



## 基于密度敏感距离的改进模糊C均值聚类算法

王治和,王淑艳,杜 辉

(西北师范大学 计算机科学与工程学院,兰州 730070)

**摘 要:** 模糊C均值(FCM)聚类算法无法识别非凸数据,算法中基于欧式距离的相似性度量只考虑数据点之间的局部一致性特征而忽略了全局一致性特征。提出一种利用密度敏感距离度量创建相似度矩阵的FCM算法。通过近邻传播算法获取粗类数作为最佳聚类数的搜索范围上限,以解决FCM算法聚类数目需要人为预先设定和随机选定初始聚类中心造成聚类结果不稳定的问题。在此基础上,改进最大最小距离算法,得到具有代表性的样本点作为初始聚类中心,并结合轮廓系数自动确定最佳聚类数。基于UCI数据集和人工数据集的实验结果表明,相比经典FCM、K-means和CFSFDP算法,该算法不仅具有识别复杂非凸数据的能力,而且能够在保证聚类性能和稳定性的前提下加快收敛速度。

**关键词:** 模糊C均值聚类算法;密度敏感距离;近邻传播;初始聚类中心;轮廓系数

开放科学(资源服务)标志码(OSID):



**中文引用格式:** 王治和,王淑艳,杜辉.基于密度敏感距离的改进模糊C均值聚类算法[J].计算机工程,2021,47(5):88-96,103.

**英文引用格式:** WANG Zhihe, WANG Shuyan, DU Hui. Improved fuzzy C-means clustering algorithm based on density-sensitive distance[J].Computer Engineering, 2021, 47(5): 88-96, 103.

## Improved Fuzzy C-means Clustering Algorithm Based on Density-Sensitive Distance

WANG Zhihe, WANG Shuyan, DU Hui

(School of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China)

**[Abstract]** The Fuzzy C-means(FCM) clustering algorithm cannot identify non-convex data, and its similarity measure based on Euclidean distance only considers the local consistency feature between data points while ignoring the global consistency feature. To address the problem, this paper proposes an improved FCM algorithm that uses the density-sensitive distance measure to create the similarity matrix. The proposed algorithm employs the Affinity Propagation (AP) algorithm to obtain the coarse number of clusters as the upper limit of the search of the optimal cluster number, to avoid the instability of clustering results of the classical FCM algorithm, which requires the clustering number to be manually set in advance and initial clustering center to be randomly selected. On this basis, the maximum and minimum distance algorithm is improved to obtain representative sample points as the initial clustering center, and the optimal cluster number is determined based on the silhouette coefficient. Experimental results on UCI and artificial data sets show that compared with the classical FCM, K-means and CFSFDP algorithms, the proposed algorithm is capable of identifying complex non-convex data, and improves the convergence speed with ensured clustering performance and stability.

**[Key words]** Fuzzy C-means(FCM) clustering algorithm; density-sensitive distance; Affinity Propagation(AP); initial clustering center; silhouette coefficient

**DOI:** 10. 19678/j. issn. 1000-3428. 0057901

### 0 概述

聚类分析是将样本对象划分成子集的过程,即

把每个子集作为一个簇,簇中的对象相似程度高,不同簇中的对象相异程度高。目前,聚类分析已被广

**基金项目:** 国家自然科学基金(61962054)。

**作者简介:** 王治和(1965—),男,教授,主研方向为数据挖掘;王淑艳,硕士研究生;杜 辉,副教授、博士。

**收稿日期:** 2020-03-30 **修回日期:** 2020-04-30 **E-mail:** wangshuyan@163.com

泛应用于数据挖掘、模式识别和图像处理等领域,很多经典算法被提出用于样本对象的聚类,主要有基于划分、层次、密度、网格和模型五大类<sup>[1]</sup>。模糊C均值(Fuzzy C-means, FCM)聚类算法是一种基于划分的聚类算法,其因简洁、高效而得到了广泛的应用<sup>[2]</sup>,但在建立相似度矩阵、随机初始化聚类中心和预先确定聚类数目等方面还存在不足。在建立相似度矩阵的过程中,FCM算法采用欧氏距离的相似性度量只对凸数据具有良好的处理性能,在复杂形状和非凸数据中往往会失败,因此,确定合适的相似度矩阵是提高FCM算法聚类性能的关键因素。

相似度矩阵依赖于距离度量这一特点,吸引了很多学者的研究与关注。文献[3]提出一种基于加权欧氏距离的改进FCM算法,其中加权欧氏距离是将特征权值合并到常用的欧氏距离中,结果表明,适当的特征权值分配可以提高FCM算法的聚类性能。文献[4]引入一种鲁棒的非欧氏距离度量方法来提高传统FCM算法的效率,从而减少噪声和异常值对聚类性能的影响。文献[5]提出使用马氏距离和闵可夫斯基距离来代替欧氏距离,提高了FCM算法对于高维数据的识别能力。文献[6]提出一种基于散度相似性度量的FCM算法,其对噪声特征的扰动具有更强的鲁棒性。以上文献虽然提高了FCM算法识别高维数据和噪声等方面的聚类性能,但这些距离度量仍然无法对非凸数据聚类。文献[7]提出一种模糊核C均值聚类算法,该算法采用基于核的距离度量代替欧氏距离作为相似性度量,可以识别任意形状的聚类,但其中核宽度 $\sigma$ 都是通过反复实验得出的,增加了算法的计算复杂度和时间复杂度。文献[8]提出一种基于传递闭包和谱聚类的多中心FCM算法,解决了FCM算法无法处理非凸数据的问题,但算法中对于子簇初始数目和子簇数目的设置都缺乏理论支持。

本文借鉴文献[9]提出的密度敏感距离度量方法,提出一种基于密度敏感距离的改进FCM算法AMMF-DSD。在建立相似度矩阵时采用密度敏感距离代替欧氏距离,以解决FCM算法无法对非凸数据聚类的问题。同时为进一步提高算法的聚类性能,利用近邻传播(Affinity Propagation, AP)聚类算法<sup>[10]</sup>获取粗类数,快速确定最佳聚类数的搜索范围上限,基于此改进最大最小距离算法获得具有代表性的采样点作为FCM算法的初始聚类中心,最后结合轮廓系数<sup>[11]</sup>在聚类数搜索范围内自动确定最佳聚类数。

## 1 相关工作

### 1.1 FCM聚类算法

给定数据集:  $X = (x_1, x_2, \dots, x_n)$ 。其中,每个数据对象  $x_i$  包含  $d$  个特征值,  $n$  是样本数据集的个数。

FCM算法将  $X$  划分为  $k$  个类,  $[v_1, v_2, \dots, v_k]$  为  $k$  个聚类中心。FCM聚类算法的目标函数如式(1)所示:

$$J = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m \|x_i - v_j\|^2 \quad (1)$$

其中,  $m$  是模糊指标,  $m > 1$ ,  $u_{ij}$  是样本点  $x_i$  在第  $j$  分组中的隶属度,  $\|x_i - v_j\|$  是样本点  $x_i$  和聚类中心  $v_j$  之间的欧式距离。在满足约束条件  $\sum_{j=1}^k u_{ij} = 1$  的情况下对

