分类号 O212/29　　密级 _____
U D C _____　　编号 10741

硕 士 学 位 论 文

论文题目 基于对称非负矩阵分解的鲁棒聚类算法研究

研 究 生 姓 名：刘万金

指导教师姓名、职称：高海燕　教授

学 科、专 业 名 称：统计学　数理统计学

研 究 方 向：复杂数据分析

提 交 日 期：2023 年 5 月 30 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：刘万金 签字日期：2023.5.30

导师签名：高海燕 签字日期：2023.5.30

关于论文使用授权的说明

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

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

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

学位论文作者签名：刘万金 签字日期：2023.5.30

导师签名：高海燕 签字日期：2023.5.30

Research on Robust Clustering Algorithm Based on Symmetric Nonnegative Matrix Factorization

Candidate : Wanjin Liu

Supervisor: Haiyan Gao

摘 要

对称非负矩阵分解 SNMF(Symmetric Nonnegative Matrix Factorization)作为一种基于图的聚类算法，能够更自然地捕获图表示中嵌入的聚类结构，并且在线性和非线性数据上获得更好的聚类结果，但对变量的初始化比较敏感。另外，标准的 SNMF 算法利用误差平方和衡量分解的质量，对噪声和异常值敏感。为了解决这些问题，在集成学习视角下，提出一种鲁棒自适应对称非负矩阵分解聚类算法 RS^{3}NMF (Robust Self-adaptived Symmetric Nonnegative Matrix Factorization)。进一步，结合训练集的标签信息增强投影矩阵的判别能力，将鲁棒性、自适应学习和标签信息集成到 SNMF 框架中，提出一种鲁棒自适应学习判别对称非负矩阵分解算法(Robust Adaptive Learning Discriminative Symmetric Nonnegative Matrix Factorization Algorithm，RADS^{3}NMF)。本文主要研究内容包括以下两部分：

（1）受鲁棒非负矩阵分解、自适应方法和集成学习的启发，建立鲁棒自适应对称非负矩阵分解聚类算法(RS^{3}NMF)，该算法将鲁棒性融入 SNMF 框架。基于$L_{2,1}$范数的 RS^{3}NMF 模型缓解了噪声和异常值的影响，保持了特征旋转不变性，提高了模型的鲁棒性。同时，在不借助任何附加信息的前提下，利用 SNMF 对初始化特征的敏感性逐步增强聚类性能。采用交替迭代方法优化，并保证目标函数值的收敛性。大量实验结果显示，所提 RS^{3}NMF 算法优于其它先进的算法，具有较强的鲁棒性。此外，对我国 31 个省市 GDP 数据进行实例应用，结果表明该鲁棒聚类算法对 GDP 数据的划分能够判断各省之间的发展差异，具有良好的实际应用价值。

（2）受空间聚类自表述学习方法的启发，通过引入$L_{2,1}$范数、自适应学习和标签信息，建立鲁棒自适应学习判别对称非负矩阵分解算法(RADS^{3}NMF)。具体地，首先由获得的自表示系数表示亲和矩阵，并利用训练集的标签信息增强投影矩阵的判别能力；其次对建立的模型进行优化求解，构造辅助函数，证明模型的收敛性，以及给出模型的算法复杂度；最后利用某一时间段北京市二氧化氮(NO_2)污染物小时浓度数据，将该算法应用于北京市空气质量监测站点布设聚类分析，结果显示 RADS^{3}NMF 算法能够较好地识别空气质量监测站点的空间布局，具有良好的适用性。

关键词：对称非负矩阵分解 鲁棒性 聚类算法

Abstract

As a graph-based clustering algorithm, symmetric nonnegative matrix factorization (SNMF) can capture the clustering structure embedded in graph representation more naturally, and get better clustering results on linear and nonlinear data, but it is sensitive to the initialization of variables. In addition, the standard SNMF algorithm uses the sum of squares of errors to measure the quality of decomposition, which is sensitive to noise and outliers. In order to solve these problems, a robust adaptive symmetric nonnegative matrix factorization clustering algorithm (RS^{3}NMF) is proposed from the perspective of ensemble learning. Furthermore, the discriminant ability of projection matrix is enhanced by combining the label information of training set, and a robust adaptive learning discriminant symmetric nonnegative matrix decomposition algorithm (RADS^{3}NMF) is proposed by integrating robustness、adaptive graph learning and label information into SNMF framework. The main research contents of this paper include the following two parts:

Inspired by robust nonnegative matrix factorization, adaptive methods and ensemble learning, a robust adaptive symmetric nonnegative matrix factorization clustering algorithm (RS^{3}NMF) is constructed, which integrates robustness into the SNMF framework. The $L_{2,1}$ norm-based RS^{3}NMF model alleviates the influence of noise and outliers, maintains the invariance of feature rotation and improves the robustness of the model. At the same time, without any additional information, the clustering performance is gradually enhanced by

using the sensitivity of SNMF to initialization features. The alternating iteration method is used to optimize and ensure the convergence of the objective function value. A large number of experimental results show that the proposed RS^{3}NMF algorithm is superior to other advanced algorithms and has strong robustness. In addition, the application of GDP data of 31 provinces and cities in China shows that the robust clustering algorithm can judge the development differences among provinces and has good practical application value.

Inspired by the spatial clustering self-expression learning method, a robust adaptive learning discriminant symmetric nonnegative matrix factorization algorithm (RADS^{3}NMF) is constructed by introducing $L_{2,1}$ norm、adaptive learning and label information. Specifically, firstly, the affinity matrix is represented by the obtained self-representation coefficient, and the discrimination ability of the projection matrix is enhanced by using the label information of the training set; Secondly, the model is optimized, the auxiliary function is constructed, the convergence of the model is proved, and the algorithm complexity of the model is given. Finally, using the hourly concentration data of nitrogen dioxide (NO_2) pollutants in Beijing in a certain period of time, the algorithm is applied to the cluster analysis of air quality monitoring stations in Beijing. The results show that RADS^{3}NMF algorithm can better identify the spatial layout of air quality monitoring stations and has good applicability.

Keywords： Symmetric nonnegative matrix factorization; Robustness; Cluster

algorithm

目 录

1 引言

1.1 研究背景

数据作为一种新型生产要素，在信息传递、科学决策、趋势预测以及报告分析中扮演着重要的角色。现实生活中，我们遇到的数据普遍存在维数高、蕴含信息量大的特点。例如一张彩色图片可以看作一个高维数据矩阵，其中图片像素的大小代表矩阵的维数。数据处理已经涉及各个领域，如人脸识别[1]、文本分析[2]、模式识别[3]以及统计分析[4]等。高维数据虽然蕴含了大量的信息，但数据处理复杂，处理的时效性难以保证；同时，数据中存在的大量噪声、异常值导致实验结果存在一定的偏差。为了挖掘数据中存在的有效信息以及降低数据处理的复杂度，通常需要对数据矩阵进行降维。目前常用的数据降维技术有主成分分析(PCA)[5]、线性判别分析(LDA)[6]、独立分量分析(ICA)[7]、奇异值分解(SVD)[8]等，这些方法只适用于小规模数据。此外，随着机器学习的发展，尽管矩阵分解在大规模数据降维中具有一定的优越性，但降维之后的矩阵中可能存在负元素，不便于对其解释分析[9]，非负矩阵分解(Nonnegative Matrix Factorization，NMF) [10]是一种旨在克服这些挑战的方法。

非负矩阵分解作为一种数据表示和聚类算法已被广泛应用。它将原始数据矩阵中呈现的样本或特征表示为基向量的线性组合。同时，通过相应的系数矩阵将隶属标签分配给每个样本或特征。当数据分布具有线性结构时，NMF 能够获得良好的聚类性能。然而，NMF 不能利用输入数据的非线性结构产生聚类结果。对称非负矩阵分解(Symmetric Non-negative Matrix Factorization，SNMF) [19]是一类特殊的约束 NMF，它将记录样本成对相似性的亲和矩阵分解成聚类指示矩阵及其转置的乘积，在线性和非线性流形上获得更好的聚类结果，但对变量的初始化比较敏感。在实际应用中，标准 NMF、SNMF 中的 Frobenius 范数利用最小二乘误差函数来计算原始数据点和预测数据点之间的差异，由于数据往往受到噪声和异常值的影响，导致模型对噪声和异常值非常敏感，降低了算法的鲁棒性。因此，有必要开发更鲁棒的损失函数。

机器学习中，鲁棒性主要用于检验模型在面对输入数据的微小变动时是否依然能保持判断的准确性，即模型面对一定变化时的表现是否稳定。在模型理论分析中，通常从误差的角度出发考虑模型的鲁棒性，也可以理解为对于带有噪声或异常值的数据，鲁棒

的模型能够取得较好的评价指标。非负矩阵分解中常见的提高模型鲁棒性的方法有基于$L_{2,1}$范数、$L_{2,p}$范数($0 < p \leq 1$)和相关熵等构造目标函数。而基于$L_{2,1}$范数的目标函数采用非平方欧氏距离，能够有效降低噪声与异常值对模型的影响，从而提升模型的鲁棒性。进一步，可结合自适应学习、标签的判别信息增强聚类性能。

鉴于现有的方法没有同时考虑鲁棒性、SNMF 初始化的敏感性等，受鲁棒非负矩阵分解、自适应方法、集成学习、空间聚类自表述学习方法等的启发，本文基于对称非负矩阵分解，利用$L_{2,1}$范数重构模型误差，结合自适应学习和标签信息，缓解噪声和异常值问题，提升模型的鲁棒性，增强聚类性能。

1.2 国内外研究现状

(1) 非负矩阵分解研究现状

Lee 等[10]提出了非负矩阵分解(NMF)，它作为一种流行的数据表示方法和聚类技术[11]被广泛应用于数据挖掘[12,13]和机器学习[14,15]。Cai 等[16]将原始数据矩阵分解为两个低秩非负矩阵的乘积，采用重构误差的平方，即 Frobenius 范数定义目标损失函数。由于非负矩阵的优良特性，基于非负矩阵分解的新的方法不断被提出，例如 zong 等[17]提出了多视角正则化非负矩阵分解；高海燕等[18]提出了基于非负矩阵分解的函数型聚类算法。当数据分布具有线性结构时，NMF 能够获得良好的聚类性能。由于 NMF 中每个数据点进入目标函数时都带有平方残差，因此很容易出现少数误差较大的异常值支配目标函数。

(2) 对称非负矩阵分解

标准 NMF 不能利用输入数据的非线性结构产生聚类结果[16]。Kuang 等[19]提出了对称非负矩阵分解算法(SNMF)，它是 NMF 的一种特殊情况，作为图聚类的通用框架，它继承了 NMF 的优点，使聚类划分矩阵具有非负性。然而，与 NMF 不同的是，SNMF 基于数据点之间的相似性度量，并分解包含两两相似值的对称矩阵。Ding 等[11]发现 SNMF 与谱聚类(Spectral Clustering，SC)具有相同的目标函数，约束条件不同，SC 寻求正交分解，而 SNMF 学习非负嵌入。因此，SNMF 可看作一种图聚类算法，可分非线性数据而且直接生成聚类指标，通常比 SC 表现的更好[20]。近年来，一些学者已经提出了一些改进的 SNMF 算法。He 等 [21]开发了有效的流形 SNMF 算法，并将其应用于概率聚类；Zhang 等[22]构建了一种用于多视角聚类的图正则化 SNMF 框架(GSNMF)；Gao 等

[23]提出了一种图正则化 SNMF(GrSymNMF)来提高其在图聚类中的性能；Jia 等[24]提出从样本数据中自适应学习亲和矩阵用于 SNMF 算法。此外，SNMF 也被推广应用于半加权学习[25]、多任务聚类问题[26]等。SNMF 在数学上被表述为一个非凸优化问题，对变量的初始化很敏感，而初始化矩阵的好坏将严重影响其聚类性能。为解决此问题，Jia 等[27]充分考虑 SNMF 对初始化的敏感性，提出了自适应加权 SNMF 算法(S^3NMF)，但 S^3NMF 未考虑到噪声与异常值的影响。

(3) 鲁棒非负矩阵分解

在许多实际应用中，数据可能含有噪声或异常值，由于平方距离，异常值对标准的 Frobenius 范数损失函数的 NMF 模型会产生较大的影响。因此，鲁棒算法相继被提出。Kong 等[28]提出使用$L_{2,1}$范数的鲁棒非负矩阵分解算法；Huang(2014)等[29]提出鲁棒的流形非负矩阵分解算法，该方法利用$L_{2,1}$范数重构误差，并使系数矩阵正交；吴月等[30]考虑噪声，提出强鲁棒性稀疏非负矩阵分解算法；蒋茂松等[31]对系数矩阵进行约束，提出稀疏正则非负矩阵分解的语音增强算法；Wu 等[32]考虑数据的几何结构，提出鲁棒流形非负矩阵聚类算法；刘国庆等[33]通过系数约束与正则化，提出稀疏图正则化的非负低秩矩阵分解算法，增加了模型的鲁棒性与可解释性；李华等[34]同时考虑系数低秩约束与数据的局部几何结构，提出基于干净数据的流形正则化非负矩阵分解算法，增加了模型的鲁棒性；董文婷等[35]通过构建邻近图和最大熵图描述数据的局部结构和非局部结构，并使用$L_{2,1}$范数增加模型的鲁棒性；Liu 等[36]考虑原始数据的低秩结构和几何信息，将系数矩阵约束为非负，提出基于$L_{2,1}$范数的鲁棒半非负低秩图嵌入算法。

综上所述，通过梳理国内外相关研究文献，发现学者们对提高模型的鲁棒性做了大量研究，取得了一些有意义的成果，可为论文后续的研究提供参考。但仍存在一些值得进一步探究的问题。

第一，$L_{2,1}$范数一定程度上能够提升模型的鲁棒性。就自适应对称非负矩阵而言，采用 Frobenius 范数重构误差，由于标准的 Frobenius 范数采用欧氏距离，导致异常值对模型产生较大的影响，模型相对不稳定。$L_{2,1}$范数采用非平方距离度量误差，因此，利用$L_{2,1}$范数重构误差，使模型对噪声和异常值不敏感，以确保鲁棒性。

第二，理论研究表明自适应学习、判别信息有助于提升聚类算法性能。因此，在考虑模型鲁棒性的基础上，进一步综合考虑自适应学习和判别信息，增强聚类性能。

1.3 研究意义与目的

相比传统模型，鲁棒模型能够降低异常值与噪声对模型的影响，使得模型更稳健。如何提高模型的鲁棒性、运算效率一直是学者们关注的焦点，也是目前机器学习研究的热点问题之一。因此研究鲁棒的对称非负矩阵分解具有重要的理论意义与现实意义。①理论意义：鲁棒算法能够缓解噪声与异常值对模型的影响，综合考虑鲁棒性、自适应学习、判别信息等相结合的鲁棒聚类算法能够确保聚类模型的精确性。②现实意义：通过将提出的鲁棒自适应对称非负矩阵分解聚类算法(RS^{3}NMF)应用于我国 31 个省市 GDP 数据聚类，并将鲁棒自适应学习判别对称非负矩阵分解算法(RADS^{3}NMF)应用到北京市 35 个空气质量监测站点 NO_2 浓度数据聚类，实例应用验证所提算法具有良好的适用性和实际应用价值。

1.4 研究内容与结构

论文的主要内容是构建鲁棒的对称非负矩阵分解模型，利用$L_{2,1}$范数提升模型的鲁棒性，并借助自适应学习和判别信息提升模型聚类性能。论文总共分为五部分，具体研究内容如下：

第一部分为引言。介绍选题的研究背景、研究现状及意义和研究内容的创新之处等。

第二部为预备知识。介绍非负矩阵分解、对称非负矩阵分解、鲁棒非负矩阵分解、自适应对称非负矩阵分解的基本原理，以及聚类评价指标。

第三部分为基于$L_{2,1}$范数的鲁棒自适应对称非负矩阵分解算法(RS^{3}NMF)。首先给出研究问题的目标函数、优化求解方法和算法复杂度分析，其次利用数据集进行实验，并进行收敛性分析，最后实例应用。

第四部分为综合考虑$L_{2,1}$范数、自适应学习和判别信息的鲁棒自适应学习判别对称非负矩阵分解算法(RADS^{3}NMF)。首先给出研究问题的目标函数、优化求解方法和算法复杂度分析，其次在公开数据集上进行实验，最后进行实证应用。

