第 16 卷第 1 期 电 脑 与 信 息 技 术 Vol.16 No.1
2008 年 2 月 Computer and Information Technology Feb. 2008
文章编号:1005-1228(2008)01-0038-03

一种改进的 K-means 聚类算法

连凤娜,吴锦林,唐 琦
(厦门大学信息科学与技术学院,福建 厦门 361005)

摘 要:K-means 算法是最常用的聚类算法之一,有很多的优点,但也存在着不足。它不仅对样本的输入顺序敏感,可能产生局部最优解,而且受孤立点的影响很大。文章正是针对这些不足,提出了一种改进的 K-means 算法,主要从数据预处理、初始聚类中心的选择方面进行了改进,并做了改进前后算法的对比实验。结果表明,改进后的算法不但更具稳定性,准确度也高,受孤立点的影响也大大降低。

关键词:K-means 算法;聚类;孤立点
中图分类号:TP311.13 **文献标识码**:A

An Improved Algorithm of K-means

LIAN Feng-na, WU Jin-lin, TANG Qi
(College of Information Science and Technology, Xiamen University, Xiamen, Fujian 361005, China)

Abstract: K-means algorithm is one of the most widespread methods in clustering, including both strong points and also shortages. Not only is it sensitive to the order of sample data, but also it may make out the local excellent and be affected by the outliers. Given these shortages, an improved algorithm is discussed, which makes improvements in data preprocessing and selection of original clustering center. Check experiment was done, which indicates the improved one is more stable, more accurate and the affection by the outliers is down to a much low figure.

Key words: k-means; clustering; outliers

0 引言

聚类就是根据最大化类内的相似性、最小化类间的相似性原则对数据对象进行分组的一个过程。其结果就是一个个由数据对象组成的簇,每个簇内的对象之间具有很高的相似性,而簇间的对象则很不相似。聚类的应用越来越广泛[1],在经济学、生物学、气象学、医药学、信息工程和工程技术等许多领域都有着十分重要的作用。因此,对聚类的要求也越来越高,提出准确且又高效的聚类算法刻不容缓。

人们已经提出了很多聚类算法,比如有基于划分的 K-MEANS[2]算法、CLARANS 算法;基于层次的 BIRCH 算法、CURE[3]算法;基于密度的 DBSCAN 算法、OPTICS 算法等。但是这些算法都存在着不足,比如 DBSCAN 算法,它的密度是一个核心对象的 ε领域内数据对象的个数,所以就存在如何选择密度参数的问题,不适当的选择将会大大影响算法的结果。本文主要是针对 K-means 算法的不足提出了改进算法。

1 相关工作

1.1 K-means 的基本思想

给定类的个数 k,随机挑选 k 个对象为初始聚类中心,利用距离最近的原则,将其余数据集对象分到 k 个类中去,聚类的结果由 k 个聚类中心来表达。算法采用迭代更新的方法,通过判定给定的聚类目标函数,每一次迭代过程都向目标函数值减少的方向进行。在每一轮中,依据 k 个参照点将其周围的点分别组成 k 个类,而每个类的几何中心将被作为下一轮迭代的参照点,迭代使得选取的参照点越来越接近真实的类几何中心,使得类内对象的相似性最大,类间对象的相似性最小。

K-means 的伪代码描述如下:

```
//输入:类的个数 k,样本数 n:
    随机选取 k 个对象,初始化 k 个聚类中心;
    设置迭代计数器 t=0;
While(r≠0)
    把样本点分到距离最近的聚类中心所代表的簇内;
    计算聚类目标函数 J(t);
    r=J(t)-J(t-1);
```

收稿日期:2007-09-30;
作者简介:连凤娜(1983-),女,硕士研究生,研究方向:数据挖掘;吴锦林(1946-),男,教授,主要研究方向:网络与数据库应用;唐 琦(1984-),女,硕士研究生,研究方向:系统结构。

重新计算各个聚类中心;

t=t+1;

输出聚类中心。

关于聚类目标函数及聚类中心的定义见参考文献[4]。

1.2 K-means 的不足之处

k-means 算法有四个缺点[2]:

(1) 通过计算样本间的距离来衡量它们间的相似度, 而大值属性往往左右样本间的距离, 所以该算法不适用于有大值属性存在的数据集。

(2) 对初始聚类中心和样本的输入顺序敏感, 不同的初始聚类中心或样本的输入顺序不同, 产生的聚类结果差别很大。

(3) 采用迭代更新的方法, 所以当初始聚类中心落在局部值最小附近时, 算法的结果生成的是局部最优解而非全局最优解。

(4) 采用同一类中的所有对象的平均值作为聚类中心, 所以算法的效果受孤立点的影响很大。

1.3 改进的 k-means 算法

针对上面的问题, 本文提出了两处改进: 一是样本数据预处理, 二是初始聚类中心的选择。

聚类的目的就是对数据对象进行区分和归类, 相似性大的对象归在同一类中, 不同类中的对象差别较大。如果用对象间的距离来描述它们的相似性, 那么就是距离越大的两点之间差别越大, 距离越小的两点之间越相似。在文献[5]中, 作者提出了基于距离和的孤立点定义, 实验验证效果是不错的。在本文中我们借用这个定义来降低孤立点对聚类结果的影响。

