计算机科学
COMPUTER SCIENCE

基于相对邻近度的自适应谱聚类算法

原泽菲, 张正军, 姜国林

引用本文

原泽菲, 张正军, 姜国林. 基于相对邻近度的自适应谱聚类算法[J]. 计算机科学, 2025, 52(10): 79-89.

YUAN Zefei, ZHANG Zhengjun, JIANG Guolin. Adaptive Spectral Clustering Algorithm Based on Relative Proximity [J]. Computer Science, 2025, 52(10): 79-89.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240800102

基于相对邻近度的自适应谱聚类算法

原泽菲　张正军　姜国林
南京理工大学数学与统计学院　南京 210094
(daichongtou1999@163.com)

摘　要　针对以高斯核函数为相似性度量的传统谱聚类算法需人为设置尺度参数,相似度与样本分布结构无关的问题,定义了在自然 k 近邻基础上的共享邻居,结合数据点的近邻信息构造了能反映区域密度的多尺度参数,以新的尺度参数重新定义了相似性度量,提出了一种基于相对邻近度的自适应谱聚类算法(Adaptive Spectral Clustering based on Relative Proximity,RPASC)。改进的尺度参数结合了间隔尺度、顺序尺度及比例尺度等特性,体现了数据点之间的相对位置关系,反映了不同密度簇的分布特征和空间结构,提高了算法对不同分布数据集的适应性。新的相似性度量通过灵活调整局部尺度参数的大小,自适应地缩小不同密度簇边界上数据点的相似度,使聚类的簇边界更明确,有利于发现真实的簇形态。通过在人工合成数据集和 UCI 真实数据集上进行的实验,验证了 RPASC 算法在多个聚类性能指标上的有效性。

关键词:谱聚类;多尺度参数;共享邻居;自然 k 近邻;相似性度量
中图分类号　TP301.6

Adaptive Spectral Clustering Algorithm Based on Relative Proximity

YUAN Zefei,ZHANG Zhengjun and JIANG Guolin
School of Mathematics and Statistics,Nanjing University of Science and Technology,Nanjing 210094,China

Abstract　The traditional spectral clustering algorithm with Gaussian kernel function as the similarity measure has the problem that the scale parameter needs to be artificially set,and the similarity is not related to the sample distribution structure. In order to solve this problem,the shared neighbors based on the natural k-nearest neighbors are defined,and a multi-scale parameter reflecting the regional density is constructed based on the nearest neighbors information of the data points,and the similarity measure is redefined with the new scale parameter. This paper proposes an adaptive spectral clustering algorithm based on relative proximity(RPASC). The improved scale parameter combines the characteristics of interval scale,sequence scale and proportional scale,embodying the relative position relationship between data points and reflecting the distribution characteristics and spatial structure of different density clusters,which improves the adaptability of the algorithm to different distribution datasets. The new similarity measure adaptively reduces the similarity of data points on cluster boundaries of different densities by flexibly adjusting the size of local scale parameter,making cluster boundaries more explicit,which is conducive to discovering the true cluster morphologies. Experiments on synthetic datasets and UCI real datasets verify the effectiveness of the RPASC algorithm on multiple clustering performance indicators.

Keywords　Spectral clustering,Multi-scale parameter,Shared neighbors,Natural k-nearest neighbors,Similarity measure

1　引言

聚类的最初目的是把具有相似特性的实物划分到一起,在实际应用中,聚类的定义通常取决于聚类对象的性质和期望得到的结果[1]。尽管聚类的定义并不十分统一,但是聚类分析还是表达了一般认为的“类内相似且类间相异”的目标,即数据集合分组为多个子集后,同子集数据间相似度最大化,不同子集数据间相似度最小化。通过聚类分析,数据对象被划分为具有现实意义的集群,方便人类分析和描述世界。聚类分析在生态学、医学、经济学、语言学、心理学等众多领域中都起着重要的作用。

经典的聚类算法,如基于划分的 K-means 算法和基于层次的 BIRCH 算法,它们易于理解且计算简单,但存在不能识别非凸数据集、初始化敏感等明显缺陷。后来,学者提出了基于密度的 DBSCAN 算法,也存在不能很好地反映高维数据、易受截断距离影响等局限。

Shi 等[2]于 2000 年提出了一种由谱图理论演化而来的谱聚类算法(Spectral Clustering,SC),该方法不用对数据的全局结构作假设,具有易收敛于全局最优解[3]、对数据分布适应性强[4]、对高维数据支持较好[5]等特点,因此被广泛应用于

到稿日期:2024-08-20　返修日期:2024-11-29
基金项目:国家自然科学基金(61773014)
This work was supported by the National Natural Science Foundation of China(61773014).
通信作者:张正军(zzjnj@163.com)

数据挖掘[6]、图像分割[7-9]、模式识别[10]等领域。

尽管谱聚类算法在实践中取得了比较好的效果，但有一些问题仍需进一步研究和解决。由于谱聚类算法最终进行聚类的对象是处理后的特征向量，特征向量是由数据集的相似矩阵经过处理后再运用特征分解得到的，因此相似矩阵的质量对算法最终的聚类效果有十分重要的影响。近年来，很多基于改进相似性度量的谱聚类算法被提出。Wang 等[11]借鉴限制与测度融合方法，采用图最短路径长度生成密度敏感的距离测度，并用成对先验信息对相似矩阵进行监督矫正。Tao 等[12]在距离测度中引入伸缩因子来反映数据分布的全局一致性和局部一致性特征，并增加相对密度敏感项来避免孤立噪声的影响。Manor 等[13]提出可以根据每个点自身的邻域信息，为其计算一个自适应的尺度参数，构造自调节的高斯核函数。Kong 等[14]在 Manor 的基础上，将数据点周围 n 个近邻计算加权距离和作为其局部尺度的值，从而实现尺度参数的自动选取。Zhang 等[15]引入共享近邻的定义，提出基于共享近邻的高斯核函数作为相似性度量，通过共同邻居的作用，拉近相同簇数据点的距离。Zhao 等[16]利用局部密度差来调整簇类样本点之间的相似度，提出一种改进的密度敏感的自适应谱聚类算法。Zhang 等[17]提出一种基于密度系数和共享近邻的谱聚类算法，通过计算样本点的密度系数阈值选取权值，构造加权的自适应核参数并结合共享近邻数计算样本点之间的相似度。Zhao 等[18]提出基于样本间变异系数改进的自适应谱聚类算法，该算法定义能够反映样本数据分布信息的变异系数，并以此构造局部尺度参数。Ge 等[19]通过一种无参数的密度自适应邻域构建方法构建无向图，将共享最近邻作为衡量样本之间的相似性度量，进而消除参数对构建相似图的影响，体现全局和局部的一致性。

针对谱聚类算法中相似矩阵对尺度参数敏感的问题，本文在现有研究的基础上，提出一种基于相对邻近度的自适应谱聚类算法(Adaptive Spectral Clustering based on Relative Proximity，RPASC)。本文算法用自适应局部尺度参数代替传统谱聚类中的全局统一尺度参数，降低了人为选取参数的随机性，且无需对数据的分布类型作出限制。该算法挖掘数据点的自然近邻信息，探索其共享邻居的相对位置关系，结合间隔尺度、顺序尺度和比例尺度的特性，构造能更好反映数据区域密度的多尺度参数，以此得到比传统谱聚类更能体现数据分布特征的相似性度量。最后，通过在人工合成数据集和 UCI 真实数据集上的实验验证了算法的性能。

2 谱聚类算法描述与分析

2.1 谱聚类算法

谱聚类是以图论当中的谱图理论为基础，构造邻接图，将聚类任务转换为图的最优划分问题，再将图划分问题转换成拉普拉斯矩阵特征值的问题，使得较为抽象的聚类问题变得具体可求解。谱聚类算法求解的核心思想是：转换数据聚类的特征空间，在新选择的特征空间中对数据运用基础算法聚类，将结果映射回原数据空间[20]。谱聚类算法的重点在于，利用数据信息构建可描述其特性的相似矩阵 $\boldsymbol{W}$，根据相似矩阵 $\boldsymbol{W}$ 计算拉普拉斯矩阵 $\boldsymbol{L}$，由 $\boldsymbol{L}$ 的特征向量构造新的解空间，在新空间中通过 K-means 聚类得到最终结果。

