



基于共享最近邻的密度自适应邻域谱聚类算法

葛君伟, 杨广欣

(重庆邮电大学 计算机科学与技术学院, 重庆 400065)

摘要:在谱聚类算法没有先验信息的情况下,对于具有复杂形状和不同密度变化的数据集很难构建合适的相似图,且基于欧氏距离的高斯核函数的相似性度量忽略了全局一致性。针对该问题,提出一种基于共享最近邻的密度自适应邻域谱聚类算法(SC-DANSN)。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。实验结果表明,SC-DANSN算法相比K-means算法和基于K最近邻的谱聚类算法(SC-KNN)具有更高的聚类精度,同时相比SC-KNN算法对参数的选取敏感性更低。

关键词:谱聚类;相似性矩阵;密度自适应邻域;共享最近邻;K最近邻

开放科学(资源服务)标志码(OSID):



中文引用格式:葛君伟,杨广欣.基于共享最近邻的密度自适应邻域谱聚类算法[J].计算机工程,2021,47(8):116-123.

英文引用格式: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.

Spectral Clustering Algorithm for Density Adaptive Neighborhood Based on Shared Nearest Neighbors

GE Junwei, YANG Guangxin

(College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

[Abstract] Without prior information, it is difficult for spectral clustering algorithms to build appropriate similarity graphs for datasets with complex shapes and different densities. At the same time, the similarity measure of Gaussian kernel functions based on Euclidean distance ignores global consistency. To address the problem, a spectral clustering algorithm (SC-DANSN) for density adaptive neighborhood based on shared nearest neighbors is proposed. An undirected graph is constructed by using a parameter-free density adaptive neighborhood construction method, and shared nearest neighbors are used to measure the similarity between samples. This measurement eliminates the influence of parameters on similarity graph construction, as it reflects both global consistency and local consistency. The experimental results show that the SC-DANSN algorithm has a higher clustering accuracy than the K-means algorithm and Spectral Clustering based on K Nearest Neighbor (SC-KNN). At the same time, SC-DANSN is less sensitive to the selection of parameters than SC-KNN.

[Key words] Spectral Clustering (SC); similarity matrix; density adaptive neighborhood; shared nearest neighbor; K Nearest Neighbor(KNN)

DOI: 10.19678/j.issn.1000-3428.0058893

0 概述

聚类是将数据按照一定的相似性度量规则分为多个不同的类别,将差异小的样本尽可能聚为同一类,将差异大的样本聚为不同的类。聚类算法在处理数据之前无需指定数据集的标签信息,属于无监

督的机器学习领域,被广泛应用在图像处理和模式识别等领域^[1]。根据算法特点大致分为基于划分的聚类算法(如K-means算法)、基于层次的聚类算法(如BIRCHIS算法)、基于密度的聚类算法(如DBSCAN算法)、基于网格的聚类算法(如STING算法等^[2])。对于凸数据集,许多聚类算法都能获得较

基金项目:重庆市重点产业共性关键技术创新重大主题专项(cstc2017zdcy-zdxx0046);重庆市基础与前沿研究计划项目(cstc2017jcyjA0755)。

作者简介:葛君伟(1961—),男,教授、博士,主研方向为大数据处理;杨广欣,硕士研究生。

收稿日期:2020-07-09 **修回日期:**2020-09-02 **E-mail:** ygx16@icloud.com

好的聚类效果,但是对于非凸数据集则难以识别复杂的数据集从而无法很好的聚类。由于谱聚类算法在非凸数据集上有较好的聚类效果,近年谱聚类算法的研究迅猛发展^[3-4]。谱聚类算法将数据集中的每个点视为图的顶点,并将任意两点之间的相似性视为连接两个顶点的边的权重。根据图划分方法,将图分为几个不连续的子图,子图中包含的点集则是聚类后生成的簇^[5]。

尽管谱聚类算法在实践中取得了良好的性能,但作为一种新的聚类方法仍处于发展阶段,有许多问题需要进一步研究。谱聚类算法的聚类性能主要由相似性矩阵的构造决定。在现有相似性矩阵的研究中,大多数研究都是基于欧几里得距离、余弦相似度或高斯核函数来评估数据点之间的相似度。文献[6]提出一种基于数据本身蕴含信息的集成度量学习方法的谱聚类算法。文献[7]通过将数据集样本表示为稀疏的线性组合,通过构造最优化问题的模型,使该方法能较好地反映数据空间的局部一致性,从而提升系统的鲁棒性。文献[8]提出一种自适应谱聚类技术,用于处理多尺度数据集,它没有选择全局参数 σ ,而是根据每个点的邻域信息计算自适应参数。文献[9]通过局部密度获得数据中隐含的簇结构特征,结合自调整高斯核函数,提出一种基于共享邻域自适应相似度的谱聚类算法。文献[10]提出流形结构数据点之间的相似度计算方法,以提高算法的聚类性能。文献[11]致力于找到核参数 σ 的最佳值以改进谱聚类算法,但最佳值是一个全局值,不适用具有密度的簇数据集。文献[12]提出一种翘曲模型,用于将数据映射到新的空间,以更准确进行相似性度量。文献[13]使用数据点之间的局部密度来缩放参数,具有放大集群内相似性的作用。文献[14]基于数据点之间的最大流量来测量相似度,以满足谱聚类算法中使用相似度度量的要求。文献[15]将相对密度敏感项引入相似性度量,在满足全局和局部一致性的同时,通过相对密度敏感项能有效规避噪声点对样本空间的干扰,尤其是对于“桥”噪声。文献[16]通过截断核范数的低秩张量分解方法。计算各视角的样本相似度矩阵和转移概率矩阵,从而解决谱聚类算法的多视角问题。

