一种基于密度的 *k*-means 聚类算法

罗军锋，锁志海
（西安交通大学 信息中心，陕西 西安 710049）

摘 要：针对 *k*-means 算法中对初始聚类中心和孤立点敏感的缺点，提出一种基于密度的改进 *k*-means 算法. 该算法引入信息熵和加权距离，从近邻密度出发，去除孤立点对算法的影响，同时确定初始聚类中心，使得聚类中心相对稳定. 实验表明，该算法在准确性、运行效率上均有 10％以上的提升.

关键词：聚类；*k*-means；信息熵；近邻密度；孤立点
中图分类号：TP311 **文献标识码**：A **文章编号**：1000－7180(2014)10－0028－04

A Density Based *k*-means Clustering Algorithm

LUO Jun-feng，SUO Zhi-hai
（Information Center，Xi'an Jiaotong University，Xi'an 710049. China）

Abstract：For *k*-means algorithm to the initial cluster centers，sensitive to outliers shortcomings，we propose a density-based improved *k*-means algorithm. The algorithm introduces entropy and weighted distance，starting from the neighbor density，remove the isolated points on the algorithm while determining the initial cluster centers，making the cluster center is relatively stable. Experimental results show that the algorithm in terms of accuracy，operating efficiency has a very good improvement.

Key words：*k*-means ；Information entropy；Neighbor density；isolated points

1 引言

聚类是数据挖掘中的重要研究方向之一. 其定义是将若干数据对象的集合分成由类似的数据对象组成的多个族的过程. 其目标是在一个族里的数据对象是尽可能最相似的，而在不同族的数据对象尽可能最不相似. 聚类一般大体上可以分为：基于划分的算法如 *k*-means，基于密度的算法如 DBSCAN，基于层次的算法如 BRICH 算法，基于网格的算法等等.

k-means[1] 算法作为一种简单实用聚类算法而成为聚类应用和研究最广泛的算法之一. 传统意义上的 *k*-means 聚类算法存在着对初始聚类中心的选择、数据对象中的孤立点等特别敏感的缺点，容易陷入局部最优，并且其聚类结果一般不是特别稳定. 这一问题的存在，其原因主要是初始聚类中心的随机选择和采用同一类中所有对象的平均值进行分类和区分.

针对这些缺点，许多研究人员对该算法进行了深入的研究与改进. 文献[2] 中用神经网络算法选取初始聚类中心，得到了较好的效果，文献[3] 中将蚁群算法与 *k*-means 算法相融合，提升了聚类的质量，文献[4] 中通过调整数据样本的分布选择初始聚类中心提出新的 *k*- means 算法，文献[5] 中则利用三角不等式性质，文献[6] 中通过设置密度参数的方法加以实现. 对孤立点的处理相对较少，主要利用数据点和最近邻居之间的距离作为检测的依据，文献[7] 中提出了一种基于孤立点数据过滤的改进算法.

我们发现以上这些文献都是分别改进传统 *k*-means 算法的两个主要缺点而加以研究的，并没有针对这两个缺点同时进行研究，但不可否认的是上文提到的这两个缺点是互相影响的. 本文提出了一种基于密度的 *k*-means 算法，本算法从计算数据集的局部可达密度出发，去除孤立点，并按照距离最远原则选择初始聚类中心，着力改善传统 *k*-means

收稿日期：2013－12－18；**修回日期**：2014－02－16

算法的两个缺陷.

2　k-means 算法

众所周知，传统 k-means 算法的思想是：首先从数据对象中随机地选择 k 个对象作为初始聚类中心；接着根据每个对象到聚类中心的距离，按最小距离原则进行划分，划给聚类最近的簇；重新计算每个聚类簇的均值. 计算收敛函数，满足函数收敛要求，算法就终止，否则，重复上述过程.

定义收敛函数为

$$E=\sum_{i=1}^{k}\sum_{x\in C_k}|x-x-_i|^2,$$

式中，E 表示所有数据对象与它所在簇的质心点的距离的总和，E 值越小表示对象与中心的距离越小，簇内的相似性就越高；反之 E 值越大表示簇内的相似性越低. x表示簇内的某一个对象；$\overline{x_i}$表示簇 C_i 的质心点；k 表示簇的数量；C_i 表示第 i 个簇.

传统 k-means 算法的描述：

输入：n 个数据对象，簇的数量 k.

输出：满足收敛函数要求的簇.

