COMPARISON OF MODERN CLUSTERING ALGORITHMS FOR TWO-DIMENSIONAL DATA

[1]Martin Kotyrba, [1]Eva Volna, [2]Zuzana Kominkova Oplatkova

[1]Department of Informatics and Computers
University of Ostrava, 70103, Ostrava, Czech Republic
martin.kotyrba@osu.cz, eva.volna@osu.cz

[2]Tomas Bata University in Zlin , Faculty of Applied Informatics
Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
kominkovaoplatkova@fai.utb.cz

KEYWORDS

Cluster analysis, K-Means, Self Organising Map Algorithm, DBSCAN.

ABSTRACT

Cluster analysis or clustering is a task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is the main task of exploratory data mining and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.The topic of this paper is modern methods of clustering. The paper describes the theory needed to understand the principle of clustering and descriptions of algorithms used with clustering, followed by a comparison of the chosen methods.

INTRODUCTION TO CLUSTER ANALYSIS

Cluster analysis itself is not one specific algorithm, but a general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to find them efficiently. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, density threshold, or number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties. Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data.
A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.
Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy and typological analysis. Subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. This often leads to misunderstandings between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals. In this paper we will compare three different algorithms in an experimental study.

MODERN CLUSTERING METHODS

There are some well used clustering algorithms out there; one of them is the famous CLARANS. Other methods are K - means, K-medoid, Hierarchical Clustering and Self-Organized Maps. Nevertheless, none of these algorithms can handle all these three mentioned problems in a good way. This report will not discuss these methods but focus on the DBSCAN(Density Based Spatial Clustering of Applications with Noise) algorithm, which introduces solutions to these problems.

DBSCAN

It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.OPTICS can be seen as generalization of DBSCAN to multiple ranges, effectively replacing the Ɛ parameter with a maximum search radius. DBSCAN's definition of a cluster is based on the notion of density reachability. Basically, a point q is directly density-reachable from a point p if it is not farther away than a given distance Ɛ (i.e., it is part of its ε-neighborhood) and if p is surrounded by sufficiently many points such that one may consider p and q to be part of a cluster. Q is called density-reachable (note the distinction from "directly density-reachable") from p if there is a sequence $p_1 \ldots p_n$ of points with $p_1=p$ and $p_n=q$ where each p_{i+1} is directly density-reachable from p_i.

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

Note that the relation of density-reachable is not symmetric. *Q* might lie on the edge of a cluster, having insufficiently many neighbors to count as dense itself. This would halt the process of finding a path that stops with the first non-dense point. By contrast, starting the process with *p* would lead to *q* (though the process would halt there, *q* being the first non-dense point). Due to this asymmetry, the notion of density-connected is introduced: two points *p* and *q* are density-connected if there is a point 0 such that both *p* and *q* are density-reachable from 0. Density-connectedness is symmetric.

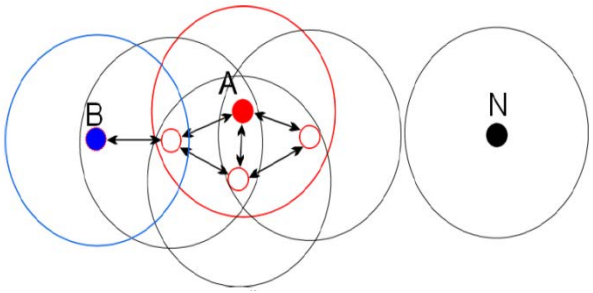

Figure 1: DBSCAN

A cluster, which is a subset of the points of the database, satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-connected to any point of the cluster, it is part of the cluster as well.

DBSCAN (Fig.1) requires two parameters: Ɛ(eps) and the minimum number of points required to form a cluster (*minPts*). It starts with an arbitrary starting point that has not been visited. This point's Ɛ -neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized Ɛ-environment of a different point and hence be made part of a cluster.If a point is found to be a dense part of a cluster, its Ɛ-neighborhood is also part of that cluster.