第五部分为研究的总结及展望。对本文主要研究内容进行总结，并梳理了未来可以开展的研究工作。

本文研究思路与技术路线图如图 1.1 所示。

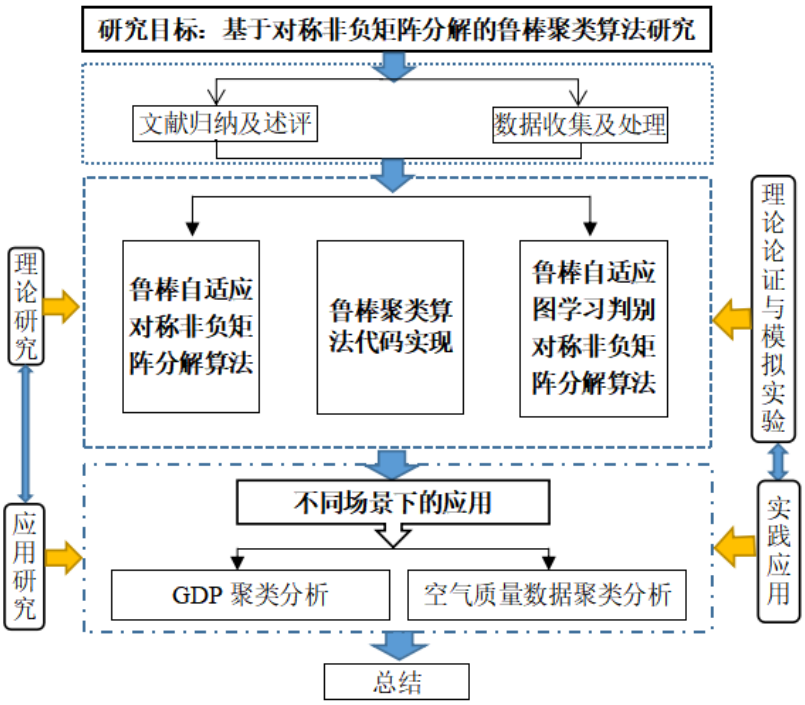


图 1.1 研究思路与技术路线图

1.5 创新点

（1）自适应对称非负矩阵分解采用欧氏距离度量误差，$L_{2,1}$范数采用非平方距离度量误差。相比标准的 Frobenius 范数，$L_{2,1}$范数对异常值不敏感，一定程度上能提高模型的鲁棒性。在自适应对称非负矩阵分解的基础上，利用$L_{2,1}$范数替换 Frobenius 范数，重构目标函数，构建鲁棒的自适应对称非负矩阵分解算法，并应用到我国 31 个省市 GDP 数据的聚类分析。

（2）标准的鲁棒模型只考虑范数对模型的影响，而忽略了自适应学习、判别信息等，此外，通过矩阵分解自适应学习的亲和矩阵，在保持原始数据固有属性的同时能更好地适应模型。因此，本文综合考虑$L_{2,1}$范数、自适应学习和判别信息，构建鲁棒自适应学习判别对称非负矩阵分解算法，进一步应用于北京市空气质量监测点的聚类。

2 预备知识

2.1 非负矩阵分解

非负矩阵分解(Nonnegative Matrix Factorization，NMF)是数学领域中应用广泛的一种矩阵因式分解方法，NMF[10]作为一种流行的数据表示方法和聚类技术[11]被广泛应用于数据挖掘和机器学习。它将原始数据矩阵中呈现的样本或特征表示为基向量的线性组合[12]。同时，通过相应的系数矩阵将隶属标签分配给每个样本或特征，对于高维数据，NMF 可以有效实现降维。给定数据$\boldsymbol{X}=(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)\in\mathbb{R}^{d\times n}$表示非负数据矩阵，其中$\boldsymbol{x}_i\in\mathbb{R}^d$是维数$d$的第$i(i=1,\cdots,n)$个样本向量。NMF 将$\boldsymbol{X}$分解为两个低秩非负矩阵的乘积，并采用重构误差的平方定义目标损失函数，即

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

其中，$\boldsymbol{U}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)\in\mathbb{R}^{d\times k}$为基矩阵，$\boldsymbol{V}=(\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n)^T\in\mathbb{R}^{n\times k}$为系数矩阵。利用乘法更新规则求解式(2.1)有

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

非负矩阵分解试图找到两个非负矩阵$\boldsymbol{U}$、$\boldsymbol{V}$，当数据分布具有线性结构时，NMF 能够获得良好的聚类性能。然而，NMF 不能利用输入数据的非线性结构产生聚类结果[16]。

2.2 对称非负矩阵分解

SNMF 是 NMF 的一种特殊情况，作为图聚类的通用框架，它继承了 NMF 的优点，使聚类划分矩阵具有非负性。然而，与 NMF 不同的是，SNMF 基于数据点之间的相似性度量，并分解包含两两相似值的对称矩阵。给定一个记录样本间成对关系的亲和矩阵$\boldsymbol{W}\in\mathbb{R}^{n\times n}$，SNMF 的目标是找到一个非负矩阵$\boldsymbol{V}\in\mathbb{R}_+^{n\times k}$，使$\boldsymbol{W}\approx\boldsymbol{V}\boldsymbol{V}^T$，且满足

$$\min_{\boldsymbol{V}\geq 0}\left\|\boldsymbol{W}-\boldsymbol{V}\boldsymbol{V}^T\right\|_F^2 \tag{2.3}$$

SNMF 与谱聚类(SC)高度相关[19,28]。SC 的目标函数为

$$\min_{\boldsymbol{V}\geq 0,\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}} \|\boldsymbol{W}-\boldsymbol{V}\boldsymbol{V}^T\|_F^2 \tag{2.4}$$

其中$\boldsymbol{I}\in\mathbb{R}^{k\times k}$是单位矩阵。式(2.3)和式(2.4)目标函数一致，约束条件不同，SC 寻求正交分解，而 SNMF 学习非负嵌入，增强分解矩阵$\boldsymbol{V}=(\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n)^T\in\mathbb{R}^{n\times k}$的可解释性，即$\boldsymbol{v}_i$最大值的位置可以指示样本$\boldsymbol{x}_i$的聚类成员[20]。具体来说，对于 SNMF，通过以下方式获得聚类划分矩阵(或聚类隶属矩阵)$\boldsymbol{M}\in\mathbb{R}^{n\times k}$ ，

$$m_{ij}=\begin{cases}1, & v_{ij}=\max\limits_j v_{ij}\\ 0, & \text{其它}\end{cases} \tag{2.5}$$

其中，m_{ij}和v_{ij}分别是$\boldsymbol{M}$和$\boldsymbol{V}$的第(i,j)元素。对于 SNMF，取$k=c$，即为聚类数目。

2.3 鲁棒非负矩阵分解

由于 NMF 中每个数据点进入目标函数时都带有平方残差，因此很容易出现少数误差较大的异常值支配目标函数。为了提高 NMF 的鲁棒性，Kong 等[28]提出了基于$L_{2,1}$范数度量重构误差的 RNMF，其目标函数为

$$\min_{\boldsymbol{U}\geq 0,\boldsymbol{V}\geq 0}\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right\|_{2,1}=\sum_{j=1}^{n}\left\|\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right)_j\right\|_2 \tag{2.6}$$

与式(2.1)相比，式(2.6)移除平方，缓解了噪声和异常值对目标函数的影响。式(2.6)迭代更新公式如下

$$\begin{aligned}\boldsymbol{U}_{ij}&\leftarrow\boldsymbol{U}_{ij}\frac{(\boldsymbol{X}\boldsymbol{J}\boldsymbol{V})_{ij}}{(\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{J}\boldsymbol{V})_{ij}}\\ \boldsymbol{V}_{ij}&\leftarrow\boldsymbol{V}_{ij}\frac{(\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{J})_{ij}}{(\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{J})_{ij}}\end{aligned} \tag{2.7}$$

其中$\boldsymbol{J}$是对角矩阵，它的对角元素由下式计算

$$J_{jj}=1/\left\|\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^T\right)_i\right\|_2 \tag{2.8}$$

实验结果表明 RNMF 可以提高 NMF 的鲁棒性，具有更好的聚类性能。

2.4 自适应对称非负矩阵分解

Jia 等[27]提出 S^{3}NMF 算法，并利用 SNMF 对初始化特性的敏感性逐步提高聚类性能。首先生成一组随机非负矩阵$\{\boldsymbol{V}_m^0\in\mathbb{R}^{n\times c}\}|_{m=1}^b$($b$是集合的大小)，由 SNMF 获得$b$个聚类划分$\{\boldsymbol{M}_m\}|_{m=1}^b$，构建更高质量的相似矩阵

$$\boldsymbol{S}=\sum_{m=1}^{b}\alpha_m\boldsymbol{M}_m\boldsymbol{M}_m^T \tag{2.9}$$

在多次初始化下重新生成一组新的更好的聚类分区。重复该过程，直到达到终止准则或最大迭代次数。约束优化模型为

$$\begin{aligned}&\min_{\boldsymbol{V_m},\boldsymbol{S},\boldsymbol{\alpha}}\sum_{m=1}^{b}(\alpha_m)^{\tau}\|\boldsymbol{S}-\boldsymbol{V_m}\boldsymbol{V_m^T}\|_F^2\\&\text{s.t.}\boldsymbol{V_m}\geq 0,\ \forall\ m,\ \boldsymbol{\alpha}^T\mathbf{1}=1,\ \boldsymbol{\alpha}\geq 0\end{aligned} \tag{2.10}$$

其中，α_m是$\boldsymbol{\alpha}\in\mathbb{R}^{b\times 1}$的第$m$个元素，权重向量平衡每个分区的贡献，$\mathbf{1}\in\mathbb{R}^{b\times 1}$表示全 1 向量，约束$\boldsymbol{\alpha}^T\mathbf{1}=1$避免了$\boldsymbol{\alpha}$的平凡解(即$\boldsymbol{\alpha}=0$)，$\boldsymbol{\alpha}\geq 0$保证每个$\alpha_m$都是有效权重，$\tau\in(1+\infty)$。

2.5 聚类评价指标

为了评估模型的聚类性能，本文选取聚类精度 (ACC)、归一化互信息(NMI)、纯度（PUR）、调整兰德指数（ARI）以及 F1 分数(F1-score)等 5 个常用指标[24]评估聚类结果。下面分别介绍这 5 个聚类指标。

(1) ACC

给定数据x_i，让r_i和s_i分别为获得的聚类标签和数据本身所提供的标签。ACC 定义如下：

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

其中n是样本总数，$\delta(x,y)$是函数，如果$x=y$，则等于 1，否则等于 0，映射$map(r_i)$是将每个聚类标签r_i映射到数据语料中的等效标签的置换映射函数。使用 Kuhn-Munkres 算法[38]可以找到最佳映射。

(2) NMI

任意两个划分聚类结果$\boldsymbol{M}_i$和$\boldsymbol{M}_j$可视为两个随机变量，归一化互信息(NMI)可评估二者之间的相关性。

$$\mathrm{NMI}_{ij}=\frac{I(\boldsymbol{M}_i,\boldsymbol{M}_j)}{(H(\boldsymbol{M}_i)+H(\boldsymbol{M}_j))/2}$$

其中$\boldsymbol{M}_i$是已经存在的类，$\boldsymbol{M}_j$为特定的聚类结果，$I(\boldsymbol{M}_i,\boldsymbol{M}_j)$为已有类标签聚类赋值的互信息，$H(\boldsymbol{M}_i)$是聚类赋值$\boldsymbol{M}_i$的熵，$\mathrm{NMI}_{ij}$的取值在$[0,1]$的范围内。NMI 值越大，说明聚类解决方案越好。

(3) ARI

兰德指数(Rand index，RI)指度量两个硬划分之间的相似度[39,40]，调整的兰德指数(Adjusted Rand Index，ARI)是它的一种变形，具体定义如下：

对于n个样本的数据集的两个划分$X=\{X_1,X_2,\ldots,X_r\}$和$Y=\{Y_1,Y_2,\ldots,Y_s\}$，记$n_{ij}=|X_i\cap Y_j|$为同时属于X_i和Y_j的样本个数，记$n_i=|X_i|=\sum_j n_{ij}$为X_i中的样本个数，记$n_{\cdot j}=|Y_j|=\sum_i n_{ij}$为$Y_j$中的样本个数，其中$i=1,\ldots,r,j=1,\ldots,s$，则有

$$\mathrm{ARI}=\frac{\sum_{i,j}\binom{n_{ij}}{2}-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i\cdot}}{2}+\sum_j\binom{\mathrm{x}_{\cdot j}}{2}\right]-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}$$

当 RI 指标等于它的期望值，即随机划分时，ARI 指标的值为 0 ，当需要比较的两个划分相同时，ARI 指标的取值为 1。

(4) PUR

纯度(PUR)是一种简单透明的评价指标，通常来说对于每一个簇，对应的标签中哪个最多就认为其属于哪一类[41]。在聚类过程中，Rand 指数会惩罚假阳性和假阴性的决策，F 测度支持对这两种类型的误差进行差分加权。计算公式定义为每个聚类被分配到聚类中最频繁的类，然后通过计算正确分配的类别数量并除以 N 来衡量这种分配的准确性，具体计算公式如下：

$$\mathrm{purity}(\Omega,\mathbb{C})=\frac{1}{N}\sum_k\max_j|\omega_k\cap c_j|$$

其中，$\Omega=\{\omega_1,\omega_2,\ldots,\omega_K\}$ 是簇的集合，$\mathbb{C}=\{c_1,c_2,\ldots,c_J\}$ 是类的集合。我们把ω_k解释为ω_k中的类别集，c_j解释为方程中c_j中的类别集。坏聚类的纯度值接近 0，完美聚类的纯度为 1。

(5) F1-score

当且仅当两个类相似时，通常希望将它们分配给同一个集群。一个真正(TP)决策将两个相似的文档分配到同一个聚类中，一个真负(TN)决策将两个不同的文档分配到不同的聚类中。我们可以犯两种类型的错误。假阳性(FP)决策将两个不同的文档分配给同一个集群。假阴性(FN)决策将两个相似的文档分配给不同的集群。

F1 分数是精度(P)和召回率(R)的加权调和平均值，权衡精度和召回率，公式定义如下：

$$F_\beta - \text{score} = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

其中，$\beta^2 = \frac{1-\alpha}{\alpha}$，$\alpha \in [0,1]$，因此$\beta^2 \in [0,\infty]$。默认的平衡 F 测度同样看重精度和召回率，通常被写成F_1，为$F_{\beta=1}$的缩写，即使在α方面的公式更透明地将 F 测度显示为加权调和平均值。当$\beta = 1$时，右侧公式简化为：

$$F1 - \text{score} = \frac{2PR}{P+R}.$$

3 鲁棒自适应对称非负矩阵分解算法

本部分在自适应对称非负矩阵分解的基础上，提出了基于$L_{2,1}$范数的鲁棒自适应对称非负矩阵分解聚类算法(RS^{3}NMF)，并给出了模型的优化求解过程。使用$L_{2,1}$范数重构模型误差，提高了模型的鲁棒性，使模型对噪声和异常值不敏感，保持了特征旋转不变性，增强了聚类性能。

3.1 目标函数

RS^{3}NMF 采用$L_{2,1}$范数度量损失函数，以缓解噪声和异常值的影响，并利用 SNMF 对初始化特性的敏感性来逐步提高聚类性能。RS^{3}NMF 的目标函数可以表示为

$$\begin{aligned} &\min_{\boldsymbol{H_r},\boldsymbol{W},\boldsymbol{\alpha}} \sum_{r=1}^{k} (\alpha_r)^{\gamma} \|\boldsymbol{W} - \boldsymbol{H}_r \boldsymbol{H}_r^T\|_{2,1} \\ &\text{s.t. } \boldsymbol{H}_r \geq 0,\ \forall\ r,\ \|\boldsymbol{\alpha}\|_1 = 1,\ \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{3.1}$$