(1) 从 n 个数据对象中任意选择 k 个对象，作为初始聚类中心；

(2) 根据每一个簇的均值，计算每一个对象与簇聚类中心的距离，并根据距离最小原则将对象划分到最近的簇；

(3) 重新计算划分后的每个簇的均值；

(4) 计算收敛函数，满足收敛条件则结束，否则回到步骤(2).

本算法中初始聚类中心是任意选取的，选取不同的初始聚类中心，得到的聚类结果也不一样，容易陷入局部最优，同时少量的孤立点的存在，对平均值会产生很大的影响. 传统的聚类算法时间复杂度为 O(KNT)(N 为样本个数，K 为分类个数，T 为算法迭代次数).

3　基于近邻密度的 k-means 算法

针对传统 k-means 的缺点，本算法主要改进包括：引入距离熵，改善距离计算公式；引入近邻密度，通过近邻密度剔除孤立点；进一步利用近邻密度选择高密度对象，并从中选择初始聚类中心.

3.1　算法基本概念

定义 1　对象 p 的 k 距离

对于任意个正整数 k，对象 p 的 k 距离表示 k-dist (p).

不失一般性，数据对象 p 和数据对象 o 同在数据集 D 中，将这两个数据对象之间的距离记为 d (p , o).

满足如下条件时，就取 k-dist (p) 等于 d (p , o) ：

(1) 至少存在 k 个数据对象 o' 满足

$d(p,o')\leqslant d(p,o)$；

(2) 至多存在 k-1 个数据对象 o' 满足

$d(p,o')<d(p,o)$.

定义 2　对象 p 的 k 近邻邻居

设对象 p 的 k 距离为 k-dist (p)，那对象 p 的 k 近邻邻居则为所有到 p 的距离小于等于 p 的 k-dist (p) 距离的数据对象的集合，定义为

$$N_{k\text{-dist}(p)}(p)=\{q\in D\backslash\{p\}\mid d(p,q)\leqslant k\text{-dist}(p)\}.$$

定义 3　对象 p 的近邻距离

设对象 o 为对象 p 的一个近邻，$d(p,o)$ 为对象 P 和对象 o 之间的距离，那么针对对象 o，对象 p 的近邻距离定义如下：

$$\text{NNdist}(p,o)=\max\{k\text{-dist}(p,o),d(p,o)\}$$

定义 4　对象 p 的近邻密度

给定集合 D，p 的近邻密度定义为

$$\text{NND}_{k\text{-dist}(p)}(p)=\frac{\sum_{o\in N_{k\text{-dist}(p)}(p)}\text{NNdist}(p,o)}{|N_{k\text{-dist}(p)}(p)|}\quad(1)$$

在定义 4 中，首先计算数据集中所考察对象 p 的 k 近邻所有邻居对象到对象 p 的近邻距离之和. 如果数据对象是个高密度区域，则它的 k 近邻邻居所涵盖的范围就非常小，其近邻密度取值就小.

3.2　算法的思想

本文提出的新算法借鉴文献[8]的思想，基本思想是：聚类的目标是聚类间要实现距离最大化，聚类内要实现距离最小化. 因此，新算法首先利用对象的近邻密度选择划分选择点，然后在近邻高密度点选择距离最远点作为初始聚类点，然后选择距离已选聚类中心最远的下一个局部高密度点作为下一个距离中心，依次类推. 因为这些选择点是局部高密度区，从而避免了随机划分的盲目性，提高了聚类的性能.

为提高算法的性能，我们对数据集中计算数据对象距离时采用加权距离，权重值的计算使用信息熵来确定.

在数据集的各维属性中，对聚类结果的影响程度不同，同时不同的相邻数据对该数据的最终聚类的影响也不相同. 于是，本文借鉴文献[9]，在度量属

性权重时引入信息熵的概念，并得到相邻数据之间的权重系数，最终求出基于熵权重的距离计算公式.

具体步骤如下：

(1) 设有如下属性值矩阵，其具有 n 个对象，m 维属性：

$$\mathbf{X}=\begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{mm} \end{pmatrix}$$

(2) 构造属性值属性比重矩阵.

首先对属性值进行了标准化处理，这是为了能将不同的量纲属性值进行比较. 处理方法为

$$r_{ij}=\frac{\max(x_{ij})-x_{ij}}{\max(x_{ij})-\min(x_{ij})},$$

式中，r_{ij} 为对象 x_i 的第 j 维属性的属性值比重.

