

基于属性集成的三支聚类算法

薛文龙, 王平心

江苏科技大学理学院, 江苏 镇江

收稿日期: 2022年9月21日; 录用日期: 2022年10月14日; 发布日期: 2022年10月25日

摘要

在三支聚类中, 每一个类簇由核心域、边界域、琐碎域三个集合来表示, 体现出聚类中的不确定性, 降低了决策风险。针对属性数较多的数据集, 本文提出了一种基于属性集成的三支聚类算法。该算法通过对样本随机选择部分属性使用kmeans算法进行聚类, 并根据每次聚类的结果采用集成策略将元素划分至核心域、边界域或琐碎域中。在UCI等数据集上测试的结果表明, 该算法相较于传统的kmeans算法, 可以较好地提升聚类结果的ACC、DBI和AS指标。

关键词

三支聚类, 属性集成, Kmeans

Three-Way Clustering Based on Attribute Integration

Wenlong Xue, Pingxin Wang

School of Science, Jiangsu University of Science and Technology, Zhenjiang Jiangsu

Received: Sep. 21st, 2022; accepted: Oct. 14th, 2022; published: Oct. 25th, 2022

Abstract

In the three-way clustering, each cluster is represented by three sets of core domains, boundary domains, and trivial domains, reflecting the uncertainty in the cluster, and reducing the risk of decision-making. In this paper, a three-branch clustering method based on attribute integration is proposed for datasets with a large number of attributes. The algorithm uses k-means algorithm to cluster by randomly selecting some attributes, and classifies elements into core, boundary, or trivial domains based on the results of each clustering. Tested on the UCI and other dataset, the results obtained are better than the conventional clustering method in ACC, DBI and AS indicators, which can improve the clustering effect.

Keywords

There-Way Clusters, Attribute Integration, Kmeans

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

聚类, 是将一组数据对象集划分成几个类簇, 处于同一个簇中的对象相似度较高, 而处于不同簇中的样本相似度低。聚类是一种无监督学习方法, 现实世界里的真实数据集没有原始分类标签, 想要挖掘数据之间信息需要使用聚类分析, 所以聚类对无标签数据学习极其重要。聚类分析经过不断地发展也已经应用到机器学习、模式识别和数据挖掘等领域。传统的聚类算法可以分为基于划分聚类、基于层次聚类、基于模糊理论聚类、基于扰动聚类、基于密度聚类、基于图论聚类、基于网格聚类、基于分型聚类和基于模型聚类等[1]。Kmeans 是一种基于划分的聚类算法, 是聚类算法中最常用的算法, 被广泛应用于计算机视觉和天文学等领域。

传统的聚类方法是二支聚类, 即对于每一个元素都有唯一确定的一个类簇与之匹配。但在实际划分中通常会遇到信息不充分的情况, 这时如果将数据对象强行划分至某一个类簇, 会增加误分类的概率, 导致聚类精度降低, 提高决策的风险。针对这一问题, 许多软聚类算法被提出。如, Hoppner *et al.* [2]通过引入模糊集的概念, 提出了模糊聚类。Lingras [3]将粗糙集的概念引入聚类中, 提出了粗糙聚类方法。Yao *et al.* [4]提出用区间集来表示类簇。Yu *et al.* [5] [6]基于三支决策思想, 提出了三支聚类方法。

除了决策风险问题, 许多聚类算法还会遇到另一个问题: 对于属性数较大的数据集, 聚类的效果并不好。在面对属性较多的数据集时, 如果同时考虑所有的属性, 会使得聚类效果并不理想; 但如果只选取部分属性, 则聚类的结果会比较片面。故许多算法在面对属性数较大的数据集时, 会采用数据降维的方法。本文提出的基于属性集成的三支聚类算法, 无需对数据集进行降维处理, 结合了三支聚类的思想, 对不确定的对象延迟决策, 从而降低决策风险。其主要思想是: 通过对数据集进行多次聚类, 每次选取其中的部分属性进行聚类, 然后对多次聚类的结果进行集成, 根据元素在聚类中出现的次数来决定将元素划分至核心域、边界域或琐碎域中。仿真实验结果表明, 该方法具有有效性和可行性。

