Année 2021 **ANALYSE DE DONNÉES - MAD - ENSIIE**

Le projet sera rendu sous la forme d'un fichier `Rmd` et de sa version compilée en pdf envoyés à l'adresse christophe.ambroise@univ-evry.fr.

Variantes des k-means pour tableaux de distances

Exercise 1 Un algorithme des k-medoïdes

1. Programmer l'algorithme décrit dans la section 2 de l'article scientifique joint au sujet.
2. Simuler des données suivant le protocole décrit par la section 3.3 et le tableau 5 de l'article.
3. Comparer votre algorithme avec PAM du package `cluster` de R, les k-means et mclust. Attention, pour une comparaison fiable il faut répéter les simulations (100 répétitions dans l'article) et montrer des statistiques sur l'ensemble des résultats.

Exercise 2 Données `iris`

1. Appliquer votre algorithme au jeu de données `iris` et comparer aux k-means et à l'algorithme `PAM` du package `cluster` de R, et mclust.
2. Visualiser les différentes partitions sur les premiers plans d'une analyse en composantes principale.
3. Commentez.

Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 3336–3341
Expert Systems with Applications
www.elsevier.com/locate/eswa

A simple and fast algorithm for K-medoids clustering

Hae-Sang Park, Chi-Hyuck Jun *

Department of Industrial and Management Engineering, POSTECH, San 31 Hyoja-dong, Pohang 790-784, South Korea

Abstract

This paper proposes a new algorithm for K-medoids clustering which runs like the K-means algorithm and tests several methods for selecting initial medoids. The proposed algorithm calculates the distance matrix once and uses it for finding new medoids at every iterative step. To evaluate the proposed algorithm, we use some real and artificial data sets and compare with the results of other algorithms in terms of the adjusted Rand index. Experimental results show that the proposed algorithm takes a significantly reduced time in computation with comparable performance against the partitioning around medoids.

Keywords: Clustering; K-means; K-medoids; Rand index

1. Introduction

Clustering is the process of grouping a set of objects into clusters so that objects within a cluster are similar to each other but are dissimilar to objects in other clusters (Han, Kamber, & Tung, 2001). K-means clustering (MacQueen, 1967) and partitioning around medoids (Kaufman & Rousseeuw, 1990) are well known techniques for performing non-hierarchical clustering. K-means clustering iteratively finds the k centroids and assigns every object to the nearest centroid, where the coordinate of each centroid is the mean of the coordinates of the objects in the cluster. Unfortunately, K-means clustering is known to be sensitive to the outliers although it is quite efficient in terms of the computational time. For this reason, K-medoids clustering are sometimes used, where representative objects called medoids are considered instead of centroids. Because it is based on the most centrally located object in a cluster, it is less sensitive to outliers in comparison with the K-means clustering. Among many algorithms for K-medoids clustering, partitioning around medoids (PAM) proposed by Kaufman and Rousseeuw (1990) is known to be most powerful. However, PAM has a drawback that it works inefficiently for a large data set due to its time complexity (Han et al., 2001). This is the main motivation of this paper. We are interested in developing a new K-medoids clustering algorithm that should be simple but efficient.

There have been some efforts in developing new algorithms for K-medoids clustering. Kaufman and Rousseeuw (1990) also proposed an algorithm called CLARA, which applies the PAM to sampled objects instead of all objects. It is reported by Lucasius, Dane, and Kateman (1993) that the performance of CLARA drops rapidly below an acceptable level with increasing number of clusters. Lucasius et al. (1993) proposed a new approach of K-medoid clustering using a genetic algorithm, whose performance is reported as better than CLARA but computational burden increases as the number of clusters increases. Wei, Lee, and Hsu (2003) also compared performance of CLARA and some other variants for large data sets. Ng and Han (1994) proposed an efficient PAM-based algorithm, which updates new medoids from some neighboring objects. van der Laan, Pollard, and Bryan (2003) tried to maximize the silhouette proposed by Rousseeuw (1987) instead of minimizing the sum of distances to the closest medoid in PAM. Zhang and Couloigner (2005) suggested a K-medoid algorithm which utilizes triangular irregular network concept when calculating the total cost of the replacement in swap step of PAM to reduce the computational time. Most

* Corresponding author. Tel.: +82 54 279 2197; fax: +82 54 279 2870.
E-mail addresses: shoo359@postech.ac.kr (H.-S. Park), chjun@postech.ac.kr (C.-H. Jun).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.039

of these algorithms are based on PAM, so the computational burden still remains.

