一种高效的 K-medoids 聚类算法

夏宁霞[1], 苏一丹[1], 覃 希[1,2]
(1. 广西大学 计算机与电子信息学院, 南宁 530004; 2. 广西工学院 计算机工程系, 广西 柳州 545006)

摘 要: 针对 K-medoids 算法初始中心点选择敏感、大数据集聚类应用中性能低下等缺点,提出一个基于初始中心微调与增量中心候选集的改进 K-medoids 算法。新算法以微调方式优化初始中心,以中心候选集逐步扩展的方式来降低中心轮换的计算复杂性。实验结果表明,相对于传统的 K-medoids 算法,新算法可以提高聚类质量,有效缩短计算时间。

关键词: 聚类; K-medoids 算法; 中心微调; 增量候选
中图分类号: TP301 **文献标志码**: A **文章编号**: 1001-3695(2010)12-4517-03
doi:10.3969/j.issn.1001-3695.2010.12.035

Efficient K-medoids clustering algorithm

XIA Ning-xia[1], SU Yi-dan[1], QIN Xi[1,2]
(1. *School of Computer & Electronic Information*, *Guangxi University*, *Nanning* 530004, *China*; 2. *Dept. of Computer Engineering*, *Guangxi University of Technology*, *Liuzhou Guangxi* 545006, *China*)

Abstract: Due to the disadvantages of sensitivity to the initial selection of the medoids and poor performance in large data set processing in the K-medoids clustering algorithm, this paper proposed an improved K-medoids algorithm based on a fine-tuned of initial medoids and an incremental candidate set of medoids. The proposed algorithm optimized initial medoids by fine-tuning and reduced computational complexity of medoids substitution through expanding medoids candidate set gradually. Experimental results demonstrate the effectiveness of this algorithm, which can improve clustering quality and significantly shorten the time in calculation compared with the traditional K-medoids algorithm.

Key words: clustering; K-medoids algorithm; medoid fine-tuning; incremental candidate

0 引言

K-medoids 算法是一种基于划分的聚类算法,具有较强的鲁棒性和较高的准确性,且相对于 K-means 算法有着明显的优势:a)在 K-means 算法中,用质心来代表簇,导致其对噪声和孤立点数据非常敏感,而 K-medoids 算法用簇中最靠近中心的一个对象(即中心点)来代表该簇,可以有效地消除这种影响[1,2]。b)K-means 算法只有在簇的平均值被定义的情况下才能使用[2]。然而,K-medoids 算法也存在一些缺点:一方面与 K-means 算法一样,它存在初始化敏感、聚类结果多样化的问题[3,4];另一方面,该算法在进行中心轮换时需遍历所有非中心点,执行代价较高。鉴于此,目前对该算法的研究主要从优化初始中心及中心替换策略两方面来进行改进。文献[5]通过计算各对象的距离关系选择初始中心,并将中心替换的范围局限于簇内,其提高了算法的效率,但聚类效果却下降了。文献[6]通过计算对象的密度信息来产生初始中心,然后执行传统的 K-medoids 算法,取得了较好的聚类效果,但是其基于传统算法的搜索策略使得该算法仍存在效率低的缺陷。其他的改进方法如基于 CF tree(clustering feature tree)[7]、基于 MAM(metric access methods)[8]等,在改进算法性能的同时却因额外组件增加了算法的复杂性。

因此,针对 K-medoids 算法在应用过程中存在的问题,结合目前对该算法的研究现状,本文提出了一个基于初始中心微调与增量中心候选集的改进 K-medoids 算法。它将在随机选择初始中心并进行首次划分后在簇内进行中心调整,然后以中心候选集逐一扩大的方式进行中心轮换,直至收敛。该算法不仅延续了原始 K-medoids 算法的优势,保证了聚类效果的优越性与稳定性,同时算法效率也取得了较大程度的提高。

1 改进的 K-medoids 算法

1.1 K-medoids 算法介绍与分析

K-medoids 算法有多种形式,其中 PAM(partitioning around medoids,围绕中心点的划分)是最早提出的 K-中心点算法之一[1]。传统的 K-medoids 算法(以 PAM 为代表)的基本思想[1,9]是:首先为每个簇随意选择一个代表对象,剩余的对象根据其与代表对象的距离分配给最近的一个簇;然后反复地用非代表对象来替代代表对象,以改进聚类的质量聚类结果的质量用一个代价函数来估算,该函数度量对象与其参照对象之间的平均相异度。

聚类的目标是使簇内差异尽可能小,所以,为了判定一个非代表对象 o_{random} 是否是当前一个代表对象 o_i 的好的替代,对

收稿日期: 2010-05-25; **修回日期**: 2010-06-28

作者简介:夏宁霞(1983-),女,江西九江人,硕士研究生,主要研究方向为数据挖掘、Web 个性化标签推荐(klxnx@163.com);苏一丹(1962-),男,教授,硕导,博士,主要研究方向为电子商务、Web 个性化挖掘、信息安全;覃希(1983-),女(壮族),广西南宁人,助教,硕士研究生,主要研究方向为 Web 标签挖掘.