The algorithm DBSCAN:

1. *Select the object from the set of*
2. *Match all reachable points from the selected item, if applicable, the ε-neighborhood contains at least MinPts, will form a new cluster.*
3. *Find objects directly reachable from these cores may be joining clusters.*
4. *Cease at the moment when none of the remaining objects can no longer be added to anv cluster.*

Figure 2: The DBSCAN algorithm

Hence, all points that are found within the Ɛ-neighborhood are added, as is their own Ɛ -neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. DBSCAN disadvantage of the method is its sensitivity to parameter settings and Ɛ *MinPts*, the main advantage lies in the ability to distinguish clusters of different shapes, resistance to remote objects, and especially the detection of clusters.
In Fig. 2 you can see basic steps of DBSCAN algorithm.

K-Means Clustering

K-means (Fig.3) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define *k* centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

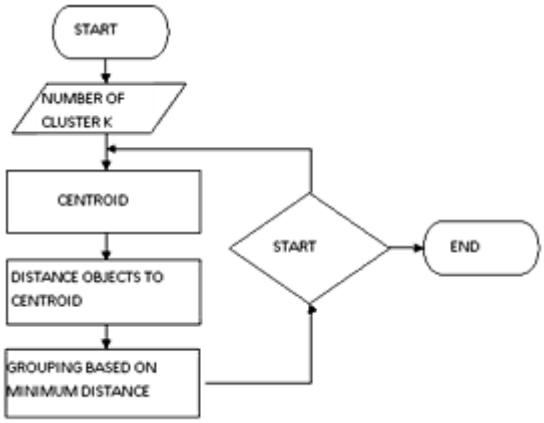

Figure 3: K-Means

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function (1)

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \qquad (1)$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster c_j is an indicator of the distance of the n data points from their respective cluster centres. The algorithm (Fig. 4) is composed of the following steps:

The algorithm K-Means

1. *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

Figure 4: The K-Means algorithm

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors.

SOM

The Kohonen Self-Organizing Feature Map (SOFM or SOM) is a clustering and data visualization technique based on a neural network viewpoint. As with other types of centroid-based clustering, the goal of SOM is to find a set of centroids (reference or codebook vector in SOM terminology) and to assign each object in the data set to the centroid that provides the best approximation of that object. In neural network terminology, there is one neuron associated with each centroid. As with incremental K-means, data objects are processed one at a time and the closest centroid is updated. Unlike K-means, SOM impose a topographic ordering on the centroids and nearby centroids are also updated. The processing of points continues until some predetermined limit is reached or the centroids are not changing very much. The final output of the SOM technique is a set of centroids that implicitly define clusters. Each cluster consist of the points closest to a particular centroid. SOM is a clustering technique that enforces neighborhood relationships on the resulting cluster centroids. Because of this, clusters that are neighbors are more related to one another than clusters that are not. Such relationships facilitate the interpretation and visualization of the clustering results. Indeed, this aspect of SOM has been exploited in many areas, such as visualizing Web documents or gene array data**.**

A distinguishing feature of SOM is that it imposes a topographic (spatial) organization on the centroids (neurons). An example of a two-dimensional SOM in which the centroids are represented by nodes that are organized in a rectangular lattice. Each centroid is assigned a pair of coordinates(i,j). Sometimes, such a network is drawn with links between adjacent nodes, but can be misleading because the influence of one centroid on another is via a neighborhood that is defined in terms of coordinates, not links. There are many types of SOM neural networks, but it will be focus on to two-dimensional SOMs with a rectangular or hexagonal organization of the centroids.

Even though SOM is similar to K-means, there is a fundamental difference. Centroids used in SOM have a predetermined topographic ordering relationship. During the training process, SOM uses each data point to update the closest centroid and centroids that are nearby in the topographic ordering. In this way, SOM produces an ordered set of centroids for any given data set. In other words, the centroids that are close to each other in the SOM grid are more closely related to each other than to the centroids that are farther away. Because of this constraint, the centroids of a two-dimensional SOM can be viewed as lying on a two-dimensional surface that tries to fit the n-dimensional data as well as possible. The SOM centroids can also be thought of as the result of a nonlinear regression with respect to the data points. At a high level, clustering using the SOM technique consists of the steps described in Algorithm which you can see in Fig.5.

