Journal of Computer Applications
计算机应用, 2020, 40(12): 3413 - 3422
ISSN 1001-9081
CODEN JYIIDU
2020-12-10
http://www.joca.cn

文章编号:1001-9081(2020)12-3413-10
DOI:10.11772/j.issn.1001-9081.2020061040

无需特征分解的快速谱聚类算法

刘静姝[1],王　莉[1*],刘惊雷[2]
(1. 太原理工大学 大数据学院,山西 晋中 030600; 2. 烟台大学 计算机与控制工程学院,山东 烟台 264005)
(*通信作者电子邮箱 wangli@tyut.edu.cn)

摘　要:为了解决样本数较大时,传统谱聚类算法执行特征分解消耗时间过大的问题,提出了一种无需特征分解的快速谱聚类算法,通过乘法更新迭代来降低时间开销。首先,利用Nyström方法进行随机采样,建立了采样矩阵和原始矩阵之间的关系;其次,基于乘法更新原理实现矩阵指示器矩阵的迭代更新;最后,在理论上对所设计算法进行了正确性和收敛性分析。在广泛使用的五个真实数据集和三个人工合成数据集上进行测试。实验结果表明,在真实数据集上,所提算法的标准互信息(NMI)平均值为0.45,与*k*-means聚类算法相比提高了12.50%;运行时间为61.73 s,与传统谱聚类算法相比减少了61.13%;而且表现性能优于层次聚类算法,验证了该算法的有效性。

关键词:谱聚类;Nyström采样;收敛性分析;特征分解;乘法更新迭代
中图分类号:TP181　　**文献标志码**:A

Fast spectral clustering algorithm without eigen-decomposition

LIU Jingshu[1], WANG Li[1*], LIU Jinglei[2]
(1. *College of Data Science, Taiyuan University of Technology, Jinzhong Shanxi* 030600, *China*;
2. *School of Computer and Control Engineering, Yantai University, Yantai Shandong* 264005, *China*)

Abstract: The traditional spectral clustering algorithm needs too much time to perform eigen-decomposition when the number of samples is very large. In order to solve the problem, a fast spectral clustering algorithm without eigen-decomposition was proposed to reduce the time overhead by multiplication update iteration. Firstly, the Nyström algorithm was used for random sampling in order to establish the relationship between the sampling matrix and the original matrix. Then, the indicator matrix was updated iteratively based on the principle of multiplication update iteration. Finally, the correctness and convergence analysis of the designed algorithm were given theoretically. The proposed algorithm was tested on five widely used real datasets and three synthetic datasets. Experimental results on real datasets show that: the average Normalized Mutual Information (NMI) of the proposed algorithm is 0.45, which is improved by 12.5% compared with that of the *k*-means clustering algorithm; the computing time of the proposed algorithm achieves 61.73 s, which is decreased by 61.13% compared with that of the traditional spectral clustering algorithm; and the performance of the proposed algorithm is superior to that of the hierarchical clustering algorithm, which verify the effectiveness of the proposed algorithm.

Key words: spectral clustering; Nyström sampling; convergence analysis; eigen-decomposition; multiplication update iteration

0　引言

随着信息技术的发展,人们在日常生活中从互联网上获取的信息以海量规模存在,并且持续高速增长[1]。聚类分析利用数据划分来找到数据间的内在联系,能够更快速、更高效、更低成本地收集存储数据[2],已经被应用到机器学习的各个领域中,例如图像分割[3-4]、特征选择[5-6]和降维[7-8]。在过去的几年,研究领域中涌现出许多应用成功的聚类方法,包括层次聚类方法[9]、中心聚类[10]。其中,使用普遍的聚类算法包括*k*-means算法[11]、模糊C均值(Fuzzy C-Means, FCM)算法[12]和最大期望(Expectation Maximization, EM)算法[13]等。诸如此类经典算法,虽然步骤简单且执行效率高,但当聚类样本集的空间为非凸结构时,算法会易陷入局部最优划分中,因此它们缺乏处理复杂簇结构的能力。

谱聚类算法的思想基于谱图理论,将数据聚类问题转化成图划分问题,通过表示出数据的低维非线性形式来实现降维,并且在降维的同时也将这些对象嵌入到欧氏空间,从而在新的空间中进行聚类[14],这种假设对数据的结构分布要求不强,使得它能够处理数据集非凸时的聚类任务[15],克服了*k*-means算法等传统聚类方法基于中心聚类而产生的缺点。除此之外,因为对误差数据和噪声数据不敏感,谱聚类方法具有较好的鲁棒性[16]。

尽管谱聚类在很多领域都取得了不错的发展和应用成果,但仍处于发展阶段,还有很多缺陷需要深入研究并进一步

收稿日期:2020-06-12;**修回日期**:2020-09-21;**录用日期**:2020-09-27。
基金项目:国家自然科学基金资助项目(61872260);山西省自然科学基金资助项目(201703D421013)。
作者简介:刘静姝(1997—),女,山东烟台人,硕士研究生,主要研究方向:大数据、矩阵分解; 王莉(1971—),女,山西太原人,教授,博士,CCF会员,主要研究方向:在线社会网络计算、移动通信; 刘惊雷(1970—),男,山东烟台人,教授,博士,CCF会员,主要研究方向:图模型推理、矩阵分解。

改进。首先，谱聚类需要计算图拉普拉斯矩阵的特征向量，所需要的时间复杂度为$O(n^3)$，导致当面向大规模数据集时，谱聚类会出现明显的速度缺陷。其次，传统的谱聚类算法存储相似度矩阵需要$O(n^2)$大小的内存空间，如此高的复杂度在处理大规模数据时是无法被接受的，导致它只适用于规模较小的数据集。

为提升谱聚类算法的扩展性，研究者设计出许多可以降低特征分解复杂度的算法来克服计算负担。Fowlkes 等[17]通过改进 Nyström 方法实现了谱聚类的快速近似特征分解，丁世飞等[18]利用自适应采样技术扩展了 Nyström 方法，扩展了谱聚类在大规模数据集中的聚类效果。此外，Cai 等[19]提出基于地标点的谱聚类（Landmark-based Spectral Clustering，LSC）算法，通过选择地标点并计算数据点与地标点之间的相似度矩阵的乘积得出近似矩阵，虽然利用这种矩阵近似性质可以实现快速特征分解，但该方法的抽样随机性会导致处理大数据集时出现样本点过于集中的问题。

本文利用 Nyström 采样思想，设计了一种无需特征值分解的快速谱聚类迭代优化（Fast Spectral Clustering using Iterative optimization，IFSC）算法。该算法克服了传统谱聚类方法处理大规模数据集时的缺陷，通过 Nyström 思想采样一小部分样本，利用小样本矩阵来构造整个原始相似矩阵，通过乘法更新迭代优化规则来实现聚类。

本文的主要工作如下：

1）基于谱聚类问题的拉格朗日函数，对聚类指示矩阵 $\boldsymbol{Y}$ 进行求导，得到关于矩阵 $\boldsymbol{Y}$ 迭代更新的乘法规则，从而避免传统谱聚类的特征值分解。

2）设计了一种不需要特征值分解的谱聚类算法 IFSC，该算法根据关于矩阵 $\boldsymbol{Y}$ 的乘法迭代规则进行更新。具体来说，基于采样的数据点间的相似矩阵 $\boldsymbol{E} \in \mathbf{R}^{m \times m}$、采样点和剩余点之间的相似矩阵 $\boldsymbol{F} \in \mathbf{R}^{m \times (n-m)}$，以及构造的采样小矩阵和原始大矩阵之间的关系对小矩阵进行迭代更新，从而实现对大矩阵的更新，实现了快速谱聚类。

3）在理论上，根据聚类指示矩阵 $\boldsymbol{Y}$ 构造辅助函数证明了算法的收敛性，使用 KKT（Karush-Kuhn-Tucker）条件证明了本文所设计乘法更新规则的正确性。

4）在五个真实数据集上进行了实验验证，实验结果表明，本文设计的快速谱聚类算法与传统谱聚类算法和其他聚类算法相比，处理大规模数据集时，计算时间有所降低；且在处理含有较多噪声的真实数据集时表现优于其他聚类方法（如层次聚类）。

1 相关工作

1.1 谱聚类

谱聚类方法利用图论思想，将数据聚类问题转化成图划分问题，通过对图拉普拉斯矩阵进行特征值分解，得到原始数据在转换后的低维空间中的向量表示。最后，对这些低维特征向量运行 k-means 算法，从而得到最终聚类结果。

其中，谱聚类的时间复杂度缺陷主要体现在三个方面：相似度矩阵的构建$O(n^2d)$、拉普拉斯矩阵的特征值分解$O(n^3)$和最终的 k-means 聚类步骤$O(nkt)$（t 为迭代次数）。它的空间复杂度缺陷主要体现在存储相似度矩阵与拉普拉斯矩阵需要的$O(n^2)$上。随着数据大小 n 的增加，谱聚类的计算复杂度过高，这使得谱聚类方法在处理大规模数据集时，无法发挥更好的性能。

