基于 K-means 聚类算法的改进

李金涛　艾　萍　岳兆新　马梦梦　边世哲
（河海大学计算机与信息学院 南京 211100）

摘　要：基于传统的 K-means 聚类方法提出一种基于密度的改进 K-means 聚类方法。改进后的方法，首先选取数据集中密度最大的点作为第 1 个聚类中心点，以此为基准，选取离此点最远的点作为第 2 个初始聚类中心，再在剩余的点中找出离这两个初始点距离最远的点作为第 3 个聚类中心，以此类推，直到找到所需的 K 个点，之后再根据 K-means 算法迭代更新聚类中心，直到收敛或达到设定的迭代次数为止。实验结果表明，提出的方法与传统 K-means 方法相比准确率及稳定性方面均有所提高，可以作为聚类研究的一个新的思路。

关键词：K-means 聚类；密度聚类；聚类稳定性
中图分类号：TP181　TN957　　**文献标识码**：A　　**国家标准学科分类代码**：520.10

Improvement of clustering algorithm based on K-means

Li Jintao　Ai Ping　Yue Zhaoxin　Ma Mengmeng　Bian Shizhe
(College of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract: This paper presents a method of clustering which improves the stability and efficiency of K-means clustering and based on density. The first step of this method is to select the largest density point of the data set . On this basis, the farthest point from this point is selected as the second initial cluster center, Then, we look for the farthest distance from the two initial points in the remaining points as the center of the third clusters, And so on, until you find the desired K points, and then according to the K-means algorithm to update the cluster center, until convergence or Up to the set number of iterations. The result of the experiments show that the proposed method is better than the traditional K-means method in terms of accuracy and stability, and can be used as a new idea of clustering research.

Keywords: K-means clustering; density clustering; clustering stability

1　引　言

聚类(clustering)在语音识别、字符识别、机器视觉、图像分割、图像处理、数据压缩和信息检索等方面有重要的作用，尤其是在识别数据的内在结构方面甚为有效。目前聚类主要应用在语音、字符、图像等的识别和分割，以及对于数据的压缩和信息检索等[1]。截止到目前，世界范围内对聚类的定义还没有一个统一的描述，本文给出的描述如下：聚类是将一个给定的数据集通过某种算法划分为不同类或簇的过程，并使得划分结果具有“高内聚，低耦合”的特征，也即：使得同一类内的数据对象的相似度较高，不同类内数据对象的相似度较低[2]。

2　聚　类

由于目前人们面临的数据量大、数据类型较多、维度高，迄今为止还没有一种能适应于所有数据类型的聚类算法。但根据最基本的聚类思想衍生出了众多的聚类算法，这些不同的算法适用于不同的需求和领域，通常根据数据间的聚类、聚类规则等标准，主要的聚类方法大致可以分为基于划分的聚类，基于层次的聚类，基于密度的聚类，基于网格的聚类，基于模型的聚类以及基于普聚类[3]的相关算法等。其中影响最为深远，且最具代表的是 K-means 聚类算法。

K-means 算法是 Lloyd 在 1957 给出的标准算法的基础上，由 MacQueen 在 1967 年最早使用 K-means 一词来代表此算法[4]，继而传播开来，K-means 算法在 1982 年由 Lloyd[5] 给出了数学证明和算法的详细步骤。K-means算法的基本思想是：给定一个样本集 $D=\{x_1,x_2,\cdots,x_m\}$ 中，每个向量为 d 维，将 m 个数据分成 k 个簇的集合 $S=\{s_1,$

收稿日期：2017-03

$s_2,\cdots,s_k\}$，其中，s_i 是样本集 D 的子集，$s_i \cap s_j = \varnothing$ 且并使得 $\bigcup_{i=1}^{k} s_i = D$。K-means 聚类的最终目标是使得每个簇内的距离的和达到最小，具体公式为：

$$\mu_i = \frac{\sum_{x \in s_i} x}{|s_i|} \tag{1}$$

其中 μ_i 是集合 s_i 的簇中心向量，用公式表示为：

$$\operatorname{argmin}_S \sum_{i=1}^{k} \sum_{x \in s_i} \|x - \mu_i\|^2 \tag{2}$$

