Clustering qualitative data based on the flow networks

Arkadiusz Lewicki[1,*], *Krzysztof Pancerz*[2]

[1]Department of Applied Informatics, University of Information Technology and Management, Sucharskiego str. 2, 35-225 Rzeszow, Poland
[2]Department of Computer Science, University of Rzeszow, Rejtana str. 16c, 35-225 Rzeszow, Poland

Abstract. The paper presents research results referring to the use of flow networks and ant colony algorithm in the problem of generating decision rules for the cluster analysis process. The experiments showed that proposed approach may prove particularly important, when we are dealing with data sets represented by categorical variables associated with the same number of objects for each variance. There are many cases when we have no knowledge about the allocation group of individual data received and in addition defining any metric to measure the distance between observations, does not give any satisfactory results. Meanwhile, the selection of features and the choice of the performance metric is the basic condition for the use of most known classifiers. The article presents a new approach to solve this problem and obtain satisfactory results. It is based on mapping the set of analyzed data into the flow network, calculating the maximum flow and determining the validity of nodes in the network to use the Ant Colony Algorithm to structuring information and determine significant relationships between data.

1 Introduction

In the process of obtaining important information from available data sets, we can apply supervised learning [1] or unsupervised learning [2] approaches. In the first case, we have to use the training data. We have a set of samples in which the desired output signals (labels) are known. If we use discrete class labels, it is a classification. Those discrete class labels are unordered values that determine the affiliation of specific objects to designated groups. That process allows prediction of class labels in new occurrences based on previous observations. In this case, we know the right answer before training the predictive model. However, when we have the unidentified data or datasets about an unknown structure and we have no knowledge about the group assignment, then it is much more difficult to organize the available sets of information into significant subsets. In that case, we can use a multidimensional statistical analysis to extract homogeneous subsets of the studied objects. This process is called cluster analysis or clustering.

The cluster analysis requires: selection of objects and variables, selection of a normalization formula for variable values, selection of distance measure, selection of a

* Corresponding author: alewicki@wsiz.rzeszow.pl

classification method, the determination of the number of classes, evaluation of the classification results and their interpretation. That approach allows to detect whether the received aggregations indicate some regularity, reduction of large data sets, and also to carry out further multidimensional analysis. The available solutions for the problem of cluster analysis allow the use of partitioning algorithms, hierarchical algorithms or density-based algorithms.

The partitioning algorithms concern the attempt to find the optimal division of a set of objects, in such a way that the objects within the cluster are more similar to each other than to objects from other clusters. This method requires defining a clustering criterion. This type of algorithm includes the k-means algorithm [3, 4] and the k-medoids algorithm [5, 6]. The first of these divides n objects into k-clusters, in which each object belongs to the cluster with the nearest average. The target function used in k-means algorithm is usually a squared error. The main disadvantage of this algorithm is that we have to determine the number of clusters in advance. Unfortunately, there is no universal method to find the optimal number of clusters. Another disadvantage is that the algorithm fails for categorical data. Also, if there are two very overlapping data, the k-funds will not be able to determine that there are two clusters. The k-medoids algorithm, in contrast to the k-means algorithm, assumes the median as the center of the cluster. However, the resulting clustering depends on the units of measurement.

The hierarchical algorithms [7, 8] rely on a hierarchical attempt to discover the structure of the set by decomposing the cluster. These algorithms build a cluster tree (a dendrogram) showing relationships between selected elements. The hierarchical clustering algorithms are monotonic, because they either increase or decrease. There are two types of hierarchical clustering: bottom up (agglomerative method) or top down (divisive method). In top-down clustering method we assign all objects to a single cluster and next partition this cluster to two least similar clusters. This step is repeated until each object is in a separate cluster or interrupt condition will be fulfilled. In bottom-up clustering method we assign each object to its own cluster. Then, these clusters are successively combined until all are in a single, hierarchical group. Clusters are combined according to a specific measure, e.g. the distance between their centers.

The last group of cluster analysis algorithms are the density-based algorithms, which divide sets of object using the probabilistic model for base clusters. This algorithm group includes, for example, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [9] and OPTICS (Ordering Points To Identify Clustering Structure) algorithm [10]. These algorithms connect objects to form clusters based on regions with high data density and separated from other clusters by regions with low data content. This means that until the number of objects around the cluster is large (with a given parameter), the cluster will grow. However, the quality of the DBSCAN and OPTICS algorithm depends to parameters, such: neighborhood radius and the number of minimum points required to form a cluster. The parameter selection process is tricky.