针对以上几点限制谱聚类应用的主要缺陷，可将现有的谱聚类方法主要划分为两种类型：

一种类型是通过约简相似度矩阵的大小来降低样本数量并减小数据集规模，例如 Martin 等[20]利用这种思路设计了 KASP（K-means Approximate SPectral clustering）算法，优先对数据集使用 k-means 等方法进行初始化聚类，从而快速地将大部分点绑定到局部的中心点上去，再针对这些中心点进行谱聚类，此时中心点聚类结果即视作绑定于中心点上的普通点的聚类结果。此外，叶茂等[21]改进了基于地标选择的谱聚类（LSC）算法，使用基于近似奇异值分解（Singular Value Decomposition，SVD）抽样方法实现快速的标点采样，克服了抽样地标点效果不稳定的缺点。与文献[14]研究思路不同的是，普通点和中心点的归一化关系是由点与点之间的最短路径来计算，由于保留了图的特性，它更能反映出图的连通状态和点与点之间的相互关系。然而即使在利用这种空间换时间的思路实现改进后，数据稠密图通过阈值限定转化成稀疏图后仍需要$O(rn \log n)$的时间复杂度，不理想的算法速度限制了它在较大数据集上的聚类应用。

另一种类型是通过选择代表对象来降低样本数据集的规模。这种方法通过在数据集中选择有代表性的对象，利用代表对象构成一个小规模数据集，从而降低可用数据集的规模。然后，利用已有的谱聚类方法划分这些代表对象构成的小规模数据集。最后，根据代表对象所属的类来分配原始数据集中数据所属的类。Chen 等[22]提出了基于子矩阵构造的研究方法，通过利用 Nyström 方法从原始数据集中随机选择 p 个代表，并建立大小为 $N \times p$ 的相似度子矩阵。张涛等[23]改进了子空间聚类算法对高斯噪声敏感的缺陷，使用优化的核范数对系数矩阵的奇异值进行正则化，能够在提高算法准确率的同时，保持其高斯噪声下的稳定性。

由此可见，特征提取是谱聚类方法完成相似度矩阵优化的主要着手点。利用数据点距离的参照采样虽然操作直观简便，但阈值的人为选择和特征空间的映射会导致代表样本和实际样本的差异。此外，使用绑定代表点的方式将图稀疏化会损害数据点集的密度，降低了聚类准确率。如果此时根据数据集的大小增加代表点的密度，依然会产生数据量越大、代表点越多，从而导致算法准确率降低的问题。

本文使用了 Nyström 方法进行随机采样间接求解相似度矩阵，使得所有的行和列都能通过映射参与到计算中，最大限度保持聚类结果的准确度。该算法利用选取的子矩阵与原始矩阵的关系来代入乘法更新迭代规则完成更新，从而实现聚类，进一步降低传统方法特征值分解步骤中所需的时间开销。

1.2 谱聚类的基本算法

谱聚类利用谱图理论思想将数据点视作无向图，这里无向图边的权重代表数据点的成对相似性，聚类的目标就是将这些数据点分配到不同的类簇中，使得簇内的数据点之间有较大的相似度，而簇之间的数据点之间的相似度较小。因此，需要构建一个相似度图，即以数据点为顶点、以相似度为权重的一个带权图，从而使构建的相似度图能够反映原始数据集中各个点之间的相似关系。其中，本文公式推导所用到的一些典型数学符号如表 1 所示。

构建相似度矩阵 $\boldsymbol{W}$ 常常使用如下的高斯核函数来作定义：

$$W_{ij}=\exp\left(\frac{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2}{2\sigma^2}\right) \tag{1}$$

其中,W_{ij}表示数据向量样本$\boldsymbol{x}_i$和$\boldsymbol{x}_j$之间的相似性,使用高斯核函数的时候,需要确定参数σ。那么在相似度图中,数据点聚类问题就转变成图划分问题。划分准则是使划分之后的子图内部的点间相似度尽可能大,不同子图之间的相似度尽可能小。下面介绍利用规范割(Normalized Cut, NCut)划分:将顶点集V划分为两部分A和B,即:$A\cup B=V, A\cap B=\varnothing$,构建出相似度矩阵后,将$A$和$B$之间的权重之和记作:$cut(A,B)=\sum_{i\in A,j\in B}W_{ij}$。此时将第$i$个节点的度定义为:$d_i=\sum_j W_{ij}$。

表 1　符号描述

Tab. 1　Symbol description

符号	描述
$\boldsymbol{X}\in\mathbf{R}^{d\times n}$	原始数据矩阵
$\boldsymbol{x}_i\in\mathbf{R}^d, i=1,2,\cdots,n$	数据集中的样本点
$\boldsymbol{W}\in\mathbf{R}^{n\times n}$	基于高斯核产生的相似度矩阵
$\boldsymbol{L}\in\mathbf{R}^{n\times n}$	图的拉普拉斯矩阵
$\boldsymbol{I}$	单位矩阵
$\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n$	$\boldsymbol{x}_i$对应的指示向量
$\boldsymbol{Y}=[\boldsymbol{y}_1^{\mathrm{T}},\boldsymbol{y}_2^{\mathrm{T}},\cdots,\boldsymbol{y}_n^{\mathrm{T}}]\in\mathbf{R}^{n\times k}$	指示向量的集合即指示矩阵
$\boldsymbol{P},\boldsymbol{Q}\in\mathbf{R}^{n\times n}$	构造指示矩阵的辅助矩阵
$\boldsymbol{E}\in\mathbf{R}^{m\times m}$	采样点间的小样本矩阵
$\boldsymbol{F}\in\mathbf{R}^{m\times(n-m)}$	采样点和其余剩余样本的相似矩阵
d	数据的维度
n	样本点个数
m	选中的样本点个数
k	聚类个数

集合的容量(volume)为该集合内所有节点度的总和:$vol(A)=\sum_{i\in A}d_i, vol(B)=\sum_{i\in B}d_i$。那么两个集合之间的规范割如下:

$$NCut(A,B)=\frac{cut(A,B)}{vol(A)}+\frac{cut(A,B)}{vol(B)} \tag{2}$$

谱聚类旨在找到使得规范割目标函数最小的子集A和子集B[24],用类的容量作归一项,兼顾了类的内部和外部的连接。利用以下公式可以计算出最优的$NCut$值:

$$\boldsymbol{y}=\arg\min NCut(A,B)=\arg\min_{\boldsymbol{y}} NCut(\boldsymbol{y})=\arg\min_{\boldsymbol{y}}\frac{\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{y}} \tag{3}$$

Malik等[25]给出归一化的拉普拉斯矩阵的定义:

$$\boldsymbol{L}=\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{D}^{-\frac{1}{2}}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$$

此时$\boldsymbol{L}$是半正定矩阵,它的特征值区间为[0,2],所以$\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$的特征值也被限制在区间[1,1]中。扩展到$k>2$的多分类问题中,公式可被重写为:

$$NCut=\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y}) \tag{4}$$

此时有$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}$,矩阵$\boldsymbol{Y}$中包含归一化拉普拉斯矩阵$\boldsymbol{L}$的前$k$个特征向量。因此,优化公式即可通过标准迹线最小化问题解决,使用文献[25]中提出的归一化谱聚类求解得出指示矩阵$\boldsymbol{Y}$。在得到矩阵$\boldsymbol{Y}$作为聚类中心后,利用k-means算法对指示矩阵$\boldsymbol{Y}$的行进行聚类,这种算法称为归一化的谱聚类算法。

为了方便在实验中与本文设计的算法进行对比,在算法1中简单描述了谱聚类算法的过程。

算法 1　谱聚类(Spectral Clustering, SC)算法。

输入　数据样本集$\boldsymbol{X}$,目标聚类个数k;

输出　k个聚好的类。

步骤 1　根据相似度函数式(1)生成相似度矩阵$\boldsymbol{W}\in\mathbf{R}^{n\times n}$;

步骤 2　计算图拉普拉斯矩阵$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$;

步骤 3　计算矩阵$\boldsymbol{L}$的前k个特征向量$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_k$;

步骤 4　将向量$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_k$作为列向量组成矩阵$\boldsymbol{U}$;

步骤 5　将$\boldsymbol{U}$的第i行作为数据点$\boldsymbol{y}_i\in\mathbf{R}^k, i=1,2,\cdots,n$;

步骤 6　使用k-means将点$\boldsymbol{y}_i$聚类到k个类$C_1,C_2,\cdots,C_k$中;

步骤 7　通过$A_i=j\,|\,y_j\in C_i$恢复类$A_1,A_2,\cdots,A_k$,得到聚类结果。

2　快速谱聚类框架

2.1　问题描述

谱聚类算法可以视作函数最小化:

$$\min\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\lambda\|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}\|_{\mathrm{F}}^2\sqrt{a^2+b^2} \tag{5}$$

其中:λ是拉格朗日常数;$\|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}\|_{\mathrm{F}}^2$是正交约束项。而式(5)的目标函数是非光滑的,因此不容易由求解拉普拉斯矩阵$\boldsymbol{L}$的特征值分解来获得有效的分辨率。非负矩阵分解(Non-negative Matrix Factorization, NMF)算法能够通过松弛技术处理聚类问题,借鉴这种思想,可以放宽离散条件,并提出乘法更新优化的思路来解决特征值分解问题。将非负约束的指标矩阵记作$\boldsymbol{Y}$,其中$Y_{ij}>0$。此外,一些传统的谱聚类相关方法将指标矩阵$\boldsymbol{Y}$放松为正交约束,即$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}$。文献[26]指出,如果指示矩阵$\boldsymbol{Y}$同时满足正交和非负,则在矩阵$\boldsymbol{Y}$的每一行中只有一个元素为正,其他元素为零。因此可以通过添加约束$\boldsymbol{Y}>0$和$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}$来获得文中定义的理想指示矩阵$\boldsymbol{Y}$,进而利用这种简单有效的方式解决特征值分解问题。考虑以上两个约束的同时放宽离散条件,则式(5)可写为:

$$\begin{aligned}E(\boldsymbol{Y})&=\min\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\lambda\|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}\|_{\mathrm{F}}^2=\\&\min\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\lambda\mathrm{Tr}((\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}))\end{aligned} \tag{6}$$

