The Performance of Objective Functions for Clustering Categorical Data*

Zhengrong Xiang[1] and Md Zahidul Islam[2]

[1] College of Computer Science, Zhejiang University, China
zolaxiang@gmail.com
[2] School of Computing and Mathematics, Charles Sturt University, Australia
zislam@csu.edu.au

Abstract. Partitioning methods, such as k-means, are popular and useful for clustering. Recently we proposed a new partitioning method for clustering categorical data: using the transfer algorithm to optimize an objective function called within-cluster dispersion. Preliminary experimental results showed that this method outperforms a standard method called k-modes, in terms of the average quality of clustering results. In this paper, we make more advanced efforts to compare the performance of objective functions for categorical data. First we analytically compare the quality of three objective functions: k-medoids, k-modes and within-cluster dispersion. Secondly we measure how well these objectives find true structures in real data sets, by finding their global optima, which we argue is a better measurement than average clustering results. The conclusion is that within-cluster dispersion is generally a better objective for discovering cluster structures. Moreover, we evaluate the performance of various distance measures on within-cluster dispersion, and give some useful observations.

Keywords: Objective Function, Clustering, Categorical data, Transfer algorithm.

1 Introduction

Clustering is an important task in data mining [1,2]. A basic idea is that objects in the same cluster are similar to each other. Usually clustering is for discovering natural structures in data. There are also utility reasons like compression or summarization. Among different clustering schemes, partitioning methods such as k-means [3] and k-medoids [1] are extremely popular in practice. They define objective functions to be the goal of clustering, and they have heuristic algorithms to optimize the objective. In this paper, the objective functions we discuss are all from partitioning methods.

Clustering for categorical data can be different from numerical data, because the distance measures for categorical data has a different nature. For example,

* The second author would like to thank the Faculty of Business Compact Fund R4 P55 in Charles Sturt University, Australia.

Y.S. Kim et al (Eds.): PKAW 2014, LNCS 8863, pp. 16–28, 2014.

the definition of center in k-means does not directly apply for categorical data. In this regard, k-modes [4] is designed specifically for categorical data in the framework of partitioning methods. It has a different definition of center than k-means, but the optimization algorithm is similar.

Recently another partitioning method [5] for clustering categorical data is proposed. The objective function is called within-cluster dispersion, and it emphasizes pairwise similarities between objects in a cluster. The optimization method is a version of transfer algorithm [6], which is a general procedure for optimizing any form of objective functions. This method is as efficient as k-modes but produces clustering results with better average quality.

In this paper, we focus on comparing the performance of three major objective functions for categorical data: k-medoids, k-modes and within-cluster dispersion. We analyze what kind of cluster structures those objectives define and experiment on how good they cluster real data sets. We measure the performance with respect to global optima, which we argue is a more convincing way to decide the goodness of objective functions.

One advantage of the within-cluster dispersion objective is that it can be used with any distance measures. In practice, it gives flexibilty for users. We can use different measures to achieve multiple clustering results [7]. Then we can either choose a best result or learn from different perspectives. It will be interesting to know how different distance measures affect clustering results when using this objective.

Another reason for evaluating performance of distance measures is the lack of study in this topic. For numerical data, the distance measure is usually Minkowski distance. For categorical data, it remains a open question. There has been a study [8] comparing distance measures on the task of outlier detection, but no study has been conducted on the task of clustering. In this paper, we show within-cluster dispersion is an objective function of good quality, thus evaluating distances on this objective is something significant to carry out.

Contributions of this paper are:

1. For partitioning methods in clustering categorical data, we analyze the quality of three objective functions: k-medoids, k-modes and within-cluster dispersion. The main conclusion is that within-cluster dispersion discovers structures better than the other two.
2. Various benchmark data sets are used to evaluate the performance of the three objective functions. We compare the global optima rather than average clustering results, to make the comparison more convincing. The results are consistent with our analysis.
3. On the within-cluster dispersion objective, we evaluate the performance of various data-driven distance measures and provide some useful observations.

