq 0$，因此有$G(u,u_{vij}^t)\geq F_{ij}(u)$。因为$G(u,u)=F_{ij}(u)$且$G(u,u_{vij}^t)\geq F_{ij}(u)$，所以$G(u,u)$是$F_{ij}(u)$的一个标准辅助函数。根据引理 2.1，可得$F_{ij}(u)$是递减的。

令$\frac{G(u_{vij}^{t+1},u_{vij}^t)}{u_{vij}^{t+1}}=0$可以求解得到$u_{vij}^{t+1}$更新规则

$$\begin{aligned}u_{vij}^{t+1}=&u_{vij}^t-\frac{u_{vij}^tF_{ij}^{'}}{2\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T\right)_{ij}}\\=&u_{vij}^t\frac{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{Y}_v\boldsymbol{V}_v+\alpha\boldsymbol{U}_v\right)_{ij}}{\left(\boldsymbol{\Phi}_v^T\boldsymbol{G}_v\boldsymbol{\Phi}_v\boldsymbol{U}_v\boldsymbol{V}_v^T\boldsymbol{V}_v+\alpha\boldsymbol{U}_v^T\mathbf{1}\mathbf{1}^T\right)_{ij}}\end{aligned} \quad (4.19)$$

以上更新规则式(4.19)等同于式(4.9)，式(4.16)是辅助函数，且在更新规则(式(4.9))下，$F_{ij}(u)$是非增的。采用相似的方法可以证得$F_{ij}(v)$在更新规则(式(4.12))下是非增的。由此，证明了算法 4.1 的收敛性。

4.4 模拟分析

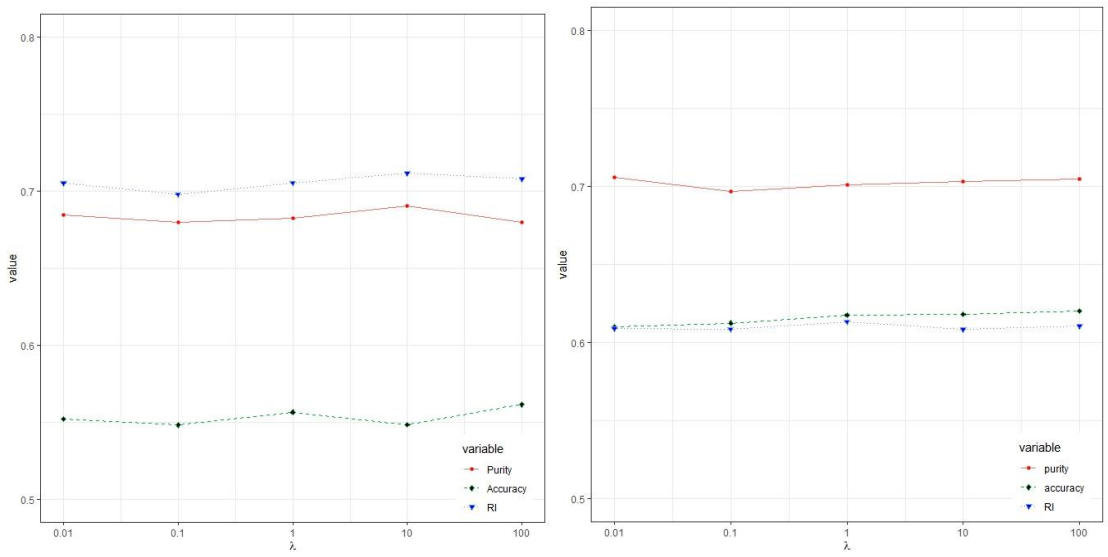
为了说明 RMNMFFC-CC 算法的有效性，下面进行模拟实验。模拟实验数据为随机模拟数据集和 Growth 数据集(见第 3 章 3.4.1)。在参数设置一致的前提下，将其与相关的函数型聚类算法进行对比。同时采用纯度(Purity)、聚类精度(Accuracy)和兰德指数(RI)作为聚类评价指标，对算法的性能进行细致的比较和分析。

4.4.1 参数设置及对比方法

在进行模拟分析时，对数据集的参数设置与 3.4.1 保持一致；本章采用的对比算法有一元函数型聚类算法：函数型子空间聚类(FFKM，Yamamoto 等，2014)[16]、函数型聚类一步法(FCOF)[21]，多元函数型聚类方法：Funclust(Jacques 等，2013)[30]、funHDDC(Julien 等，2014)[29]与 RMNMFFC(见第三章)五种方法进行对比。需要说明 funHDDC 聚类方法通过在特定函数子空间内对各组进行建模，其包含 6 个子模型(详见 Schmutz 等(2020)[31]的描述)，该方法的结果是对 6 个子模型的聚类结果取最优。

4.4.2 参数敏感性分析

考虑超参数λ对聚类性能的影响，取$\lambda=\{0.01,0.1,1,10,100\}$，在随机模拟数据集(图 3.1)和 Growth 数据集(图 3.2)上分别进行聚类，聚类的评价指标结果如图 4.1 所示：



(a) 随机模拟数据 (b) Growth 数据

图 4.1 正则化参数λ与聚类性能的关系

从图 4.1 可以看出，在随机模拟数据集和 Growth 数据集上，聚类结果对超参数λ的变化不敏感。因此，说明 RMNMFFC-CC 算法对正则化参数λ具有很好的鲁棒性。

4.4.3 聚类结果分析

在参数设置一致的基础上，对随机模拟数据集和 Growth 数据集进行聚类分析，聚类评价结果如表 4.1 和表 4.2 所示。其中，表 4.1 和表 4.2 中粗体表示聚类评价结果最优。

表 4.1 随机模拟数据集的聚类评价结果(均值±标准差)

评价指标 / 聚类算法		聚类纯度 (Purity)	聚类精度 (Accuracy)	兰德指数 (RI)
一元方法	FFKM	0.33 ± 0.0080	0.32 ± 0.0014	0.55 ± 0.0071
	FCOF	0.68 ± 0.0732	0.51 ± 0.0818	0.67 ± 0.0605
多元方法	Funclust	0.49 ± 0.0531	0.36 ± 0.0433	0.50 ± 0.0736
	funHDDC	0.58 ± 0.0507	0.42 ± 0.0435	0.61 ± 0.0471
	RMNMFFC	0.62 ± 0.0939	0.46 ± 0.0863	0.63 ± 0.0729
	RMNMFFC-CC	**0.69 ± 0.0526**	**0.54 ± 0.0246**	**0.72 ± 0.0209**

表 4.2 Growth 数据集的聚类评价结果(均值±标准差)