求解 K-means 的 Lloyd 经典算法具体描述如表 1 所示。

表 1　K-means 算法伪代码

输入：样本集 $D=\{x_1,x_2,\cdots,x_m\}$；最终聚类簇数 k
过程：
1：从 D 中随机选取 K 个样本作为初始均值向量$\{\mu_1,\mu_2,\cdots,\mu_k\}$
2：repeat
3：令 $S_i=\varnothing(<1\leqslant i\leqslant k)$
4：for $j=1,2,\cdots,m$ do
5：计算样本 x_j 与个均值向量 μ_i 的距：$d_{ji}=\|x_j-\mu_i\|_2$
6：根据距离最近的均值向量确定 x_j 的簇标记：$\lambda_j = \arg\min_{k\{1,2,\cdots,k\}} d_{ji}$
7：将样本 x_j 划入相应的簇：$s_{\lambda j}=s_{\lambda j}\cup\{x_j\}$
8：end for
9：for $i=1,2,\cdots,k$ do
10：计算新均值向量：$\mu'_i = \frac{1}{|S_i|}\sum_{x\in s_i} x$
11：if $\mu'_i \neq \mu_i$ then
12：将当前均值向量 μ_i 更新为 μ'_i
13：else
14：保持当前均值向量不变
15：end if
16：end for
17：until 当前均值向量均未更新
输出：簇划分 $s_i = \{s_1,s_2,\cdots,s_k\}$

Lloyd 提出的 K-means 方法是目前在数据挖掘以及机器学习等相关领域应用最多的算法，对于一般的数据 K-means 本身具有简单、快速、易于解释、性能高，伸缩性好[6]等多项优点，但是，此方法也存在一些缺陷[7-8]：1）对初始选取的聚类中心值非常敏感[9]；2）通常聚类结果都是圆状或球状簇；3）对离群值和“噪音”非常敏感；4）容易陷入局部最优解[10]。

由于经典 K-means 方法的上述缺陷，本文提出了一种基于密度和 K-means 相结合的方法，实验证明能对经典 K-means 聚类效果有所改进。

3　基于密度和 K-means 结合的聚类算法

3.1　改进算法的基本思想

选取数据集中密度最大的点为第 1 个聚类中心点[3,11]，以此为基准，选取离此点最远的数据点为第 2 个初始聚类中心，再选择距离最近的点作为同一簇内的数据点。之所以要这样选择，原因之一是为了避免传统 K-means 算法在聚类过程中，其选取的初始点都是本应该属于同一个类的点；第二个原因是避免传统 K-means 算法在聚类过程中使得某个簇把数据集中过多的点错误的划在其中，以致出现聚类效果不佳。大致来说，对于给定的某个数据集 $D=\{x_1,x_2,\cdots,x_m\}$，改进后的算法会由给定的阈值来计算每个点的密度，根据计算的结果来确定一个点，并以此点作为第 1 个初始聚类中心，设此数据点为 K_1，找到 K_1 后，再以 K_1 为基准来计算其他点到 K_1 的距离，距离公式使用欧氏距离公式：

$$D(\boldsymbol{x}_i,\boldsymbol{x}_j) = ((x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{im}-x_{jm})^2)^{1/2} \tag{3}$$

式中：$\boldsymbol{x}_i,\boldsymbol{x}_j$ 分别是有 m 个属性构成的向量，也即 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{im})$，$x_j=(x_{j1},x_{j2},\cdots,x_{jm})$，选择这些向量中距离最大的一个数据点作为第 2 个初始中心点，反复执行此步，直到得到 K 个初始聚类中心点。