The characteristics of cluster analysis methods show that we need to define a metric for the space representing objects [11]. The metric d (1) can be a mapping of the X x X Cartesian product to a set of non-negative numbers R.

$$d: X\ x\ X \rightarrow [0, +\infty) \tag{1}$$

The defined assumptions for all vector pairs $x^a, x^b \in X(a,b=1,2,\ldots)$ are related to rules 2, 3 and 4:

$$d(x^a, x^b) = 0 \leftrightarrow x^a \equiv x^b \tag{2}$$

$$d(x^a, x^b) = d(x^b, x^a) \tag{3}$$

$$d(x^a, x^b) \leq d(x^a, x^c) + d(x^c, x^b) \quad (4)$$

One of the most commonly used metrics in cluster analysis is the Euclidean metric. However, a generalized measure of the distance between the points of the Euclidean space is the Minkowski metric (5):

$$d(x, y) = (\sum_{i=1}^{N} |x_i - y_i|^{\alpha})^{1/\alpha} \quad (5)$$

If we use α= 2, then we will get the Euclidean metric. For α= 2 we will get the get a Manhattan metric. Other popular distance measures for measurable features are: the Mahalanobis measure (6) and Canberry (7).

$$d_{Ma}(x, y) = \sum_{ij}(x_i - y_i) \sum (x_i - y_i)^{-1} \quad (6)$$

$$d_{Ca}(x, y) = \sum_{i=1}^{N} \frac{|x_i - y_i|}{|x_i + y_i|} \quad (7)$$

where x, y are vectors, $\sum$ is the covariance matrix of the x and y, and N is the number of vector features.

The cluster analysis algorithms often use also Hamming metric and the Jaccard similarity coefficient. Hamming metric (8) is used for data vectors that have attributes represented by binary values. The Jaccard similarity coefficient (9) measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$d_H(x, y) = \sum_{i=1}^{N} \overline{(x_\iota \oplus y_\iota)} \quad (8)$$

where $\oplus$ means the sum of modulo two.

$$J(x, y) = \frac{|x \cap y|}{|x \cup y|} \quad (9)$$

The values assumed by the Jaccard coefficient are contained in a subset of a set of real numbers <0.1>. If the Jaccard coefficient assumes values close to zero, then the sets are different from each other. If the Jaccard coefficient assumes values close to 1, then the sets are similar to each other. This is particularly important for objects represented by vectors of binary data. For example, we can consider two objects (O_1 and O_2) represented by vectors of binary data, where there are 4 dichotomic variables (Table 1).

Table 1. The example objects represented by vectors of binary variables

Items of variables	**x1**	**x2**	**x3**	**x4**
O_1	1	0	1	0
O_2	1	1	0	1

In order to measure similarity of asymmetric categorical data, we can calculate: number of attributes where O_1 is 1 and O_2 is 0 – formula (10), number of attributes where O_1 and O_2 are 1 - formula (11), number of attributes where O_1 and O_2 are 0 - formula (12), number of attributes where O_1 is 0 and O_2 is 1 – formula (13).

$$D = \sum_i x_{O_1}(1 - x_{O_2}) \quad (10)$$

$$E = \sum_i x_{O_1} x_{O_2} \tag{11}$$

$$F = \sum_i (1 - x_{O_1})(1 - x_{O_2}) \tag{12}$$

$$G = \sum_i (1 - x_{O_1})\, x_{O_2} \tag{13}$$

where x_{O1} is value of observation O_1 in the variable i-th, and x_{O2} is value of observation O_2 in the variable i-th.
The Jaccard coefficient for similarity of asymmetric binary attributes of object O_1 and object O_2 is defined by formula (14):

$$J(O_1, O_2) = \frac{E}{E+G+D} \tag{14}$$

Which algorithm and metric are therefore the most effective for the problem of cluster analysis? The research and analysis conducted by the authors this article have shown that there is no such classifier that would work effectively (or even satisfactorily) in all possible situations. There are also such cases in the field of problem analysis of categorical data sets that without a priori knowledge, even the density-based algorithms will return erroneous results. The solution presented in the further part of the article deals with this type of problem.

2 Problem definition

The categorical data are variables representing one of the finite number of categories. Examples of such data are ID numbers, nationality, purchased product, interests, etc. We can use the Jaccard coefficient (14) as a measure of dissimilarity (or similarity) for many collections of categorical variables. This coefficient is defined as the ratio of the number of variables for which the objects differ to the total number of all variables describing the objects. However, what if this measure for many case objects returns the same values, but these objects should be assigned to different groups?