设有数据集 $X=\{x_1,x_2,\cdots,x_N\}$，$x_i=\{x_{i1},x_{i2},\cdots,x_{im}\}$，$x_{iv}$ 表示第 i 个数据点的第 v 维属性值，$v=1,2,\cdots,m$。根据数据分布构造相似性图，可得到数据点间的相似程度。在谱聚类算法中，一般用高斯核函数作为数据点间的相似性度量，高斯核函数基于距离度量，将数据映射到高维空间中，数据在高维空间中更容易被分离。对于任意两点，由高斯核函数定义的相似度计算式如下：

$$w_{ij}=\exp\left(-\frac{d^2(x_i,x_j)}{2\sigma^2}\right) \tag{1}$$

其中，d 表示计算数据点间距离，一般设置为欧氏距离；σ 为尺度参数，用于控制函数的局部衰减速度。相似度计算对 σ 的变化较为敏感，对于不同点集，σ 均需多次实验才能确定最佳取值。相似矩阵 $\boldsymbol{W}$ 抽象地表达了数据集的基本特征，谱聚类算法的性能很大程度上取决于相似矩阵 $\boldsymbol{W}$ 的质量，因此合理定义相似性度量十分关键。

图拉普拉斯矩阵是谱聚类的重要工具，分为非规范拉普拉斯矩阵和规范拉普拉斯矩阵，也非规范拉普拉斯矩阵定义为：

$$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W} \tag{2}$$

其中，$\boldsymbol{D}$ 是对角矩阵，也称为度矩阵，其对角元素为度值，计算方式如式(3)所示：

$$d_{ij}=\sum_j w_{ij} \tag{3}$$

两种形式的规范拉普拉斯矩阵如式(4)和式(5)所示：

$$\boldsymbol{L}_{\text{sym}}=\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}} \tag{4}$$

$$\boldsymbol{L}_{\text{rw}}=\boldsymbol{D}^{-1}\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{W} \tag{5}$$

其中，$\boldsymbol{I}$ 为单位矩阵。

在机器学习特征提取中，最大特征值所对应的特征向量的方向通常包含了最多的信息量[21]，因此，一般可选取拉普拉斯矩阵前 c 个特征值所对应的特征向量构造新的解空间，在新空间上采用 K-means 聚类算法得到最终的 c 个类。谱聚类将原始数据点转换为新的解空间上的点，这种表示的转换增强了数据中的集群属性，使 K-means 算法检测到不同集群的难度相比变换前大大降低。

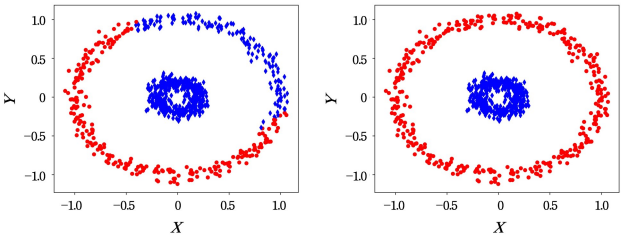
2.2 谱聚类算法的缺陷

本文主要针对谱聚类算法存在的以下缺陷进行研究。

1)尺度参数的选取问题。尺度参数 σ 控制高斯核函数的局部作用范围，当两点距离处于某一个区间范围内时，σ 对函数的影响很大，取不同 σ 值时函数的衰减速度会有明显差异。图 1(a)和图 1(b)分别为同一数据集上高斯核函数取不同 σ 值的聚类结果，相同形状颜色的点表示识别为同簇数据。可以看到，当 $\sigma=4.5$ 时数据点被正确聚类，但 $\sigma=4.0$ 时聚类效果不佳。在实际问题中，很难找到适合一个数据集的全局 σ 值，使谱聚类获得令人满意的结果。在一些数据分布较复杂的问题中，σ 也并非越大越好，且由于人工取值的随机性，一般需进行多次实验比较后才能决定 σ 的取值，其过程费时费力。

2)相似度的计算未能合理利用数据分布结构信息。在传统谱聚类算法中，相似矩阵 $\boldsymbol{W}$ 的构造仅利用了数据点之间的距离信息。根据高斯核函数的定义，指定 σ 值后，相似度的大小仅与点间距离有关，但实际上，相似度也会受数据点所处邻域环境的影响。如图 2 所示，point 2 与 point 3 到 point 1 的距离相等，但 point 1 与 point 3 处于同一个较高密度区域，倾向于被划分到 cluster 1，所以两点应具有比较高的相似度；

point 1 与 point 2 之间存在数据点分布稀疏的区域，point 2 倾向于被划分到 cluster 2，与 point 1 不同簇，两点的相似度应相对较低。显然，已有的计算式不能满足对相似度的要求。



(a)$\sigma=4.0$ (b)$\sigma=4.5$

图 1 谱聚类算法在同一数据集上取不同 σ 值的聚类结果

Fig. 1 Clustering results of SC algorithm on the same dataset with different σ values

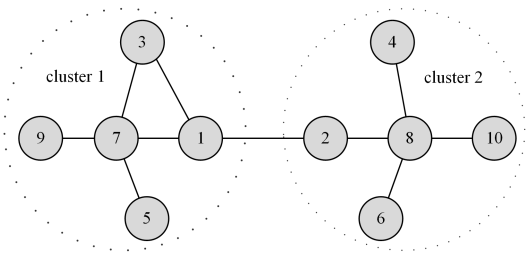


图 2 点 1、点 2 和点 3 的示意图

Fig. 2 Sehematic diagram of point 1, point 2 and point 3

3 RPASC 算法

3.1 最近邻关系

在研究数据属性特征的过程中，最近邻居的概念被提出。根据数据聚类的局部一致性特征，在空间位置上相邻的数据点具有更高的相似性。最近邻居代表距离当前数据点最近的点，在分布较密集的区域，数据点与其最近邻之间距离较近；在分布较稀疏的区域，数据点与其最近邻之间距离较远。因此，最近邻能够很好地揭示数据点与其附近点的亲密程度，可以有效地描述数据的局部信息。最近邻还可以根据数据分布或密集或稀疏的情况来调整邻域半径的大小，降低某些参数带来的不确定性影响。常用的最近邻方法是由 Stevens[22] 提出的 k-邻域和 ε-邻域。本文将在 k-邻域的基础上研究最近邻关系并改进相似性度量。

定义 1(k-最近邻) 设有数据集 $X=\{x_1,x_2,\cdots,x_N\}$，对于 X 中的任意点 x_i，与其距离最近的 k 个数据点被称为点 x_i 的 k-最近邻。一般以欧氏距离作为距离度量，将与点 x_i 最近的数据点称为第一最近邻，记为 x_{i1}；将除 x_{i1} 外与点 x_i 最近的数据点称为第二最邻，记为 x_{i2}。依此类推，点 x_i 的 k-最近邻集合记为 $knn_i=\{x_{i1},x_{i2},\cdots,x_{ik}\}$。以点 x_i 为中心，以它到第 k 近邻的距离为半径，所围成的区域被称为点 x_i 的 k-邻域。

由 k-最近邻的定义可知，密集区域的数据点的 k-邻域较小，而稀疏区域的数据点的 k-邻域较大，因此，最近邻关系所描述的邻域大小可以根据不同数据所处区域密度进行自动缩放[23]。假设各簇中数据点间绝对距离不同，但近邻点分布的结构和相对位置相似，使用 k-邻域能够把各簇内部的点(无论簇是稀疏还是稠密)更紧密地联系起来。

最近邻数 k 是算法中可调节的参数，且 k 的取值决定了每个数据点可考虑的近邻范围，对构造的相似矩阵的质量有较大影响。在各个数据集的数据量和结构不一致的前提下，最近邻数 k 也应根据实际情况取合适的值，而非所有数据集统一取相同值。引入自然近邻的思想，每一个数据集的最近邻数 k 都是由该数据集内部数据点的整体分布情况决定的。

定义 2(自然 k 近邻) 若 X 中任一点 x_i 都至少出现在它的其中一个 k-最近邻的 k-邻域中，当 k 取最小值时，称点 x_i 的 k-最近邻为点 x_i 的自然 k 近邻。点 x_i 的自然 k 近邻集合记为 $nknn_i$，此时点 x_i 的 k-邻域称为自然 k-邻域，记为 nkn_i。

通常自然 k 近邻的搜索可以从一个较小的 k 值开始($k>1$，否则无意义)，k 值连续增加并判断在数据集 X 上是否能够满足自然 k 近邻的定义，若满足则说明找到了该数据集的 k 值，若不满足则继续向上搜索。

Zhang 等引入共享最近邻的定义，若两点的共享最近邻较多，可以认为这两点的联系较紧密。在数值上取共享最近邻的计数来量化对两点相似度的贡献大小，计数更大，则相似度更高。其中，共享最近邻由两点的 ε-邻域交集构成。

定义 3(共享最近邻) 对于数据集 X 中的任意两点 x_i 和 x_j，其自然 k 近邻集合分别为 $nknn_i$ 和 $nknn_j$，两点的共享最近邻是两点自然 k 近邻集合的交集，表示为：

$$snn_{ij}=\{h_{ijn}\mid h_{ijn}\in nknn_i\cap nknn_j,n=1,2,\cdots\} \tag{6}$$

通常情况下，处于同一密集区域且距离相近的数据点往往具有更多的共享最近邻，相近但处于不同密集区域的数据点拥有较少的共享最近邻。共享邻居的数量能够在一定程度上体现出两点联系的紧密程度。然而，使用 ε-邻域作为搜索区域会出现一个问题：若各簇密度不均，则会因为稀疏簇内部点距较大，点间共享邻居较少，而使得簇内部点之间的相似度降低。为了让同一稀疏簇数据点的联系更紧密，我们采用自然 k-邻域而非 ε-邻域。