3.2　改进算法的基本流程

假设现在有 n 个数据点，每个数据都是 d 维的，要对这些数据聚类，类数为 K。聚类流程首先会根据密度聚类算法及其阈值找出 K 个初始聚类中心点，然后再在此基础上用 K-means 聚类方法进行迭代划分，直到得到最终的结果。步骤如表 2 所示。

表 2　改进算法的伪代码

1：输入：样本集 $D=\{x_1,x_2,\cdots,x_m\}$；
最终聚类簇数 k，邻域(ϵ，MinPts)。

2：任选数据集中的一个核心对象为“种子(seed)”点，再由此出发根据给定的邻域参数找出所有的核心对象，选取所有核心对象在给定阈值内数据点最多的哪个核心对象，设为初始 Q1 点。

3：在改进的算法中，简单起见，采用欧氏距离作为相似度度量函数，以 Q1 为中心进行计算其他点距离与其的距离，选取最远的一个点作为第 2 个初始聚类中心，记为 Q2。

4：反复执行 3，直到得到 K 个初始点。

5：在 4 的基础上用表 1 的 K-means 算法中的迭代过程反复执行和更新中心点(根据最近邻条件)，直到最终聚成的 K 个类簇不再变化。

6：输出 K 个类簇结果。

由上述算法可知，此算法主要分为两步：1）选择合适正确的初始聚类中心；2）聚类以及更新的迭代过程。分析时间复杂度可得第 1 步的时间复杂度为 $O(n^2)$，第 2 步为 $O(nkt)$，k 和 t 分别为聚类数目及迭代次数，两者时间复杂度相加简化后可得时间复杂度为 $O(n^2)$。此外由

于第2步并不是随机的选取中心点，所以聚类精度会提高，迭代的次数也会大大的减少，进而缩短聚类过程的时间。

下面通过实例来说明本文提到的聚类方法是怎么选取初始中心，并最终划分出聚类结果。以20个二维数据点的数据集为例来说明，具体数据为：(1,2),(1,3) ,(1,4),(2,2),(2,3),(3,1),(3,3),(3,6),(4,3),(4,4),(4,5),(5,4),(5,6),(5,7),(6,3),(6,6),(7,0),(7,1),(7,2),(7,7)，需要将这20个数据划为3类，这些数据点如图1所示。

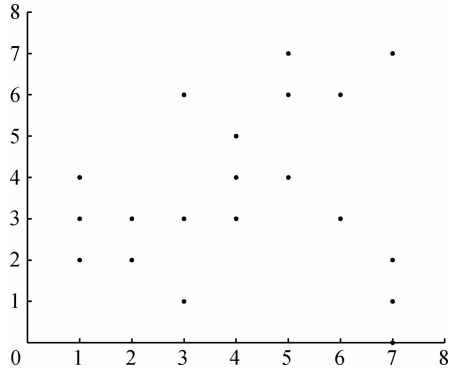

图1 二维数据点实例

这20个点的具体聚类步骤如下：

1)给定阈值 $d=2$，求出这些点钟密度最大的点(2,3)，作为初始聚类的第1个点，紧接着求出离其最远的点(7,7)作为第2个初始点。

2)以(2,3),(7,7)这两个点为基准，在剩下的点中选择距离(2,3),(7,7)距离之和最近的点(7,0)作为第3个聚类点，此时3个初始点已全部找到，再根据最近邻原则分别计算剩余点中距离这3个点哪个最近，并划分到离其最近的初始点所代表的聚类簇，同时计算准则函数。

