

高效率的 K -means 最佳聚类数确定算法

王 勇*, 唐 靖, 饶勤菲, 袁巢燕

(重庆理工大学 计算机科学与工程学院, 重庆 400054)

(* 通信作者电子邮箱 ywang@cqut.edu.cn)

摘 要:针对 K -means 聚类算法通常无法事先设定聚类数,而人为设定初始聚类数目容易导致聚类结果不够稳定的问题,提出一种新的高效率的 K -means 最佳聚类数确定算法。该算法通过样本数据分层来得到聚类数搜索范围的上界,并设计了一种聚类有效性指标来评价聚类后类内与类间的相似性程度,从而在聚类数搜索范围内获得最佳聚类数。仿真实验结果表明,该算法能够快速、高效地获得最佳聚类数,对数据集聚类效果良好。

关键词: K -means 聚类; 数据分层; 聚类有效性指标; 相似性程度; 最佳聚类数

中图分类号: TP393 **文献标志码:** A

High efficient K -means algorithm for determining optimal number of clusters

WANG Yong*, TANG Jing, RAO Qingfei, YUAN Chaoyan

(College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China)

Abstract: The cluster number is not generally set by K -means clustering algorithm beforehand, and artificial initial clustering number easily leads to the problem of unstable clustering results. A high-efficient algorithm for determining the K -means optimal clustering number was presented. The algorithm got the upper bound of the number of clustering search range through stratified sample data and designed a new kind of effective clustering indicator to evaluate the clustering degree of similarity between and within class after clustering. Thus the optimal number of clusters was obtained in the search range of the clusters number. The simulation results show that the algorithm can obtain the optimal clustering number fast and accurately, and the dataset clustering effect is good.

Key words: K -means clustering; data stratification; clustering validity index; degree of similarity; optimal number of clusters

0 引言

聚类是一个将整体的数据对象划分为以类或簇存在的包含局部数据对象的过程。聚类^[1-3]源于数据挖掘、统计学、生物学、机器学习等众多领域,现如今聚类分析已经广泛应用于模式识别、数据分析以及图像处理等研究领域。经过专家学者的研究,目前的聚类算法可以归纳为如下几类:划分方法、层次方法、基于密度的方法、基于网格的方法、基于模型的方法和高维数据的方法。其中 K 均值聚类算法^[4] (K means clustering algorithm, K -means) 是基于划分的经典聚类算法之一,因其简洁、高效而得到了广泛的应用。但是 K -means 聚类算法也不可避免地存在缺点:无法事先确定合适的聚类数目,导致聚类质量不高。获取良好聚类效果关键在于确定最佳的聚类数目。为克服传统 K -means 算法的不足,文献[5]提出采用类内距离和类间距离的比值作为评价准则函数,将准则函数取得最小值时对应的聚类数作为最佳聚类数。该算法有效解决了用户在缺乏经验时对样本聚类数随机确定的问题,但随着样本数据量的增大,算法的运行时间也随之增加。针对大多数聚类算法要求事先给定聚类数目的难题,文献[6]提

出利用二分思想递归分裂簇内相似度大于给定阈值的簇,同时,合并簇间相似度小于给定阈值的簇来获得最终聚类数目。该算法有效解决了聚类数无法事先确定的问题,但是该算法簇内相似度阈值 λ 和簇间相似度阈值 γ 的确定是个难题,取值过高或过低都会影响聚类的效果和质量。文献[7]提出一种新的最佳聚类数方法,该算法利用近邻传播 (Affinity Propagation, AP) 聚类算法产生的聚类数 K_{\max} 作为聚类数搜索范围的上界,并运用 Sil (Silhouette) 指标分析聚类效果,确定最佳聚类数。但 AP 算法对于比较松散的聚类结构,倾向于产生较多的局部聚类,使得算法产生的聚类数往往偏多,最终不能给出准确的聚类结果。