其中$\boldsymbol{Y}>0$。为了实现损失函数式(6)的最小化,有:

$$\frac{1}{2}\frac{\partial E(\boldsymbol{Y})}{\partial\boldsymbol{Y}}=\boldsymbol{L}\boldsymbol{Y}+2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y} \tag{7}$$

其中$\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$,则式(6)为:

$$\begin{aligned}&(\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})\boldsymbol{Y}+2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y}=\\&\boldsymbol{Y}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{Y}+2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y}=\\&(\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})-(2\lambda\boldsymbol{Y}+\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{Y})\end{aligned} \tag{8}$$

经过上面推导能观察出,式(6)可以视作两部分,即$\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$和$2\lambda\boldsymbol{Y}+\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$。因为$\boldsymbol{Y}>0$,$\boldsymbol{D}>0$且$\boldsymbol{W}\geqslant 0$,这两部分都是非负的。为便于描述,将前一个因子表示为$\boldsymbol{Q}=\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$,后者表示为$\boldsymbol{P}=2\lambda\boldsymbol{Y}+\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$。根据文献[27]中提出的标准NMF算法的乘法更新规则,可通过更新$\boldsymbol{Y}$的方式来使得式(6)中的损失函数最小化,即:

$$\boldsymbol{Y}=\boldsymbol{Y}\circ\boldsymbol{P}\oslash\boldsymbol{Q} \tag{9}$$

其中,"$\circ$"和"$\oslash$"分别表示Hadamard乘法和Hadamard除法(即逐元素乘法和除法),且有:

$$Y_{ij}\leftarrow Y_{ij}\cdot\sqrt{\frac{P_{ij}}{Q_{ij}}} \tag{10}$$

然后经过一系列迭代后使损失函数收敛。由于在指示矩阵 $\boldsymbol{Y}$ 的每一行中仅有一个元素为正，其他元素接近零，因此在实现聚类时，可以视作完美的约束指示矩阵，而不像传统的谱聚类中还需要对指示矩阵 $\boldsymbol{Y}$ 进行松弛和特征分解的处理。

2.2 算法设计和描述

2.2.1 设计模型

为了将谱聚类算法应用于大规模数据集，需要进一步降低计算相似度矩阵的时间和存储复杂度，因此本文在优化模型中选用 Nyström 方法来扩展近似有限样本的原始相似度矩阵。

根据 Nyström 采样原理，将 n 个数据点视作两部分：m 个随机采样得到的样本点和剩余的 $n-m$ 数据点。则相似度矩阵 $\boldsymbol{W}$ 可以写成：

$$\boldsymbol{W}=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F}^{\mathrm{T}} & \boldsymbol{C}\end{bmatrix} \tag{11}$$

其中：矩阵 $\boldsymbol{E}\in\mathbf{R}^{m\times m}$ 表示 m 个采样数据点之间的相似度矩阵，它可以用特征分解形式 $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$ 来表示，特征向量 $\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{I}$。矩阵 $\boldsymbol{F}\in\mathbf{R}^{m\times(n-m)}$ 表示采样点和剩余点之间的相似度矩阵，而矩阵 $\boldsymbol{C}\in\mathbf{R}^{(n-m)\times(n-m)}$ 代表剩余点之间的相似度矩阵。令 $\overline{\boldsymbol{U}}$ 表示 $\boldsymbol{W}$ 的近似特征向量，由 Nyström 扩展可得到：

$$\overline{\boldsymbol{U}}=\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{F}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\end{bmatrix} \tag{12}$$

相应的，$\boldsymbol{W}$ 的近似矩阵 $\hat{\boldsymbol{W}}$ 为：

$$\hat{\boldsymbol{W}}=\overline{\boldsymbol{U}}\boldsymbol{\Lambda}\overline{\boldsymbol{U}}^{\mathrm{T}}=\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{F}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\end{bmatrix}\boldsymbol{\Lambda}\begin{bmatrix}\boldsymbol{U}^{\mathrm{T}} & \boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{F}\end{bmatrix}=\begin{bmatrix}\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{\mathrm{T}} & \boldsymbol{F}\\ \boldsymbol{F}^{\mathrm{T}} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}\end{bmatrix}=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F}^{\mathrm{T}} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}\end{bmatrix} \tag{13}$$

由此可见，Nyström 扩展技术使得矩阵 $\boldsymbol{C}$ 可以用 $\boldsymbol{C}=\boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}$ 来近似逼近，则 $\boldsymbol{W}$ 可写为：

$$\hat{\boldsymbol{W}}=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F}^{\mathrm{T}} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}\end{bmatrix}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-1}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix} \tag{14}$$

由于 $m\ll n$，抽样后的剩余点个数很多，而 Nyström 技术利用近似逼近降低了计算剩余点间相似度这一步骤所需的时间和空间复杂度。定理 1 给出了关于矩阵 $\hat{\boldsymbol{W}}$ 正交特征向量表达式的证明。

定理 1 若给定一个矩阵 $\boldsymbol{E}$ 为正定矩阵，且定义矩阵 $\boldsymbol{Q}=\boldsymbol{E}+\boldsymbol{E}^{-1/2}\boldsymbol{F}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1/2}$，将其对角化 $\boldsymbol{Q}=\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}\boldsymbol{U}_{\boldsymbol{Q}}^{\mathrm{T}}$，则 $\hat{\boldsymbol{W}}$ 的正交特征向量为：

$$\boldsymbol{V}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-\frac{1}{2}}\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-\frac{1}{2}} \tag{15}$$

证明

1）首先证明 $\boldsymbol{V}$ 是 $\boldsymbol{W}$ 的特征向量：

$$\hat{\boldsymbol{W}}=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F}^{\mathrm{T}} & \boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}\end{bmatrix}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-1}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix}=\left\{\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-\frac{1}{2}}\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-\frac{1}{2}}\right\}\boldsymbol{\Lambda}_{\boldsymbol{Q}}\left\{\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-\frac{1}{2}}\boldsymbol{U}_{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{E}^{-\frac{1}{2}}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix}\right\}=\boldsymbol{V}\boldsymbol{\Lambda}_{\boldsymbol{Q}}\boldsymbol{V}^{\mathrm{T}}$$

2）然后证明 $\boldsymbol{V}$ 和 $\boldsymbol{V}^{\mathrm{T}}$ 是正交的。

$$\boldsymbol{I}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}=\left\{\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-\frac{1}{2}}\boldsymbol{U}_{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{E}^{-\frac{1}{2}}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix}\right\}\left\{\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-\frac{1}{2}}\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-\frac{1}{2}}\right\}$$

将式中括号里的两项分别乘以 $\boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-1/2}$ 和 $\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-1/2}\boldsymbol{U}_{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{Q}}^{-1/2}$，即可得到 $\boldsymbol{Q}$。

将 Nyström 扩展矩阵用于谱聚类，相似矩阵需要作归一化处理，即 $\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}$，其中 $\boldsymbol{D}$ 是对角矩阵，它的对角线元素 D^{ii} 等于矩阵 $\hat{\boldsymbol{W}}$ 的第 i 行元素和。

文献[17]中给出了节点度的计算方法：

$$\boldsymbol{d}=\hat{\boldsymbol{W}}\boldsymbol{l}=\begin{bmatrix}\boldsymbol{E}\boldsymbol{l}_m+\boldsymbol{F}\boldsymbol{l}_n\\ \boldsymbol{F}^{\mathrm{T}}\boldsymbol{l}_m+\boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}\boldsymbol{l}_n\end{bmatrix}=\begin{bmatrix}\boldsymbol{e}_r+\boldsymbol{f}_r\\ \boldsymbol{f}_c+\boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{f}_r\end{bmatrix} \tag{16}$$

其中：用 $\boldsymbol{e}_r=\boldsymbol{E}\boldsymbol{l}_m$ 来代表矩阵 $\boldsymbol{E}$ 的行和；$\boldsymbol{F}\boldsymbol{l}_n$ 代表矩阵 $\boldsymbol{F}$ 的行和；$\boldsymbol{f}_c$ 表示矩阵 $\boldsymbol{F}$ 的列和；$\boldsymbol{l}$ 表示元素均为 1 的列向量。则不需要求解 $\boldsymbol{C}=\boldsymbol{F}^{\mathrm{T}}\boldsymbol{E}^{-1}\boldsymbol{F}$，仅利用 $\boldsymbol{d}$ 就可以将矩阵 $\boldsymbol{E}$ 和 $\boldsymbol{F}$ 归一化：

$$\begin{cases}E_{ij}\leftarrow\dfrac{E_{ij}}{\sqrt{d_i d_j}}\\ F_{ij}\leftarrow\dfrac{F_{ij}}{\sqrt{d_i d_{j+m}}}\end{cases} \tag{17}$$