Let's consider a set of only 24 objects represented by 4 categorical variables. For example, contact lenses dataset. Each of the 24 objects of this set (Table 2) is described by the vector of the following attributes: age {young, pre-presbyopic, presbyopic}, spectacle prescription {myope, hypermetrope}, astigmatism {no, yes}, tear-prod-rate {reduced, normal}.

Table 2. The characteristics of the variable objects from the lenses dataset

Object/ attributes	Age	Spectacle prescription	Astigmatism	Tear-prod rate
O_1	young	myope	no	reduced
O_2	young	myope	no	normal
O_3	young	myope	yes	reduced
O_4	young	hypermetrope	no	reduced
O_5	young	hypermetrope	no	normal
…	…	…	…	…
O_{10}	pre-presbyopic	myope	no	normal
…	…	…	…	…
O_{24}	presbyopic	hypermetrope	yes	normal

The set presented in table 2 is available from the UCI Machine Learning Repository. this object collection is very interesting, because contains all possible combinations of values of categorical data. Therefore, we have the same number of young (8) patients, pre-presbyopic (8) and presbyopic (8). The same number of patients having myope (12) and hypermetrope (12). The same number of patients who have astigmatism (12) and don't have astigmatism (12). The same number of patients who have a reduced tear production rate (12) and the same number who have normal (12). We could apply the algorithm of rule induction or one of the density-based algorithms. However, we need to know the number of clusters for the correct division of objects and the decision attribute. We know from the repository that each of the 24 objects can belong to one of three groups: hard contact lenses (class1), soft contact lenses (class2) or none contact lenses (class3). Then we could point out the classification rules. We could use the proposed by Cendrowska an algorithm for inducing modular rules. These rules are:

1) If astigmatism = no
   and tear-prod-rate = normal
   and spectacle-prescrip = hypermetrope then soft contact lenses
2) If astigmatism = no
   and tear-prod-rate = normal
   and age = young then soft contact lenses
3) If age = pre-presbyopic and astigmatism = no and tear-prod-rate = normal then soft
4) If astigmatism = yes
   and tear-prod-rate = normal
   and spectacle-prescrip = myope then hard contact lenses
5) If age = young
   and astigmatism = yes
   and tear-prod-rate = normal then hard contact lenses
6) If tear-prod-rate = reduced then none
7) If age = presbyopic and tear-prod-rate = normal
   and spectacle-prescrip = myope
   and astigmatism = no then none contact lenses
8) If spectacle-prescrip = hypermetrope
   and astigmatism = yes
   and age = pre-presbyopic then none contact lenses
9) If age = presbyopic
   and spectacle-prescrip = hypermetrope
   and astigmatism = yes then none contact lenses

However, let's consider the case, that we have this set of data without the knowledge about the decision attribute and without the knowledge about the correct number of classification groups. The experiments carried out by the authors showed that both the k-means algorithm and k-medoids algorithm, proved to be ineffective. The density-based algorithms also required knowledge on the expected parameters. The cluster analysis using similarity metrics for such a dataset is difficult, because even Jaccard's coefficient does not give correct results. If we consider an objects: O_3 [young, myope, yes, reduced] and object O_8 [young, hypermetrope, yes, reduced], and next O_8 [young, hypermetrope, yes, reduced] and object O_{24} [presbyopic, hypermetrope, yes, normal], we will see that they differ in one argument (first or last) and they belong to 3 different classes. However, the difference and the rule is the same in case O_7 [young, hypermetrope, yes, normal] and object O_8 [young, hypermetrope, yes, reduced], and next O_{16} [pre-presbyopic, hypermetrope, yes, reduced] and object O_{24} [presbyopic, hypermetrope, no, normal], but the objects belong to the same

class. Therefore, we propose using the flow network and Ant Colony Classification Algorithm, as a possibility to solve the described above problem.

3 The proposed solution, experiments and results

We do not know the criterion of division for the objects of described above dataset. Let's assume that we also do not know how many object groups should be. If we considering this type of problem, we can map the dataset of qualitative data as a bipartite graph, where vertices will represent possible values of categorical variables. The edges between the vertices in the bipartite graph should be related to the number of occurrences of the categorical data values (represented by given nodes). The bipartite graph may represent a flow network, but at the beginning with zero flows. This is illustrated in Figure 1.