The remaining parts of this paper are organized as follows: The proposed method is introduced in the next section and performance comparison is presented for two real data sets and some artificial data sets. Other methods to find initial medoids are discussed and finally conclusions are given.

2. Proposed K-medoids algorithm

Suppose that n objects having p variables each should be grouped into k ($k < n$) clusters, where k is assumed to be given. Let us define jth variable of object i as X_{ij} ($i = 1,\ldots,n$; $j = 1,\ldots,p$). The Euclidean distance will be used as a dissimilarity measure in this study although other measures can be adopted. The Euclidean distance between object i and object j is given by

$$d_{ij} = \sqrt{\sum_{a=1}^{p}(X_{ia} - X_{ja})^2} \quad i = 1,\ldots,n;\ j = 1,\ldots,n \tag{1}$$

It should be noted that the above Euclidean distance will be adopted in K-means and PAM algorithms in this study.

The proposed algorithm is composed of the following three steps.

Step 1: (Select initial medoids)
- 1-1. Calculate the distance between every pair of all objects based on the chosen dissimilarity measure (Euclidean distance in our case).
- 1-2. Calculate v_j for object j as follows:

$$v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}, \quad j = 1, ..., n \tag{2}$$

- 1-3. Sort v_j's in ascending order. Select k objects having the first k smallest values as initial medoids.
- 1-4. Obtain the initial cluster result by assigning each object to the nearest medoid.
- 1-5. Calculate the sum of distances from all objects to their medoids.

Step 2: (Update medoids)
Find a new medoid of each cluster, which is the object minimizing the total distance to other objects in its cluster. Update the current medoid in each cluster by replacing with the new medoid.

Step 3: (Assign objects to medoids)
- 3-1. Assign each object to the nearest medoid and obtain the cluster result.
- 3-2. Calculate the sum of distance from all objects to their medoids. If the sum is equal to the previous one, then stop the algorithm. Otherwise, go back to the Step 2.

The above algorithm is a local heuristic that runs just like K-means clustering when updating the medoids. In Step 1, we proposed a method of choosing the initial medoids. This method tends to select k most middle objects as initial medoids. The performance of the algorithm may vary according to the method of selecting the initial medoids. We will consider some other possibilities of choosing the initial medoids and their performance will be compared with each other through simulation study in Section 3.4.

3. Numerical experiments

In order to see the performance of the proposed method, we first applied the method to two real data sets, 'Iris' data and 'Soybean' data, whose true classes are known. Performance was measured by the accuracy, which is the proportion of objects that are correctly grouped together against the true classes. To investigate the performance more objectively, a simulation study was carried out by generating artificial data sets repetitively and calculating the average performance of the method.

3.1. Iris data

The Iris data set is available in UCI repository (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/), which set includes 150 objects (50 in each of three classes – 'Setosa', 'Versicolor', and 'Virginica') having four variables ('sepal length', 'sepal width', 'petal length', and 'petal width'). We applied the proposed method and K-means with $k = 3$ to this data without using the class information. When implementing K-means, the initial centroids were chosen randomly although many other alternatives are available including Al-Daoud and Roberts (1996), Khan and Ahmad (2004), etc.

