计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(1):129−134 [doi: 10.15888/j.cnki.csa.007694]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于成对约束的 SubKMeans 聚类数确定算法①

高　波, 何振峰

(福州大学 数学与计算机科学学院, 福州 350108)
通讯作者: 高　波, E-mail: 1020792578@qq.com

摘　要: 随着数据维度的增加, 传统聚类算法会出现聚类性能差的现象. SubKMeans 是一种功能强大的子空间聚类算法, 旨在为 K-Means 类算法搜索出一个最佳子空间, 降低高维度影响, 但是该算法需要用户事先指定聚类数目 K 值, 而在实际使用中有时无法给出准确的 K 值. 针对这一问题, 引入成对约束, 将成对约束与轮廓系数进行结合, 提出了一种基于成对约束的 SubKMeans 聚类数确定算法. 改进后的轮廓系数能够更加准确的评价聚类性能, 从而实现 K 值确定, 实验结果证明该方法的有效性.

关键词: 子空间聚类; 聚类数; 成对约束; 轮廓系数

引用格式: 高波,何振峰.基于成对约束的 SubKMeans 聚类数确定算法.计算机系统应用,2021,30(1):129–134. http://www.c-s-a.org.cn/1003-3254/7694.html

SubKMeans Algorithm for Determining Number of Clusters Based on Pairwise Constraints

GAO Bo, HE Zhen-Feng

(School of Mathematics and Computing Science, Fuzhou University, Fuzhou 350108, China)

Abstract: With the increase of data dimension, the traditional clustering algorithm will have poor clustering performance. SubKMeans is a powerful subspace clustering algorithm, which aims to search the best subspace for K-Means algorithm and reduce the impact of high dimensions. However, the algorithm requires users to specify the number of clusters K value in advance, and sometimes it can not give accurate K value in actual use. In order to solve this problem, the pairwise constraint is introduced, which is combined with the silhouette coefficient. A SubKMeans algorithm for determining the number of clusters based on the pairwise constraint is proposed. The improved silhouette coefficient can evaluate the clustering performance more accurately, so that the K value can be determined. The experimental results proves the effectiveness of the proposed method.

Key words: subspace clustering; number of clusters; pairwise constraints; silhouette coefficient

聚类是一种无监督学习方法, 它根据样本间相似度把样本划分到若干簇[1]. K-Means 算法是聚类算法的一种典型代表, 它因其简单而又有效的特性备受欢迎, 并且在十大经典数据挖掘算法中排名第二[2]. 该算法根据用户指定的 K 值, 基于某种距离度量方式, 把样本划分为 K 个不同的簇, 使得簇内样本相似性高, 簇间样本相似性低[1]. 高维数据空间中数据分布稀疏且存在着大量无关属性, 数据的重要结构信息会隐藏在海量的噪声数据中, 因此使用 K-Means 算法在高维数据上进行聚类很难发现数据的内在结构, 使得聚类效果差 [3,4]. 然而在现实的聚类分析应用场景中, 数据维度通常很高, 比如图片视频或文本数据, 其维度一般为千万级, 甚至更高. 针对这一问题, Mautz 等人于 2017 年提出了 SubKMeans 算法[5], 该算法能够将数据映射到子空

① 基金项目: 福建省自然科学基金 (2018J01794)
Foundation item: Natural Science Foundation of Fujian Province (2018J01794)
收稿时间: 2020-04-09; 修改时间: 2020-05-10; 采用时间: 2020-05-28; csa 在线出版时间: 2020-12-31

间中进行聚类, 降低维度影响, 提升 K-Means 类算法聚类性能.

SubKMeans 算法将数据空间划分为一个包含有大部分重要信息的子空间和一个基本不包含重要信息的子空间, 通过映射矩阵能够把数据投影到包含有大部分重要信息的子空间中进行聚类, 从一定程度上减轻“维度灾难”对 K-Means 类算法的影响. 但是 SubKMeans 算法只是对经典 K-Means 类算法的一种扩展, 它依然会受到 K-Means 类算法固有缺陷的限制[6]. SubKMeans 算法是无监督聚类算法, 需要用户事先指定 K 值, 而在现实中部分数据集的种类数是未知的, 这给使用者带来巨大的困扰, 因此 SubKMeans 算法的聚类数确定研究具有重要的现实意义.