目标函数使用拉格朗日(Lagrange)乘数法,得到隶属度矩阵和聚类中心,分别如式(2)和式(3)所示:

$$u_{ij} = \frac{1}{\sum_{c=1}^k \left( \frac{\|x_i - v_j\|}{\|x_i - v_c\|} \right)^{\frac{2}{m-1}}} \quad (2)$$

$$V_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m} \quad (3)$$

FCM聚类算法具体步骤如下:

#### 算法1 FCM聚类算法

输入 聚类数  $k$ , 初始聚类中心, 模糊指标  $m$ , 终止误差  $\varepsilon$

输出 聚类中心, 隶属度矩阵

步骤1 按式(2)更新隶属度矩阵。

步骤2 按式(3)更新聚类中心。

步骤3 如果  $\|J^{(k)} - J^{(k-1)}\| < \varepsilon$ , 则算法终止; 否则回到步骤1, 继续进行迭代。

### 1.2 密度敏感距离

根据上述FCM算法的过程可以明显看出, 所获得相似度矩阵的准确性直接影响聚类性能。此外, 相似度矩阵主要取决于距离度量的确定。因此, 选择合适的距离度量方法对于提高FCM算法聚类性能至关重要。基于该距离获得的数据点之间的相似性度量必须满足以下两个一致性关系<sup>[9]</sup>: 1) 局部一致性, 即空间上相邻的数据点之间应具有较高的相似性; 2) 全局一致性, 即位于同一流形上的数据点之间应具有较高的相似性。

传统的FCM算法通常采用欧氏距离来确定数据点之间的相似性, 然而欧氏距离只考虑数据点之间的局部一致性特征, 忽略了全局一致性特征。因此, 对于复杂数据和非凸数据, 基于欧氏距离的相似性矩阵往往无法准确地捕获实际的数据结构, 从而导致聚类性能较差。如图1所示, 根据相似测度的全局一致性要求, 同一流形上的数据点应具有较高的相似性, 即点1与点3之间的相似性应高于点1与点2之间的相似性, 但是在按照欧氏距离进行相似性度量时, 点1与点3的相似性要明显小于点1与点2, 这与期望不一致, 即将欧氏距离作为相似性度量不能满足全局一致性。

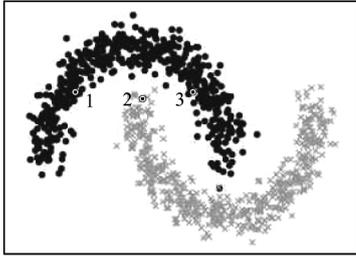


图1 欧式距离无法满足样本全局一致性的情况

Fig.1 The case of Euclidean distance not satisfying the global consistency of samples

为满足聚类结果的全局一致性,使相同流形结构中数据对的相似度高于不同的流形结构,必须使得穿过高密度区域以较短边相连的路径长度低于穿过低密度区域直接相连的两点间距离,即  $\overline{af} + \overline{fe} + \overline{ed} + \overline{dc} + \overline{cb} < \overline{ab}$ ,如图2所示。

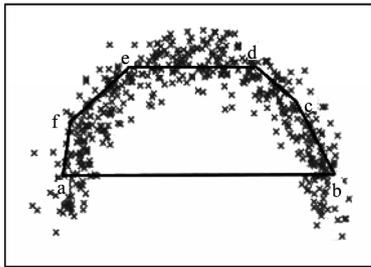


图2 全局一致性距离

Fig.2 Global consistency distance

本文提出一种基于密度敏感距离度量创建相似性矩阵的算法,通过引入密度敏感距离能够同时考虑全局一致性和数据分布的局部一致性,使获得的相似性矩阵可以更准确地捕获真实数据结构,从而解决FCM算法无法识别复杂非凸数据的问题。具体如下:

**定义1** 密度调整长度如式(4)所示:

$$L(x, y) = e^{\rho d(x, y)} - 1 \tag{4}$$

其中,  $d(x, y)$ 表示点  $x$  与点  $y$  间的欧氏距离,  $\rho$  为伸缩因子,通过调节伸缩因子  $\rho$  来放大或缩短两点间线段长度,可以同时满足全局一致性和局部一致性。基于密度调整长度进一步定义密度敏感距离,通过在图中寻找最短路径来测量一对点之间的距离。

**定义2** 将数据点看作一个加权无向图  $G = \{V, E\}$ ,  $V$ 表示顶点集合,  $E$ 表示边集合。令  $P \in V^l$  为图上长度为  $l = |P| - 1$  的连接点  $p_1, p_{|P|}$  之间的路径,其中,边  $(p_k, p_{k+1}) \in E, 1 \leq k < l, p_{ij}$  为连接数据点对  $\{x_i, x_j\}$  的所有路径的集合,  $1 \leq i, j < n, x_i$  与  $x_j$  之间的密度敏感距离如式(5)所示:

$$D_{ij}^\rho = \frac{1}{\rho^2} \ln(1 + d_{sp}(x_i, x_j)) \tag{5}$$

$$d_{sp}(x_i, x_j) = \min_{p \in P_{ij}^\rho} \sum_{k=1}^{|p|-1} (e^{\rho d(p_k, p_{k+1})} - 1) \tag{6}$$

其中,  $d_{sp}(x_i, x_j)$ 表示图  $G$  上节点  $x_i$  和  $x_j$  之间的最短路径距离,  $d(p_k, p_{k+1})$ 是节点  $x_i$  到  $x_j$  最短路径上任意相邻两点的欧氏距离。不难看出,本文提出的距离度量方法可同时满足距离度量的以下4种特性:

- 1) 自反性:  $D_{i,j}^\rho = 0$ , 当且仅当  $x_i = x_j$ 。
- 2) 对称性:  $D_{i,j}^\rho = D_{j,i}^\rho$ 。
- 3) 非负性:  $D_{i,j}^\rho \geq 0$ 。
- 4) 三角不等式:  $D_{i,j}^\rho \leq D_{i,k}^\rho + D_{k,j}^\rho$ 。

### 1.3 轮廓系数

轮廓系数是由KAUFMAN等人提出的一种用于评价算法聚类质量的有效性指标。该指标结合了凝聚度和分离度,不仅能够评价聚类质量,而且还可用于获取最佳聚类数。假设数据集的样本对象  $x_i$  属于类  $A$ 。数据集的聚类轮廓系数  $S_k$  (平均轮廓系数) 定义如式(7)所示:

$$S_k = \frac{1}{n} \sum_{i=1}^n \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{7}$$

其中,  $n$  为数据集中的样本个数,  $a(i)$  表示样本点  $i$  与同簇剩余样本对象的平均距离,  $b(i)$  表示样本点  $i$  与剩余每个簇的样本对象平均距离的最小值。轮廓系数的取值范围在  $[-1, 1]$  之间,其值越大,表明聚类的质量越好。对于现有的分类数,求取轮廓系数的最大值,与之对应的  $k$  值就是最佳聚类数<sup>[12]</sup>。结合相关资料和实验情况可知,本文采用轮廓系数来评价聚类效果从而获得最佳聚类数的方法是有效的。