2 Related Work

The objective functions we discuss here are from partitioning relocation clustering methods. They are highly efficient, and easily explainable because of clear goals defined by objective functions. K-means is most popular and typical, which is usually for numerical data. For categorical data, three major algorithms are k-medoids, k-modes and the transfer algorithm.

The algorithm of k-medoids [1] has the same structure as k-means, except that the centers of clusters are medoids. A medoid is defined to be the object whose sum of distances to other objects is minimal. One advantage of k-medoids over k-means is that the medoid can be computed with respect to any distance measures.

K-modes [4,9] is also a k-means-like algorithm, but it's specifically used for categorical data. The center here is called mode. It takes the same form as an object, with each attribute value being the most frequent value in the cluster.

Recently another method is proposed to use transfer algorithm for clustering categorical data [5]. The objective, called within-cluster dispersion, is traditionally a cluster evaluation measure [10]. With the appropriate design of the transfer algorithm, the objective can be locally optimized as efficient as k-modes. The average of clustering results shows that this method discovers more real structures than k-modes.

For numerical data, there have been empirical performance evaluations of various objective functions. Some evidence suggest that the classical k-means objective performs better than others, although the k-means objective tends to result in clusters of approximately the same size and shape [3].

For performance evaluation of data-driven distance measures of categorical data, there is some good work in [8]. They bring together fourteen distance measures, and evaluated them in the task of outlier detection.

Books on cluster analysis [2,10] usually do not include these data-driven distance measures. They discuss more about measures for binary data that are independent from data sets, while data-driven distances use helpful information from data sets.

Some clustering algorithms define distance measures based on neighbors [11,12]. Definition of neighbor uses simple distance measures like the simple matching distance. The distances we use here are different, in that they directly calculate the distances between two objects.

3 Quality Analysis of Objective Functions

3.1 Perspective of Cluster Structures

For the objective function of k-means, there is an equivalence as follows:

$$\sum_{k=1}^{K}\sum_{i\in C_k}\left(x_i-\overline{x}^{(k)}\right)^2=\sum_{k=1}^{K}\frac{1}{2n_k}\sum_{i\in C_k}\sum_{j\in C_k}(x_i-x_j)^2 \qquad (1)$$

In the equation, K is the number of clusters, x_i is the ith object of a data set. For the simplicity of notations, we assume the data is one-dimensional. $\bar{x}^{(k)}$ is the mean of cluster k. n_k is the number of objects in cluster k.

The left side of the equation is based on centers, while the right side of the equation computes all pairwise distances between objects in the same cluster. Because the k-means algorithm is closely related to centers (assign objects to nearest cluster center and update centers), the perspective reflected by the right side of the equation is often neglected. It's actually an important perspective and a basic idea of cluster definition: objects in a cluster are similar with each other. This definition is obviously very useful because k-means has been proved to be very successful in discovering structures in many applications.

The objective of within-cluster dispersion is the same as the right side of equation (1), except that it replaces the squared Euclidean distance with a general distance measure for categorical data. See equation (2). So within-cluster dispersion also defines clusters by calculating pairwise similarities between all objects in a cluster (Figure 2).

$$Dispersion = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} \sum_{j \in C_k} d(x_i, x_j) \tag{2}$$

Now let's check k-medoids and k-modes from this perspective. Their objective functions have a same form:

$$\sum_{k=1}^{K} \sum_{i \in C_k} d(x_i, m_k) \tag{3}$$

Where m_k is medoid and mode respectively for the kth cluster. For k-medoids, the medoid is defined to be one of the objects that minimizes the sum of distances, thus its objective function sums over much less number of distances than all pairwise distances (see the comparison between Figure 1 and Figure 2).

