Advances in Applied Mathematics 应用数学进展, 2024, 13(9), 4381-4390
Published Online September 2024 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2024.139418

Hans汉斯

基于空间平移的K-Means初始簇心选取

朱家乐

长安大学理学院，陕西 西安

收稿日期：2024年8月23日；录用日期：2024年9月17日；发布日期：2024年9月24日

摘　要

K-means聚类算法因其算法简单、计算效率高，在机器学习、数据挖掘等多个领域得到了广泛应用。然而，传统K-means算法在初始簇心的选取上存在随机性，这可能导致聚类结果的不稳定性。为了解决这一问题，本研究提出了一种基于空间平移的初始簇心选取算法。该算法首先将包含所有样本集的最小空间通过单位空间以一定步长遍历，在单位空间内统计样本点的密度，以此降低计算量。通过逐一选出密度最高的个点作为初始簇心，从而提高了K-means算法的聚类性能。在UCI的12种数据集上进行的实验表明，与传统的K-means、K-means++等算法相比，改进的算法在迭代次数上有所降低，聚类准确率得到了显著提高。

关键词

K-Means，初始聚类中心，密度，空间平移

K-Means Initial Cluster Center Selection Based on Spatial Translation

Jiale Zhu

School of Science, Chang'an University, Xi'an Shaanxi

Received: Aug. 23rd, 2024; accepted: Sep. 17th, 2024; published: Sep. 24th, 2024

Abstract

K-means clustering algorithm is an important content in the field of machine learning and is widely used because of its simplicity and efficiency. In order to solve the problem that the initial cluster center selection of traditional K-means algorithm is random, an initial cluster center selection algorithm based on space segmentation is proposed. The minimum space containing all sample sets is divided to calculate the density, and the initial cluster centers with the highest density are selected one by one. The selected cluster centers are replaced by random initial cluster centers for K-

文章引用: 朱家乐. 基于空间平移的 K-Means 初始簇心选取[J]. 应用数学进展, 2024, 13(9): 4381-4390.
DOI: 10.12677/aam.2024.139418

means clustering. Twelve datasets were tested separately at UCI. The experimental results show that compared with traditional K-means, K-means++ and other algorithms, the improved algorithm has lower iteration times and higher clustering accuracy.

Keywords

K-Means, Initial Cluster Center, Density, Spatial Translation

1. 引言

聚类试图将数据集的样本划分为若干个通常是不相交的子集，每个子集称为一个“簇”[1]。Han [2]等人按不同性质将聚类划分为五类：基于划分、基于分层、基于密度、基于网格以及基于模型的聚类。聚类分析作为数据挖掘和机器学习中的重要技术，广泛应用于医疗[3]、能源[4]、交通[5]、金融[6]、图像处理[7]。

K-means 作为最简单高效的聚类算法之一，是学者 MacQueen [8]在 1967 年提出的，虽然 K-means 聚类算法简单高效，但也存在着一些不足，如 k 值需要事先确定、算法对离群值较为敏感、聚类效果对初始簇心较为敏感。针对这些问题，许多学者也提出了相应的改进。

对于 k 值的确定，Rezaee [9]等人提出 $k \in \left[2, \sqrt{n}\right]$，其中 n 为样本个数；成卫青[10]等人根据平方误差(SSE)和 k 值的关系提出一种自适应确定 k 值的算法；王建仁[11]等人针对手肘法在确定 k 值的过程中存在的“肘点”位置不明确问题，提出了一种改进的 k 值选择算法 ET-SSE 算法，提升了寻找 k 值的效率；Kristina [12]等人提出了一种新颖的无监督 K 均值(UK 均值)聚类算法，可以自动找到最佳的聚类数量，而无须进行任何初始化和参数选择；何选森[13]等人定义了新的聚类有效性评价指标，k 值就是使得此指标达到最小的整数值。

在筛除离群点方面，唐东凯[14]等人根据 LOF [15] (Local Outlier Factor)算法计算每个点的可达密度来判断样本点是否为离群点；朱利[16]等人设计了一种用于连接数据点间的信息数据结构，且开发了一个新的离群因子计算公式，实现了对簇类数量的自适应评估，从而能够高效地识别出离群点。刘凤[17]等人通过局部密度筛选离群点，剔除离群点后进行聚类。

