THE MODIFICATION OF THE K-MEANS METHOD FOR CREATING NON-CONVEX CLUSTERS

Marta Žambochová

Abstract

Cluster analysis involves many different methods based on a variety of principles. Each of these methods has its advantages and disadvantages. The aim of authors of the new algorithms is to search for new methods that use existing methods positives and minimize the negatives. In this article, we discuss one of the most famous and most used method, the *k*-means method. The inability to search non-convex clusters is one of known weak points of the *k*-means method. Nevertheless, this method has many advantages such as simplicity and speed. This algorithm is also implemented in a lot of statistical software. The article presents a proposal for a solution. The second phase of the process is proposed. We create a certain number of clusters which is larger than desired in the first stage of processing by the classical algorithm *k*-means. Then we combine some clusters using appropriate agglomerative methods and reduce the number of clusters required in the second phase. There is the proposed procedure described and analyzed. The second aim was to find out how this new feature of the algorithm adversely affects the positive benefits of the original *k*-means method.

Key words: modification of algorithm, *k*-means method, non-convex clusters

JEL Code: C38, C44

Introduction

The *k*-means method is one of the basic methods of the cluster analysis. This is a non-hierarchical method that works on the principle of optimization. The algorithm finds the decomposition of objects into clusters for which the sum of distances of individual objects from their cluster centroid is minimal, ie minimization of the function (1).

$$Q = \sum_{x} \left\| x - c(x) \right\|^2 \qquad (1)$$

where x ... object, $c(x)$ ... the closest centroid of object x.

The basic *k*-means procedure:

- Step 0: An initial distribution of file data into *k* clusters.
- Step 1: The counting of every cluster's centroid.
- Step 2: The assignment of every object to the centroid.
- Step 3: If there is a change when compared with the previous iteration, return to Step 1.

Advantages of the *k*-means method

- Simple principle.
- Acceptable speed - the applicability to large data sets.
- Relatively good results (due to minimizing intra-variability).

Disadvantages of the *k*-means method

- It is applicable only for numeric data.
- It is necessary to enter the desired number of clusters.
- It searches only convex spherical clusters.
- It searches the local minimum only.
- The strong influence of outliers.
- It is very time consuming especially for large files.
- The strong influence of initial distribution into clusters.

Advantages of the *k*-means method are described for example in (Zambochova, 2008), (Rezankova, Husek & Snasel, 2009). The problem of finding only the local optima is determined by the principle of the algorithm, and therefore can not be removed. For example, (Rezankova, Loster & Husek, 2011) or (Rezankova, & Loster, 2013) is dealing with valuation of clusters in the case of categorical data. The way to set the desired number of clusters is proposed by the authors in (Rezankova, Husek & Snasel, 2008). The big impact of outliers is discussed in (Zambochova, 2010), (Zambochova, 2009). The authors of (Rezankova, Husek & Snasel, 2009) are highlighting restrictive time demands especially for large files. Reduction of the strong influence of the initialization distribution is proposed in (Zambochova, 2009a). Practical use of the described method is described for example in (Loster, & Pavelka, 2013) or (Pivonka, & Loster, 2013).

We offer description and solutions of the latest disadvantages of *k*-means method. We show the possible way to find non-convex clusters, using the modification of *k*-means method. All the described algorithms were programmed in the MATLAB development environment. They were tested and compared.

