Computer Science and Application 计算机科学与应用, 2018, 8(8), 1164-1171
Published Online August 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.88128

Hans汉斯

A K-Means Algorithm Based on Feature Weighting

Yan Xu, Xueliang Fu, Honghui Li, Gaifang Dong, Qing Wang

College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot Inner Mongolia
Email: 1925195149@qq.com

Received: Jul. 25th, 2018; accepted: Aug. 5th, 2018; published: Aug. 10th, 2018

Abstract

Cluster analysis is a statistical analysis technique that divides the research objects into relatively homogeneous groups. The core of cluster analysis is to find useful clusters of objects. K-means clustering algorithm has been receiving much attention from scholars because of its excellent speed and good scalability. However, the traditional K-means algorithm does not consider the influence of each attribute on the final clustering result, which makes the accuracy of clustering have a certain impact. In response to the above problems, this thesis proposes an improved feature weighting algorithm. The improved algorithm uses the information entropy and ReliefF feature selection algorithm to weight the features and correct the distance function between clustering objects, so that the algorithm can achieve more accurate and efficient clustering effect. The simulation results show that compared with the traditional K-means algorithm, the improved algorithm clustering results are stable, and the accuracy of clustering is significantly improved.

Keywords

K-Means Clustering, Information Entropy, ReliefF Algorithm, Feature Weighting

一种基于特征加权的K-Means算法研究

徐　艳，付学良，李宏慧，董改芳，王　晴

内蒙古农业大学计算机与信息工程学院，内蒙古　呼和浩特
Email: 1925195149@qq.com

收稿日期：2018年7月25日；录用日期：2018年8月5日；发布日期：2018年8月10日

摘　要

聚类分析是将研究对象分为相对同质的群组的统计分析技术，聚类分析的核心就是发现有用的对象簇。

文章引用: 徐艳, 付学良, 李宏慧, 董改芳, 王晴. 一种基于特征加权的 K-Means 算法研究[J]. 计算机科学与应用, 2018, 8(8): 1164-1171. DOI: 10.12677/csa.2018.88128

K-means聚类算法由于具有出色的速度和良好的可扩展性，一直备受广大学者的关注。然而，传统的K-means算法，未考虑各个属性对于最终聚类结果的影响差异性，这使得聚类的精度有一定的影响。针对上述问题，本文提出一种改进的特征加权算法。改进算法通过采用信息熵和ReliefF特征选择算法对特征进行加权选择，修正聚类对象间的距离函数，使算法达到更准确更高效的聚类效果。仿真实验结果表明，与传统的K-means算法相比，改进后的算法聚类结果稳定，聚类的精度有明显提升。

关键词

K-means聚类，信息熵，ReliefF算法，特征加权

1. 引言

数据挖掘是目前人工智能和数据库领域研究的热点问题，指从大量的数据中提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。聚类分析现在已经成为数据挖掘领域中一个非常重要的研究方向。MacQueen 提出[1]的 K-means 算法是聚类分析中最常用的方法之一。它采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把得到紧凑且独立的簇作为最终目标[2]。K-means 算法假设样本的每个特征对最终聚类的贡献程度一样，但在实际情况中某些特征在聚类的过程中起到很大的作用，而某些特征的作用却很小，甚至对聚类过程没有影响。

针对传统 K-means 算法的这一问题，学者们进行了大量研究，研究表明：通过对特征赋予不同的特征权值，能够有效解决上述问题并提高聚类性能。目前，计算特征权重的算法有很多种：刘铭[3]等人提出一种结合限制数据的特征权值量化函数，该函数通过用户指定的限制数据进行特征权值量化并对不同的限制数据赋予不同的置信度，解决了限制数据分布不均匀和限制数据中可能包含不一致性的问题；Li Jie [4]等人提出将针对分类问题的 ReliefF 算法应用于聚类问题，通过 ReliefF 算法计算特征权重值，并对各维特征进行加权，提高聚类的性能；Meng Qian [5]等人提出通过梯度下降技术最小化特征评估函数 FLearning (w)为每个特征分配权重并进行加权，该算法采用遗传算法和模拟退火算法的优点，减弱冗余特征的影响，解决了容易陷入局部最优解的问题。Songtao Shang [6]等人提出一种改进的基尼指数算法计算特征权重，该算法克服了原始 Gini 的缺点，将条件概率与后验概率结合，抑制训练集不平衡时的影响。杨玉梅[7]利用信息论中的信息熵计算特征权重并对各位特征加权，有效的解决了特征对聚类的影响。