现有的子空间聚类算法可以划分为硬子空间聚类和软子空间聚类[7,8]. 硬子空间聚类把所有属性都看成同等重要, 按照搜索子空间方式的不同, 可以进一步划分为自底向上的子空间聚类算法和自顶向下的子空间聚类算法. 软子空间聚类算法认为每个属性对于每个簇的贡献程度不一样, 因此给每个属性赋予不同的权重. 文献 [9] 和文献 [10] 中提出基于惩罚机制的竞争学习来逐步合并聚类簇, 消除冗余聚类, 最后为子空间聚类确定聚类数目. 文献 [11] 基于类内紧凑性和类间分离性提出了一种新的聚类有效性指标, 通过在 K 的取值范围内得出最佳指标值来为子空间聚类确定 K 值. 文献 [12] 把现有的 K 值确定方法分为 3 类, 分别为传统方法、基于合并分裂方法和基于进化的方法. 传统方法把最佳聚类有效性指标对应的 K 值作为最佳 K 值. 基于合并分裂的方法根据聚类有效性指标的值是否更优来决定是否合并或分裂, 达到稳定时的 K 值即为最佳 K 值. 基于进化的方法使用特定的编码方式将可能的划分方式编码到个体或染色体中, 通过遗传变异的方式得到适应性最好的个体, 把该个体对应的 K 值作为最终 K 值.

为解决 SubKMeans 聚类数确定问题, 考虑到现实中有时能获取到类似成对约束之类的监督信息, 参考文献 [13] 中成对约束与轮廓系数的结合方法, 用成对约束改变轮廓系数计算方式, 并用成对约束的满足度给轮廓系数加权. 将改进后的轮廓系数作为聚类有效性评价指标, 通过尝试不同的 K 值来找到一个最佳指标值, 把该最佳指标值对应的 K 值作为最佳 K 值. 第 1 节介绍 SubKMeans 算法, 第 2 节介绍改进的 SubKMeans 算法, 第 3 节对改进算法进行实验并分析实验结果, 第 4 节对所做工作进行总结.

1 SubKMeans 算法简介

SubKMeans 算法又称子空间 K 均值算法, 它通过寻找数据的最佳子空间来发现数据的隐藏结构, 降低维度影响, 使得 K-Means 类算法在高维数据上也能够有不错的表现[5]. 它的主要思想是: 假设在一个数据集中大部分重要的信息会隐藏在某一个维度更低的子空间中, 而其它的子空间能够提供的有用信息很少. 根据这一假设把数据空间划分为两个子空间, 包含大部分重要信息的子空间称为聚类子空间, 基本不包含重要信息的子空间称为噪声子空间[5]. 为了提高聚类性能, 挖掘出数据的内在结构, 需要把数据映射到聚类子空间上进行聚类.

给定数据集 $D=\{x_1,x_2,\cdots,x_n\}\in R^{d\times n}$, 其中 n 是数据集 D 的规模, d 是样本的维度. 假设要把数据聚为 K 个簇 $\{C_i\}_{i=1}^{K}$, 在经典 K-Means 算法中, 最优化目标是使得每个样本到其聚类中心点的距离总和最小[1,5], 即优化下式:

$$\sum_{i=1}^{K}\sum_{x\in C_i}\|x-u_i\|^2 \tag{1}$$

其中, u_i 为第 i 个簇的簇中心, $\|\cdot\|$ 表示欧几里得范数. SubKMeans 算法需要将样本映射到聚类子空间中进行聚类, 两个样本在聚类子空间中的距离可以通过式 (2) 计算.

$$dist(x_i,x_j)=\left\|P_C^{\mathrm{T}}V^{\mathrm{T}}x_i-P_C^{\mathrm{T}}V^{\mathrm{T}}x_j\right\|^2 \tag{2}$$