The algorithm SOM:

1. *Determine the number of clusters.*
2. *Initialize the cluster centers.*
3. *Compute partitioning for data.*
4. *Compute (update) cluster centers.*
5. *If the partitioning is unchanged (or the algorithm has converged), stop; otherwise, return to step 3.*

Figure 5: The SOM algorithm

EVALUATION CRITERIA

On the basis of the experiments these parameters were chosen in individual methods: the ability to determine the number of clusters, sensitivity to outlying values, ability to distinguish clusters of arbitrary shape, sensitivity settings from the user. Table 1 shows a comparison of different methods based on the monitored parameters

Table 1: Parameter setting for individual algorithms

Name of method	Arbitrary shape of clusters	Sensitivity settings	Sensitivity to outlying objects	Determination of the number of clusters
DBSCAN	yes	yes	no	yes
SOM	yes	yes	no	yes
K-Means	no	no	yes	no

Based on this comparison, the following was chosen for implementation of these methods: method of k-means, SOM method and method DBSCAN. The SOM method was chosen as a representative model-based methods, DBSCAN is selected as the representative method based on density. The K-means method was chosen to be

implemented due to comparison with these modern methods. Another reason for its selection was illustrated by the inability of the clustering method to distinguish clusters of arbitrary shape. An equally important reason is that it is one of the most used clustering methods.

EXPERIMENTAL STUDY

In an experimental study, the methods were compared on the three data sets of two-dimensional data. The experimental data set contains three clusters arranged in shape. For demonstration we present experiments on only one data set. This data set has been selected for implementation because of the presentation capabilities of clustering methods to deal with the cluster complex shape such as used in a spiral. In the results for each method, there is a table containing the column number of the cluster, which indicates the number of clusters where appropriate. The column number of objects in a cluster that indicates how many objects are assigned to a cluster, the column contains the percentage distribution of the percentage of the size of the cluster to the total number of objects, see Tab.2.

Table 2: Data sets represent clusters arranged in a two-dimensional spiral

Number of objects	529
Clusters	3
Dimension	2
Noise	0%
The he shape of clusters	spiral

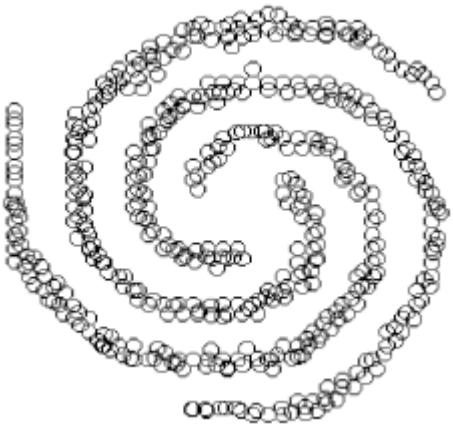

Figure 6: The shape of used spiral

The result of the using a method K-Means when the number of clusters k = 3, see Tab.3 and Fig.7.

Table 3: Results for K-Means method

Cluster	Number of objects in cluster	Percentage distribution
0	170	32%
1	171	32%
2	188	36%

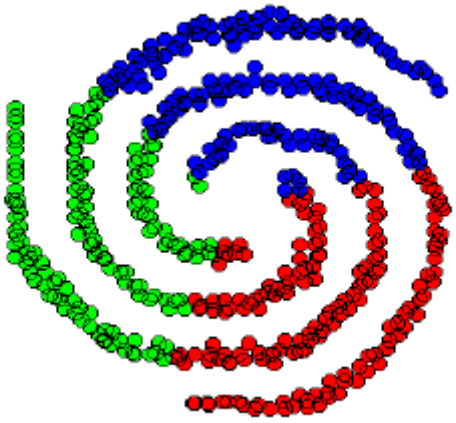

Figure 7: Results for K-Means methods

The result of the using the DBSCAN method when was following set: MinPts=6 and Ɛ=20, see Tab.4 and Fig. 8.