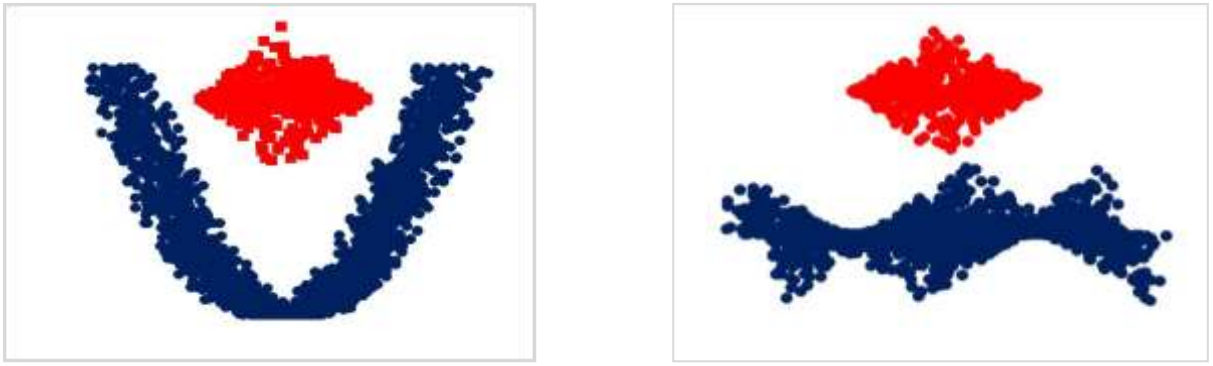
1 Using the *k*-means method for files containing non-convex clusters

To minimize the sum of squared distances of each object from certain centres is the basic principle of the *k*-means algorithm. It implies a spherical shape formed agglomerates. This fact is considered to be one of the negative characteristics of the *k*-means algorithm.

Inappropriate division into clusters in the case of files containing non-convex clusters is clearly visible from these two sets of simple images. First, the natural clusters as they really appear in the files are shown graphically. Also the clusters are seen as created using the basic *k*-means algorithm with a different parameter, in which the specific number of clusters is required to enter.