对于构造谱聚类算法中相似性矩阵,目前研究大多采用K最近邻(K-Nearest Neighbor, KNN)图构

建相似图,并基于密度敏感项改进的欧几里得距离相似性度量来评价样本点之间的相似性。然而基于KNN图构建的相似图通常会引入冗余连接,并且易受噪声影响,噪声样本点使聚类质心不准确。这两个因素会降低谱聚类性能使其出现错误的聚类结果^[17],需根据经验确定K最近邻的值,使最终聚类效果具有不确定性。基于欧几里得距离的相似性度量也存在着无法揭示某些复杂数据集的真实簇的问题,尤其是不能很好地分离数据集。因不同群集中某些数据点相距不远,噪声的存在也会使数据集无法很好地分离。由于许多实际数据集没有很好地分离,很难找到正确的聚类,因此提出一种对未分离的数据集具有鲁棒性的聚类算法是十分必要。

本文以相似性矩阵构造为切入点,提出一种基于共享邻域的密度自适应邻域谱聚类算法(SC-DANSN),构建无需指定初始参数的相似图,并将共享最近邻作为样本点之间的相似性度量,以减少噪声信息在分离数据集上的影响,并解决谱聚类算法相似性度量无法准确反映数据空间结构的问题。

1 谱聚类

谱聚类是一种基于图论的聚类方法。通过对数据集相似矩阵进行谱分析得到较好的聚类结果。谱聚类的概念起源于谱划分,谱划分将数据聚类转换为无向图的多向划分问题^[18]。谱聚类算法将数据点定义为无向图 $G=(V,E)$ 的顶点,其中 V 是顶点集, E 是顶点之间的边集。建立图上各顶点之间的相似度矩阵 S ,其元素 S_{ij} 可被视为连接第 i 个数据点和第 j 个数据点的边上的权重。相似度矩阵的元素 S_{ij} 由高斯核函数表示:

$$S_{ij} = \begin{cases} \exp(-d^2(p_i, p_j)/(2\sigma^2)), & i \neq j \\ 0, & i = j \end{cases} \quad (1)$$

其中, P_i 和 P_j 分别表示第 i 个数据点和第 j 个数据点,而 $d^2(P_i, P_j)$ 表示 P_i 和 P_j 之间的欧几里得距离, σ 是确定数据点之间相似度的比例参数。NJW算法使用归一化相似度矩阵作为图拉普拉斯矩阵,并通过考虑对应于最大特征值的特征向量,基于归一化割准则优化分区问题,谱聚类算法描述如下:

算法1 谱聚类

输入 数据集 D ,聚类数目 K

输出 K 个簇的集合 $C=\{C_1, C_2, \dots, C_K\}$

步骤1 通过KNN邻域或 ϵ 邻域生成无向图,基

于式(1)构建相似性矩阵 S , 生成对应的度矩阵 B , 其中 $B_{ij} = \sum_{j=1} S_{ij}, i, j = 1, 2, \dots, n$ 。

步骤2 生成对应规范化的拉普拉斯矩阵 L 。

步骤3 计算拉普拉斯矩阵 L 对应的特征向量, 并选取前 K 个最大特征值对应的特征向量构建矩阵 X 。

步骤4 用 K-means 算法对 Y 进行聚类, 输出 K 个簇的集合 $C = \{C_1, C_2, \dots, C_K\}$ 。

2 基于共享邻的密度自适应邻域谱聚类

2.1 密度自适应邻域构建算法

谱聚类算法主要是构建相似性矩阵, 在此之前, 先构建能反映样本空间的无向图。主要方法是基于 KNN 图或 ε 邻域图, 其缺点是对参数的选取敏感, 不能较好地反映数据空间。

聚类是基于相似性度量形成点组, 因此同组内的点之间相似度很高, 而来自不同群体点的相似度很低。邻居包括本地相似的点, 因此数据点及其邻居应位于同一簇中。

在 2 个数据集上使用 KNN 和 ε 邻域算法构建的邻域如图 1 所示。根据欧氏距离来衡量相似度, 图 1(a) 中有 2 个簇。当 $K \leq 3$ 时, 簇 2 中点 i 的 KNN 邻域包括来自同一簇的点, 即点 j, m 和 n ; 当 $K \geq 4$ 时, 簇 1 中点 p 包含在点 i 的邻域中。因此, KNN 无法提取点 i 和 o 之间超过点 m 的间接连接, 并导致混合邻域。图 1(b) 中的数据集有 3 个簇, 其中 1 个簇具有内部密度变化。对于固定半径 ε , 点 s 的邻域包括来自同一簇簇 1 的点, 但是点 r 的邻域为空, 即使它不是离群值也是如此。当 ε 值增大时, 点 r 不再是异常值, 但点 s 的邻域也增大, 并引起了簇 2 和簇 3 的邻域混合^[19]。从图 1 可以看出, 聚类问题中参数对邻域方法的敏感性。

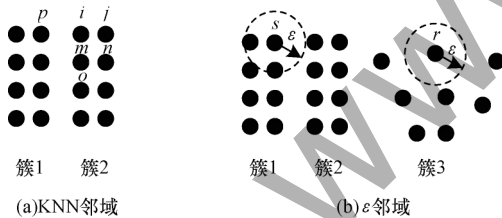


图 1 聚类问题示例邻域的构建

Fig.1 Construction of example neighborhoods for clustering problem

本文采用一种密度自适应的邻域构造(Neighborhood Construction, NC)算法克服 KNN 和 ε 邻域方法的局限性。算法的流程如图 2 所示。