## 2 基于密度敏感距离的改进FCM算法

### 2.1 基于密度敏感距离的FCM距离度量

改进后的FCM算法距离度量采用密度敏感距离,目标函数如式(8)所示:

$$J = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m D_{ij}^\rho \tag{8}$$

其中,  $D_{ij}^\rho$  为样本  $x_i$  与聚类中心  $v_j$  之间的密度敏感距离,采用式(5)计算,  $u_{ij}$  是样本点  $x_i$  在第  $j$  分组中的隶属度。在满足约束条件  $\sum_{j=1}^k u_{ij} = 1$  的情况下对目标函数使用拉格朗日乘数法求得隶属度矩阵,如式(9)所示:

$$u_{ij} = \frac{1}{\sum_{c=1}^k \left( \frac{D_{ij}^\rho}{D_{ic}^\rho} \right)^{\frac{2}{m-1}}} \tag{9}$$

基于密度敏感距离度量的FCM算法具体步骤如下:

**算法2** 基于密度敏感距离度量的FCM算法

输入 聚类数  $k$ , 模糊指标  $m$ , 初始聚类中心, 终止误差  $\varepsilon$

**输出** 聚类中心,隶属度矩阵

**步骤1** 第 $c$ 次迭代,根据式(9)更新隶属度矩阵。

**步骤2** 根据式(11)更新聚类中心。

**步骤3** 根据新得的聚类中心从密度敏感距离矩阵中获得新的 $D_{ij}$ ,重新计算隶属度函数 $u_{ij}$ ,迭代循环,直到聚类中心不发生变化,算法结束。

算法2中的聚类中心更新方式如下:将数据集中的样本点作为聚类中心,在确定初始聚类中心后,由上述密度敏感距离得到 $D_{ij}^p$ 。目标函数如式(10)所示:

$$J = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m D_{ij}^p = u_{j1}^m D_{j1}^p + u_{j2}^m D_{j2}^p + \cdots + u_{jm}^m D_{jm}^p \quad (10)$$

已知 $u_{ij}^m$ 和任意两样本间的密度敏感距离矩阵 $D^M$ ,即:

$$\begin{bmatrix} D_{11}^p & D_{12}^p & \cdots & D_{1n}^p \\ D_{21}^p & D_{22}^p & \cdots & D_{2n}^p \\ \vdots & \vdots & \ddots & \vdots \\ D_{n1}^p & D_{n2}^p & \cdots & D_{nn}^p \end{bmatrix} \begin{bmatrix} u_{11}^m & u_{21}^m & \cdots & u_{k1}^m \\ u_{12}^m & u_{22}^m & \cdots & u_{k2}^m \\ \vdots & \vdots & \ddots & \vdots \\ u_{1n}^m & u_{2n}^m & \cdots & u_{kn}^m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix} = A_{n \times k}$$

则第 $j$ 类的聚类中心 $v_j$ 如式(11)所示:

$$v_j = \left\{ x_i | i = \underset{c}{\operatorname{argmin}} A_{ij} \right\}, j = 1, 2, \dots, k \quad (11)$$

## 2.2 最佳聚类数的确定

传统的FCM算法采用多种方法获取最佳聚类数 $k_{opt}$ 。文献[13]提出一些检验聚类有效性的函数来评估聚类结果并确定 $k_{opt}$ ,但是这些有效性函数自身存在一定的问题,一般很难直接确定。文献[14-15]提出使用一种新的紧密度和分离度的指标,文献[16-17]使用轮廓系数来确定 $k_{opt}$ ,这些算法在每次迭代中,通过有效性函数衡量聚类结果,最后利用指标值来估计 $k_{opt}$ ,但是对于大型数据集,需要较高的计算量资源。以上研究虽然确定了最佳聚类数,但仍存在 $k$ 值最优搜索不稳定、计算复杂度高等问题,因此,需要事先确定 $k$ 的搜索范围,即确定 $k_{opt}$ 的上下限 $[k_{min}, k_{max}]$ 。由于 $k_{min} = 1$ 表示样本均匀分布,无法区分样本基本特征,因此一般情况下设置 $k_{min}$ 最小为2。关于如何确定 $k_{max}$ ,目前尚无明确的理论指导,很多学者根据经验规则 $k_{max} \leq \operatorname{int}(\sqrt{n})$ 来获取,其中 $n$ 为样本点的个数<sup>[18]</sup>,而文献[19]中所有数据集的样本数和实际类数并不具有这样的性质。由此可见,仅仅利用有效性指标或者经验规则确定FCM算法的最佳聚类数不具有普遍性,且FCM算法聚类中心的随机初始化更是对聚类质量造成了极大的影响。本文在引入密度敏感距离的基础上,利用AP算法获取粗聚类数作为 $k_{max}$ ,并结合轮廓系数自动确定最佳聚类数。AP算法的基本原理是经过样本对象彼此的消息传递以获取高质量的聚类中心,对于类内紧密、类

间远离的聚类结构,AP算法能获得比较准确的聚类结果,但对于比较松散的聚类结构,算法倾向于产生较多的局部聚类,这使得算法产生的聚类数往往偏多,从而不能给出准确的聚类结果<sup>[20]</sup>。AP算法中的偏向参数 $p(i)$ 表示样本点 $x_i$ 被选作聚类中心的倾向性,它对聚类数的大小有重要影响, $p(i)$ 越大,倾向于产生的聚类数越多。本文将 $p(i)$ 统一设置为相似度矩阵的最小值 $s_{min}$ <sup>[10]</sup>。经实验可证明,当 $p = s_{min}$ 时,算法结束时得到的聚类数和经验规则 $\operatorname{int}(\sqrt{n})$ 相比,AP算法获得的聚类数 $k_{AP}$ 更接近正确类数。

## 2.3 初始聚类中心的确定

在上述聚类数搜索范围确定的前提下,基于密度敏感距离度量的FCM算法搜索聚类空间逐步增加聚类数。当聚类数为 $k_{min}$ 时,基于最大最小距离算法原则<sup>[21]</sup>选取 $k_{min}$ 个样本点初始化FCM算法的聚类中心,之后每增加一个聚类数,在保持上一次初始聚类中心不变的基础上,再按照最大最小距离算法原则增加一个初始聚类中心,从而保持聚类结果的稳定性和延续性。基于最大最小距离算法选出的聚类中心倾向于属于不同类别的可能性比较大,这样可以得到较好的聚类结果。传统的最大最小距离算法利用比例系数 $\theta$ 作为限制条件来确定聚类数对聚类结果影响很大,而本文是在聚类数已知的前提下进行的,因此无需设定比例系数 $\theta$ 。此外,最大最小距离算法随机选择初始聚类中心,会使聚类结果不稳定。根据数据的实际分布情况,选取密度最大点作为最大最小距离算法的第一个聚类中心,这样所有的初始聚类中心都是确定的,其最终聚类结果也就保证了唯一且稳定,同时此方法有效地避免了噪声点的选取。

改进的最大最小距离算法具体步骤如下:

**算法3** 改进的最大最小距离算法

**输入** 聚类数搜索范围 $k_{min} = 2, k_{max} = k_{AP}$

**输出**  $k$ 个聚类中心。

**步骤1** 求出各样本点之间的距离 $d_{ij}$ ,将密度最大的一个样本点作为第1个聚类中心 $Z_1$ <sup>[22]</sup>。根据 $\rho_i = \sum_j x(d_{ij} - d_c)$ 确定 $i$ 点的密度大小,以 $i$ 点为圆心,包含在以截断距离 $d_c$ 为半径的圆内点的个数,即为 $i$ 点的密度大小。

**步骤2** 当聚类数为2时,计算剩余样本对象到 $Z_1$ 的距离,找到距离 $Z_1$ 最大的样本点作为第2个聚类中心 $Z_2$ 。

**步骤3** 当聚类数为3时,计算剩余样本对象与 $Z_1, Z_2$ 之间的距离,并求出它们之中的最小值 $D_{zr}$ ,将第 $r$ 个样本作为第3个聚类中心。

**步骤4** 当聚类数为 $k$ 且 $k \leq k_{max}$ 时,对于已有的 $(k-1)$ 个聚类中心,计算剩余不属于聚类中心的样

本对象分别到每个聚类中心的距离  $D_{ij}$ , 并计算  $D_i = \max\{\min(Z_{i1}, Z_{i2}, \dots, Z_{i(k-1)})\}$ , 将第  $t$  个样本作为第  $k$  个聚类中心。当算法满足结束条件时, 算法结束。

#### 2.4 AMMF-DSD 算法

为提高传统 FCM 算法对复杂数据和非凸数据的聚类性能, 提高算法聚类结果的稳定性, 本文在原有 FCM 算法思想的基础上, 提出基于密度敏感距离度量创建相似度矩阵的改进 FCM 算法 AMMF-DSD。首先利用密度敏感距离代替欧式距离创建相似度矩阵; 然后通过设定 AP 算法的偏向参数  $p = s_{\min}$ , 获取粗类数作为  $k_{\max}$ , 基于此改进最大最小距离算法获取一些有代表性的样本点初始化 FCM 算法的聚类中心; 最后结合轮廓系数自动确定最佳聚类数。AMMF-DSD 算法流程如图 3 所示。

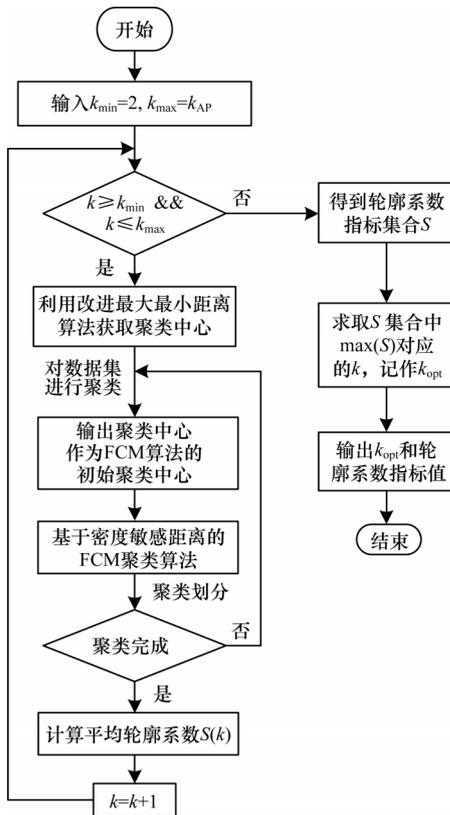


图3 AMMF-DSD 算法流程

Fig.3 Procedure of AMMF-DSD algorithm

对 AMMF-DSD 算法的时间复杂度进行分析, 主要包含以下 3 个部分:

- 1) 利用 AP 算法遍历整个数据集, 以获取粗类数作为  $k_{\max}$ , 其时间复杂度为  $O(n^2)$ , 其中  $n$  是数据点的个数。
- 2) 利用改进最大最小距离算法获取具有代表性的样本点作为 FCM 算法的初始聚类中心, 其时间复杂度为  $O(n^2)$ 。
- 3) 改进的 FCM 算法中主要涉及欧氏距离的计算, 其计算复杂度为  $O(n^2)$ , 引入的密度敏感距离度

量由于采用 Dijkstra 的最短路径算法<sup>[23]</sup>来实现最小路径距离的计算, 其时间复杂度也为  $O(n^2)$ 。

综上所述, AMMF-DSD 算法的时间复杂度为 3 个部分时间复杂度之和  $O(n^2)$ , 即 AMMF-DSD 算法的时间复杂度相比原 FCM 算法没有改变。但 AMMF-DSD 算法具有明显的优势: 按照本文方法获取的  $k_{\max}$  由  $n$  降低为粗类数  $k_{AP}$ , 同时改进最大最小距离算法确定的聚类中心避免了 FCM 算法聚类中心初始化时可能出现的初始聚类中心过于邻近以及多个初始聚类中心都选自同一个类中而小类中没有初始聚类中心的情况。因此, AMMF-DSD 算法收敛速度较快, 可有效减少迭代次数。

### 3 实验与结果分析

通过在人工数据集和 UCI 数据集上进行实验评估和分析本文算法性能。实验环境为 Intel®Core™i5-1035G1CPU@ 1.00 GHz, 内存为 8 GB。编程环境为 Eclipse, MATLAB R2016b 显示实验结果。在 Windows 10 操作系统的计算机上运行通过。实验数据集包括 UCI 数据集 (Iris、Wine、TAE、Seeds、CMC、Blood、Heart-stat-log、Thyroid、Haber-man、Bu-pa) 和人工数据集 (Three-circles、Spiral、Line-blobs、Aggregation、Square1)。对比算法包括 FCM、K-means 和 CFSFDP 算法, 其中, CFSFDP 算法是一种快速搜索查询的利用决策图确定中心的算法<sup>[22]</sup>, K-means 算法采用欧氏距离建立相似度矩阵, 是一种只适用于凸数据的聚类算法<sup>[24]</sup>。本文采用聚类准确率 (ACC)<sup>[25]</sup> 和调整兰德系数 (ARI)<sup>[26]</sup> 对算法的聚类性能进行评估。

聚类准确率 (ACC) 用于评估算法的准确性, 如式 (12) 所示, 其中,  $C_i$  是所提算法的类标签,  $\bar{C}_i$  是数据真实的类标签,  $\delta(x, y)$  表示函数,  $\text{map}(x)$  作为最好的映射函数使用了匈牙利算法进行映射, 对获得的中心和真实的中心进行映射。

$$A_{\text{ACC}} = \frac{\sum_{i=0}^n \delta(\bar{C}_i, \text{map}(C_i))}{n} \quad (12)$$

调整兰德系数 (ARI) 如式 (13) 所示, 其中,  $a$  是属于  $U$  的同类且属于  $V$  的同类的数据对数目,  $b$  是属于  $U$  的同类但属于  $V$  的不同类的数据对数目,  $c$  是属于  $U$  的不同类而属于  $V$  的同类的数据对数目,  $d$  是属于  $U$  的不同类且属于  $V$  的不同类的数据对数目。ARI 数值越接近 1 代表聚类结果越好, 越接近 0 代表聚类结果越差。

$$A_{\text{ARI}} = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} \quad (13)$$