3)计算每次划分好类簇后的所有同一簇内所有对象的均值，作为新的聚类中心，更新聚类中心，再以新的聚类中心计算准则函数的值。

4)对比步聚2)和步聚3)步的准则函数值，如果小于或等于预定的阈值或达到设定的迭代次数，则停止迭代，否则继续步骤3)迭代。

5)最终的聚类结果如图2所示。

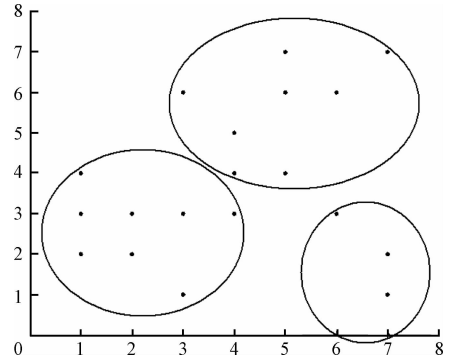

图2 二维数据点实例聚类结果

4 实验及分析

实验环境：Windows 7 旗舰版操作系统，处理器 I7-6700，实验软件 MATLAB R2015a。

所用数据集为4维随机生成的数据集，共150条数据，分为3类，每类的个数分别为48,52,50，具体实验数据如表3所示。

表3 第1次 K-means 随机实验结果

实验数据	实验数据	实验数据	实验数据
5.1,3.5,1.4,0.2	5.5,4.2,1.4,0.2	4.9,3.1,1.5,0.1	5.0,3.2,1.2,0.2
5.5,3.5,1.3,0.2	4.9,3.1,1.5,0.1	4.4,3.0,1.3,0.2	5.1,3.4,1.5,0.2
5.0,3.5,1.3,0.3	4.5,2.3,1.3,0.3	4.4,3.2,1.3,0.2	5.0,3.5,1.6,0.6
5.1,3.8,1.9,0.4	4.8,3.0,1.4,0.3	5.1,3.8,1.6,0.2	4.6,3.2,1.4,0.2
5.3,3.7,1.5,0.2	5.0,3.3,1.4,0.2	7.0,3.2,4.7,1.4	6.4,3.2,4.5,1.5
6.9,3.1,4.9,1.5	5.5,2.3,4.0,1.3	6.5,4.8,2.6,1.3	5.2,2.7,3.9,1.4
6.3,3.3,4.7,1.6	4.9,2.4,3.3,1.0	6.6,2.9,4.6,1.3	5.2,2.7,3.9,1.4
5.0,2.0,3.5,1.0	6.1,2.8,4.7,1.2	6.4,2.9,4.3,1.3	6.6,3.0,4.4,1.4
6.8,2.8,4.8,1.4	6.7,3.0,5.0,1.7	6.0,2.9,4.5,1.5	5.7,2.6,3.5,1.0
5.5,2.4,3.8,1.1	5.5,2.4,3.7,1.0	5.8,2.7,3.9,1.2	6.0,2.7,5.1,1.6
5.4,3.0,4.5,1.5	6.0,3.4,4.5,1.6	6.7,3.1,4.7,1.5	6.3,2.3,4.4,1.3
5.6,3.0,4.1,1.3	5.5,2.5,4.0,1.3	5.5,2.6,4.4,1.2	6.1,3.0,4.6,1.4
5.8,2.6,4.0,1.2	5.0,2.3,3.3,1.0	5.6,2.7,4.2,1.3	5.7,3.0,4.2,1.2
5.7,2.9,4.2,1.3	6.2,2.9,4.3,1.3	5.1,2.5,3.0,1.1	5.7,2.8,4.1,1.3
6.3,3.3,6.0,2.5	5.8,2.75.1,1.9	7.1,3.0,5.9,2.1	6.3,2.9,5.6,1.8
6.5,3.0,5.8,2.2	7.6,3.0,6.6,2.1	4.9,2.5,4.5,1.7	7.3,2.9,6.3,1.8
6.7,2.5,5.8,1.8	7.2,3.6,6.1,2.5	6.5,3.2,5.1,2.0	6.4,2.7,5.3,1.9