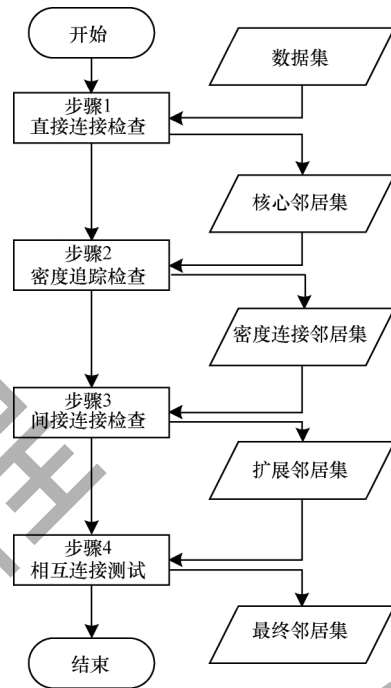


图 2 密度自适应邻域构建流程

Fig.2 Construction process of density adaptive neighborhood

在构建密度自适应邻域算法之前, 定义以下概念:

D : 代表一个数据集;

i, j, p : 代表数据点;

d_{ij} : 代表点 i 和 j 之间点欧氏距离;

CC_i : 代表点 i 的核心近邻点;

BC_i : 代表点 i 的密度连接近邻点;

PC_i : 代表点 i 的扩展近邻点;

CS_i : 代表点 i 的最终近邻点;

$B(i, j, d_{ij})$: 以 i 和 j 为圆上两点, d_{ij} 为直径, 作一个圆, 圆内点的集合是 $B(i, j, d_{ij})$;

直接连接: 当且仅当 $B(i, j, d_{ij}) \cap D = \emptyset$ 时, 称点 i 和 j 为直接连接, 即 i 和 j 之间的集合 $B(i, j, d_{ij})$ 内不存在任何点;

间接连接: 如果 $B(i, j, d_{ij}) \cap D \neq \emptyset$ 时, 称点 i 和 j 为间接连接, 即 i 和 j 之间的集合 $B(i, j, d_{ij})$ 内至少存在一个点;

$Density_{ij}$: 代表点 i 和 j 之间的密度, 该值定义为集合 $B(i, j, d_{ij}) \cap D$ 内点的个数。

NC 算法描述如下:

算法 2 密度自适应邻域构建算法

输入 数据集 D

输出 密度自适应最终邻居集合 CS_i

步骤 1 列出 D 中除点 i 以外的所有点, 按其到点 i 的距离由近到远排列, 并形成有序集 T_i 。

步骤 2 初始化 $CC_i = \emptyset, j = 0$ 遍历有序集合 T_i ,

计算 $\text{Density}_{i, T_i[j]}$, 对于 $\text{Density}_{i, T_i[j]} = 0$, 更新 $CC_i = CC_i \cup \{T_i[j]\}$, $j = j + 1$, 直到 $\text{Density}_{i, T_i[j]} \neq 0$, 结束遍历, 并设置 $\text{indirect } i = j$ 。

步骤3 初始化 $BC_i = CC_i$, $j = \text{indirect } i$ 遍历有序集合 T_i , 计算 $\text{Density}_{i, T_i[j]}$, 对于 $\text{Density}_{i, T_i[j]} - \text{Density}_{i, T_i[j-1]} \geq 0$, 更新 $BC_i = BC_i \cup \{T_i[j]\}$, $j = j + 1$, 直到 $\text{Density}_{i, T_i[j]} - \text{Density}_{i, T_i[j-1]} < 0$, 结束遍历, 并设置 $\text{break } i = j - 1$ 。

步骤4 初始化 $PC_i = BC_i$, $j = \text{break } i$, 遍历有序集合 T_i , 计算 $\text{Density}_{i, T_i[j]}$, 对于 $\text{Density}_{i, T_i[j]} - \text{Density}_{i, T_i[j-1]} \geq 0$, 更新 $PC_i = PC_i \cup \{T_i[j]\}$, $j = j + 1$, 直到 $BC_i \cup BC_{T_i[j]} = \emptyset$, 结束遍历, 并设置 $CS_i = PC_i$ 。

步骤5 初始化 $j = 0$, 遍历有序集合 T_i , 对于 $CC_i \cup CS_{CS[j]} = \emptyset$, 从集合 CS_i 中移除 $CS_i[j]$, 更新 $j = j + 1$, 直到集合 CS_i 不再发生变化, 算法结束, 输出最终邻居集 CS_i 。

NC算法最终输出的密度自适应邻居集可以构造密度自适应的无向图 $G = (V, E)$, 本文把图 G 记为 DAN (Density Adaptive Neighborhood)。构造相似矩阵还需要对无向图 DAN 进行加权, 权重样本点之间的相似性度量。

2.2 基于共享最近邻的相似性度量

在构建局部自适应邻域后可以生成无向图。为实现谱聚类算法应先对该无向图加权, 权重即为数据点之间的相似度。目前谱聚类算法广泛使用基于欧氏距离的相似性度量, 这种相似性度量有时难以反映数据点之间的真实相似程度^[18]。因此考虑一种基于共享最近邻的新相似性度量, 并将其与局部自适应邻域构建相结合, 从而形成一种改进的谱聚类算法。本文基于局部自适应邻域图中共享的最近邻来测量数据点之间的相似度。