其中$\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$为度量$\boldsymbol{X} \in \mathbb{R}^{m\times n}$第$i$个和第$j$个样本之间相似性的重构相似矩阵。$\boldsymbol{H}_r \in \mathbb{R}_+^{n\times c}$中每一行的最大值的位置指示样本$\boldsymbol{x}_i$的聚类成员，表示聚类数。$\alpha_r$是权重向量$\boldsymbol{\alpha} \in \mathbb{R}^{k\times 1}$的第$r$个元素，用于平衡每组聚类结果的贡献。约束$\|\boldsymbol{\alpha}\|_1 = 1$避免了$\boldsymbol{\alpha}$的平凡解(即$\boldsymbol{\alpha} = 0$)，$\boldsymbol{\alpha} \geq 0$保证每个都是有效权重。超参数$\gamma$可控制$\boldsymbol{\alpha}$项的分布，防止极端稀疏的解产生，即大部分$\boldsymbol{\alpha}$等于或非常接近 0。注意到，当$\gamma \to 1$时，只有$\boldsymbol{\alpha}$的少数条目将支配向量，而当$\gamma \to +\infty$时，式(3.1)将赋予$\boldsymbol{\alpha}$相等的权重。$\gamma$取值太大，式(3.1)所赋权重相等，$\gamma$取值太小，起作用的因子太少，因此，$\gamma$的取值不宜太大，也不宜太小，$\gamma$的建议取值对整体数据集是可接受的。

具体地，随机生成一组非负矩阵$\{\boldsymbol{H}_r^0 \in \mathbb{R}^{n\times c}\}|_{r=1}^{k}$($k$表示集合中元素的个数)，获得$k$个聚类划分矩阵(或聚类隶属矩阵)$\boldsymbol{M}_r \in \mathbb{R}^{n\times c}$，且

$$m_{ij} = \begin{cases} 1, & v_{ij} = \max\limits_{j} v_{ij} \\ 0, & \text{其它} \end{cases} \tag{3.2}$$

从而得到重构相似矩阵

$$\boldsymbol{W} = \sum_{r=1}^{k} \alpha_r \boldsymbol{M}_r \boldsymbol{M}_r^T \tag{3.3}$$

这里h_{rij}、m_{rij}分别是$\boldsymbol{H}_r$和$\boldsymbol{M}_r$的第i第j列的元素。重复该过程，直到达到终止准则或最大迭代次数。

3.2 优化算法

下面利用乘法更新规则推导 RS^{3}NMF 算法式(3.1)的更新公式。

a)优化$\boldsymbol{H}_r$。固定$\boldsymbol{\alpha}$，更新$\boldsymbol{H}_r$。利用一个固定的$\boldsymbol{W}^0$以及多个随机非负初始化矩阵$\boldsymbol{H}_r^0$。关于$\boldsymbol{H}_r$的目标函数可写为

$$\min_{\boldsymbol{H}_r}\sum_{r=1}^{k}(\alpha_r)^\gamma\|\boldsymbol{W}-\boldsymbol{H}_r\boldsymbol{H}_r^T\|_{2,1},\quad \text{s.t. } \boldsymbol{H}_r\geq 0,\ \forall\, r \tag{3.4}$$

由于k个分解矩阵$\boldsymbol{H}_r$是相互独立的，可以分别求解每个$\boldsymbol{H}_r$，式(3.4)可写为r个子问题

$$\min_{\boldsymbol{H}_r\geq 0}(\alpha_r)^\gamma\|\boldsymbol{W}-\boldsymbol{H}_r\boldsymbol{H}_r^T\|_{2,1} \tag{3.5}$$

式(3.5)是具有非负约束的非凸优化问题，不存在解析解。为此，通过构造辅助函数求解式(3.5)的数值解。

辅助函数的定义如下。

定义 称$g(x)$是$f(x)$的辅助函数，如果下列条件成立

$$f(x)\leq g(x),\quad g(x=x^t)=f(x=x^t)$$

其中$g(x=x^t)=f(x=x^t)$表示在点x^t处$f(x)$与$g(x)$有相同的值。

在每次迭代中减少$g(x)$，即$g(x^{t+1})<g(x^t)$。注意到

$$f(x^t)=g(x^t)>g(x^{t+1})\geq f(x^{t+1})$$

所以原函数$f(x)$也会减少，即$f(x^{t+1})<f(x^t)$。

下面构造式(3.5)的辅助函数。式(3.5)的目标函数可以扩展为

$$\begin{aligned}\mathcal{O}(\boldsymbol{H}_r)=&\ (\alpha_r)^\gamma\|\boldsymbol{W}-\boldsymbol{H}_r\boldsymbol{H}_r^T\|_{2,1}\\ =&\ (\alpha_r)^\gamma\left[\mathrm{tr}(\boldsymbol{W}^T\widetilde{\boldsymbol{G}}_r\boldsymbol{W})-2\mathrm{tr}(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T)+\mathrm{tr}(\boldsymbol{H}_r\boldsymbol{H}_r^T\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T)\right]\end{aligned} \tag{3.6}$$

其中$\boldsymbol{G}_r=\mathrm{diag}(g_{r11},g_{r22},\cdots,g_{rnn})\in\mathbb{R}^{n\times n}$是第$r$个随机非负矩阵对应的对角权重矩阵

$$g_{rii}=1/\left\|(\boldsymbol{W}-\boldsymbol{H}_r\boldsymbol{H}_r^T)_i\right\|_2 \tag{3.7}$$

$$\widetilde{\boldsymbol{G}}_r=(\alpha_r)^\gamma\boldsymbol{G}_r \tag{3.8}$$

式(3.6)中的第一项$\boldsymbol{W}^T\widetilde{\boldsymbol{G}}_r\boldsymbol{W}$看作常数，第二项可写为

$$\mathrm{tr}(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T)=\sum_i\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T\right)_{ii}\sum_{ijk}w_{ik}\widetilde{g}_{rkk}h_{rkj}h_{rij}$$

其中$\widetilde{g}_{rkk}=\alpha_r^\gamma g_{rkk}$。注意到，对$\forall x>0$，有$x\geq 1+\log x$。

令$x=\frac{h_{rkj}h_{rij}}{h_{rkj}^t h_{rij}^t}$，则有

$$\mathrm{tr}(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T)\geq\sum_{ijk}\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}h_{rij}^t\left(1+\log\frac{h_{rkj}h_{rij}}{h_{rkj}^t h_{rij}^t}\right) \tag{3.9}$$

参考文献[17]，式(3.6)的第三项可化为

$$\begin{aligned}&\mathrm{tr}(\boldsymbol{H}_r\boldsymbol{H}_r^T\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T)\\=&\sum_i\left(\boldsymbol{H}_r\boldsymbol{H}_r^T\widetilde{\boldsymbol{G}}_r\boldsymbol{H}_r\boldsymbol{H}_r^T\right)_{ii}=\sum_{ik}\left(\boldsymbol{H}_r\boldsymbol{H}_r^T\widetilde{\boldsymbol{G}}_r\right)_{ik}\left(\boldsymbol{H}_r\boldsymbol{H}_r^T\right)_{ki}\\=&\sum_{ijk}\left(\boldsymbol{H}_r\boldsymbol{H}_r^T\widetilde{\boldsymbol{G}}_r\right)_{ik}h_{rkj}^t h_{rij}\leq\sum_{ij}\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{V}_r^t\right)_{ij}\frac{(h_{rij})^4}{(h_{rij}^t)^3}\end{aligned} \tag{3.10}$$

结合式(3.9)和式(3.10)，在第t次迭代中，构造式(3.4)的辅助函数为

$$\begin{aligned}g(\boldsymbol{H}_r)=&(\alpha_r)^\gamma\sum_{ij}\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{(h_{rij})^4}{(h_{rij}^t)^3}\\&-2(\alpha_r)^\gamma\sum_{ijk}\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)h_{rij}^t\left(1+\log\frac{h_{rkj}h_{rij}}{h_{rkj}^t h_{rij}^t}\right)\end{aligned} \tag{3.11}$$

对式(3.11)分别求一阶导数和二阶导数，有

$$\begin{aligned}\frac{\partial g(\boldsymbol{H}_r)}{\partial h_{rij}}=&4(\alpha_r)^\gamma\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{(h_{rij})^3}{(h_{rij}^t)^3}\\&-2(\alpha_r)^\gamma\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{h_{rij}^t}{h_{rij}}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\frac{\partial^2 g(\boldsymbol{H}_r)}{\partial h_{rij}^2}=&12(\alpha_r)^\gamma\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{(h_{rij})^2}{(h_{rij}^t)^3}\\&+2(\alpha_r)^\gamma\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{h_{rij}^t}{v_{rij}^2}\end{aligned} \tag{3.13}$$

$$\begin{aligned}\frac{\partial^2 g(\boldsymbol{H}_r)}{\partial h_{rij}\partial h_{rkl}}=&12(\alpha_r)^\gamma\delta_{ik}\delta_{jl}\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{(h_{rij})^2}{(h_{rij}^t)^3}\\&+2(\alpha_r)^\gamma\delta_{ik}\delta_{jl}\left(\boldsymbol{W}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}\frac{h_{rij}^t}{h_{rij}^2}\end{aligned} \tag{3.14}$$

其中，当$i=j$时，$\delta_{ij}=1$，否则，$\delta_{ij}=0$。由于$\frac{\partial^2 g(\boldsymbol{H}_r)}{\partial h_{rij}\partial h_{rkl}}>0$，因此，构造的辅助函数$g(\boldsymbol{H}_r)$是凸函数，当$\frac{\partial g(\boldsymbol{H}_r)}{\partial h_{rij}}=0$时，函数$g(\boldsymbol{H}_r)$取到全局最小值。具体来说，在每次迭代中，$\boldsymbol{H}_r$的更新公式为

$$h_{rij}^{t+1}=h_{rij}^t\left(\frac{\left(\boldsymbol{w}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}}{2\left(\boldsymbol{H}_r^t\boldsymbol{H}_r^{tT}\widetilde{\boldsymbol{G}}_r^t\boldsymbol{H}_r^t\right)_{ij}}\right)^{\frac{1}{4}},\ \forall\ i,j \tag{3.15}$$

根据辅助函数的性质，式(3.15)减小，原始函数式(3.4)也会随之减小。

b)优化$\boldsymbol{\alpha}$。固定$\boldsymbol{H}_r$，$\forall\, r$，更新$\boldsymbol{\alpha}$。利用一个固定的$\boldsymbol{W}^0$以及多个随机非负初始化矩阵$\boldsymbol{H}_r^0$，$\boldsymbol{\alpha}$的子问题为

$$\min_{\boldsymbol{\alpha}} \sum_{r=1}^{k}(\alpha_r)^{\gamma}\boldsymbol{Q}_r, \quad \text{s.t. } \|\boldsymbol{\alpha}\|_1 = 1, \ \boldsymbol{\alpha} \geq 0 \tag{3.16}$$

其中$\boldsymbol{Q}_r = \left\|\boldsymbol{W} - \boldsymbol{H}_r\boldsymbol{H}_r^T\right\|_{2,1}$，式(3.16)的拉格朗日函数为

$$\mathcal{O}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\alpha}}(\alpha_r)^{\gamma}\boldsymbol{Q}_r - \mu(\sum_{r}\alpha_r - 1) \tag{3.17}$$

式(3.17)关于$\boldsymbol{\alpha}$求一阶导数，并令$\frac{\partial\mathcal{O}(\boldsymbol{\alpha})}{\partial\alpha_r} = 0$，则有

$$\alpha_r = \left(\frac{\mu}{\gamma\boldsymbol{Q}_r}\right)^{\frac{1}{\gamma-1}}, \ \forall\, r \tag{3.18}$$

利用约束条件$\|\boldsymbol{\alpha}\|_1 = 1$，$\boldsymbol{\alpha} \geq 0$，可得

$$\mu = \left(\frac{1}{\sum_{r}(\gamma\boldsymbol{Q}_r)^{\frac{1}{1-\gamma}}}\right)^{\gamma-1}$$

将μ代入式(3.18)得

$$\alpha_r = \frac{(\gamma\boldsymbol{Q}_r)^{\frac{1}{1-\gamma}}}{\sum_{r}(\gamma\boldsymbol{Q}_r)^{\frac{1}{1-\gamma}}} \tag{3.19}$$

式(3.19)分子分母均大于 0，有$\alpha_r > 0$，$\forall\, r$，满足条件$\boldsymbol{\alpha} \geq 0$，并且式(3.19)满足式(3.16)的 KKT 条件，因此是局部最优的。由于式(3.16)是凸问题，所以式(3.19)也是式(3.16)的全局最优解。

c)优化$\boldsymbol{W}$和$\boldsymbol{G}_r$。固定$\boldsymbol{H}_r$和$\boldsymbol{\alpha}$。利用式(3.3)更新$\boldsymbol{W}$。固定$\boldsymbol{H}_r$、$\boldsymbol{W}$和$\boldsymbol{\alpha}$，利用式(3.8)更新$\boldsymbol{G}_r$。

综上分析，依次根据式(3.15)、式(3.19)、式(3.3)和式(3.8)，交替迭代更新$\boldsymbol{H}_r$、$\boldsymbol{\alpha}$、$\boldsymbol{W}$和$\boldsymbol{G}_r$，可以完成非凸优化问题式(3.4)的求解，亦即实现了鲁棒自适应对称非负矩阵分解聚类算法(RS^{3}NMF)。 RS^{3}NMF 算法见表 3.1。

表 3.1 RS³NMF 算法表

算法 1 鲁棒自适应对称非负矩阵分解聚类算法(RS³NMF)

输入：经验亲和矩阵$\boldsymbol{P}$，超参数γ，k，$\varepsilon = 10^{-3}$
输出：一组聚类结果$\{\boldsymbol{M}_r\}|_{r=1}^{k}$
1：初始化：iter=1，maxIter=10，$t=1$，一组随机的非负矩阵$\boldsymbol{H}_r^0$，$\boldsymbol{W}=\boldsymbol{P}$；
2：while iter < maxIter do
3：　while $t<500$ do
4：　　for $r\in\{1,\cdots,b\}$ do
5：　　利用式(3.15)更新$\boldsymbol{H}_r$；
6：　　end for
7：　　利用式(3.29)更新$\boldsymbol{\alpha}$；
8：　　if $\|\boldsymbol{H}_r^{t+1}-\boldsymbol{H}_r^t\|_\infty<\varepsilon$and $\|\boldsymbol{\alpha}^{t+1}-\boldsymbol{\alpha}^t\|_\infty<\varepsilon$ then
9：　　break；
10：　　end if
11：　　$t=t+1$ ；
12：　end while
13：　根据式(3.3)更新$\boldsymbol{W}$；根据式(3.8)更新$\boldsymbol{G}_r$；
14：　if ANMI 值开始下降 then
15：　break
16：　end if
17：　iter= iter+1；
18：end while
19：通过式(3.2)生成聚类划分矩阵$\{\boldsymbol{M}_r\}|_{r=1}^{k}$；

3.3 算法终止准则

归一化互信息(NMI)用来评估两个聚类分区之间的相似性。NMI 值越大，说明聚类解决方案越好。不妨假设，在开始的几次迭代中，所有划分的聚类性能可以逐步提高，划分之间的一致性也可以提高，当达到最大一致性时，它们之间的一致性将保持在相当的水平，甚至可能由于迭代中变量初始化的随机性而减少和波动。互信息是衡量两个随机变量之间关系的一个量。我们将两个分区的聚类结果视为两组随机变量，然后使用归一化互信息来评估它们的相关性，NMI 取值在$[0,1]$的范围内，使用平均 NMI (ANMI)来衡量所有分区的共识水平，即

$$\mathrm{ANMI}=\frac{1}{k(k-1)}\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathrm{NMI}_{ij} \tag{3.20}$$

其中，ANMI 的取值范围为$[0,1]$，ANMI 的取值较大意味着不同划分之间的一致性水平较高。当 ANMI 开始下降时，终止 RS³NMF 算法，并将 ANMI 最高取值的迭代作为最终聚类结果。