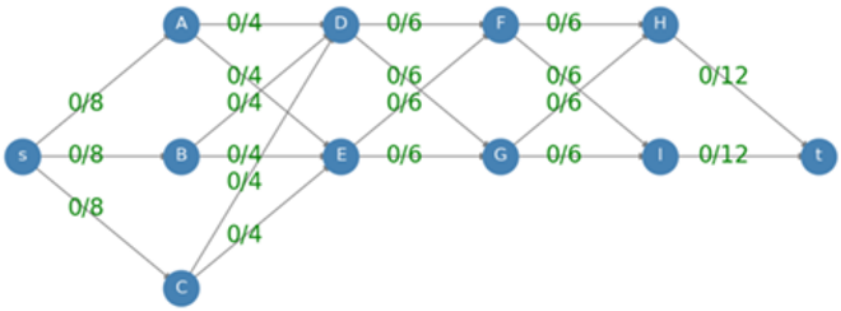

Fig. 1. The mapping of lenses dataset structure as a bipartite graph.

The nodes A, B and C represent the values of the 'age' attribute - {young, pre-presbyopic, presbyopic}, i.e. A corresponds to young, B corresponds to pre-presbyopic and C corresponds to presbyopic. The nodes D and E represent the values of the 'spectacle prescription' attribute - {myope, hypermetrope}. The nodes F and G represent the values of the 'astigmatism' attribute - {no, yes}. The nodes H and J represent the values of the ' tear-prod-rate' attribute - {reduced, normal}, s in the flow network represents the source, and t - outlet. If we next apply the Ford-Fulkerson algorithm or its modified version, we will begin the process of saturating the edges of the transition from the source to the outlet. This is shown in Figures 2 and Figure 3.

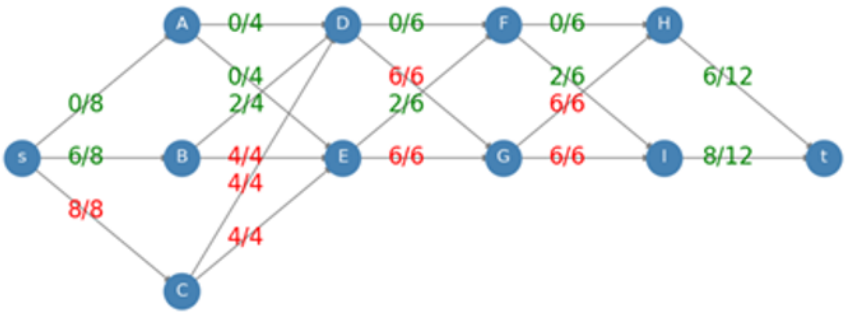

Fig. 2. The process of saturating the edges for a flow network (representing the lenses dataset).

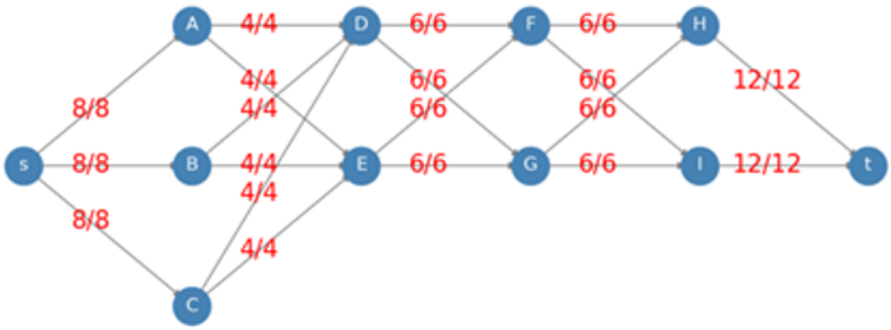

Fig. 3. The maximum flow in the flow network.

The maximum flow in our network is 24. However, the process of saturating individual transitions in the our flow network is important here. It is associated with transitions: s – C-E – G – I – t, s – D – E – G – I – t, s – D – E – G – H – t, s – B – E – G – H – t, s – B – E – F – I – t, s – B – D – G – I – t, s – B – D – F – I – t, s – A – E – F – I – t, s – A – D – F – H – t. The order in which the attributes are mapped is irrelevant because the weights of the A, B and C attributes are the same. The same situation is for the attributes D and E, F and G and H and I. The indicated paths are associated with the set of objects: {O_{24}, O_{20}, O_{19} , O_{15}, O_{14}, O_{12}, O_{10}, O_6, O_1}. This subset will be indicated as a decision collection.