为了获得更科学的初始簇心，张玉芳[18]等人事先设定簇数 $k' > k$，选择合适的样本集大小，采取 J 次取样并进行 K-means 聚类，将平方误差 SSE 最小的聚类结果的簇心作为初始聚类簇心并聚类，再将最近的两个类合并直到簇个数为 k；袁方[19]等人先设置一个类簇所含样本点个数的阈值，将位置最近的一些点归为一类直到该簇的样本点个数达到阈值，再从这些点之外的样本点重复此操作直至选出 k 个类，将这 k 个簇的质心作为初始簇心；赖玉霞[20]等人先选出密度较大的样本点，在这些点中选出相互相距最远的 k 个初始簇心；汪中[21]等人先计算各点密度，选出密度最大的点作为第一个初始点将其周围的点剔除，再选出剩下点中密度最大的点作为下一个初始点，直至找出 k 个初始点；陈光平[22]等人首先将数据集最远的两个点作为初始簇心，将其余点归为最近的簇，在簇最多的点里继续找相距最远的两个点作为新的初始簇心，直到找到 k 个初始簇心；谢娟英[23]等人首先定义样本点的方差，在 k 个不同区域选出方差最小的样本点作为初始簇心；郭永坤[24]等人给出点到簇的距离定义和类簇内样本点个数阈值 α，将部分样本点聚成 k 个簇，将每个簇的簇心作为初始簇心。

大多数选取初始簇心的文献都是基于密度来选取的，虽然这些文献对 K-means 初始簇心的优化有所

改进，但却忽略了统计各样本点密度时的庞大计算量。针对此问题，本文提出一种划分空间的算法 SK-means 算法(Space-divided-based K-means)，该算法通过划分空间来降低统计样本点密度时的计算量；另一方面针对文献[25]提出的算法中密度最大样本点不止一个的问题，提出了局部平均距离，选出更为紧凑的样本点，使得聚类效果更优。实验表明，改进的算法降低了迭代次数，提高了聚类结果准确率。

2. K-Means 聚类算法

K-means 算法是一种基于划分的聚类算法，通过在迭代中更新簇及簇心将数据集划分到若干个不相交的簇。该算法原理为：给定类簇数 k，随机选取 k 个样本点作为初始簇心，并将样本点划分到距离最近的簇，在一次划分完毕后更新簇心，继续重复划分和更新簇心直至簇心不再变化或达到最大迭代次数[26]。

定义 1 [1]设 $X=\{x_1,x_2,\cdots,x_n\}$ 为一组维度为 d 的数据集、$x_i=(x_{i1},x_{i2},\cdots,x_{id})$ 为第 i 个样本、k 为聚类簇数，记 $C_1^t,C_2^t,\cdots,C_k^t$ 为第 t 次迭代聚类结果。对任意 $x_i,x_j\in X$，x_i,x_j 间的欧氏距离为

$$dist(x_i,x_j)=\sqrt{\sum_{l=1}^{d}(x_{il}-x_{jl})^2}, \tag{1}$$

第 t 次迭代簇 C_i^t 的簇中心为

$$\mu_i^t=\frac{1}{|C_i^t|}\sum_{x\in C_i^t}x,\quad i=1,2,\cdots,k, \tag{2}$$

其中 $|C_i^t|$ 为簇 C_i^t 的基数，即 C_i^t 中所含样本点的个数。第 t 次迭代聚类结果的平方误差为

$$\mathrm{SSE}=\sum_{i=1}^{k}\sum_{x_j\in C_i^t}(x_j-\mu_i)^2 \tag{3}$$

簇心依据距离度量将一个簇内的样本点平均化，作为一个簇中心；平方误差是评价聚类好坏的最基本的指标，它直观地反映了样本点与其所属簇心之间的距离关系。传统 K-means 聚类算法具体步骤如下：

算法 1 [27]传统 K-means 聚类算法

输入：数据集 $X=\{x_1,x_2,\cdots,x_n\}$，类簇数 k。
输出：聚类的 k 个簇。

输入：数据集 $X=\{x_1,x_2,\cdots,x_n\}$，类簇数 k。
输出：聚类的 k 个簇。

步骤 1 $t=0$，随机选取 k 个样本点 $\mu_1^t,\mu_2^t,\cdots,\mu_k^t$ 作为初始簇心，$C_i^t=\{\mu_i^t\}$，$i=1,2,\cdots,0$；

步骤 2 根据公式(1)计算每个点到簇心的距离，将其划分到距离最近的簇心所在的簇，即 $\forall x\in X-\{\mu_1^t,\mu_2^t,\cdots,\mu_k^t\}$，令 $i=\arg\min_{1\le i\le k}dis(x,\mu_i^t)$，$C_i^t:=C_i^t\cup\{x\}$；

步骤 3 $t:=t+1$，根据公式(2)更新簇心 $\mu_i^t=\frac{1}{|C_i^t|}\sum_{x\in C_i^t}x$，$i=1,2,\cdots,k$，$C_i^t=\varnothing$；

步骤 4 根据公式(1)计算每个点到簇心的距离，将其划分到距离最近的簇心所在的簇，即 $\forall x\in X$，令 $i=\arg\min_{1\le i\le k}dis(x,\mu_i^t)$，$C_i^t:=C_i^t\cup\{x\}$；