2. 相关工作

2.1. 三支聚类

设有数据集 $U = \{x_1, x_2, \dots, x_n\}$, 在传统的二支聚类中, 每一个类簇用集合来表示, 即寻找一组集合 C_1, C_2, \dots, C_k 满足

$$\bigcup_{i=1}^k C_i = U \quad (1)$$

$$C_i \cap C_j = \emptyset, i, j = 1, 2, \dots, k; i \neq j \quad (2)$$

在传统二支聚类中, 一个对象和一个类簇之间只有两个关系: 对象属于类簇或者对象不属于类簇。但在实际情况中, 可能因为信息的缺失或者对象本身的模糊性而导致无法确定对象是否属于类簇。如果在此时将对象强行划分至某类簇中, 会降低聚类的精确度。三支聚类是解决这一问题的方法之一。

三支聚类是由 Yu *et al.* [7]在引入了三支决策的思想后提出的。2010年 Regina 大学的姚一豫教授在研究粗糙集三个域和统计学中的假设验证基础上提出了三支决策理论[8] [9] [10]。他将研究对象分为正域、负域和边界域。正域中的对象是被接受的, 负域中的元素是被拒绝的, 而边界域中的元素是不做决定或者推迟决定的。

在三支聚类中, 每一个类簇用三个互不相交的集合 C_i^P, C_i^B, C_i^N 来表示, 分别称为核心域、边界域和琐碎域, 它们满足:

$$C_i^P \cup C_i^B \cup C_i^N = U \quad (3)$$

核心域中的元素一定属于这个类簇, 琐碎域中的元素一定不属于这个类簇, 而边界域中的元素可能属于也可能不属于这个类簇。有时可以不用表示出琐碎域, 使用核心域和边界域来表示一个类簇。三支聚类的结果可以表示为 $\{(C_1^P, C_1^B), (C_2^P, C_2^B), \dots, (C_k^P, C_k^B)\}$, 其满足

$$C_i^P \neq \emptyset, i = 1, 2, \dots, k \quad (4)$$

$$\bigcup_{i=1}^k (C_i^P \cup C_i^B) = U \quad (5)$$

其中式(4)保证论域非空, 式(5)保证每一个点都属于一个类中。

若在三支聚类中满足 $C_i^B = \emptyset, i = 1, 2, \dots, k$, 则此时三支聚类退化成了二支聚类。从而可以看出三支聚类是二支聚类的一个推广。

由式(2)可知, 传统的二支聚类是一种硬聚类, 即每一个元素都属于唯一的类簇中。与硬聚类相对的是软聚类, 即一个样本可以属于多个类簇, 或者类簇之间的交集不为空。三支聚类既可以是一种硬聚类也可以是一种软聚类。具体的, 如果存在 $i, j = 1, 2, \dots, k$ 且 $i \neq j$ 满足以下四种条件:

$$C_i^P \cap C_j^P \neq \emptyset \quad (6)$$

$$C_i^P \cap C_j^B \neq \emptyset \quad (7)$$

$$C_i^B \cap C_j^P \neq \emptyset \quad (8)$$

$$C_i^B \cap C_j^B \neq \emptyset \quad (9)$$

之一的的话, 则该聚类是一种软聚类, 否则是一种硬聚类。本文的聚类是一种软聚类。

2.2. Kmeans 算法

Kmeans 算法是一种基于划分的聚类算法。其主要思想是在给定类簇个数 k 以及 k 个初始类簇中心点的情况下, 将每个对象分配到与之最近的类簇中心所属的类簇中。每个对象都被分配之后, 重新计算每个类簇的中心点。迭代上述过程, 直到满足一定的条件, 如类簇中心点的变动很小、达到给定的迭代次数。优点是简单易懂且方便实现。Kmeans 算法中的主要定义如下:

定义 1 (距离): 在数据集 U 中, 有 n 个对象 $\{x_1, x_2, \dots, x_n\}$, 每个对象有 p 个属性 $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ 。对象与聚簇中心的距离 $D(x_i, x_j)$ 采用欧式距离

$$D(x_i, c_j) = \|x_i - c_j\|_2 = \sqrt{\sum_{k=1}^p (x_{ik} - c_{jk})^2} \quad (10)$$

其中 x_i 为第 i 个对象, c_j 为第 j 个聚簇的中心, x_{ik} 为 x_i 的第 k 个属性, c_{jk} 为 c_j 的第 k 个属性。

定义 2 (类簇中心): 类簇的中心是该类簇中所有对象各个维度的平均值, 即

$$c_{ij} = \frac{1}{k_i} \sum_{t=1}^{k_i} x_{ij}^{(i)} \quad (11)$$

其中, c_{ij} 为第 i 个聚簇的中心 c_i 的第 j 个属性, k_i 为第 i 个聚簇中的元素个数, $x_{ij}^{(i)}$ 为第 i 个聚簇中第 t 个元素的第 j 个属性。

Kmeans 算法步骤如下:

算法 1 kmeans 算法

输入: 数据集 U ($n \times p$ 维矩阵), 划分的聚类数 k

输出: n 维列向量 t , 其中 t_i 为点 i 的所属的类簇。

初始化参数:

$i_{\max} = 50$ 最大迭代次数

1: 从 U 中随机选择 k 个对象作为初始聚类中心 $c_1^0, c_2^0, \dots, c_k^0$

2: for $i = 1$ to i_{\max} do

 for $j = 1$ to n do

$$t_j = \arg \min_k \{D(x_j, c_k)\}$$

 end for

 for $s = 1$ to k do

 用式(11)计算新的类簇中心 $c_1^i, c_2^i, \dots, c_k^i$

 end for

 如果最小误差变化小于给定值, 则结束循环

end for

3: 输出 t 。

3. 基于属性集成的三支聚类

本算法适合处理属性数比较多的数据集, 其主要思想是对数据集的不同的属性进行多次聚类, 再根据总的聚类结果进行分类。首先选择数据集中的一部分属性, 针对这些属性使用 kmeans 算法进行聚类。重复上述过程多次, 并记录下每一次聚类的结果。最后根据聚类的结果进行分类, 依据每个点在聚类中出现的次数, 来确定将其划分到核心域、边界域还是琐碎域之中。

属性选择的方法并不是固定的, 可以根据实际的需要采用不同的方法, 如将属性分成不同的部分, 依次对这些部分进行选择; 或每次随机选择一定比例的属性; 或根据属性的含义进行选择等。本实验采用随机选择属性的方法, 每次随机选择其中的 80% 的属性, 重复 10 次。

由于 kmeans 算法是无监督的聚类方法, 有可能出现虽然聚类结果一样但每个对象的标签却截然不同。故要对分类的结果进行标签匹配, 来保证尽可能多的元素在不同的聚类结果中属于相同的类簇。标签匹配有两种方式, 第一种方式是为每一个标签尽量选择匹配程度最高的标签, 优点是简单, 复杂度低, 但容易陷入局部最优解; 另一种方式就是从全局出发, 寻找一种标签的分配方法, 确保尽可能多的对象在不同的聚类结果中属于相同的类簇, 这种方式的获得的效果更加理想, 但复杂度更高。但并非采用第二种匹配方式的效果就会比第一种方法好。在实验中发现, 在聚类过程中使用第一种方式进行标签匹配时聚类效果最佳。在进行数据评估的时候, 使用第二种方式可以保证结果的准确性。