#### 3.1 聚类数搜索范围

本节运用 AP 算法确定聚类数的搜索范围上限,  $k_{\max} = k_{AP}$ , 其中设定 AP 算法中的参考度为相似度矩

阵  $S$  的最小值,即  $p = s_{\min}$  (忽略参数对聚类结果的影响)。与经验规则  $k_{\max} = \text{int}(\sqrt{n})$  进行对比实验,实验结果如表 1 所示。

表 1 AP 算法确定的  $k_{\max}$   
Table 1  $k_{\max}$  determined by the AP algorithm

数据集	样本数	维数	正确类数	$k_{AP}$	$\text{int}(\sqrt{n})$
Iris	150	4	3	3	12
Wine	178	3	3	3	13
TAE	151	5	3	4	12
Seeds	210	7	3	4	14
CMC	1 473	9	3	4	38
Blood	748	4	2	3	27
Heart-stat-log	270	13	2	2	16
Thyroid	215	5	3	4	15
Haber-man	306	3	2	4	17
Bu-pa	345	6	2	2	19
Three-circles	299	2	3	4	17
Spiral	312	2	3	6	18
Line-blobs	266	2	3	6	16
Aggregation	788	2	7	7	28
Square1	1 000	2	4	7	32

从表 1 可以看出,当运用 AP 算法确定  $k_{\max}$  时,UCI 数据集 Iris、Wine、Heart-stat-log、Bu-pa 和人工数据集 Aggregation 获取的聚类数目等于正确类数,而 UCI 数据集 TAE、Seeds、CMC、Blood、Thyroid、Haber-man 和人工数据集 Three-circles、Spiral、Line-blobs、Square1 获取的聚类数均大于正确类数,但是与经验规则  $\text{int}(\sqrt{n})$  确定的  $k_{\max}$  相比,显然 AP 算法获取的聚类数更接近正确类数,大幅缩小了  $k_{\text{opt}}$  的搜索范围,由此验证了将 AP 算法获取的聚类数作为  $k_{\max}$  是合理的。

### 3.2 算法对比与分析

在聚类数搜索范围确定的基础上,分别对 UCI 数据集 Iris、Wine、TAE、Seeds、CMC、Blood、Heart-stat-log、Thyroid、Haber-man、Bu-pa 和人工数据集 Three-circles、Spiral、Line-blobs、Aggregation、Square1 进行的实验。其中,Line-blobs、Three-circles 的伸缩因子  $\rho$  设置为  $e^3$ ,Iris、Wine、TAE、Seeds、CMC、Thyroid 的伸缩因子  $\rho$  设置为  $e^2$ ,Spiral、Aggregation、Square1、Blood、Heart-stat-log、Haber-man、Bu-pa 的伸缩因子  $\rho$  设置为  $e$ 。

#### 3.2.1 最佳聚类数

本节将 AMMF-DSD 算法和随机选取初始聚类中心的 FCM 算法进行实验对比,比较这两种算法关于聚类中心的不同初始化方法对轮廓系数 Silhouette

的影响,进而比较对最佳聚类数  $k_{\text{opt}}$  的确定造成的影响。为减少误差,对每个数据集实验重复运行 10 次,所确定的最佳聚类数  $k_{\text{opt}}$  如表 2 所示。

表 2 最佳聚类数  
Table 2 The optimal number of clusters

数据集	正确类数	$k_{\max}$	FCM 算法		AMMF-DSD 算法	
			$k_{\text{opt}}$	Silhouette	$k_{\text{opt}}$	Silhouette
Three-circles	3	4	4	0.74	3	0.82
Spiral	3	6	6	0.80	3	0.85
Line-blobs	3	6	5	0.86	3	0.93
Aggregation	7	7	7	0.90	7	0.94
Square1	4	7	7	0.89	4	0.91
Iris	3	3	2	0.88	3	0.94
Wine	3	3	4	0.76	3	0.79
TAE	3	4	2	0.28	3	0.62
Seeds	3	4	4	0.81	3	0.86
CMC	3	4	6	0.24	3	0.78
Blood	2	3	3	0.89	2	0.97
Heart-stat-log	2	2	2	0.73	2	0.77
Thyroid	3	4	4	0.71	4	0.94
Haber-man	2	4	3	0.75	2	0.79
Bu-pa	2	2	2	0.59	2	0.94

从表 2 可以看出,在聚类数搜索范围确定时,AMMF-DSD 算法对于各种数据集获得的最佳聚类数都等于正确类数,而 FCM 算法只有 Aggregation、Heart-stat-log、Bu-pa 数据集的  $k_{\text{opt}}$  等于正确类数,且 AMMF-DSD 算法得到的  $k_{\text{opt}}$  对应的轮廓系数均大于 FCM 算法,这进一步验证了改进后的算法 AMMF-DSD 是有效的且获得的最佳类数是合理的。

由于传统的 FCM 算法随机选取初始聚类中心,使聚类结果存在不稳定的现象,因此随机选取 4 个数据集(Spiral、Line-blobs、Iris 和 Wine)对 AMMF-DSD 和 FCM 算法进行算法稳定性对比,实验结果如图 4 所示。从图 4 可以看出,FCM 算法的轮廓系数会随着实验次数的不同而呈现出不同的聚类结果,其原因是 FCM 算法的初始聚类中心是随机选取的,因此聚类结果也表现出不稳定的状态,而 AMMF-DSD 算法是对传统 FCM 算法的改进,避免了初始聚类中心随机选取的问题,且聚类数的搜索范围又是确定的,其聚类结果就表现出较强的稳定性。AMMF-DSD 算法和 FCM 算法聚类时得到的迭代次数如图 5 所示。从图 5 可以看出,AMMF-DSD 算法的迭代次数明显小于 FCM 算法,即 AMMF-DSD 算法加快了算法的收敛速度,而 FCM 算法的迭代次数仍在不断变化。

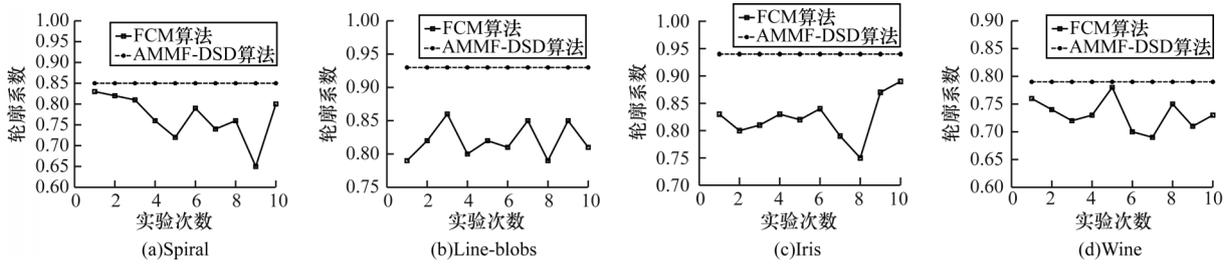


图4 FCM和AMMF-DSD算法在4个数据集上的稳定性对比

Fig.4 Stability comparison of FCM algorithm and AMMF-DSD algorithm on four data sets