步骤 5 重复步骤 3、步骤 4 直至簇心不再发生变化，即 $\mu_i^t=\mu_i^{t-1}$ 时，停止迭代。

3. 基于空间平移的 K-Means 初始点选取算法

由于初始点选取的随机性，若选取到离群点作为初始簇心，则初始簇心距离最终聚类结果簇心较远，

容易陷入局部最优解[28]。文献[20]指出聚类中心应位于簇内密度较高的位置，而根据文献[23] [24] [29]的理论推导和实验分析可知，聚类中心若设置在样本密度较大的区域，则能够显著提高聚类的准确性。以往基于密度选取初始点的文献在计算样本点密度时考虑了所有样本点，为了避免大量计算，快速找到密度较大的样本点，提出了 SK-means 初始点选取算法。

3.1. SK-Means 算法思想

SK-means 算法思想是：首先构造包含所有样本点的最小 d 维多面体，记该多面体体积为V_S，用一个体积为V_S/k (k 为类簇数)的 d 维正多面体从包含所有样本点的最小 d 维多面体的一角出发，以一定步长遍历上述 d 维多面体，在各个较小的正多面体中找出 k 个密度最大的初始点作为初始簇心。

3.1.1. 第一步：构造棱空间

设 $X=\{x_1,x_2,\cdots,x_n\}$ 为一组 d 维数据集，样本 $x_i=(x_{i1},x_{i2},\cdots,x_{id})$，记 $\alpha_i=\inf\{x_{1i},x_{2i},\cdots,x_{ni}\}$，$\beta_i=\sup\{x_{1i},x_{2i},\cdots,x_{ni}\}$。称包含数据集 X 中所有样本的最小 d 维多面体

$$S=[\alpha_1,\beta_1]\times[\alpha_2,\beta_2]\times\cdots\times[\alpha_d,\beta_d]=\overset{d}{\underset{i=1}{\times}}[\alpha_i,\beta_i] \tag{4}$$

为容纳空间，则容纳空间体积$V_S=\prod_{i=1}^{d}(\beta_i-\alpha_i)$。

定义 2 设 $X=\{x_1,x_2,\cdots,x_n\}$ 为一组维度为 d 的数据集，数据集 X 的容纳空间 $S=\overset{d}{\underset{i=1}{\times}}[\alpha_i,\beta_i]$，称 $S(l_1,l_2,\cdots,l_d)=\overset{d}{\underset{i=1}{\times}}[\alpha_i+l_ib,\alpha_i+l_ib+a)$ 是索引指标为$(l_1,l_2,\cdots,l_d)$的棱空间，其中棱长 $a=\sqrt[d]{V_S/k}$，k 为类簇数，步长$b=\dfrac{a}{2}$，$l_i\in\{0,1,2,\cdots,m_i\}$，$m_i=\left[\dfrac{\beta_i-\alpha_i}{b}\right]$。

例 1 考虑一组 2 维数据集 $X=\{x_1,x_2,\cdots,x_{16}\}$ (见表 1)，类簇数 $k=4$。显然 $\alpha_1=\alpha_2=0$，$\beta_1=\beta_2=10$，容纳空间 $S=[0,10]\times[0,10]$ 且 $V_S=100$，边长 $a=\sqrt{V_S/k}=5$，步长 $b=2.5$，所有的棱空间见表 2。

Table 1. The 2-dimensional data set
表 1. 2 维数据集

样本点	样本点	样本点	样本点
$x_1=(0,1)$	$x_5=(9,7)$	$x_9=(4,9)$	$x_{13}=(5,0)$
$x_2=(0,2)$	$x_6=(8,8)$	$x_{10}=(5,9)$	$x_{14}=(6,0)$
$x_3=(1,1)$	$x_7=(10,8)$	$x_{11}=(5,10)$	$x_{15}=(5,1)$
$x_4=(1,2)$	$x_8=(9,9)$	$x_{12}=(6,10)$	$x_{16}=(6,1)$

Table 2. All edge space
表 2. 所有棱空间

棱空间	棱空间	棱空间
$S(0,0)=[0,5)\times[0,5)$	$S(1,4)=[2.5,7.5)\times[10,15)$	$S(3,3)=[7.5,12.5)\times[7.5,12.5)$
$S(0,1)=[0,5)\times[2.5,7.5)$	$S(2,0)=[5,10)\times[0,5)$	$S(3,4)=[7.5,12.5)\times[10,15)$
$S(0,2)=[0,5)\times[5,10)$	$S(2,1)=[5,10)\times[2.5,7.5)$	$S(4,0)=[10,15)\times[0,5)$
$S(0,3)=[0,5)\times[7.5,12.5)$	$S(2,2)=[5,10)\times[5,10)$	$S(4,1)=[10,15)\times[2.5,7.5)$
$S(0,4)=[0,5)\times[10,15)$	$S(2,3)=[5,10)\times[7.5,12.5)$	$S(4,2)=[10,15)\times[5,10)$