3.4 算法收敛性与复杂性

根据辅助函数的性质，可以证明算法 1 是收敛的。根据式(3.15)更新$\{\boldsymbol{H}_r\}_{r=1}^{k}$不会增加目标函数式(3.4)，即$\mathcal{O}(\boldsymbol{H}_r^{t+1},\boldsymbol{\alpha}^t)\leq\mathcal{O}(\boldsymbol{H}_r^{t},\boldsymbol{\alpha}^t)$；固定$\{\boldsymbol{H}_r\}_{r=1}^{k}$，根据式(3.19)更新$\boldsymbol{\alpha}$，得到$\mathcal{O}(\boldsymbol{H}_r^{t+1},\boldsymbol{\alpha}^{t+1})\leq\mathcal{O}(\boldsymbol{H}_r^{t+1},\boldsymbol{\alpha}^t)$。因此，在每次迭代中，有$\mathcal{O}(\boldsymbol{H}_r^{t+1},\boldsymbol{\alpha}^{t+1})\leq\mathcal{O}(\boldsymbol{H}_r^{t},\boldsymbol{\alpha}^t)$。此外，对$\forall\ r$，参数$\boldsymbol{\alpha}$、$\|\boldsymbol{W}-\boldsymbol{H}_r\boldsymbol{H}_r\|_{2,1}$是非负的，根据构造的辅助函数可知目标函数不增且有下界，从而验证 RS^{3}NMF 算法的收敛性。

进一步讨论 RS^{3}NMF 算法的计算复杂度。最大迭代次数为τ，n是总样本量，c是聚类数，k表示初始化的次数。对于算法 1，交替迭代求解$\boldsymbol{H}_r$的子问题，计算复杂度为$O(n^3k+n^2ck)$，更新求解$\boldsymbol{\alpha}$子问题的计算复杂度为$O(n)$，构造$\boldsymbol{W}$的复杂度为$O(n^2c)$，$\boldsymbol{G}_r$的计算复杂度$O(n^2ck)$。因此，算法 1 每次迭代的复杂度为$O(n^3k\tau+n^2ck\tau)$。

3.5 实验

本节我们通过实验来验证所提出 RS^{3}NMF 聚类算法的有效性。在 5 个公开数据集和 5 个聚类指标上，将所提出的 RS^{3}NMF 算法与 7 种具有代表性的聚类算法进行了比较。实验利用 MATLAB 软件实现，实验的计算机环境为：Intel(R) Core(TM) i7-10875H CPU2.30 GHz，内存 16GB，Windows10 64 位操作系统。

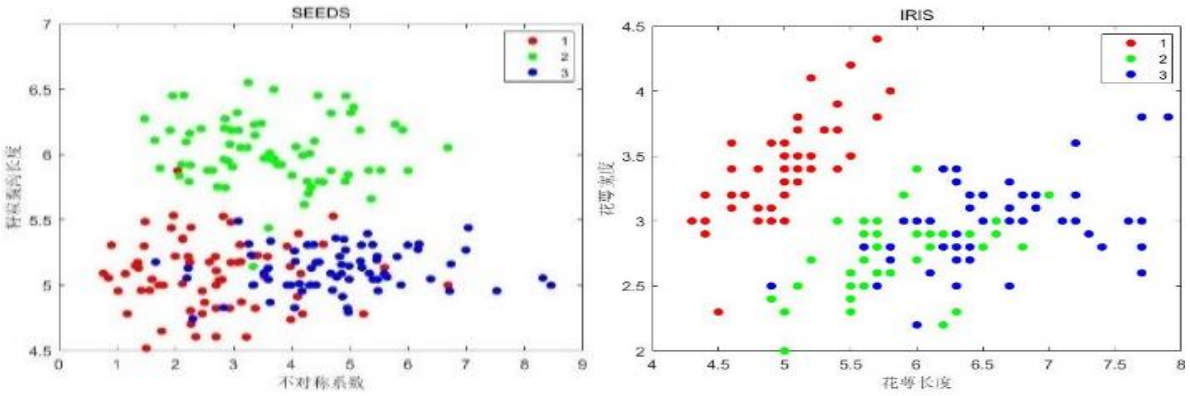
3.5.1 数据集

表 3.2 实验使用的数据集

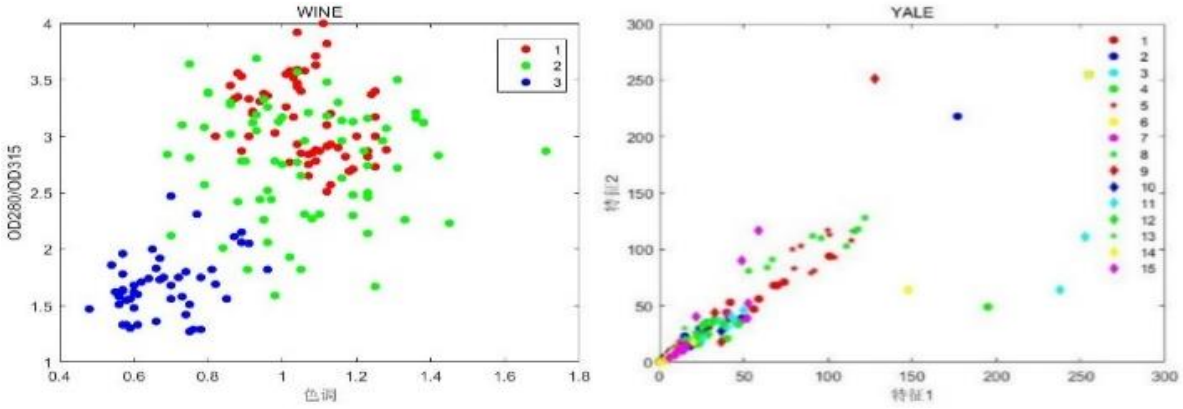
数据集	样本总数(n)	特征维度(k)	类别数(c)	数据类型
SEEDS	210	7	3	小麦种子数据
IRIS	150	4	3	鸢尾花卉数据
WINE	150	3	3	葡萄酒数据
YALE	165	1024	15	图像，人脸数据
3-Sources	169	3560	6	新闻数据

实验选用 5 个公开数据集：图像数据集 YALE [1]，新闻数据集 3-Sources，以及来自 UCI 机器学习库[2](简称 UCI)的 3 个数据集 IRIS、SEEDS 和 WINE。数据集的详细信息参见表 3.2。选取 5 个数据集的部分特征绘制散点图，如图 3.1 所示。

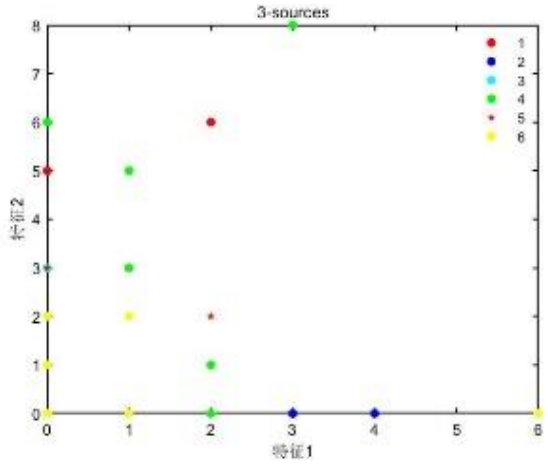
[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
[2] http://archive.ics.uci.edu/ml



SEEDS 数据散点图　　IRIS 数据散点图



WINE 数据散点图　　YALE 数据散点图



3-sources 数据散点图

图 3.1　数据集示例散点图

从图 3.1 中的散点图可以看出，数据中存在一定数量异常偏离的数据，异常值的数据对分析造成了干扰，因此，有必要提出更鲁棒的模型来减少异常值模型结果的干扰，以增强模型的聚类效果。

3.5.2 实验设置

在 5 个数据集上，所提出的 RS^{3}NMF 聚类算法与 7 种算法进行对比分析，验证 RS^{3}NMF 算法的有效性。包括三类：第一类是 K-means 、NMF[10] 、GNMF[16]与 RNMF[28]，第二类是图聚类算法 SC[42]与 SNMF[20]，第三类集成聚类算法 S^{3}NMF[12]。为了使 8 种方法实现结果具有可比性，对 8 种方法的结构参数进行统一设置。具体设置如下：

K-means 设置为默认参数设置。RS^{3}NMF 与 S^{3}NMF 的迭代次数设为 20 次，超参数取$\gamma = 2, k = 20$; GNMF 迭代次数为 300 次，超参数$\lambda = 100$。特别地，当$\lambda = 0$时，GNMF 退化为 NMF。NMF、 SC、RNMF 和 SNMF 的迭代次数为 300 次。RS^{3}NMF 和 S^{3}NMF、SNMF 采用相同的亲和矩阵作为输入。为了排除随机性对 K-means 和初始化的影响，我们将每种方法独立重复试验 20 次，并报告平均性能。

3.5.3 聚类性能分析

聚类结果通过 5 个常用指标进行评估[24]：聚类精度(ACC)、归一化互信息(NMI)、纯度(PUR)、调整兰德指数(ARI)和 F1 分数(F1-score)。除 ARI 之外的所有度量都在$[0, 1]$的范围内，而 ARI 的值范围为$[-1, 1]$。所有的指标值越大，说明聚类性能越好。表 3.3-表 3.7 展示了 8 种不同方法在 5 个数据集上的聚类性能。

表 3.3 SEEDS 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.6857	0.4046	0.6857	0.3794	0.5857
NMF	0.8809	0.6508	0.8890	0.6850	0.7893
GNMF	0.8095	0.6116	0.8095	0.5535	0.7050
SC	0.8286	0.6337	0.8286	0.5862	0.7259
RNMF	0.6714	0.3091	0.6714	0.3061	0.5370
SNMF	0.7571	0.4916	0.7571	0.4004	0.6212
S^{3}NMF	0.8810	0.6670	0.8810	0.6880	0.7920
RS^{3}NMF	**0.9048**	**0.7031**	**0.9048**	**0.7373**	**0.8242**

表 3.4 IRIS 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.8933	0.7515	0.8933	0.7302	0.8207
NMF	0.7267	0.5893	0.7267	0.5099	0.6713
GNMF	0.9067	0.7696	0.9067	0.7576	0.8384
SC	0.7333	0.4434	0.7333	0.3347	0.5830
RNMF	0.7067	0.5841	0.7067	0.5072	0.6735
SNMF	0.8733	0.7536	0.8733	0.6956	0.8004
S^3NMF	0.7533	0.5398	0.7533	0.4775	0.6575
RS^{3}NMF	**0.9333**	**0.8027**	**0.9333**	**0.8176**	**0.8776**

表 3.5 WINE 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.7023	0.4287	0.7023	0.3711	0.5835
NMF	0.6407	0.3084	0.6629	0.3424	0.5935
GNMF	0.7023	0.4 224	0.7023	0.3598	0.5762
SC	0.6854	0.3664	0.6854	0.3101	0.5476
RNMF	0.5337	0.3539	0.6517	0.3178	0.5712
SNMF	0.5449	0.3466	0.6573	0.2891	0.5816
S^3NMF	0.5933	0.2135	0.5955	0.1096	0.4688
RS^{3}NMF	**0.7247**	**0.3921**	**0.7247**	**0.3955**	**0.5973**

表 3.6 YALE 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.3939	0.4384	0.4061	0.1797	0.2411
NMF	0.4121	0.4707	0.4424	0.2117	0.2654
GNMF	0.3091	0.4012	0.3515	0.1305	0.1921
SC	0.3818	0.4391	0.3818	0.1935	0.2492
RNMF	0.4303	0.4547	0.4424	0.1994	0.2502
SNMF	0.4424	0.5126	0.4606	0.2363	0.2854
S^3NMF	0.4658	0.5043	0.4727	0.2540	0.3022
RS^{3}NMF	**0.4667**	**0.5178**	**0.4727**	**0.2619**	**0.3092**

表 3.7 3-Sources 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.4260	0.1790	0.4675	0.0722	0.3716
NMF	0.4260	0.2825	0.5503	0.0769	0.3514
GNMF	0.3669	0.2003	0.4970	0.0798	0.2926
SC	0.4438	0.2172	0.5621	0.1370	0.4000
RNMF	0.3728	0.2349	0.4793	0.0686	0.3534
SNMF	0.4675	0.3438	0.5740	0.2282	0.3951
S^3NMF	0.5030	0.3862	0.6036	0.3139	0.4613
RS^{3}NMF	**0.5207**	**0.4039**	**0.6331**	**0.3580**	**0.4852**

由表 3.3-表 3.7 得到以下结论：

(1) RS^{3}NMF 算法显著优于 S^{3}NMF 算法，特别在 IRIS 和 WINE 数据集上 5 个聚类评价指标结果有明显改进，这表明 RS^{3}NMF 算法优于 S^{3}NMF 算法，更具鲁棒性。例如，在 IRIS 数据集上，ACC 值从 0.7533 提高到 0.9333，NMI 值从 0.5398 增加到 0.8027，PUR、ARI 和 F1-score 的值也分别提高了 0.1800、0.3401 和 0.2201。

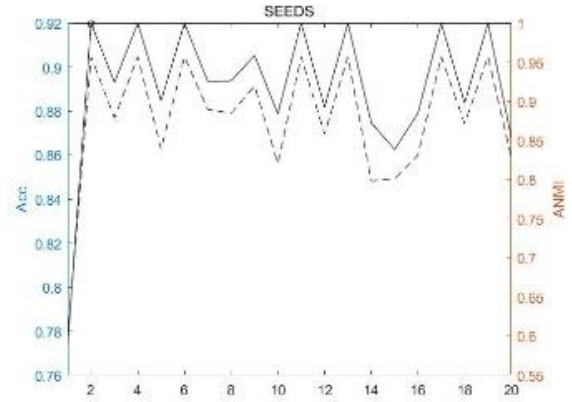
(2) 与鲁棒聚类算法 RNMF 相比，RS^{3}NMF 算法的优势也很明显。例如在 SEEDS、IRIS 和 3-Sources 数据集上，RS^{3}NMF 的 ACC 值分别提升了 35%、32%和 40%。这意味着 RS^{3}NMF 算法对噪声和异常值具有十分良好的鲁棒性。

(3) 与图聚类算法 SC 和 SNMF 相比，RS^{3}NMF 算法的聚类性能也显著提升。例如，在 IRIS 数据集上，与 SC 算法比较，ACC 值提高了 27%；在 SEEDS 数据集上，与 SNMF 算法比较，ACC 值提高了 20%。总的来说，相比其他 7 种聚类算法，RS^{3}NMF 算法在这 5 个数据集上始终产生最好或较好的聚类性能，验证了其鲁棒性。

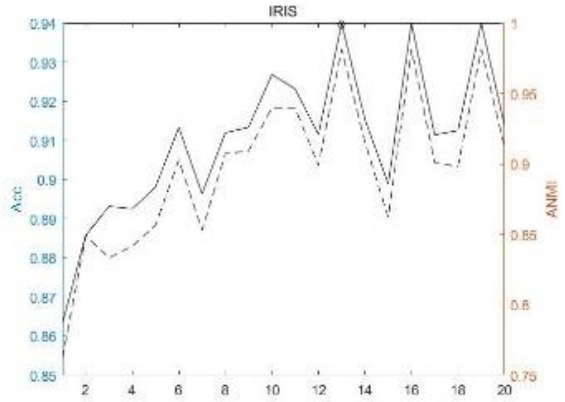
3.5.4 迭代次数对聚类性能的影响

下面研究 RS^{3}NMF 算法中迭代次数对聚类性能的影响。如图 3.2 所示，虚线表示所提 RS^{3}NMF 在不同迭代次数下的 ACC 值，实线表示 RS^{3}NMF 算法终止准则 ANMI 的值，圈表示在该点达到了所建议的终止准则。图 3.2 显示在 SEEDS、WINE 和 YALE 数据集上，所提算法 RS^{3}NMF 的 ACC 值在最初的 3 次迭代中迅速增加，同时随着迭代次数的增加其最高值基本保持不变，波动范围在$0.02 \sim 0.06$之间，趋于相对稳定，RS^{3}NMF 算法的终止准则 ANMI 可以选择最高的 ACC。对于 IRIS 数据集，虽没有选择最高的 ACC，

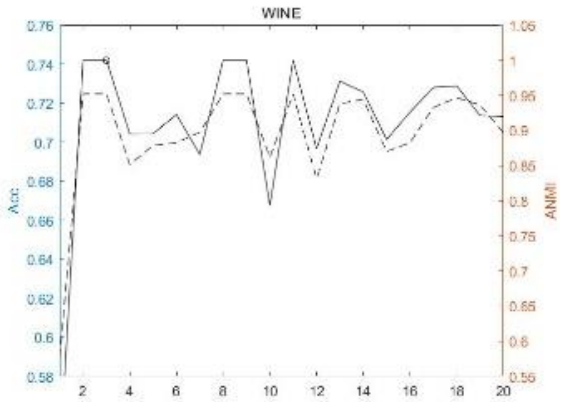
但终止准则 ANMI 也可以为 RS^{3}NMF 产生一个满意的 ACC，并保持最大取值基本不变。注意到，在 SEEDS、WINE 和 YALE 数据集上，2 或 3 次迭代内就可终止算法 1，有效降低计算成本。因此，RS^{3}NMF 算法的终止准则 ANMI 是有效且高效的。