The class of an object cannot be predicted by a clustering algorithm but it may be estimated by examining the cluster result for the class-labeled data. Table 1 shows the confusion matrix by K-means clustering method, whereas Table 2 shows the result by the proposed method. The clustering accuracy against the true classes by K-means is

Table 1
Cluster result of iris data by K-means

From algorithm			
True	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	47	3
Virginica	0	14	36

Table 2
Cluster result of iris data by the proposed method

From algorithm			
True	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	41	9
Virginica	0	3	47

88.7%, whereas the clustering accuracy by the proposed method is 92%. K-means is wrongly grouping a considerable amount of objects in Virginica class mixed with the objects in Versicolor class.

3.2. Soybean data

We also considered 'Soybean' data set in UCI repository for comparing the performance of the proposed algorithm with K-means. This data set has 47 objects having 35 variables each. Four classes are labeled to the objects, but they have not been used when clustering.

Table 3 shows the confusion matrix resulted from K-means and Table 4 summarizes the result from the proposed method. The clustering accuracy by K-means for this data set is only 44.7%, whereas the accuracy by the proposed method is as high as 80.9%. Particularly, K-means is wrongly grouping all objects in class 2 mixed with objects in class 1.

3.3. Artificial data sets

In order to evaluate the performance of the proposed method more objectively, some artificial data sets will be generated and clustered by using the proposed method, K-means and PAM. We assumed that in each data set there are three clusters (clusters A, B and C) and 120 two-variable objects in each cluster. We generated x-coordinates and y-coordinates in cluster A independently from normal distribution with mean μ_x^A and standard deviation σ^A, denoted by $N(\mu_x^A, \sigma^A)$, and from $N(\mu_y^A, \sigma^A)$, respectively. Similarly, x- and y-coordinates in cluster B were generated from $N(\mu_x^B, \sigma^B)$ and $N(\mu_y^B, \sigma^B)$, respectively. However, objects in cluster C were generated somewhat differently. A specified proportion of objects (called noisy objects) in cluster C are assumed to have a larger standard deviation σ_L^C, say, than the standard deviation of the rest (σ^C), while the means of x- and y-coordinates of all objects in cluster C are same as μ_x^C and μ_y^C, respectively. For example, when 10% of noisy objects are specified, x- and y-coordinates of 12 objects in cluster C will be generated from $N(\mu_x^C, \sigma_L^C)$ and $N(\mu_y^C, \sigma_L^C)$, respectively, and two coordinates of the rest 108 objects in cluster C will be generated from $N(\mu_x^C, \sigma^C)$ and $N(\mu_y^C, \sigma^C)$, respectively. We selected the above parameters as in Table 5.

Fig. 1 shows one example of our artificial data set generated from the above setting when there are no noisy objects in cluster C. The objects marked by points, triangles and circles belong to clusters A, B and C, respectively.

In order to compare the performance of the proposed method with K-means clustering and PAM, the adjusted Rand index was employed. The adjusted Rand index proposed by Hubert and Arabie (1985) is popularly used for comparison of clustering results when the external criterion or the true partition is known. Suppose that U and V represent two different partitions of the objects under consideration and that U is the true partition and V is a clustering result. Then the adjusted Rand index for the clustering result V is calculated by

$$\mathrm{RI}_{\mathrm{adj}} = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} \tag{3}$$

where a is the number of pairs of objects that are placed in the same class in U and in the same cluster in V, b is the number of pairs in the same class in U but not in the same cluster in V, c is the number of pairs in the same cluster in

Table 3
Cluster result of Soybean data by K-means

From algorithm				
True	Class 1	Class 2	Class 3	Class 4
Class 1	10	0	0	0
Class 2	10	0	0	0
Class 3	0	1	4	5
Class 4	0	5	5	7

Table 4
Cluster result of Soybean data by the proposed method

From algorithm				
True	Class 1	Class 2	Class 3	Class 4
Class 1	10	0	0	0
Class 2	0	10	0	0
Class 3	0	0	8	2
Class 4	0	0	7	10