其中, $P_C\in R^{m\times d}$, m 为聚类子空间维度且 $m<d$, $V\in R^{d\times d}$ 是一个维度为 d 的正交矩阵. 通过 $P_C^{\mathrm{T}}V^{\mathrm{T}}x$ 能够将样本 x 映射到聚类子空间中, P_C 定义为:

$$P_C=\begin{bmatrix} I_m \\ O_{d-m,m}\end{bmatrix} \tag{3}$$

其中, I_m 是维度为 m 的单位矩阵, $O_{d-m,m}\in R^{m\times(d-m)}$ 为零矩阵. 重新定义样本间距离计算公式后, SubKMeans 优化目标可以表示为:

$$\sum_{i=1}^{K}\sum_{x\in C_i}\left\|P_C^{\mathrm{T}}V^{\mathrm{T}}x-P_C^{\mathrm{T}}V^{\mathrm{T}}u_i\right\|^2+\sum_{x\in D}\left\|P_N^{\mathrm{T}}V^{\mathrm{T}}x-P_N^{\mathrm{T}}V^{\mathrm{T}}u_D\right\|^2 \tag{4}$$

其中, $P_N\in R^{(d-m)\times d}$, $(d-m)$ 为噪声子空间维度, $u_D\in R^{d\times 1}$ 为数据集 D 的列均值. 将式 (4) 展开, 利用矩阵迹的特

性, 可以表示为:

$$Tr\left(P_C P_C^{\mathrm{T}} V^{\mathrm{T}} S_{iD} V\right) + Tr\left(V^{\mathrm{T}} S_D V\right) \tag{5}$$

$$S_{iD} = \left[\sum_{i=1}^{K} S_i\right] - S_D \tag{6}$$

$$S_i = \sum_{x \in C_i} (x - u_i)(x - u_i)^{\mathrm{T}} \tag{7}$$

$$S_D = \sum_{x \in D} (x - u_D)(x - u_D)^{\mathrm{T}} \tag{8}$$

其中, Tr 表示矩阵的迹, V 是一个正交矩阵, 根据正交矩阵的特性, 可知$V^{\mathrm{T}} S_D V$相乘后, 只改变矩阵S_D特征向量的方向, 不改变其特征值本身. 因此对于任意的正交矩阵 V, $V^{\mathrm{T}} S_D V$的特征值是常量. 矩阵的迹是其所有特征值之和, 所以$Tr\left(V^{\mathrm{T}} S_D V\right)$是一个常量, 在式 (5) 中可以忽略. 令矩阵 V 为S_{iD}特征分解后的特征向量, 并且这些特征向量按照特征值的大小进行升序排序, 最小的 m 个特征值对应的特征向量将数据映射到聚类子空间中, 其它 $(d-m)$ 个特征值对应的特征向量将数据映射到噪声子空间中, 令 m 为S_{iD}特征分解后特征值中小于 0 的个数, 可解决 (4) 的最优化问题. 使用式 (2) 计算样本距离, 不断迭代更新簇中心, 更新矩阵 V 和子空间维度 m, 算法最终趋于稳定得到固定维度的聚类子空间和聚类簇. SubKMeans 算法框架如算法 1 所示.

算法 1. SubKMeans 算法

输入: 数据集 D, 聚类数量 K

输出: 聚类簇$\{C_1, C_2, \cdots, C_K\}$, 正交变换矩 V, 聚类子空间维度 m

1) 初始化聚类子空间维度$m=\lfloor d/2 \rfloor$ //$\lfloor\rfloor$表示向下取整
2) 计算数据集列平均u_D
3) 采用式 (8) 计算数据集的散列矩阵S_D
4) 随机产生初始聚类中心u_i, $i=1,2,\cdots,K$
5) 随机矩阵执行 QR 分解产生正交矩阵 V
6) While(簇中心改变)
7) for each $x \in D$
8) 采用式 (2) 计算样本到簇中心的距离
9) 将样本划分到距离最近的簇
10) end for
11) 更新簇中心u_i
12) 采用式 (7) 计算簇的散列矩阵S_i
13) 更新矩阵$V, \varepsilon = eig(S_{iD})$// eig 表示特征分解, V 为特征分解后的特征向量, ε为特征值
14) 更新维度$m=|\{e|e \in \varepsilon, e<0\}|$ //|| 表示取集合中元素个数
15) end while

虽然 SubKMeans 算法能够自动确定聚类子空间维度, 但需要用户指定聚类数量 K. 聚类数的确定是实际应用中的一个重大问题, 因为在实际的应用场景中, 需要聚类的数据往往是未知数据, 我们不知道哪些数据应该分配到同一类中, 对于给出的 K 值, 我们也无法验证其是否是当前数据的准确 K 值.