Table 4: Results for DBSCAN method

Cluster	Number of objects in cluster	Percentage distribution
0	196	37%
1	168	32%
2	165	31%

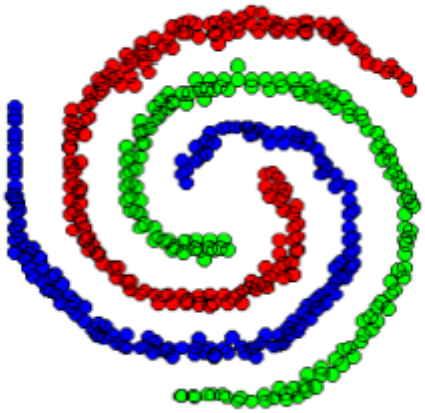

Figure 8: Results for DBSCAN methods

The result of using the SOM method when the output layer was 8x8 neurons and α=1.0. SOM method using the size of the output layer 8x8 where was formed 62 clumps in Figure X is visible placement of neurons in the area of clusters with respect to their shape. A neuron is indicated by a black circle with the numerical designation of its position in the output layer, see Tab.5 and Fig. 9.

Table 5: The first column represents neuron, the second number of objects in cluster and the third percentage distribution.

$n_{0,0}$	8	2%	$n_{2,0}$	7	1%	$n_{4,0}$	8	2%	$n_{6,0}$	7	1%
$n_{0,1}$	5	1%	$n_{2,1}$	10	2%	$n_{4,1}$	7	1%	$n_{6,1}$	4	1%
$n_{0,2}$	6	1%	$n_{2,2}$	12	2%	$n_{4,2}$	19	4%	$n_{6,2}$	8	2%
$n_{0,3}$	8	2%	$n_{2,3}$	14	3%	$n_{4,3}$	5	1%	$n_{6,3}$	7	1%
$n_{0,4}$	10	2%	$n_{2,4}$	10	2%	$n_{4,4}$	8	2%	$n_{6,4}$	6	1%
$n_{0,5}$	10	2%	$n_{2,5}$	7	1%	$n_{4,5}$	7	1%	$n_{6,5}$	10	2%
$n_{0,6}$	8	2%	$n_{2,6}$	8	2%	$n_{4,6}$	0	0%	$n_{6,6}$	8	2%
$n_{0,7}$	7	1%	$n_{2,7}$	10	2%	$n_{4,7}$	12	2%	$n_{6,7}$	10	2%
$n_{1,0}$	7	1%	$n_{3,0}$	9	2%	$n_{5,0}$	8	2%	$n_{7,0}$	4	1%
$n_{1,1}$	10	2%	$n_{3,1}$	10	2%	$n_{5,1}$	9	2%	$n_{7,1}$	7	1%
$n_{1,2}$	9	2%	$n_{3,2}$	14	3%	$n_{5,2}$	7	1%	$n_{7,2}$	10	2%
$n_{1,3}$	10	2%	$n_{3,3}$	0	0%	$n_{5,3}$	9	2%	$n_{7,3}$	10	2%
$n_{1,4}$	8	2%	$n_{3,4}$	9	2%	$n_{5,4}$	7	1%	$n_{7,4}$	8	2%
$n_{1,5}$	8	2%	$n_{3,5}$	7	1%	$n_{5,5}$	3	1%	$n_{7,5}$	8	2%
$n_{1,6}$	9	2%	$n_{3,6}$	12	2%	$n_{5,6}$	8	2%	$n_{7,6}$	9	2%
$n_{1,7}$	10	2%	$n_{3,7}$	7	1%	$n_{5,7}$	8	2%	$n_{7,7}$	9	2%