聚类算法 \ 评价指标		聚类纯度 (Purity)	聚类精度 (Accuracy)	兰德指数 (RI)
一元方法	FFKM	0.66 ± 0.0023	0.53 ± 0.0015	0.56 ± 0.0032
	FCOF	0.69 ± 0.0035	0.57 ± 0.0020	0.58 ± 0.0020
多元方法	Funclust	0.63 ± 0.0379	0.55 ± 0.0370	0.55 ± 0.0487
	funHDDC	0.60 ± 0.0547	0.52 ± 0.0893	0.53 ± 0.0788
	RMNMFFC	**0.72 ± 0.0105**	0.60 ± 0.0084	0.59 ± 0.0087
	RMNMFFC-CC	0.71 ± 0.0235	**0.61 ± 0.0184**	**0.60 ± 0.0173**

表 4.1 和表 4.2 的结果表明，在随机模拟数据集以及 Growth 数据集上，聚类精度和兰德指数，RMNMFFC-CC 算法的结果明显优于多元函数型聚类算法 Funclust、funHDDC 以及第三章提出的算法 RMNMFFC，同时也略优于一元函数型聚类算法 FFKM 和 FCOF。对于聚类纯度，可以发现 RMNMFFC-CC 在 Growth 数据集上的结果仅低于第三章提出的算法 RMNMFFC。

为了进一步验证算法的性能，另外使用 TIMIT 语音数据集进行分析。该数据集收集了使用美国不同方言读出给定句子时的语音数据，在语音识别分类研究领域有广泛的应用价值。本次研究选取数据库中名为“SA1”的语音数据集[③]进行分析，该数据集具有 256 个维度，划分为 5 个类别，含有 4509 个样本。

对 TIMIT 语音数据集的原始数据进行数字信号处理，得到相应的音素数据和对应的音素类别标签。任取一个样本的部分语音数据，采用对数对数周期图法(Log-periodgram)进行处理，并进行可视化操作，如图 4.2 所示。

[③] “SA1”语音数据的文本内容为“She had your dark suit in greasy wash water all year”，其划分为 5 各类别，“sh”代表“she”这一单词起始的辅音音素，“dcl”则是“dark”这一单词中辅音部分的标记，“iy”作为“she”中的元音音素，“aa”则对应“dark”中的元音部分，“ao”是“water”这一单词中首个元音的标记。

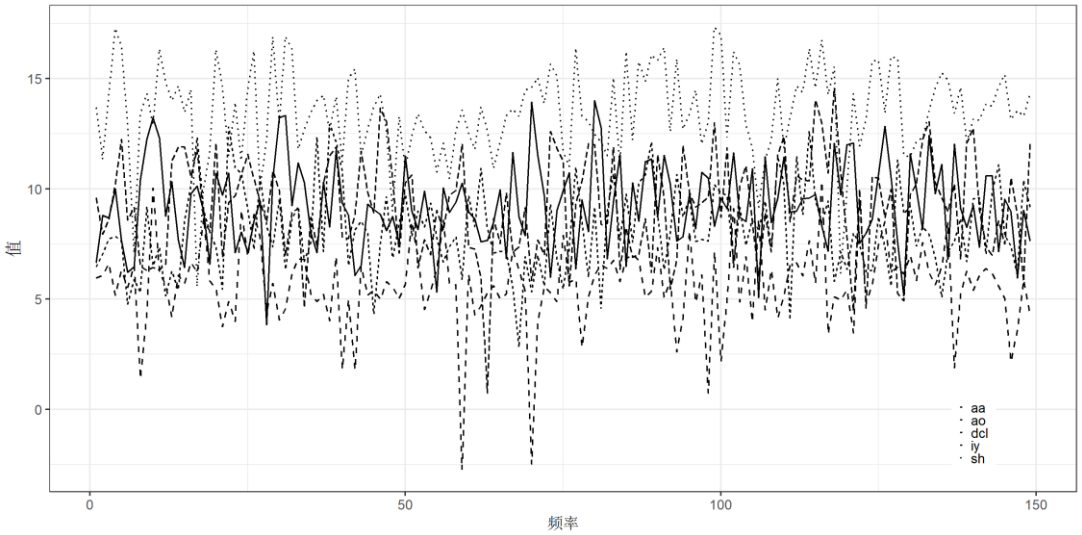


图 4.2 TIMIT 单个样本的语音数据及类别标签

图 4.2 中，纵坐标是对音频数据进行处理后的数值，横坐标表示频率(Frequency)的变化，此外，为了区分不同的数据类别，采用了不同的线型进行表示。

对 TIMIT 语音数据集进行聚类分析时，取视角数 n_v=256，类别数 $K=5$，采用等距节点 B-样条基底，将基底数量设为 20。将 RMNMFFC-CC 算法与相应的函数型聚类算法进行比较，聚类评价结果如表 4.3 所示。

表 4.3 TIMIT 语音数据聚类评价结果(均值±标准差)

聚类算法 \ 评价指标		聚类纯度 (Purity)	聚类精度 (Accuracy)	兰德指数 (RI)
一元方法	FFKM	0.70 ± 0.0210	0.58 ± 0.0142	0.81 ± 0.0165
多元方法	Funclust	0.58 ± 0.0341	0.56 ± 0.0233	0.78 ± 0.0346
	RMNMFFC	0.68 ± 0.0451	0.58 ± 0.0563	0.76 ± 0.0392
	RMNMFFC-CC	**0.73 ± 0.0122**	**0.64 ± 0.0238**	**0.82 ± 0.0161**

表 4.3 的结果表明，在 TIMIT 语音数据集上，RMNMFFC-CC 算法的三个聚类评价结果明显优于多元函数型聚类算法 Funclust 及第三章提出的算法 RMNMFFC，同时也优于一元函数型聚类算法 FFKM。因此，该算法的综合表现较好，在处理高维大规模数据时具有相应的优势。

4.5 实例分析——以甘肃省行政区划气象数据聚类为例

为了获得更加准确的气候分区，需要综合考虑多种气象要素。本节使用 RMNMFFC-CC 算法对甘肃省行政区划的气象数据进行聚类分析，从而进行气候分区。

甘肃省是中国地理条件十分复杂的省份，辖区内共有 12 个地级市，2 个自治州，位于三大高原(黄土高原、青藏高原和内蒙古高原)的交汇地带，同时也是东部季风区、西北干旱半干旱区和青藏高寒区的交界处。正因其独特的地理位置，甘肃省的气候类型从南向北呈现出了多样化的特点，涵盖了亚热带季风气候、温带季风气候、温带大陆性气候(干旱)和高原高寒气候四种气候类型[④]。

在不同的标准下，对于中国气候区域的划分结果略有不同。基于甘肃省的气候情况参考甘肃气候类型图[⑤]，将甘肃省各地区依照气候类型划分为 4 种类别，具体的划分见表 4.4 所示。