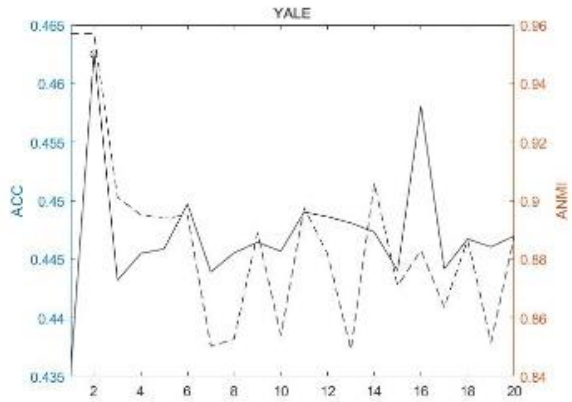
(a)SEEDS 数据迭代次数图



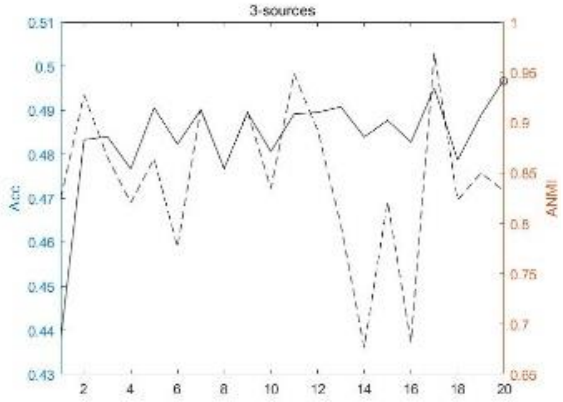
(b)IRIS 数据迭代次数图



(c)WINE 数据迭代次数图



(d) YALE 数据迭代次数图



(e)3-sources 数据迭代次数图

图 3.2　迭代次数与 ACC 和 ANMI 的关系

3.5.5 收敛性分析

我们通过实验验证 RS^{3}NMF 算法的收敛性，并研究算法的收敛速度。收敛曲线如图 3.3 所示。图 3.3 呈现了当$\gamma = 2$，$k = 50$时，5 个数据集上目标函数值的对数与迭代次数的关系。随着迭代次数的增加，目标函数值的对数快速单调递减。算法 RS^{3}NMF 对 5 个数据集通常在 30 次迭代内收敛，说明本文提出的优化算法是有效的，收敛速度较快。

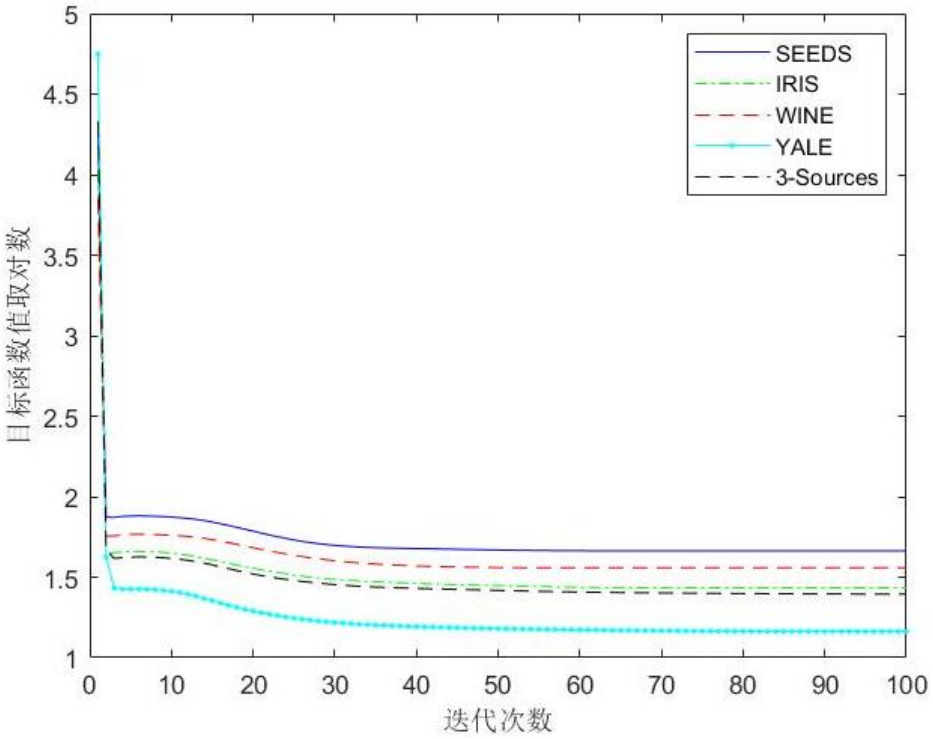


图 3.3 RS^{3}NMF 算法在 5 个数据集上的收敛曲线

3.6 实证应用

我国是一个地域辽阔、人口众多的发展中国家。31 个省市根据所处地理位置可以划分为东中西三个区域，具体划分如下：东部地区（包括北京、天津、河北、辽宁、上海、江苏、浙江、福建、山东、广东、海南）、中部地区（包括黑龙江、安徽、江西、河南、湖北、湖南、山西、吉林）、西部地区（包括内蒙古、广西、重庆、四川、贵州、云南、西藏、陕西、甘肃、青海、宁夏、新疆）。各省市由于地理位置和资源的制约导致经济发展不平衡，如果实施同一套经济政策，采用一刀切的方法将制约其发展，可能会引发一系列社会问题。因此，研究我国各省市的 GDP 发展状况，利用聚类分析的方法对各省市 GDP 进行聚类分析，对国家的经济发展有着重要的现实意义。数据来源：本文数据来源为中国经济社会大数据研究平台(https://data.cnki.net/)。

(1) S^{3}NMF 聚类

对 1999-2021 年我国 31 个省市 GDP 总值数据构造新的亲和矩阵，首先利用 S^3NMF 进行聚类，然后利用 RS^{3}NMF 进行聚类，将我国 31 个省市按照 GDP 数据聚为 5 类，并分析聚类效果。S^3NMF 聚类结果如表 3.8 所示。

表 3.8 S^3NMF 聚类结果分类表

分类	省市
第一类	上海市、江苏省、浙江省、山东省、广东省、四川省
第二类	北京市、河北省、安徽省、福建省、河南省、湖南省、湖北省
第三类	山西省、辽宁省、云南省、陕西省
第四类	内蒙古自治区、江西省、广西壮族自治区、重庆市、贵州省
第五类	天津市、吉林省、黑龙江省、海南省、西藏自治区、甘肃省、青海省、宁夏回族自治区 、新疆维吾尔族自治区

由表 3.8 可知，我国 31 个省市按照 GDP 总值分为 5 类，其中上海市、江苏省、浙江省、山东省、广东省、四川省为第一类，就第一类省份而言，高新技术产业发达，人才驱动经济发展；北京市、河北省、安徽省、福建省、河南省、湖南省、湖北省为第二类，就第二类省份而言，经济产业比较丰富，产业结构较为完善；山西省、辽宁省、云南省、陕西省为第三类，就第三类省份而言，以特色产业为导向驱动经济增长；内蒙古自治区、江西省、广西壮族自治区、重庆市、贵州省为第四类，就第四类省份而言，拥有一定优势资源，没有明显强劲产业驱动经济快速发展；天津市、吉林省、黑龙江省、海南省、西藏自治区、甘肃省、青海省、宁夏回族自治区、新疆维吾尔族自治区为第五类，就第五类省份而言，区位优势相对较弱，产业结构不完善，人才流失比较严重。分析聚类结果可以发现，整理后的聚类出现一定的阶梯性，即从第 1 类到第 5 类整体体现为从东到西整体 GDP 由强到弱的变化过程，东部地区 GDP 总体较高，西部地区 GDP 整体较低。

利用 ArcGIS 软件可视化展示聚类结果，如图 3.4 所示。

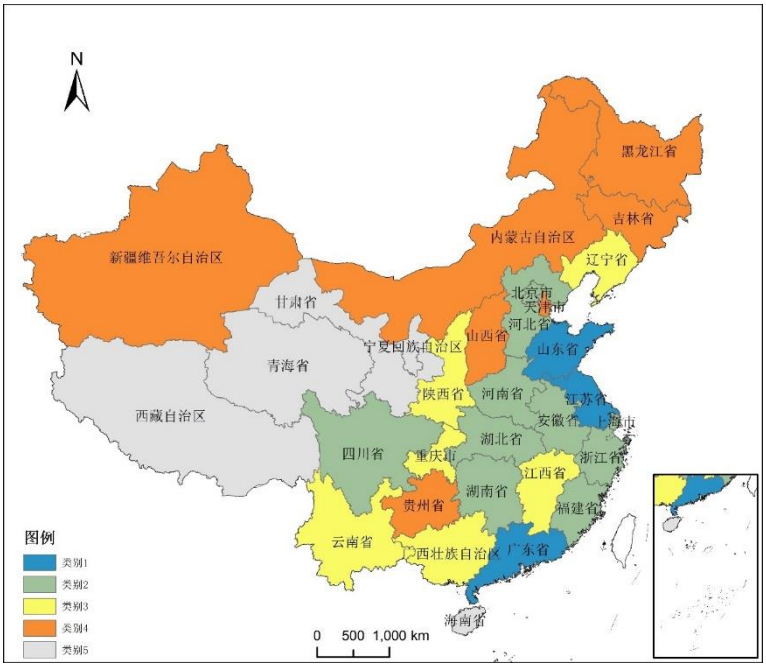


图 3.4 自适应对称非负矩阵分解聚类结果

(2) RS^{3}NMF 聚类

利用 RS^{3}NMF 进行分解，将我国 GDP 数据聚为 5 类，具体聚类结果如表 3.9 所示。

表 3.9 RS^{3}NMF 聚类结果分类表

分类	省市
第一类	北京市、上海市、河北省、安徽省、福建省、湖北省、湖南省
第二类	江苏省、浙江省、山东省、河南省、广东省、四川省
第三类	天津市、山西省、内蒙古自治区
第四类	辽宁省、江西省、广西壮族自治区、重庆市、贵州省、云南省、陕西省
第五类	吉林省、黑龙江省、海南省、西藏自治区、甘肃省、青海省、宁夏回族自治区 、新疆维吾尔族自治区

由表 3.9 可知，RS^{3}NMF 将北京市、上海市、河北省、安徽省、福建省、湖北省、湖南省聚分为一类；将江苏省、浙江省、山东省、河南省、广东省、四川省聚为一类；将天津市、山西省、内蒙古自治区聚分为一类；将辽宁省、江西省、广西壮族自治区、重庆市、贵州省、云南省、陕西省聚为一类；将吉林省、黑龙江省、海南省、西藏自治区、甘肃省、青海省、宁夏回族自治区 、新疆维吾尔族自治区聚为一类。结合表 3.8、

表 3.9 可知，产生的共同聚类结果为将江苏省、浙江省、山东省、广东省、四川省聚为一类；将北京市、河北省、安徽省、福建省、湖北省、湖南省聚为一类；将吉林省、黑龙江省、海南省、西藏自治区、甘肃省、青海省、宁夏回族自治区、新疆维吾尔族自治区聚为一类，综合而言，RS^{3}NMF 产生的聚类结果由东部到西部类别层次差异更加明显，呈现明显的区域差异，总体来说聚类结果更加准确。利用 ArcGIS 软件在地图上显示的聚类效果，见图 3.5。

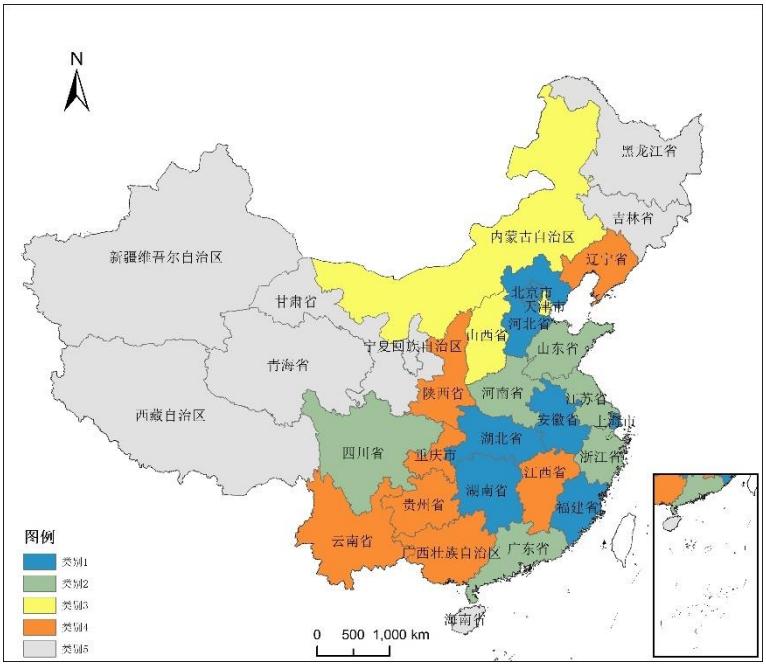


图 3.5 鲁棒自适应对称非负矩阵分解聚类结果

3.7 本章小结

本部分提出了基于$L_{2,1}$范数的鲁棒自适应对称非负矩阵分解聚类算法(RS^{3}NMF)，并给出了优化求解过程。使用$L_{2,1}$范数提高了模型的鲁棒性，使模型对噪声和异常值不敏感，保持了特征旋转不变性，增强了聚类性能。实验结果证实了 RS^{3}NMF 算法比 7 种先进的聚类算法具有更好的聚类性能。针对我国 31 省份 GDP 数据的实例应用表明，该鲁棒聚类算法对 GDP 数据的划分能够判断各省之间的发展差异，具有良好的实际应用价值。

4 鲁棒自适应学习判别对称非负矩阵分解算法

利用$L_{2,1}$范数测量重构误差，以确保鲁棒性。受空间聚类自表述学习方法的启发，由获得的表示系数[43]表示亲和矩阵(权重矩阵)，通过矩阵分解自适应学习的$\boldsymbol{W}$，保存原始数据固有的属性，相比集成学习的$\boldsymbol{W}$更灵活，能更好地适应模型。此外，结合文献[35,41]指出类指示矩阵可用于获得更好的判别能力。基于$L_{2,1}$范数，同时考虑鲁棒性、自适应学习和判别信息，将有助于提高模型的聚类效果。

4.1 目标函数

在考虑$L_{2,1}$范数对模型影响的同时考虑自适应学习、标签的判别信息，建立鲁棒自适应学习判别对称非负矩阵分解算法(Robust Adaptive Learning Discriminative Symmetric Nonnegative Matrix Factorization Algorithm，RADS^{3}NMF)，并给出模型的优化求解过程，建立的模型目标函数为

$$\begin{aligned}&\min_{\boldsymbol{V},\boldsymbol{W}} \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\|_{2,1}+\alpha\|\boldsymbol{W}-\boldsymbol{V}\boldsymbol{V}^T\|_{2,1}+\beta\|\boldsymbol{S}-\boldsymbol{A}\boldsymbol{V}\|_{2,1}\\&\text{s.t.}\boldsymbol{V}\geq 0,\ \boldsymbol{W}\geq 0,\ \text{diag}(\boldsymbol{W})=0,\ \boldsymbol{A}\geq 0\end{aligned} \tag{4.1}$$