2 基于成对约束的 SubKMeans 聚类数确定算法

轮廓系数是一种常用的聚类有效性指标, 可用于确定 K 值. 在轮廓系数的计算方式中, 聚类的轮廓系数为数据集中所有样本的轮廓系数的平均值, 其把每个样本看成同等重要, 把该指标作为聚类有效性指标用于确定聚类数量时, 往往效果不好. 而在实际的聚类过程中, 存在部分样本对簇的贡献程度不一样的情况. 为了体现这种差异, 基于文献 [13], 本文引入成对约束, 用轮廓系数的满足度给单个样本和整个聚类进行加权, 并将违反的成对约束作为惩罚项, 改进轮廓系数的计算方式, 为 SubKMeans 算法提出一种成对约束与轮廓系数结合的 K 值确定方法, 称为 Constrained Weighted SubKMeans, 简称 CSWKM. CSWKM 算法把改进后的轮廓系数作为一种新的聚类有效性指标, 在 K 的取值范围内, 计算出各个 K 值时的指标值, 把最佳指标值对应的 K 值作为最佳 K 值. CSWKM 算法框架如下算法 2 所示.

算法 2. CSWKM 算法

输入: 数据集 D, 成对约束 Cst, 最大迭代次数 $Count$

输出: 聚类簇$\{C_1, C_2, \cdots, C_K\}$, 正交变换矩 V, 聚类子空间维度 m, 聚类数量 K

1) for $K=K_{\min}$ to $K_{\max}$
2) *SubKMeans* 算法 //迭代时需判断迭代次数是否超过限制
3) if (簇迭代次数小于 *Count*)
4) 采用式 (13) 计算出此次划分的轮廓系数
5) if(计算得出的轮廓系数小于 0)
6) 令轮廓系数为 0
7) else
8) 令此次划分的轮廓系数为 0
9) end for

CSWKM 需要分别计算出各个 K 值时的轮廓系数值, 把最大轮廓系数对应的 K 值作为最终 K 值. 在计算单个 K 值的轮廓系数时, 需要迭代更新簇中心点、更新矩阵 V 和子空间维度 m, 同时在进行迭代时需要先判断当前迭代次数是否超过最大迭代次数, 若超过, 则停止迭代. $K_{\min}$一般取 2, $K_{\max}$根据经验为样本数量

的平方根取整, 算法输出部分中, 簇$\{C_1,C_2,\cdots,C_K\}$、V 和 m 对应于最佳 K 值的簇、V 和 m. 与 SubKMeans 算法相比, CSWKM 算法对簇的迭代次数进行了限制, 计算了每次簇划分后对应的轮廓系数值.

2.1 簇迭代次数限制

CSWKM 算法不同于 SubKMeans 算法, CSWKM 算法需要尝试 K 值范围内的每个 K 值. 由于 CSWKM 算法中对簇的个数进行了限制, 强制每个簇里面的样本个数必须大于 5, 在实验中发现当给出的 K 值与实际的 K 值相差较大时, 会出现划分簇的迭代次数过多或者不收敛的现象. 为了解决这一问题, 给簇的迭代加上次数限制, 使得超过迭代次数的 K 值划分认为是不合适的划分, 直接令此次 K 值划分的簇轮廓系数为 0, 一般情况下令迭代次数为 50.

2.2 轮廓系数

轮廓系数是目前使用最为频繁的聚类有效性评价指标之一, 其要求同一个簇内样本间距离小, 相似性高, 不同簇间距离大, 相似性低[13,14]. 聚类的轮廓系数为数据集中所有样本的轮廓系数平均值, 单个样本 x 的轮廓系数计算公式如式 (9) 所示:

$$Si(x)=\frac{b(x)-a(x)}{\max(a(x),b(x))} \tag{9}$$

其中, $a(x)$表示样本 x 与其所属簇的其他样本之间的平均距离, 为类内距离, $b(x)$表示样本 x 到其他簇的平均距离中的最小值, 为类间距离.