Fig. 1: The natural clusters

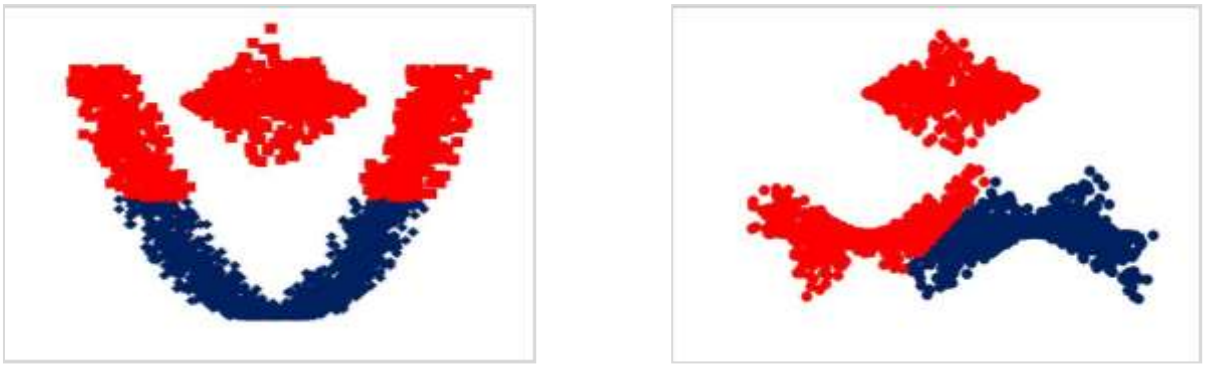
a) File I. **b) File II.**



Source: Own

Fig. 2: K-means metod, 2 clusters

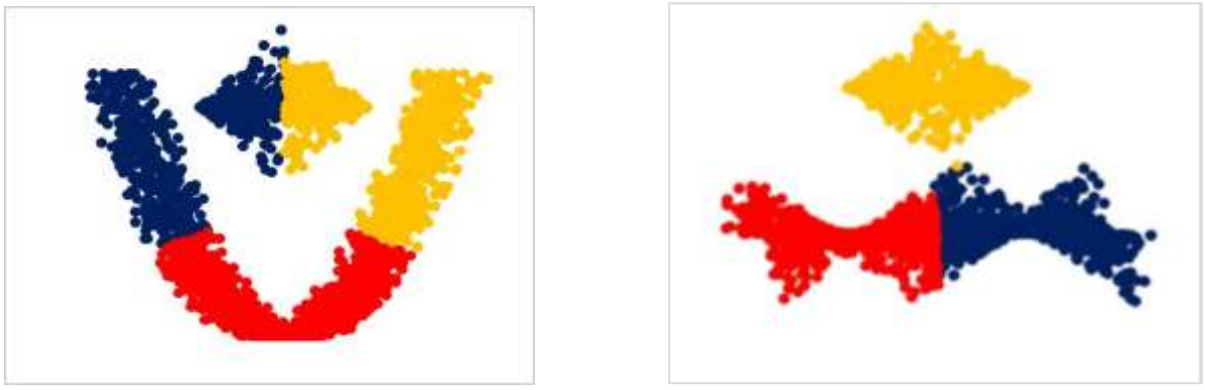
a) File I. **b) File II.**



Source: Own

Fig. 3: K-means metod, 3 clusters

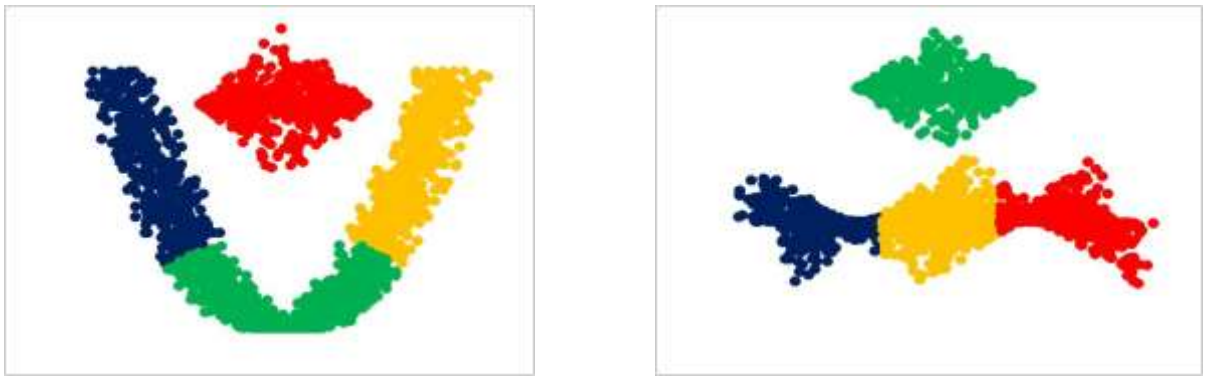
a) File I. **b) File II.**



Source: Own

Fig. 3: K-means metod, 4 clusters

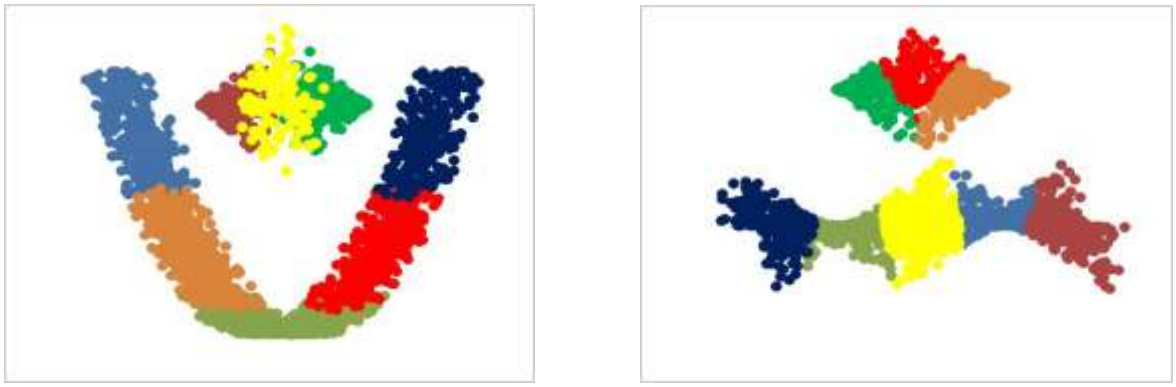
a) File I. **b) File II.**



Source: Own

Fig. 4: K-means metod, 8 clusters

a) File I. **b) File II.**



Source: Own

It is evident that the distribution is totally unsatisfactory, in case we create exactly the desired number of clusters using the *k*-means method. However, when we gradually divide the processed files to a larger number of clusters than is required, we will get into the situation where each of non-convex clusters is divided into several convex parts.