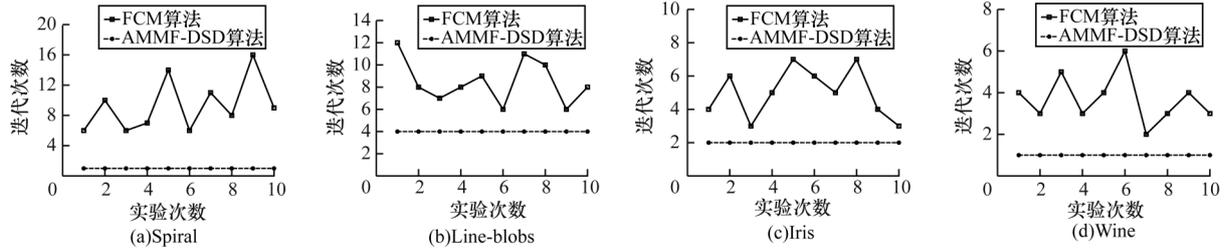


图5 FCM和AMMF-DSD算法在4个数据集上的迭代次数对比

Fig.5 Iteration time comparison of FCM algorithm and AMMF-DSD algorithm on four data sets

3.2.2 人工数据集上的实验

分别在 Three-circles、Spiral、Line-blobs、Aggregation 和 Square1 这5个人工数据集上使用4种聚类算法进行实验,实验数据集见表1,聚类结果如图6~图10所示。从图6可以看出,FCM、K-means和CFSFDP算法在 Three-circles数据集上的聚类效果都不理想,而AMMF-DSD算法能够正确划分数据类别。从图7可以看出,FCM、K-means和CFSFDP算法在Spiral数据集上依然聚类效果不佳,不能正确聚类,而AMMF-DSD算法将

正确地划分了数据类别。从图8可以看出,FCM和K-means算法在Line-blobs数据集上的聚类效果不理想,CFSFDP和AMMF-DSD算法则得到了正确的聚类结果。从图9可以看出,AMMF-DSD算法的聚类效果最好,CFSFDP算法次之,FCM和K-means算法在Aggregation数据集上的聚类效果都不好。从图10可以看出,在Square1数据集上,AMMF-DSD算法聚类效果最优,FCM和CFSFDP算法仅次之,而K-means算法的聚类效果最差。