单独使用轮廓系数作为聚类有效性评价指标效果并不理想, 基于样本对簇的贡献程度不同, 本文引入监督信息对轮廓系数进行改进. 监督信息可以分为两类, 一类是数据样本类别标签, 另一类是数据样本之间的成对约束信息. 成对约束一般是指 *must-link* 与 *cannot-link* 两种关联约束关系, 正关联约束关系 *must-link*(x,y) 表示样本 x 和样本 y 属于同一类, 负关联约束关系 *cannot-link*(x,y) 表示样本 x 和样本 y 属于不同类. 由于成对约束信息获取成本低, 容易得到, 因此本文使用的监督信息为成对约束. 为了体现出各个样本对簇的贡献大小, 我们认为成对约束满足程度高的样本对簇的贡献程度更大, 应该赋予更高的权重. 但是当两个样本成对约束满足程度一致时, 其对簇的贡献程度也可能不一样. 文献 [15] 认为不同的成对约束的包含的信息不一样, 应该区分对待. 因此我们把未得到满足的成对约束之间的平均距离作为一个惩罚项, 用来体现当成对约束满足程度一致时, 样本对簇的贡献程度.

在 *must-link* 约束关系中, 距离更大的约束包含的信息更多, 违反后应该受到更大惩罚, 应使得其轮廓系数更小. 根据轮廓系数计算方式, 通常类内距离越大轮廓系数越小. 在不考虑权重的情况下, 对同一个样本来说, 违反约束后, 其轮廓系数值应该更小, 因此改进后的类内距离不应该比原先的类内距离小. 所以令改进后的类内距离为$a(x)$与惩罚项两者中的最大值[13], 如式 (10) 所示, $a(x)$表示为改进时的类内距离.

$$a(x)'=\max(a(x),avg(x,x_{ML})) \tag{10}$$

其中, x_{ML}表示与样本 x 具有正关联约束关系但在实际划分簇的过程中没有划分到同一个簇的样本集合, $avg(x,x_{ML})$表示样本 x 到集合x_{ML}的平均距离.

在 *cannot-link* 约束关系中, 距离更小的约束包含的信息更多, 违反后应该受到更大惩罚. 根据轮廓系数计算方式, 一般类间距离越小轮廓系数越小, 同理, 应该使得改进后的类间距离为$b(x)$与惩罚项两者中的最小值[13], 如式 (11) 所示, $b(x)$表示未改进前的类间距离.

$$b(x)'=\min(b(x),avg(x,x_{CL})) \tag{11}$$

其中, x_{CL}表示与样本 x 具有负关联约束关系但在实际划分簇的过程中划分到同一个簇的样本集合, $avg(x,x_{CL})$表示样本 x 到集合x_{CL}的平均距离.

改进后的单个样本轮廓系数如式 (12) 所示. 此时可能会出现轮廓系数为负数的情况, 而轮廓系数不为负数, 因此令小于 0 的轮廓系数为 0.

$$Si(x)'=\frac{b(x)'-a(x)'}{\max(a(x)',b(x)')} \tag{12}$$

加权的方式分为划分权重与样本权重. 划分权重是从整个聚类划分的角度出发, 为在此次 K 值划分中满足的约束关系个数占总约束关系个数的比例. 样本权重是从单个样本的角度出发, 若样本 x 具有约束关系, 则其样本权重为样本 x 满足的约束关系个数占样本 x 总约束关系个数的比例. 若样本 x 没有约束关系但其所在的簇里面其它样本具有约束关系, 那么其样本权重为簇中满足的约束关系个数占簇中总约束关系个数的比例. 若样本 x 本身没有约束关系并且其所在的簇中其它样本也没有约束关系, 那么其样本权重为 1.