续表

$S(1,0)=[2.5,7.5)\times[0,5)$	$S(2,4)=[5,10)\times[10,15)$	$S(4,3)=[10,15)\times[7.5,12.5)$
$S(1,1)=[2.5,7.5)\times[2.5,7.5)$	$S(3,0)=[7.5,12.5)\times[0,5)$	$S(4,4)=[10,15)\times[10,15)$
$S(1,2)=[2.5,7.5)\times[5,10)$	$S(3,1)=[7.5,12.5)\times[2.5,7.5)$	
$S(1,3)=[2.5,7.5)\times[7.5,12.5)$	$S(3,2)=[7.5,12.5)\times[5,10)$	

3.1.2. 第二步：计算棱空间中样本点密度

由于各棱空间的位置明确，易得各棱空间所包含的样本点及包含的样本点数量。记$S(l_1,l_2,\cdots,l_d)$包含的样本点数量为$n(l_1,l_2,\cdots,l_d)$，包含样本点数量最多的棱空间的索引指标集合为$M_p=\arg\max\limits_{\substack{l_i\in\{0,1,2,\cdots,m_i\}\\ i=1,2,\cdots,d}}\left(n(l_1,l_2,\cdots,l_d)\right)$，显然$|M_p|\geq 1$。只需找到索引指标在$M_p$的棱空间中密度最大的样本点即可，样本点密度定义为一定范围内样本点的个数与数据集大小的比值。

定义 3 设$X=\{x_1,x_2,\cdots,x_n\}$为一组 d 维的数据集，样本$x_i=(x_{i1},x_{i2},\cdots,x_{id})\in S(\gamma)$，$n(\gamma)$为$S(\gamma)$的样本数量，$M_p$为包含样本点数量最多的棱空间的索引指标集合，阈值半径为

$$R=\frac{1}{|M_p|}\sum_{\gamma\in M_p}\frac{1}{C_{n(\gamma)}^2}\sum_{\substack{x_i,x_j\in S(\gamma)\\ 0\leq i<j\leq n(\gamma)}}dist(x_i,x_j), \tag{5}$$

样本x_i在$S(\gamma)$中的R邻域样本集为

$$U(x_i,R\,|\,S(\gamma))=\{x_j\in S(\gamma)\,|\,dist(x_i,x_j)\leq R\}, \tag{6}$$

样本x_i在$S(\gamma)$中的密度为

$$\rho(x_i\,|\,S(\gamma))=\frac{|U(x_i,R\,|\,S(\gamma))|}{|X|}。 \tag{7}$$

3.1.3. 第三步：确定初始簇心

在得到包含样本点数量最多的棱空间后，需要在这些棱空间中找到密度最大的点。设γ为棱空间索引指标，记$S(\gamma)$中密度最大的样本点集合为$M_a(\gamma)=\arg\max\limits_{x_i\in S(\gamma)}\left(\rho(x_i\,|\,S(\gamma))\right)$，则$|M_a(\gamma)|\geq 1$。考虑到$|M_p|\geq 1$，$|M_a(\gamma)|\geq 1$，即密度最大的样本点可能不唯一。需要进一步细化算法，挑选出更为紧凑的样本点。

定义 4 设$X=\{x_1,x_2,\cdots,x_n\}$为一组维度为 d 的数据集，M_p为包含样本点数量最多的棱空间的索引指标集合，$U(x,R\,|\,\gamma)$为x在$S(\gamma)$中的R邻域样本集。$\forall\gamma\in M_p$，$\forall x\in S(\gamma)$，x在$S(\gamma)$中的紧凑度为

$$com(x\,|\,\gamma)=\frac{\sum\limits_{x_i\in U(x,R|\gamma)\setminus\{x\}}dist(x_i,x)}{|U(x,R\,|\,\gamma)|-1} \tag{8}$$

紧凑度定义为样本周围的点到自身距离的平均值，紧凑度越小，反映了周围点到该点的距离越小，该点越紧凑。记$\mu_i^0=\arg\min\limits_{\substack{x\in M_a(\gamma)\\ \gamma\in M_p}}com(x\,|\,\gamma)$为挑选出的第 i 个初始簇心，为了使得第$i+1$个初始簇心不在已经挑选出的簇心周围，需要对样本点进行剔除。在挑选出样本点后，将$U(\mu_i^0,R\,|\,S(\gamma))$从$X$中剔除，从$X\setminus U(\mu_i^0,R\,|\,S(\gamma))$中重复统计棱空间样本点数量、计算样本点密度、挑选初始点及剔除样本点，直至选出 k 个初始簇心。