将式(17)中的 $\boldsymbol{E}$ 和 $\boldsymbol{F}$ 代入标准化的相似度矩阵 $\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}$ 中得到：

$$\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-1}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix} \tag{18}$$

由于矩阵 $\boldsymbol{E}^{-1}$ 中含有负数元素，故标准化结果 $\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}$ 也可能是负的。为了保证乘法迭代规则的约束性，需要满足标准化结果非负，即 $\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}\geqslant 0$，将矩阵 $\boldsymbol{E}$ 中的元素拆分为正负两部分并分别记作：$\boldsymbol{E}^{+}=(|\boldsymbol{E}|+\boldsymbol{E})./2$ 和 $\boldsymbol{E}^{-}=(|\boldsymbol{E}|-\boldsymbol{E})./2$（其中./代表矩阵中的元素逐个相除），那么此时的 $\boldsymbol{E}^{+}$ 和 $\boldsymbol{E}^{-}$ 都是非负矩阵。

将 $\boldsymbol{E}$ 和 $\boldsymbol{F}$ 放在指示矩阵 $\boldsymbol{Y}$ 中，则更新规则中的 $\boldsymbol{P}$、$\boldsymbol{Q}$ 可以写作：

$$\begin{cases}\boldsymbol{P}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{+}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix}\boldsymbol{Y}+2\lambda\boldsymbol{Y}\\ \boldsymbol{Q}=\begin{bmatrix}\boldsymbol{E}\\ \boldsymbol{F}^{\mathrm{T}}\end{bmatrix}\boldsymbol{E}^{-}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\end{bmatrix}+\boldsymbol{Y}+2\lambda\end{cases} \tag{19}$$

此时，矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 都是非负的，即可按照式(9)来进行乘法更新迭代。

2.2.2 算法描述

第 2.2.1 节中介绍了基于乘法更新迭代的快速谱聚类算法优化模型。本文提出了基于乘法更新迭代思想的快速谱聚类算法，框架具体实现过程见算法 2。

算法 2 基于乘法更新迭代的快速谱聚类算法 IFSC。

输入 数据集 $\boldsymbol{X}$，聚类个数 k，参数 λ；

输出 指示矩阵 $\boldsymbol{Y}$，聚类结果。

初始化 初始化参数 λ 并随机生成指示矩阵 $\boldsymbol{Y}\in\mathbf{R}^{n\times k}$。

步骤 1 利用式(1)计算出数据集样本间的相似度矩阵 $\boldsymbol{W}$，从数据集中随机选择 m 个样本，通过式(11)构建数据点间的相似度矩阵 $\boldsymbol{E}\in\mathbf{R}^{m\times m}$；采样点和剩余点间的相似度矩阵 $\boldsymbol{F}\in\mathbf{R}^{m\times(n-m)}$，并根据式(16)计算 $\boldsymbol{d}$。

步骤 2 利用式(17)更新矩阵 $\boldsymbol{E}$ 和矩阵 $\boldsymbol{F}$，并根据式(18)计算出近似值 $\hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-1/2}$ 用于乘法更新；

当聚类指示矩阵 $\boldsymbol{Y}$ 不收敛；

则计算分子函数 $\boldsymbol{P}=\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}+2\lambda\boldsymbol{Y}$；

计算分母函数 $\boldsymbol{Q}=\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$；

根据式(10)进行乘法更新迭代：$Y_{ij}\leftarrow Y_{ij}\cdot\sqrt{\dfrac{P_{ij}}{Q_{ij}}}$。

步骤 3 输出指示矩阵 $\boldsymbol{Y}$，并将 $\boldsymbol{Y}$ 输入 k-means 聚类算法得到聚类结果。

传统谱聚类算法可以视作三个阶段：第一步为预处理阶

段,对由数据点计算出的相似度矩阵进行标准化;第二步为谱映射阶段,计算相似度矩阵的特征向量;第三部为分组处理阶段,使用常见的分组算法来得到聚类结果。本文的算法利用乘法更新迭代的Nyström扩展思想来实现快速谱聚类,从而降低了前两个步骤所需要的时间损耗。

首先,在输入包含n个点的数据集$\boldsymbol{X}=\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$中,依据已知的相似性度量方法(式(1)),构造数据点集的相似性矩阵$\boldsymbol{W}$,根据式(18),利用Nyström方法选取出一部分样本来近似整个相似度矩阵。

由于需要在利用乘法更新迭代思想实现对目标函数的优化之后得到指示矩阵,再完成谱聚类最后的处理或分组步骤,但是从式(16)中可以看出,$\boldsymbol{E}^{-1}$中可能含有负数元素,使得$\hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-1/2}$可能为负。而设计条件的迭代规则中要求满足非负约束性$\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}\geqslant 0$,因此在处理数据时,将矩阵$\boldsymbol{E}^{-1}$中的元素拆分为正负两部分并分别记作:$\boldsymbol{E}^{+}=(|\boldsymbol{E}|+\boldsymbol{E})./2$和$\boldsymbol{E}^{-}=(|\boldsymbol{E}|-\boldsymbol{E})./2$,那么此时二者都是非负矩阵。其次,由于$\boldsymbol{D}^{-1/2}\hat{\boldsymbol{W}}\boldsymbol{D}^{-1/2}$中的大部分元素是正的,则近似值中的大部分元素也为正,此时可以将近似值$\hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-1/2}$中的负数元素视作噪声,直接记作零元素处理。因此在公式$Y_{ij}\leftarrow Y_{ij}\cdot\sqrt{\frac{P_{ij}}{Q_{ij}}}$中满足$\boldsymbol{P}>0$和$\boldsymbol{Q}>0$的条件,即能按照更新规则进行后续计算。

再依据$\boldsymbol{P}=\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}+2\lambda\boldsymbol{Y}$和$\boldsymbol{Q}=\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$,通过更新矩阵$\boldsymbol{Y}$来使式(5)中的损失函数最小化,直至目标函数收敛。指示矩阵$\boldsymbol{Y}$中元素为Y_{ij},指示矩阵形成了一个低维嵌入空间$\mathbf{R}^{n\times k}$,其第i行与原始数据集的点$\boldsymbol{x}_i$相对应。以上步骤相当于将原始数据点映射到基于一个或多个特征向量确定的低维空间中,得到的指示矩阵$\boldsymbol{Y}$即为数据点在新空间中的表示。

最后对矩阵$\boldsymbol{Y}$的行向量聚类,使用k-means算法将数据点划分,若第i行被分到第j类中,则将数据点$\boldsymbol{x}_i$归到第j类,从而得到k个聚类簇,输出聚类结果。

3　算法理论分析

3.1　正确性和收敛性

在本节中,参照Ding等[28]的思想,通过不同的对象和辅助函数来证明所提算法的正确性和收敛性。

KKT条件是非线性规划最佳解的必要条件。KKT条件将拉格朗日乘数法所处理涉及等式的约束优化问题推广至不等式。为了验证所提算法的正确性,需要引入满足KKT互补条件的拉格朗日函数:通过将它的梯度设置为零,能够得到一个解必定收敛于固定点的不动点方程。如果可以证明式(10)中的更新规则满足这些定点方程以及KKT定点条件,则证明了本文所设计的IFSC算法的正确性。

3.1.1　正确性

命题1　算法IFSC的正确性。

式(5)中给出了目标函数,如式(17)中所示的更新规则,此时得到的约束解满足规则下的KKT互补条件。

证明　为了解决优化问题,需要引入拉格朗日函数,由于矩阵计算规则,有$\mathrm{Tr}(\boldsymbol{AB})=\mathrm{Tr}(\boldsymbol{BA})$且$\mathrm{Tr}(\boldsymbol{A})=\mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}})$。则:

$$\begin{aligned}L(\boldsymbol{Y},\lambda)&=\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\lambda\|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}\|_{\mathrm{F}}^{2}=\\&\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y})+\lambda\mathrm{Tr}((\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-\boldsymbol{I}))=\\&\mathrm{Tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Y}+\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}+\lambda\boldsymbol{I})\end{aligned}\tag{20}$$

其中,拉格朗日乘数$\lambda>0$,$\mathrm{Tr}(\boldsymbol{Y}^{t}\boldsymbol{L}\boldsymbol{Y})$用于谱聚类,$\|\boldsymbol{Y}^{T}\boldsymbol{Y}-\boldsymbol{I}\|_{\mathrm{F}}^{2}$用于实现正交约束,这个函数满足KKT的互补松弛条件。将梯度设置为零,可得:

$$\begin{aligned}\frac{\partial L(\boldsymbol{Y},\lambda)}{\partial\boldsymbol{Y}}&=2\boldsymbol{L}\boldsymbol{Y}+4\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-4\lambda\boldsymbol{Y}=\\&2(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{Y}+4\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-4\lambda\boldsymbol{Y}\end{aligned}\tag{21}$$

故有:

$$\begin{aligned}\frac{1}{2}\frac{\partial L(\boldsymbol{Y},\lambda)}{\partial\boldsymbol{Y}}&=\left(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}\right)\boldsymbol{Y}+2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y}=\\&\boldsymbol{Y}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{Y}+2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}-2\lambda\boldsymbol{Y}=\\&(\boldsymbol{Y}+2\lambda\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})-(2\lambda\boldsymbol{Y}+\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{Y})\end{aligned}\tag{22}$$