Let's denote by $\Phi(N)$ the maximum flow associated with the node (where N represents A or B or C, etc.). Let's denote by $max\Phi(G)$ the maximum flow network. It will allow us to define the validity of a particular node N as (15):

$$valid(N) = \frac{\Phi(N)}{max\Phi(G)} \tag{15}$$

where vaild(N) is the validity of the rule attribute represented by N node. We have: valid(A) =0.33, valid(B) =0.33, valid(C) =0.33, valid(D) =0.5, valid(E) =0.5, valid(F) =0.5, valid(G) =0.5, valid(H) =0.5 and valid(I) =0.5. So, we can see that our vector consists of the attributes [A | B | C, D | E, F | G, H | I], but the most important for the decision process are pairs {H | I, F | G}, {H | I, D | E} and {F| G, D | E}. We will have three different groups of objects for each pair considered in the set of objects: {O_{24}, O_{20}, O_{19} , O_{15}, O_{14}, O_{12}, O_{10}, O_6, O_1}. For the pair of attributes {H | I, F | G} it will be: class1=[G-I], class2 = [G-H] and class3=[F-H]. In addition, the saturation of the last possible transition is associated only with one pair of values [F-H].

Let's examine how the proposed approach will work for another case and other types of objects, for example for the problem of grouping quantitative data. For this purpose, let's choose the most popular dataset – iris dataset. This collection has 150 objects with 4 elements (sepal length, sepal width, petal length, petal width). The three expected classes are: Iris Setosa, Iris Versicolor and Iris Virginica. The sample dataset of Iris Setosa is shown in Table 3, Iris Versicolor - in Table 4 and Iris Virginica - in Table 5. However, let's assume also for this set, that we do not know how to group objects and what is the expected number of classes. All attributes of the iris dataset have real values. Therefore, they should be rounded to integers, according to the laws of mathematics. It will be a set of vertices of the bipartite graph. This is illustrated in Figure 4.

Table 3. The sample dataset of Iris Setosa

Sepal length	Sepal width	Petal length	Petal width
5.0	3.5	1.6	0.5
5.1	3.5	1.5	0.5
5.4	3.9	1.7	0.4
5.7	4.4	1.3	0.4

Table 3. The sample dataset of Iris Versicolor

Sepal length	Sepal width	Petal length	Petal width
7.0	3.2	4.7	1.4
6.4	3.2	4.5	1.5
5.5	2.3	4.9	1.5
6.5	2.8	4.0	1.3

Table 3. The sample dataset of Iris Virginica

Sepal length	Sepal width	Petal length	Petal width
6.3	3.3	6.0	2.5
5.8	2,7	5.1	2.9
7.1	3.0	5.9	2.1
6.3	2.9	5.6	1.8

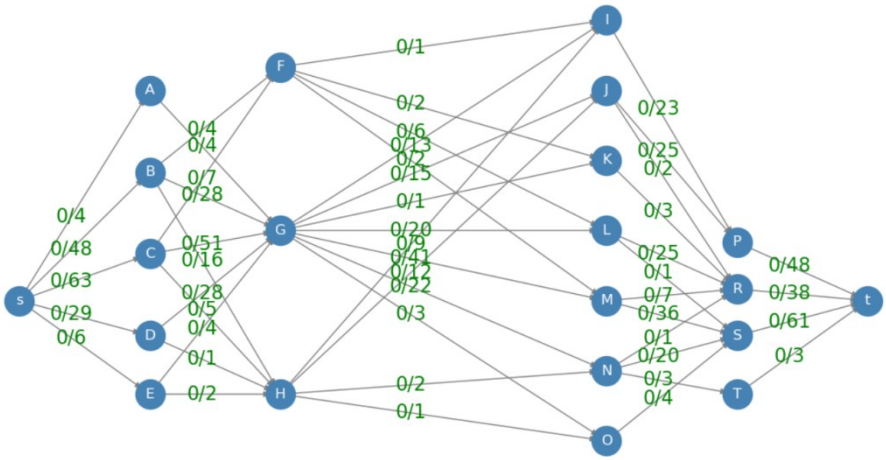

Fig. 4. The labelled iris dataset for flow network.