综上所述，为了提高传统 K-means 算法的聚类精度，国内外学者对 K-means 算法进行了大量改进探索研究，并取得了一些阶段性的成果。本文拟研究传统 K-means 算法在聚类过程中聚类对象的每个特征对聚类结果的贡献度，使贡献程度大的特征优先利用，理论上讲可以有效提升 K-means 算法聚类的准确率和精度。因此，本文提出将熵值法和 ReliefF 特征选择算法有机融合，通过采用信息熵和 ReliefF 特征选择算法对特征进行加权选择，修正聚类对象间的距离函数，使算法达到更准确更高效的聚类效果。实验结果表明，改进后的算法聚类结果稳定，且具有较高的准确率，达到预期目的。

2. K-means 算法

K-means 算法的核心思想是通过迭代把数据对象划分到不同的簇中，以求目标函数最小化，从而使

生成的簇尽可能的紧凑和独立，算法具体流程如下。

输入：簇的数目 k，包含 n 个对象的数据集 D。

输出：k 个簇。

步骤如下：

1) 从 D 中任意选择 k 个对象作为初始聚类中心；

2) 计算每个对象与这些中心对象的距离；并根据最小距离重新对相应对象进行划分；

3) 重新计算每个聚类的均值；

4) 当满足一定条件，如没有对象再被重新分配给其他的类簇、聚类中心不再发生变化、误差平方和(SSE)最小，则算法终止；如果条件不满足则回到步骤(2)。

其中，每个对象与中心对象的距离为欧氏距离，距离公式如下：

$$d(x,y)=\sqrt{\sum_{j=1}^{m}\left(x_j-y_j\right)^2} \tag{1}$$

上式中：x，y 分别表示样本和聚类中心；j 表示第 j 维特征。

3. 基于特征加权的改进算法

3.1. 信息熵

1948 年，信息论之父克劳德•艾尔伍德•香农在他发表的论文“通信的数学理论(A Mathematical Theory of Communication)”中，提出了“信息熵”的概念。香农指出，任何信息都存在冗余，冗余大小与信息中每个符号(数字、字母或单词)的出现概率或者说不确定性有关。他用“信息熵”描述信息的不确定性。信息熵的数学表达式如下：

$$H(x)=-\sum_{i=1}^{m}p_i\log_2 p_i \tag{2}$$

其中，p_i 表示事件发生的概率。

在权重方面，对于任意两个特征 A_i、A_j，若 $H(A_i)>H(A_j)$，则表示特征 A_i 比特征 A_j 好，在聚类过程中，特征 A_i 比特征 A_j 起到的作用大。

3.2. RliefF 算法

1994 年，Knonenko 提出了 ReliefF 算法，它是 Relief 算法的扩展，主要处理多分类问题[8]。ReliefF 算法的基本思想是从训练样本集中随机取出一个样本 x_i；然后从与 x_i 同类的样本中取出 k 个近邻样本 H_i；再从其他与 x_i 不同的类中分别取出 k 个样本 M_i；根据权重公式更新每个特征的权重；随机抽取 m 次，得到最终的特征权重。

权重表达式如下：

$$w(j)=w(j)+\sum_{c\neq class(i)}\frac{\dfrac{p(c)}{1-p(class(x_i))}\sum_{j=1}^{k}d(x_i(j),M_i(j))}{mk}-\sum_{j=1}^{k}\frac{d(x_i(j),H_i(j))}{mk} \tag{3}$$

上式中：$class(i)$ 表示样本 x_i 所属的类别；c 表示除样本 x_i 所属的类别外的类别；$p(c)$表示类别 c 的先验概率。$x_i(j)$ 表示样本 x_i 关于第 j 个特征的值；m 是随机抽取样本的次数；$d\left(x_i(j),M_i(j)\right)$ 表示距离函数，用于计算两个样本关于第 j 个特征的距离，计算公式如下：

$$d\left(x_i(j), M_i(j)\right) = \left| \frac{x_i(j) - M_i(j)}{\max(j) - \min(j)} \right| \tag{4}$$

其中， $\max(j), \min(j)$ 表示第 j 个特征的所有取值中的最大值和最小值。

ReliefF 算法用于处理多分类问题，每个样本都要有明确的类标记。但聚类分析中的样本没有类标记。所以，需要先对样本集进行一次初聚类，获得样本的类标记，再用 ReliefF 算法进行特征权重的计算。

3.3. 基于特征加权的改进算法 ER_Kmeans 算法(EntropyReliefF_Kmeans)

传统的 K-means 算法在聚类过程中假设每个特征对于聚类的影响是相同的，忽略了特征对于聚类过程的影响，导致最终的聚类结果准确率不高。基于特征加权的改进算法则有效的解决了这一问题。