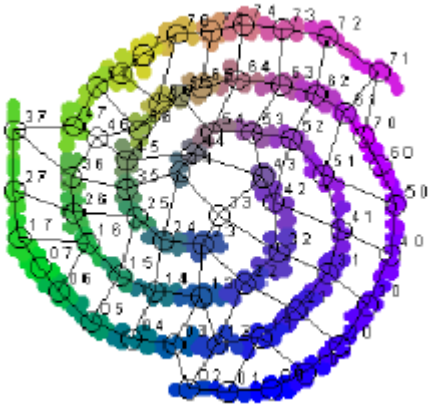

Figure 9: Results for SOM method

The data set in spiral shape is a clearly evident inability of K-Means method to take into account the shape of cluster. The DBSCAN method has solved this problem very well, discovered all the clusters taking into account their shape. The SOM method in the grid smaller than 8x8 could not take into account the shape of a spiral occurred linking individual shoulder, but Figure 9 shows a 8x8 grid settings distinguish 62 clusters, the largest had 19 objects, the method create a higher number of smaller clusters (Sefar, 2013).

CONCLUSION

Our experiments have shown that the best method for two-dimensional data clustering is DBSCAN. The method was able to distinguish all clusters correctly, its drawback lies in the possibility of a more difficult setting of initial parameters, which do not provide the desired result immediately. This quality of the BDSCAN method can be a problem when we do not have information on the number of clusters in the data set. The SOM method suffers from a similar drawback as DBSCAN. It is necessary to adequately set the number of neurons in the output layer. A low number can result in grid over twisting or in representing more objects by one neuron, which results in failing to distinguish the shape of the cluster. The k-means method, as presumed, wasn't able to distinguish clusters of arbitrary shape and deal with remote objects. Using the k-means method seems to be favorable if we do not need to take into consideration the cluster shape and we have information on their number, or we require their exact number.

ACKNOWLEDGEMENT

The research described here has been financially supported by University of Ostrava grant SGS/PřF/2014 and by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

REFERENCES

MacQueen, J. B. 1967, Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297

Sefar, S. 2013, Modern clustering methods, Bachelor thesis, University of Ostrava.

Moore, A. 2005, K-means and Hierarchical Clustering - Tutorial Slides http://www-2.cs.cmu.edu/~awm/tutorials/ kmeans. html

Cabanes, G. & Bennani, Y. 2007. A simultaneous two-level clustering algorithm for automatic model selection. In *Proceedings of the International Conference on MachineLearning and Applications* (ICMLA'07) Cincinnati, Ohio, USA.

Kohonen, T. 2001. *Self-Organizing Maps* . Berlin: Springer-Verlag.

Achtert, E., Böhm, C., Kriegel, H. P., Kröger, P., Müller-Gorman, I., Zimek, A. 2007. Detection and Visualization of Subspace Cluster Hierarchies. *LNCS: Advances in Databases: Concepts, Systems and Applications*. Lecture Notes in Computer Science 4443: 152–163.

Chakraborty, S., Nagwani, N.K., Dey, L., 2011, Performance Comparison of Inceremental K-means and DBSCAN algorithms. *International Journal of Computers Applications* (097-8887), Volume 27 – No.11, August.

EVA VOLNA is an associate professor at the University of Ostrava. Her interests include artificial intelligence, artificial neural networks, evolutionary algorithms, and cognitive science. She is an author of more than 50 papers in technical journals and conference proceedings.

MARTIN KOTYRBA His interests include artificial intelligence, formal logic, soft computing methods and fractals. He is an author of more than 30 papers in conference proceedings.

ZUZANA KOMINKOVA OPLATKOVA is an associate professor at Tomas Bata University in Zlin. Her research interests include artificial intelligence, soft computing, evolutionary techniques, symbolic regression, neural networks. She is an author of around 100 papers in journals, book chapters and conference proceedings. Her e-mail address is: oplatkova@fai.utb.cz