把划分权重与样本权重结合起来, 聚类的轮廓系数计算公式如式 (13) 所示,其中$SI(D)$表示聚类轮廓系数, $Si(x)'$为单个样本 x 的轮廓系数, $w(x)$为样本权重,

|D|为数据集 D 中的样本个数, $weight$ 为划分权重.

$$SI(D)=\frac{\sum_{x\in D} w(x)Si(x)'}{|D|}weight \tag{13}$$

3 实验与分析

实验阶段使用 6 个 UCI 数据集和 1 个 UCR 数据集, 如表 1 所示. Wdbc、Seeds、Iris、Wine、Vertebral column、Glass Identification、Breast Tissue 来自于 UCI 数据集, Plane 来自于 UCR 数据集, Wdbc 表示的是 Breast Cancer Wisconsin (Diagnostic) 数据集. 每组数据都采用了标准化 (将一组数的每个数都减去这组数的平均值后再除以这组数的均方差) 的预处理方式, 采用结合成对约束的轮廓系数作为聚类有效性评价指标, 聚类准确性使用标准互信息 (NMI).

表 2 是 CSWKM 算法对比实验的结果, 在 CSWKM 算法对比实验中, 迭代次数 $Count$ 取 50, 聚类数量 K 的最大取值范围为$\sqrt{n}$向下取整, n 表示数据集的规模, Pre_K 表示实验重复 100 次时, 算法选出的最佳聚类数与原数据集中种类数一致的次数, “无”表示算法迭代 10 000 次后未收敛, 括号中的数字为成对约束的对数, NMI 的值为 10 次十折交叉验证的平均值. 把仅仅使用轮廓系数而不加成对约束作为聚类有效性评价指标, 用来为 SubKMeans 确定 K 值的算法称为 SIKM.

表 1 数据集相关信息

数据集	样本	属性	类数
Wdbc	569	31	2
Seeds	210	7	3
Iris	150	4	3
Wine	178	13	3
Vertebral Column (Column)	310	6	3
Glass Identification (Glass)	214	9	6
Breast Tissue (BT)	106	9	6
Plane	210	144	7

表 2 CSWKM、SIKM 和 SubKMeans 算法对比

数据集	CSWKM			SIKM			SubKMeans
	NMI(10)	Pre_K(10)	NMI(100)	Pre_K(100)	NMI	Pre_K	NMI
Wdbc	0.548	84	0.568	100	0.565	99	0.566
Seeds	0.724	74	0.785	100	0.596	0	0.785
Iris	0.713	17	0.700	62	0.737	0	0.685
Wine	0.890	78	0.915	92	0.729	26	0.915
Column	0.366	35	0.362	23	0.367	0	0.341
Glass	0.423	10	0.463	27	无	0	0.470
BT	0.650	8	0.693	10	0.448	0	0.678
Plane	0.882	16	0.897	24	0.775	6	0.879

从表 2 中 CSWKM 与 SIKM 算法的对比实验数据中可以明显看到 CSWKM 算法的 K 值确定准确率不论在成对约束对数为 10 或 100 时, 均要高于 SIKM 算法, 预测 K 值更加精准, 使得 NMI 系数也要高于 SIKM 算法. 这一结果表明结合成对约束后的轮廓系数更能够表示聚类性能, 验证了 CSWKM 算法在确定 K 值上的有效性. 在 Glass 数据集上, 由于有一类只有 9 个样本, 在进行十折交叉验证的时候会出现有的簇中无法满足样本数大于 5 的要求, 导致不收敛, 而 CSWKM 算法对簇的迭代次数进行了限制, 因此不会出现不收敛的现象. 从 10 对成对约束与 100 对成对约束的实验结果中可以看到, CSWKM 算法的 NMI 系数随着预测 K 值准确率的提高而提升, 由于在大多数的数据集中预测的 K 值准确率不能达到百分百, 因而 NMI 系数普遍要比 SubKMeans 算法低. 当 K 值预测准确率达到百分百时, CSWKM 算法的 NMI 系数不低于 SubKMeans 算法, 可以从 Wdbc 和 Seeds 数据集的实验结果中看出. 在部分数据集上, 可能聚为其它簇的效果要更好, 因而预测 K 值准确率虽然没有达到百分百, 但是 CSWKM 算法的 NMI 系数还是要高于 SubKMeans 算法.