ER_Kmeans 算法的基本思想是将聚类对象的特征权重与信息熵作为 K-means 算法的特征权重进行聚类。设信息熵权重为 w_1，特征权重为 w_2，则最终的特征权重为

$$w = \frac{w_1 + w_2}{2} \tag{5}$$

ER_Kmeans 算法的步骤如下(图 1)：

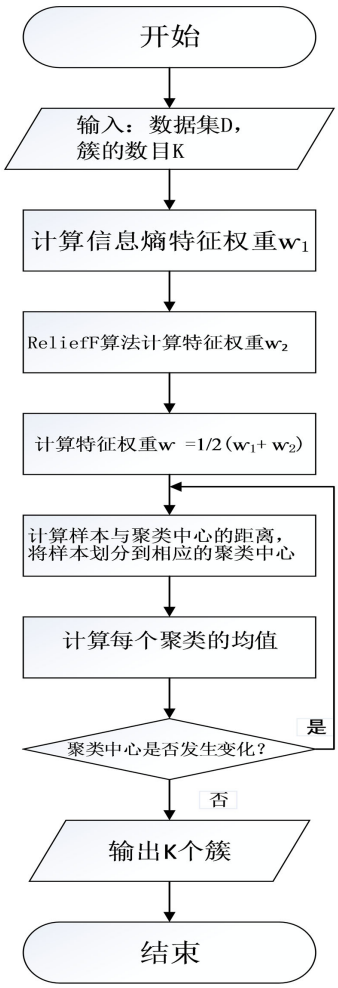

Figure 1. ER_Kmeans algorithm flowchart
图 1. ER_Kmeans 算法流程图

输入：数据集 D，簇的数目 K

输出：K 个簇

1) 随机选择 K 个初始聚类中心；

2) 计算信息熵特征权重 w1；

3) ReliefF 算法计算特征权重 w2；

4) 计算特征权重 w；

5) 计算每个样本与这些聚类中心的距离 $d(x,y)=\sqrt{\sum_{j=1}^{m}\omega_j(x_j-y_j)^2}$；并根据最小距离将样本划分到相应的聚类中心；

6) 重新计算每个聚类的均值；

7) 若聚类中心不再发生变化，则算法终止；如果聚类中心发生变化则回到步骤(5)。

4. 仿真实验

4.1. 实验环境与数据集

实验的硬件环境为 Intel(R)Core(TM)i5-6500 3.20 GHz，8 G 内存，软件环境为 Matlab2016b，Windows7 操作系统。实验选取的数据集为 UCI 数据库[9]中的 Iris、Balance-scale、Stalog 数据集，数据集的主要信息如表 1 所示。

4.2. 实验结果与分析

为了验证在聚类过程中聚类对象的每个特征对聚类结果的贡献度不同，在 Iris、Balance-scale、Stalog 3 个数据集各计算 20 次得到各特征对应的权重值如图 2、图 3、图 4 所示。图中一条线代表一次计算。

由图 2、图 3、图 4 可以看出，每个特征对于聚类结果具有不同的影响。以图 4 Stalog 数据集的特征权重值为例，从图中可以看出，特征 7 和特征 12 的权重值较高，说明其对聚类结果的影响较大；特征 6 和特征 15 的权重值较低，几乎趋近于 0，说明对聚类结果的影响较小甚至可能没有影响。传统的 K-means 算法忽略了这一问题，导致最终的聚类结果在精度方面较低。

为了验证算法的有效性及稳定性，在相同的实验环境下，ER_Kmeans 算法与传统 K-means 算法、文献[4]的基于 ReliefF 算法的加权 K-means 算法 ReliefF-Kmeans、文献[7]的基于信息熵的加权 K-means 算法 entropy-Kmeans 在相同的数据集下，都进行了 20 次的单独实验，并取平均值，在准确率、误差平方和(SSE)、迭代次数和运行时间 4 个方面进行比较。结果如表 2~表 6 所示。

从表 2 可以看出，在准确率方面，ER_Kmeans 算法明显高于其他 3 种算法，由于传统 K-means 算法忽略了特征对于聚类结果的影响，聚类结果是不稳定的，所以准确率低于其他 3 种算法；从表 3 可以看出，ER_Kmeans 算法的误差平方和低于其他 3 种算法，误差平方和越小，说明聚类中的对象越相似，所以 ER_Kmeans 算法每类的类内对象相似度高，聚类质量优于其他 3 种算法，达到了聚类分析的最终目的，