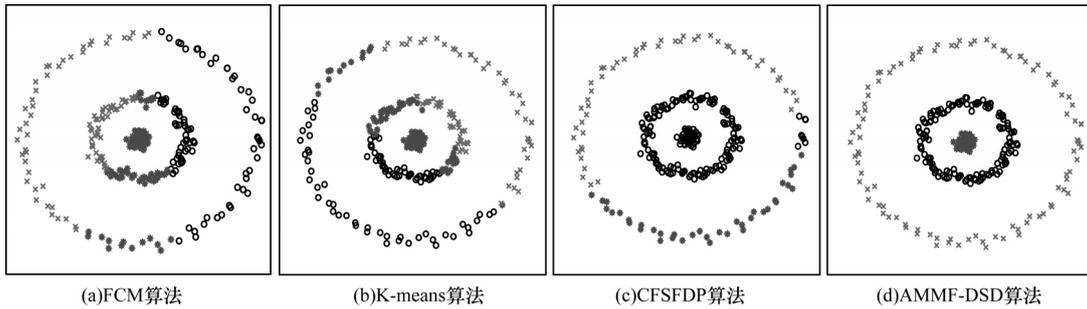


图6 4种聚类算法对数据集 Three-circles的聚类结果

Fig.6 Clustering results of four clustering algorithms on Three-circles data set

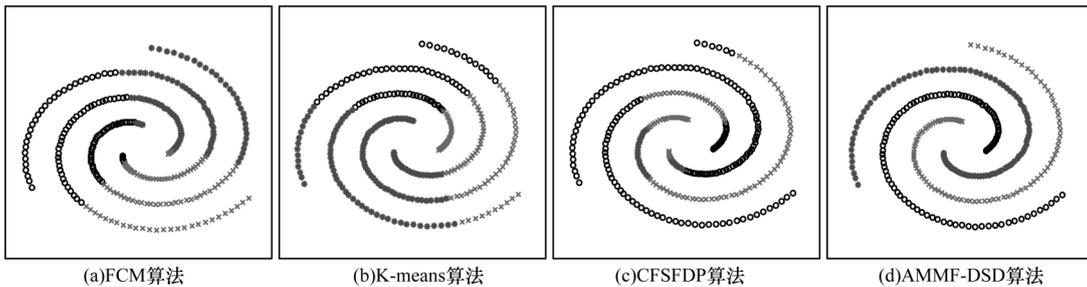


图7 4种聚类算法对数据集 Spiral的聚类结果

Fig.7 Clustering results of four clustering algorithms on Spiral data set

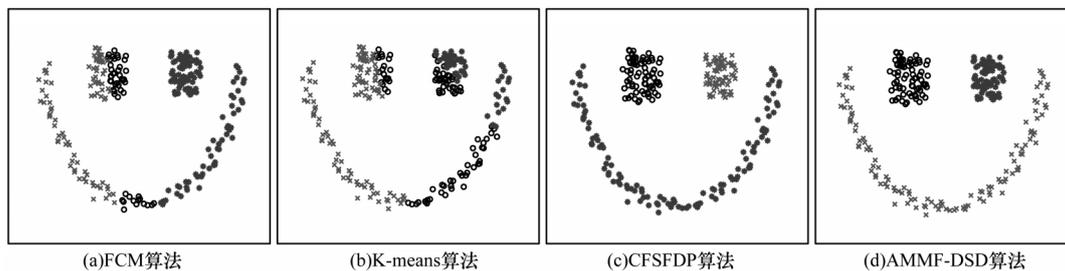


图 8 4 种聚类算法对数据集 Line-blobs 的聚类结果

Fig.8 Clustering results of four clustering algorithms on Line-blobs data set

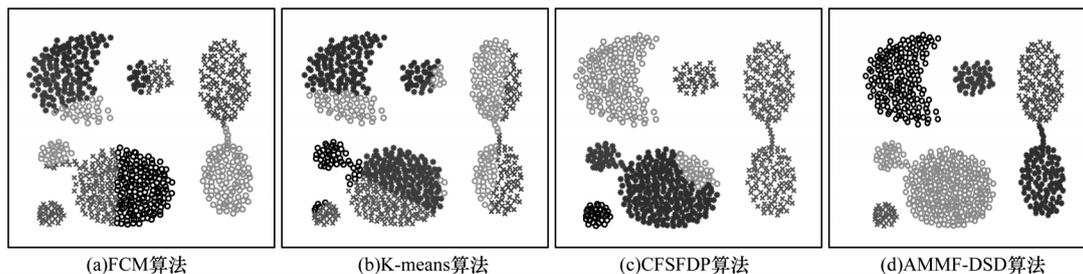


图 9 4 种聚类算法对数据集 Aggregation 的聚类结果

Fig.9 Clustering results of four clustering algorithms on Aggregation data set

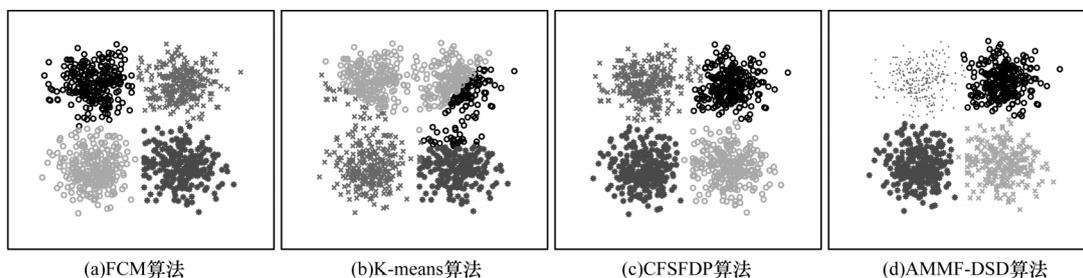


图 10 4 种聚类算法对数据集 Square1 的聚类结果

Fig.10 Clustering results of four clustering algorithms on Square1 data set

通过对图 6~图 10 实验的可视化对比实验分析可知, AMMF-DSD 算法比 K-means、FCM 和 CFSFDP 算法更擅长对非凸数据和复杂形状的数据进行聚类。以上 4 种聚类算法在人工数据集上的性能对比如表 3 所示。从表 3 可以看出, AMMF-DSD 算法在 Three-circles、Spiral、Line-blobs 和 Square1 数据集上的聚类指标值都是 1, 在 Aggregation 数据集上的聚类指标值均大于对比算法, 聚类性能最好, CFSFDP 算法仅在 Line-blobs 数据集上的聚类指标值是 1。从聚类指标值来看, AMMF-DSD 算法聚类性能最优, CFSFDP 算法次之, 而 FCM 和 K-means 算法最差。可见, 用密度敏感距离代替欧氏距离创建相似度矩阵大幅提高了原始 FCM 算法的聚类性能, 聚类数搜索范围的确定和初始聚类中心的确定也提高了 AMMF-DSD 算法的稳定性, 聚类效果较好。

表 3 4 种聚类算法在人工数据集上的性能对比

Table 3 Performance comparison of four clustering algorithms on artificial data sets

数据集	FCM 算法		K-means 算法		CFSFDP 算法		AMMF-DSD 算法	
	ACC	ARI	ACC	ARI	ACC	ARI	ACC	ARI
Three-circles	0.515	0.140	0.508	0.133	0.642	0.434	1.000	1.000
Spiral	0.346	0.004	0.407	0.011	0.532	0.279	1.000	1.000
Line-blobs	0.594	0.264	0.496	0.158	1.000	1.000	1.000	1.000
Aggregation	0.688	0.665	0.622	0.585	0.821	0.833	0.989	0.982
Square1	0.989	0.978	0.767	0.716	0.986	0.972	1.000	1.000

### 3.2.3 UCI 数据集上的实验

本组实验选取 10 个 UCI 数据集将 AMMF-DSD 算法的聚类结果同 CFSFDP、FCM 和 K-means 算法的聚类结果进行比较, 实验数据集见表 1, 各算法得到的 ACC 和 ARI 指标值见表 4。为了减少实验误差, 每个数据集独立运行 10 次。从表 4 可以看出: AMMF-DSD 算法在这 10 个 UCI 数据集上的聚类指标值均高于 K-means、FCM 和 CFSFDP 算法, 聚类性

能最好;本文算法的聚类结果是相对稳定的,因此聚类效果较好;CFSFDP算法次之;K-means、FCM算法的指标值随着实验次数的不同而呈现出不同的聚类结果,聚类效果欠佳。通过上述分析可以看出,AMMF-DSD算法具有较好的聚类性能,并且聚类结果也更稳定。

表4 4种聚类算法在UCI数据集上的性能对比

Table 4 Performance comparison of four clustering algorithms on UCI data set

数据集	FCM 算法		K-means 算法		CFSFDP 算法		AMMF-DSD 算法	
	ACC	ARI	ACC	ARI	ACC	ARI	ACC	ARI
Iris	0.757	0.573	0.720	0.502	0.701	0.533	0.942	0.850
Wine	0.410	0.009	0.466	0.007	0.444	0.013	0.899	0.851
TAE	0.368	0.010	0.366	0.012	0.369	0.003	0.782	0.607
Seeds	0.783	0.621	0.726	0.454	0.703	0.494	0.910	0.737
CMC	0.301	0.367	0.391	0.405	0.412	0.448	0.690	0.549
Blood	0.707	0.668	0.599	0.607	0.656	0.686	0.778	0.785
Heart-stat-log	0.585	0.020	0.590	0.029	0.552	0.216	0.756	0.758
Bu-pa	0.510	0.547	0.538	0.551	0.542	0.551	0.617	0.571
Thyroid	0.617	0.172	0.651	0.200	0.521	0.078	0.740	0.694
Haber-man	0.562	0.667	0.601	0.682	0.572	0.663	0.742	0.719

#### 4 结束语

针对传统FCM算法无法识别非凸数据,同时对复杂形状的数据聚类性能不佳的问题,本文提出使用密度敏感距离代替欧氏距离创建相似度矩阵的AMMF-DSD算法。该距离度量通过调整伸缩因子 $\rho$ ,可以同时满足全局一致性和局部一致性,使得到的相似度矩阵能够更准确地捕获真实的数据结构,从而实现非凸数据的聚类。同时,使用AP算法确定最佳聚类数的搜索范围上限 $k_{\max}$ ,基于此改进最大最小距离算法获取代表点初始化FCM算法的聚类中心,并结合轮廓系数确定最佳聚类数。实验结果表明,AMMF-DSD算法能够对非凸数据和复杂形状的数据进行聚类并提高算法的聚类性能和稳定性,同时加快算法的收敛速度。但是该算法在处理大规模数据时需要较大的存储空间和计算时间,并且算法结果受参数取值的影响大,下一步将结合Spark框架和抽样技术实现算法的并行化,改善算法的大数据聚类性能并减小对参数的敏感度。

#### 参考文献

[1] HAN J, KAMBER M, PEI J. Data mining concept and techniques[M]. [S. l.]: Morgan Kaufmann, 2011.  
[2] BEZDEK J C. Pattern recognition with fuzzy objective function algorithms[J]. Advanced Applications in Pattern

Recognition, 1981, 22(1171): 203-239.