4 总结与展望

针对 SubKMeans 算法需要用户指定 K 值的问题, 提出了一种基于成对约束的 SubKMeans 聚类数确定算法. 将成对约束运用到轮廓系数中, 首先用成对约束改进轮廓系数的计算方式, 其次用成对约束的满足程度给轮廓系数加权, 将改进后的轮廓系数作为聚类有效性评价指标, 在 K 的取值范围内根据最佳指标值挑选出对应的最佳 K 值, 有效的解决了 SubKMeans 算法在确定聚类数量方面的难题. 最后, 通过在 UCI 和 UCR

数据集上进行实验, 对比没有使用成对约束改进轮廓系数的 SIKM 算法和 SubKMeans 算法. 实验结果表明, CSWKM 算法的 K 值确定准确率和聚类效果优于 SIKM 算法, 验证了 CSWKM 算法的有效性. 并且 CSWKM 算法在给出 100 对成对约束时, 聚类效果优于 SubKMeans 算法. 未来的工作将致力于如何把子空间信息作为确定 K 值的一个考虑因素.

参考文献

1 Saxena A, Prasad M, Gupta A, *et al*. A review of clustering techniques and developments. Neurocomputing, 2017, 267: 664–681. [doi: 10.1016/j.neucom.2017.06.053]

2 Wu XD, Kumar V, Quinlan JR, *et al*. Top 10 algorithms in data mining. Knowledge and Information Systems, 2008, 14(1): 1–37. [doi: 10.1007/s10115-007-0114-2]

3 Fränti P, Sieranoja S. K-means properties on six clustering benchmark datasets. Applied Intelligence, 2018, 48(12): 4743–4759. [doi: 10.1007/s10489-018-1238-7]

4 Mautz D, Ye W, Plant C, *et al*. Discovering non-redundant K-means clusterings in optimal subspaces. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Kassel, Germany. 2018. 1973–1982.

5 Mautz D, Ye W, Plant C, *et al*. Towards an optimal subspace for K-Means. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Halifax, NS, Canada. 2017. 365–373.

6 Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 2010, 31(8): 651–666. [doi: 10.1016/j.patrec.2009.09.011]

7 Harikumar S, Akhil AS. Semi supervised approach towards subspace clustering. Journal of Intelligent & Fuzzy Systems, 2018, 34(3): 1619–1629.

8 Deng ZH, Choi KS, Jiang YZ, *et al*. A survey on soft subspace clustering. Information Sciences, 2016, 348: 84–106. [doi: 10.1016/j.ins.2016.01.101]

9 Jing LP, Li JJ, Ng MK, *et al*. SMART: A subspace clustering algorithm that automatically identifies the appropriate number of clusters. International Journal of Data Mining, Modelling and Management, 2009, 1(2): 149–177. [doi: 10.1504/IJDMMM.2009.026074]

10 Jia H, Cheung YM. Subspace clustering of categorical and numerical data with an unknown number of clusters. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(8): 3308–3325. [doi: 10.1109/TNNLS.l2017.2728138]

11 Chen LF, Wang SR, Wang KJ, *et al*. Soft subspace clustering of categorical data with probabilistic distance. Pattern Recognition, 2016, 51: 322–332. [doi: 10.1016/j.patcog.2015.09.027]

12 Hancer E, Karaboga D. A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. Swarm and Evolutionary Computation, 2017, 32: 49–67. [doi: 10.1016/j.swevo.2016.06.004]

13 He ZF. Evolutionary K-Means with pair-wise constraints. Soft Computing, 2016, 20(1): 287–301. [doi: 10.1007/s00500-014-1503-6]

14 Starczewski A, Krzyżak A. A modification of the silhouette index for the improvement of cluster validity assessment. Proceedings of the 15th International Conference on Artificial Intelligence and Soft Computing. Zakopane, Poland. 2016. 114–124.

15 Yu ZW, Luo PN, Liu JM, *et al*. Semi-supervised ensemble clustering based on selected constraint projection. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(12): 2394–2407. [doi: 10.1109/TKDE.2018.2818729]