续表 3

实验数据	实验数据	实验数据	实验数据
6.8,3.0,5.5,2.1	5.7,2.5,5.0,2.0	5.8,2.8,5.1,2.4	6.4,3.2,5.3,2.3
6.5,3.0,5.5,1.8	7.7,3.8,6.7,2.2	7.7,2.6,6.9,2.3	6.0,2.5,5.0,1.5
6.9,3.2,5.7,2.3	5.6,2.8,4.9,2.0	7.7,2.8,6.7,2.0	6.3,2.7,4.9,1.8
6.7,3.3,5.7,2.1	7.2,3.2,6.0,1.8	6.2,2.8,4.8,1.8	6.1,3.0,4.9,1.8
6.4,2.8,5.6,2.1	7.2,3.0,5.8,1.6	7.4,2.8,6.1,1.9	7.9,3.8,6.4,2.0
6.4,2.8,5.6,2.2	6.3,2.8,5.1,1.5	6.1,2.6,5.6,1.4	7.7,3.0,6.1,2.3
6.3,3.4,5.6,2.4	6.4,3.1,5.5,1.8	6.0,3.0,4.8,1.8	6.9,3.1,5.4,2.1
6.7,3.1,5.6,2.4	4.9,3.0,1.4,0.2	4.7,3.2,1.3,0.2	4.6,3.1,1.5,0.2
4.4,2.9,1.4,0.2	4.9,3.1,1.5,0.1	5.4,3.7,1.5,0.2	4.8,3.4,1.6,0.2
4.8,3.0,1.4,0.1	4.3,3.0,1.1,0.1	5.8,4.0,1.2,0.2	5.7,4.4,1.5,0.4
5.4,3.9,1.3,0.4	5.1,3.5,1.4,0.3	5.7,3.8,1.7,0.3	5.1,3.8,1.5,0.3
5.4,3.4,1.7,0.2	5.1,3.7,1.5,0.4	4.6,3.6,1.0,0.2	5.1,3.3,1.7,0.5
6.0,2.2,5.0,1.5	5.8,2.8,5.1,2.4	5.0,3.0,1.6,0.2	4.8,3.1,1.6,0.2
6.9,3.2,5.7,2.3	6.4,3.2,5.3,2.3	5.0,3.4,1.6,0.4	5.4,3.4,1.5,0.4
5.6,2.8,4.9,2.0	6.5,3.0,5.5,1.8	5.2,3.5,1.5,0.2	5.2,4.1,1.5,0.1
7.7,2.8,6.7,2.0	7.7,3.8,6.7,2.2	5.2,3.4,1.4,0.2	5.5,4.2,1.4,0.2
6.3,2.7,4.9,1.8	7.7,2.6,6.9,2.3	4.7,3.2,1.6,0.2	4.9,3.1,1.5,0.1
6.8,2.8,4.8,1.4	6.7,3.0,5.0,1.7	6.0,2.9,4.5,1.5	5.7,2.6,3.5,1.0
5.0,3.6,1.4,0.2	5.4,3.9,1.7,0.4	4.6,3.4,1.4,0.3	5.0,3.4,1.5,0.2
5.1,2.9,4.8,0.6	6.8.3.9,5.0,1.2	5.3,3.6,4.4,1.3	5.5,4.6,3.0,0.7
7.6,1.5,5.8,2.6	5.3,3.8,5.0,1.0		

4.1 第 1 次实验

为验证本文所提算法的准确性，下面分别以此 150 条数据为基础，进行多次对比实验，其中 K-means 方法选取 3 组，由于本文所提基于密度的方法比较稳定，所以选取 1 组。

第 1 次 K-means 随机实验结果如表 4 所示。

表 4 第 1 次 K-means 随机实验结果

类别	聚类中心						正确	准确率
	初始		最终		实际			
一类	4.799	3.401	5.125	3.407	5.008	3.419	41	0.746 7
	1.900	0.200	3.287	0.982	1.463	0.243		
二类	5.201	2.699	8.812	2.810	5.935	2.771	44	
	3.900	1.400	3.947	1.324	4.263	1.327		
三类	7.900	3.800	6.303	2.870	6.588	2.973	27	
	6.400	2.000	5.013	2.032	5.551	2.028		