- [3] WANG Xizhao, WANG Yadong, WANG Lijuan. Improving fuzzy C-means clustering based on feature-weight learning[J]. Pattern Recognition Letters, 2004, 25(10): 1123-1132.  
[4] KANNAN S R, DEVI R, RAMATHILAGAM S, et al. Effective FCM noise clustering algorithms in medical images[J]. Computers in Biology & Medicine, 2013, 43(2): 73-83.  
[5] GUEORGUEVA N, VALOVA I, GEORGIEV G. M&MFCM: fuzzy C-means clustering with Mahalanobis and Minkowski distance metrics[J]. Procedia Computer Science, 2017, 114: 224-233.  
[6] SEAL A, KARLEKAR A, KREJCAR O, et al. Fuzzy C-means clustering using Jeffreys-divergence based similarity measure[J]. Applied Soft Computing, 2020, 88: 1-5.  
[7] KANG Jiayin, JI Zhicheng, GONG Chenglong. Kernelized fuzzy C-means clustering algorithm and its application[J]. Chinese Journal of Scientific Instrument, 2010, 31(7): 1657-1663. (in Chinese)  
康家银, 纪志成, 龚成龙. 一种核模糊C均值聚类算法及其应用[J]. 仪器仪表学报, 2010, 31(7): 1657-1663.  
[8] ZENG Shan, TONG Xiaojun, SANG Nong. Study on multi-center fuzzy C-means algorithm based on transitive closure and spectral clustering[J]. Applied Soft Computing, 2014, 16: 89-101.  
[9] TAO Xinmin, WANG Ruotong, CHANG Rui, et al. Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies[J]. Knowledge-Based Systems, 2019, 170: 26-42.  
[10] FREY B J, DUECK D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.  
[11] SUBBALAKSHMI C, KRISHNA G R, RAO S K M, et al. A method to find optimum number of clusters based on fuzzy silhouette on dynamic data set[J]. Procedia Computer Science, 2015, 46: 346-353.  
[12] ESTIRI H, OMRAN B A, MURPHY S N. kluster: an efficient scalable procedure for approximating the number of clusters in unsupervised learning[J]. Big Data Research, 2018, 13: 38-51.  
[13] ZHU Erzhou, ZHANG Yuanxiang, WEN Peng, et al. Fast and stable clustering analysis based on grid-mapping K-means algorithm and new clustering validity index[J]. Neurocomputing, 2019, 363: 149-170.  
[14] PHAM V N, NGO L T, PEDRYCZ W. Interval-valued fuzzy set approach to fuzzy Co-clustering for data classification[J]. Knowledge-Based Systems, 2016, 107: 1-13.  
[15] HANMANDLU M, VERMA O P, SUSAN S, et al. Color segmentation by fuzzy co-clustering of chrominance color features[J]. Neurocomputing, 2013, 120: 235-249.  
[16] de AMORIM R C, HENNIG C. Recovering the number of clusters in data sets with noise features using feature rescaling factors[J]. Information Sciences, 2015, 324: 126-145.

(下转第103页)

- of Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Stroudsburg, USA: ACL, 2009: 1003-1011.
- [11] ZHANG Yingcheng, YANG Yang, JIANG Rui, et al. Commercial intelligence entity recognition model based on BiLSTM-CRF[J]. Computer Engineering, 2019, 45(5): 308-314. (in Chinese)  
张应成, 杨洋, 蒋瑞, 等. 基于 BiLSTM-CRF 的商情实体识别模型[J]. 计算机工程, 2019, 45(5): 308-314.
- [12] MIWA M, BANSAL M. End-to-end relation extraction using LSTMs on sequences and tree structures [C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, USA: ACL, 2016: 1105-1116.
- [13] ZHENG G, MUKHERJEE S, DONG X L, et al. Opentag: open attribute value extraction from product profiles [C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, USA: ACM Press, 2018: 1049-1058.
- [14] YU A W, DOHAN D, LUONG M T, et al. Qanet: combining local convolution with global self-attention for reading comprehension [EB/OL]. [2020-02-08]. <https://openreview.net/forum?id=B14TIG-RW>.
- [15] SEO M, KEMBHAVI A, FARHADI A, et al. Bidirectional attention flow for machine comprehension [EB/OL]. [2020-02-08]. <https://openreview.net/forum?id=HJ0UKP9ge>.
- [16] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]// Proceedings of NIPS'13. New York, USA: ACM Press, 2013: 3111-3119.
- [17] PENNINGTON J, SOCHER R, MANNING C D. GloVe: Global vectors for word representation [C]// Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, USA: ACL, 2014: 1532-1543.
- [18] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C]// Proceedings of 2019 Conference of the 9th American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, USA: ACL, 2019: 4171-4186.
- [19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]// Proceedings of NIPS'13. New York, USA: ACM Press, 2017: 5998-6008.
- [20] KAISER L, GOMEZ A N, CHOLLET F. Depthwise separable convolutions for neural machine translation [EB/OL]. [2020-02-08]. <https://openreview.net/forum?id=S1jBcucAb>.
- [21] BA J L, KIROS J R, HINTON G E. Layer normalization [EB/OL]. [2020-02-28]. <https://arxiv.org/abs/1607.06450>.
- [22] KONDREDDI S K, TRIANTAFILLOU P, WEIKUM G. HIGGINS: knowledge acquisition meets the crowds [C]// Proceedings of the 22nd International Conference on World Wide Web. New York, USA: ACM Press, 2013: 85-86.
- [23] LI Qi, JIANG Meng, ZHANG Xikun, et al. Truepie: discovering reliable patterns in pattern-based information extraction [C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, USA: ACM Press, 2018: 1675-1684.
- [24] LI Peng, LI Wei, HE Zhengyan, et al. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering [EB/OL]. [2020-02-08]. <https://arxiv.org/pdf/1607.06275.pdf>.

编辑 索书志

(上接第96页)

- [17] LING Huilinag, WU Jiansheng, ZHOU Yi, et al. How many clusters? A robust pso-based local density model [J]. Neurocomputing, 2016, 207: 264-275.
- [18] CHENG Weiqing, LU Yanhong. Adaptive clustering algorithm based on maximum and minimum distances and SSE [J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2015, 35(2): 102-107. (in Chinese)  
成卫青, 卢艳红. 一种基于最大最小距离和 SSE 的自适应聚类算法 [J]. 南京邮电大学学报(自然科学版), 2015, 35(2): 102-107.
- [19] FREY B J, DUECK D. Response to comment on "clustering by passing messages between data points" [J]. Science, 2008, 319(5864): 726-726.
- [20] WANG Kaijun, LI Jian, ZHANG Junying, et al. Semi-supervised affinity propagation clustering [J]. Computer Engineering, 2007, 33(23): 197-198, 201. (in Chinese)  
王开军, 李健, 张军英, 等. 半监督的仿射传播聚类 [J]. 计算机工程, 2007, 33(23): 197-198, 201.
- [21] SUN Jixiang. Modern pattern recognition [M]. Changsha: National University of Defense Technology, 2002. (in Chinese)
- 孙即祥. 现代模式识别 [M]. 长沙: 国防科技大学出版社, 2002.
- [22] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.
- [23] WU T F, TSAI P S, HU N T, et al. Combining turning point detection and Dijkstra's algorithm to search the shortest path [J]. Advances in Mechanical Engineering, 2017, 9(2): 1-12.
- [24] MACQUEEN J. Some methods for classification and analysis of multivariate observations [C]// Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, USA: University of California Press, 1967: 281-297.
- [25] SHANG Fanhua, JIAO Licheng, SHI Jiarong, et al. Fast affinity propagation clustering: a multilevel approach [J]. Pattern Recognition, 2012, 45(1): 474-486.
- [26] VINH N X, EPPS J, BAILEY J. Bibliometrics: information theoretic measures for clusterings comparison [C]// Proceedings of International Conference on Machine Learning. New York, USA: ACM Press, 2010: 2837-2854.

编辑 金胡考