设 N_i 表示局部自适应邻域图中数据点 X_i 的邻居。 X_i 与 X_j 之间共享邻居的集合为 $N_i \cap N_j$, 通过集合 $N_i \cap N_j$ 来测量成对相似度 S_{ij} 。通常 N_i 不包括 X_i , 因此 $N_i \cap N_j$ 不包括两个测量点 X_i 和 X_j 。 $N_i \cap N_j$ 是否包含 X_i 和 X_j 取决于 X_i 和 X_j 的关系, 并影响相似性度量。在图 3(a) 和图 3(b) 中两种不同情况下显示点 1 和点 2 的邻居。在图 3(b) 中, 点 1 和点 2 是彼此的邻居, 在图 3(a) 中则不是。如果不考虑点 1 和点 2 的关系, 在两种情况下点 1 和点 2 的共享邻居集相同。因在图 3(b) 中点 1 和点 2 是彼此的邻居, 图 3(b) 中的点 1 和点 2 的相似度高于图 3(a)。

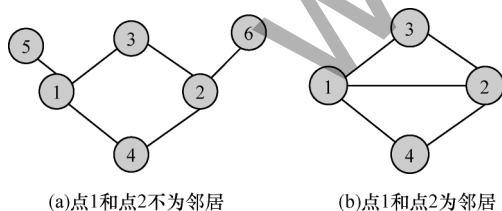


图3 点1和点2的共享邻居示意图

Fig.3 Schematic diagram of shared neighbors of point 1 and point 2

通过考虑两个测量点之间的关系来重新定义图中两个点的共享邻居集合。 N_i 是 X_i 的邻居集合, 并且不包括 X_i 。 $N_i \cap N_j$ 是点 X_i 和 X_j 之间共享邻居的集合, 其重新定义为:

$$N_i \cap N_j = \begin{cases} N_i \cap N_j \cup \{x'_{ij}\}, & X_i \text{ 和 } X_j \text{ 互为邻居} \\ N_i \cap N_j, & \text{其他} \end{cases} \quad (2)$$

其中, x'_{ij} 是一个虚拟数据点, 表示 X_i 是 X_j 的邻居, 并同时表示 X_j 是 X_i 的邻居。如果不考虑其他共享邻居, 则 X_i 和 X_j 具有一个共享邻居。

根据式 (2) 定义共享邻居的集合来测量成对相似性。许多基于共享邻居的聚类方法都将成对相似性视为共享邻居数量的函数。如果 2 个数据点具有更多共享的邻居, 则它们具有较高的成对相似性。但仅考虑共享最近邻居的数量可能会导致测量结果错误。在 2 种不同情况下点 1 和点 2 的 3 个邻居如图 4 所示。点 1 和点 2 在图 4(a) 中具有 2 个共享邻居, 而在图 4(b) 中它们具有 1 个共享邻居。如果仅考虑共享邻居数量, 图 4(a) 中的点 1 和点 2 的成对相似度高于图 4(b) 中的点。图 4(b) 中点 1 和点 2 非常接近, 它们的成对相似度应高于图 4(a)。因此仅考虑共享邻居的数量可能会忽略数据点的紧密度。

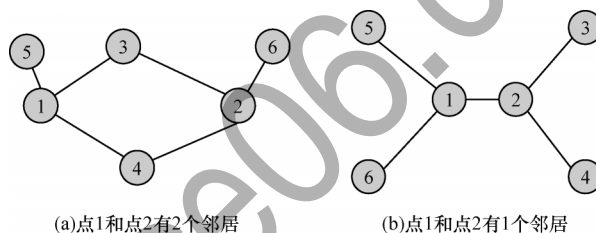


图4 点1和点2的3个最近邻居

Fig.4 Three nearest neighbors of point 1 and point 2

为测量成对相似度 S_{ij} , 根据 $N_i \cap N_j$ 中共享邻居对数据点 X_i 和 X_j 的权重。令 W_{ij} 表示 $N_i \cap N_j$ 中共享的邻居的权重。令 X_r 为 $N_i \cap N_j$ 中共享的邻居之一。假设 X_r 是 X_i 的第 i_r^{th} 个邻居和 X_j 的第 j_r^{th} 个邻居, 则邻居 W_{ij} 的权重表达式为:

$$W_{ij} = \sum_{X_r \in N_i \cap N_j} (k - i_r^{\text{th}} + 1)(k - j_r^{\text{th}} + 1) \quad (3)$$

为了进行统计分析, 考虑在 $S_{ij} \in [0, 1]$ 范围内的成对相似性。计算成对相似度 S_{ij} 为:

$$S_{ij} = \frac{W_{ij}}{\max \{W_{ij}\}} \quad (4)$$

相似矩阵 S 为:

$$S = (S_{ij})_{i,j=1,2,\dots,n} \quad (5)$$

通过对 DAN 进行加权构建一种基于共享邻的局部自适应邻域相似矩阵。

2.3 基于共享最近邻的密度自适应邻域谱聚类

为提高传统谱聚类算法对相似性矩阵构造参数的敏感性, 以及相似性度量难以准确反映复杂数据和非凸数据的结构, 增加聚类算法结果的稳定性。

本文在原有谱聚类算法的基础上,结合一种密度自适应邻域构建方法,并通过共享最近邻进行样本点的相似性衡量,提出一种新的基于共享近邻的密度自适应邻域谱聚类算法 SC-DANSN。利用密度自适应邻域构建方法构建无向图 DAN;通过最终邻居集基于共享最近邻对 DAN 的边进行加权;生成相似矩阵,并计算相应的拉普拉斯矩阵和度矩阵,再进行特征向量的计算,最终通过 K-means 算法进行聚类得到最终的聚类结果。