(3) 分别计算第 j 维属性的熵值，权值：

$$\text{熵值：}H_j=-\frac{\sum_{i=1}^{n} r_{ij}\ln r_{ij}}{\ln n}$$

$$\text{权值：}w_j=\frac{1-H_j}{\sum_{j=1}^{m}(1-H_j)}$$

$$(0\leqslant w_j\leqslant 1,\sum_{j=1}^{m} w_j=1).$$

(4) 计算相邻对象间的权重系数.

设对象 x_j 是数据对象 x_i 的某个邻居，则二者间的权重系数的计算公式如下：

$$w_{ij}=\sum_{p=1}^{m} w_p\times\frac{x_{jp}}{\sum x_{op}} \tag{2}$$

式中：x_{op} 是对象 x_i 的第 p 维的属性取值，对象 x_o 表示对象 x_i 的相邻数据对象，w_p 是第 p 维的属性的权值.

从式(2) 中可以看出，相邻对象的权重系数是由对象的所有属性，其所有邻居共同决定，这样在随后计算距离时就最大限度的考虑了相邻对象之间的相互影响及其所有属性的影响.

那么基于信息熵的距离计算公式为

$$d(x_i,x_j)=w_{ij}\times\sqrt{\sum_{k=1}^{g}(x_{ik}-x_{jk})^2} \tag{3}$$

利用改进的距离计算公式充分考虑到了属性值得影响，更精确的计算出各个对象之间的差异程度，在实际聚类算法中可以提高聚类的精确度.

3.3 算法步骤

改进后的算法分三大步骤：

首先根据式(3)求取对象的加权距离，根据式(1)计算各个对象的近邻密度.

然后，对各个对象的近邻密度排序，根据预先设置的阈值剔除离群点，以剔除后的所有对象作为新对象集进行聚类，并构造新的近邻密度矩阵.

最后，在新的近邻密度排序下，选择近邻密度最小的 $2k$ 个对象作为初始聚类中心选择集，首先在初始聚类中心选择集中选取近邻密度最小的 2 个数据对象作为初始聚类中心，然后在剩余的初始聚类中心选择集中选取距离初始的 2 个聚类中心最远的一个数据对象加入到初始聚类中心中，以此类推，直到选取了 k 个聚类中心，以改进后的距离公式进行聚类.

具体算法如下

输入：d 维数据集 D，k，近邻离群阈值 ξ

输出：满足准则函数最小的 k 个族

(1) 计算各个属性的信息熵，同时采用式(3) 计算各对象之间的加权距离，然后进一步计算出各对象的 k-dist 近邻邻居矩阵；

(2) 根据式(1) 计算各对象的近邻密度 NND，并将 NND 降序排序，然后将所有 NND 大于离群阈值 ξ 的对象作为离群点去除，将剩余的对象作为训练数据集 reduct，并更新 k-dist 近邻邻居矩阵和近邻密度 NND 矩阵，同时选择 NND 最小的 $2k$ 个对象作为准初始聚类中心集待用；

(3) 从准初始聚类中心集中选取距离最远的两个数据对象 k_1 和 k_2，将其作为初始的 2 个聚类中心，同时在准初始聚类中心集中删除这两个对象；

(4) 继续从准初始聚类中心集中寻找距离 k_1 和 k_2 最远的对象作为 k_3，并从准初始聚类中心集中删除该对象，依次类推，直到找到 k 个初始聚类中心；

(5) 在训练集 reduct 上，从得到的这 k 个初始聚类中心出发，运用 k-means 算法进行聚类.

3.4 性能分析

从上面的算法描述可看出，本算法的时间复杂度主要分为 2 个部分. 第一部分复杂度为 $O(N_{k\text{-dist}})$，是为计算 p 的第 k-dist 距离的近邻邻居，第二部分复杂度为 $O(|N_{k\text{-dist}}|\times O(N_{k\text{-dist}}))$，是遍历 p 的第 k-dist 距离近邻邻居的邻居，所以算法时间复杂度为

$$O(N_{k\text{-dist}})+O(|N_{k\text{-dist}}|\times O(N_{k\text{-dist}})).$$

由于算法在计算出各个对象的 k-dist 距离后将其自身的近邻邻居保存起来，这样大幅度的降低算法的时间复杂度的价，其付出的代价就是增加规模为 N^2 的空间开销. 实验表面，这种以空间换取时间的方式至少提高 10 倍以上的时间效率.