第 2 次 K-means 随机实验结果如表 5 所示。

表 5 第 2 次 K-means 随机实验结果

类别	聚类中心						正确	准确率
	初始		最终		实际			
一类	5.401	3.899	5.108	3.566	5.008	3.419	38	0.84
	1.301	0.401	1.374	0.275	1.463	0.243		
二类	6.101	2.801	5.931	2.875	5.935	2.771	45	
	4.702	1.201	4.146	1.305	4.263	1.327		
三类	6.503	3.202	6.600	2.972	6.588	2.973	43	
	5.100	2.000	5.343	1.692	5.551	2.028		

第3次K-means随机实验结果如表6所示。

表6 第3次K-means随机实验结果

类别	聚类中心						正确	准确率
	初始		最终		实际			
一类	4.300	3.000	4.719	3.246	5.008	3.419	33	
	1.100	0.100	1.179	0.204	1.463	0.243		
二类	6.700	3.100	6.247	2.915	5.935	2.771	47	0.653 3
	4.400	1.400	4.448	1.312	4.263	1.327		
三类	7.900	3.800	7.430	3.476	6.588	2.973	28	
	6.400	2.000	5.589	2.012	5.551	2.028		

基于密度的K-means实验结果如表7和8所示。

表7 基于密度的K-means第1次随机实验结果

类别	聚类中心						正确	准确率
	初始		最终		实际			
一类	5.100	3.400	5.043	3.401	5.008	3.419	38	
	1.500	0.200	1.469	0.204	1.463	0.243		
二类	5.800	2.700	5.914	2.767	5.935	2.771	45	0.853 3
	4.100	1.000	4.235	1.320	4.263	1.327		
三类	6.400	2.800	6.439	3.016	6.588	2.973	3	
	5.600	2.200	5.548	2.033	5.551	2.028		

表8 基于密度的K-means第2次随机实验结果

类别	聚类中心						正确	准确率
	初始		最终		实际			
一类	5.100	3.399	5.043	3.400	5.008	3.419	39	
	1.500	0.201	1.469	0.241	1.463	0.243		
二类	5.800	2.701	5.914	2.767	5.935	2.771	46	0.866 6
	4.102	1.001	4.245	1.320	4.263	1.327		
三类	6.398	2.800	6.439	3.016	6.588	2.973	45	
	5.601	2.200	5.548	2.032	5.551	2.028		

从以上几组对比可看出，本文所提出的算法在聚类准确率上有一定的提升。

4.2 第2次实验

为了从总体趋势及稳定性上有一个直观的认识，对表3中的150条数据分别用K-means和本文所提的基于密度的K-means算法进行10次抽取，并进行第2次实验，10次抽取的数据个数分别为数据分别为22，46，65，84，93，104，116，123，134，144，并进行聚类，为能对比的更加直观，图3显示了两种聚类算法分别在这10次随机试验中的准确率对比图，结果如图3所示。

由表3～8和图3可知，本文所提出的算法在第1次数次实验中准确率较高，在第2次随机抽取数据的实验中，当数据量较少时，K-means表现和基于密度的K-means基本相同，但随着数据量的增多，渐渐表现出聚类效果不稳定的情况。

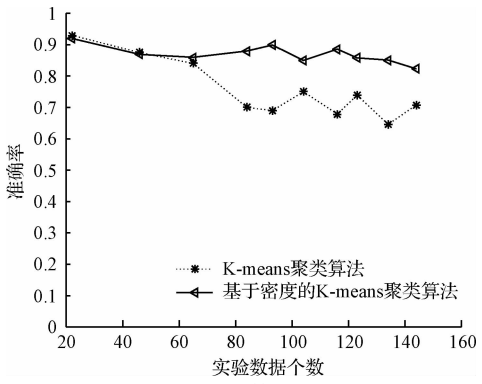

图3 K-means聚类算法和基于密度的K-means聚类算法对比

综合对比来说，本文所提的算法表现比较稳定，上下波动范围比较小，是对传统K-means的一个改进。

（下转第21页）