由互补松弛条件得出:

$$-\frac{1}{2}(\boldsymbol{L}\boldsymbol{Y})(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}-\boldsymbol{Y})_{i,j}^{-1}Y_{i,j}=\lambda Y_{i,j}=0\tag{23}$$

可以看出对于固定点,该等式的解必收敛,且更新规则式(17)中的极限解满足固定点方程,在极限处有:$\boldsymbol{Y}^{(\infty)}=\boldsymbol{Y}^{(t+1)}=\boldsymbol{Y}^{(t)}$,其中$t$为迭代次数。

$$Y_{ij}\leftarrow Y_{ij}\cdot\sqrt{\frac{P_{ij}}{Q_{ij}}}\tag{24}$$

则式(24)递减到:

$$(\boldsymbol{L}\boldsymbol{Y})(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}-\boldsymbol{Y})_{i,j}^{-1}Y_{i,j}^{2}=0\tag{25}$$

当约束区域包含目标函数的原有可行解时,此时加上约束可行解仍落在约束区域内,对应$g(x)<0$的情况,此时约束条件不起作用,故此时可以让$\lambda=0$,因为约束条件没有作用。当约束区域不包含目标函数的可行解时,此时加上约束后可行解落在边界$g(x)=0$上,所以无论哪种情况都会得到:$\lambda g(x)=0$。因此,式(17)中具有更新规则的约束解,满足式(23)和KKT定点条件。

3.1.2　收敛性

为了证明IFSC算法的收敛性,需要构造辅助函数,从而使得在更新规则下,目标函数单调递减。

命题2　算法IFSC的收敛性。

在更新规则(式(10))下,式(5)中所示的目标函数单调递减。

证明　将目标函数式(5)记作关于矩阵$\boldsymbol{X}$的函数:

$$J(\boldsymbol{X})=\mathrm{Tr}(-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{+}+2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{-}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{+}\boldsymbol{X}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{-}\boldsymbol{X})\tag{26}$$

其中,$\boldsymbol{A}=\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$,$\boldsymbol{B}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$。这时,文献[29]中指出需要通过构造辅助函数来证明等式在更新规则式(10)下单调递减,此时辅助函数$Z(\boldsymbol{X},\boldsymbol{X}')$和$J(\boldsymbol{X})$应同时满足以下两个条件:

$$\begin{cases}Z(\boldsymbol{X},\boldsymbol{X}')\geqslant J(\boldsymbol{X})\\Z(\boldsymbol{X},\boldsymbol{X})=J(\boldsymbol{X})\end{cases}\tag{27}$$

此时,定义:

$$\boldsymbol{X}^{t+1}=\min_{\boldsymbol{X}}Z(\boldsymbol{X}^{t+1},\boldsymbol{X}^{t})\tag{28}$$

则由推导可得:

$$J(\boldsymbol{X}^{t})=Z(\boldsymbol{X}^{t},\boldsymbol{X}^{t})\geqslant Z(\boldsymbol{X}^{t+1},\boldsymbol{X}^{t})\geqslant J(\boldsymbol{X}^{t+1})\tag{29}$$

此时$J(\boldsymbol{X}^{t})$单调递减。因此基于这种条件,首先需要构造一个合适的辅助函数$Z(\boldsymbol{X},\boldsymbol{X}')$使得$J(\boldsymbol{X}^{t})$单调递减,其次需要求得其最小值。根据下文中的命题3,在式(31)中定义的是J的辅助函数$Z(\boldsymbol{X},\boldsymbol{X}')$,其最小值由式(32)给出。根据式(28),有$\boldsymbol{X}^{(t+1)}\leftarrow\boldsymbol{X}$,$\boldsymbol{X}^{(t)}\leftarrow\boldsymbol{X}'$,用$\boldsymbol{A}=\boldsymbol{F}\boldsymbol{F}$便还原出式(10)的更新规则。

首先构造关于矩阵 $\boldsymbol{Y}$ 的辅助函数。由于 $\lambda\boldsymbol{I}$ 是一个常数矩阵，因此在以下证明步骤中将其省略。而需要证明的式(10)中 $\boldsymbol{Y}$ 的更新步骤恰好是式(28)中所构辅助函数的更新。

命题 3

对于满足如下公式形式的函数 $J(\boldsymbol{X})$：

$$J(\boldsymbol{X}) = \mathrm{Tr}(-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{+} + 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{-} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{+}\boldsymbol{X} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{-}\boldsymbol{X}) \tag{30}$$

其中所有矩阵均为非负值，则函数：

$$Z(\boldsymbol{X},\boldsymbol{X}') = -\sum_{ij}2B_{ij}^{+}X'\left(1+\log\frac{X_{ij}}{X'_{ij}}\right)+\sum_{ij}B^{-}\frac{X_{ij}^{2}+X'^{2}_{ij}}{X'_{ij}}+$$
$$\sum_{ij}\frac{(\boldsymbol{X}'\boldsymbol{A}^{+})_{ij}X_{ij}^{2}}{X'_{ij}}-\sum_{ij}A_{jk}^{-}X'_{ij}X'_{jk}\left(\log\frac{X_{ij}X_{jk}}{X'_{ij}X'_{jk}}\right) \tag{31}$$

为目标函数 $J(\boldsymbol{X})$ 对应的辅助函数，此处的 log 为函数计算值。该辅助函数不仅满足：$J(\boldsymbol{X}) \leqslant Z(\boldsymbol{X},\boldsymbol{X}')$ 和 $J(\boldsymbol{X})=Z(\boldsymbol{X},\boldsymbol{X})$，且是关于 $\boldsymbol{X}$ 的凸函数。故有全局最小值：

$$X_{ij} = \underset{X}{\mathrm{argmin}}\, Z(\boldsymbol{X},\boldsymbol{X}') = X'_{ij}\sqrt{\frac{B_{ij}^{+}+(\boldsymbol{X}'\boldsymbol{A}^{-})_{ij}}{B_{ij}^{-}+(\boldsymbol{X}'\boldsymbol{A}^{+})_{ij}}} \tag{32}$$

证明

$$J(\boldsymbol{X}) = \mathrm{Tr}(-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{+}) + 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{-} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{+}\boldsymbol{X} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}^{-}\boldsymbol{X} \tag{33}$$

为了找到两个正项的上限，两个负项下限，对于函数 $J(\boldsymbol{X})$ 中的第三项，使用命题并令 $\boldsymbol{A}\leftarrow\boldsymbol{I},\boldsymbol{B}\leftarrow\boldsymbol{A}^{+}$，得到一个上限：

$$\mathrm{Tr}(\boldsymbol{X}\boldsymbol{A}^{+}\boldsymbol{X}^{\mathrm{T}}) \leqslant \sum_{ij}\frac{(\boldsymbol{X}'\boldsymbol{A}^{+})_{ij}X_{ij}^{2}}{X'_{ij}} \tag{34}$$

对于函数的第二项，由于任意 $a,b>0$，不等式 $a\leqslant(a^{2}+b^{2})/(2ab)$ 始终成立，故可以推导出其边界如下：

$$\mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}) = \sum_{ij}X_{ij}B_{ij}^{-} \leqslant \sum_{ij}B_{ij}^{-}\frac{X_{ij}^{2}+X'^{2}_{ij}}{2X'_{ij}}\cdot\sqrt{b^{2}-4ac} \tag{35}$$

而对于任意 $z>0$，都有 $z\geqslant 1+\log z$ 始终成立。故可以利用这个不等式，继续推导函数 $J(\boldsymbol{X})$ 其余两个项的下界：

$$\frac{X_{ij}}{X'_{ij}} \geqslant 1+\log\frac{X_{ij}}{X'_{ij}} \tag{36}$$

且

$$\frac{X_{ij}X_{jk}}{X'_{ij}X'_{jk}} \geqslant 1+\log\frac{X_{ij}X_{jk}}{X'_{ij}X'_{jk}} \tag{37}$$

由式(36)可推导出函数 $J(\boldsymbol{X})$ 的第一项的边界：

$$\mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}^{+}) = \sum_{ij}B_{ij}^{+}X_{ij} \geqslant \sum_{ij}\boldsymbol{B}_{ij}^{+}X'_{ij}\left(1+\log\frac{X_{ij}}{X'_{ij}}\right) \tag{38}$$

由式(37)可推导出函数 $J(\boldsymbol{X})$ 的最后一项的边界：

$$\mathrm{Tr}(\boldsymbol{X}\boldsymbol{A}^{-}\boldsymbol{X}^{\mathrm{T}}) \geqslant \sum_{ijk}A_{jk}^{-}X'_{ij}X'_{jk}\left(1+\log\frac{X_{ij}X_{jk}}{X'_{ij}X'_{jk}}\right) \tag{39}$$

汇总以上边界值，就能够验证之前提出的目标式(31)满足 $J(\boldsymbol{X})\leqslant Z(\boldsymbol{X},\boldsymbol{X}')$ 和 $J(\boldsymbol{X})=Z(\boldsymbol{X},\boldsymbol{X})$ 的条件。此时为了求解 $Z(\boldsymbol{X},\boldsymbol{X}')$ 的最小值，对函数进行求导得：

$$\frac{\partial Z(\boldsymbol{X},\boldsymbol{X}')}{\partial X_{ij}} = -2B_{ij}^{+}\frac{X'_{ij}}{X_{ij}}+2B_{ij}^{-}\frac{X_{ij}}{X'_{ij}}+\frac{2(\boldsymbol{X}'\boldsymbol{A}^{+})_{ij}X_{ij}}{X'_{ij}}-$$
$$2\frac{(\boldsymbol{X}'\boldsymbol{A}^{-})_{ij}X'_{ij}}{X_{ij}} \tag{40}$$