Table 1. Experimental data set
表 1. 实验数据集

数据集	数据个数	属性个数	类别个数
Iris	150	4	3
Balance	625	4	3
Stalog	846	18	4

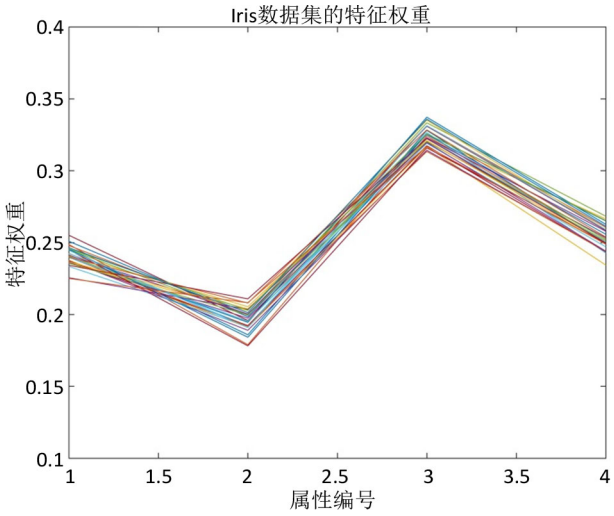

Figure 2. Feature weights of the Iris dataset
图 2. Iris 数据集的特征权重

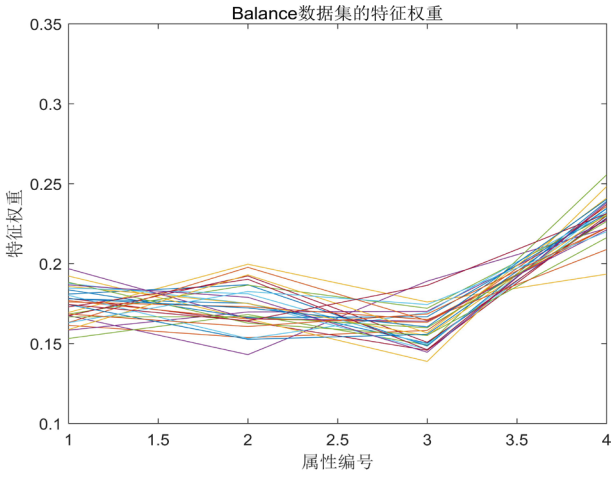

Figure 3. Feature weights for Balance-scale datasets
图 3. Balance 数据集的特征权重

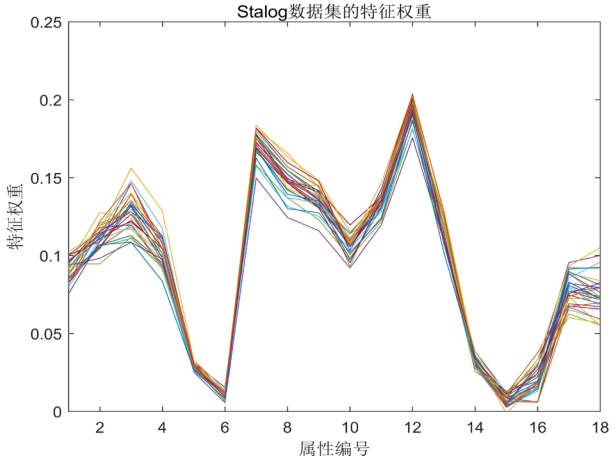

Figure 4. Feature weights of the Stalog data set
图 4. Stalog 数据集的特征权重

Table 2. Comparison of accuracy of each algorithm in UCI dataset (%)
表 2. 各算法在 UCI 数据集的准确率比较(%)

数据集	传统 K-means 算法	entropy-Kmeans [7]	ReliefF-Kmeans [4]	本文算法(ER_Kmeans 算法)
Iris	83.40	85.10	87.07	89.50
Balance	63.03	64.95	64.65	66.09
Stalog	43.97	44.56	44.58	45.31

Table 3. Comparison of error square sum (SSE) of each algorithm in UCI dataset (/ms^2)
表 3. 各算法在 UCI 数据集的误差平方和(SSE)比较(/ms^2)

数据集	传统 K-means 算法	entropy-Kmeans [7]	ReliefF-Kmeans [4]	本文算法(ER_Kmeans 算法)
Iris	96	89	93	79
Balance	3492	3489	3510	3478
Stalog	3,740,979	3,626,124	3,562,829	3,556,605

Table 4. Comparison of the number of iterations of each algorithm in the UCI data set (/times)
表 4. 各算法在 UCI 数据集的迭代次数比较(次)

数据集	传统 K-means 算法	entropy-Kmeans [7]	ReliefF-Kmeans [4]	本文算法(ER_Kmeans 算法)
Iris	8	7	9	7
Balance	14	16	5	9
Stalog	17	17	13	13