式(4.1)中的第一项自适应学习$\boldsymbol{W}$，使获得的亲和矩阵能更好的适应模型，第二项和第三项保证了学习后的图能够保留数据的内在几何结构，并具有判别能力，同时利用$L_{2,1}$范数提升模型的鲁棒性。因此，学习到的图不仅符合其固有的几何结构，而且具有判别能力与鲁棒性。 其中$\boldsymbol{X}\in\mathbb{R}^{m\times n}$，$\boldsymbol{W}\in\mathbb{R}^{n\times n}$为学习的亲和矩阵，$\boldsymbol{V}\in\mathbb{R}^{n\times c}$是$c$类的聚类指示矩阵，$\boldsymbol{A}\in\mathbb{R}^{r\times n}$是一个非负矩阵。$\boldsymbol{S}\in\mathbb{R}^{r\times c}$为类指标矩阵，

$$S_{ij}=\begin{cases}1, & v_{ij}=\max\limits_{j} v_{ij}\\ 0, & \text{其它}\end{cases}$$

参数α、β是两个非负常数。

4.2 优化算法

下面我们利用乘法更新规则推导 RADS^{3}NMF 算法式(4.1)的更新公式。

$$\min_{\boldsymbol{V},\boldsymbol{W},\boldsymbol{A}} \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\|_{2,1}+\alpha\left\|\boldsymbol{W}-\boldsymbol{V}\boldsymbol{V}^T\right\|_{2,1}+\beta\|\boldsymbol{S}-\boldsymbol{A}\boldsymbol{V}\|_{2,1}$$

目标函数的拉格朗日函数为

$$\begin{aligned}\boldsymbol{L} &= \mathrm{tr}((\boldsymbol{X}-\boldsymbol{XW})\boldsymbol{P}(\boldsymbol{X}-\boldsymbol{XW})^T) + \alpha\mathrm{tr}((\boldsymbol{W}-\boldsymbol{VV}^T)\boldsymbol{Q}(\boldsymbol{W}-\boldsymbol{VV}^T)^T) \\ &+ \beta\mathrm{tr}((\boldsymbol{S}-\boldsymbol{AV})\boldsymbol{R}(\boldsymbol{S}-\boldsymbol{AV})^T) - \mathrm{tr}(\lambda_1\boldsymbol{V}^T) - \mathrm{tr}(\lambda_2\boldsymbol{W}^T) - \mathrm{tr}(\lambda_3\boldsymbol{A}^T)\end{aligned} \tag{4.2}$$

其中$\|\boldsymbol{X}-\boldsymbol{XW}\|_{2,1} = \mathrm{tr}\left((\boldsymbol{X}-\boldsymbol{XW})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{XW})^T\right)$，$\boldsymbol{P}$是对角矩阵，主要对角元素由以下公式给出

$$\boldsymbol{P}_{kk} = 1/\sqrt{\sum_{i=1}^{m}(\boldsymbol{X}-\boldsymbol{XW})_{ik}^2} = 1/(x_k - xw_k)^2 \tag{4.3}$$

$\boldsymbol{Q}$、$\boldsymbol{R}$的公式类似于$\boldsymbol{P}$，$\mathrm{tr}(\boldsymbol{XY}) = \mathrm{tr}(\boldsymbol{YX})$，$\mathrm{tr}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X})^T$，令$\lambda_1, \lambda_2, \lambda_3$分别为$\boldsymbol{V}$，$\boldsymbol{A}$，$\boldsymbol{W}$的拉格朗日乘数，式(4.3)展开为

$$\begin{aligned}\boldsymbol{L}_f &= \mathrm{tr}(\boldsymbol{XPX}^T) - 2\mathrm{tr}(\boldsymbol{XPW}^T\boldsymbol{X}^T) + \mathrm{tr}(\boldsymbol{XWPW}^T\boldsymbol{X}^T) \\ &+ \alpha\mathrm{tr}(\boldsymbol{WQW}^T) - 2\alpha\mathrm{tr}(\boldsymbol{WQVV}^T) + \alpha\mathrm{tr}(\boldsymbol{VV}^T\boldsymbol{QVV}^T) \\ &+ \beta\mathrm{tr}(\boldsymbol{RS}^T) - 2\beta\mathrm{tr}(\boldsymbol{SRV}^T\boldsymbol{A}^T) + \beta\mathrm{tr}(\boldsymbol{AVRV}^T\boldsymbol{A}^T) \\ &- \mathrm{tr}(\lambda_1\boldsymbol{V}^T) - \mathrm{tr}(\lambda_2\boldsymbol{W}^T) - \mathrm{tr}(\lambda_3\boldsymbol{A}^T)\end{aligned} \tag{4.4}$$

关于$\boldsymbol{V}$、$\boldsymbol{A}$、$\boldsymbol{W}$的项分别为

$$\begin{aligned}L_v &= -2\alpha\mathrm{tr}(\boldsymbol{WQVV}^T) + \alpha\mathrm{tr}(\boldsymbol{VV}^T\boldsymbol{QVV}^T) - 2\beta\mathrm{tr}(\boldsymbol{SRV}^T\boldsymbol{A}^T) \\ &+ \beta tr(\boldsymbol{AVRV}^T\boldsymbol{A}^T) - \mathrm{tr}(\lambda_1\boldsymbol{V}^T)\end{aligned}$$

$$L_A = -2\beta\mathrm{tr}(\boldsymbol{SRV}^T\boldsymbol{A}^T) + \beta\mathrm{tr}(\boldsymbol{AVRV}^T\boldsymbol{A}^T) - \mathrm{tr}(\lambda_3\boldsymbol{A}^T)$$

$$\begin{aligned}L_w &= -2\mathrm{tr}(\boldsymbol{XPW}^T\boldsymbol{X}^T) + \mathrm{tr}(\boldsymbol{XWPW}^T\boldsymbol{X}^T) + \alpha\mathrm{tr}(\boldsymbol{WQW}^T) \\ &- 2\alpha\mathrm{tr}(\boldsymbol{WQVV}^T) - \mathrm{tr}(\lambda_2\boldsymbol{W}^T)\end{aligned}$$

分别求偏导，令其为 0，得

$$\lambda_1 = 2(\alpha\boldsymbol{VV}^T\boldsymbol{QV} + \alpha\boldsymbol{QVV}^T\boldsymbol{V} + \beta\ \boldsymbol{A}^T\boldsymbol{AVR} - \alpha\boldsymbol{WQV} - \beta\boldsymbol{A}^T\boldsymbol{SR}) \tag{4.5}$$

$$\lambda_2 = 2(\boldsymbol{X}^T\boldsymbol{XWP} + \alpha\boldsymbol{WQ} - \boldsymbol{XX}^T\boldsymbol{P} - \alpha\boldsymbol{V}\ \boldsymbol{V}^T\boldsymbol{Q}) \tag{4.6}$$

$$\lambda_3 = 2\beta(\boldsymbol{AVRV}^T - \boldsymbol{SR}\ \boldsymbol{V}^T) \tag{4.7}$$

由 KKT 条件：$\lambda_1 \odot \boldsymbol{V} = 0$，$\lambda_2 \odot \boldsymbol{W} = 0$，$\lambda_3 \odot \boldsymbol{A} = 0$。

因此$\boldsymbol{V}$，$\boldsymbol{A}$，$\boldsymbol{W}$，的更新规则如下：

$$\boldsymbol{W}_{ij} \longleftarrow \boldsymbol{W}_{ij}\frac{\left(\boldsymbol{X}^T\boldsymbol{XP} + \alpha\boldsymbol{VV}^T\boldsymbol{Q}\right)_{ij}}{\left(\boldsymbol{X}^T\boldsymbol{XWP} + \alpha\boldsymbol{WQ}\right)_{ij}} \tag{4.8}$$

$$\boldsymbol{V}_{jk} \longleftarrow \boldsymbol{V}_{jk}\frac{\left(\alpha\boldsymbol{QW}^T\boldsymbol{V} + \alpha\boldsymbol{WQV} + \beta\boldsymbol{A}^T\boldsymbol{SR}\right)_{jk}}{\left(\alpha\boldsymbol{VV}^T\boldsymbol{QV} + \alpha\boldsymbol{QVV}^T\boldsymbol{V} + \beta\boldsymbol{A}^T\boldsymbol{AVR}\right)_{jk}} \tag{4.9}$$

$$\boldsymbol{A}_{kk} \longleftarrow \boldsymbol{A}_{kk}\frac{\left(\boldsymbol{SRV}^T\right)_{kk}}{\left(\boldsymbol{AVRV}^T\right)_{kk}} \tag{4.10}$$

4.3 算法流程

RADS^{3}NMF 算法流程见表 4.1。

表 4.1 RADS^{3}NMF 算法表

算法 2 RADS^{3}NMF 算法

输入：数据矩阵$\boldsymbol{X} \in \mathbb{R}^{m\times n}$，类指示矩阵$\boldsymbol{S} \in \mathbb{R}^{r\times c}$，参数$\alpha$、$\beta$。

初始化：随机初始化三个非负矩阵$\boldsymbol{W} \in \mathbb{R}^{n\times n}$，$\boldsymbol{A} \in \mathbb{R}^{r\times n}$，$\boldsymbol{V} \in \mathbb{R}^{n\times c}$，$\mathrm{diag}(\boldsymbol{W}) = 0$

重复：

根据式(4.8)更新$\boldsymbol{W}$

根据式(4.9)更新$\boldsymbol{A}$

根据式(4.10)更新$\boldsymbol{V}$

直到收敛

输出：$\boldsymbol{W}$、$\boldsymbol{A}$、$\boldsymbol{V}$

4.4 算法收敛性与复杂性

在定理证明过程中，矩阵计算的相关知识可以参考参考书[44]。为了证明定理，需要证明目标函数在式(4.8)、式(4.9)、式(4.10)更新步骤下是不增加的。对于目标函数，如果更新$\boldsymbol{W}$，需要固定$\boldsymbol{V}$和$\boldsymbol{A}$；更新$\boldsymbol{V}$，需要固定$\boldsymbol{W}$和$\boldsymbol{A}$；更新$\boldsymbol{A}$，需要固定$\boldsymbol{W}$和$\boldsymbol{V}$。因此，RADS^{3}NMF 中可以像求解 NMF 一样给出$\boldsymbol{W}$、$\boldsymbol{V}$和$\boldsymbol{A}$的更新公式，可以使用 NMF 的收敛性证明来证明目标函数在式(4.8)、式(4.10)中的更新步骤下是不增加的。这些细节可以在文献[10]中找到。因此证明目标函数在式(4.9)中的更新步骤下是不增加的。遵循文献[10]中构造辅助函数证明的类似过程。下面给出了辅助函数的定义。

定义 1 如果G是F的辅助函数，则F在如下的更新中不增。

$$v^{(t+1)} = \arg\min_{v} G\left(v, v^{(t)}\right)$$

其中$F\left(v^{(t+1)}\right) \leq G\left(v^{(t+1)}, v^{(t)}\right) \leq G\left(v^{(t)}, v^{(t)}\right) = F\left(v^{(t)}\right)$。

现在，证明式(4.9)中$\boldsymbol{V}$的更新步骤正是辅助函数中的更新，具有适当的辅助函数。考虑到$\boldsymbol{V}$中的任何元素v_{ab}，使用F_{ab}来表示目标函数中仅与v_{ab}相关的部分。容易得到以下导数：

$$F'_{ab}=\left(\frac{\partial \boldsymbol{L}_v}{\partial \mathbf{V}}\right)_{ab}=(2\alpha \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{V}+2\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{V}+2\beta\ \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{R}$$
$$-2\alpha \boldsymbol{Q}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{V}-2\alpha \boldsymbol{W}\boldsymbol{Q}\boldsymbol{V}-2\beta \boldsymbol{A}^T\boldsymbol{S}\boldsymbol{R})_{ab}$$
$$F''_{ab}=2(2\alpha \boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{V}+3\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T+\alpha \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{Q}$$
$$+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{R}-\alpha \boldsymbol{W}\boldsymbol{Q}-\alpha \boldsymbol{Q}\boldsymbol{W}^T)_{ab}$$

可证每个F_{ab}在式(4.8)的更新步骤下不增加。因此，引入以下引理。

引理 1 函数

$$G\left(v,v_{ab}^{(t)}\right)=F_{ab}\left(v_{ab}^{(t)}\right)+F'_{ab}\left(v_{ab}^{(t)}\right)\left(v-v_{ab}^{(t)}\right)$$
$$+\frac{\left(2\alpha \mathbf{V}\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{V}+3\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{V}+\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{R}\boldsymbol{V}\right)_{ab}}{v_{ab}^{(t)}}\left(v-v_{ab}^{(t)}\right)^2 \tag{4.11}$$

是F_{ab}的辅助函数。

证明 只需要证明$G\left(v,v_{ab}^{(t)}\right)\geq F_{ab}(v)$，因为$G(v,v)=F_{ab}(v)$是显而易见的。因此，首先考虑$F_{ab}(v)$的泰勒级数展开。

$$F_{ab}(v)=F_{ab}\left(v_{ab}^{(t)}\right)+F'_{ab}\left(v_{ab}^{(t)}\right)\left(v-v_{ab}^{(t)}\right)+[(2\alpha \boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{V}+3\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T$$
$$+\alpha \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{Q}+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{R}-\alpha \boldsymbol{W}\boldsymbol{Q}-\alpha \boldsymbol{Q}\boldsymbol{W}^T)_{ab}]\left(v-v_{ab}^{(t)}\right)^2 \tag{4.12}$$

比较式(4.12)和式(4.11)，发现$G\left(v,v_{ab}^{(t)}\right)\geq F_{ab}(v)$等价于

$$\frac{\left(2\alpha \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\mathbf{Q}\mathbf{V}+3\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}+\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}+\beta \boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{V}\right)_{ab}}{v_{ab}^{(t)}}$$
$$\geq\left(2\alpha \boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{V}+3\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T+\alpha \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{Q}+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{R}-\alpha \boldsymbol{W}\boldsymbol{Q}-\alpha \boldsymbol{Q}\boldsymbol{W}^T\right)_{ab}$$

对构造的辅助函数进行求导，令其等于 0，则

$$v_{ab}^{(t+1)}=v_{ab}^{(t)}-v_{ab}^{(t)}\frac{F'_{ab}\left(v_{ab}^{(t)}\right)}{2\left(\alpha \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}+\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{V}+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{R}\right)_{ab}}$$
$$=v_{ab}^{(t)}\frac{\left(\alpha \boldsymbol{Q}\boldsymbol{W}^T\boldsymbol{V}+\alpha \boldsymbol{W}\boldsymbol{Q}\boldsymbol{V}+\beta \boldsymbol{A}^T\boldsymbol{S}\boldsymbol{R}\right)_{ab}}{\left(\alpha \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{V}+\alpha \boldsymbol{Q}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{V}+\beta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{R}\right)_{ab}}$$

因为式(4.11) 是一个辅助函数，并且F_{ab}在这个更新规则下是不增加的，所以整个模型是收敛的。

进一步讨论算法的复杂度。设最大迭代次数为τ，n是总样本量，c是聚类数，k表示初始化的次数。对于算法 2，交替迭代求解$\boldsymbol{W}$、$\boldsymbol{V}$、$\boldsymbol{A}$的子问题，计算复杂度为$O(nck\tau)$。因此，算法 2 每次迭代的复杂度为$O(nck\tau)$。相比参考文献[35]中模型算法复杂度，本文提出的方法算法复杂度没有增加。

4.5 实验

实验所用数据集为 6 个公开数据集，在 5 个聚类指标上将所提出的 RADS^{3}NMF 算法与 7 种具有代表性的聚类算法进行比较。设置参数为$\alpha = 20$、$\beta = 0.005$。

表 4.2 实验使用的数据集

数据集	样本总数(n)	特征维度(k)	类别数(c)	数据类型
IRIS	150	4	3	鸢尾花卉数据
TIMI*	300	256	5	语音数据集
Letters*	300	16	26	手写字母识别数据集
Pendigits*	300	16	10	手写样本数字数据库
SEEDS	210	7	3	小麦种子数据
WINE	150	3	3	葡萄酒数据

注：TIMI*、Letters*、Pendigits*为从原始数据集抽取的300个样本。

在 IRIS、TIMI*、Letters*、Pendigits*、SEEDS、WINE 数据集合的聚类指标如表 4.3、表 4.4、表 4.5、表 4.6、表 4.7 和表 4.8 所示。

表 4.3 IRIS 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.8933	0.7515	0.8933	0.7302	0.8207
NMF	0.7267	0.5893	0.7267	0.5099	0.6713
GNMF	0.9067	0.7696	0.9067	0.7576	0.8384
SC	0.7333	0.4434	0.7333	0.3347	0.5830
RNMF	0.7067	0.5841	0.7067	0.5072	0.6735
SNMF	0.8733	0.7536	0.8733	0.6956	0.8004
S^{3}NMF	0.7533	0.5398	0.7533	0.4775	0.6575
RADS^{3}NMF	**0.9533**	**0.8559**	**0.9267**	**0.8683**	**0.9117**