此时二阶导数：

$$\frac{\partial^{2}Z(\boldsymbol{X},\boldsymbol{X}')}{\partial X_{ij}^{2}\partial X_{jk}} = \delta_{ij}\delta_{jk}M_{ij} \tag{41}$$

其中：

$$M_{ij} = \frac{4[(\boldsymbol{B}^{+})_{ij}+(\boldsymbol{X}'\boldsymbol{A}^{-})_{ij}]X'_{ij}}{X_{ij}^{2}}+2\frac{B_{ij}^{+}+(\boldsymbol{X}'\boldsymbol{A}^{+})_{ij}}{X'_{ij}} \tag{42}$$

由以上证明可知，辅助函数 $Z(\boldsymbol{X},\boldsymbol{X}')$ 是关于 $\boldsymbol{X}$ 的凸函数，通过设置 $\frac{\partial Z(\boldsymbol{X},\boldsymbol{X}')}{\partial X_{ij}}=0$ 求导，如式(40)所示，整理得到关于 $\boldsymbol{X}$ 的全局最小值即式(32)。

命题 4

对于任意矩阵 $\boldsymbol{A}\in\mathbf{R}_{+}^{n\times n},\boldsymbol{B}\in\mathbf{R}_{+}^{n\times n}$，矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 均为对称矩阵，$\boldsymbol{S}\in\mathbf{R}_{+}^{n\times k},\boldsymbol{S}'\in\mathbf{R}_{+}^{n\times k}$，对于第 i 个点和第 p 个聚类簇进行双重求和，$i\in[1,n],p\in[1,k]$，则有以下不等式始终成立：

$$\sum_{i=1}^{n}\sum_{p=1}^{k}\frac{(\boldsymbol{A}\boldsymbol{S}'B_{ip})S_{ip}^{2}}{S'^{2}_{ip}} \geqslant Tr(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{S}\boldsymbol{B}) \tag{43}$$

证明　令 $S_{ip}=S'_{ip}V_{ip}$，用一个指定符号来表示不等式左右两端的差异值：

$$\Delta = \sum_{i,j=1}^{n}\sum_{p,q=1}^{k}A_{ij}S'_{jq}B_{qp}S'_{ip}(V_{ip}^{2}-V_{ip}V_{jq})$$

由于 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 是对称矩阵，即：

$$\Delta = \sum_{i,j=1}^{n}\sum_{p,q=1}^{k}A_{ij}S'_{jq}B_{qp}S'_{ip}\left(\frac{V_{ip}^{2}+V_{jq}^{2}}{2}-V_{ip}V_{jq}\right)=$$
$$\frac{1}{2}\sum_{i,j=1}^{n}\sum_{p,q=1}^{k}A_{ij}S'_{jq}B_{qp}S'_{ip}(u_{ip}-u_{jq})^{2}$$

在出现 $\boldsymbol{B}=\boldsymbol{I}$ 和 $\boldsymbol{S}$ 为列向量的特殊情况下，此结果的详细表述在文献[29-30]中。

3.2 算法优势对比

根据乘法更新迭代规则，本文设计的改进算法与传统谱聚类算法相比在时间开销上更有优势。传统谱聚类算法花费了更多的时间来求解拉普拉斯矩阵的特征值分解，而由于利用了乘法更新优化，本改进算法在这个方面提供了更有效的解决方案。同时，由于样本数量和类别的增加，拉普拉斯矩阵特征值分解的时间快速增长，在处理高维数据集时需要花费更多时间。

由于非负约束和正交约束为聚类过程提供了更好的指示矩阵，这使得在后续处理的步骤(例如 k-means)中能够得到更好的聚类结果。因此，基于乘法更新迭代的算法在聚类性能方面也略胜于传统谱聚类算法。

4 实验验证

4.1 实验环境

由于文中涉及对算法性能的评估，实验环境可能会对最终结果造成一定影响，因此本文在实验过程中对于实验公平性做了充分考虑。实验过程中使用 Matlab 语言统一对谱聚类算法的输入和输出接口进行了设定，输入接口为数据样本及其相似度矩阵、相关类数及参数，输出接口为聚类结果。

在实验过程中，提取优化了原作者所提供文献或代码中的算法公用模块，即相似度矩阵的计算、拉普拉斯矩阵的构建及 k-means 步骤等部分。例如，实验在完成 IFSC、SC 算法最后一步的 k-means 算法步骤中，并没有使用 Matlab 自带的内置实现，从而避免了当数据量较大时，计算内存不足使用到磁盘的交换区而影响到以秒为精确度的算法评估结果，进行了单独的抽离和统一的调用。在使用 k-means 算法进行矩阵运算时，统一设定阈值使分批计算的步骤能够在内存中完成；并对算法公用模块进行了优化，从而避免公用模块与内置实现差距太大导致的实验误差。

在准确性评估中的计算都基于式(1)中高斯核函数：

$$W_{ij} = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right)$$

实验中将计算相似度指定为统一的带宽参数σ,且统一设置了各个算法所需要的采样点数m,低秩估计值统一设定为k。

本文中的所有实验都是在一台8 GB DDR3内存和主频为2.8 GHz的Intel Pentium CPU的计算机上进行的,计算机的系统环境为Windows 10 64位,实验均在Matlab R2014a中完成。为了防止计算机的读写速度对实验结果造成影响,已经将实验的输入数据在测试前完成读写预热。

4.2　实验数据

为了在实验中对IFSC算法的聚类性能进行验证测试,实验分别在由UCI(University of California Irvine)机器学习数据库中选取的5个真实数据集上和3个人工合成数据集上进行验证。

4.2.1　真实数据集

以下给出实验部分所使用的真实数据集的介绍:

1)Mnist数据集是一个手写数字数据集,数据集包含60 000个用于训练的示例和10 000个用于测试的示例。这些数字已经过尺寸标准化并位于图像中心,图像为固定大小(28像素×28像素),每个图像都被平展并转换为784个特征的一维numpy数组。本文实验使用了Mnist数据集中包含5 000、7 500、10 000、12 500、15 000、17 500和20 000个样本的子集。

2)Corel数据集是图像处理应用任务中广泛使用的数据集,包含巴士、建筑、恐龙等10个类别的图片,有很好的颜色、纹理、形状等144个属性特征,利用这些属性特征能够描述并区别每一幅图片的类别。在本文实验中,分别使用2、6和10类的3个子集来用于评估实验中的算法。

3)WebKB数据集[31]中的示例是从4所大学(康奈尔大学、德克萨斯大学、华盛顿大学、威斯康星大学)数据集主页中下载的网页,这些网页相应的标记分类为学生、教职员工、教职员工、部门、课程、项目以及其他。

4)RCV1数据集[32]是人工对新闻故事分类整理得到文本分类测试集合,每篇文档都是由一个词频-逆向文件频率(Term Frequency-Inverse Document Frequency, TF-IDF)向量表示,实验中选取了一个RCV1子集,包含1 925个文档,包含29 992个不同的词,包括四个类别"C15""ECAT""GCAT"和"MCAT"。

5)Waveform数据集是常用于分类和聚类任务中的噪声波形数据集,波形数据被分为3类且各占33%,在第40维属性之后的19维为噪声数据,噪声的均值为0、方差是1。

由于真实数据集中含有更多的噪声,且样本边界模糊,因此,在实验过程中调整了部分数据集的大小,表2中给出了实验中使用数据集的详细信息。

表2　实验中使用的真实数据集

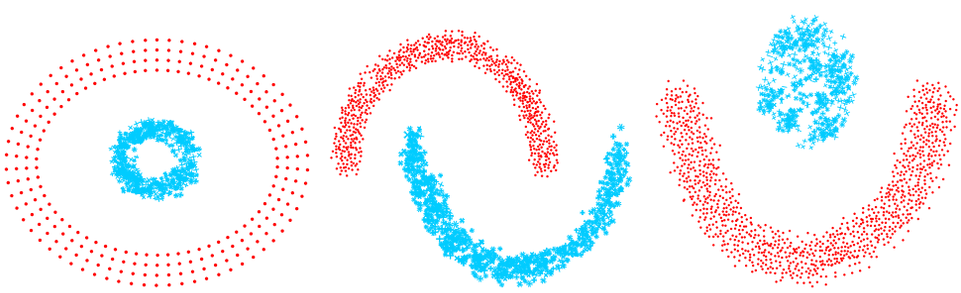
Tab. 2　Real datasets used in experiments

数据集名称	样本数	维数	类数
Mnist	60 000	784	10
Corel	2 074	144	18
WebKB	814~1 210	4 029~4 189	7
RCV1	1 925	29 992	4
Waveform	1 658	40	3

4.2.2　人工数据集

为了进一步验证算法的正确性,实验中除使用了真实数据集外,还在三个聚类任务中常使用的人工合成数据集[33]上进行了验证,其主要目的是验证IFSC算法的有效性,即在不需要特征值分解的情况下同样能够达到相应的聚类效果。图1中给出了所使用的合成数据集的流形结构。