Table 5
Parameters used when generating objects

	Cluster A	Cluster B	Cluster C
Means	$\mu_x^A = 0,\ \mu_y^A = 0$	$\mu_x^B = 6,\ \mu_y^B = -1$	$\mu_x^C = 6,\ \mu_y^C = 2$
Standard deviations	$\sigma^A = 1.5$	$\sigma^B = 0.5$	$\sigma^C = 0.5$ $\sigma_L^C = 2$

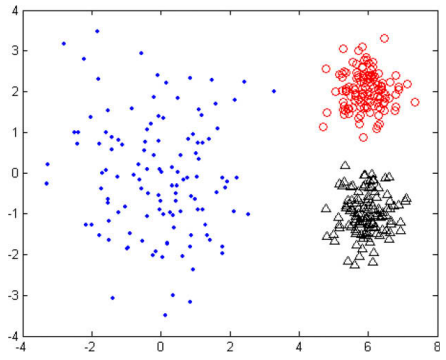

Fig. 1. Example of artificial data set generated.

Table 6
Adjusted Rand indices by various clustering methods

% Noisy objects	K-means	PAM	Proposed method
0	0.7903	0.9679	0.9629
5	0.8376	0.9534	0.9335
10	0.7836	0.9430	0.9430
15	0.7957	0.9288	0.9189
20	0.7305	0.9150	0.9115
25	0.7708	0.9053	0.8904
30	0.7750	0.8952	0.8915
35	0.7595	0.8782	0.8609
40	0.7624	0.8667	0.8671

V but not in the same class in U, and d is the number of pairs in different classes in U and different clusters in V.

Table 6 shows the calculated adjusted Rand index of each method under comparisons according to a different proportion of noisy objects contained. For example, when % noisy objects is 10 in Table 6, 120 objects will be generated for each of classes A and B, but 108 objects plus 12 noisy objects will be generated for class C. The result in Table 6 is in fact the average adjusted Rand index from 100 repetitions.

It can be clearly seen from Table 6 that the proposed method and the PAM perform much better than K-means clustering. Fig. 2 shows one cluster result from an artificial data set. As seen in this figure, K-means clustering sometimes divides class A into two groups and combines classes B and C as one group. The performance of the proposed method and PAM is very similar to each other regardless of the proportion of noisy objects although the PAM slightly outperforms the proposed method. The computational time required for PAM, however, increases rapidly as the number of objects increases. Fig. 3 shows the computation times required for PAM and the proposed method as a function of the number of objects. PAM takes about 20 s when there are 360 objects but it takes more than 100 s when the number of objects increases to 750, whereas the proposed method takes about the constant time near zero regardless of the number of objects. In fact, the complexity