Therefore, the idea of performing the clustering in two stages originated. In the first phase, we use the *k*-means algorithm and a distribution to a larger number of clusters than is required. In the second stage of processing some clusters are appropriately merged to get the desired number of clusters. These clusters will be closer to the natural clusters shape than clusters created directly by the *k*-means.

2 Linking clusters methods

It was necessary to choose a suitable method for joining clusters. It can be used, for example one of the basic methods of agglomerative hierarchical clustering, which are mentioned in (Rezankova, Husek & Snasel, 2009), (Everit, Landau & Leese, 2001), (Starczewski, 2012). Unfortunately, many of them do not create the appropriate result. Some are too slow and thus unusable for large files. Other options are offered within the other algorithms. For example, the authors of (Karypis, Han & Kumar, 1999), (Guha, Rastogi & Shim, 2001) or (Kogan, Nicholas, Charles & Teboulle, 2006) offer some variants.

2.1 Simple linkage

- The method calculates the distance between clusters by taking the shortest of the distances of every two objects from two different clusters.
- The resulting clumps might not have a spherical shape.
- The disadvantage of this method is that if there are objects at the same distance from already existing clusters. This may lead to concatenation.
- High time-consuming.

2.2 Centroid linkage

- The Euclidean metric is used for calculating dissimilarity between objects. The distances of the clusters centres are measured.
- Less time-consuming.

2.3 Complete linkage

- The method calculates the distance between two clusters by taking the greatest possible distance from the distance of every two objects from two different clusters.
- Produces tight clusters of approximately equal size.
- It tends to form spherical clusters.
- Prevents chained clusters.
- High time-consuming.

2.4 Ward's method

- The method minimizes the increase of total intra sum of squares.
- The method forms clusters of comparable size.
- It forms spherical clusters.
- Very high time-consuming.

2.5 CURE algorithm

- The algorithm uses random sampling in progress.
- First, it randomly divides data into several parts.
- Clustering is carried out separately in each part. It creates the given number of representatives of each part.
- Clustering of the representatives of the pre-processing is performed and the basis for the creation of the resulting clusters is created.
- The complexity of the algorithm is $O(n^2 \cdot \log n)$, where n ... number of objects.

2.6 CHAMELEON algorithm

- The algorithm uses merging clusters based on dynamic programming in progress.
- It takes into account two characteristics:
    - Relative Inter-Connectivity
    - Relative closeness
- It uses user-adjustable threshold constants *TRI* and *TRC*, which are dynamically changing. They are changing in case that there are more merging opportunities, or none.
- The complexity of the algorithm is $O(n \cdot k + n \cdot \log n + k^2 \cdot \log k$, where n ... number of objects, k ... number of clusters.

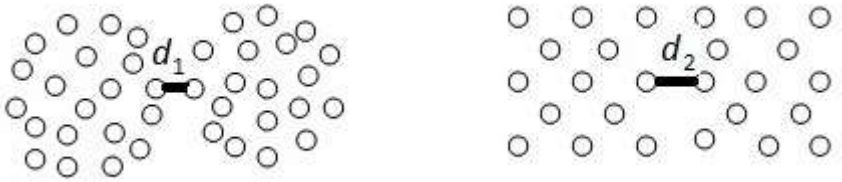
2.7 Selected problems of merging

The methods Simple linkage and CURE reflected the similar problem while connecting clusters. They take into account the minimum absolute distance between clusters as a criterion for joining. However, these methods do not consider clusters size and shape as criteria. This implies a problem that is well noticeable in the fig.5 or fig.6. Here we see that the left clusters are significantly differentiated clusters than the right ones. However, since the distance d_1 on the left is smaller than the distance d_2 between clusters on the right, just these two clusters on the left will be merged. This is obviously not right.

Fig. 5: The first problem

Source: (Karypis, Han & Kumar, 1999)

Fig. 6: The second problem



Source: (Karypis, Han & Kumar, 1999)

Conclusion

We tested the procedure on several artificially generated files containing non-convex clusters of various shapes and numbers. It has been shown in experiments that qualitatively better results are achieved if we set a higher value of the parameter determining the number of resulting clusters in the first stage of processing. Generally, the number of clusters formed during the first stage should be several times greater than the desired target number. On the other hand, the processing time increases with the increasing value of this parameter. Fortunately, it grows only linearly. Still, this number is a compromise between the quality of the resulting clustering and processing efficiency.