根据流形结构将数据集分别命名为环形分布数据集Circle Cluster(CC)、双螺旋结构数据集Two Spirals(TS)和双月型数据集Two Moons(TM)。实验在每个合成数据集中选取了10 000个样本点并划分为两个类簇。此时在合成数据集,尤其是环形分布数据集CC和双螺旋结构数据集TS中,仅依靠数据样本点间距离作为标准的聚类算法就很难得到较好鲁棒的聚类结果。



(a) Circle Cluster　(b) Two Spirals　(c) Two Moons

图1　合成数据集的流形结构

Fig. 1　Manifold structures of synthetic datasets

4.3　评价指标

使用标准化互信息(Normalized Mutual Information, NMI)来评估不同聚类方法的表现性能,即互信息分数的归一化,用熵做分母将结果调整到0与1之间,由如式(44)来定义:

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,j}\log\frac{n_{i,j}}{n_i\hat{n}_j}}{\sqrt{\left(\sum_{i=1}^{c} n_i\log\frac{n_i}{n}\right)\left(\sum_{i=1}^{c}\hat{n}_i\log\frac{\hat{n}_i}{n}\right)}} \tag{44}$$

其中,n_i、n_j、n_{ij}分别代表样本属于聚类簇$C_i(1\leqslant i\leqslant c)$、属于类别$L_j(1\leqslant j\leqslant c)$和同时属于两者时的样本数量。NMI值越大,则聚类结果越好。

4.4　实验过程

为了验证本文所设计IFSC算法的正确性、准确率和效率,在实验中对以下方法进行比较:

1)使用k-means(KM)聚类算法。

2)使用本文所设计的基于乘法更新迭代思想的快速谱聚类(IFSC)算法。

3)传统谱聚类(SC)算法,即需要特征值分解的谱聚类

算法。

4)层次聚类(Hierarchical Clustering, HC)算法。

实验结果除了给出本文算法与传统谱聚类的性能对比,还能得到IFSC算法相较于常用的基于距离迭代和基于层次分解的聚类算法在进行大规模数据集处理时的性能优点。

4.5 正确性和收敛性

4.5.1 聚类性能比较

在真实数据集上的实验主要用来评估算法正确率和执行性能,实验结果中,需要将速度和准确率两方面结合来综合评价算法的表现;而在人工合成数据集上的实验主要是验证所设计的算法是否符合谱聚类的一般特征。

表3中展示了不同算法在不同真实数据集上的执行速度和准确率。通过表3可以看出,本文设计的IFSC算法与传统的谱聚类(SC)算法相比,极大地降低了算法所需要的时间成本,当选用Corel数据集,类数 $Cls = 10$ 时,所消耗的时间仅是SC算法的28.4%,原因是IFSC算法避免了对拉普拉斯矩阵的创建和分解,在处理真实数据集,尤其数据规模较大和维数较高时最有效。考虑到Matlab内置的特征值分解是高度优化的版本,迭代的方法完全是代码,实际性能提升可以更高。另外可以看到,SC算法虽然在处理WebKB和Waveform数据集时,在选用Washington、Texas、Wiscosin这三个类上进行聚类的NMI值优于IFSC算法,但仅增高了约21%,并且可以观察到,在其他四个数据集上的聚类结果指标均不如IFSC算法有效。结合表3中的速度对比,证明了不需要特征值分解的乘法迭代算法能够完成谱聚类过程中对数据矩阵的度量和处理。

观察IFSC算法和KM算法的速度性能对比可以发现,随着数据集维数的升高,KM算法的运算消耗时间急剧增长。虽然KM算法在Mnist数据集上的速度性能表现良好,但在数据集RCV1此类高维数据集上表现较差。正如1.2节中提到的,传统谱聚类算法在处理大规模和高维数据时需要在计算相似度矩阵方面消耗大量时间,从表3中计算时间的对比可以发现,它的速度性能与KM相比虽然有所提升,但效果仍然不佳。通过以上比较可以看出,IFSC算法在处理小型数据集时的计算时间与 k-means方法和SC类似。当数据集规模和维数较大时,IFSC算法的时间开销能少,速度性能更优,这是由于IFSC算法简化了计算相似度矩阵所需要的复杂步骤,且同时能有效处理谱分解问题。

表4中展示了不同算法在不同人工合成数据集上的执行速度和准确率。由实验结果可以观察到,使用KM算法的聚类结果准确性很低,这是由于KM方法是通过将距离最近样本分配给最近的聚类中心,这种忽略数据全局分布的特点会导致它在处理流形数据集时的聚类能力有限。此外,可以观察到,HC算法在处理合成数据集时能表现出相对较好的聚类性能,因此更适用于处理边界清晰的合成数据;但由于对含噪声和模糊样本边界的数据较敏感,在处理真实数据集时表现不佳。

表3 真实数据集的聚类结果

Tab. 3 Clustering results on real datasets

数据集		KM算法		IFSC算法		SC算法		HC算法	
		NMI	时间/s	NMI	时间/s	NMI	时间/s	NMI	时间/s
Mnist	*No*=5 000	0.47	6.96	**0.52**	15.63	0.47	17.36	0.02	8.95
	No=7 500	0.47	11.63	**0.48**	30.23	0.47	42.93	0.02	20.03
	No=10 000	0.49	12.84	**0.52**	52.74	0.48	78.95	0.02	44.84
	No=12 500	0.48	31.36	0.48	83.80	**0.49**	137.36	0.02	60.27
	No=15 000	0.48	22.31	0.47	119.58	**0.48**	215.32	0.01	101.34
	No=17 500	0.48	36.65	**0.49**	162.72	0.47	288.51	0.01	118.58
	No=20 000	0.48	37.18	**0.50**	214.27	0.47	428.67	0.01	154.67
Corel	*Cls*=2	0.69	19.42	**0.76**	13.41	0.74	48.24	0.02	13.04
	Cls=6	0.61	362.37	0.67	78.47	**0.70**	306.04	0.02	81.84
	Cls=10	0.63	717.83	0.62	172.75	**0.64**	682.24	0.04	193.78
WebKB	Corell	0.12	6.59	**0.22**	2.03	0.17	5.24	0.02	1.52
	Washington	0.21	6.78	0.22	2.05	**0.26**	4.26	0.05	1.14
	Texas	0.16	6.05	0.22	4.11	**0.27**	11.17	0.04	1.79
	Wiscosin	0.13	10.27	0.24	4.59	**0.27**	12.11	0.03	1.65
RCV1	*Cls*=4	0.12	168.54	**0.32**	47.28	0.27	200.07	0.01	53.84
Waveform	*Cls*=3	0.47	226.88	0.46	33.63	**0.49**	97.76	0.13	23.78
平均值		0.40	116.36	**0.45**	61.73	**0.45**	158.83	0.03	53.23

表4 合成数据集的聚类结果

Tab. 4 Clustering results on synthetic datasets

算法	环形分布数据集(CC)		双螺旋结构数据集(TS)		双月型数据集(TM)		平均值	
	NMI	时间/s	NMI	时间/s	NMI	时间/s	NMI	时间/s
KM算法	0.01	0.05	0.17	0.03	0.35	0.02	0.18	0.03
HC算法	1.00	2.52	1.00	3.23	1.00	3.34	1.00	3.03
SC算法	1.00	9.07	0.81	8.35	1.00	8.48	0.94	8.63
IFSC算法	1.00	7.86	0.95	11.27	0.95	15.36	0.97	11.49

其次,从表4中可以看出,IFSC算法在TM数据集上结果较好,但是在CC和TS数据集上则需要更多的迭代才能获得更好的结果,这是因为CC和TS数据集符合流形分布。尽管如此,在相同精度下,IFSC相较于传统SC方法在处理现实任务中更有效。本文中的IFSC算法利用基于指示矩阵的乘法迭代完成聚类,验证结果表明可达到与传统谱聚类相当的聚

类效果。

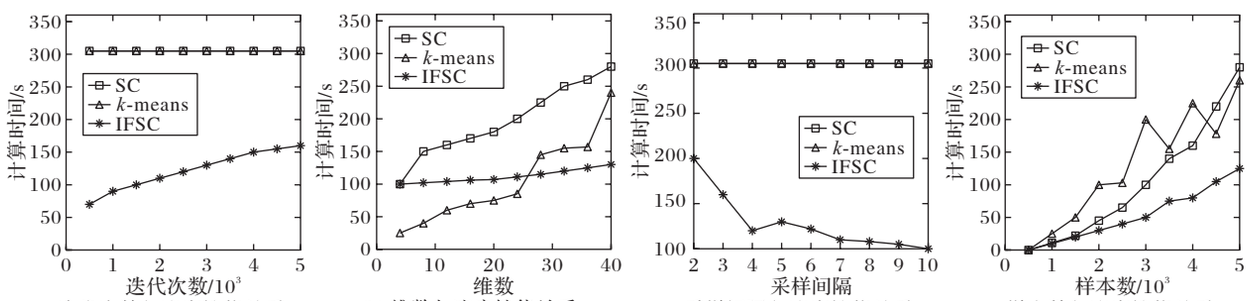
4.5.2　计算时间比较

为了比较IFSC算法的速度性能,在数据集Waveform上进行了实验验证。计算时间主要受到迭代次数、采样间隔、样本维数和样本大小的影响,因此在实验过程中控制变量,将迭代次数设置为5 000,采样间隔设置为5,且Waveform数据集的维度为40,样本个数为5 000。通过保持其中三个变量不变来评估剩余的一个变量,每种方法分别独立运行20次,参数λ = 0.5。实验结果如图2所示。