表 4.4 甘肃省各地区类别划分及特征

类别	地区	气候特征
亚热带季风气候	1 陇南市	夏季高温多雨，冬季低温少雨，雨热同期。
温带季风气候	2 庆阳市 3 平凉市 4 天水市 5 定西市 6 临夏回族自治州	夏季高温多雨，冬季寒冷干燥，雨热同期。
温带大陆性气候(干旱)	7 兰州市 8 白银市 9 武威市 10 金昌市 11 张掖市 12 酒泉市 13 嘉峪关市	冬冷夏热，降水稀少，气温年差较大。
高原高寒气候	14 甘南藏族自治州	降水较少，气温年较差小，气温日差较大。

将表 4.4 的地区按照气候类别划分，采用 ArcGIS 10.2 进行可视化空间展示，如图 4.3 所示。

④ http://www.gstb.gov.cn/zjgs/zrdl/201711/t20171116_11867994.html

⑤ http://map.ps123.net/china/19805.html

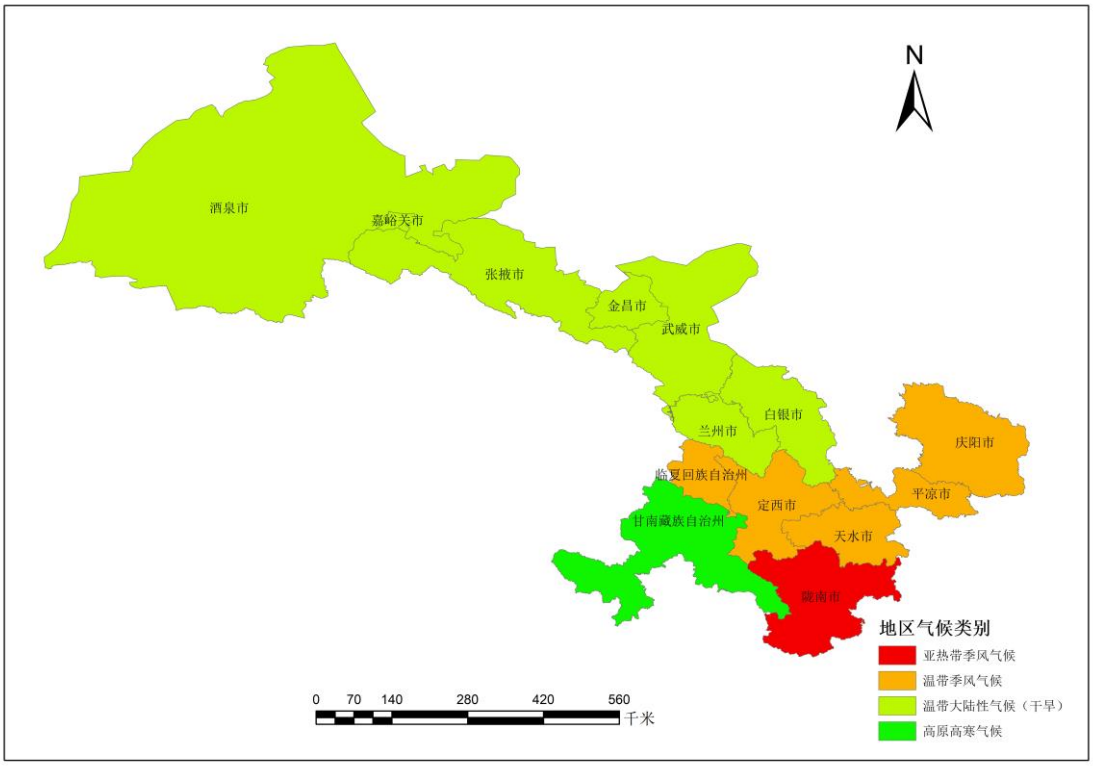


图 4.3 甘肃省各地区气候类别

从图 4.3 中可以发现甘肃省的气候类别由北向南、自西向东，逐渐由温带大陆性气候(干旱)过渡到其它气候，且气候类型为温带大陆性气候(干旱)的城市最多。

在国家环境信息中心(NICE, https://www.ncei.noaa.gov)网站上收集甘肃省 14 个地区自 2000 至 2021 年这 21 年间的逐日气象数据进行实证分析研究，具体考虑以下 5 个类别的气象数据：降水数据、湿度数据、日照数据、风速数据、平均气温数据。

对于原始数据，基于气象站点的经纬度，采用反距离权重法插值得到甘肃省范围的逐日平均栅格图，然后再基于甘肃省地级市的行政边界数据和栅格图，统计得到各个地级市的逐日平均相关气象数据。由于在进行数据处理时，发现气温数据存在负值，而该聚类算法需要保证数据的非负性，因此对数据进行预处理保证数据的非负性。

数据处理之后，运用 RMNMFFC-CC 算法对甘肃省的气象数据进行聚类分析。为了更好地进行分析，对相关参数做出以下设定：①各变量的调节参数相

等，即$\alpha_v = \frac{1}{5}(v = 1,2,\cdots\cdots, 5)$；②采用等距节点 3 次 B-样条基底拟合曲线；③进行聚类时，将迭代次数设为 200，聚类次数设为 100；④以各个地区的气候类别为标签检验聚类效果，设定聚类数$K = 4$。

对甘肃省 14 个地区 5 种气象数据的 RMNMFFC-CC 聚类结果如表 4.5 所示。

表 4.5 RMNMFFC-CC 聚类结果

类别	聚类结果
第一类	1 陇南市 2 庆阳市 3 平凉市 4 天水市 5 定西市
第二类	6 临夏回族自治州 14 甘南藏族自治州
第三类	7 兰州市 8 白银市
第四类	9 武威市 10 金昌市 11 张掖市 12 酒泉市 13 嘉峪关市