算法 3 基于共享最近邻的密度自适应邻域谱聚类算法

输入 数据集 D 聚类数目 K

输出 K 个簇的集合 $C=\{C_1, C_2, \dots, C_K\}$

步骤 1 通过 NC 算法构建密度自适应邻域,并生成无向图 DAN,列出 DAN 中共享最近邻集合。

步骤 2 如果 DAN 中的连通子图数大于 K ,则在最近的连通子图之间插入边。相应地更新已连通子图的数量和 DAN 图,直到连通子图数等于 K 。

步骤 3 基于共享最近邻测量成对相似度 S_{ij} ,对 DAN 进行加权,构建相似矩阵 S 。

步骤 4 根据相似矩阵 S ,计算度矩阵 B 及其对应的规范化拉普拉斯矩阵 L 。

步骤 5 计算拉普拉斯矩阵 L 对应的特征向量,并选取前 K 个最大特征值对应的特征向量构成矩阵 X 。

步骤 6 通过 K-means 算法对 X 进行聚类,最终输出 K 个簇的集合 $C=\{C_1, C_2, \dots, C_K\}$,算法结束。

SC-DANSN 算法主要由 3 部分构成,第 1 部分为通过 NC 算法构造 DAN 图,具体分为 5 步,步骤 1 和步骤 2 的时间复杂度由密度计算确定,该步骤的程序执行次数与样本点个数 n 成 10^3 增长,即时间复杂度为 $O(n^3)$ 。步骤 3 和步骤 4 在整个过程中的时间复杂度为 $O(n^2)$ 。步骤 5 在 $O(n^2)$ 时间内检查点与其邻域的相互连通性,重复进行直到邻域不再变化。所以构造 DAN 图即 NC 算法的时间复杂度为 $O(n^3)$ 。第 2 部分为测量成对相似的构造相似性矩阵 S ,该部分的时间复杂度为 $O(n^2d)$, d 表示数据集的维度即特征数目。第 3 部分为聚类部分,采用 K-means 算法,故该部分的时间复杂度为 $O(tkn)$, t 表示迭代次数, k 表示类别数目。结合上述 3 部分分析,由于构建 DAN 时的算法复杂度较高,本文所提算法 SC-DANSN 的时间复杂度为 $O(n^3)$ 比传统谱聚类算法时间复杂度 $O(n^2)$ 高,

3 实验与结果分析

实验环境为 Intel® Core™i7-6700HQ CPU@2.60 GHz,内存为 8.0 GB;编程环境为 PyCharm;在 Windows10 操作系统的计算机上进行测试。通过在人工数据集和 UCI 数据集上进行实验,评估和分析所提算法。

为了比较和分析聚类结果,在以下实验中采用评估 2 种聚类性能方法归一化互信息 (NMI)^[18] 和兰德指数 (RI)^[20]。

归一化互信息 (NMI) 被广泛用于评估聚类算法性能。令 $C=\{C_1, C_2, \dots, C_K\}$ 表示正确的聚类结果, $C'=\{C'_1, C'_2, \dots, C'_K\}$ 表示通过聚类算法获得的预测聚类结果。 $P(c_i)=|c_i|/n$ 是数据点属于簇 $|c_i|$ 的概率,其中 $|c_i|$ 是簇 c_i 的基数, n 是数据点的总数。 $P(c_i \cap c'_j)=|c_i \cap c'_j|/n$ 是数据点属于群集 c_i 和 c'_j 交集的概率。NMI 计算如下:

$$N(C, C') = \frac{2\varphi(C, C')}{\varphi(C) + \varphi(C')}$$

其中: $\varphi(C) = -\sum_{i=1}^K P(c_i) \lg P(c_i)$; $\varphi(C, C') = \sum_{i=1}^K \sum_{j=1}^K P(c_i \cap c'_j)$

$$\lg \frac{P(c_i \cap c'_j)}{P(c_i)P(c'_j)}。$$

N 值越大表示聚类结果越好, N 最大为 1, 表示所有数据点均被正确分类。

兰德指数 (RI) 用于测量 2 个群集的相似性,考虑到同群集和不同群集中存在的数据点数量。RI 将群集标签分配视为数据点之间的成对关系,表明每对数据点可以分配给相同的群集,还可以属于不同的群集。对于具有 N 个数据点的数据集, RI 的计算如下:

$$R(U, V) = \frac{\binom{N_k}{2} - \left[\sum_l \binom{N_l}{2} \cdot \sum_k \binom{N_k}{2} \right] / \binom{N}{2}}{\left((1/2) \left[\sum_l \binom{N_l}{2} + \sum_k \binom{N_k}{2} \right] - \left[\sum_l \binom{N_l}{2} \cdot \sum_k \binom{N_k}{2} \right] / \binom{N}{2} \right)}$$

其中, N_{lk} 表示在 U 中属于相同簇 l 的数据点和 V 中属于相同簇 k 数据点的数量, N_l 表示分配给 U 中的相同簇 l 并属于其中不同簇数据点的数量, N_k 表示属于 U 中不同簇并分配给 V 中的相同簇 k 数据点的数量。RI 的值在 0 ~ 1, 其中 RI 值越高表示聚类效果越好。