首先,观察图2中(a)和(b)的结果可以看出,在其他条件相同的情况下,同k-means方法和传统谱聚类(SC)方法相比,本文所设计的基于乘法更新迭代的快速谱聚类(IFSC)算法在处理大规模高维数据集时仅花费了一半的时间。

其次,由图2(b)和(d)中的结果分析样本维数和样本大小对算法性能的影响可知,使用k-means方法所需的运行时间随着样本维数大小的升高而快速增加。

此外,从图2(c)和(d)中还可以看出:传统的谱聚类方法对样本量的增加更敏感,这是由于谱聚类方法在计算特征向量这一运算步骤中需要$O(n^3)$的计算复杂度,其中n为样本个数。



(a) 迭代次数与速度性能关系　(b) 维数与速度性能关系　(c) 采样间隔与速度性能关系　(d) 样本数与速度性能关系

图2　不同方法的运行时间比较

Fig. 2　Comparison of running time of different methods

5　结语

本文简要介绍了谱聚类技术处理复杂数据集时需要解决的两个主要问题,即相似度矩阵构造和拉普拉斯矩阵的特征值分解。基于乘法更新迭代规则,在聚类指示矩阵$\boldsymbol{Y}$的基础上设计了一种快速谱聚类优化(IFSC)算法。该算法利用Nyström方法对数据集随机采样后,根据由子矩阵表示出的指示矩阵完成迭代更新。实验结果表明,所设计的算法在保证聚类精度的同时,提高了传统谱聚类方法的效率,弥补了谱聚类在处理大样本数据集时需要对拉普拉斯矩阵完成特征分解的时间消耗缺陷。接下来工作将使用其他采样方法(如自适应采样方法)来完成算法设计模型的更新迭代,并从理论层面上进一步分析所设计的算法的误差界和泛化界。

参考文献 (References)

[1] 朝乐门,邢春晓,张勇. 数据科学研究的现状与趋势[J]. 计算机科学,2018,45(1):1-13. (CHAO L M, XING C X, ZHANG Y. Data science studies: state-of-the-art and trends [J]. Computer Science, 2018, 45(1): 1-13.)

[2] CAO L. Data science: a comprehensive overview [J]. ACM Computing Surveys, 2017, 50(3): Article No. 43.

[3] WANG L, DONG M. Multi-level low-rank approximation-based spectral clustering for image segmentation [J]. Pattern Recognition Letters, 2012, 33(16): 2206-2215.

[4] 周莉莉,姜枫. 图像分割方法综述研究[J]. 计算机应用研究,2017,34(7):1921-1928. (ZHOU L L, JIANG F. Survey on image segmentation methods [J]. Application Research of Computers, 2017, 34(7): 1921-1928.)

[5] NIE F, ZHU W, LI X. Unsupervised feature selection with structured graph optimization [C]// Proceedings of the 2016 30th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2016: 1302-1308.

[6] WANG Q, ZHANG F, LI X. Optimal clustering framework for hyperspectral band selection [J]. IEEE Transaction on Geoscience and Remote Sensing, 2018, 56(10): 5910-5922.

[7] GU G, HOU Z, CHEN C, et al. A dimensionality reduction method based on structured sparse representation for face recognition [J]. Artificial Intelligence Review, 2016, 46(4): 431-443.

[8] 管涛,李玉玲. 大规模矩阵降维的随机逼近方法[J]. 数学的实践与认识,2016,46(24):184-193. (GUAN T, LI Y L. Stochastic approximation approaches of large-scale matrix dimension reduction [J]. Mathematic in Practice and Theory, 2016, 46(24): 184-193.)

[9] WU J, XIONG H, CHEN J. Towards understanding hierarchical clustering: a data distribution perspective [J]. Neurocomputing, 2009, 72(10/11/12): 2319-2330.

[10] NAGPAL A, JATAIN A, GAUR D. Review based on data clustering algorithms [C]// Proceedings of the 2013 IEEE Conference on Information and Communication Technologies. Piscataway: IEEE, 2013: 298-303.

[11] WU J, LIU H, XIONG H, et al. K-means-based consensus clustering: a unified view [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(1): 155-169.

[12] WANG Y, CHEN L. Multi-view fuzzy clustering with minimax optimization for effective clustering of data from multiple sources [J]. Expert Systems with Applications, 2017, 72: 457-466.

[13] KADIR S N, GOODMAN D F M, HARRIS K D. High-dimensional cluster analysis with the masked EM algorithm [J]. Neural Computation, 2014, 26(11): 2379-2394.

[14] YANG Y, MA Z, YANG Y, et al. Multitask spectral clustering by exploring intertask correlation [J]. IEEE Transactions on Cybernetics, 2015, 45(5): 1083-1094.

[15] NG A Y, JORDAN M I, WEISS Y. On spectral clustering: analysis and an algorithm [C]// Proceedings of the 2001 14th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2001: 849-856.

[16] VON LUXBURG U. A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395-416.

[17] FOWLKES C, BELONGIE S, CHUNG F, et al. Spectral grouping

using the Nyström method [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(2): 214-225.

[18] 丁世飞,贾洪杰,史忠植. 基于自适应Nyström采样的大数据谱聚类算法[J]. 软件学报,2014,25(9):2037-2049. (DING S F, JIA H J, SHI Z Z. Spectral clustering algorithm based on adaptive Nyström sampling for big data analysis [J]. Journal of Software, 2014, 25(9): 2037-2049.)

[19] CHEN X, CAI D. Large scale spectral clustering via landmark-based sparse representation [C]// Proceedings of the 2011 25th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2011: 313-318.

[20] MARTIN L, LOUKAS A, VANDERGHEYNST P. Fast approximate spectral clustering for dynamic networks [C]// Proceedings of the 2018 35th International Conference on Machine Learning. New York: International Machine Learning Society, 2018: 3423-3432.

[21] 叶茂,刘文芬. 基于快速地标采样的大规模谱聚类算法[J]. 电子与信息学报,2017,39(2):278-284. (YE M, LIU W F. Large scale spectral clustering based on fast landmark sampling [J]. Journal of Electronics and Information Technology, 2017, 39(2): 278-284.)

[22] CHEN W Y, SONG Y, BAI H, et al. Parallel spectral clustering in distributed systems [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568-586.

[23] 张涛,唐振民,吕建勇. 一种基于低秩表示的子空间聚类改进算法[J]. 电子与信息学报,2016,38(11):2811-2818. (ZHANG T, TANG Z M, LYU J Y. Improved algorithm based on low rank representation for subspace clustering [J]. Journal of Electronics and Information Technology, 2016, 38(11): 2811-2818.)

[24] GALLIER J. Spectral theory of unsigned and signed graphs applications to graph clustering: a survey [J]. Computing Research Repository, 2016, 16(4): 1601-692.

[25] SHI J, MALIK J. Normalized cuts and image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[26] NIE F, DING C, LUO D, et al. Improved minmax cut graph clustering with nonnegative relaxation [C]// Proceedings of the 2010 Joint European Conference on Machine Learning and Knowledge Discovery in Databases, LNCS 6322. Berlin: Springer, 2010: 451-466.

[27] TÜRKMEN A C. A review of nonnegative matrix factorization methods for clustering [EB/OL]. [2020-05-10]. https://www.researchgate.net/profile/Ali_Caner_Turkmen/publication/280062357_A_Review_of_Nonnegative_Matrix_Factorization_Methods_for_Clustering/links/57fd28a908ae49db475537b0.pdf.

[28] DING C H Q, LI T, JORDAN M I. Convex and semi-nonnegative matrix factorizations [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 32(1): 45-55.

[29] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization [C]// Proceedings of the 2000 13th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2000: 535-541.

[30] LEE D D, SEUNG H S. Unsupervised learning by convex and conic coding [C]// Proceedings of the 1996 9th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 1996: 515-521.

[31] CRAVEN M, DIPASQUO D, FREITAG D, et al. Learning to extract symbolic knowledge from the World Wide Web [C]// Proceedings of the 1998 15th National on Artificial Intelligence/10th Conference on Innovative Applications of Artificial Intelligence. Palo Alto: AAAI Press, 1998: 509-516.

[32] SEMERTZIDIS T, RAFAIIDIS D, STRINTZIS M G, et al. Large-scale spectral clustering based on pairwise constraints [J]. Information and Management, 2015, 51(5): 616-624.

[33] ZELNIK-MANOR L, PERONA P. Self-tuning spectral clustering [C]// Proceedings of the 2004 17th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2004: 1601-1608.

This work is partially supported by the National Natural Science Foundation of China (61872260), the Natural Science Foundation of Shanxi Province (201703D421013).

LIU Jingshu, born in 1997, M. S. candidate. Her research interests include big data, matrix decomposition.

WANG Li, born in 1971, Ph. D., professor. Her research interests include online social network computing, mobile communication.

LIU Jinglei, born in 1970, Ph. D., professor. His research interests include graph model reasoning, matrix decomposition.