针对上述算法在确定最佳聚类数时都存在一定的问題,本文设计了一种新的聚类有效性指标,并在此基础上,提出一种基于 K -means 高效率的最佳聚类数确定算法。通过对样本数据进行阈值分层快速确定 K -means 算法的聚类数搜索范围上限,并确定聚类数搜索范围,利用新的聚类有效性指标评价聚类后类内与类间的相似性程度,从而在聚类数搜索范围内获得最佳聚类数。

收稿日期: 2013-11-27; 修回日期: 2013-12-25。

基金项目: 重庆市教委资助项目 (KJ100821); 重庆理工大学研究生创新基金资助项目 (YCX2013218)。

作者简介: 王勇 (1974 -), 男, 重庆人, 副教授, 博士, 主要研究方向: 多媒体、网络; 唐靖 (1988 -), 女, 湖南永州人, 硕士研究生, 主要研究方向: 图像处理; 饶勤菲 (1990 -), 男, 江西吉安人, 硕士研究生, 主要研究方向: 图像处理; 袁巢燕 (1987 -), 女, 安徽合肥人, 硕士研究生, 主要研究方向: 无线传感器网络、嵌入式技术。

1 相关工作

1.1 K-means 算法

K-means 聚类算法的基本思想: 首先随机选择 k 个初始聚类中心, 人为确定分簇数目 k ; 遍历每一个点, 计算每一个点到 k 个初始聚类中心的欧氏距离, 通过比较知道该点离哪个初始聚类中心最近, 并把该点与其初始聚类中心归为一簇; 将点集分好簇以后, 重新确定聚类中心; 当簇内数目不发生变化或者达到了最大的迭代次数时, 算法结束。

K-means 聚类算法的优点是对大数据集合聚类效果明显, 聚类快速且易于实现。同时 K-means 算法也存在一定的局限性, 如对“噪声”和孤立点(异常点)敏感; 无法掌握数据分布情况, 人为设定分簇数目, 需增加迭代次数以获取良好的聚类效果; 算法时间复杂度较高, 经常以局部最优结束, 难以达到全局最优。

1.2 确定最佳聚类数的相关算法

为解决 K-means 算法无法事先准确确定聚类数目的问题, 周世兵等^[7]提出了新的 K-means 算法最佳聚类数确定方法。该算法的基本思想如下: 针对给定的数据集, 首先确定聚类数的搜索范围 $[k_{\min}, k_{\max}]$, 一般取 $k_{\min} = 2, k_{\max} = k_{AP}$, 其中 k_{AP} 为 AP 算法产生的聚类数搜索范围上界; 然后通过运行 K-means 算法得到不同聚类数目所对应的聚类结果; 最后根据 Silhouette 有效性指标对聚类结果评估, 有效性指标最大值对应的类数为最佳聚类数 K_{opt} 。该算法的流程如图 1 所示。

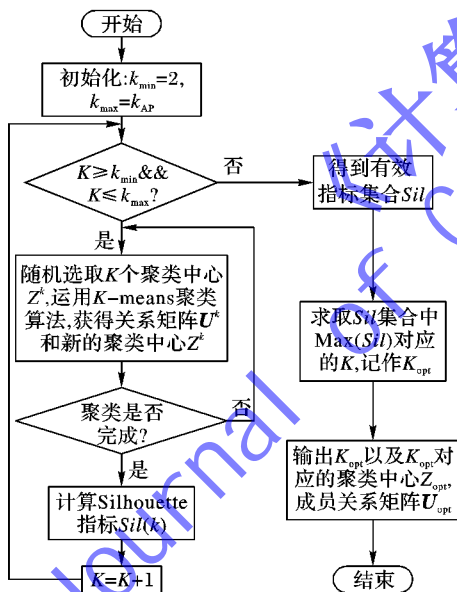


图1 确定最佳聚类数的算法流程

由于通过该算法确定聚类数搜索范围的时间复杂度较高, 导致算法运行效率不高。因此本文需要重新确定聚类数搜索范围的上限, 从而最终确定最佳聚类数。

2 新的确定最佳聚类数的算法

2.1 样本数据阈值分层方法

本文样本数据分层方法如下:

- 1) 固定样本数据的某一列, 利用直方图方法统计其数据分布。
- 2) 将一定阈值范围内的点近似归为一个层面, 设定合适

的阈值 T ; 确定分层数目 k_{\max} , 即为聚类数搜索范围的上界。

3) 根据层内点的法向量是否平行, 证明该数据分层方法所分离的各组点是否在同一平面内。

2.2 新聚类有效性指标

聚类有效性指标^[7]是衡量聚类算法产生的聚类结果是否达到最优的标准, 该指标将最优聚类结果所对应的聚类数作为最佳聚类数。目前已有的聚类有效性指标主要有 CH (Calin-ski-Harabasz)^[8]指标、Wint (Weighted inter-intra)^[9]指标和 Sil (Silhouette)^[10]指标等。鉴于以上指标不一定都能快速准确地确定最佳聚类数, 本文提出一种新的聚类有效性指标, 该指标可以对 K-means 算法的聚类结果进行评价, 根据评价结果获得最佳聚类数。

2.2.1 新指标及相关定义

定义 1 令待分类的样本数据集为 $S = \{x_1, x_2, \dots, x_n\}$, 每个样本点的维数为 m 维, 假设 n 个样本数据被聚类为 h 类, 聚类中心为 $C = \{c_1, c_2, \dots, c_h\}$, 定义第 j 类的第 i 个样本的最小类间夹角余弦值的平均值为 $bc(j, i)$, 即:

$$bc(j, i) = \min_{1 \leq k \leq h, k \neq j} \left(\frac{1}{n_k} \sum_{p=1}^{n_k} \frac{\sum_{q=1}^m x_{pq}^{(k)} x_{iq}^{(j)}}{\sqrt{\sum_{q=1}^m x_{pq}^{(k)^2}} \sqrt{\sum_{q=1}^m x_{iq}^{(j)^2}}} \right) \quad (1)$$

其中: k 和 j 表示类标, $x_{pq}(k)$ 表示第 k 类的第 p 个样本的第 q 维, $x_{iq}(j)$ 表示第 j 类的第 i 个样本的第 q 维, n_k 表示第 k 类的样本个数。

定义 2 令待分类的样本数据集为 $S = \{x_1, x_2, \dots, x_n\}$, 每个样本点的维数为 m 维, 假设 n 个样本数据被聚类为 h 类, 聚类中心为 $C = \{c_1, c_2, \dots, c_h\}$, 定义第 j 类的第 i 个样本的类内夹角余弦值的平均值为 $wc(j, i)$, 即:

$$wc(j, i) = \frac{1}{n_j - 1} \sum_{t=1, t \neq i}^{n_j} \frac{\sum_{q=1}^m x_{iq}^{(j)} x_{tq}^{(j)}}{\sqrt{\sum_{q=1}^m (x_{iq}^{(j)})^2} \sqrt{\sum_{q=1}^m (x_{tq}^{(j)})^2}} \quad (2)$$

其中: $x_{iq}(j)$ 表示第 j 类的第 i 个样本的第 q 维, 并且 $t \neq i$; n_j 表示第 j 类中的样本个数。

定义 3 令待分类的样本数据集为 $S = \{x_1, x_2, \dots, x_n\}$, 每个样本点的维数为 m 维, 假设 n 个样本数据被聚类为 h 类, 聚类中心为 $C = \{c_1, c_2, \dots, c_h\}$, 定义第 j 类的第 i 个样本的聚类有效性指标为类间夹角余弦值平均值的最小值与类内夹角余弦值的平均值的比值 (Between-Within Angle Cosine Ratio, BWACR), 即:

$$BWACR(j, i) = \frac{wc(j, i)}{bc(j, i)} = \frac{\frac{1}{n_j - 1} \sum_{t=1, t \neq i}^{n_j} \frac{\sum_{q=1}^m x_{iq}^{(j)} x_{tq}^{(j)}}{\sqrt{\sum_{q=1}^m (x_{iq}^{(j)})^2} \sqrt{\sum_{q=1}^m (x_{tq}^{(j)})^2}}}{\min_{1 \leq k \leq h, k \neq j} \left(\frac{1}{n_k} \sum_{p=1}^{n_k} \frac{\sum_{q=1}^m x_{pq}^{(k)} x_{iq}^{(j)}}{\sqrt{\sum_{q=1}^m x_{pq}^{(k)^2}} \sqrt{\sum_{q=1}^m x_{iq}^{(j)^2}}} \right)} \quad (3)$$

2.2.2 新指标及最佳聚类数确定

夹角余弦取值范围为 $[-1, 1]$ 。夹角余弦越大表示两个向量的夹角越小,夹角余弦越小表示两向量的夹角越大。对样本数据集中两个 n 维样本点 $a(x_{11}, x_{12}, \dots, x_{1n})$ 和 $b(x_{21}, x_{22}, \dots, x_{2n})$,利用夹角余弦的概念来衡量它们之间的相似程度。夹角余弦值的计算公式如下:

$$\cos(\theta) = \frac{a \cdot b}{|a| |b|} \quad (4)$$

即:

$$\cos(\theta) = \frac{\sum_{k=1}^n x_{1k} x_{2k}}{\sqrt{\sum_{k=1}^n x_{1k}^2} \sqrt{\sum_{k=1}^n x_{2k}^2}} \quad (5)$$

根据聚类的最佳效果是类内越紧密越好,类间越分散越好,本文算法希望样本的类间夹角余弦值越小越好,即式(1)的 $bc(j, i)$ 越小越好;样本的类内夹角余弦值越大越好,即式(2)的 $wc(j, i)$ 越大越好。通过对聚类后每一类计算该类的类间及类内的夹角余弦值的比值的大小即式(3)来衡量聚类效果的好坏,类内夹角余弦值越大越好,类间夹角余弦值越小越好,其比值取得最大值所对应的聚类数目为最佳聚类数。

为了验证不同聚类数所对应的聚类效果,需计算该聚类数聚类后所有样本的BWACR指标值的平均值,来评价该数据集的聚类效果,通过比较不同聚类数的平均BWACR指标值来确定最佳聚类数。平均BWACR指标值取得最大值所对应的聚类数为最佳聚类数。即:

$$avg_{BWACR}(h) = \frac{1}{h} \sum_{j=1}^k \sum_{i=1}^{n_j} BWACR(j, i) \quad (6)$$

$$h_{opt} = \arg \max_{2 \leq h \leq k_{max}} \{ avg_{BWACR}(h) \} \quad (7)$$

2.3 高效率的最佳聚类数目确定算法

确定最佳聚类数目的算法步骤如下。

1) 对样本数据进行阈值分层确定最佳分层数目作为聚类数范围的上限 k_{max} 。

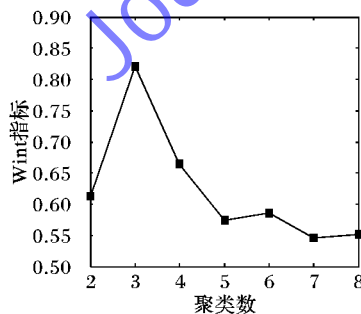
2) 确定聚类数范围: for $k = k_{min} : k_{max}$, $k_{min} = 2$;

①运行K-means聚类算法;