表 4.4 TIMI*数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.4046	0.6857	0.4046	0.6857	0.3794
NMF	0.8809	0.6507	0.8808	0.6850	0.7893
GNMF	0.8095	0.6116	0.8095	0.5535	0.7050
SC	0.8285	0.6337	0.8286	0.5862	0.7260
RNMF	0.6714	0.3091	0.6714	0.3061	0.5370
SNMF	0.7571	0.4916	0.7571	0.4004	0.6212
S^3NMF	0.8300	0.8117	0.8300	0.7488	0.8004
RADS^{3}NMF	**0.8900**	**0.7999**	**0.8900**	**0.7751**	**0.8203**

表 4.5 Letters*数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.3133	0.4724	0.3633	0.1178	0.1567
NMF	0.3000	0.4491	0.3233	0.1043	0.1434
GNMF	0.2900	0.4704	0.3233	0.1095	0.1470
SC	0.3033	0.4804	0.3400	0.1197	0.1556
RNMF	0.3067	0.4515	0.3267	0.1076	0.1489
SNMF	0.3167	0.5154	0.3600	0.1422	0.1754
S^3NMF	0.3133	0.4602	0.3267	0.1405	0.1755
RADS^{3}NMF	**0.3900**	**0.5439**	**0.4167**	**0.1972**	**0.2280**

表 4.6 Pendigits*数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.6967	0.7089	0.7400	0.5603	0.6066
NMF	0.6867	0.6928	0.7167	0.5441	0.5919
GNMF	0.6067	0.6961	0.6700	0.5173	0.5729
SC	0.6100	0.6519	0.6933	0.4306	0.5012
RNMF	0.6333	0.6786	0.6767	0.4942	0.5501
SNMF	0.7033	0.7374	0.7367	0.5612	0.6074
S^3NMF	0.7800	0.7085	0.7800	0.6788	0.7124
RADS^{3}NMF	**0.8367**	**0.7863**	**0.8367**	**0.7117**	**0.7410**

表 4.7 WINE 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.7023	0.4287	0.7023	0.3711	0.5835
NMF	0.6407	0.3084	0.6629	0.3424	0.5935
GNMF	0.7023	0.4224	0.7023	0.3598	0.5762
SC	0.6854	0.3664	0.6854	0.3101	0.5476
RNMF	0.5337	0.3539	0.6517	0.3178	0.5712
SNMF	0.5449	0.3466	0.6573	0.2891	0.5816
S^3NMF	0.5933	0.2135	0.5955	0.1096	0.4688
RADS^{3}NMF	**0.6854**	**0.3835**	**0.6854**	**0.3452**	**0.5703**

表 4.8 SEEDS 数据集上的聚类结果

方法	ACC	NMI	PUR	ARI	F1-score
K-means	0.6857	0.4046	0.6857	0.3794	0.5857
NMF	0.8809	0.6508	0.8890	0.6850	0.7893
GNMF	0.8095	0.6116	0.8095	0.5535	0.7050
SC	0.8286	0.6337	0.8286	0.5862	0.7259
RNMF	0.6714	0.3091	0.6714	0.3061	0.5370
SNMF	0.7571	0.4916	0.7571	0.4004	0.6212
S^3NMF	0.8810	0.6670	0.8810	0.6880	0.7920
RADS^{3}NMF	**0.8762**	**0.6420**	**0.8761**	**0.6744**	**0.7819**

结合表 4.3、表 4.4、表 4.5、表 4.6、表 4.7、表 4.8 可知：

(1) RADS^{3}NMF 算法整体优于其它算法，特别在 IRIS 和 TIMI*、Letters*、Pendigits* 数据集上 5 个聚类评价指标结果有明显改进，更具鲁棒性。例如，在 IRIS 数据集上，相比 S^3NMF，将 ACC 值从 0.7533 提高到 0.9533，NMI 值从 0.5398 增加到 0.8559，PUR、ARI 和 F1-score 的值分别提高了 0.1734、0.3908 和 0.2542；在 TIMI*数据集上，将 ACC 值从 0.83 提高到 0.89，PUR 值提高了 7%，ARI 值从 0.7488 增加到 0.7751，增加了 0.0263，提高了 3.5%，F1-score 值从 0.8004 增加到 0.8203，提高了 2.49%；在 Letters*集上，将 NMI 值从 0.5154 提高到 0.5439，提高了 5.5%，PUR 从 0.3633 提高到 0.4167，提高了 14.7%；在 Pendigits*数据集上，将 ACC 值从 0.78 提高到 0.8367，提高了 7.3%，NMI

值从 0.7374 提高到 0.7863，提高了 6.6%，F1-score 值从 0.7124 提高到 0.7410，提高了 4%。

(2) 与鲁棒聚类算法 RNMF 相比，RADRS^{3}NMF 算法的优势也很明显。例如在 IRIS、TIMI*、Letters*、Pendigits*数据集上，RADRS^{3}NMF 的 ACC 值分别提升了 35%、33%、23.1%、15.6%。这意味着 RADS^{3}NMF 算法对噪声和异常值具有十分良好的鲁棒性。

(3) 与图聚类算法 SC 和 SNMF 相比，RS^{3}NMF 算法的聚类性能也显著提升。例如，在 IRIS 数据集上，与 SC 算法比较，ACC 值提高了 30%。在 TIMI*数据集上，与 SNMF 算法比较，ACC 值提高了 8%；在 Letters*数据集上，ACC 提高了 28.5%。Pendigits*数据集上，ACC 提高了 19%。

(4) 在 SEEDS 数据集上，RADS^{3}NMF 的聚类指标跟 NMF、 RS^{3}NMF，评价指标比较接近，高于其它指标。在 WINE 数据集上，跟 K-means、GNMF 相比数值比较接近，但高于其它方法的指标。综合 6 个数据集，总的来说，相比其他 7 种聚类算法，RADS^{3}NMF 算法在这 6 个数据集上能产生较好的聚类性能，一定程度上提升了模型的鲁棒性。

4.6 消融性分析

在本节中，我们评估了所提出方法的不同组成部分的重要性。具体地说，取$\lambda=0$、$\beta=0$、$\alpha=15,\beta=0.01$与本文参数下的指标(α、β是拉格朗日乘子，文中的α、β参数取值分别为 20、0.005，算法迭代次数为 500 次)进行比较，分析模型在不同系数取值下的评价指标值，具体结果见下表 4.9、表 4.10、表 4.11、表 4.12、表 4.13 和表 4.14。

表 4.9 IRIS 消融性分析结果表

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.9533**	**0.8559**	**0.9267**	**0.8683**	**0.9117**
Alpha=0	0.6800	0.5816	0.6800	0.5539	0.7312
beta =0	0.3333	0.0131	0.3333	0.1762	0.4949
Alpha=15 beta=0.01	0.7400	0.6169	0.7400	0.5596	0.7271

表 4.10 TIMI* 数据消融性分析结果表

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.8900**	**0.7999**	**0.8900**	**0.7751**	**0.8203**
Alpha=0	0.2767	0.0211	0.2800	0.0048	0.2069
beta =0	0.8633	0.7816	0.8633	0.7414	0.7932
Alpha=15 beta=0.01	0.8767	0.7820	0.8767	0.7546	0.8038

表 4.11 Letters* 数据消融性分析结果表

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.3900**	**0.5439**	**0.4167**	**0.1972**	**0.2280**
Alpha=0	0.0567	0.0780	0.0567	0.0000	0.0783
beta =0	0.3467	0.5066	0.3667	0.1432	0.1758
Alpha=15 beta=0.01	0.3367	0.5185	0.3700	0.1501	0.1827

表 4.12 Pendigits* 数据消融性分析结果表

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.3900**	**0.5439**	**0.4167**	**0.1972**	**0.2280**
Alpha=0	0.0567	0.0780	0.0567	0.0000	0.0783
beta =0	0.3467	0.5066	0.3667	0.1432	0.1758
Alpha=15 beta=0.01	0.3367	0.5185	0.3700	0.1501	0.1827

表 4.13 WINE 数据集消融性分析结果表

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.6854**	**0.3835**	**0.6854**	**0.3452**	**0.5703**
Alpha=0	0.3989	0.0114	0.3989	0.0148	0.5052
beta =0	0.6631	0.3610	0.6685	0.3194	0.5494
Alpha=15 beta=0.01	0.6798	0.3606	0.6798	0.3289	0.5615

表 4.14 SEEDS 数据集

方法	ACC	NMI	PUR	ARI	F1-score
RADS^{3}NMF	**0.8762**	**0.6420**	**0.8761**	**0.6744**	**0.7819**
Alpha=0	0.3333	0.0094	0.3333	0.0048	0.4964
beta =0	0.7381	0.4603	0.7381	0.4315	0.6283
Alpha=15 beta=0.01	0.6762	0.3720	0.6762	0.3671	0.5816

综合表 4.9、表 4.10、表 4.11、表 4.12、表 4.13 和表 4.14 可知，当模型缺少判别信息或者数据结构部分时，模型的评价指标相对比较小，此外，在其余不同参数下，模型的评价指标并不是最优，即模型缺失判别信息或者自适应学习的亲和矩阵分解部分，模型的评价指标 ACC、NMI、PUR、ARI、F_1-score 均减少，分开的模型效果并不是最优结果，即模型中的每一部分都起着重要作用，不可缺失。基于此，综合考虑$L_{2,1}$范数、自适应学习和判别信息的模型能够很好的增强模型的性能。

4.7 实证应用

本文数据来自于北京市 35 个空气质量监测站点(http://www.bjmemc.com.cn/)，包含 2018 年 1 月 1 日至 2018 年 12 月 31 日各监测站点的 NO_2 污染物小时浓度($\mu g/m^3$) 数据。35 个站点根据职能被分为四类，有一个站点为“城市清洁对照站点”，因此在聚类的时候将其删除，各个监测站点的经纬度信息[45]见表 4.15。利用 RADS^{3}NMF 进行聚类分析，将北京市 35 个空气质量监测站点聚为 3 类，聚类结果见表 4.16。

表 4.15 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)
城市环境评价点	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)
	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）				

表 4.16 RADS^{3}NMF 聚类结果分类表

类别	站点名称
第一类	西城官园、奥体中心、海淀万柳、丰台花园、古城、前门、西直门、东四环、永定门、南三环
第二类	云岗、顺义新城、双峪（门头沟）、密云镇、夏都（延庆）、京西北（八达岭）、怀柔镇、京东北
第三类	东四、天坛、万寿西宫、农展馆、北部新区、香山（植物园）、良乡（房山）、黄村（大兴）、亦庄、通州北苑、昌平镇、平谷镇、京东（东高村）、京东南、京南（榆垡）、京西南

由表 4.16 可知，利用 RADS^{3}NMF 将 34 个监测站点聚为 3 类，西城官园、奥体中心、海淀万柳、丰台花园、古城、前门、西直门、东四环、永定门、南三环聚为一类；云岗、顺义新城、双峪（门头沟）、密云镇、夏都（延庆）、京西北（八达岭）、怀柔镇、京东北聚为一类；东四、天坛、万寿西宫、农展馆、北部新区、香山（植物园）、良乡（房山）、黄村（大兴）、亦庄、通州北苑、昌平镇、平谷镇、京东（东高村）、京东南、京南（榆垡）、京西南聚为一类。类别 1 包含所有的交通控制污染点，类别 2 包含大部分区域背景传输点，类别 3 包含大部分城市环境评价点，利用 ArcGIS 在地图上显示结果如图 4.1 所示，可以看出 RADS^{3}NMF 聚类结果整体呈现上中下的分布，结合文献[45]的聚类结果，可以得出 RADS^{3}NMF 整体聚类相对比较准确，但相对实际站点存在差异，可能是实际样点数目太少或过于分散导致。总体而言 RADS^{3}NMF 利用 35 个监测站点进行聚类分析能够产生相对比较好的聚类效果。

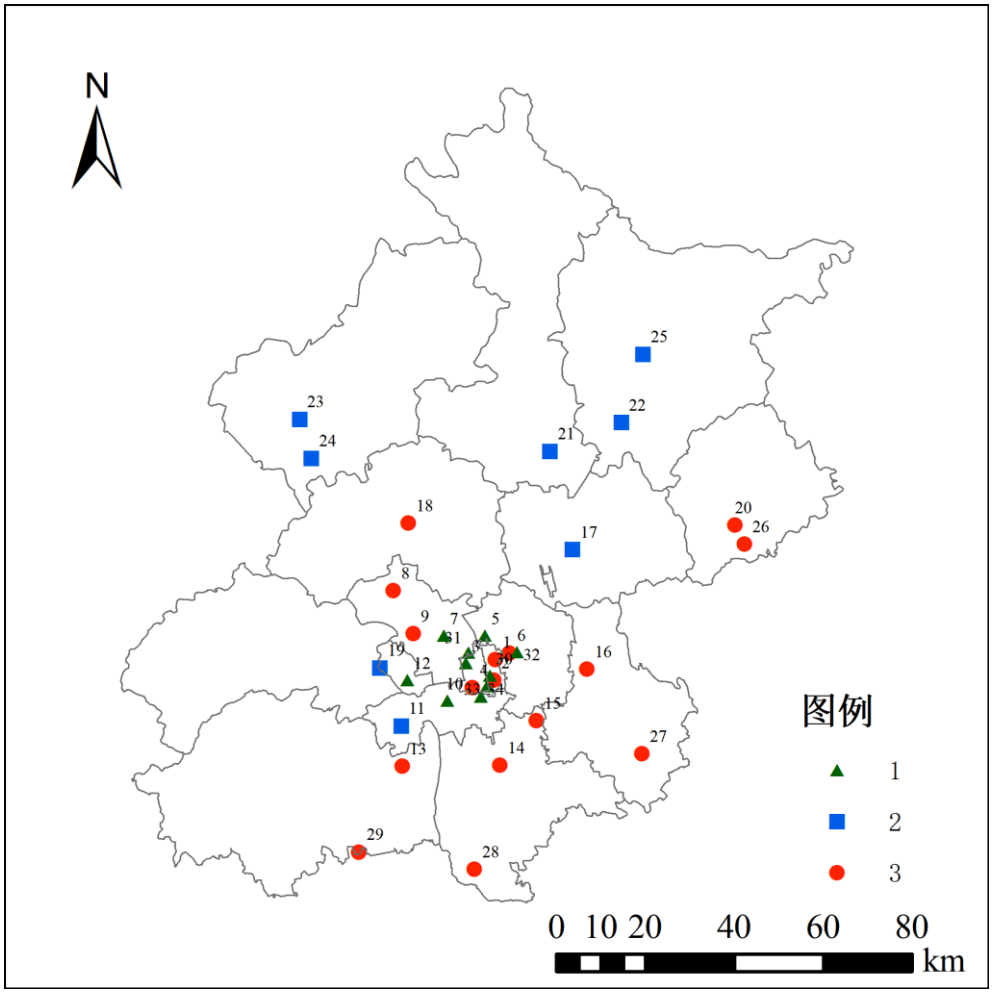


图 4.1 各监测点聚类结果空间展示图

4.8 本章小结

本章在考虑$L_{2,1}$范数重构误差的基础上，考虑自适应学习亲和矩阵，对自适应学习的亲和矩阵进行分解，保证数据的本身结构信息，同时加入判别信息，建立 RADS^{3}NMF 算法，并进行优化求解，构造辅助函数，证明模型的收敛性，以及给出模型的算法复杂度。最后利用北京市二氧化氮(NO_2)污染物小时浓度数据，将该算法应用于北京市空气质量监测站点布设聚类分析，结果显示 RADS^{3}NMF 算法能够较好地识别空气质量监测站点的空间布局，具有良好的适用性。

5 研究的总结及展望

5.1 主要结论

现实生活中的大多数据，存在一定的异常值，即数据存在一定的噪声，为了降低噪声数据对模型的影响，本文提出了两项主要工作：

(1) 构建基于$L_{2,1}$范数的 RS^{3}NMF。RS^{3}NMF 算法利用 SNMF 对初始化特征敏感的特点，结合$L_{2,1}$范数提高模型的鲁棒性，缓解噪声和异常值的影响，进一步提高聚类性能。此外，采用交替迭代优化算法推导 RS^{3}NMF 的乘法更新规则，并分析算法的收敛性和计算复杂度。实验结果证实RS^{3}NMF算法比7种先进的聚类算法具有更好的聚类精度，并将算法应用于我国 31 省份 GDP 数据聚类分析。

(2) 构建同时考虑鲁棒性、自适应学习和判别信息的 RADS^{3}NMF 聚类算法，进一步提高模型的聚类性能。此外，采用交替迭代优化算法推导 RADS^{3}NMF 的乘法更新规则，构造辅助函数分析算法的收敛性和计算复杂度。实验结果验证 RADS^{3}NMF 算法相比 7 种先进的聚类算法能产生较好的聚类性能。对北京市空气质量监测站点聚类开展应用，验证了 RADS^{3}NMF 算法的可行性与合理性。

5.2 展望

尽管所建立的RS^{3}NMF算法、RADS^{3}NMF算法在聚类任务中表现出了良好的效果，但未来还可以做一些改进：

首先，图正则化采用拉普拉斯矩阵分解，充分考虑了数据的内在局部几何结构，进一步探索数据的内在局部几何结构，将数据的局部结构并入RS^{3}NMF和RADS^{3}NMF，考虑用于聚类的图正则化RS^{3}NMF算法和RADS^{3}NMF算法。

其次，本文主要采用了鲁棒的$L_{2,1}$范数损失函数，进一步可采用相关熵作为相似度度量来降低噪声和异常值的影响，构建基于相关熵的自适应SNMF算法。

最后，尝试将RS^{3}NMF算法和RADS^{3}NMF算法推广到解决多视角聚类问题和函数型聚类问题。

参考文献

[1] Shapiro P N, Penrod S. Meta-analysis of facial identification studies[J]. Psychological bulletin, 1986, 100(2): 139.