We performed subsequent merging using various methods mentioned above. Many of them formed clusters that were different from natural ones. Good results were given by the simple linkage method, CURE and CHAMELEON.

Big differences are in speed. The simple linkage method appeared to be the slowest. Conversely, methods Centroid linkage and CURE were faster. The CHAMELEON was the fastest method. The procedure of merging was therefore the most successful while using CHAMELEON method.

The work contains a description of the variant *k*-means method for finding clusters of generally non-spherical shape. Computational complexity of the algorithm is linear, which is a very positive feature. In further research, however, several tasks are still to be undertaken. It is necessary to verify the recommended number of clusters in the first phase. It is also necessary to verify the functionality of the algorithm for greater diversity in appearance clusters for different types of data distribution.

References

Everit, B. S., Landau, S., & Leese, M. (2001). *Cluster analysis*. (4 ed., p. 256). London: Hodder Arnold.

Guha, S., Rastogi, R., & Shim, K. (2001). CURE: An efficient clustering algorithm for large databases.INFORMATION SYSTEMS, 26(1), 35-58.

Karypis, G., Han, E., & Kumar, V. (1999). Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, *32*(8), 68-75.

Kogan, J., Nicholas, Charles, K., & Teboulle, M. (2006). Grouping multidimensional data: recent advances in clustering. (1 ed., p. 268). Springer.

Loster, T., & Pavelka, T. (2013). Evaluating of the Results of Clustering in Practical Economic Tasks. In Loster Tomas, Pavelka Tomas (Eds.), The 7th International Days of Statistics and Economics (pp. 804-818).

Pivonka, T., & Loster, T. (2013). Clustering of the countries before and during crisis. In Loster Tomas, Pavelka Tomas (Eds.), The7th International Days of Statistics and Economics (pp. 1110-1121).

Rezankova, H., Husek, D., & Snasel, V. (2008). *Clusters number determination and statistical software packals*. In *19TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, PROCEEDINGS*(pp. 549-553). Springer Verlag.

Rezankova, H., Husek, D., & Snasel, V. (2009). *Shluková analýza*. (2nd ed., p. 218). Prague: Professional Publishing.

Rezankova, H., Loster, T., & Husek, D. (2011). *Evaluation of categorical data clustering*. In *ADVANCES IN INTELLIGENT WEB MASTERING 3* (pp. 173-182). SPRINGER-VERLAG BERLIN.

Rezankova, H., & Loster, T. (2013). Shlukova analyza domacnosti charakterizovanych kategorialnimi ukazateli. *E+M. Ekonomie a Management ,16*(3), 139-147.

Starczewski, A. (2012). *A new hierarchical clustering algorithm*. In *11th International Conference on Artificial Intelligence and Soft Computing (ICAISC)* (pp. 175-180). SPRINGER-VERLAG BERLIN.

Zambochova, M. (2008). *Shlukování velkých souborů dat pomocí metod rozkladu*. In *Sborník ROBUST 2008* (pp. 533 – 540). Praha.

Zambochova, M. (2009). *Odlehlé objekty a shlukovací algoritmy*. In *Sborník MSED na VŠE* (pp. 1 – 6). VŠE Praha.

Zambochova, M. (2009a). *Inicializační rozdělení do shluků a jeho vliv na konečné shlukování v metodách k-průměrů*. In *Sborník prací účastníků vědeckého semináře doktorského studia FIS VŠE* (pp. 243 – 250).

Zambochova, M. (2010). Shlukování v souborech s odlehlými objekty pomocí metod k-průměrů. *Informační bulletin České statistické společnosti*, *22*(3), 123-130.

Contact

Marta Žambochová

Faculty of Social and Economic Studies, J. E. Purkyne University in Usti nad Labem

Moskevská 54, Ústí nad Labem, 400 96

marta.zambochova@ujep.cz