3.2 自适应尺度参数

在传统谱聚类算法中，尺度参数 σ 是需要人为设定的常值，σ 确定后，两点相似度大小仅与两点间距离有关。无论数据点所在区域分布情况如何，只要距离确定，相似度就确定，这样会对区别不同区域的数据点造成困难。引入自然 k-邻域上的共享最近邻，在一定程度上能够让同簇内部点联系更紧密，但当相邻的簇中数据点分布的密集程度不同时，稀疏簇边界上的点因自然 k-邻域较大，容易与相近的异簇点存在共享最近邻，从而被划分到错误的簇中。

如果要更有效地识别出稀疏簇，就需要让边界上异簇点的联系更松散。如图 3 所示，point 1 与 point 2 分别是两个簇边界上相近的点，point 1 属于 sparse cluster，point 2 属于 dense cluster。分别从 point 1 及 point 2 出发，连接它们的 k 个最近邻点(图中假设 $k=7$)，point 3 和 point 4 是 point 1 和 point 2 的共享最近邻。若在计算 point 1 和 point 2 相似度时赋予 point 3 和 point 4 更低的重要性，降低其对相似度大小的贡献，则有利于将 point 1 划分到正确的簇中。若两点所属区域密度有较大差异，由于密集区域点的自然 k-邻域相对较小，共享最近邻会更靠近两点中所属区域密度更大的一方，且两边密度相差越大，这种偏向越明显。如果能够量化共享最近邻所在位置的“偏向”，并体现在对原数据点对相似度的重

要性上，就能更有效地划分不同区域的边界点。因此，改进相似性度量的思路在于：当两点区域密度相差较大时，表现为共享最近邻的偏向水平较大，为了降低两点相似度，此时该共享最近邻对于相似度的重要性应该较小。若构造能反映重要性大小关系的度量，并作为权重赋给共享最近邻，使相似度与权重同向变动，就能有效利用密度信息动态调整相似度的大小。

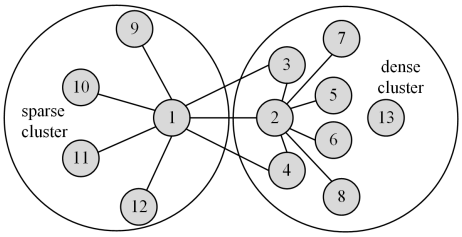


图 3 点 1 和点 2 的最近邻示意图

Fig. 3 Sehematic diagram of the nearest neighbors of point 1 and point 2

在体现数据的分布特征方面，不同的衡量尺度发挥着不同的作用。以欧氏距离为计算方法的间隔尺度可以衡量点间位置的绝对差距，且能表示差距的大小关系，但仅用间隔尺度无法表现出各个共享最近邻对原数据点的重要程度的差异。从图 3 中可以看到，共享最近邻 point 3 到 point 1 和 point 2 的距离不同，在两点的最近邻中的排列次序也不同。可以看到，point 3 到 dense cluster 中 point 2 的距离更小，且在 point 2 的最近邻中排序也更靠前。尝试在搜索共享最近邻时，引入能够描述强弱程度的顺序尺度以及能够体现相对差异大小的比例尺度，目的是根据各点所处区域密度动态增减其共享最近邻的重要性权重，以降低不同密度异簇边界点的相似度。

本文根据以上思想，构造能够随数据点的最近邻分布自主调整大小的自适应尺度参数。

定义 4（相对邻近度） 设 $h_{ij(n)}$ 是点 x_i 和点 x_j 的共享最近邻集合 snn_{ij} 中的点，$h_{ij(n)}$ 必然满足既是 x_i 的自然 k 近邻，又是 x_j 的自然 k 近邻。为了量化点 $h_{ij(n)}$ 对 x_i，x_j 两点相似度的贡献，定义 $h_{ij(n)}$ 到 x_i，x_j 的相对邻近度（Relative Proximity）如式(7)所示：

$$rp_{ij(n)}=\frac{\min(l_{i(n)}\cdot d(x_ih_{ij(n)}),l_{j(n)}\cdot d(x_jh_{ij(n)})}{\max(l_{i(n)}\cdot d(x_ih_{ij(n)}),l_{j(n)}\cdot d(x_jh_{ij(n)})} \tag{7}$$

其中，$l_{i(n)}$ 表示点 $h_{ij(n)}$ 在点 x_i 的自然 k 近邻集合中的顺序排次，属于顺序尺度，若 $h_{ij(n)}$ 是 x_i 的第 p 近邻，则有 $l_{i(n)}=p$。$rp_{ij(n)}$ 属于比例尺度，数值越小，说明 $h_{ij(n)}$ 对两点其中一方的位置偏向越明显，两点属于不同密度区域的可能性越大，相似程度越小。

定义 5（基于相对邻近度的尺度参数） 对于任意两点 x_i 和 x_j，定义它们的尺度参数 γ_{ij} 如式(8)所示：

$$\gamma_{ij}=1+\sum_{h_{ij(n)}\in snn_{ij}}rp_{ij(n)} \tag{8}$$

其中，等式右边第二项是点 x_i 和点 x_j 的共享最近邻集合中的每一点以相对邻近度为重要性权重的加权计数项。

定义 6（改进的相似性度量） 对于任意两点 x_i 和 x_j，基于改进后尺度参数的相似性度量如式(9)所示：

$$w_{ij}=\begin{cases}\exp\left(-\dfrac{d^2x_ix_j}{\gamma_{ij}}\right), & d(x_ix_j)\leqslant\max(nkn_i,nkn_j)\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

新的相似性度量具有如下特点：

1）对应的图连接方式不是全连接。若两点距离较远，彼此都不在对方的自然 k-邻域内，此时相似度为 0。由此，最终得到的相似矩阵 $\boldsymbol{W}$ 是一个稀疏矩阵，降低了后期计算的复杂度。

2）自然 k 近邻搜索使得同数据集内的 k 值相同，数据集间的 k 值不同，自适应的 k-邻域既有利于平衡数据集内部不同密度区域点的联系，又能适应各个数据集不同的分布情况。

3）自适应尺度参数根据数据点最近邻的分布情况，自动调节大小，无需人为设定。引入共享最近邻的概念来反映两点周围区域的密度，构造可根据区域密度灵活调整大小的多尺度参数，弥补了传统的高斯核函数尺度参数取值单一的缺点。多尺度参数利用不同尺度的特性描述共享最近邻到两点的位置关系，以相对邻近度动态缩放共享最近邻的重要性大小。该尺度参数中包含了数据点分布结构等信息，有利于发现真实的簇形态。

4）当两点位于不同簇且所处区域密度不同时，通过尺度参数的自动调节，可适当减小两点的相似度；当两点位于同一稀疏簇时，通过参数自动调节，适当增大两点的相似度，更容易将稀疏区域的点凝聚到一起。

5）任意两点 x_i 和 x_j 中有一方在对方的自然 k-邻域内但两点无共享最近邻点时，两点的相似度可表示为 $w_{ij}=\exp(-d^2x_ix_j)$，此时相似度计算式退化为普通高斯核函数，尺度参数 $\gamma_{ij}=1$。

3.3 RPASC 算法描述

RPASC 算法具体步骤如算法 1 所示。

算法 1 RPASC

输入：数据集 X，聚类数目 c

输出：聚类结果 T

步骤 1 计算数据集 X 中各点之间的相似度，构造实对称的相似矩阵 **W**；

步骤 1.1 对任意两点 x_i 和 x_j，计算它们之间的欧氏距离，得到距离矩阵 **S**；

步骤 1.2 根据定义 1 和定义 2，遍历距离矩阵 **S** 搜索自然 k 近邻，确定数据集 X 的 k 值；

步骤 1.3 对任意点 x_i，得到其自然 k 近邻集合 $nknn_i$ 和自然 k-邻域 nkn_i，将 k 个最近邻到点 x_i 的距离由小到大排序，记录每个最近邻的次序；

步骤 1.4 根据定义 3，在任意两点 x_i 和 x_j 的自然 k-邻域中搜索两点的共享最近邻点 $h_{ij(n)}$，得到共享最近邻集合 snn_{ij}；