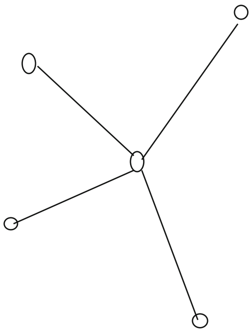

Fig. 1. Star Structure of K-medoids

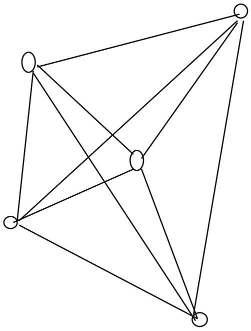

Fig. 2. Net Structure of Within-cluster Dispersion

For k-modes, the way objects connect with each other is a little more complicated. The mode is not necessarily a real object like a medoid, but a virtual object: each attribute of a mode can take any value from that attribute. The mode is chosen to minimize the sum of distances from objects in the respective cluster to the mode. On each attribute of a mode, the value minimizes the sum of distanes between values of objects and the value of the mode. So for each attribute, k-modes is like doing a one-dimensional k-medoids (except that some one-dimensional objects are duplicated).

In the following example, we assume the data has two attributes. In Figure 3, assume object x_1 and object x_2 have a same attribute value. Also reasonably assume that this value is the most frequent value on the first attribute, thus it's the minimizer. So all other objects are connected to these two objects because these pairs of distances are included in the objective function. We used dashed lines to represent a "partial connection" because the connection is only on one attribute rather than both attributes. In Figure 4, x_2 and x_3 have the minimizer value on the second attribute, and dashed lines are similarly connected. In Figure 5, we add the effects of two attributes together to get the whole picture of how objects interact. For a pair of objects, if there are dashed lines in both Figure 3 and Figure 4, a solid line is plotted in Figure 5 meaning that they have full connection. From Figure 5, we can see that the k-modes considers only a part of the pairwise distances as in the case of within-cluster dispersion.

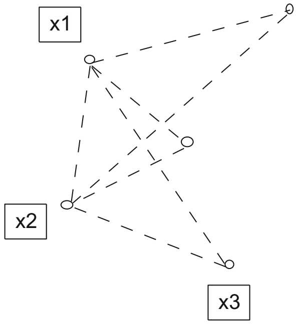

Fig. 3. K-modes on the First Attribute

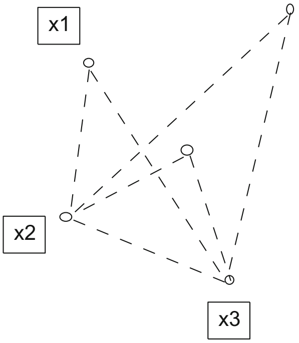

Fig. 4. K-modes on the Second Attribute

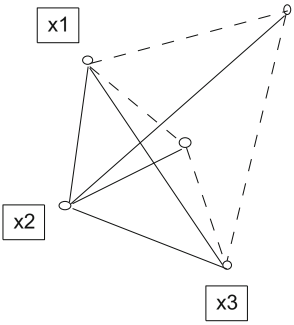

Fig. 5. Adding up Figure 3 and Figure 4: Partial-Net Structure of k-modes

We have presented the different cluster structures that three objectives define. Now the question is which cluster structure is more dominant in real data sets, because in exploratory data analysis, we want to find the natural structures in data. In the fully connected net structure, all objects are supposed to be similar with each other. It's a more compact cluster than k-modes and k-medoids. If real-life clusters are such good quality clusters, then within-cluster dispersion is better suited to discover them. One way to know what real-life clusters are like is to look at k-means. K-means defines clusters to be the net structure and k-means is widely recognized as being successful.

Another argument from us is that the net structure is more dominant because that's how objects in natural clusters interact with each other. For populations of plants, pollination happens to all adjacent plants. For a human society, people are constantly moving and communicating with others. Objects interact with many others that are near them, not with only one "central" object. In this reasoning, within-cluster dispersion also performs better than k-modes, and k-modes better than k-medoids.

In the experiment section, we show the superiority of within-cluster dispersion at discovering structures in real data sets.

3.2 Perspective of Informativeness

In this perspective, we compare how informative the objective functions are in terms of describing clusters. From the discussion of cluster structures, we can see that the computation of within-cluster dispersion involves the most amount of distances/similarities. Obviously, the more distances computed, the more information an objective provides.