在选择初始聚类中心上我们采用了以下方法: 先计算样本对象两两之间的距离, 再筛掉 m 个与其他对象之间的距离和最大的对象, 然后从剩下的数据集合中选出距离最大的两个点作为两个不同类的聚类中心, 接着从其余的样本对象中找出已经选出来的所有聚类中心的距离和最大的点为另一个聚类中心, 直到选出 k 个聚类中心。这样得到的初始聚类中心不受样本的输入顺序影响, 因为筛选是基于样本间的距离进行的, 降低了孤立点的影响, 所以事先排除了可能是孤立点的对象, 这些将在后面的实验中给出验证。

在此之前, 为了防止某些大值属性左右样本间的距离, 要先对样本数据进行正规化处理。$X_j'=(X_j-\mathrm{avg}(X_j))/\sigma(X_j)$, 其中 $\mathrm{avg}(X_j)$是均值, $\sigma(X_j)$是均方差值, X_j' 是 X_j 处理后的值, 下面举例说明。

假设有一个样本数据集, 有 11 个样本对象, 它们的分布如图 1 所示, 现在采用改进后的算法对此数据集进行聚类。假设聚类个数 k 为 3, 可能的孤立点数 m 取 1, 首先筛掉与其它样本的距离和最大的点 a, 然后从剩下的数据集{b, c, d, e, f, g, h, i, j, k}中选出距离最远的两个点 i 和 d, 因为点 g 到点 i 和 d 的距离和最大, 所以 g 为另一个类的中心; 接着根据样本与两个中心点之间的距离进行聚类, 所以{h, j, b}归到 i 类, {c, e}归到 d 类, {k, f}归到 g 类; 接下来计算各类聚类中心, 新的 i 类聚类中心就是{h, j, i, b}四个点的平均值, d 类聚类中心就是{c, e, d}三个点的平均值, g 类聚类中心就是{k, f, g}三个点的平均值, 然后以新的聚类中心为参照点展开新一轮的迭代, 直到聚类目标函数不再发生改变。

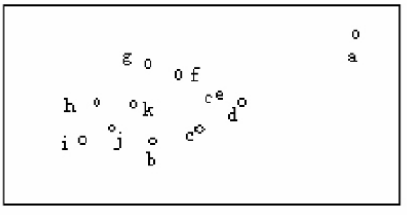

图 1　样本对象的分布

2 改进后的算法实现及实验分析

2.1 实现算法

改进后的 k-means 伪代码描述如下:

//输入: 类的个数 k, 样本数 n, 可能孤立点数 m;

对样本数据进行正规化处理;

计算样本对象两两间的距离 dist[i][j];

筛掉与其它所有样本的距离和最大的 m 个对象;

初始化聚类中心集合 center=Φ;

选出值最大的 dist[i][j], 则 center={i,j};

for(h=2;h<k; h++)

找出与 center 中的所有对象距离和最大的对象 t, 把 t 加入到 center 中;

设置迭代计数器 t=0;

While(r≠0)

把样本点分到距离最近的聚类中心所代表的簇内;

计算聚类目标函数 J(t);

r=J(t)-J(t-1);

重新计算各个聚类中心;

t=t+1;

输出聚类中心。

2.2 实验分析

本文的实验数据采用 UCI 数据库(http://www.ics.uci.edu/~mlearn/MLSummary.html) 中的 Iris 数据集和 wine 数据集, 其中 Iris 数据集包含有 4 个属性, 150 个数据对象, 可分为 3 类; Wine 数据集包含 14 个属性

(其中第一个属性是类别标识，实验中我们只选取后13个属性),178个数据对象,可分为3类。

下面通过两组实验对改进后的算法和改进前的算法进行测试和比较，为了方便，我们只选取Iris的第三、第四维(petal length、petal width)作为实验对象。

2.2.1 实验一

该实验的目的是在没有噪声的影响下，比较改进前后算法的稳定性和准确性。

参数设置如下:原k-means算法,k取3;改进后的算法,k也取3,m取0。实验结果见表1:

表1 无孤立存在的聚类结果

算法	初始聚类中心	错分数	
		Iris数据集	Wine数据集
K-means 1	(1.125, 0.175) (1.523, 0.270) (4.925, 1.682)	11	53
K-means 2	(1.810, 0.385) (4.161, 1.267) (5.281, 1.861)	5	55
K-means 3	(2.770, 0.752) (5.133, 2.100) (5.706, 1.942)	8	76
K-means 4	(1.810, 0.385) (4.413, 1.374) (5.549, 2.024)	8	57
K-means 5	(1.492, 0.263) (4.452, 1.437) (5.791, 2.129)	8	53
K-means 6	(1.293, 0.267) (1.666, 0.300) (4.959, 1.696)	9	53
K-means 7	(1.560, 0.291) (3.667, 1.083) (5.044, 1.736)	6	56
K-means 8	(1.462, 0.246) (4.333, 1.372) (5.665, 2.079)	9	53
K-means 9	(2.186, 0.528) (4.676, 1.532) (5.715, 2.092)	8	53
改进后算法	(1.489, 0.181) (2.236, 0.594) (5.092, 1.766)	6	16