4　实验结果与分析

本文实验采用 MATLAB 2012b 开发环境，在 Intel(R) Core(TM) i5　CPU 3.4 GHz，4 GB 内存，windows XP 操作系统上运行.

本实验选取 UCI 机器学习数据库中的 Iris，Wine，Glass Identification 作为实验数据.

实验方法：分别采用传统的 k-means 算法、本文提出的新算法进行实验.

实验结果：由于传统的 k-means 算法对初始聚类中心的敏感原因导致聚类结果不稳定，因此求出 8 次运算后所计算的平均准确率来进行比较，实验结果如表 1 所示.

表 1　两种算法运行结果的准确度比较

算法	聚类精度/%		
	Iris	Glass	Wine
k-means	73.16	71.12	72.51
改进的 k-means 算法	87.36	82.61	83.84

从表中可以看到，改进后的 k-means 算法准确度比传统算法提高了至少 10 个百分点.

此外，考虑到改进的算法相对原始 k-means 算法多了距离矩阵的计算，近邻密度的计算等等这些过程，势必造成大量的时间消耗，但是初始聚类中心的快速定位能够大大抵消计算所用的时间，更为重要的是，本算法中将距离矩阵只计算一次后保存，以便后续的计算，大大减少了重复计算过程. 实验结果如表 2 所示.

表 2　比较两个算法运行的时间消耗

算法	聚类时间消耗/%		
	Iris	Glass	Wine
k-means	2 450	3 160	2 870
改进的 k-means 算法	2 680	3 210	2 900

从表中可以看出，虽然改进后的算法要计算两个矩阵，但最终时间消耗并没有增加多少.

更为重要的是，相比较原始 k-means 算法，随着数据集数量的增加，本算法的时间复杂度大大降低，如图 1 所示.

这是因为随着数据集的增大，原始 k-means 算法由于随机选择初始聚类中心，造成时间大量消耗在随后的反复计算中.

5　结束语

本文在传统距离公式中引入信息熵，并据此计算各个对象的近邻距离和近邻密度，去除孤立点对聚类结果的影响. 利用近邻密度选择距离最远的对象作为初始聚类中心. 实验结果显示，新的聚类算法改进了传统 k-means 算法的缺点，准确度提高了约 10%以上.

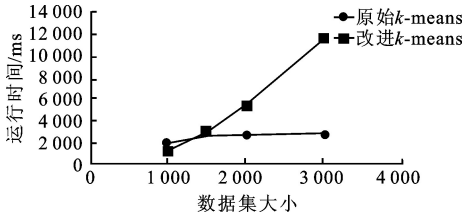

图 1　算法运行时间与数据集关系图

参考文献：

[1] James MacQueen. Some methods for classification and analysis of multivariate observations[C]// Fifth Berkeley Symposium oon Mathematical Statistics and Probability. Berkeley, Calif. University of California Press, 1967:666.

[2] Wang Huaibin, Yang Hongliang, Xu Zhijian. A clustering algorithm use SOM and K-Means in Intrusion Detection[C]// E-Business and E-Government (ICEE), 2010 International Conference. Guangzhou, IEEE, 2010:1281-1284.

[3] 莫锦萍，陈琴，马琳，等. 一种新的 K—Means 蚁群聚类算法[J]. 广西科学院学报，2008，24(4)：284-286.

[4] 曹志宇，张忠林，李元韬. 快速查找初始聚类中心的 k-means 算法[J]. 兰州交通大学学报，2009，28(6)：15-18.

[5] 王守强，朱大铭，徐小平. 求解 K-means 聚类更有效的算法[J]. 计算机工程与设计，2008，29(2)：378-380.

[6] 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.

[7] 张建民. 一种改进的 K-means 聚类算法[J]. 微计算机信息，2010(9)：233-234.

[8] 胡彩平，秦小麟. 一种基于密度的局部离群点检测算法 DLOF[J]. 计算机研究与发展，2010(12)：2110-2116.

[9] 唐波. 改进的 K-means 聚类算法及应用[J]. 软件，2012(3)：36.

作者简介：

罗军锋　男，(1976-)，硕士，工程师. 研究方向为数据挖掘、高校信息化.

锁志海　男，(1971-)，硕士，助理研究员. 研究方向为数据挖掘，高校信息化.