We can also reach this conclusion by comparing the centers of the objective functions. Within-cluster dispersion can also be written in a center-based form. We use simple matching distance to illustrate. Simple matching distance between objects x and y on d attributes is defined as:

$$d(x,y) = \sum_{j=1}^{d} d_j(x_j, y_j) \quad d_j(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

If we transform the categorical data into binary data, and treat the binary data as numerical data, then within-cluster dispersion on the original data is equivalent to the objective function of k-means on the transformed data. This is because simple matching distance is essentially equivalent to Euclidean distance on 0/1 data. So the center of k-means is our virtual "center" for within-cluster dispersion. Obviously, the dimension of the center is the same as the number of all attribute values, and the entry on each dimension is proportional to the number of objects taking the respective attribute value:

$$(\frac{f(A_{11})}{N}, ..., \frac{f(A_{1p_1})}{N}, ..., \frac{f(A_{d1})}{N}, ..., \frac{f(A_{dp_d})}{N}) \tag{5}$$

Where A_{ij} denotes the jth value of the ith attribute, and $f(A_{ij})$ is the number of objects taking value A_{ij}, N being the total number of objects, p_d being the number of values in attribute A_d.

This center obviously has more information than the centers of k-modes and k-medoids. For each attribute, k-modes takes the attribute value with the biggest entry in vector (5), while k-medoids restricts the values to be from one real object. So although mode and medoid both have the same dimension as the number of attributes, mode has more significant information. If we see objective functions as a measure for the information in clusters, the more informative an objective is, the better. In this sense, within-cluster dispersion is the best, and k-modes better than k-medoids.

Informativeness means how well an objective can distinguish the quality of a cluster. If a distance between two objects is not reflected in the objective, changing it might not affect the value of the objective. Thus the two clusters before and after change are not distinguished by the objective. In practice, lacking the ability to distinguish between clusters can result in a problem: for a single objective value, several different structures are found. If this objective value is the minimum and chosen to be the best result, there are no apparent ways to choose one structure among several options.

Note that there is no way to strictly prove that one objective function is better than another one. The reason is that clustering in real data sets can be arbitrary. That's why there are so many different clustering methods that define different concepts of clusters. For one data set, say an objective function A finds structure S_A, and it's better than S_B found by objective B. However, we can always change the data labels to something else (creating a new data set), so that B performs better than A. So to summarize this section, the argument we are trying to make is: within-cluster dispersion is "generally" better than the other two objectives.

4 Distance Measures

One advantage of within-cluster dispersion over k-modes is that it doesn't define a specific distance measure. Since within-cluster dispersion is a very good objective function, it is useful to evaluate how different distance measures affect its clustering performance. In this section, we introduce the distance measures [8] we use for evaluation.

The limitation of the simple matching distance is that it treats all categorical values the same. Data-driven measures use the characteristics from a particular data set to define distances. For example, if an attribute of two objects has the same categorical value, and that value is rare in the data set, it might be a good idea to decide that this rare match shows more similarity (less distance).

Assume a data set has N objects, each has d attributes, $A_1, ...A_d$. Then we can use the following information from the data set:

n_k: The number of attribute values for attribute A_k.

$f_k(x)$: The number of times (frequency) attribute value x appears in attribute A_k.

$\hat{p}_k(x)$: The sample probability of attribute A_k takes value x, given by: $f_k(x)/N$.

$p_k^2(x)$: Another probability measure of attribute A_k takes value x, given by:

$$p_k^2(x) = \frac{f_k(x)(f_k(x)-1)}{N(N-1)} \tag{6}$$

From Boriah [8], some of the measures are distance measures in the original form. They can be directly applied here. For others, we use a simple but effective way to transform similarity measures into distance:

$$distance = 1 - similarity \tag{7}$$

All the measures are calculated by summing over per-attribute distances:

$$d(X, Y) = \sum_{k=1}^{d} d_k(X_k, Y_k) \tag{8}$$

The measures are listed in Table 1. They have different ideas of how to incorporate the characteristics of a data set. For example, IOF says that mismatches between higher frequency values are stronger, thus a larger distance is assigned. Some of the measures from [8] are too complicated and they have similar ideas about how to use the information in data sets. These measures are not included in our study. Note that for all these measures, the transfer algorithm can be carried out in a time complexity that is linear to data size N.