步骤 1.5 根据定义 4 中式(7)，对任意两点 x_i 和 x_j，计算其共享最近邻集合中每个点的相对邻近度 $rp_{ij(n)}$；

步骤 1.6 根据定义 5 中式(8)，对任意两点 x_i 和 x_j，计算得到自适应尺度参数 γ_{ij}；

步骤 1.7 根据定义 6 计算点间相似度，构造相似矩阵 **W**；

步骤 2 根据式(2)和式(3)计算非规范化的拉普拉斯矩阵 **L** 和度矩阵 **D**，代入式(4)得到规范化的拉普拉斯矩阵 $\mathbf{L}_{sym}$；

步骤 3 计算矩阵 $\mathbf{L}_{sym}$ 的特征值和相应的特征向量，选取前 c 个最大特征值对应的特征向量构成一个新的矩阵 $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_c]$；

步骤 4 矩阵 **V** 的行向量可以看作 c 维空间中的点 $\mathbf{y}_i$，$i=1,2,\cdots,q$，将 q 个行向量看作 q 个 c 维样本点，使用 K-means 算法聚类，得到 $\mathbf{y}_i$ 的聚类标签 t_i。根据输出结果，若 $t_i=j$，则把原始数据点 x_i 分配到类 T_j 中。

RPASC 算法的主要流程如图 4 所示。

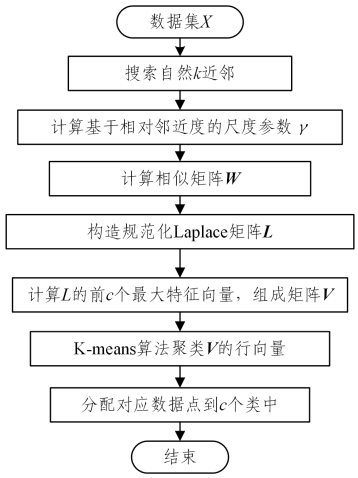


图 4 RPASC 算法的主要流程

Fig. 4 Main process of RPASC algorithm

4 实验结果与分析

为了验证本文算法的有效性，分别在人工数据集和真实数据集上，对 K-means 算法、传统谱聚类算法(SC)以及 5 个改进算法(基于共享最近邻的谱聚类算法(SNN-SC)[15]、基于密度敏感的谱聚类算法(DS-SC)[16]、基于密度系数和共享近邻的谱聚类算法(DSCN-SC)[17]、基于样本间变异系数的谱聚类算法(CV-SC)[18]、基于共享最近邻和自适应密度邻域构建的谱聚类算法(DANSN-SC)[19])和本文基于相对邻近度的自适应谱聚类算法(RPASC)共 8 种算法进行了对比实验。表 1 列出了其中 7 种谱聚类算法构造的相似性度量的特点，并根据不同相似性度量推测其拉普拉斯矩阵的谱可能具有的特性。

本文选取的聚类评价指标有：调整兰德系数(Adjusted Rand Index，ARI)[24]、调整互信息(Adjusted Mutual Information，AMI)[24]以及 Fowlkes-Mallows 指数(FMI)[25]。对于数据集 X，设真实标签集合为 C，聚类得到的标签集合为 T。将在 C 中和在 T 中均属于相同簇的数据点对的数目记为 n_1；只在 C 中属于相同簇的数据点对的数目记为 n_2；只在 T 中属于相同簇的数据点对的数目记为 n_3；在 C 中和在 T 中均不属于相同簇的数据点对的数目记为 n_4。

表 1 各谱聚类算法的特点概述

Table 1 Characteristics overview of spectral clustering algorithms

算法	相似性度量的特点	图拉普拉斯矩阵的谱特性
SC	尺度参数 σ 决定了相似度的范围和衰减速度。σ 值较大时一定范围内相似度波动大；反之则相似度波动小	零特征值的重数和较小的非零特征值能够反映图的连通性，在合理选择尺度参数 σ 的情况下，相似性矩阵通常可以保证图的连通性
SNN-SC	共享最近邻计数加权的尺度参数在高密度区域减少相似度的变化；在低密度区域突出局部相似性	当簇间密度差距较大时，低密度簇的连通性变低，图的割裂可能会比较明显，可能出现较多零特征值
DS-SC	定义局部密度差以调整簇类样本点间相似度，降低了尺度参数的敏感性，新的相似矩阵可能在不同的尺度参数下表现出更稳定的特征	低密度区域的点的相似度有所提高，相比 SNN-SC 算法改善了图的连通性，且特征值分布对尺度参数变化的敏感度下降
DSCN-SC	定义密度系数，并以其均值为阈值将数据点划分为密集区和稀疏区，以不同分区对数据点的尺度参数进行不同的加权计算	密集区的划分可能会形成明显的社区结构，这种结构可能在拉普拉斯矩阵的特征值中表现为一些较小的特征值
CV-SC	定义变异系数，降低噪声点对相似性计算的影响。变异系数可体现数据的离散程度，变异系数大则数据离散程度大，相似度相应降低	图中因噪声而分离的分量减小，图可能变得更连通，零特征值减少，与噪声相关的特征值减少，代数连通度(最小的非零特征值)增大
DANSN-SC	该算法不使用高斯核函数，构造共享邻居的权重表达式，并对表达式计算出的相似度进行归一化，考虑在[0,1]范围内的成对相似度	归一化的相似度使得矩阵的非对角线元素(即图的边权重)不会有极端值，这种平滑性可能使特征值分布更加均匀
RPASC	定义自然 k-邻域，对共享最近邻构造能反映区域密度的加权表达式作为新的尺度参数。新的相似性度量结合了多个尺度关系的特性，能更好地反映数据分布结构	相似矩阵为稀疏矩阵，图的连通性较低，可能表现为多重零特征值和较小的非零特征值

调整兰德系数是兰德系数(RI)的一种调整形式，可以用于评估将样本点分为多个簇的聚类算法。ARI 的计算式如式(10)所示：

$$ARI=\frac{2(n_1\times n_4-n_2\times n_3)}{(n_1+n_2)\times(n_3+n_4)+(n_1+n_3)\times(n_2+n_4)} \quad (10)$$

ARI 的取值范围为[−1,1]，其中值越接近 1 表示聚类结果越准确，值接近 0 表示聚类结果与随机结果相当，值接近 −1 表示聚类结果与真实情况几乎完全不同。

调整互信息是互信息(MI)的一种调整形式，也是一种用于衡量多簇聚类算法性能的指标。与 MI 相比，AMI 更加稳健，能够更好地反映数据分布的吻合程度[26]。AMI 的计算式如下：

$$AMI=\frac{MI-E(MI)}{\max(H(C),H(T))-E(MI)} \quad (11)$$

其中，MI 代表真实标签 C 与聚类结果 T 之间重叠的信息；$H(C)$与 $H(T)$表示对应样本的边缘熵值。AMI 的取值范围是[−1,1]，值越接近 1，聚类结果与真实情况越吻合。

Fowlkes-Mallows 指数(FMI)主要基于数据的真实标签和聚类结果的交集、联合集以及簇内和簇间点对数的比值来评价聚类效果。FMI 的计算式如下：

$$FMI=\frac{n_1}{\sqrt{(n_1+n_2)\times(n_1+n_3)}} \quad (12)$$

FMI 在数值上是准确率与召回率的几何平均数，取值范围为[0,1]，值越接近 1 表示聚类效果越好[27]。

本文的实验环境为 Windows 10 64 位操作系统，K-means 算法、SC 算法、SNN-SC 算法、DS-SC 算法、DSCN-SC 算法、CV-SC 算法、DANSN-SC 算法和 RPASC 算法均使用 Python

3.8 进行编程实验。其中 K-means 算法和几种谱聚类算法构造相似矩阵后的矩阵计算操作由 Python 调用 sklearn 包中函数实现。另外，为了更公平地评价不同聚类算法的聚类效果，进行以下操作：

1)实验中不同算法的参数值默认设置为该算法论文中的推荐取值，未有推荐取值的参数统一设置为相同数值。DSCN-SC 算法 k 值取样本量 4%大小；DS-SC 算法取 $k=6$，但图连接方式为全连接；SNN-SC 算法中的 ε-邻域使每个点至少有 6 个最近邻；SC 算法和 SNN-SC 算法的尺度参数均取 $2\sigma^2=1$(对应本文自适应尺度参数取 $\gamma_{ij}=1$ 时)。

2)各算法中聚类数目 c 均取正确个数。

4.1 人工合成数据集实验

比较 K-means 算法、SC 算法、SNN-SC 算法、DS-SC 算法、DSCN-SC 算法、CV-SC 算法、DANSN-SC 算法和 RPASC 算法在人工合成数据集上的聚类效果。本文选取的 6 个人工合成数据集包含环形簇类、半月形簇类、针叶形簇类、螺旋形簇类等，基本信息如表 2 所列。

表 2 人工合成数据集基本信息

Table 2 Basic information of synthetic datasets

数据集	样本数	维数	簇个数
Ring	500	2	2
Ring-unbalanced	500	2	3
Moon	373	2	2
Leaves	200	2	4
Spiral	312	2	3
Spiral-unbalanced	567	2	2