3.1 人工合成数据集

在人工合成数据集中随机选取 4 个数据集,分别测试 K-means、SC-KNN、SC-DANSN 算法的聚类效果。采用的数据集属性如表 1 所示。

表 1 人工合成数据集参数设置

数据集	样本数	维度	类别
ThreeCircles	3 603	2	3
TwoMoons	1 502	2	2
Five_cluster	2 000	2	5
Cluto_t4	8 000	2	7

聚类结果如图 5~图 8 所示。在凸数据集 Five_cluster, K-means、SC-KNN 和 SC-DANSN 算法均能正确的聚类如图 7 所示,而在其他 3 个非凸数据集, K-means 算法效果最差(见图 5、图 6、图 8)。本次实验固定 SC-KNN 算法的核参数为 5, K 最近邻为 20 即构建相似图时选取 20 个最近的邻居作为构图参数。由于 SC-KNN 算法受限于相似矩阵构建时参数的影响, 聚类效果不稳定, 需要不断地尝试选择参数来实现正确的聚类。本次实验中 SC-DANSN 算法的共享最近邻数量也设为 20。由于 SC-DANSN 算法采用一种密度自适应邻域方法来构造相似图, 无需指定构建相似图的参数信息, 只需指定共享最近邻的数量以测量成对相似性。由于 SC-DANSN 算法无参构造相似图的特性, 无需像传统谱聚类算法要不断尝试构造相似图的参数, 所以其聚类效果稳定并能正确聚类。对于加入噪声的数据集 Cluto_t4, SC-KNN 算法无论进行何种调参, 噪声样本都无法很好地分离。而 SC-DANSN 算法通过引入共享最近邻能较好地分离噪声信息, 在 3 种算法中抗噪声性能最优。

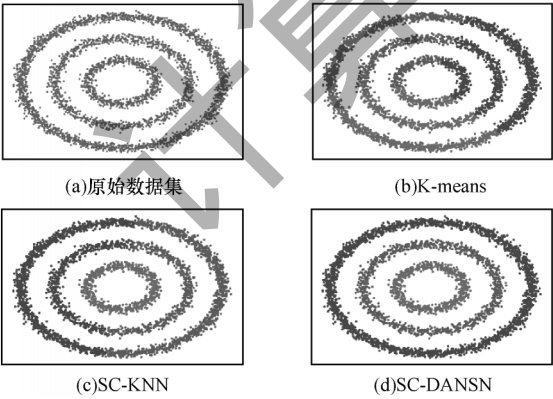


图 5 ThreeCircles 数据集聚类结果
Fig.5 The clustering results of ThreeCircles datasets

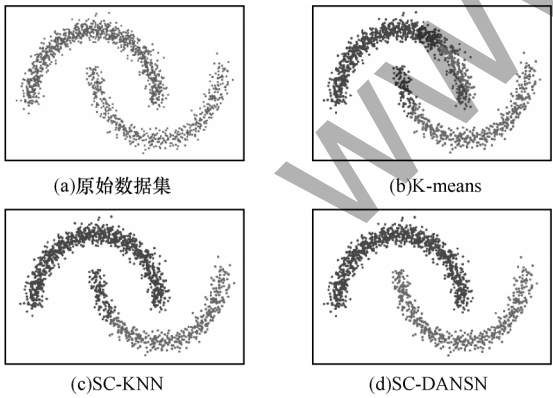


图 6 TwoMoons 数据集聚类结果
Fig.6 The clustering results of TwoMoons datasets

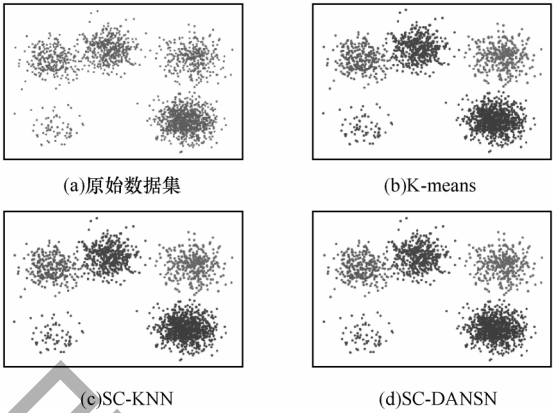


图 7 Five_cluster 数据集聚类结果
Fig.7 The clustering results of Five_cluster datasets

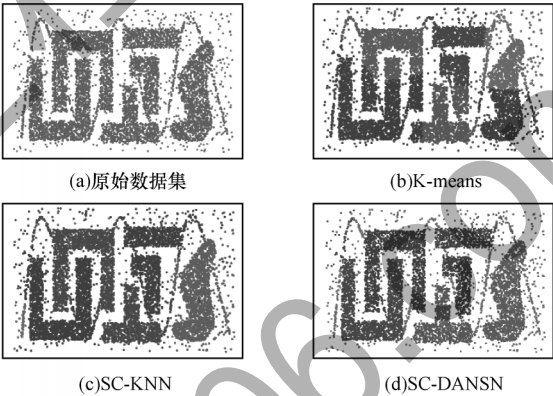


图 8 Cluto_t4 数据集聚类结果
Fig.8 The clustering results of Cluto_t4 datasets

3.2 UCI 真实数据集

为进一步验证 SC-DANSN 算法的有效性, 随机选取 UCI 机器学习库的 4 个真实数据集。4 个数据集的属性如表 2 所示。

表 2 UCI 数据集参数设置			
Table 2 Parameters setting of UCI datasets			
数据集	样本数	维度	类别
Iris	150	4	3
Breast	699	9	2
Wine	178	13	3
Ecoli	336	7	8