[2] Welbers K, Van Atteveldt W, Benoit K. Text analysis in R[J]. Communication Methods and Measures, 2017, 11(4): 245-265.

[3] Jain A K, Duin R P W, Mao J. Statistical pattern recognition: A review[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(1): 4-37.

[4] Bryant E C, Center on national security at fordham law. statistical analysis[M]. New York: McGraw-Hill, 1966.

[5] Hu Z, Pan G, Wang Y, et al. Sparse principal component analysis via rotation and truncation [J]. IEEE transactions on neural networks and learning systems, 2016, 27(4): 875-890.

[6] Treder MS, Porbadnigk AK, Avarvand FS, et al. The LDA beamformer: optimal estimation of ERP source time series using linear discriminant analysis [J]. NeuroImage, 2016, 129: 279-291.

[7] Matteson DS, Tsay RS. Independent component analysis via distance covariance [J]. Journal of the American Statistical Association, 2017: 1-16.

[8] Zhang Q, Wang Y, Levine MD, et al. Multisensor video fusion based on higher order singular value decomposition[J]. Information Fusion, 2015, 24: 54-71.

[9] Ergul E. Relative attribute based incremental learning for image recognition [J]. CAAI Transactions on Intelligence Technology, 2017, 2(1): 1-11.

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

[11] Ding C, He X, Simon H D. On the equivalence of nonnegative matrix factorization and spectral clustering[J]. Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining. 2005(5): 606-610.

[12] Wang K, Liao R J, Yang L J, et al. Nonnegative matrix factorization aided principal component analysis for high-resolution partial discharge image compression in transformers[J]. Int'l. Rev. Electr. Eng., 2013, 8(1): 479-490.

[13]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.

[14]Ge S, Li H, Luo L. Constrained dual graph regularized orthogonal nonnegative matrix tri-Factorization for co-clustering[J]. Mathematical Problems in Engineering, 2019, 2019(1):1-17.

[15]Lu Z, Liu G,Wang S . Sparse neighbor constrained co-clustering via category consistency learning[J]. Knowledge-Based Systems, 2020, 201-202(9):105987.

[16]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.

[17]Zong L, Zhang X, Zhao L, et al. Multi-view clustering via multi-manifold regularized non-negative matrix factorization[J]. Neural Networks, 2017, 88: 74-89.

[18]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(08):91-103.

[19]Kuang D, Ding C, Park H. Symmetric nonnegative matrix factorization for graph clustering[C]//Proceedings of the 2012 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2012: 106-117.

[20]Kuang D, Yun S, Park H. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering[J]. Journal of Global Optimization. 2015, 62(3): 545-574.

[21]He Z, Xie S, Zdunek R, et al. Symmetric nonnegative matrix factorization: algorithms and applications to probabilistic clustering[J]. IEEE Transactions on Neural Networks. 2011, 22(12): 2117-2131.

[22]Zhang X, Wang Z, Zong L, et al. Multi-view clustering via graph regularized symmetric nonnegative matrix factorization[C]//IEEE International Conference on Cloud Computing and Big Data Analysi. IEEE, 2016: 109-114.

[23]Gao Z, Guan N, Su L. Graph regularized symmetric non-negative matrix factorization for graph clustering[C]//IEEE International Conference on Data Mining Workshops. IEEE, 2018: 379-384.

[24]Jia Y, Liu H, Hou J, et al. Clustering-aware graph construction: a joint learning

perspective[J]. IEEE Transactions on Signal and Information Processing over Networks. 2020, 6: 357-370.

[25] Yang L, Cao X, Jin D, et al. A Unified semi-supervised community detection framework using latent space graph regularization[J]. IEEE Transactions on Cybernetics. 2015, 45(11): 2585-2598.

[26] Al-Stouhi S, Reddy C K. Multi-task clustering using constrained symmetric non-negative matrix factorization[C]//Proceedings of the 2014 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2014: 785-793.

[27] Jia Y, Liu H, Hou J, et al. Self-supervised symmetric nonnegative matrix factorization[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(7): 4526-4537.

[28] Kong D, Ding C, Huang H. Robust nonnegative matrix factorization using $L_{2,1}$-norm[C]//Proceedings of the 20th ACM international conference on Information and knowledge management. 2011: 673-682.

[29] Huang J, Nie F, Huang H, et al. Robust manifold nonnegative matrix factorization[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2014, 8(3): 1-21.

[30] Wu B, Wang E, Zhu Z, et al. Manifold NMF with $L_{2,1}$ norm for clustering[J]. Neurocomputing, 2018, 273: 78-88.

[31] 蒋茂松,王冬霞,牛芳琳,曹玉东.稀疏正则非负矩阵分解的语音增强算法[J].计算机应用,2018,38(04):1176-1180.

[32] Ng A, Jordan M, Weiss. Y. On spectral clustering: Analysis and an algorithm [C]// Advances in neural information processing systems. MA: MIT Press, 2002: 849-856.

[33] Yang S, Hou C, Zhang C, et al. Robust non-negative matrix factorization via joint sparse and graph regularization for transfer learning[J]. Neural Computing and Applications, 2013, 23(2): 541-559.

[34] Huang J, Nie F, Huang H, et al. Robust manifold nonnegative matrix factorization[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2014, 8(3): 1-21.

[35] Long X, Lu H, Peng Y, et al. Graph regularized discriminative non-negative matrix f actorization for face recognition[J]. Multimedia Tools and Applications. 2014, 72(3): 2679-2699.

[36] Liu G, Ge H, Wang S. Robust semi non-negative low-rank graph embedding algorithm via the L21 norm[J]. Applied Intelligence, 2022, 52(8): 8708-8720.

[37] Cai D, He X, Han J. Document clustering using locality preserving indexing[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(12): 1624-1637.

[38] Lovász L, Plummer M D. Matching theory[M]. American Mathematical Soc, 2009.

[39] Hubert L, Arabie P. Comparing partitions[J]. Journal of classification, 1985, 2: 193-218.

[40] 李锋. 信任函数理论框架下新聚类分析方法的研究[D]. 北京工业大学,2020.DOI:10.26935/d.cnki.gbjgu.2020.000061.

[41] Schütze H, Manning C D, Raghavan P. Introduction to information retrieval[M]. Cambridge: Cambridge University Press, 2008.

[42] Ng A, Jordan M, Weiss. Y. On spectral clustering: Analysis and an algorithm [C]// Advances in neural information processing systems. MA: MIT Press, 2002: 849-856.

[43] Li C G, You C, Vidal R. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework[J]. IEEE Transactions on Image Processing, 2017, 26(6): 2988-3001.

[44] Petersen K B, Pedersen M S. The matrix cookbook[J]. Technical University of Denmark, 2008, 7(15): 510.

[45] 姚晓红,高海燕,吕家奇,黄恒君.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(04):689-702.

攻读硕士学位期间所发表的论文

[1] 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.

[2] 刘万金,赵芳芳.自监督对称非负矩阵在 GDP 聚类分析中的应用[J].甘肃科技纵横,2022,51(07):69-73.

致谢

慕然回首，时光如梭，研究生三年仿佛如在昨日。研一准备分享论文的焦虑与慌张、面对新知识的无知，研二写论文的惆怅与参加比赛的匆忙，些许遗憾的就是研三，疫情影响下学习稍有懈怠。回首三年，倍感收获：从阅读文献时的毫无头绪，到逐渐具有清晰的研究思路；从论文投稿、修回、接受到发表，不断学习积累终有成果；从竞赛作品的反复打磨到获奖，得到了宝贵的学习经验和实践积累；从精心准备简历、不懈努力面试到顺利收到 offer，最终找到了心仪的满意工作。在此，非常感谢曾经给予我指导和鼓励的老师、同学和亲朋好友们，对他们充满了感激之情，在此向他们表达我最诚挚的感谢！

首先，特别感谢我的导师高海燕教授。记得研一时，刚入学就要正式确定导师，听师姐师兄说高老师是一位认真负责的老师，除了上课之余，常在图书馆碰到高老师在自习，而且上课也很敬业，于是，我便毫不犹豫的选择了高老师作为导师。一开始读文献、写论文时，面对高出本科的理论知识，特别需要理论推导时，让我一脸茫然、无从下手，为此，老师亲自在黑板推导，面对理论知识薄弱的我，老师细心指导，不曾放弃，从论文的发表到毕业论文的顺利完成，都离不开高老师的耐心指导和帮助。高老师时常教导我们要端正学习态度、多读文献、不能懒惰，老师严谨的工作态度、广博的学识时刻影响着我，是我永远值得学习的榜样。在这里，衷心的感谢高老师的用心引导和教育，向我的导师说声谢谢，老师，您辛苦了！祝愿老师永远身体健康，事业顺利！

其次，感谢我最爱的家人们。大善无言，至爱无声，是他们为我撑起了一片蓝天，在我迷失方向时给我指引方向，在我经受挫折时给予给我力量，是我坚强的后盾。感谢家人给予给我无私的爱，是他们无私的爱包容了我所有的好与不好、包容了我不成熟的小脾气、包容了我的年少无知，是他们让我伸展开双臂去拥抱我想要的生活，让我在求学期间做到心无旁骛、专心学习，是他们默默无闻支持我许多年以来的求学之路。未来的时光，我将用我最大的努力去回报、去感谢，去让一直为我付出的父母少一点担忧，多一点快乐。

同时，感谢统计学院的各位老师。饮水思其源，学成念吾师，谆谆教诲如春风、似润雨，永铭我心。他们驾驶着智慧的船，载着我们航行在知识的海洋，并邀请相关领域的专家们，向同学们普及学术知识和更深层次的统计学，丰富了我们的见识，谨此向他们表示最诚挚的敬意和感谢！

最后感谢我的师妹们，感谢你们的帮助，感谢你们在学习和生活中给予的关心和陪伴，也感谢几位博士师兄学习与生活的关爱，在此向他们表示感谢！也感谢不曾放弃的自己，衷心感谢评审我论文的专家学者们，感谢你们利用宝贵的时间审阅并提出意见，谢谢！

道阻且长，行则将至，惆怅忧郁的日子，都是成长的足迹，心若向阳，何惧道阻且长。学无止境，勤则可达，路漫漫其修远兮，吾将上下而求索！三年的研究生，满是感恩和庆幸；展望未来，心中有无限憧憬。我始终相信，冬去春来，一树花开，当下与未来，同样精彩!

愿将来我们都能成为我们想要成为的人。

附录

主函数：

```
addpath(genpath('.\'))
clear
clc
rng('default')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% load data
load('seeds.mat')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C=3; % number of class
n=210; % number of data samples

%%%    load the affinity matrix, all the affinity matrix construction methods
%%%    can be used here
load seeds_affinity_matrix
%% Progressive Ensemble
%addpath('\\144.214.36.164\jyh\ProEnSymNMF')
nClass=C;
iter=20;
num_en=20;
mode=2; % 2 stands for the hard manner, while 1 stands for the soft manner

%%% init the input matrix
rng(102)
V=rand(n,C,num_en,iter);


[acc_PEsymNMF,nmi_PEsymNMF,aveNMI,PairwiseNMI,SS,HH,res_PEsymNMF]=...
    progressive_EnSymNMF_v3_outputS_fixdV(W,nClass,iter,num_en,mode,gnd,V)
%%
figure
for i=1:20
    aa(i)=mean(cell2mat(acc_PEsymNMF{i}));
end
% plot(aa)
yyaxis left
plot(aa,'k--');
%set(gca,'ytick',[0.6:0.2:1])
ylabel('Acc')
%ylim([0.6 1])
for i=1:20
    bb(i)=mean(PairwiseNMI{i});
```

```
end
yyaxis right
plot(bb,'k-');
ylabel('ANMI')
xlim([1 20]);
title('SEEDS')
ACC:
function [ACCssnmf,NMIssnmf,Cres]=cal_ACC_NMF_symNMF_v3(H,gnd)
[~,res]=max(H');
labelnew = res;
                %                       gndnew=gnd;
NMIssnmf= MutualInfo(gnd,labelnew);

labelnew = bestMap(gnd,labelnew);
ACCssnmf= length(find(gnd == labelnew))/length(gnd);

Pur=purity(max(gnd),labelnew,gnd);
ARI=RandIndex(labelnew,gnd);

[f,p,r] = compute_f(gnd,labelnew);
Cres.ACC=ACCssnmf;
Cres.NMI=NMIssnmf;
Cres.Pur=Pur;
Cres.ARI=ARI;
Cres.F1=f;
Cres.Pre=p;
Cres.Rec=r;
%% add the metrics for F1 score, Precision and Recall
NMI
function [aveNMI,PairwiseNMI]=cal_aveNMI_symNMF(H)
for i=1:length(H)
    [~,res{i}]=max(H{i}');
end
% PairwiseNMI=zeros(length(H))*(zeros(length(H))-1)/2;
aveNMI=0;
PairwiseNMI=0;

n=length(H);

for i=1:length(H)
    for j=1:length(H)
        try
            temp= MutualInfo(res{i},res{j});
        catch
```

```
                temp=mean(PairwiseNMI);
            end
            PairwiseNMI((i-1)*n+j)=temp;
            aveNMI=temp+aveNMI;
        end
end
```

F得分:

```
function [f,p,r] = compute_f(T,H)

    if length(T) ~= length(H),
        size(T)
        size(H)
    end;

    N = length(T);
    numT = 0;
    numH = 0;
    numI = 0;
    for n=1:N,
        Tn = (T(n+1:end))==T(n);
        Hn = (H(n+1:end))==H(n);
        numT = numT + sum(Tn);
        numH = numH + sum(Hn);
        numI = numI + sum(Tn .* Hn);
    end;
    p = 1;
    r = 1;
    f = 1;
    if numH > 0,
        p = numI / numH;
    end;
    if numT > 0,
        r = numI / numT;
    end;
    if (p+r) == 0,
        f = 0;
    else
        f = 2 * p * r / (p + r);
    end;
```

PUR:

```
function val = purity(CCC, x, y)        %x is obtained, y is true
%% Computing Purity between x and y labels

ind_set=cell(CCC,CCC);
```

```
for ii=1:CCC
    for jj=1:CCC
        ind_set{ii,jj}= intersect(find(x==ii), find(y==jj));
    end
end

val0=0;
for ii=1:CCC
    val_set=[];
    for jj=1:CCC
        val_set=[val_set,length(ind_set{ii,jj})];
    end
    val0=val0+max(val_set);
end

val=val0/length(x);
```