由表 2 可知，选取的 6 个数据集特征数均为 2，故可以用二维平面图直观地展示各个数据集的聚类结果，如图 5 所示。其中，数据集 Ring-unbalanced，Moon 和 Spiral-unbalanced 中各簇密度差异较大，可用来测试各算法在不平衡数据集上的聚类效果。

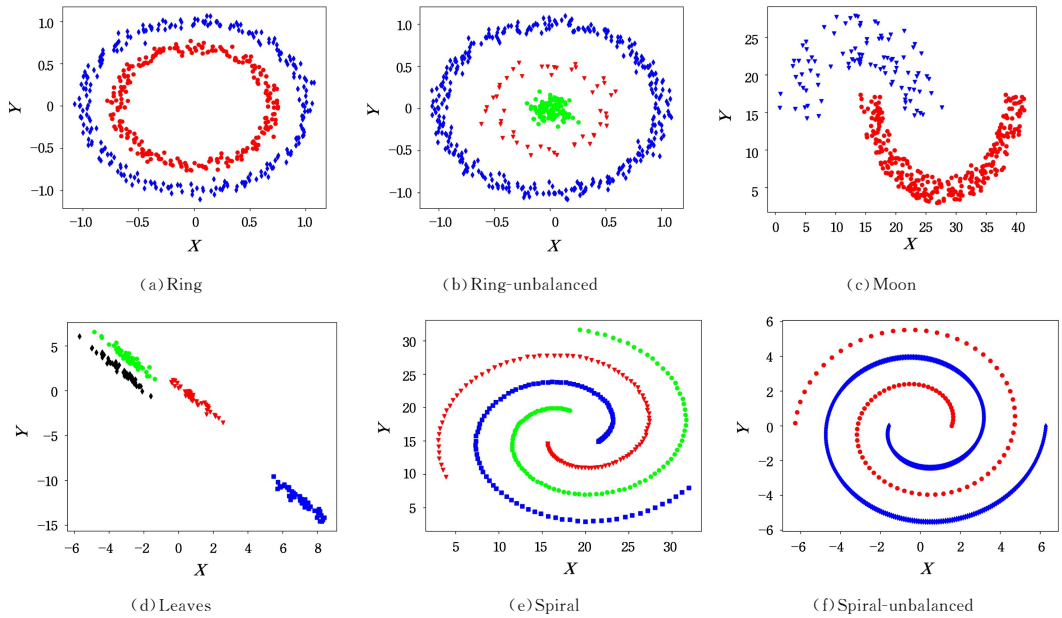
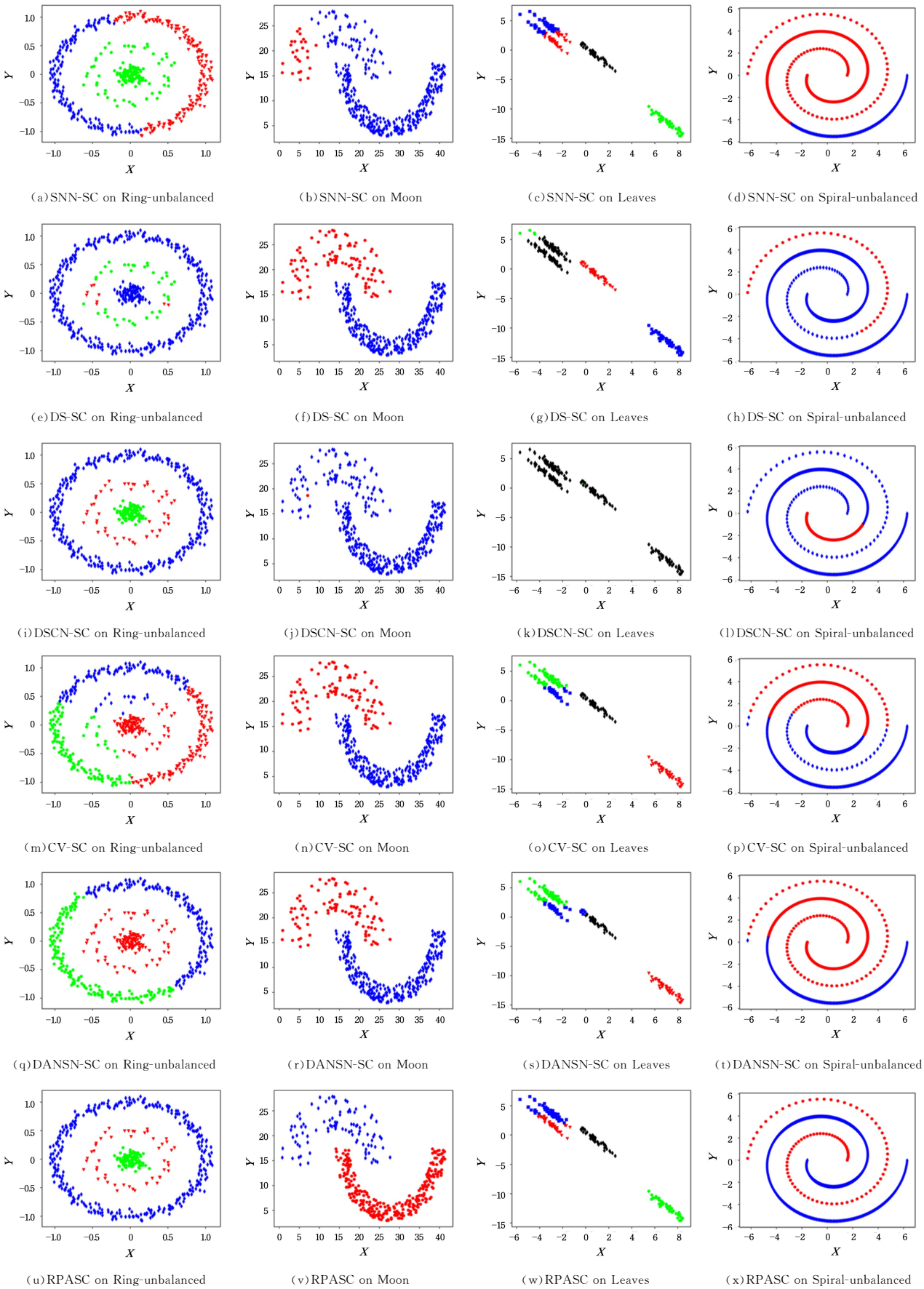


图 5 人工合成数据集的正确聚类结果

Fig. 5 Correct clustering results of synthetic datasets

图 6 为 SNN-SC 算法、DS-SC 算法、DSCN-SC 算法、CV-SC 算法、DANSN-SC 算法和 RPASC 算法在其中 4 个人工合成数据集上的聚类结果。在 Ring-unbalanced 数据集上，DS-SC 算法(图 6(e))将中心簇和最外层簇划分在一起，由于 DS-SC 算法中自适应尺度参数的大小由局部密度差决定，因此不属于同一区域但局部密度大小相近的两点联系更紧密，造成了跨簇的错误划分；CV-SC 算法(图 6(m))错误地将环状簇都识别为团块状簇；SNN-SC 算法(图 6(a))和 DANSN-SC 算法(图 6(q))未能划分开中心的两个不同密度簇，而最外层的簇被错误地拆为两部分；DSCN-SC 算法(图 6(i))仅在两簇交界有一个点错误；RPASC 算法(图 6(u))能做到正确聚类。在 Moon 数据集上，DSCN-SC 算法(图 6(j))错误地将两个不同密度的月牙形簇合并为一个簇，没有识别出簇间差异，考虑到可能是 k 的取值不适合该数据集；SNN-SC 算法(图 6(b))受到簇间距离和区域密度的影响，将稀疏簇中的大部分点划分到稠密簇中；RPASC 算法(图 6(v))、DS-SC 算法(图6(f))、CV-SC 算法(图 6(n))和 DANSN-SC 算法(图 6(r))均能正确聚类。在 Leaves 数据集上，DSCN-SC 算法(图 6(k))仍没有找到各簇的分界，错误地把绝大多数点划分为一个簇；DS-SC 算法(图 6(g))基本将距离较近的两个条形簇识别为一个团形簇；SNN-SC 算法(图 6(c))、CV-SC 算法(图 6(o))和 DANSN-SC 算法(图 6(s))也将两个相近的簇中较多的点做了错误的划分；RPASC 算法(图 6(w))的正确率最高。在 Spiral-unbalanced 数据集上，CV-SC 算法(图 6(p))未能识别出簇的螺旋形态，只简单划分为上下两部分；SNN-SC 算法(图 6(d))、DSCN-SC 算法(图 6(l))和 DANSN-SC 算法(图6(t))只识别出稠密簇的一部分；DS-SC 算法(图 6(h))受到簇的形态和密度差异的影响，将相近的点划分在一起，使得稀疏簇螺旋中心部分的点被划分到稠密簇中；而 RPASC 算法(图 6(x))能做到正确聚类。