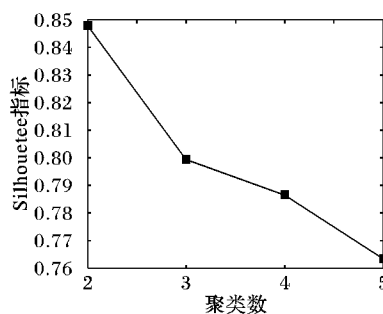
②根据式(1)和式(2)分别计算类间夹角余弦值和类内夹角余弦值;

③根据式(6)计算不同聚类数所对应的类内类间夹角余弦值比值。

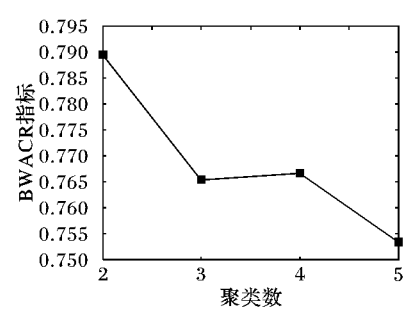
3) 根据式(7)计算类内类间夹角余弦值比值最大时所对



(a) dataset2 聚类数—Wint 指标关系



(b) dataset2 聚类数—Silhouette 指标关系



(c) dataset2 聚类数—BWACR 指标关系

图2 dataset2 聚类数与指标关系

图2显示了分别运用Wint指标、Silhouette指标和BWACR指标对数据集dataset2的聚类数进行评价的情况,由

应的聚类数为最佳聚类数。

4) 输出聚类结果、BWACR指标、最佳聚类数。

与参考文献[7]中的算法相比,本文算法采用阈值分层来确定聚类数搜索范围的上界,在获得最佳聚类数的同等情况下,大大缩小了最佳聚类数的搜索范围,从而使本文算法具有更高的执行效率。通过对本文和文献[7]中的算法过程进行分析,当K-means聚类算法的迭代次数较少时,本文算法的运行效率略高于文献[7]中的算法;但随着迭代次数的增加,本文算法的运行效率会得到明显的提高。

3 仿真实验与分析

本文实验采用Matlab 2010开发环境编程实现,在Windows 7操作系统的计算机上运行通过。

为检测本文提出的高效率最佳聚类数确定算法的有效性和运行效率。本章通过三组实验分别对人工数据集和真实数据集进行仿真测试,并进行算法对比仿真测试。

人工数据集包括三个,分别是dataset1、dataset2和dataset3。样本数据集1(dataset1)是人工随机生成的样本数据集,样本个数为150,真实聚类数目为5。样本数据集2(dataset2)由中心分别为(0,0),(20,20)的二维两高斯分布数据组成,各类分别有400个样本。样本数据集3(dataset3)由中心分别为(0,0),(5,7),(12,17),(19,24)的二维四高斯分布数据组成,各类分别有400个样本。

真实数据集:由UCI真实数据集组成,包括Iris、Wine数据集。

实验所用的人工数据集和UCI真实数据集的标准聚类数和数据及来源如表1所示。

表1 不同数据集的标准聚类数

数据集	标准聚类数	数据集来源	数据集	标准聚类数	数据集来源
dataset1	5	人工	Iris	3	文献[11]
dataset2	2	人工	Wine	3	文献[12]
dataset3	4	人工			

3.1 人工数据集仿真测试

以数据集dataset2为例,运用Wint指标、Silhouette指标和BWACR指标确定人工数据集最佳聚类数的情况如图2所示。

图2(a)可以得到最佳聚类数为3,对应的Wint指标值为0.8210。由图2(b)可以得到最佳聚类数为2,对应的

Silhouette 指标值为 0.8480。由图 2(c) 可以得到最佳聚类数为 2, 对应的 BWACR 指标值为 0.7895。

几种有效性指标估计出的人工数据集最佳聚类数情况如表 2 所示。

表 2 几种有效性指标估计出的人工数据集最佳聚类数

数据集	标准聚类数	Wint	Silhouette	BWACR
dataset1	5	4	5	5
dataset2	2	3	2	2
dataset3	4	3	4	4