3.2. 具体算法步骤

SK-means 算法避开了计算所有样本点两两之间的距离，只在个别棱空间计算部分样本点的密度。在

降低计算量的同时，又细化了算法，解决了密度最大的样本点同时存在只能随机挑选的问题。下面给出SK-means初始簇心选取算法的具体步骤。在已经挑选出的簇心周围，需要对样本点进行剔除。在挑选出样本点后，将$U\left(\mu_i^0,R|S(\gamma)\right)$从$X$中剔除，从$X\setminus U\left(\mu_i^0,R|S(\gamma)\right)$中重复统计棱空间样本点数量、计算样本点密度、挑选初始点及剔除样本点，直至选出k个初始簇心。

算法2 SK-means初始簇心选取算法

输入：数据集$X=\{x_1,x_2,\cdots,x_n\}$，簇个数k。

输出：k个初始簇心。

步骤1 构造棱空间；

步骤2 $\forall x\in X$，判断x属于哪些棱空间；

步骤3 统计各空间样本点个数$n(l_1,l_2,\cdots,l_d)$，$l_i\in\{0,1,2,\cdots,m_i\}$，$i=1,2,\cdots,d$。找到包含样本点个数最多的棱空间的索引指标集合$M_p$；

步骤4 $j=1$，$\forall\gamma\in M_p$，计算$S(\gamma)$中样本点的密度，找到$S(\gamma)$中密度最大的样本点$M_a(\gamma)$，$\forall x\in M_a(\gamma)$，找出使得$md(x|\gamma)$最小的x以及所属的棱空间的索引指标γ，初始簇心$\mu_j^0=\arg\min\limits_{\substack{x\in M_a(\gamma)\\ \gamma\in M_p}} md(x|\gamma)$，$j:=j+1$；

步骤5 在X中剔除$U\left(\mu_j^0,R|S(\gamma)\right)$中的样本点，$X:=X\setminus U\left(\mu_j^0,R|S(\gamma)\right)$；

步骤6 重复步骤3-5直至选出k个初始簇心。

4. 实验结果及分析

4.1. 实验背景

实验设备的处理器是12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz，内存为16.0 GB，Microsoft Windows11的操作系统，系统类型为64位操作系统，基于x64的处理器，算法编写和编译是在Python3.12.2环境下实现的。

为了验证本文算法对降低聚类迭代次数的有效性，本文选取了UCI数据库中的Iris等十二个数据集来进行实验分析。这些数据集的维度从几维到十几维不等，数据量也比较广泛，从而反映出SK-means算法有一定的适用性。具体数据集及基本信息见表3。

Table 3. Basic data set information
表3. 数据集基本信息

数据集	数据集大小	数据维度	数据类别
Balance	625	4	3
Tae	151	5	3
Haberman	306	3	2
Iris	150	4	3
Led	500	7	10
Seeds	210	7	3
Titanic	2201	3	2
Wine	178	13	3
Heart	270	13	2
Appendicitis	106	7	2
Phoneme	5405	5	2
Hayes-Roth	160	4	3

4.2. 聚类评价指标

为了验证所提算法在聚类分析上的有效性，采用三个指标来评价算法，分别为准确率(ACC)、平方误差(SSE)、轮廓指数[30] (SI)。准确率是指预测正确的样本个数与总个数的比值，其值位于[0, 1]，值越大则表示聚类效果越接近真实聚类结果。平方误差为各点到所属簇簇心的距离平方和，值越小反映出簇越集中。若某个样本 $x_i \in C_j$，则 x_i 的轮廓系数为

$$S(x_i)=\frac{b(x_i)-a(x_i)}{\max\{a(x_i),b(x_i)\}} \tag{9}$$

其中 $a(x_i)=\frac{1}{|C_j|-1}\sum_{x_l\in C_j,l\neq i} dist(x_i,x_l)$，$b(x_i)=\min_{t\neq j}\frac{1}{|C_t|}\sum_{x_l\in C_t} dist(x_i,x_l)$。则整个聚类结果的轮廓指数为

$$\mathrm{SI}=\frac{1}{n}\sum_{i=1}^{n}s(x_i) \tag{10}$$

轮廓指数的取值范围为$[-1,1]$，取值越大聚类结果越紧凑，聚类效果越好。

4.3. 实验结果

为了验证 SK-means 算法在降低迭代次数和提高聚类准确性方面是否有效，本文将 SK-means 算法与传统 K-means、K-means++聚类算法，与文献[24]提出的算法在不同方面进行对比。由于传统的 K-means 和 K-means++对初始簇心的选取都具有一定随机性，造成了迭代次数与准确率不稳定，于是本文对每个数据集都做了 100 次实验，取平均值作为传统 K-means 和 K-means++的代表。实验结果见表 3。与文献[24]对比的实验结果见于表 4。考虑到表格容量，表格内数值仅保留了前几位。在各方面实数值的比较中，效果较好的算法所对应的数值已在表格内加粗。