(a) SNN-SC on Ring-unbalanced (b) SNN-SC on Moon (c) SNN-SC on Leaves (d) SNN-SC on Spiral-unbalanced

(e) DS-SC on Ring-unbalanced (f) DS-SC on Moon (g) DS-SC on Leaves (h) DS-SC on Spiral-unbalanced

(i) DSCN-SC on Ring-unbalanced (j) DSCN-SC on Moon (k) DSCN-SC on Leaves (l) DSCN-SC on Spiral-unbalanced

(m) CV-SC on Ring-unbalanced (n) CV-SC on Moon (o) CV-SC on Leaves (p) CV-SC on Spiral-unbalanced

(q) DANSN-SC on Ring-unbalanced (r) DANSN-SC on Moon (s) DANSN-SC on Leaves (t) DANSN-SC on Spiral-unbalanced

(u) RPASC on Ring-unbalanced (v) RPASC on Moon (w) RPASC on Leaves (x) RPASC on Spiral-unbalanced

图 6 各个算法在不同人工合成数据集上的聚类结果

Fig. 6 Clustering results of various algorithms on different synthetic datasets

由图 6 可知，RPASC 算法能更好地识别出簇的分布形态，且能更有效地划分稠密区域与稀疏区域。

表3列出了各个算法在6个人工合成数据集上的聚类评价指标，在当前数据集上相对最优的评价指标值均加粗表示。

表3 人工合成数据集上不同谱聚类算法的评价指标比较

Table 3 Comparison of evaluation indexes of different spectral clustering algorithms on synthetic datasets

数据集	算法	ARI	AMI	FMI	数据集	算法	ARI	AMI	FMI
Ring	K-means	−0.0018	−0.0013	0.4981	Leaves	K-means	0.6481	0.7154	0.7350
	SC	−0.0019	−0.0014	0.4980		SC	0.6491	0.8075	0.7618
	SNN-SC	0.9919	0.9811	0.9959		SNN-SC	0.6880	0.7603	0.7648
	DS-SC	0.0001	0.0079	0.7021		DS-SC	0.7049	0.8272	0.8033
	DSCN-SC	0.9919	0.9811	0.9959		DSCN-SC	0.0024	0.0349	0.4807
	CV-SC	−0.0019	−0.0014	0.4980		CV-SC	0.7511	0.8112	0.8144
	DANSN-SC	−0.0018	−0.0013	0.4981		DANSN-SC	0.7082	0.7607	0.7816
	RPASC	**1.0000**	**1.0000**	**1.0000**		RPASC	**0.8474**	**0.8645**	**0.8851**
Ring-unbalanced	K-means	0.0337	0.1644	0.4611	Spiral	K-means	−0.0059	−0.0055	0.3277
	SC	0.0217	0.1435	0.4562		SC	**1.0000**	**1.0000**	**1.0000**
	SNN-SC	0.4438	0.6423	0.6926		SNN-SC	0.0526	0.2168	0.5221
	DS-SC	0.4034	0.5504	0.8079		DS-SC	**1.0000**	**1.0000**	**1.0000**
	DSCN-SC	0.9975	0.9868	0.9988		DSCN-SC	0.0151	0.1523	0.5234
	CV-SC	0.0218	0.1672	0.4589		CV-SC	**1.0000**	**1.0000**	**1.0000**
	DANSN-SC	0.4433	0.6422	0.6922		DANSN-SC	0.0016	0.0015	0.3326
	RPASC	**1.0000**	**1.0000**	**1.0000**		RPASC	**1.0000**	**1.0000**	**1.0000**
Moon	K-means	0.3241	0.3676	0.7005	Spiral-unbalanced	K-means	0.0080	0.0078	0.6033
	SC	**1.0000**	**1.0000**	**1.0000**		SC	0.0537	0.0342	0.6292
	SNN-SC	0.2562	0.8037	0.8037		SNN-SC	0.0813	0.0817	0.6739
	DS-SC	**1.0000**	**1.0000**	**1.0000**		DS-SC	0.4903	0.4060	0.8949
	DSCN-SC	0.0098	0.0070	0.7839		DSCN-SC	0.0139	0.0101	0.8496
	CV-SC	**1.0000**	**1.0000**	**1.0000**		CV-SC	0.0658	0.0343	0.6403
	DANSN-SC	**1.0000**	**1.0000**	**1.0000**		DANSN-SC	0.0671	0.0440	0.6360
	RPASC	**1.0000**	**1.0000**	**1.0000**		RPASC	**1.0000**	**1.0000**	**1.0000**

比较ARI，AMI，FMI这3个指标可以发现，在6个数据集上RPASC算法的3项评价指标均能达到最优，聚类效果优于其他几种算法。在类似Ring-unbalanced和Spiral-unbalanced这样不平衡的数据集或Leaves这样各簇分布比较接近的数据集中，RPASC算法更具优势。

RPASC算法在6个人工合成数据集上的聚类效果较为理想，能够识别出形状不同、密度不均的簇，尤其是在流线形的数据集上效果较好。

4.2 真实数据集实验

进一步比较K-means算法、SC算法、SNN-SC算法、DS-SC算法、DSCN-SC算法、CV-SC算法、DANSN-SC算法和RPASC算法在真实数据集上的聚类效果。本文选取的8个真实数据集Iris[28]，Control[29]，Thyroid[30]，Breast[31]，Optdigits[32]，Musk[33]，Rice[34]和Pendigits[35]均来自UCI数据库，基本信息如表4所列。这些数据集在样本数、维数和簇个数上不尽相同，具有一定的代表性。

表4 真实数据集基本信息

Table 4 Basic information of real datasets

数据集	样本数	维数	簇个数
Iris	150	4	3
Control	600	60	6
Thyroid	215	5	3
Breast	569	30	2
Optdigits	1923	64	10
Musk	2598	166	2
Rice	3810	7	2
Pendigits	4992	16	10

表5列出了各种算法分别在8个真实数据集上的3项聚类评价指标。由表5可知：在Iris，Thyroid，Optdigits和Rice这4个数据集上，RPASC算法的各个指标值均高于其余7种算法，这是由于RPASC算法将自然k-邻域作为最近邻的范围，最近邻点数不受区域密度不均影响，在一定程度上增大了将稀疏区域内部点联系起来的可能；在此基础上，构造自适应的尺度参数代替单一尺度参数，考虑点间最近邻的重合程度和亲疏关系，赋予不同位置的点以不同的参数值，更好地反映了数据的分布信息，使稀疏区域与稠密区域的划分更明显，有利于发现真实的簇形态。在Control，Musk和Pendigits这3个数据集上，RPASC算法均有两项指标值优于其余7种算法。在Breast数据集上，RPASC算法的ARI值最优，其AMI值和FMI值低于DANSN-SC算法，但3项指标的差距都比较小。相较而言，RPASC算法总体聚类效果更好。

相比其他数据集，Musk的维度更高，将其作为高维数据的代表测试各个聚类算法，8种算法的聚类结果都不理想。这样的结果可能有多个原因：首先，谱聚类中构建数据点的相似度矩阵依赖于距离度量，在高维空间中，距离度量的对比能力减弱，相似度矩阵的元素趋于相同，降低了谱分解过程中特征向量对数据真实结构的捕捉能力；再者，高维数据中可能包含大量噪声和冗余特征，使得算法使用更多的计算内存和时间却难以提取出数据的有效结构，影响聚类的准确性；最后，由于DANSN-SC算法和RPASC算法构造的相似矩阵是稀疏矩阵，这样可能会丢失部分有效信息，使得算法在计算拉普拉斯矩阵及其特征向量时，忽略了真实的聚类结构。

为了缓解谱聚类在高维数据上遇到的问题，提高聚类的精度和可靠性，可以考虑对源数据降维，例如使用UMAP方法构建高维数据的流形，然后在低维空间中尽可能保留数据结构，使得相似的数据点在低维空间中仍然尽可能接近。也

可考虑运用专业领域知识选择对聚类任务具有实际意义的特征，以减少噪声和冗余。

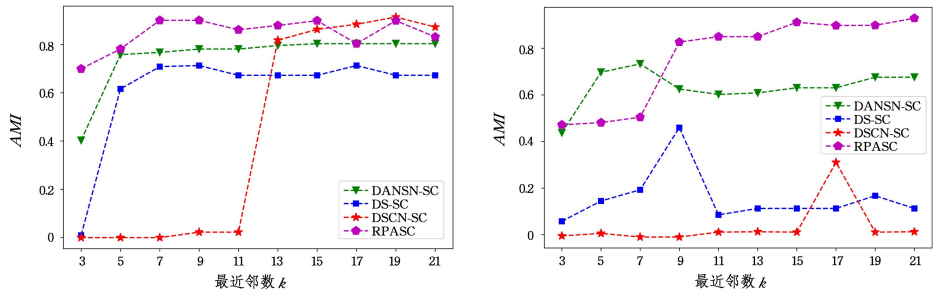
表 5 真实数据集上不同谱聚类算法的评价指标比较

Table 5 Comparison of evaluation indexes of different spectral clustering algorithms on real datasets