从实验结果可以看出，原k-means算法随机选出的初始聚类中心相当的不稳定，从而影响到最后聚类结果的稳定性，而且对于有大值属性存在的Wine数据集,错分数大大的增加;而改进后的算法,不受大值属性的影响,而且错分数较少。

2.2.2 实验二

为了测试两个算法受孤立点的影响程度，我们在原有的Iris数据集中任意加入了5个孤立点，分别为(1.5,5),(20,0.2),(0,0),(14,0),(1.4,9),同样,也在Wine数据集中加入5个孤立点,分别为(0,0,0,0,0,0,0,0,0,0,0,0,0,0),(13.34,94,2.36,1.7,110,0.55,0.42,3.17,1.02,1.93,750),(14.34,1.68,2.7,25,98,2.8,31,0.53,2.7,13,0.57,1.96,660),(14.2,1.76,2.45,15.2,1.12,3.27,3.39,0.34,1.97,6.75,1.05,2.85,450),(12.67,0.98,2.24,18,99,2.2,1.94,0.3,1.46,2.62,123,3.16,450),这些孤立点是根据可能的输入错误创造出来的，它们均匀地分布在数据集中。

参数设置如下:k-means算法,k取3;改进后的算法,k也取3。实验结果见表2。

比较实验一和实验二的结果,K-means算法在有孤立点存在的情况下,错分数大大地增加,而且初始聚类中心更加的不稳定，即K-means算法受孤立点的影响很大。改进后的算法只要我们的m参数选择得当(一般m≥孤立点的实际数目)，就可以大大的降低孤立点的影响。

表2 有孤立点存在的聚类结果

算法	初始聚类中心	错分数	
		Iris数据集	Wine数据集
K-means 1	(1.162, 0.177) (1.520, 0.271) (5.069, 1.747)	40	53
K-means 2	(2.150, 0.589) (5.138, 1.905) (17, 0.100)	54	73
K-means 3	(1.423, 0.246) (4.368, 1.374) (5.956, 2.220)	20	52
K-means 4	(2.836, 0.780) (5.393, 2.533) (6.300, 1.629)	52	59
K-means 5	(5.207, 1.622)(5.071, 3.385)(1.600, 0.405)	57	52
K-means 6	(1.600, 0.406) (4.856, 1.773) (9.528, 1.471)	53	53
K-means 7	(1.179, 1.178) (3.875, 1.260) (9.100, 1.525)	53	78
K-means 8	(2.305, 0.578) (4.464, 1.618) (5.860, 2.067)	28	52
K-means 9	(3.096, 1.028) (5.643, 2.077)(9.443, 1.643)	53	78
改进后算法 m=3	(1.423, 0.246) (4.900, 1.673) (1.5, 5.0)	52	68
改进后算法 m=4	(1.423, 0.246) (4.670, 1.567) (6.075, 2.219)	9	69
改进后算法 m=5	(1.453, 0.251) (4.662, 1.518) (5.824, 2.280)	8	11
改进后算法 m=7	(1.475, 0.183) (2.091, 0.536) (5.014, 1.733)	8	9
改进后算法 m=9	(1.461, 0.254) (4.645, 1.552) (5.979, 2.173)	8	9

3 结论

K-means算法计算速度快,资源消耗小,对于处理大数据集是相对可伸缩的和高效的,但是对样本的输入顺序敏感,有可能产生局部最优解,而且容易受到孤立点的影响。本文针对这些不足提出了一种改进的K-means算法,从数据预处理和初始聚类中心的选择方面作了改进，通过实验证明该算法比K-means算法更具稳定性和准确性。

参考文献:

[1] Han J, Kamber M. 数据挖掘概念与技术[M]. 范明，孟小峰译.北京:机械工业出版社, 2001.

[2] Kaufan L, Rousseeuw Pj. Finding Groups in Data: an Introduction to Cluster Analysis[M]. New York: John Wiley & Sons, 1990.

[3] Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for large databased[C]. In Haas LM, Tiwary A eds. Proceedings of the ACM SIGMOD International Conference on Management of Data, Sesttle: ACM Press, 1998:73-84.

[4] 金微.基于遗传算法的k-means聚类方法的研究[D].南京:河海大学, 2007.

[5] 陆声链,林士敏.基于距离的孤立点检测研究[J].计算机工程与应用, 2004, 33: 73-75.

[6] 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, 36: 177-178.