进一步采用 ArcGIS10.2 将表 4.5 中 RMNMFFC-CC 的聚类结果进行空间布局可视化展示，如图 4.3 所示。

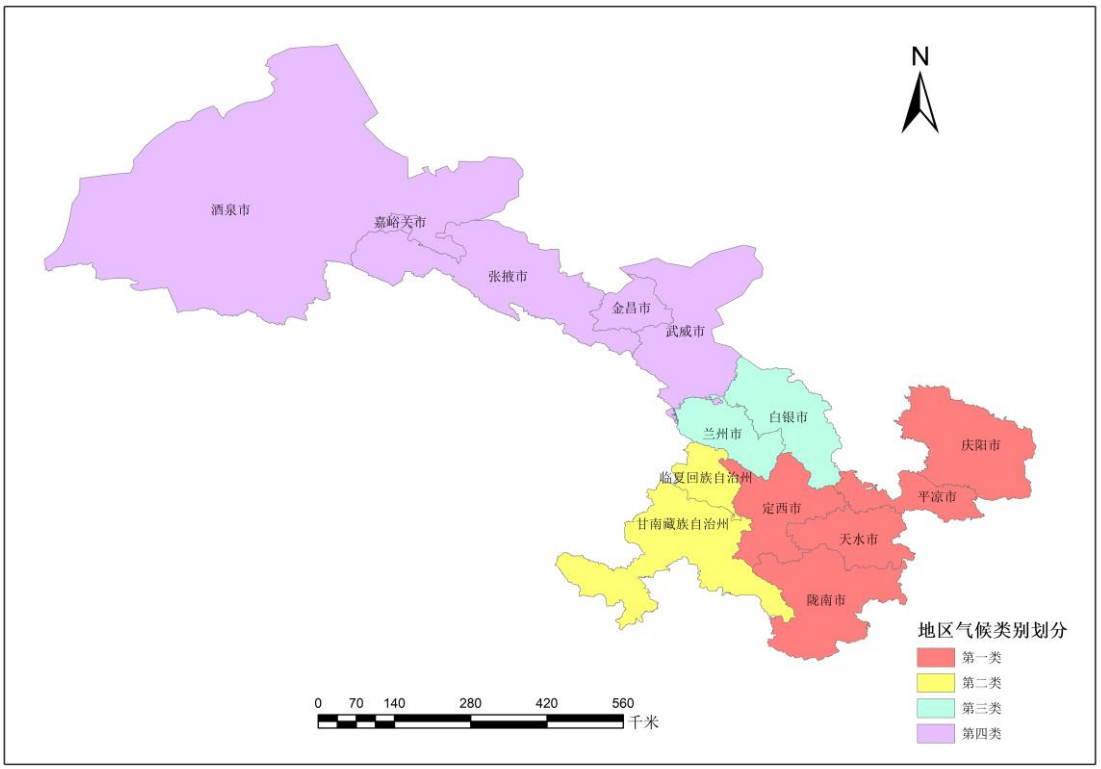


图 4.3　甘肃省 14 个地区聚类结果空间分布图

将图 4.3 中的聚类结果与甘肃省各地区气候类别划分(表 4.4)进行比较，可以看出：聚类结果中的第一类城市陇南市、庆阳市、平凉市、天水市、定西市五个城市，这些城市都位于甘肃省的东部，其中，庆阳市、平凉市和定西市地处黄土高原，这些地区水土流失较为严重，气候类型以温带季风气候为主，具有典型的四季分明特征，而天水市和陇南市位于甘肃东南部，紧邻秦岭山脉，气候湿润，多雨，以亚热带季风气候为主；聚类结果的第二类城市临夏回族自治州和甘南藏族回族自治虽在气候上有所差异，前者属于温带季风季候，后者则是高原高寒气候，但它们的地理位置有着紧密的联系。这两个地区相邻，都位于甘南高原，特别是甘南藏族自治州，它位于我国地形的第一阶梯“青藏高原”，与黄土高原在此交汇。这一特殊的地理位置使得两个地区都呈现出地形崎岖、海拔较高、气候寒冷，以高原山地气候为主。因此，尽管气候类型略有不同，但他们在地理环境和气候特征上的相似性使得它们可以划分为同一类别；聚类结果的第三类城市：兰州市、白银市属于温带大陆性气候(干旱)，两地位于甘肃省的中部地区，属于甘肃省的政治、经济中心，位于黄河上游，地形较为复杂，从这两个地区开始，气温开始向北过渡为温带大陆性气候，这两个城市属于温带大陆性气候和季风气候的过渡区，可划分为一个类别；聚类结果的第四类城市包括武威市、金昌市、张掖市、酒泉市、嘉峪关市，这些城市都位于甘肃省的西北部地区，靠近新疆和内蒙古，属于西北干旱半干旱区，地处内陆，干燥少雨，温差大，是典型的温带大陆性气候(干旱)区域。

通过对聚类结果与甘肃省各地区气候类别划分的比较，不难发现 RMNMFFC-CC 算法在识别城市气候空间布局方面的准确性和有效性，这一算法可用来处理高维大规模的函数型数据，不仅可以为农业、旅游等行业的发展提供有价值的参考，还可以为城市规划、生态保护等领域提供重要的决策支持。

4.6 本章小结

本章提出的 RMNMFFC-CC 是一种基于非负矩阵分解的鲁棒双正交多视角函数型聚类算法，其用于处理高维大规模的函数型数据。该算法结合多视角学习，采用$l_{2,1}$范数，对噪声和异常值具有鲁棒性；引入图正则化，考虑数据的局

部几何特征，集成多视角异构特征；同时采用表示矩阵和基矩阵的正交性在达到最优降维的同时提高聚类性能。在实验数据集上的分析证明了 RMNMFFC-CC 算法的良好性能；最后在甘肃省气象数据上的实例应用表明该算法在实际应用中拥有一定的价值。

5 结论与展望

5.1 结论

现实生活中获取的函数型数据，往往存在一定的噪声和异常值，同时，由于数据收集设备的更新，获取到的函数型数据存在高维且大规模的特点。基于此，本文围绕函数型数据，基于上述两个问题，提出两个多视角函数型聚类算法：

（1）针对函数型数据中的噪声和异常值，构建鲁棒图正则化非负矩阵分解多视角函数型聚类算法(RMNMFFC)，该算法采用$l_{2,1}$范数代替F范数以此提高模型的鲁棒性，缓解噪声和异常值的影响；引入图拉普拉斯正则化项，保持数据集内在的几何结构，进一步提高聚类性能。此外，采用交替迭代方法对目标函数进行优化，并给出模型的迭代更新求解算法，证明算法的收敛性，讨论算法的计算复杂度。在随机模拟数据集和 Growth 数据集上进行模拟实验，表明该方法不仅具有鲁棒性，相较于其它聚类算法还具有更好的聚类精度，将其应用于对北京市空气质量监测站点的空间布局识别，验证了该算法的现实意义。

（2）针对函数型数据高维且大规模的特点，构建鲁棒双正交多视角函数型聚类算法(RMNMFFC-CC)。该算法采用$l_{2,1}$范数，对噪声和异常值具有鲁棒性；引入图正则化，考虑数据的局部几何特征，集成多视角异构特征；同时对 NMF 添加约束条件，利用表示矩阵和基矩阵的正交性在降维同时提高聚类性能。此外，采用交替迭代方法优化，给出了模型的迭代更新求解算法，证明了算法的收敛性。在随机模拟数据集、Growth 数据集以及 TIMIT 语音数据集上的模拟实验表明，该方法在在具有鲁棒性的同时，相较于其它聚类算法具有更好的聚类性能，将其应用于对甘肃省行政区划气象数据的聚类研究，验证了 RMNMFFC-CC 算法的可行性与合理性。

5.2 展望

本文围绕函数型数据并基于非负矩阵分解，提出了两个多视角函数型聚类算法，但现实生活中，在函数型数据的获取时，有时会存在缺失，针对于缺失

函数型数据的相关多视角聚类研究是接下来要研究的一大重点内容。此外，本文提出的两个多视角函数型聚类算法都是基于非负矩阵分解，而在机器学习领域中，除了非负矩阵分解，高斯混合模型、子空间学习、贝叶斯方法等在聚类领域也得到了相关的应用，基于其它方法的多视角函数型聚类算法研究，也是接下来要进一步研究的内容。

参考文献

[1] Ramsay J O. When the data are functions[J]. Psychometrika, 1982,47(4):379-396.