数据集	算法	ARI	AMI	FMI	数据集	算法	ARI	AMI	FMI
Iris	K-means	0.7163	0.7386	0.8112	Control	K-means	0.5830	0.7510	0.6803
	SC	0.6230	0.6541	0.7502		SC	0.5563	0.7180	0.6577
	SNN-SC	0.7322	0.7880	0.8237		SNN-SC	0.5050	0.6656	0.6135
	DS-SC	0.5637	0.7125	0.7635		DS-SC	0.5669	0.7268	0.6672
	DSCN-SC	0.0001	0.0001	0.5696		DSCN-SC	**0.6230**	0.7830	0.7026
	CV-SC	0.5910	0.6247	0.7275		CV-SC	0.5653	0.7004	0.6452
	DANSN-SC	0.7591	0.8032	0.8407		DANSN-SC	0.6130	0.7923	0.7079
	RPASC	**0.8856**	**0.8605**	**0.9232**		RPASC	0.6169	**0.8028**	**0.7115**
Thyroid	HK-means	0.6282	0.5908	0.8545	Breast	K-means	0.7301	0.6225	0.8769
	SC	0.3781	0.3667	0.7872		SC	0.4971	0.4511	0.7914
	SNN-SC	0.1352	0.1397	0.7440		SNN-SC	0.0122	0.0217	0.7267
	DS-SC	0.1209	0.0953	0.7422		DS-SC	0.0048	0.0067	0.7281
	DSCN-SC	0.0156	0.0127	0.7300		DSCN-SC	0.0024	0.0017	0.7286
	CV-SC	0.8418	0.7828	0.9309		CV-SC	0.6891	0.5656	0.8547
	DANSN-SC	0.7323	0.6756	0.8892		DANSN-SC	0.7921	**0.7018**	**0.9052**
	RPASC	**0.9063**	**0.8497**	**0.9572**		RPASC	**0.7922**	0.6942	0.9048
Optdigits	K-means	0.6161	0.7388	0.6594	Musk	K-means	−0.0362	**0.0287**	0.6236
	SC	0.4139	0.6947	0.5273		SC	−0.0006	−0.0004	0.8628
	SNN-SC	0.5375	0.7344	0.6062		SNN-SC	−0.0006	−0.0004	0.8628
	DS-SC	0.4112	0.6959	0.5257		DS-SC	−0.0006	−0.0004	0.8628
	DSCN-SC	0.7593	0.8122	0.7844		DSCN-SC	−0.0394	0.0258	0.6271
	CV-SC	0.6851	0.7430	0.7174		CV-SC	−0.0006	−0.0004	0.8628
	DANSN-SC	0.8404	0.8913	0.8571		DANSN-SC	−0.0609	0.0261	0.8103
	RPASC	**0.9092**	**0.9164**	**0.9183**		RPASC	**0.0000**	0.0000	**0.8633**
Rice	K-means	0.6885	0.5755	0.8481	Pendigits	K-means	0.6024	0.6977	0.6440
	SC	0.6780	0.5669	0.8436		SC	0.4846	0.6822	0.5567
	SNN-SC	0.6831	0.5818	0.8475		SNN-SC	0.2958	0.5883	0.4448
	DS-SC	−0.0001	−0.0001	0.7141		DS-SC	0.0003	0.0170	0.3100
	DSCN-SC	0.6876	0.5754	0.8479		DSCN-SC	0.0003	0.0006	0.3163
	CV-SC	0.6850	0.5715	0.8462		CV-SC	0.5275	0.6495	0.5761
	DANSN-SC	0.6893	0.5792	0.8492		DANSN-SC	0.5712	**0.7889**	0.6468
	RPASC	**0.6936**	**0.5838**	**0.8513**		RPASC	**0.6059**	0.7647	**0.6565**

4.3 参数敏感性分析

最近邻数 k 的取值决定了每个数据点可考虑的近邻范围，对构造的相似矩阵的质量有较大影响。图 7 展示了不使用自然 k 近邻搜索时 RPASC 算法与 DANSN-SC 算法、DS-SC 算法和 DSCN-SC 算法在 Iris 和 Thyroid 数据集上取不同 k 值（k 的范围是[3,21]）的聚类指标 AMI。从图 7 中的结果来看，RPASC 算法与 DANSN-SC 算法的聚类结果更稳健，而 DS-SC 算法与 DSCN-SC 算法的聚类结果波动较大，稳定性较差。



(a) Iris (b) Thyroid

图 7 不同算法取不同 k 值时的 AMI

Fig. 7 AMI for various algorithms with different k values

在各个数据集的数据量和结构不一致的前提下，自然 k 近邻能够根据数据集内部数据点的整体分布情况取相对合适的 k 值。从图 7 可以看到，在[3,21]范围内人工调试 k 的取值，RPASC 算法在 Iris 和 Thyroid 数据集上能达到比较好的 AMI 指标结果，都在 0.8 以上。从表 5 可知，RPASC 算法通过自然 k 近邻搜索得到 k 值，在 Iris 和 Thyroid 上的 AMI 指标结果分别为 0.8605 和 0.8497，可见自然 k 近邻搜索方法得到的 k 值能够较好地适应实验数据集。

自然 k 近邻搜索可以设置初始搜索值，一般来说，初始 k 值不宜太小，避免过早停止搜索。k 值过小会使图的连通性极低，研究意义不大。若已知数据集分布较均匀，初始值可设定适中（例如取 $k\geqslant 6$）；若已知数据集存在类别不平衡，或存在噪声、异常值的情况，则初始值也可设定较大（例如取 $k\geqslant 10$）；若已知数据维度较高，这时距离度量的可靠性下降，可能需要更大的 k 来获得更稳定的结果，这种情况可以选择更大的初始值（例如取 $k\geqslant 20$）以减少迭代搜索次数。

4.4 RPASC 算法复杂度分析

表 6 列出了各算法在 Control 和 Musk 两个数据集上运

行的内存和时间信息，指标最大值被加粗显示。Usage 表示获取追踪内存块的当前大小，Peak 表示获取追踪内存块的分配峰值大小，Time 表示运行时间。从表 6 可以看出，相比其他几种算法，RPASC 算法与 DSCN-SC 算法在最终占用和分配过程中需要的内存资源都是比较大的，运行时间也比较长。

表 6 各算法在数据集上的占用内存和运行时间

Table 6 Memory usage and running time of each algorithm on datasets

数据集	算法	Usage/MB	Peak/MB	Time/s
Control	K-means	0.0170	0.6454	0.0887
	SC	0.0194	12.2447	8.6967
	SNN-SC	0.0188	12.2446	157.3546
	DS-SC	0.0188	12.2446	11.7255
	DSCN-SC	**0.0402**	12.2446	272.0277
	CV-SC	0.0210	12.2446	7.5938
	DANSN-SC	0.0156	12.2446	27.6964
	RPASC	0.0228	**15.5938**	**374.7781**
Musk	K-means	0.0233	6.9714	0.0947
	SC	0.0240	229.4912	8349.3285
	SNN-SC	0.0244	229.4911	30058.5801
	DS-SC	0.0219	229.4911	3755.2656
	DSCN-SC	0.0267	229.4911	**42585.6814**
	CV-SC	0.0238	229.4911	2939.2590
	DANSN-SC	0.0216	229.4911	583.5231
	RPASC	**0.0319**	**289.7354**	38722.7780

对 RPASC 算法进行时间复杂度分析，其中 N 代表样本数，k 代表近邻数，c 代表簇的数目，t 代表二阶段 K-means 算法迭代的次数。

1)计算相似矩阵：计算欧氏距离的时间复杂度为 $O(N^2)$，搜索自然 k 近邻和共享最近邻以及构造尺度参数的总体时间复杂度为 $O(N^3)$，计算相似矩阵 $\boldsymbol{W}$ 的整体时间复杂度为 $O(N^3)$；

2)计算规范化 Laplacian 矩阵，时间复杂度为 $O(N^2)$；

3)计算 Laplacian 矩阵的特征分解，时间复杂度为 $O(N^3)$；

4)K-means 算法聚类特征向量，时间复杂度为 $O(c\times t\times N)$。

各个算法的区别主要在于相似矩阵的构造，在这部分，SC 算法、DS-SC 算法和 CV-SC 算法的时间复杂度为 $O(N^2)$，SNN-SC 算法的时间复杂度为 $O(N^2+k^2\times N)$，DANSN-SC 算法的时间复杂度为 $O(k^2\times N^2)$，RPASC 算法和 DSCN-SC 算法的时间复杂度则达到 $O(N^3)$。图 8 给出了各算法在 Optdigits 数据集中取不同样本数量时的运行时间，从图中可以看到，RPASC 算法和 DSCN-SC 算法的运行时间随数据量上升而增长的速率明显快于其他几种算法，这与这两种方法计算相似矩阵的时间复杂度更高相符。在 Optdigits 数据集上，RPASC 算法相较于 DSCN-SC 算法的耗时更长，总体时间成本最高。