本次实验对每种数据集进行 10 次独立的实验, 对于 SC-KNN、SC-DANSN 算法, 取 K 值从 5~50 步长为 5, K 在 SC-KNN 算法中代表 K 最近邻数量, 在 SC-DANSN 算法中代表共享最近邻的数量。经过 10 次独立的试验后, K-means、SC-KNN、SC-DANSN 算法在 UCI 数据集的 RI 和 NMI 值的平均值如表 3 所示。结果表明基于 RI 和 NMI 准则, 在 4 种 UCI 数据集中, SC-DANSN 算法的聚类效果最好, SC-KNN 次之, 两种算法聚类结果相似。由于 K-means 基于划分的聚类规则, 对于非凸数据集分离效果不好, 相比其他谱聚类算法, 其聚类效果最差。

表3 UCI数据集实验结果

Table 3 Experimental results of UCI datasets

算法 数据集	K-means		SC-KNN		SC-DANSN	
	RI	NMI	RI	NMI	RI	NMI
Iris	0.551	0.625	0.812	0.781	0.868	0.857
Breast	0.703	0.689	0.872	0.790	0.876	0.792
Wine	0.694	0.475	0.867	0.875	0.878	0.899
Ecoli	0.654	0.622	0.723	0.741	0.735	0.764

为验证 SC-DANSN 算法对参数选择的不敏感性,使用 SC-DANSN 和 SC-KNN 算法对 4 个 UCI 数据集的最近邻数 K 的变化进行聚类实验。SC-DANSN 和 SC-KNN 算法在 K 值选取 4 个最佳结果时的敏感性如图 9~图 12 所示,可以看出,SC-DANSN 算法对参数 K 的敏感性更小。

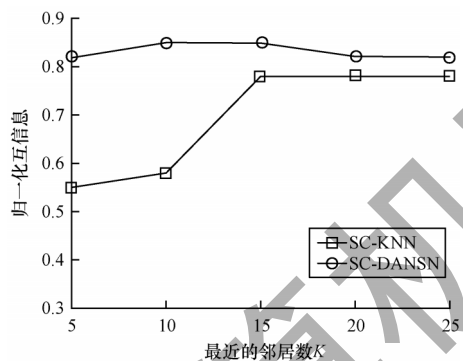


图9 Iris数据集实验结果

Fig.9 Experimental results of Iris dataset

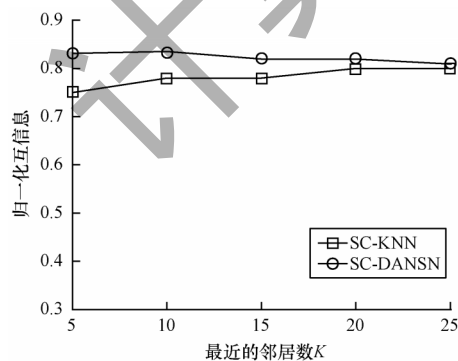


图10 Breast数据集实验结果

Fig.10 Experimental results of Breast dataset

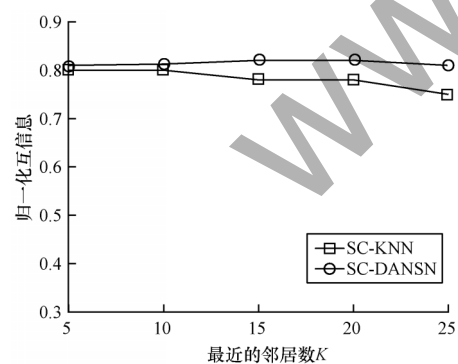


图11 Wine数据集实验结果

Fig.11 Experimental results of Wine dataset

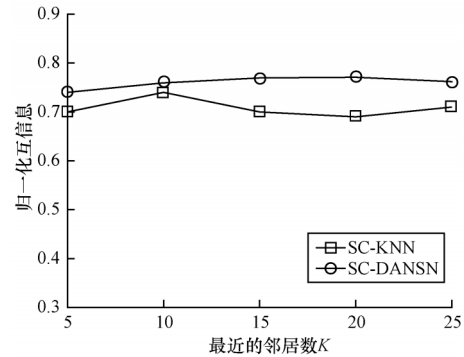


图12 Ecoli数据集实验结果

Fig.12 Experimental results of Ecoli dataset

4 结束语

本文提出一种基于共享最近邻的密度自适应邻域谱聚类算法 SC-DANSN,以降低谱聚类算法对构造相似图的参数敏感性并充分分离数据集。该算法的相似性度量是基于无向 DAN 图中共享的最近邻居的接近度,与基于无向 KNN 图的传统谱聚类算法相比,其对共享最近邻居数 K 不敏感。实验结果表明,在人工合成数据集和 UCI 真实数据集上,SC-DANSN 分离复杂结构和含噪声数据集的性能优于传统的谱聚类算法。下一步考虑使用分布式和并行算法构造相似性矩阵并计算特征向量,并将谱聚类算法应用于大数据集。