[2] Ramsay J O, Silverman B W. Functional Data Analysis[M]. 1997.

[3] Ramsay J O, Dalzell C J. Some Tools for Functional Data Analysis[J]. Journal of the Royal Statistical Society. Series B (Methodological), 1991,53(3):539-561.

[4] Bouveyron C, Brunet-Saumard C. Model-based clustering of high-dimensional data: A review[J]. Computational Statistics & Data Analysis, 2014,71(1):52-78.

[5] Jacques J, Preda C. Functional data clustering: A survey[J]. Advances in Data Analysis and Classification, 2014,8(3):231-255.

[6] Muslea I, Minton S, Knoblock C A. Active learning with multiple views[J]. Journal Of Artificial Intelligence Research, 2006,27(2006):203-233.

[7] Chang X, Dacheng T, Chao X. A Survey on Multi-view Learning[J]. arXiv.org, 2013,1304:5634.

[8] Fu L, Lin P, Vasilakos A V. An overview of recent multi-view clustering[J]. Neurocomputing, 2020,402:148-161.

[9] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999,401(6755):788-791.

[10]王春杰, 何进荣, 王文发. 基于图的多视角聚类算法综述[J]. 计算机与数字工程, 2022,50(2):229-237.

[11]张二华, 王明合, 唐振民. 加性噪声条件下鲁棒说话人确认[J]. 电子学报, 2019,47(6):1244-1250.

[12]贺超波, 汤庸, 张琼, 等. 基于增量式鲁棒非负矩阵分解的短文本在线聚类[J]. 电子学报, 2019,47(5):1086-1093.

[13]James G M, Sugar C A. Clustering for Sparsely Sampled Functional Data[J]. Journal of the American Statistical Association, 2003,98(462):397-408.

[14]Fabrice R, Villa, Rossi F, et al. Support vector machine for functional data classification[J]. Neurocomputing, 2006,69(7-9):730-742.

[15]Kayano M, Dozono K, Konishi S. Functional Cluster Analysis via Orthonor-

malized Gaussian Basis Expansions and Its Application[J]. Journal of Classification, 2010,27:211-230.

[16]Yamamoto, Michio, Yoshikazu T. Functional factorial K -means analysis[J]. Computational Statistics and Data Analysis, 2014,79(2014):133-148.

[17]王德青, 朱建平, 王洁丹. 基于自适应权重的函数型数据聚类方法研究[J]. 数理统计与管理, 2015,34(1):84-92.

[18]王德青, 刘晓葳, 朱建平. 基于自适应迭代更新的函数型数据聚类方法研究[J]. 统计研究, 2015,32(4):91-96.

[19]王德青, 刘晓葳, 朱建平. 函数型自适应权重聚类分析的再拓展[J]. 数理统计与管理, 2016,35(1):81-88.

[20]梁银双, 刘黎明, 卢媛. 基于函数型数据聚类的京津冀空气污染特征分析[J]. 调研世界, 2017(5):43-48.

[21]黄恒君, 高海燕, 张梦瑶. 函数型聚类分析:基于距离的一步法框架[J]. 数理统计与管理, 2019,38(06):986-995.

[22]高海燕, 黄恒君, 王宇辰. 基于非负矩阵分解的函数型聚类算法[J]. 统计研究, 2020,37(8):91-103.

[23]姚晓红, 黄恒君. 非负半监督函数型聚类方法[J]. 计算机科学与探索, 2021,15(12):2438-2448.

[24]Zhong Q, Lin H, Li Y. Cluster non-Gaussian functional data[J]. Biometrics, 2021,77(3):852-865.

[25]孟银凤, 杨佳宇, 曹付元. 函数型数据的分裂转移式层次聚类算法[J]. 山东大学学报(工学版), 2022,52(01):19-27.

[26]Wang T, Qin L, Dai C, et al. Heterogeneous Learning of Functional Clustering Regression and Application to Chinese Air Pollution Data[J]. Int J Environ Res Public Health, 2023,20(5):4155.

[27]Singhal A, Seborg D E. Clustering multivariate time-series data[J]. Journal of Chemometrics, 2005,19(8):427-438.

[28]任娟. 多指标面板数据聚类方法及其应用[J]. 统计与决策, 2012(04):92-95.

[29]Julien, Jacques, Cristian P. Model-based clustering for multivariate functional

data[J]. Computational Statistics and Data Analysis, 2014,71:92-106.

[30]Julien, Jacques, Cristian P. Funclust: A curves clustering method using functional random variables density approximation[J]. Neurocomputing, 2013,112(7):164-171.

[31]Schmutz A, Jacques J, Bouveyron C, et al. Clustering multivariate functional data in group-specific functional subspaces[J]. Computational statistics, 2020,35(3):1101-1131.

[32]Ieva F, Paganoni A M, Pigoli D, et al. Multivariate Functional Clustering for the Morphological Analysis of Electrocardiograph Curves[J]. Journal of the Royal Statal Society, 2013,62(3):401-418.

[33]Yamamoto M, Hwang H. Dimension-Reduced Clustering of Functional Data via Subspace Mixing Two Stages Dimension Reduction and Nonparametric Approaches[J]. Computational Statistics, 2017,2(34):631-652.

[34]Misumi T, Matsui H, Konishi S. Multivariate functional clustering and its application to typhoon data[J]. Behaviormetrika, 2019,46(1):163-175.

[35]姚晓红. 基于多视角学习的若干多元函数型聚类方法研究[D]. 兰州财经大学, 2022.

[36]Wang S, Zhu W. Sparse Graph Embedding Unsupervised Feature Selection[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018,48(3):329-341.

[37]Tang J, Wan Z. Orthogonal Dual Graph-Regularized Nonnegative Matrix Factorization for Co-Clustering[J]. Journal of scientific computing, 2021,87(3):61-66.

[38]张骏. 基于类别信息和稀疏表示的非负矩阵分解[J]. 哈尔滨商业大学学报（自然科学版）, 2017,33(5):607-610.

[39]Wang K, Liao R, Yang L J, et al. Nonnegative Matrix Factorization Aided Principal Component Analysis for High-Resolution Partial Discharge Image Compression in Transformers[J]. International review of electrical engineering, 2013,8(1):479-490.