Table 4. Results of accuracy and number of iterations of each dataset
表 4. 各数据集准确率和迭代次数结果

算法	Iris			Wine			Tae		
	准确率	SSE	迭代次数	准确率	SSE	迭代次数	准确率	SSE	迭代次数
K-means	0.813	89.272	7.25	0.682	2.43e+7	7.84	0.337	17,315	6.41
K-means++	0.889	78.943	6.39	0.663	2.48e+7	6.58	0.318	17,325	7.77
SK-means	0.893	78.940	5	0.702	2.37e+7	9	0.364	17,666	10
算法	Heart			Haberman			Seeds		
	准确率	SSE	迭代次数	准确率	SSE	迭代次数	准确率	SSE	迭代次数
K-means	0.500	5.82e+6	8.55	0.448	31,330	8.74	0.890	588.04	8.99
K-means++	0.469	5.82e+6	8.38	0.520	31,455	7.46	0.893	587.85	6.08
SK-means	0.590	5.82e+6	6	0.520	30,533	10	0.895	587.31	5
算法	Titanic			Phoneme			Appendicitis		
	准确率	SSE	迭代次数	准确率	SSE	迭代次数	准确率	SSE	迭代次数
K-means	0.709	4273.5	3.04	0.667	12,947	12.25	0.804	17.522	7.38
K-means++	0.732	4172.7	2.71	0.666	12,916	12.3	0.807	17.530	6.53
SK-means	0.776	4059.6	2	0.668	12,838	12	0.830	17.482	5

续表

算法	Hayes-Roth			LED			New-Thyroid		
	准确率	SSE	迭代次数	准确率	SSE	迭代次数	准确率	SSE	迭代次数
K-means	0.332	358.67	7.43	0.540	271.97	6.41	0.801	28,968	10.03
K-means++	0.386	356.88	7.27	0.643	240.84	4.77	0.771	28,942	9.71
SK-means	0.450	360.04	5	0.742	222.27	4	0.860	28,917	8

Table 5. Accuracy improves quantitative data
表 5. 准确率提升量化数据

数据集	Iris	Wine	Tae	Heart
K-means	9.84%	2.93%	8.01%	18.00%
K-means++	0.45%	5.88%	14.47%	25.80%
数据集	Haberman	Seeds	Titanic	Phoneme
K-means	16.07%	0.56%	9.45%	0.15%
K-means++	0	0.22%	6.01%	0.30%
数据集	Appendicitis	Hayes-Roth	LED	New-Thyroid
K-means	3.23%	35.54%	37.41%	7.37%
K-means++	2.85%	16.58%	15.40%	11.54%

Table 6. Each algorithm evaluates indexes in different data clustering classes
表 6. 各算法在不同数据集聚类评价指标

UCI 数据集	K-means		K-means++		文献[24]		SK-means	
	SI	ACC	SI	ACC	SI	ACC	SI	ACC
Iris	0.546	0.825	0.552	0.888	0.549	0.887	0.553	0.893
Wine	0.567	0.669	0.565	0.631	0.571	0.702	0.571	0.702
Hayes-Roth	0.198	0.442	0.204	0.450	0.571	0.432	0.201	0.450
Heart	0.365	0.500	0.365	0.469	0.377	0.589	0.367	0.590
Tae	0.313	0.337	0.312	0.318	0.328	0.358	0.293	0.364
Haberman	0.395	0.447	0.395	0.520	0.393	0.500	0.399	0.520

表 4 为 SK-means 算法与传统 K-means、K-means++在准确率、SSE 及迭代次数的比较。对比发现 SK-means 算法在大部分数据集有着较好的聚类结果。SK-means 算法在所有数据集的准确率相较于传统 K-means 和 K-means++均有提升、SSE 和迭代次数除在极个别数据集略有增加，其余均有降低。表 6 为 SK-means 算法在 6 个数据集与 K-means、K-means++及文献[24]的比较。对比发现 SK-means 的准确率在 6 个数据集均最优，而轮廓系数较文献[24]效果稍差。