从表 2 可以看出对真实类数分别为 5, 2, 4 的 dataset1、dataset2、dataset3 数据集, 运用 Wint 指标、Silhouette 指标和 BWACR 指标确定最佳聚类数的情况。其中 Silhouette 指标和 BWACR 指标均能确定出三个数据集的最佳聚类数, 而 Wint 指标则不够稳定, 得到的最佳聚类数是不准确的。

3.2 UCI 数据集仿真测试

仿真测试选用 UCI 数据库内的 Iris、Wine 数据集作为测试数据, 数据来源 (<http://archive.ics.uci.edu/ml/>)。

以数据集 Iris 为例, 运用 Wint 指标、Silhouette 指标和 BWACR 指标确定 Iris 数据集最佳聚类数的情况如图 3 所示。

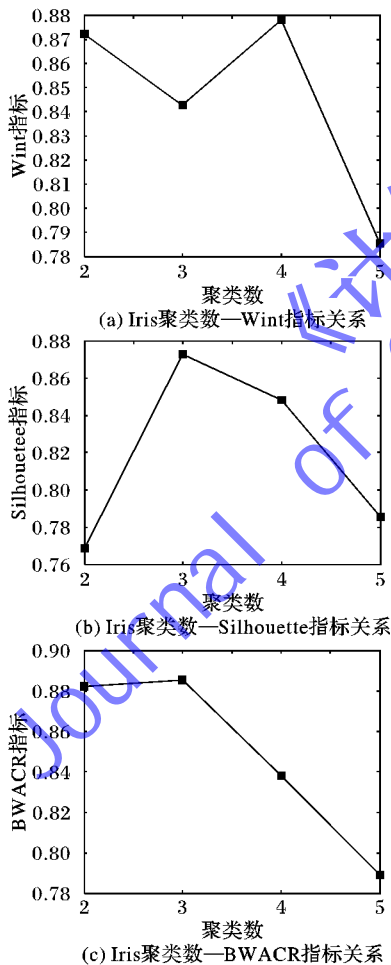


图 3 Iris 聚类数与指标关系

图 3 显示了分别运用 Wint 指标、Silhouette 指标和 CVR 指标对数据集 Iris 的聚类数进行评价的情况, 由图 3(a) 可以得到最佳聚类数为 4, 对应的 Wint 指标值为 0.8781。由图 3(b) 可以得到最佳聚类数为 3, 对应的 Silhouette 指标值为

0.8727。由图 3(c) 可以得到最佳聚类数为 3, 对应的 BWACR 指标值为 0.8856。

几种有效性指标估计出的 UCI 数据集最佳聚类数情况如表 3 所示。

表 3 几种有效性指标估计出的 UCI 数据集最佳聚类数

数据集	标准聚类数	Wint	Silhouette	BWACR
Iris	3	4	3	3
Wine	3	4	3	3

从表 3 可以看出对真实类数都为 3 的 Iris、Wine 数据集, 运用 Wint 指标、Silhouette 指标和 BWACR 指标确定最佳聚类数的情况。其中 Silhouette 指标和 BWACR 指标均能确定出三个数据集的最佳聚类数; 而 Wint 指标则不够稳定, 得到的最佳聚类数是不准确的。

3.3 算法对比仿真测试

此实验对人工随机生成的数据集 dataset1 ($n = 150$) 分别运用文献[7]提出的确定最佳聚类数的算法 (Method1) 和本文新提出的快速确定最佳聚类数的算法 (Method2) 进行聚类, 并对两种聚类结果的随机正确率和聚类算法的效率 (包括算法运行时间和迭代次数) 进行比较。随机聚类准确率 (RCRA) 是指在给定正确聚类数的情况下, 对数据集运行聚类算法 w 次 (取 $w = 10000$), 能够正确聚类的次数与 w 的比值的百分数。这里的运行时间是指 K-means 算法运行的时间, 迭代次数是指在给定正确聚类数的情况下 K-means 的迭代次数。为保证算法结果的有效性, 在实验中两种算法均重复运行 50 次, 运行时间和迭代次数取平均值。