聚类的结果使用一个 $u \times k$ 维矩阵 R 来表示, 其中 $R_{ij} = \{0, 1, 2\}$ 。若 $R_{ij} = 0$, 则说明对象 i 在类簇 j 的琐碎域中; 若 $R_{ij} = 1$, 则说明对象在类簇 j 的边界域中; 若 $R_{ij} = 2$, 则说明对象 i 在类簇 j 的核心域中。

该算法的算法流程如下:

算法 2 基于属性集成的三支聚类

输入: 数据集 U ($n \times p$ 维矩阵), 划分的聚类数 k

输出: 数据点 x_i 的所属类簇的表示矩阵 R ($u \times k$ 维矩阵, 元素及其含义如上)

初始化参数:

$C = (0)_{n \times k}$ (其中元素 C_{ij} 表示对象 i 被划分到类簇 j 中的次数)

M ($m \times t$ 矩阵, 其行向量是随机生成的 t 个不同的介于 $1, p$ 的整数, 表示选择的属性的列, $t = [0.6p]$, 其中 $[x]$ 为高斯函数, 表示不大于 x 的整数的值)

$l = 0$

$R = (0)_{u \times k}$

1: for $c = 1$ to m do

对数据集 U 根据 M 的第 c 行选择的属性的列使用 kmeans 算法进行聚类, 对于每一个对象 i , 如果它被分类到了聚类 j 中, 则 $C_{ij} = C_{ij} + 1$ 。

end for

2: for $j = 1$ to k do

$l = \max \{C_{ij} \mid i = 1, 2, \dots, n\}$

for $i = 1$ to n do

if $C_{ij} > 0.6l$

$R_{ij} = 2$

else if $C_{ij} > 0$

$R_{ij} = 1$

end if

end for

3: 输出矩阵 R 。

4. 实验仿真

4.1. 数据集介绍

本文采用 4 组 UCI 数据集: LSVT Voice Rehabilitation (LSVT) [11]、Statlog (Image Segmentation) (Statlog)、Urban Land Cover (Urban)、dermatology_numeric (dermatology) [12], 2 组 Shape 数据集[13]: D31、R15 和 2 组 sipu 数据集[14]: mk2、z2 对算法进行测试, 具体信息如表 1 所示。

4.2. 实验性能评价指标

聚类标准评价大致分为两类: 外部聚类和内部聚类。本文采用评价指标为准确率(Accuracy, ACC)、DBI (Davies-Bouldin Index)和平均轮廓系数(AS, Average Silhouette)。

4.2.1. 准确率

ACC 是一种常见的评价聚类结果好坏的外部指标, 将预测的结果与真实值做对比, 此值越高说明聚类结果越好。

定义 3 (ACC) [15]:

$$ACC = \frac{1}{N} \sum_{i=1}^k C_i \quad (12)$$

其中, N 为样本容量, C_i 为正确划分到类簇 i 的样本个数, k 为聚类数。

4.2.2. DBI

DBI 是一种聚类性能度量的内部指标。其基本原理是计算计算所有簇的类内距离和类间距离的比值，越接近 0，则说明聚类效果越好。

定义 4 (DBI) [16]:

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \frac{c_i + c_j}{\|w_i - w_j\|_2} \quad (13)$$

其中, c_i 表示第 i 类中所有样本到聚类中心 w_i 的平均距离, $\|w_i - w_j\|_2$ 表示类簇 i 与类簇 j 聚类中心之间的欧式距离, k 表示聚类数。

4.2.3. 轮廓系数

轮廓系数 S_i 是由 Rousseeuw 提出的一种评价聚类好坏效果的指标。它结合了凝聚度和分离度两个因素, 可以评估一个点被分类在类簇中的合理程度。

定义 5 样本 i 的轮廓系数 S_i [16]

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (14)$$

其中 a_i 为样本 i 到同簇其它样本的平均值, 称为簇内不相似度, b_i 为样本 i 到其他某类簇中的点的平均距离的最小值, 称为簇间不相似度。 S_i 的取值范围为 $[-1, 1]$, 接近 1 说明对样本 i 的分类合理, 接近 -1 说明样本 i 的应该被划分到其他类簇之中, 接近 0 说明样本 i 处在多个类簇的交界处。