5 Experiments

5.1 Quality of Objective Functions

In Section 3, the conclusion of the analysis is that the within-cluster dispersion discovers better (more real) structures than k-medoids and k-modes. In this section, the goal is to evaluate the objective functions with real data sets. If the structures an objective discovered fit well with the real clusters, then it's a good objective.

The data sets we use are from UCI machine learning repository [13]. Their characteristics are listed in Table 2. There are more than 5 categorical data sets in UCI repositary. We didn't include more data sets to the experiment because real data sets can be hard for all three objective functions to handle. For example, if objects in a real cluster (class) are hardly similar with others, any clustering methods don't work. The 5 data sets we do have are more or less suitable for the task of cluster analysis, but they are not deliberately chosen in favor of any of the three objectives.

In this paper we compare the quality of different objective functions when their global optima are reached. One other option is to compare average clustering results, which means the average of any optima (global or local) when the algorithms converged. We argue that global optimum is a better criterion for deciding the goodness of objective functions. There are two reasons. One is that the quality of local optima depends on the optimization algorithm. For example, in k-means, Hartigan's method finds better optima than the common Lloyd's method [14,15]. However, the thing we want to find out is how good the objective function is, not the goodness of the optimization algorithm. Global optimum is the result that an objective function can provide at its best. The other reason is, for heuristic algorithms like k-means, the standard way is to run the algorithm multiple times (say 1000) and pick the result with the minimum

Table 1. Distance Measures For Categorical Data

$Measures$	$d_k(X_k, Y_k)$
$Eskin$	$= \begin{cases} 0; & if\ X_k = Y_k \\ \frac{2}{n_k^2}; & otherwise \end{cases}$
IOF	$= \begin{cases} 0; & if\ X_k = Y_k \\ \log f_k(X_k) log f_k(Y_k); & otherwise \end{cases}$
OF	$= \begin{cases} 0; & if\ X_k = Y_k \\ \log \frac{N}{f_k(X_k)} log \frac{N}{f_k(Y_k)}; & otherwise \end{cases}$
$Goodall1$	$= \begin{cases} \sum_{q \in Q} p_k^2(q); & if\ X_k = Y_k \\ 1; & otherwise \end{cases}$ $\{Q \subseteq A_k : \forall q \in Q, p_k(q) \leq p_k(X_k)\}$
$Goodall2$	$= \begin{cases} \sum_{q \in Q} p_k^2(q); & if\ X_k = Y_k \\ 1; & otherwise \end{cases}$ $\{Q \subseteq A_k : \forall q \in Q, p_k(q) \geq p_k(X_k)\}$
$Goodall3$	$= \begin{cases} p_k^2(X_k); & if\ X_k = Y_k \\ 1; & otherwise \end{cases}$
$Goodall4$	$= \begin{cases} 1 - p_k^2(X_k); & if\ X_k = Y_k \\ 1; & otherwise \end{cases}$
$Gambaryan$	$= \begin{cases} 1 + \hat{p}_k(X_k)\log_2 \hat{p}_k(X_k) + & \\ (1 - \hat{p}_k(X_k)) \log_2 (1 - \hat{p}_k(X_k)); & if\ X_k = Y_k \\ 1; & otherwise \end{cases}$

Table 2. Characteristics of Benchmark Data Sets

	Mushroom	Congress	Promoter	Soybean	Splice
Number of Objects	8124	435	106	47	3190
Number of Attributes	22	16	58	35	61
Number of Classes	2	2	2	4	3

objective value. As the computing power grows in modern days, the best result from thousands of runs is very likely to be the global optima [16].