4.4. 聚类可视化

为了更直观的展示 SK-means 算法的聚类效果，将 Iris 数据集前 2 维分别用 K-means、K-means++聚类，得到效果如图 1。

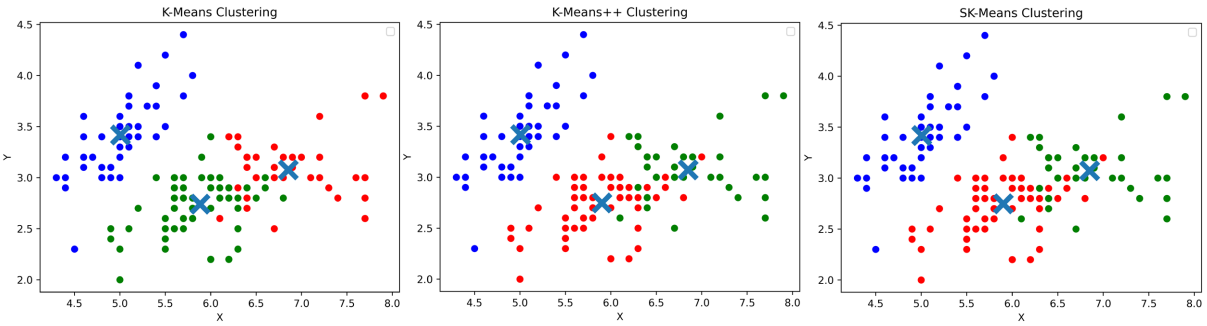

Figure 1. The clustering effect of three algorithms in Iris data set
图 1. Iris 数据集三种算法的聚类效果

图中相同的颜色的样本点为一类，叉号为最终聚类簇心。可以看到 K-means 算法将右下角都聚为一类，左上角分为两类。Iris 数据集真实标签为三类均分，每个簇各有 50 个样本点。就数量而言，K-means 算法聚类效果不佳。而 K-means++和 SK-means 算法效果相似，但 SK-means 算法迭代次数更少且准确率更高。

4.5. 结果分析

由表 4 可知，与传统 K-means 聚类算法和 K-means++算法相比，改进的算法准确率均有相应的提升，且在大多数数据集提高明显。特别是 LED 数据集，相较于传统 K-means 聚类算法和 K-means++算法准确率分别提高了 15.4%、37.4%。在降低 SSE 和迭代次数方面，SK-means 算法在大部分数据集都有起到一定的作用。但在 Wine 等数据集本文在提升准确率的情况下，改进的算法迭代次数与 K-means++算法相比略有逊色。分析发现这些数据集某些维度数据范围过大，造成每个点 R 范围内样本点个数都比较少，不易比较哪个点更适合作为初始簇心。于是该算法对于样本点稀疏的数据集可能会造成迭代次数增加。由表 4 可知 SK-means 算法在大多数数据集轮廓指数高于 K-means、K-means++，且准确率相较于文献[24]提出的算法均有提高。

5. 结束语

随着大数据时代的到来，传统的 K-means 聚类算法已经不能满足数据挖掘的需求，为了提高聚类效果，本研究在传统 K-means 聚类算法的基础上，提出了一种数据预处理的算法，通过对高密度差异数据集的处理，成功降低了 K-means 的迭代次数，提高了聚类的准确率。实验结果表明，SK-means 算法排除了传统 K-means 聚类算法在选取初始簇心的随机性，科学选取初始簇心，降低了陷入局部最优的可能性。通过对比准确率，SK-means 算法有着较好的聚类效果。尽管本研究主要关注了初始聚类中心和样本数据密度差异对算法性能的影响，但仍有许多方面值得进一步研究和探索。譬如在 SK-means 算法步骤 4 中，剔除样本点的范围可以根据数据集的稀疏程度来选取，以期获得更快的迭代和更优的聚类效果；可以排除掉离群点对内群点单独做聚类，聚类稳定后再将其归为最近的簇。未来的工作将重点关注这两方面以及优化算法的收敛性和复杂度，以期发现更优的方式提高算法的效率和应用范围。

参考文献

[1] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[2] Han, J., Kamber, M. and Pei, J. (2001) Data Mining Concepts and Techniques Orlando. Morgan Kaufmann Publishers, San Francisco.

[3] Zhang, W., Zhang, X., Zhao, J., Qiang, Y., Tian, Q. and Tang, X. (2017) A Segmentation Method for Lung Nodule

Image Sequences Based on Superpixels and Density-Based Spatial Clustering of Applications with Noise. *PLOS ONE*, **12**, e0184290. https://doi.org/10.1371/journal.pone.0184290

[4] Shahbazitabar, M. and Abdi, H. (2018) A Novel Priority-Based Stochastic Unit Commitment Considering Renewable Energy Sources and Parking Lot Cooperation. *Energy*, **161**, 308-324. https://doi.org/10.1016/j.energy.2018.07.025

[5] Andrienko, G., Andrienko, N., Fuchs, G. and Garcia, J.M.C. (2018) Clustering Trajectories by Relevant Parts for Air Traffic Analysis. *IEEE Transactions on Visualization and Computer Graphics*, **24**, 34-44. https://doi.org/10.1109/tvcg.2017.2744322