Table 5. Runtime comparison of each algorithm in UCI dataset (/ms)
表 5. 各算法在 UCI 数据集的运行时间比较(/ms)

数据集	传统 K-means 算法	entropy-Kmeans [7]	ReliefF-Kmeans [4]	本文算法(ER_Kmeans 算法)
Iris	16.8	4.2	4.3	3.5
Balance	136.7	36.7	11.4	34.6
Stalog	172.2	63.7	48.4	40.6

Table 6. The average run time of each algorithm in the UCI data set is compared with each iteration (/ms)
表 6. 各算法在 UCI 数据集的平均每次迭代运行时间比较(/ms)

数据集	传统 K-means 算法	entropy-Kmeans [7]	ReliefF-Kmeans [4]	本文算法(ER_Kmeans 算法)
Iris	2.1	0.6	0.5	0.5
Balance	9.8	2.3	2.3	3.8
Stalog	10.1	3.7	3.7	3.1

即类内相似度高，类间相似度低；从表 4 和表 5 可以看出，ER_Kmeans 算法在迭代次数和运行时间上低于传统 K-means 算法和 entropy-Kmeans 算法，在 Iris 数据集和 Stalog 数据集上低于 ReliefF-Kmeans 算法，但在 Balance 数据集上高于 ReliefF-Kmeans 算法，原因是算法的初始聚类中心是随机选择的，从而导致了算法迭代次数的不稳定、运行时间的长短不同。但在平均运行时间上二者差别不大，从表 6 可以看出，ER_Kmeans 算法在平均每次迭代运行时间上低于传统 K-means 算法和 entropy-Kmeans 算法，在 Balance

数据集上高于 ReliefF-Kmeans 算法，但 ER_Kmeans 算法平均每次迭代的时间与 ReliefF-Kmeans 算法相差不大，说明迭代次数和运行时间较高是受到初始聚类中心的影响。

5. 结束语

本文提出一种利用信息熵和 ReliefF 算法对特征加权的 K-means 算法 ER_Kmeans 算法，有效的解决了不同的特征对于聚类具有不同的影响这一问题。实验结果表明：改进后的 K-means 算法在准确率和聚类误差上都优于传统 K-means 算法和其他两种特征加权方法，取得了较好的聚类结果。

基金项目

本研究获得国家自然科学基金(61363016, 61063004)；内蒙古自然科学基金(NO.2015MS0605, NO.2015MS0626, NO.2015MS0627, NO.2017MS0605)；内蒙古教育厅高校研究项目(NJZC059)；内蒙古自治区高等学校科学研究重点项目(NJZZ14100)；教育部留学人员基金([2014]1685)；内蒙古自治区科技计划项目：穿透降水量 GSM 网络在线监测与数据传输系统的资助。

参考文献

[1] Alexandropoulos, A., Plessas, F. and Birbas, M. (2010) A Dynamic DFI-Compatible Strobe Qualification System for Double Data Rate (DDR) Physical Interfaces. *IEEE International Conference on Electronics*, Athens, 12-15 December 2010, 277-280. https://doi.org/10.1109/ICECS.2010.5724507

[2] 卓金武, 周英. 量化投资: MATLAB 数据挖掘技术与实践[M]. 北京: 电子工业出版社, 2017: 217-224.

[3] 刘铭, 吴冲. 基于特征权重量化的相似度计算方法[J]. 计算机学报, 2015, 38(7): 1420-1433.

[4] Li, J. and Gao, X.B. (2006) A New Feature Weighted Fuzzy Clustering Algorithm. *Proceedings of SPIE*, *The International Society for Optical Engineering*, **3641**, 412-420.

[5] Meng, Q. (2015) An Improved Clustering Algorithm Based on Feature-Weight Learning. *Journal of Information and Computational Science*, **12**, 3519-3526. https://doi.org/10.12733/jics20106074

[6] Shang, S.T. and Shi, M.Y. (2016) Improved Feature Weight Algorithm and Its Application to Text Classification. *Mathematical Problems in Engineering*, **2016**, Article ID: 7819626. https://doi.org/10.1155/2016/7819626

[7] 杨玉梅. 基于信息熵改进的 K-means 动态聚类算法[J]. 重庆邮电大学学报(自然科学版), 2016, 28(2): 254-259.

[8] 菅小艳, 韩素青. 不平衡数据集上的 Relief 特征选择算法[J]. 数据采集与处理, 2016, 31(4): 838-844.

[9] UCI Machine Learning Repository: Data Sets [DB]. http://archive.ics.uci.edu/ml/datasets.html