于每一个非中心点对象 P,每当重新分配发生时,平方—误差 $E=\sum_{i=1}^{k}\sum_{p\in C_j}|p-o_i|^2$ 所产生的差别对代价函数有影响。替换的总代价是所有非中心点对象所产生的代价之和。如果总代价是负的,那么实际的平方—误差将会减小,即表明替换后的簇内差异变小了,o_i 可以被 o_{random} 替代。如果总代价总是正的或为零,表明算法不能产生一个有效的替换了,即此时算法收敛。

K-medoids 算法是基于使聚类准则函数最优的原则,采用最接近于聚类中心的数据点作为类的中心以增强算法的鲁棒性,对小的数据集合非常有效。但其在处理过程中还存在着以下问题:

a)传统的 K-medoids 算法对初始中心是随机选择的,从而导致聚类结果会随着初始点选择的变化而发生改变,即其对初始中心敏感,且容易陷入局部最优解。所以要想获得较好的聚类结果,可以从初始中心的选择上进行改进,降低随机性,以达到优化的目的。

b)在中心替换策略上,原始的 K-medoids 算法采取全局搜索的策略,即在中心替换时,考虑所有的非代表对象,根据相异度函数重新划分,保证了聚类精度,但搜索效率却很低。然而,针对这个问题,Park 等人[5]提出将替换范围限制在簇内,具有较高的搜索效率,但会遗漏一些可能的更优替换点,从而使聚类精度得不到保证。因此,若需保证聚类精度的同时提高搜索效率,可以结合它们的优势,先从簇内开始查找,采用逐步递增的方式,最终将搜索范围扩大至所有非代表对象,以寻找最优替换对象,保证聚类质量。

由上面的分析可知,改进的算法可从优化初始中心与改进中心替换策略两方面进行,以此来提高聚类质量与算法的运行效率。

1.2　新算法的改进思路

针对传统的 K-medoids 算法存在的一系列问题,改进算法的具体改进思路如下:

a)初始中心微调。在随机产生初始中心并将所有对象进行首次划分后,对各初始中心在簇内进行微调,使之向较优的中心偏移。具体方法为:以簇内各对象轮换为此簇中心,计算该中心与其他所有对象的距离和,选取距离和最小的对象为此簇微调后的中心。

b)采用基于增量中心候选集的中心替换策略。对于一个中心点,若要对其进行替换,一些较优的候选点最有可能出现在该中心点附近,如同一簇内或其附近簇中的点。因此在搜索新的中心点时可以只在这些范围内进行。具体替换方法为:中心点替换分 k(k 为目标聚类数目,下同)次进行,每次进行替换前首先生成中心候选集,候选集在迭代过程中不断扩大。设当前为第 i 次,在对上一次的 k 个中心点进行替换时,其搜索的范围设定为离该中心最近的 i 个簇(包含被替换中心对应簇)内的所有非中心对象。这样新中心的候选集就从 1,2,…,k 个簇不断增大, 形成增量中心的候选集,其最后的搜索范围将扩大到所有的非中心点集,与原始算法相一致。

1.3　新算法描述

根据改进思路,设计了一个新的 K-medoids 算法,其算法流程描述如下:

输入:结果簇的数目 k,包含 n 个对象的数据库。
输出:k 个簇,使得所有对象与其最近中心点的相异度总和最小。
1. 选择、优化初始中心点并作初步划分
　1.1. 在 n 个对象中随意选择 k 个对象作为初始的中心点;
　1.2. 以当前中心点对其他对象进行初始划分;
　1.3. 在划分得的各簇中对中心点进行微调(按改进思路(a));
　1.4. 以当前中心点对其他对象进行划分;
2. 执行增量中心候选集 K-medoids 算法
　令 $i=1$;
　while $i<=k$
　2.1. 轮换中心并计算代价
　　repeat
　　选择一个未选择过的中心点 O;
　　生成替换 O 的候选集 C(C 由距 O 最近的 i 个簇的所有非中心对象构成);
　　2.1.1. 替换当前中心
　　repeat
　　在 C 中选择一个未选择过的非中心点 P,计算用 P 代替 O 的代价并记录在 S 中;
　　untill C 中所有的非中心点都被选择过;
　　untill 所有的中心点都被选择过;
　2.2. 实施中心替换
　if min(S) <0(设 min(S)表示代价集 S 中最小值)
　　2.2.1. 用 min(S)对应的非中心点与中心点进行中心替换,形成一个新的 k 个中点的集合;
　　2.2.2. 指派 $n-k$ 个剩余的对象给离它最近的中心点所代表的簇;
　　2.2.3. 重新从 2.1 开始执行;
　else
　　$i=i+1$
　end if
　end while
3. 算法结束

对于本文改进的新算法,通过初始中心的调整,使得更多的中心点极有可能就是最后聚类结果对应的中心点,可减少算法迭代的次数,提高算法的执行效率。然而通过整体的调整,将得到相对较优的初始中心,从一定程度上可提高聚类质量。

另外,从时间复杂度上分析,起决定作用的是算法流程 2,在这个阶段,新算法进行 k 次搜索范围不同的中心轮换过程。对一般情况,设聚类的数据点分布均匀,即每次对数据进行划分后各簇中数据点的数量相当,约为 $(n-k)/k$。根据算法流程,在进行第 $i(i=1,2,\cdots,k)$ 轮替换时,将以 i 个最近的簇组成中心候选集,此时该候选集中数据点的数量为 $i\times(n-k)/k$。在算法的 2.1.1 中,当一一以该候选集中的一个点替换当前中心点,并对剩余的 $n-k$ 个非中心点进行簇划分时,算法的时间复杂度为 $O(i\times(n-k)^2/k)$。所以,对所有的 k 个中心点进行替换与验证(算法流程 2.1)的时间复杂度为 $O(i(n-k)^2)$。而根据算法流程 2,上述整个替换过程需进行 k 次,则总的时间复杂度为 $O(\sum_{i=1}^{k}i(n-k)^2)$,即 $O(k(k+1)(n-k)^2/2)$。根据文献[1]的比较准则,需计算每轮迭代的时间复杂度,即平均时间复杂度。改进算法一共进行 k 轮,则其平均时间复杂度为 $O((k+1)(n-k)^2/2)$。与传统算法的平均时间复杂度 $O(k(n-k)^2)$[1]比较,尽管没有数量级上的降低,但其拥有更小的系数,因此其必然能节省执行时间,后面的实验将对此进行验证。

2　实验

2.1　实验设计

为了检验该算法的有效性,实验采用传统的 K-medoids 算

法(简称 KM)与本文改进的 K-medoids 算法(简称 IKM)进行对比实验,且在聚类性能和算法运行时间两方面进行比较。该实验是在配置 2.69 GHz Pentium® E5400 CPU、2 GB 内存的 PC 机上进行,操作系统为 Windows XP,程序用 MATLAB 编写。

实验中聚类性能将采用 f-measure 值来衡量。f-measure 组合了信息检索中查准率(precision)与查全率(recall)的思想。一个聚类 j 相对分类 i 的 precision 和 recall 分别定义为 $P=\text{precision}(i,j)=\frac{n_{ij}}{n_i}$,$R=\text{recall}(i,j)=\frac{n_{ij}}{n_j}$。其中:$n_{ij}$ 是在聚类 j 中分类 i 的数据;n_i 是分类 i 中所有对象的数目;n_j 是聚类 j 中所有对象的数目。分类 i 的 f-measure 值定义为 $F=\frac{2PR}{(P+R)}$。对分类 i 而言,f-measure 值最高的聚类用来表示分类 i 的映射,即该 f-measure 值表示分类 i 的评判分值。就聚类结果而言,最终的 f-measure 值由分类 i 的加权平均 f-measure 值得到:$F=\frac{\sum_i(|i|\times F(i))}{\sum_i|i|}$。其中 $|i|$ 为分类 i 中所有对象的数目。由此可知,f-measure 值越大,代表聚类结果越好,所获得的簇与原类别越吻合。

2.2　实验结果与分析

实验的数据集采用 UCI 公共数据库提供的四个数据集[10],对于每个数据集合将数据标度到[-1,1]的范围内,所有结果皆取 10 次实验的平均值。测试的四个数据集如表 1 所示。