[6] Caruso, G., Gattone, S.A., Fortuna, F. and Di Battista, T. (2021) Cluster Analysis for Mixed Data: An Application to Credit Risk Evaluation. *Socio-Economic Planning Sciences*, **73**, Article ID: 100850. https://doi.org/10.1016/j.seps.2020.100850

[7] Ji, X., Vedaldi, A. and Henriques, J. (2019) Invariant Information Clustering for Unsupervised Image Classification and Segmentation. 2019 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Seoul, 27 October-2 November 2019, 9864-9873. https://doi.org/10.1109/iccv.2019.00996

[8] MacQueen, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the* 5*th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, 281-297.

[9] Ramze Rezaee, M., Lelieveldt, B.P.F. and Reiber, J.H.C. (1998) A New Cluster Validity Index for the Fuzzy C-Mean. *Pattern Recognition Letters*, **19**, 237-246. https://doi.org/10.1016/s0167-8655(97)00168-2

[10] 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.

[11] 王建仁, 马鑫, 段刚龙. 改进的 K-means 聚类 k 值选择算法[J]. 计算机工程与应用, 2019, 55(8): 27-33.

[12] Sinaga, K.P. and Yang, M. (2020) Unsupervised K-Means Clustering Algorithm. *IEEE Access*, **8**, 80716-80727. https://doi.org/10.1109/access.2020.2988796

[13] 何选森, 何帆, 徐丽, 等. K-Means 算法最优聚类数量的确定[J]. 电子科技大学学报, 2022, 51(6): 904-912.

[14] 唐东凯, 王红梅, 胡明, 等. 优化初始聚类中心的改进 K-means 算法[J]. 小型微型计算机系统, 2018, 39(8): 1819-1823.

[15] Breunig, M.M., Kriegel, H., Ng, R.T. and Sander, J. (2000) LOF: Identifying Density-Based Local Outliers. *Proceedings of the* 2000 *ACM SIGMOD International Conference on Management of Data*, Dallas, 15-18 May 2000, 93-104. https://doi.org/10.1145/342009.335388

[16] 朱利, 邱媛媛, 于帅, 等. 一种基于快速 k-近邻的最小生成树离群检测方法[J]. 计算机学报, 2017, 40(12): 2856-2870.

[17] 刘凤, 戴家佳, 胡阳. 基于局部密度离群点检测 K-means 聚类算法[J]. 重庆工商大学学报(自然科学版), 2021, 38(4): 30-35.

[18] 张玉芳, 毛嘉莉, 熊忠阳. 一种改进的 K-means 聚类算法[J]. 计算机应用, 2003, 23(8): 31-33.

[19] 袁方, 孟增辉, 于戈. 对 K-means 聚类算法的改进[J].计算机工程与应用, 2004, 40(36): 177-178.

[20] 赖玉霞, 刘建平. K-means 算法的初始聚类中心的优化[J]. 计算机工程与应用, 2008, 44(10): 147-149.

[21] 汪中, 刘贵全, 陈恩红. 一种优化初始中心点的 K-means 算法[J]. 模式识别与人工智能, 2009(2): 299-304.

[22] 陈光平, 王文鹏, 黄俊. 一种改进初始聚类中心选择的 K-means 聚类算法[J].小型微型计算机系统, 2012, 33(6): 1320-1323.

[23] 谢娟英, 王艳娥. 最小方差优化初始聚类中心的 K-means 算法[J]. 计算机工程, 2014, 40(8): 205-211.

[24] 郭永坤, 章新友, 刘莉萍, 等. 优化初始聚类中心的 K-means 聚类算法[J]. 计算机工程与应用, 2020, 56(15): 172-178.

[25] 韩凌波, 王强, 蒋正锋, 等. 一种改进的 K-means 初始聚类中心选取算法[J]. 计算机工程与应用, 2010, 46(17): 150-152.

[26] Steinley, D. (2006) K-Means Clustering: A Half-Century Synthesis. *British Journal of Mathematical and Statistical Psychology*, **59**, 1-34. https://doi.org/10.1348/000711005x48266

[27] 杨俊闯, 赵超. K-Means 聚类算法研究综述[J]. 计算机工程与应用, 2019, 55(23): 7-14.

[28] 王森, 刘琛, 邢帅杰. K-means 聚类算法研究综述[J]. 华东交通大学学报, 2022, 39(5): 119-126.

[29] Rousseeuw, P.J. (1987) Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, **20**, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

[30] 薛印玺, 许鸿文, 李羚. 基于样本密度的全局优化 K 均值聚类算法[J]. 计算机工程与应用, 2018, 54(14): 143-147.