In our experiment, in order to increase the chance of finding the global optima, we run the algorithms for as many times as possible. Note that we use random initial conditions to make the clustering results as diverse as possible. Although we can not be 100% sure that global optima are found, but it's very likely to be true: for all data sets the optima we get from 1000 runs are already highly duplicated.

The results are shown in Table 3 and Table 4. In Table 3, different numbers of clusters are set on the Mushroom data set. In Table 4, the number of clusters of the four data sets is the same as the number of true clusters(classes). The performance measure is purity [10] (also called accuracy in literature like [17]), which measures how well cluster results correspond to real structures in data sets. a_k is the number of objects from the most dominant class in cluster k.

$$Purity = \frac{\sum\limits_{k=1}^{K} a_k}{N} \tag{9}$$

From the results, we can see that within-cluster dispersion outperforms the other two, which is consistent with the analysis in Section 3. We can also see that in most cases, k-modes outperforms k-medoids. This is also reflected in our analysis: k-modes defines a more real cluster structure than k-medoids.

Table 3. Clustering Performance of Three Objectives on Mushroom Data Set

	k=2	k=3	k=4	k=5	k=6
k-medoids	0.879	0.772	0.854	0.85	0.855
k-modes	0.891	0.719	0.856	0.888	0.888
Dispersion	**0.892**	**0.894**	**0.894**	**0.894**	**0.894**

Table 4. Clustering Performance of Three Objectives on Four Data Sets

	Soybean	Promoter	Congress	Splice
k-medoids	0.936	0.864	0.509	0.519
k-modes	**1**	0.864	0.528	0.519
Dispersion	**1**	**0.880**	**0.623**	**0.845**

In Section 3.2, we mentioned that due to the uninformativeness of k-modes and k-medoids, different clustering structures can have a same value of objective function. This problem is exposed in the experiments on real data sets. For example, using k-modes in the Mushroom data set (number of clusters set to 2), for a local optimum of 62534, some results have an accuracy of 0.871, others have an accuracy of 0.884. In practice, if this kind of local optimum happens to be the minimum after some runs, it's a problem for users to choose among different clusterings.

5.2 Evaluation of Distance Measures

We evaluated nine data-driven distance measures on the five benchmark data sets in Table 2. The results are shown in Table 5 and Table 6. Again, the goodness of discovered structure is measured by purity. In Table 5 the results are averaged from 1000 runs, while in Table 6, the results are recorded when global optima are achieved. Although the number of data sets is not quite big, but we can still make some interesting observations:

1. For one data set, different distance measures can have significantly different performances in discovering structures. For example, for the Mushroom data set, Eskin is a lot worse than IOF. So it's important to choose a suitable measure for a particular data set.
2. For data sets of a similar nature, distance measures can have consistent performances. For example, Goodall4 is good for two gene data sets (Promoter and Splice). OF is not good for the two plant data sets (Mushroom and Soybean in Table 6). So in practice, we can use the knowledge from previous data sets to make the choice of an appropriate distance.
3. Distances with opposite philosophies have significantly different results over one data set. For example, the performance of Goodall3 and Goodall4 on data set Splice. This is easily expected and in practice, if one measure doesn't work well, it's a good idea to choose an "opposite" one.
4. No distance measure performs badly across all data sets. For example, OF has bad performance on most data sets. But for the Congress data set, it is the best measure. This implies that these various distance measures can all somehow be useful, and they are worth a try in practical clustering tasks.

Table 5. Average Performance of Different Distance Measures

	Mushroom	Congress	Promoter	Soybean	Splice
SMD	0.7534	0.8805	0.8022	0.9650	0.8392
Eskin	0.7856	0.8805	0.8035	0.9122	0.8384
IOF	0.8182	0.8805	0.8160	0.9656	0.8687
OF	**0.8352**	**0.8828**	0.5881	0.8875	0.5972
Goodall1	0.7435	0.8805	0.7874	0.9350	0.7523
Goodall2	0.7404	0.8805	0.7663	0.9192	0.7430
Goodall3	0.7580	0.8805	0.7761	**0.9698**	0.7415
Goodall4	0.7480	0.8805	**0.9528**	0.9600	**0.8919**
Gambaryan	0.7652	0.8805	0.8226	0.9583	0.8707