参考文献

- [1] AGGARWAL C C, REDDY C K. Data clustering: algorithms and applications [M]. London, UK: Taylor and Francis Group, 2014: 4-7.
- [2] ANTER A, HASSENIAN A E, OLIVA D. An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural [J]. Expert Systems with Applications, 2019, 118: 340-354.
- [3] DING S, JIA H, ZHANG L, et al. Research of semi-supervised spectral clustering algorithm based on pairwise constraints [J]. Neural Computer & Applications, 2014, 24: 211-219.
- [4] WANG L, DING S, JIA H. An improvement of spectral clustering via message passing and density sensitive similarity [J]. IEEE Access, 2019, 7: 101054-101062.
- [5] LUXBURG U V, A tutorial on spectral clustering [J]. Statist. Comput, 2007, 17(4): 395-416.
- [6] 牛科, 张小琴, 贾郭军. 基于距离度量学习的集成谱聚类 [J]. 计算机工程, 2015, 41(1): 207-210.
- NIU K, ZHANG X Q, JIA G J. Integrated spectral clustering based on distance metric learning [J]. Computer Engineering, 2015, 41(1): 207-210. (in Chinese)
- [7] 乔晓明, 潘晓英. 基于稀疏图的鲁棒谱聚类算法 [J].

- 计算机应用研究,2018,35(6):1-2.
- QIAO X M, PAN X Y. Robust spectral clustering algorithm based on sparse graph [J]. Application Research of Computers, 2018, 35(6): 1-2. (in Chinese)
- [8] ZELNIK-MANOR L, PERONA P. Self-tuning spectral clustering [C]//Proceedings of the Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2004: 1601-1608.
- [9] LIU X Y, LI J W, YU H, et al. Adaptive spectral clustering based on shared nearest neighbors [J]. Journal of Chinese Computer System, 2011, 32(9): 1876-1880.
- [10] TAO X M, SONG S Y, CAO P D, et al. A spectral clustering algorithm based on manifold distance kernel [J]. Information and Control, 2012, 41(3): 307-313.
- [11] NG A Y, JORDAN M I, WEISS Y. On spectral clustering: analysis and an algorithm [C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2002: 849-856.
- [12] LI Z, LIU J, CHEN S, et al. Noise robust spectral clustering [C]//Proceedings of the 11th IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2007: 361-368.
- [13] ZHANG X, LI J, YU H. Local density adaptive similarity measurement for spectral clustering [J]. Pattern Recognition Letters, 2011, 32(2): 352-358.
- [14] CAO J, CHEN P, YUN Z, et al. A max-flow-based similarity measure for spectral clustering [J]. ETRI Journal, 2013, 35(2): 311-320.
- [15] XIONG C, JOHNSON D M, CORSO J J. Spectral active clustering via purification of the k -nearest neighbor graph [C]//Proceedings of International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2012.
- [16] 程士卿,郝问裕,李晨,等. 低秩张量分解的多视角谱聚类算法 [J]. 西安交通大学学报, 2019, 54(3): 119-125.
- CHENG S Q, HAO W Y, LI C, et al. Low-rank tensor decomposition based multi-view spectral clustering algorithm [J]. Journal of Xi'an Jiaotong University, 2019, 54(3): 119-125. (in Chinese)
- [17] SUN L, LIU R, XU J, et al. An affinity propagation clustering method using hybrid Kernel function with LLE [J]. IEEE Access, 2018, 6: 68892-68909.
- [18] JANANI R, VIJAYARANI S. Text document clustering using spectral clustering algorithm with particle swarm optimization [J]. Expert Systems with Applications, 2019, 134: 192-200.
- [19] NKAYA T, KAYALGIL S, ÖZDEMIRAL N E. An adaptive neighborhood construction algorithm based on density and connectivity [J]. Pattern Recognition Letters, 2014, 52: 17-24.
- [20] TAO X M, WANG R T, CHANG R, et al. Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies [J]. Knowledge-Based Systems, 2019, 170: 26-42.
- 编辑 薛晋栋
-
- (上接第115页)
- [24] WANG Z, ZHAO Y, XI J, et al. Fast ranking influential nodes in complex networks using a k-shell iteration factor [J]. Physica A: Statistical Mechanics and its Applications, 2016, 461(1): 171-181.
- [25] SHEIKHAHMADI A, NEMATBAKHSI M A, ZAREIE A. Identification of influential users by neighbors in online social networks [J]. Physica A: Statistical Mechanics and its Applications, 2017, 486(15): 517-534.
- [26] MOLINERO X, RIQUELME F, SERNA M. Cooperation through social influence [J]. European Journal of Operational Research, 2015, 242(3): 960-974.
- [27] IRFAN M T, ORTIZ L E. On influence, stable behavior, and the most influential individuals in networks: a game-theoretic approach [J]. Artificial Intelligence, 2014, 215: 79-119.
- [28] BASSOLAS A, BARBOSA-FILHO H, DICKINSON B, et al. Hierarchical organization of urban mobility and its connection with city livability [J]. Nature Communications, 2019, 10(1): 33-45.
- [29] KITSACK M, GALLOS L K, HAVLIN S, et al. Identification of influential spreaders in complex networks [J]. Nature Physics, 2010, 6(11): 888-893.
- [30] ZAREIE A, SHEIKHAHMADI A. A hierarchical approach for influential node ranking in complex social networks [J]. Expert Systems with Application, 2018, 93: 200-211.
- [31] BIANCHINI M, GORI M, SCARSELLI F. Inside PageRank [J]. ACM Transactions on Internet Technology, 2005, 5(1): 92-128.
- [32] Wikipedia. PageRank algorithm [EB/OL]. [2020-07-01]. <http://wiki.swarma.net/index.php?title=PageRank算法&variant=zh-hant>.
- 编辑 陆燕菲