图 4 显示了 Method1 和 Method2 对人工数据集 dataset1 的聚类效果, 由图可以看出两种算法均能得到较好的聚类效果。

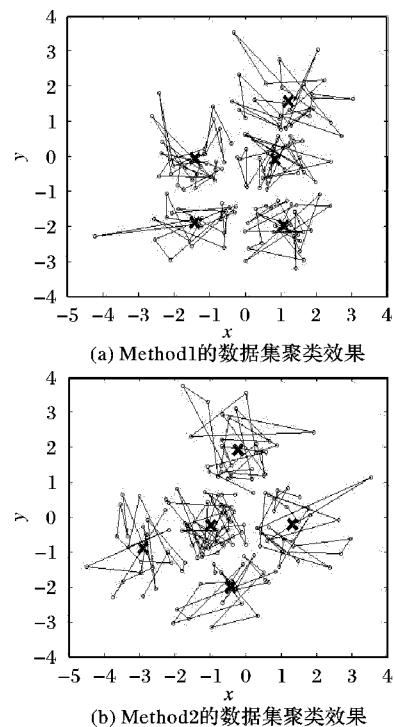


图 4 聚类效果

两种算法聚类结果的随机正确率和聚类算法的效率 (包括算法运行时间和迭代次数) 比较情况如表 4 所示。

表4 不同设定方法得到的聚类算法的正确率和算法效率

方法	RCAR/ %	运行 时间/s	迭代 次数	方法	RCAR/ %	运行 时间/s	迭代 次数
Method1	100	0.1528	3	Method1	100	0.1524	6
Method2	100	0.0621	3	Method2	100	0.0623	6

表4结果表明:在两种聚类算法准确率一致的情况下, Method2 的聚类算法运行时间低于 Method1。由此看出本文的算法运行效率高于 Method1,并能达到较好的聚类效果。由此证明本文算法具有较强的可行性和实际参考价值。

4 结语

本文针对现有 K -means 算法存在的不足,提出一种快速、高效的确定最佳聚类数的算法,在对样本数据进行有效分层后,快速确定聚类范围,并设计了新的聚类有效性指标,利用类内、类间夹角余弦值的比值作为衡量聚类效果好坏的评价标准,该算法克服了以往 K -means 聚类算法无法确定最佳聚类数目,导致聚类质量不高、聚类效果不好、算法时间复杂度较高等缺点,与同类算法相比较运行效率得到了提高。能够更加快速、高效地确定最佳聚类数,得到良好聚类效果。

参考文献:

- [1] SUN J, LIU J, ZHAO L. Clustering algorithms research [J]. Journal of Software, 2008, 19(1): 48–61. (孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48–61.)
 - [2] YU H, LI Z, YAO N. Research on optimization method for K -means clustering algorithm [J]. Journal of Chinese Computer Systems, 2012, 33(10): 2273–2277. (于海涛, 李梓, 姚念民. K -means 聚类算法优化方法的研究[J]. 小型微型计算机系统, 2012, 33(10): 2273–2277.)
 - [3] XING X, PAN J, JIAO L. A novel K -means clustering based on the immune programming algorithm [J]. Chinese Journal of Computers, 2003, 26(5): 605–610. (行小帅, 潘进, 焦李成. 基于免疫规划的 K -means 聚类算法[J]. 计算机学报, 2003, 26(5): 605–610.)
 - [4] XU X, XIAO Y. KBAC: K -means based adaptive clustering for massive dataset [J]. Journal of Chinese Computer Systems, 2012, 33(10): 2268–2272. (徐晓旻, 肖仰华. KBAC: 一种基于 K -means 的自适应聚类[J]. 小型微型计算机系统, 2012, 33(10): 2268–2272.)
 - [5] ZHANG L, CHEN Y, JI Y, et al. Research on K -means algorithm based on density [J]. Application Research of Computers, 2011, 28(11): 4071–4074. (张琳, 陈燕, 汲业, 等. 一种基于密度的 K -means 算法研究[J]. 计算机应用研究, 2011, 28(11): 4071–4074.)
 - [6] ZHANG Z, WANG A, CHAI X. Easy and efficient algorithm to determine number of clusters [J]. Computer Engineering and Applications, 2009, 45(15): 166–168. (张忠平, 王爱杰, 柴旭光. 简单有效的确定聚类数目算法[J]. 计算机工程与应用, 2009, 45(15): 166–168.)
 - [7] ZHOU S, XU Z, TANG X. New method for determining optimal number of clusters in K -means clustering algorithm [J]. Computer Engineering and Applications, 2010, 46(16): 27–31. (周士兵, 徐振源, 唐旭清. 新的 K -均值算法最佳聚类数确定方法[J]. 计算机工程与应用, 2010, 46(16): 27–31.)
 - [8] CALINSKI T, HARABASZ J. A dendrite method for cluster analysis [J]. Communications in Statistics, 1974, 3(1): 1–27.
 - [9] DIMITRIADOU E, DOLNICAR S, WEINGESSEL A. An examination of indexes for determining the number of cluster in binary data sets [J]. Psychometrika, 2002, 67(3): 137–160.
 - [10] DUDOIT S, FRIDLAND J. A prediction-based resampling method for estimating the number of clusters in a dataset [J]. Genome Biology, 2002, 3(7): 1–21.
 - [11] DEMBÉLÉ D, KASTNER P. Fuzzy C -means method for clustering microarray data [J]. Bioinformatics, 2003, 19(8): 973–980.
 - [12] BLAKE C L, MERZ C J. UCI repository of machine learning databases (University of California) [EB/OL]. [2013-06-21]. <http://mllearn.ics.uci.edu/MLRepository.html>.
- (上接第1295页)
- [14] AKAVIA A, GOLDWASSER S, HAZAY C. Distributed public key schemes secure against continual leakage [C]// Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing. New York: ACM, 2012: 155–164.
 - [15] BRAKERSKI Z, KATZ J, KATZ J, et al. Overcoming the hole in the bucket: public-key cryptography resilient to continual memory leakage [C]// Proceedings of the 2010 51st Annual IEEE Symposium on Foundations of Computer Science. Washington, DC: IEEE Computer Society, 2010: 501–510.
 - [16] HALEVI S, LIN H. After-the-fact leakage in public-key encryption [C]// TCC '11: Proceedings of the 8th Theory of Cryptography Conference. Berlin: Springer-Verlag, 2011: 107–124.
 - [17] KILTZ E, PIETRZAK K. Leakage resilient elgamal encryption [C]// ASIACRYPT '10: Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer-Verlag, 2010: 595–612.
 - [18] NGUYEN M H, TANAKA K, YASUNAGA K. Leakage-resilience of stateless/stateful public-key encryption from hash proofs [C]// ACISP '12: Proceedings of the 17th Australasian Conference on Information Security and Privacy. Berlin: Springer-Verlag, 2012: 208–222.
 - [19] ZHANG F, SUN Y, ZHANG L, et al. Research on certificateless public key cryptography [J]. Journal of Software, 2011, 22(6): 1316–1332. (张福泰, 孙银霞, 张磊, 等. 无证书公钥密码体制研究[J]. 软件学报, 2011, 22(6): 1316–1332.)
 - [20] DODIS Y, REYZIN L, SMITH A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data [C]// EUROCRYPT '04: Proceedings of the 2004 International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer-Verlag, 2004: 523–540.
 - [21] GENTRY C. Practical identity-based encryption without random oracles [C]// EUROCRYPT '06: Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer-Verlag, 2006: 445–464.