Table 6. Performance of Different Distance Measures with Respect to Global Optima

	Mushroom	Congress	Promoter	Soybean	Splice
SMD	0.8922	0.8805	0.6226	1	0.8445
Eskin	0.8385	0.8805	0.6226	1	0.8455
IOF	**0.8987**	0.8805	0.6226	1	0.8774
OF	0.8469	**0.8828**	0.6226	0.8511	0.6009
Goodall1	0.8978	0.8805	0.6321	1	0.7520
Goodall2	0.8936	0.8805	0.6226	1	0.7508
Goodall3	0.8954	0.8805	0.6226	1	0.7455
Goodall4	0.8865	0.8805	**0.9528**	1	0.8928
Gambaryan	0.8912	0.8805	0.6226	1	**0.8962**

6 Conclusions

In this paper, we focused on the performance of objective functions for clustering categorical data. First, we analyzed the quality of three objective functions, by presenting what kind of structures each of them define, and how informative they are to measure cluster quality. Our conclusion is that within-cluster dispersion is generally better than k-medoids and k-modes for discovering structures. In experiments on benchmark data sets we measure the performance of objectives functions with respect to their global optima, and the results are consistent with the previous analysis. Secondly, for the objective of within-cluster dispersion, we evaluated how various distance measures affect the performance of clustering results. Experiments exposed several interesting insights for the practice of cluster analysis.

References

1. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley & Sons (2009)
2. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis, 5th edn. John Wiley & Sons (2011)
3. Steinley, D.: K-means clustering: a half - century synthesis. British Journal of Mathematical and Statistical Psychology 59(1), 1–34 (2006)
4. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge discovery 2(3), 283–304 (1998)
5. Xiang, Z., Ji, L.: The use of transfer algorithm for clustering categorical data. In: Motoda, H., Wu, Z., Cao, L., Zaiane, O., Yao, M., Wang, W. (eds.) ADMA 2013, Part II. LNCS, vol. 8347, pp. 59–70. Springer, Heidelberg (2013)
6. Banfield, C.F., Bassill, L.C.: Algorithm AS 113. A transfer algorithm for nonhierarchical classification. Applied Statistics 26, 206–210 (1977)
7. Muller, E., Gunnemann, S., Farber, I., et al.: Discovering multiple clustering solutions: Grouping objects in different views of the data. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 1207–1210. IEEE (2012)
8. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. Red 30(2), 3 (2008)
9. Chaturvedi, A., Green, P.E., Caroll, J.D.: K-modes clustering. Journal of Classification 18(1), 35–55 (2001)
10. Pang-Ning, T., Steinbach, M., Kumar, V.: Introduction to data mining. Library of Congress (2006)
11. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. In: Proceedings of 15th International Conference on Data Engineering, pp. 512–521. IEEE (1999)
12. Palmer, C.R., Faloutsos, C.: Electricity based external similarity of categorical attributes. In: Whang, K.-Y., Jeon, J., Shim, K., Srivastava, J. (eds.) PAKDD 2003. LNCS (LNAI), vol. 2637, pp. 486–500. Springer, Heidelberg (2003)
13. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2013), `http://archive.ics.uci.edu/ml`

14. Telgarsky, M., Vattani, A.: Hartigan's Method: k-means Clustering without Voronoi. In: International Conference on Artificial Intelligence and Statistics, pp. 820–827 (2010)
15. Slonim, N., Aharoni, E., Crammer, K.: Hartigan's K-means versus Lloyd's K-means: is it time for a change? In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 1677–1684. AAAI Press (2013)
16. Steinley, D.: Local optima in K-means clustering: what you don't know hurt you. Psychological Methods 8(3), 294 (2003)
17. Ng, M.K., Li, M.J., Huang, J.Z., et al.: On the impact of dissimilarity measure in k-modes clustering algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 503–507 (2007)