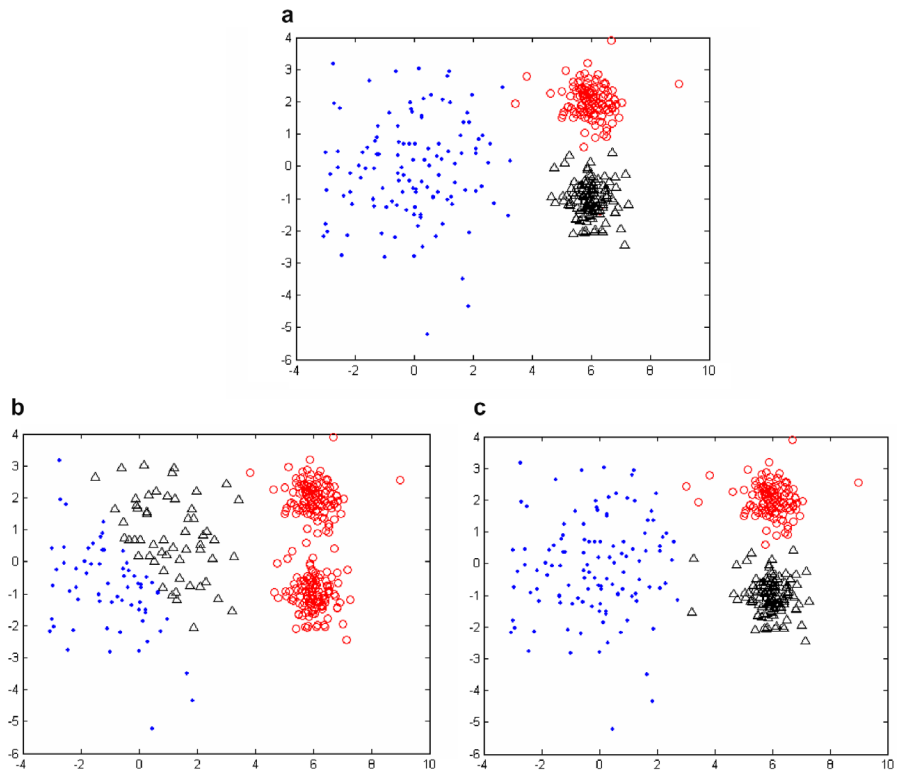

Fig. 2. True partitions and cluster results. (a) True partitions, (b) cluster result from K-means and (c) cluster result from PAM and the proposed method.

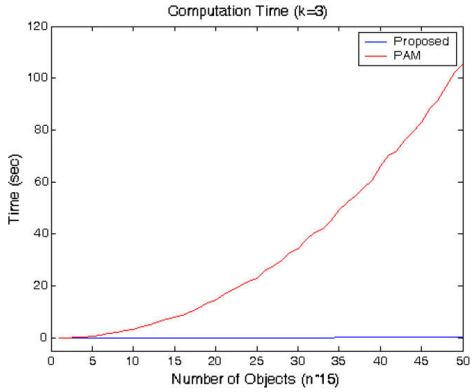

Fig. 3. Computation times of the proposed method with PAM.

of PAM is $O(k(n-k)^2)$ but that of the proposed method is $O(nk)$ which is equivalent to K-means clustering (Ng & Han, 1994). So, we may conclude that the proposed method is more efficient than PAM.

One may be curious of the performance of the proposed method in the case of a larger number of clusters. In order to investigate this, we carried out an additional experiment by generating an artificial data set having nine clusters as shown in Fig. 4. Each object has two variables, x- and y-coordinates and 100 objects were generated for each cluster. The two coordinates of each object were randomly generated from normal distributions having means at the center of a cluster and specified standard deviations. The first three clusters were generated at centers having 0 as x-coordinates, next three clusters at centers with $4\sqrt{2}$, and the last three clusters at centers with $8\sqrt{2}$ as x-coordinates. The standard deviations used when generating the first, the second and the last clusters were 1, 1.5 and 1, respectively so that the second clusters may overlap with the first and the last three clusters.

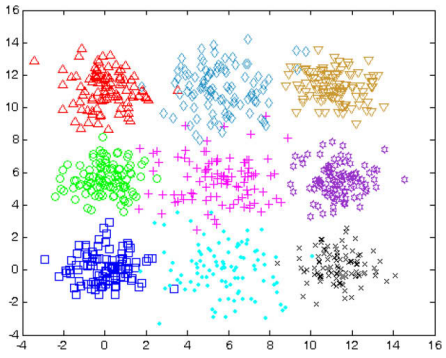

Fig. 4. Data set for nine clusters.

To see the performance of the proposed algorithm of clustering, the adjusted Rand index was calculated as before. In fact, we repeated the above experiment 1000 times and calculated the average value of the adjusted Rand index. It turned out that the average adjusted Rand index of the proposed algorithm was 0.8852 with standard deviation of 0.0682, whereas K-means gave the average value of 0.8561 with standard deviation of 0.0802. The performance was not compared with PAM because PAM requires huge amount of computational time, but it is expected that they are similar to each other.

3.4. Performance of the proposed method according to different initial medoids selection

The proposed method involves one way of selecting initial medoids in Step 1. Obviously, the performance may vary according to a different method of selecting initial medoids. Other alternatives may include the followings:

Method 1 (random selection): Select k objects randomly among all objects.