平均轮廓系数是所有样本的轮廓系数的平均值, 它可以用来评估聚类划分的合理性。

定义 6 AS

$$AS = \frac{1}{n} \sum_{i=1}^n S_i \quad (15)$$

其中, S_i 为样本 i 的轮廓系数。平均轮廓系数的取值范围为 $[-1, 1]$, 趋于 1 说明聚类效果比较好, 小于 0 说明簇内的平均距离小于最近的其他簇, 表示聚类效果不好。

5. 实验结果

本实验对每组数据分别使用 kmeans 与三支聚类进行 100 次聚类, 计算每次聚类结果的 ACC, DBI 与 AS, 并取平均数, 实验结果如表 2 所示。

从实验数据中可以看出, 无论是对于属性较多的数据集, 还是对于属性较少的数据集, 本文提出的方法与 kmeans 相比, 所得的聚类更加准确、效果也更好。从 R15、LSVT 数据集上的结果可以看出, 三支聚类在 DBI 和 AS 的指标上优于二支聚类, 同时准确率类也略优于二支聚类。从其他的数据集上也可以看出类似的效果, 由此可以证实, 基于属性集成的三支聚类算法是有效的。

Table 1. Datasets used in experiments

表 1. 实验中使用的数据集

数据集	样本维度	样本数量	类别数
LSVT	256	126	2
Statlog	18	2310	7
Urban	147	675	9

Continued

dermatology	34	366	6
D31	2	3100	31
R15	2	600	15
mk2	2	1000	2
z2	2	900	5

Table 2. Results of experiment

表 2. 实验结果

name	Algorithm	ACC	DBI	AS
LSVT	kmeans	0.5892	1.7391	0.5350
	three-way-kmeans	0.6082	1.4255	0.6819
Statlog	kmeans	0.5922	1.2532	0.4685
	three-way-kmeans	0.6004	1.1873	0.4826
Urban	kmeans	0.6095	1.9810	0.2484
	three-way-kmeans	0.6126	1.8859	0.2493
dermatology	kmeans	0.7062	1.9524	0.3633
	three-way-kmeans	0.7294	1.6746	0.3883
D31	kmeans	0.8661	0.6668	0.7057
	three-way-kmeans	0.8731	0.5592	0.7504
R15	kmeans	0.8843	0.5065	0.8022
	three-way-kmeans	0.9085	0.3471	0.8770
mk2	kmeans	0.5435	1.1328	0.5249
	three-way-kmeans	0.5449	1.1277	0.5265
z2	kmeans	0.6909	0.7523	0.6136
	three-way-kmeans	0.6910	0.7339	0.6247

6. 总结

传统的聚类方法有两个问题：一是强行将所有对象划入某一类簇中会带来决策风险；二是对于属性较多的数据集效果会很差。本文提出了一种基于属性集成的三支聚类方法，通过对同一数据集多次根据不同的属性列进行聚类，并根据聚类的结果来决定对象该划分至核心域或边界域中，既能提高聚类的准确性，又可以处理高维度数据。实验结果表明本文提出的方法可以有效提高聚类的准确性，同时在 *DBI* 与 *AS* 指标上相较于二支聚类也有较好的效果。实验过程中发现，该算法中的许多因素，如对象划分至核心域的标准或属性的选取方式、选取次数等，均会影响结果的准确性。后续实验将定量分析这些参数对实验结果的影响，并提升算法的效率。