[40]Li H, Li K, An J, et al. An efficient manifold regularized sparse non-negative matrix factorization model for large-scale recommender systems on GPUs[J]. Information Sciences, 2019,496:464-484.

[41]Malik S, Bansal P. Matrix Factorization-based Improved Classification of Gene Expression Data[J]. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 2020,13(5):858-863.

[42]Cai D, He X, Han J, et al. Graph Regularized Nonnegative Matrix Factorization for Data Representation.[J]. IEEE transactions on pattern analysis and machine intelligence, 2011,33(8):1548-1560.

[43]Li X, CUI G, DONG Y. Graph regularized non-negative low-rank matrix factorization for image clustering[J]. IEEE transactions on cybernetics, 2017,47(11):3840-3853.

[44]Meng Y, Shang R, Shang F, et al. Semi-Supervised Graph Regularized Deep NMF With Bi-Orthogonal Constraints for Data Representation[J]. IEEE transaction on neural networks and learning systems, 2020,31(9):3245-3258.

[45]Liang N, Yang Z, Li Z, et al. Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints[J]. Knowledge-based systems, 2020,194(prepublish):105582.

[46]Li S T, Li W G, Hu J W, et al. Semi-supervised bi-orthogonal constraints dual-graph regularized NMF for subspace clustering[J]. Applied Intelligence, 2021,52(3):3227-3248.

[47]陈献, 胡丽莹, 林晓炜, 等. 基于核非负矩阵分解的有向图聚类算法[J]. 计算机应用, 2021,41(12):3447-3454.

[48]李向利, 范学珍, 逯喜燕. 基于非负矩阵分解的修正模糊聚类算法[J]. 吉林大学学报(理学版), 2022,60(06):1416-1422.

[49]李向利, 毕胜, 王佩源. 层次预处理的非负矩阵分解加权集成聚类算法[J]. 重庆师范大学学报(自然科学版), 2023,40(05):136-144.

[50]黄路路, 唐舒宇, 张伟, 等. 基于Lp范数的非负矩阵分解并行优化算法[J]. 计算机科学, 2024,51(02):100-106.

[51]Balcan M F, Blum A, Yang K. Co-training and Expansion: Towards Bridging Theory and Practice[J]. Advances in neural information processing systems, 2004,17:89-96.

[52]Sonnenburg S, Rätsch G, Schäfer C. A General and Efficient Multiple Kernel Learning Algorithm[J]. Advances in neural information processing systems, 2005,18:1273-1280.

[53]Lai Z H, Mo D M, Wen J J, et al. Generalized Robust Regression for Jointly Sparse Subspace Learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019,29(3):756-772.

[54]Yang Y, Wang H. Multi-view clustering: A survey[J]. Big Data Mining and Analytics, 2018,1(2):83-107.

[55]Liu X, Dou Y, Yin J, et al. Multiple Kernel k-Means Clustering with Matrix-Induced Regularization[J]. Proceedings of the . AAAI Conference on Artificial Intelligence, 2016,30(1):1888-1894.

[56]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):1724-1736.

[57]Wang H, Yang Y, Liu B. GMC: Graph-Based Multi-View Clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2020,32(6):1116-1129.

[58]林燕铭, 陈晓云. 流形正则引导的自适应加权多视角子空间聚类[J]. 模式识别与人工智能, 2022,35(11):965-976.

[59]王丽娟, 邢津萍, 尹明, 等. 基于一致性图的权重自适应多视角谱聚类算法[J]. 计算机工程, 2024,50(02):122-131.

[60]Xie Q, Zhu H, Liu W. Image Retrieval Based on Multiview Constrained Nonnegative Matrix Factorization and Gaussian Mixture Model Spectral Clustering Method[J]. Mathematical problems in engineering, 2016,2016:1-15.

[61]刘正, 张国印, 陈志远. 基于特征加权和非负矩阵分解的多视角聚类算法[J]. 电子学报, 2016,44(3):535-540.

[62]宗林林, 张宪超, 赵乾利, 等. 一种多流形正则化的多视图非负矩阵分解算法

[J]. 南京大学学报(自然科学), 2017,53(03):557-568.

[63] Mekthanavanh V, Li T, Meng H, et al. Social web video clustering based on multi-view clustering via nonnegative matrix factorization[J]. International journal of machine learning and cybernetics, 2019,10(10):2779-2790.

[64] 李骜, 冯聪, 牛宇童, 等. 面向视角非对齐数据的多视角聚类方法[J]. 通信学报, 2022,43(7):143-152.

[65] 郝敬琪, 胡立华, 张素兰, 等. 基于非负矩阵分解的均方残差多视图聚类算法[J]. 计算机技术与发展, 2023,33(12):65-71.

[66] 林虹燕, 杜元花, 周楠, 等. 基于多视角自适应图正则的非负矩阵分解聚类[J]. 成都信息工程大学学报, 2023,38(05):526-534.

[67] Wang S, Chen L, Sun Y, et al. Multiview nonnegative matrix factorization with dual HSIC constraints for clustering[J]. International journal of machine learning and cybernetics, 2023,14(6):2007-2022.

[68] 姚晓红, 高海燕, 吕家奇, 等. 一种基于多视角学习的多元函数型聚类方法[J]. 数理统计与管理, 2022,41(4):689-702.

[69] 王伟祥, 孙广磊. 求解非光滑全局优化问题的单参数填充函数算法[J]. 上海第二工业大学学报, 2022,39(03):251-255.

[70] Guan N, Liu T, Zhang Y, et al. Truncated Cauchy Non-Negative Matrix Factorization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019,41(1):246-259.

[71] Peng S, Ser W, Chen B, et al. Robust semi-supervised nonnegative matrix factorization for image clustering[J]. Pattern Recognition, 2021,111:107683.

[72] 高海燕, 刘万金, 黄恒君. 鲁棒自适应对称非负矩阵分解聚类算法[J]. 计算机应用研究, 2023,40(4):1024-1029.

[73] 余沁茹, 卢桂馥, 李华. 自适应图正则化的低秩非负矩阵分解算法[J]. 智能系统学报, 2022,17(2):325-332.

[74] Belhumeur P N, Hespanha J P, Kriegman D J. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997,19(7):711-720.

[75]葛红妨, 沈雷. 基于主特征ICA的欺骗干扰检测识别算法[J]. 杭州电子科技大学学报, 2022,42(4):19-26.

[76]Ding C, Li T, Peng W. Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering[J]. Information Processing & Management, 2006,46(5):126-135.

[77]Wang J, Zhang P, Luo L. Nonnegative Component Representation with Hierarchical Dictionary Learning Strategy for Action Recognition[J]. IEICE Transactions on Information and Systems, 2016,99(4):1259-1263.