Method 2 (systematic selection): Sort all objects in the order of values of the chosen variable (first variable will be used in this study). Divide the range of the above values into k equal intervals and select one object randomly from each interval.

Method 3 (sampling): Take 10% randomly from all objects as a sample and perform a preliminary clustering on these sampled objects using the proposed algorithm. The resultant k medoids are used as the initial medoids.

Method 4 (outmost objects): Select k objects which are furthest from the center.

To compare the above methods for selecting initial medoids, data set is generated by the same way as before with 10% noisy objects in class C. Table 7 summarizes the results according to different numbers of total objects and different methods of selecting initial medoids, where the adjusted Rand indices were reported. Here again, the result is the average of 100 times of repetitions. It may be interesting to see that method 2 and 4 seem to be worse than the random selection. It was expected that method 3 should

Table 7
Adjusted Rand indices by different initial medoids selection

#Objects	Proposed	Method 1	Method 2	Method 3	Method 4
300	0.93927	0.8456	0.68002	0.91237	0.71532
600	0.92889	0.82134	0.6562	0.93896	0.78439
900	0.92832	0.81601	0.65237	0.92356	0.70749
1200	0.93135	0.84926	0.63543	0.92593	0.76650
1500	0.92939	0.81001	0.63680	0.92376	0.75256
1800	0.93771	0.83955	0.63531	0.93278	0.77791
2100	0.92736	0.79899	0.59487	0.91689	0.72579
2400	0.93755	0.82880	0.67166	0.92734	0.74584
2700	0.93284	0.78849	0.65120	0.94318	0.73119
3000	0.92201	0.80911	0.65068	0.9322	0.71507

be better than the proposed method and it is true when the number of objects is quite large. However, the improvement is not significant. It may be concluded that the initial medoids selection employed in the proposed method performs quite well as compared with other methods of naively selecting initial medoids. We may apply any clustering technique to obtain a cluster result and then use medoids of the clusters as initial medoids for the proposed algorithm.

4. Conclusion

In this paper, we propose a new algorithm for K-medoids clustering which runs like the K-means clustering. The algorithm has an excellent feature that it requires the distance between every pair of objects only once. The result from various simulations using artificial data sets shows that the proposed method has better performance than K-means clustering and that it takes a significantly reduced computation time than PAM with comparable performance. It can be also seen that the initial medoids selection employed in the proposed method performs quite well as compared with other methods of naively selecting initial medoids.

Acknowledgement

This work was supported by KOSEF through System Bio-Dynamics research center at POSTECH.

References

Al-Daoud, M. B., & Roberts, S. A. (1996). New methods for the initialization of clusters. *Pattern Recognition Letters, 17*, 451–455.

Han, J., Kamber, M., & Tung, A. K. H. (2001). Spatial clustering methods in data mining: A survey. In H. J. Miller & J. Han (Eds.), *Geographic data mining and knowledge discovery*. Taylor & Francis.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: Wiley.

Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for K-means clustering. *Pattern Recognition Letters, 25*, 1293–1302.

Lucasius, C. B., Dane, A. D., & Kateman, G. (1993). On k-medoid clustering of large data sets with the aid of a genetic algorithm: Background, feasibility and comparison. *Analytica Chimica Acta, 282*, 647–669.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). Berkeley: University of California Press.

Ng, R., & Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th international conference on very large databases, Santiago, Chile* (pp. 144–155).

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*(1), 53–65.

van der Laan, M. J., Pollard, K. S., & Bryan, J. (2003). A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation, 73*(8), 575–584.

Wei, C.-P., Lee, Y.-H., & Hsu, C.-M. (2003). Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems with Applications, 24*(4), 351–363.

Zhang, Q., & Couloigner, I. (2005). A new and efficient k-medoid algorithm for spatial clustering. *Lecture Notes in Computer Science, 3482*, 181–189.