表 1　实验数据集基本情况

数据集	类别数	属性个数	样本总数
soybean	4	35	47
zoo	7	17	101
wine	3	13	178
breast cancer(BC)	2	10	699

在四个数据集中,soybean 与 zoo 的数据类型包括二元变量与分类变量,用海明距离[1]表示其数据对象进行相异度;wine 与 breast cancer(BC)的数据类型为区间标度变量,用欧几里德距离[1]度量数据库中对象之间的相异度。

表 2 是传统 K-medoids 算法与本文算法在聚类效果上的实验对比情况。

表 2　f-measure 值对比情况

数据集	f-measure 值		f-measure 值
	KM	IKM	提高百分比/%
soybean	0.811 2	0.815 4	0.517 8
zoo	0.745 1	0.758 7	1.825 3
wine	0.896 9	0.899 2	0.256 4
breast cancer	0.956 9	0.956 9	0

由表 2 可知,新算法所获得的聚类结果的 f-measure 比原始算法有所提升,表明改进后的算法不仅能够保证聚类的性能,并能获得更好的聚类效果。特别针对类别数与属性个性相对较多的 zoo 数据集,其改进效果更加明显,而对于 breast cancer 这种聚类效果较好的数据集能够维持原来的精度,验证了新算法在聚类效果上的稳定性。

表 3 为两算法在运行时间上的对比情况。从表 3 可知,与 KM 比较,IKM 在运行时间上表现出了明显的优势,对每一数据集都能减少一半左右的运行时间。对于不同的数据集其算法效率会受数据集类别数与样本数的影响,随着分类数与样本数的增加,聚类所耗费的时间也大幅度增加,上面的实验对比情况可以对此验证。表 3 的实验结果与上文对算法时间复杂度的分析相吻合,从另一个角度说明了 IKM 的性能的稳定性。

表 3　算法效率的对比情况

数据集	运行时间/s		运行时间
	KM	IKM	节省百分比/%
soybean	0.510 9	0.234 0	54.198 5
zoo	4.572 4	2.269 3	50.369 6
wine	4.133 0	1.721 2	58.354 7
breast cancer	25.269 1	12.630 6	50.015 6

综上所述,改进的新算法在同样的实验环境下,既能保证聚类性能又能大幅度提高算法效率。上述实验结果验证了新算法的可行性与有效性。

3　结束语

在原始的 K-medoids 算法进行聚类时,会遇到初始中心敏感、聚类结果不稳定、执行效率低下等问题,为此本文提出了一种高效的 K-medoids 改进聚类算法。该算法通过调整初始中心,并在中心替换操作过程中采用增量式思想逐步扩大搜索范围。与原始聚类算法相比,本文算法不仅能改进聚类精度而且能提高算法的效率,这对促进 K-medoids 算法的发展与应用具有积极的意义。

参考文献:

[1] HAN Jia-wei, KAMBER M. 数据挖掘概念与技术[M]. 2 版. 北京:机械工业出版社,2008:263-266.

[2] CHEN Xin-quan, PENG Hong, HU Jing-song. K-medoids substitution clustering method and a new clustering validity index method [C]//Proc of the 6th World Congress on Intelligent Control and Automation. 2006:5896-5900.

[3] HE Zeng-you. Farthest-point heuristic based initialization methods for K-modes clustering[EB/OL]. (2006-10-10). http://arxiv.org/ftp/cs/papers/0610/0610043.pdf.

[4] 宗瑜,金萍. BK-means:骨架初始解 K-means [J]. 计算机工程与应用, 2009, 45(14):49-52.

[5] PARK H S, JUN C H. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems with Applications, 2009,36(2):3336-3341.

[6] PARDESHI B, TOSHNIWAL D. Improved K-medoids clustering based on cluster validity index and object density[C]//Proc of the 2nd IEEE International Advance Computing Conference. 2010:379-384.

[7] GAO Dan-yang, YANG Bing-ru. An improved K-medoids clustering algorithm[C]//Proc of the 2nd International Conference on Computer and Automation Engineering(ICCAE). 2010:132-135.

[8] BARIONI C N M, RAZENTE H L, TRAINA A J M, *et al*. Accelerating K-medoid-based algorithms through metric access methods[J]. The Journal of Systems and Software,2008, 81(3):343-355.

[9] PARTYKA J, KHAN L, THURAISINGHAM B. Semantic schema matching without shared instances[C]//Proc of IEEE International Conference on Semantic Computing. 2009:297-302.

[10] DataSet[EB/OL]. (2010). http://archive.ics.uci.edu/ml/datasets.html.