[78]Wang J, Tian F, Yu H. Diverse Nonnegative Matrix Factorization for Multi-view Data Representation[J]. IEEE Tranctions on Cybernetics, 2018,48(9):2620-2632.

附录

附录一：基于图正则化非负矩阵分解的鲁棒多视角函数型聚类算法主程序代码

```
code<- function(Y, Phi, U, V, G,  alpha,W,lambda,D)
{
  U <-
 U * ((t(Phi) %*% G %*% Y%*% V)/(t(Phi) %*% G %*% Phi %*% U %*% t(V) %*
% V + alpha *U))
  V <-
 V * ((t(Y)%*% G %*% Phi %*% U + lambda * W %*% V)/(V %*% t(U) %*% t(Ph
i) %*% G %*% Phi %*% U + lambda * D %*% V))
  return(list(U = U, V = V))
}
accMtr <- matrix(0, nrow = ntest, ncol = 3)
colnames(accMtr) <- c("purM","accM","riM")
class(lst_Y[[1]])
for(kk in 1:ntest)
{
  print(paste0("Round ",kk))
  lst_Y <- list(CO,no2,o3,pm10,so2,pm2.5)
  lst_X <- vector("list", nview)
  lst_Knot <- vector("list", nview)
  lst_Phi <- vector("list", nview)
  lst_mtrCoef <- vector("list", nview)
  for(i in 1:nview)
  {
    x <- 1:nrow(lst_Y[[i]])
    knots <- seq(range(x)[1], range(x)[2], length.out = n_knot[i])[-
c(1,n_knot[i])]
    mtr_ds <- bs(x, knots= knots, intercept = TRUE)
    lst_X[[i]] <-  x
    lst_Knot[[i]] <- knots
    lst_Phi[[i]] <- mtr_ds
    lst_mtrCoef[[i]] <- solve(t(lst_Phi[[i]]) %*% lst_Phi[[i]]+1e-
2) %*% t(lst_Phi[[i]]) %*% as.matrix(lst_Y[[i]])
  }
  lst_U <- vector("list", nview)
  for(i in 1:nview)
```

```
30.  {
31.    U <-
 matrix(runif(ncol(lst_Phi[[i]])*K,0,1), nrow = ncol(lst_Phi[[i]]), nco
l = K)
32.    lst_U[[i]] <- U
33.  }
34.  lst_V <- vector("list", nview)
35.  for(i in 1:nview)
36.  {
37.    ini_KM <- kmeans(t(lst_mtrCoef[[i]]), K)
38.    V <- matrix(0, nrow = ncol(lst_Y[[i]]), ncol = K)
39.    for(j in 1:ncol(V))
40.    {
41.      V[ini_KM$cluster==j, j] <- 1
42.    }
43.    lst_V[[i]] <- V %*% solve(sqrt(diag(apply(V, 2, sum))))
44.  }
45.  D<-list()
46.  W<-list()
47.  L<-list()
48.  DD<-list()
49.  for (i in 1:nview)
50.  {
51.    DD[[i]] <- matrix(0,ncol(lst_Y[[i]]),ncol(lst_Y[[i]]))
52.    for (ii in 1:ncol(lst_Y[[i]]))
53.    {
54.      for (j in 1:ncol(lst_Y[[i]]))
55.      {
56.        DD[[i]][ii,j] <- round(sqrt(sum((lst_Y[[i]][,ii] -
 lst_Y[[i]][,j])^2)),2)
57.      }
58.    }
59.    for (ij in 1: nrow(DD[[i]]))
60.    {
61.      DD[[i]][ij,order(DD[[i]][ij,])[1:kk+1]] = 1
62.      DD[[i]][ij,order(DD[[i]][ij,])[kk+2:ncol(DD[[i]])]] = 0
63.    }
64.    W[[i]] <- DD[[i]]
65.    W[[i]] <- (W[[i]] + t(W[[i]]))/2
66.    D[[i]] <- diag(apply(W[[i]],2,sum))
67.    L[[i]] <- D[[i]] - W[[i]]
68.  }
69.  for(j in 1:maxiter)
70.  {
```

```
    lst_G <- vector("list", nview)
    for(i in 1:nview)
    {
      lst_G[[i]] <- diag(apply((lst_Y[[i]] -
 lst_Phi[[i]] %*% lst_U[[i]] %*% t(lst_V[[i]]))^2,1,sum))
    }
    for(i in 1:nview)
    {
      res <-
 RRMVFC(lst_Y[[i]], lst_Phi[[i]], lst_U[[i]], lst_V[[i]],lst_G[[i]], al
pha[i],W[[i]],lambda[i],D[[i]])
      lst_U[[i]] <- res$U
      lst_V[[i]] <- res$V
    }
  }
  cl <- kmeans(V, K)
  lbl<-c(rep(1,39),rep(2,54))
  accMtr[kk, 1] <- com_accuracy(cl$cluster, lbl)
  accMtr[kk, 2] <- com_accuracy(cl$cluster, lbl, method = 1)
  accMtr[kk, 3] <- com_accuracy(cl$cluster, lbl, method = 4)
}
round(colMeans(accMtr),4)
round(apply(accMtr,2,sd),4)
```