综上，RPASC 算法由于增加了搜索自然 k 近邻的过程以及融合了多个尺度度量的计算方式，使得计算步骤增加，过程也更加复杂。该算法对时间和内存的需求大，需要进一步研究改进，降低其计算成本。

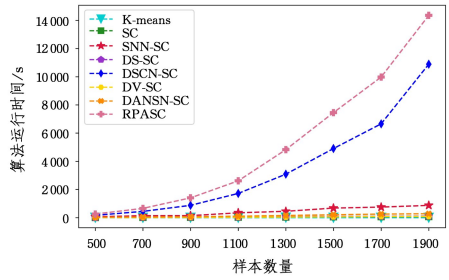


图 8 各个算法在不同样本数量上的聚类时间

Fig. 8 Clustering time of each algorithm on a different number of samples

结束语 针对传统谱聚类算法中尺度参数的选取问题，本文提出了具有自适应尺度参数的 RPASC 算法。实验结果表明，该方法弥补了传统谱聚类算法对参数选取较敏感这一不足，并且能够更好地利用样本信息，得到更高质量的相似矩阵，提高了算法的聚类精度。但是，本文算法由于增加了搜索自然近邻和计算多尺度度量的操作，复杂度较高，计算量较大，在具有大样本量的数据集上计算的时间成本较高；另一方面，在高维数据中，特征选择更加困难，算法的计算复杂度随着维度的增加而急剧上升，这些原因都让谱聚类在高维数据中的应用难度增大，本文算法以及相比较的多种改进谱聚类算法在高维度数据上的聚类效果均不佳。因此，下一步研究工作是探索如何平衡算法的精度和复杂度，例如考虑使用 Nystrom 矩阵近似方法降低谱分解的计算消耗，使其能更高效地应用于大数据集；思考能否使用降维方法或学习其他聚类方法的优势，将改进的谱聚类算法推广到更高维情形。

参 考 文 献

[1] ZHANG Y L, ZHOU Y J. Review of clustering algorithms [J]. Journal of Computer Applications, 2019, 39(7): 1869-1882.

[2] SHI J, MALIK J. Normalized Cuts and Image Segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[3] LI P, LIU L J, HUANG Y D. Landmark-based Spectral Clustering by Joint Spectral Embedding and Spectral Rotation [J]. Computer Science, 2021, 48(S1): 220-225.

[4] XU X, ZHANG H, YANG C M, et al. Fair Method for Spectral Clustering to Improve Intra-cluster Fairness [J]. Computer Science, 2023, 50(2): 158-165.

[5] LI L L. A survey of spectral clustering algorithms and their applications[J]. Software Guide, 2016, 15(7): 54-56.

[6] LIAO L C, JIANG X H, ZOU F M, et al. A Spectral Clustering Method for Big Trajectory Data Mining with Latent Semantic Correlation[J]. Acta Automatica Sinica, 2015, 43(5): 956-964.

[7] HE L, LI Y, ZHANG X, et al. Incremental spectral clustering via fastfood features and its application to stream image segmentation[J]. Symmetry, 2018, 10(7): 272.

[8] ALSHAMMARI M, TAKATSUKA M. Approximate spectral clustering with eigenvector selection and self-tuned k[J]. Pattern Recognition Letters, 2019, 122: 31-37.

[9] XIA K J, GU X Q, ZHANG Y D. Oriented groupingconstrained spectral clustering for medical imaging segmentation[J]. Multi-

media Systems，2020，26(1)：27-36.
[10] XU D H，LI C，CHEN T，et al. A novel low rank spectral clustering method for face identification[J]. Recent Patents on Engineering，2019，13(4)：387-394.
[11] WANG L，BO L F，JIAO L C. Density-Sensitive Semi-Supervised Spectral Clustering[J]. Journal of Software，2007，18(10)：2412-2422.
[12] TAO X M，WANG R T，CHANG R，et al. Low Density Separation Density Sensitive Distance-based Spectral Clustering Algorithm[J]. Acta Automatica Sinica，2020，46(7)：1479-1495.
[13] MANOR L Z，PERONA P. Self-Tuning Spectral Clustering [C]// Proceeding of NIPS. 2005：1601-1608.
[14] KONG W Z，SUN C S H，ZHANG J H，et al. Spectral clustering based onneighboringadaptive local scale[J]. Journal of Image and Graphics，2012，17(4)：523-529.
[15] ZHANG X，LI J，YU H. Local density adaptive similarity measurement for spectral clustering[J]. Pattern Recognition Letters，2011，32(2)：352-358.
[16] ZHAO X Q，LIU X L. Improved adaptive spectral clustering algorithm based on density sensitivity[J]. Journal of Lanzhou University of Technology，2018，44(6)：102-106.
[17] ZHANG T，GE H W. Spectral Clustering Based on Density Coefficient and Shared Nearest Neighbors[J]. Journal of Chinese Computer Systems，2017，38(8)：1829-1833.
[18] ZHAO Y L，CHE W G，JIN R Z. A self-adaptive spectral clustering algorithm based on an improved coefficient of variation between samples[J]. Journal of Lanzhou University (Natural Sciences)，2022，58(6)：812-818.
[19] GE J W，YANG G X. Spectral Clustering Algorithm for Density Adaptive Neighborhood Based on Shared Nearest Neighbors [J]. Computer Engineering，2021，47(8)：116-123.
[20] BAI L，ZHAO X，KONG Y T，et al. Survey of Spectral Clustering Algorithms[J]. Computer Engineering and Applications，2021，57(14)：15-26.
[21] REBAGLIATI N，VERRI A. Spectral clustering with more than K eigenvectors[J]. Neurocomputing，2011，74(9)：1391-1401.
[22] STEVENS S S. Mathematics，measurement，and psychophysics [M]// Handbook of Experimental Psychology. London：Wiley，1951：1-49.
[23] ZHANG L P，ZHAO J Q，LI S，et al. Research on Methods of Construction of Voronoi Diagram and Nearest Neighbor Query in Constrained Regions[J]. Computer Science，2014，41(9)：220-224.
[24] VINH N X，EPPS J，BAILEY J. Information theoretic measures for clusterings comparison：Variants，properties，normalization and correction for chance[J]. Journal of Machine Learning Research，2010，11：2837-2854.
[25] FOWLKES E B，MALLOWS C L. A method for comparing two hierarchical clusterings[J]. Journal of the American Statistical Association，1983，78(383)：553-569.
[26] YIN S Z，WANG T，CHEN Q C，et al. A Class of Classification Algorithm for Binary Protocol Based on Adjusting Mutual Information[J]. Ordnance Industry Automation，2020，39(6)：37-41.
[27] WEI Y，ZHANG Z J，HE K L，et al. Density Peak Clustering Algorithm Based on Relative Density[J]. Computer Engineering，2023，49(6)：53-61.
[28] FISHER R A. The UCI Machine Learning Repository[EB/OL]. [1988-06-30]. https://archive.ics.uci.edu/dataset/53/iris.
[29] ALCOCK R. The UCI Machine Learning Repository[EB/OL]. [1999-06-07]. https://archive.ics.uci.edu/dataset/139/synthetic+control+chart+time+series.
[30] ROSS Q. The UCI Machine Learning Repository [EB/OL]. [1986-12-31]. https://archive.ics.uci.edu/dataset/102/thyroid+disease.
[31] STREET W，WOLBERG W，MANGASARIAN O. The UCI Machine Learning Repository[EB/OL]. [1995-10-31]. https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.
[32] ALPAYDIN E，KAYNAK C. The UCI Machine Learning Repository[EB/OL]. [1998-06-30]. https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits.
[33] CHAPMAN D，JAIN A. The UCI Machine Learning Repository [EB/OL]. [1994-09-11]. https://archive.ics.uci.edu/dataset/75/musk+version+2.
[34] MARKELLE K，RACHEL L，KOLBY N. The UCI Machine Learning Repository[EB/OL]. [2019-10-05]. https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik.
[35] ALPAYDIN E，ALIMOGLU F. The UCI Machine Learning Repository[EB/OL]. [1998-06-30]. https://archive.ics.uci.edu/dataset/81/pen+based+recognition+of+handwritten+digits.

YUAN Zefei，born in 1999，postgraduate. Her main research interests include data mining and machine learning.

ZHANG Zhengjun，born in 1965，Ph.D，associate professor. His main research interests include data mining，graphics technology and image processing.

（责任编辑：何杨）