基金项目

本文受到江苏省高等学校大学生创新创业训练计划项目(项目编号 202210289036Z)资助。

参考文献

- [1] Xu, D. and Tian, Y. (2015) A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, **2**, 165-193. <https://doi.org/10.1007/s40745-015-0040-1>
- [2] Hoppner, F., Klawonn, F., Kruse, R., et al. (1999) *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, New York.
- [3] Lingras, P. and West, C. (2004) Interval Set Clustering of Web Users with Rough k-Means. *Journal of Intelligent Information Systems*, **23**, 5-16. <https://doi.org/10.1023/B:JIIS.0000029668.88665.1a>
- [4] Yao, Y.Y., Lingras, P., Wang, R.Z., et al. (2009) Interval Set Cluster Analysis: A Re-Formulation. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., et al., Eds., *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Springer, Berlin, 398-405. https://doi.org/10.1007/978-3-642-10646-0_48
- [5] Yu, H., Chu, S.S. and Yang, D.C. (2012) Autonomous Knowledge-Oriented Clustering Using Decision-Theoretic Rough Set Theory. *Fundamenta Informaticae*, **115**, 141-156. <https://doi.org/10.3233/FI-2012-646>
- [6] Yu, H., Liu, Z.G. and Wang, G.Y. (2014) An Automatic Method to Determine the Number of Clusters Using Decision Theoretic Rough Set. *International Journal of Approximate Reasoning*, **55**, 101-115. <https://doi.org/10.1016/j.ijar.2013.03.018>
- [7] Yu, H., Chu, S.S. and Yang, D.C. (2010) Autonomous Knowledge-Oriented Clustering Using Decision-Theoretic Rough Set Theory. In: Yu, J., Greco, S., Lingras, P., et al., Eds., *Rough Set and Knowledge Technology*, Springer, Berlin, 687-694. https://doi.org/10.1007/978-3-642-16248-0_93
- [8] Yao, Y.Y. (2010) Three-Way Decisions with Probabilistic Rough Sets. *Information Sciences*, **180**, 341-353. <https://doi.org/10.1016/j.ins.2009.09.021>
- [9] Yao, Y.Y. (2011) The Superiority of Three-Way Decisions in Probabilistic Rough Set Models. *Information Sciences*, **181**, 1086-1096. <https://doi.org/10.1016/j.ins.2010.11.019>
- [10] Yao, Y.Y. (2012) An Outline of a Theory of Three-Way Decisions. In: Yao, J., Ed., *Rough Sets and Current Trends in Computing*, Springer, Berlin, 1-17. https://doi.org/10.1007/978-3-642-32115-3_1
- [11] Little, M.A., McSharry, P.E., Hunter, E.J. and Ramig, L.O. (2009) Suitability of Dysphonia Measurements for Tele-monitoring of Parkinson's Disease. *IEEE Transactions on Biomedical Engineering*, **56**, 1015-1022. <https://doi.org/10.1109/TBME.2008.2005954>
- [12] Tsanas, A., Little, M.A., Fox, C. and Ramig, L.O. (2014) Objective Automatic Assessment of Rehabilitative Speech Treatment in Parkinson Disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **22**, 181-190. <https://doi.org/10.1109/TNSRE.2013.2293575>
- [13] Fänti, P. and Sieranoja, S. (2018) K-Means Properties on Six Clustering Benchmark Datasets. *Applied Intelligence*, **48**, 4743-4759. <https://doi.org/10.1007/s10489-018-1238-7>
- [14] Veenman, C.J., Reinders, M.J.T. and Backer, E. (2002) A Maximum Variance Cluster Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 1273-1280. <https://doi.org/10.1109/TPAMI.2002.1033218>
- [15] Schölkopf, B., Platt, J. and Hofmann, T. (2007) A Local Learning Approach for Clustering. In: *International Conference on Neural Information Processing Systems*, MIT Press, Vancouver, 1529-1536. <https://doi.org/10.7551/mitpress/7503.001.0001>
- [16] Fahad, A., Alshatri, N., Tari, Z., et al. (2014) A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. *IEEE Transactions on Emerging Topics in Computing*, **2**, 267-279. <https://doi.org/10.1109/TETC.2014.2330519>