For the examined set of objects (where A represents 4, B–5, C6, D–7, E–8, F-1, G-1, H-4, I-1, J-2, K-3, L-4, M5, N-6, O-7, P-0, R-1, S-2 and T-3), we have: valid(T)=0.02,

valid(S)=0.406, valid(R)=0.25, valid(P)=0.32, etc. As we can see, the value of valid(T) is very small and differs significantly from other output nodes. Besides, valid(P) + valid(R) + valid(S)= 0.976. Therefore, we will consider a maximum of only three classes. The edge saturation process in this flow network returned a set of decision, in the form of nodes with the highest decision-making importance. These pairs are: {I, P}, {J, P}, {N, S}, {M, S}, {L, R}. Earlier attributes (A, B, C, D, etc.) are less important due to the significant weight P, R, S, I, M, N nodes. We have indicated decision rules and the knowledge about the number of classes. Then we can use any classification algorithm that uses the Euclidean distance (Figure 5). The carried out experiments showed that the proposed method returns the result which is 84.6% of the expected accuracy.

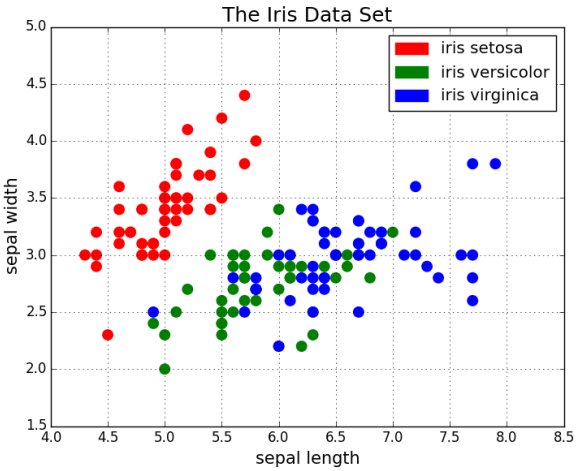

Fig. 5. The classified iris dataset objects

We have checked the proposed method of returning the decision-making collection also on the following datasets: car evaluation dataset, nursery dataset and dermatology dataset. We have achieved a result confirming the correctness of selecting the number of groups and a set of decision attributes, for each of these datasets.

Thanks to the construction of a flow network for categorical data sets such as the lenses dataset, car evaluation dataset, nursery dataset and dermatology dataset, we have received data on the number of classes. Moreover, we have received the data set, from which a specific number of reference groups should be created (according to the validity of attributes). We next used the Ant Colony algorithm as a classifier. The ant based clustering algorithm used in our experiments is mainly based on the versions proposed by Lumer and Faieta [12]. The difference is the calculation of similarity to the reference objects. It is calculated based on the formula (16):

$$d(x,y) = \frac{k-w}{k} \tag{16}$$

where k is the total number of variables but not related to the decision attribute pairs, and w is the number of variables whose value is the same for both objects (for attributes other than decision pairs),

The proposed approach allowed to obtain for the lenses dataset an accuracy of 94.7%, 87.3% - for car evaluation dataset, 87.9 – for nursery dataset and 95.2% - for dermatology dataset. These results are fully acceptable.

3 Conclusions

In the paper, we have examined the problem of clustering data categorical, for which there was no a priori knowledge about the expected number of groups. The decisive attribute was also unknown. The proposed use of flow networks and method is an option to solve this problem. The carried out experiments allowed to achieve satisfactory results. However, proposed approach requires further research for other data sets.

References

1. T. G. Dietterich, Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, (1998).
2. N. Grira, M. Crucianu, N. Boujemaa, Unsupervised and semi-supervised clustering: a brief survey, A review of machine learning techniques for processing multimedia content, (2004).
3. A. K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters, (2010).
4. P. S. Bradley, K. P. Bennett, A. Demiriz, Constrained K-Means Clustering, Machine Learning, (2000).
5. S. K. Popad, Review and Comparative Study of Clustering Techniques, International Journal of Computer Science and Information Technologies, (2014).
6. T. Velmurugan and T. Santhanam, Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points, Journal of Computer Science (2010).
7. Y. Zhao, G. Karypis, U. Fayyad, Hierarchical Clustering Algorithms for Document Datasets, Data Mining and Knowledge Discovery, (2005).
8. R. Lior, M. Oded, Clustering methods, Data mining and knowledge discovery handbook. Springer US, (2005).
9. R. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data, (2015).
10. M. Ankerst, M. M. Breunig, H. P. Kriegel, J. Sander, OPTICS: Ordering Points To Identify the Clustering Structure. ACM SIGMOD international conference on Management of data. ACM Press, (1999).
11. P. M. B. Vitanyi, Information Distance in Multiples, IEEE Transactions on Information Theory, (2011).
12. E. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants, Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, MIT Press, Cambridge, (1994).