附录二：基于非负矩阵分解的鲁棒双正交多视角函数型聚类算法主程序代码

```
code <- function(Y, Phi, U, V, G,  alpha,W,lambda,mu,D)
{
  A<-matrix(1, nrow(U), ncol(U))
  U <-
 U * ((t(Phi) %*% G %*% Y%*% V + alpha * U)/(t(Phi) %*% G %*% Phi %*% U
 %*% t(V) %*% V + alpha * U %*% t(A) %*% A))
  V <-
 V * ((t(Y)%*% G %*% Phi %*% U + lambda *  W %*% V + V)/(V %*% t(U) %*%
 t(Phi) %*% G %*% Phi %*% U + lambda * W %*% V +  V %*% t(V) %*% V))
  return(list(U = U, V = V))
}
require(splines)
require(lava)
source("聚类评价指标.r")
nview <- 256
K <- 5
alpha <- 1/nview
lambda <- 100
maxiter <- 100
n_knot <- c(20,20)
ntest <- 5
accMtr <- matrix(0, nrow = ntest, ncol = 3)
colnames(accMtr) <- c("purM","accM","riM")
class(lst_Y[[1]])
data<-as.matrix(data)
gnd<- read.csv("C:/Users/apple/Desktop/gnd.csv")
for(kk in 1:ntest)
{
  print(paste0("Round ",kk))
  lst_Y <- list(data)
  lbl<- lst_Y$gnd
  lst_X <- vector("list", nview)
  lst_Knot <- vector("list", nview)
  lst_Phi <- vector("list", nview)
  lst_mtrCoef <- vector("list", nview)
  for(i in 1:nview)
  {
    x <- 1:nrow(lst_Y[[i]])
```

```
    knots <- seq(range(x)[1], range(x)[2])
    mtr_ds <- bs(x, knots= knots, intercept = TRUE)
    lst_X[[i]] <-  x
    lst_Knot[[i]] <- knots
    lst_Phi[[i]] <- mtr_ds
    lst_mtrCoef[[i]] <- solve(t(lst_Phi[[i]]) %*% lst_Phi[[i]]+ 1e-3) %*% t(lst_Phi[[i]]) %*% as.matrix(lst_Y[[i]])
  }
  lst_U <- vector("list", nview)
  for(i in 1:nview)
  {
    U <- matrix(runif(ncol(lst_Phi[[i]])*K,0,1), nrow = ncol(lst_Phi[[i]]), ncol = K)
    lst_U[[i]] <- U
  }
  lst_V <- vector("list", nview)
  for(i in 1:nview)
  {
    ini_KM <- kmeans(t(lst_mtrCoef[[i]]), K)
    V <- matrix(0, nrow = ncol(lst_Y[[i]]), ncol = K)
    for(j in 1:ncol(V))
    {
      V[ini_KM$cluster==j, j] <- 1
    }
    lst_V[[i]] <- V %*% solve(sqrt(diag(apply(V, 2, sum))))
  }
  D<-list()
  W<-list()
  L<-list()
  DD<-list()
  for (i in 1:nview)
  {
    DD[[i]] <- matrix(0,ncol(lst_Y[[i]]),ncol(lst_Y[[i]]))
    for (ii in 1:ncol(lst_Y[[i]]))
    {
      for (j in 1:ncol(lst_Y[[i]]))
      {
        DD[[i]][ii,j] <- round(sqrt(sum((lst_Y[[i]][,ii] - lst_Y[[i]][,j])^2)),2)
      }
    }
    for (ij in 1: nrow(DD[[i]]))
    {
```

```
      DD[[i]][ij,order(DD[[i]][ij,])[1:kk+1]] = 1
      DD[[i]][ij,order(DD[[i]][ij,])[kk+2:ncol(DD[[i]])]] = 0
    }
    W[[i]] <- DD[[i]]
    W[[i]] <- (W[[i]] + t(W[[i]]))/2
    D[[i]] <- diag(apply(W[[i]],2,sum))
    L[[i]] <- D[[i]] - W[[i]]
  }
  for(j in 1:maxiter)
  {
    lst_G <- vector("list", nview)
    for(i in 1:nview)
    {
      lst_G[[i]] <- diag(apply((lst_Y[[i]] -
 lst_Phi[[i]] %*% lst_U[[i]] %*% t(lst_V[[i]]))^2,1,sum))
    }
    for(i in 1:nview)
    {
      res <-
 code(lst_Y[[i]], lst_Phi[[i]], lst_U[[i]], lst_V[[i]],lst_G[[i]], alph
a[i],W[[i]],lambda[i],D[[i]])
      lst_U[[i]] <- res$U
      lst_V[[i]] <- res$V
    }
  }
  cl <- kmeans(V, K)
  lbl<-c(rep(1,1),rep(2,5),rep(3,7),rep(4,1))
  accMtr[kk, 1] <- com_accuracy(cl$cluster, lbl)
  accMtr[kk, 2] <- com_accuracy(cl$cluster, lbl, method = 1)
  accMtr[kk, 3] <- com_accuracy(cl$cluster, lbl, method = 4)
}
round(colMeans(accMtr),4)
round(apply(accMtr,2,sd),4)
```

攻读硕士学位期间承担的科研任务及主要成果

发表论文：

[1]程莞莞,赵芳芳.基于函数型聚类的重庆市主城区PM_(2.5)防治区域划分[J].甘肃科技纵横,2024,53(02):39-46.

参与科研项目：

甘肃教育科技创新项目：大数据背景下基于非负矩阵分解的多视角聚类方法及应用研究(2020A-059)，2020.6—2022.9，已结项。

竞赛获奖：

“空气污染物防治区域划分研究——以重庆市 $PM_{2.5}$ 为例”荣获 2022 年（第八届）全国大学生统计建模大赛省级三等奖，2022 年 8 月。

“函数型数据视角下我国 GDP 的聚类分析及预测研究”荣获第五届全国应用统计专业学位研究生案例大赛全国三等奖，2022 年 8 月。

致 谢

行文至此，落笔为终。这一年我二十五岁，学生生涯也就此告一段落，光阴似箭，日月如梭，三年的硕士研究生求学即将结束。回首过去，每一步都有迹可循，走的每一步都显示着未来方向。

盛行千里，感恩总总。

感谢我的导师高海燕老师。在读研的三年中，给予了我许多学术上的指导和生活上的帮助。在毕业论文的选题与撰写中，耐心地给予了许多建议和指导，从而使我的论文得以顺利完成；读研期间遇到困难时，老师都以鼓励为主，总能让我重拾希望，继续努力；疫情封校期间，老师也不忘学生，给予了许多温暖和帮助。感谢老师让我在这三年异地求学的旅途中感受到真诚与温暖，愿老师桃李满园，学术长青。

感谢我的父母。我生于农村长于农村，生活在一个普通又温暖的家庭，父母虽没有富足的精神世界富裕的物质生活，但仍然将我成功的送入大学，并不断深造。他们用朴素又真挚的爱陪我走过人生的每一个阶段，感谢他们，供我上学，靠他们勤劳的双手和伟大的爱让我无后顾之忧的完成学业；感谢他们，让我在健康、快乐、幸福的家庭中成长；感谢他们，让我站在他们的肩膀上看到了更大的世界，拥有了更广阔的天空。

感谢我的朋友们。三年前，即将进入陌生城市与未知环境的抗拒与迷茫，来到学校的第一个星期，面对简陋的宿舍环境及对新环境的不适，感恩有你们，让我回忆起三年的研究生生活，满是笑容与欢乐、感动与美好。很幸运一路走来，在每个阶段都遇到美好、真诚的人，我们见证彼此的青葱岁月。不管今后是继续陪伴还是各奔东西，都要感谢你们温暖着我的每一刻时光。

道阻且长，行则将至。感谢那个缓慢前行但未曾放弃的自己。一路跌跌撞撞至今，无论是“孔乙己的长衫”还是“范进中举”，我都见到了更多的沿途风景，也意识到了自己的渺小。我的这一路虽不优秀，但从未放弃。未来山高水远，请继续前行吧。愿未来仍能保持初心，仍不轻易放弃，始终勇敢真诚。

凡是过往，皆为序章。怀着努力与真诚、勇气与信心，奔赴下一篇山海。谨以此文敬我的青春，我们在枝繁叶茂时再见，在理想的远方见。毕业快乐！