UNSUPERVISED CLASSIFICATION AND PARTICLE SWARM OPTIMIZATION

Adam Truszkowski, Magdalena Topczewska

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: This article considers three algorithms of unsupervised classification - *K-means*, *Gbest* and the *Hybrid* method, the last two have been proposed in [14]. All three algorithms belong to the class of non-hierarchical methods. At first, the initial split of objects into known in advance number of classes is performed. If it is necessary, some objects are then moved into other clusters to achieve better split - between cluster variation should be much larger than within cluster variation. The first algorithm described in this paper (*K-means*) is well-known classical method. The second one (*Gbest*) is based on the particle swarm intelligence idea. While the third is a hybrid of two mentioned algorithms. Several indices assessing the quality of obtained clusters are calculated.

Keywords: unsupervised classification, clustering, particle swarm optimization

1. Introduction

The aim of unsupervised classification or cluster analysis is to search the data for a structure of natural groupings to understand the complex nature of multivariate relationships. This group of methods are exploratory techniques and can be widely used in many fields, for example in medicine to search for sets of symptoms or treatments, in marketing to segment of target groups, etc. Thus it can provide an informal means for suggesting hypotheses concerning relationships. It can be also helpful to assess dimensionality of data and to identify outliers [5].

The process of grouping is performed on the basis of similarities or dissimilarities (distances) of objects and this is the only assumption of this group of methods. Of course, to obtain good results some kind of basic exploration should be made to decide how to measure the association between the objects. Due to the fact that we usually have to deal with large data sets, we can rarely examine all group combinations. So we can use some of the wide variety of algorithms which emerged to find

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 9, pp. 119-132, 2012.

reasonable clusters in data.
Each algorithm solving clustering problem belongs to one of two types – hierarchical or non-hierarchical. Hierarchical methods create a tree structure called dendrogram by splitting or merging recursively existing groups of objects. In the non-hierarchical methods objects are divided into known in advance number of clusters, and this division is then corrected by moving some objects between clusters to obtain better split. Among others *K-means* is the most popular of this kind. The applications of this algorithm cover many areas and range from market segmentation [7], portfolio analysis [12], to image retrieval [8] or academic performance [11]. The algorithms inspired by nature can be alternative proposal. They have also many applications, for instance image classification [10], robotic mapping [3], document clustering [1], etc. Comparing with other artificial intelligence algorithms, like genetic or evolutionary algorithms, *swarm intelligence* (*SI*) is relatively new group of methods, but regard to its increasing popularity it seems promising. *Particle swarm optimization* (*PSO*) belongs to the *SI* group of methods. It is inspired by social behavior of bird flocking and fish schooling [6]. Each element (bird, fish) called a *particle* respects few primary rules – moves toward the best position of the swarm leader and shares information with its neighbours.
This article considers thee non-herarchical algorithms of clustering: *K-means*, *Gbest* (based on particle swarm intelligence idea) and the *Hybrid* method. Several indices of obtained quality of splits, like adjusted Rand index, mistakes matrix, inter cluster, intra cluster and validity quantities, are calculated.

2. Methods

The first algorithm described in this paper (*K-means*) is a well-known classical method. The second one – *Gbest* – is based on the particle swarm intelligence idea. While the third one is a hybrid of two mentioned algorithms. In general, at first, the initial split of objects into known in advance number of classes is performed. If it is necessary, some objects are then moved into other clusters to achieve better split and improve results - between cluster variation should be much more larger than within cluster variation.
The crucial issue is how to measure the distance between objects. In the pattern recognition systems, for example during examination of profiles or patterns, it is much better to use correlation measures than distance ones to avoid connection of objects with different profiles. To calculate the similarity between elements the most commonly used is Euclidean distance. We can also apply Manhattan, Jaccard or Gower distances depending on the considered problem.

In this paper only the Euclidean distance is applied due to the fact that it is a measure calculated as a distance between two points joined with a straight line. If there is no additional information about the data and the objects are points in $\mathbb{R}^{N_d}$, where $\mathbb{R}^{N_d}$ is a real N_d-dimensional vector space, this distance is a natural way to calculate the similarity between objects.

Given a set of N_o objects $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{N_o}\}$, each object is described by N_d values of attributes. The notation used in the article to all three methods is as follows:
N_o – number of vectors (objects in the data set),
N_d – number of dimensions (attributes describing each object),
N_c – number of clusters,
n_j – number of objects in j-th cluster,
$\mathbf{m}_j$ – the vector describing j-th cluster centroid,
$\mathbf{C}_j$ – the subset of objects belonging to the j-th cluster.

2.1 *K-Means* algorithm

The *K-means* algorithm was described in [9] and assigns each object to the cluster having the nearest centroid. Instead of K in this article the N_C denotes the number of clusters. The steps of this method are presented in the listing 1. Three steps are performed in this algorithm. In the first step the initial partition of objects into predefined N_C clusters is made. In the second one, the objects are allocated into clusters - the object is located to the cluster according to the rule of the nearest centroid. The distance between $\mathbf{z}_p$ object and $\mathbf{m}_j$ centroid is calculated as Euclidean distance

$$d(\mathbf{z}_p, \mathbf{m}_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \tag{1}$$

Then the mean values of clusters (centroids) are recalculated

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\forall \mathbf{z}_p \in \mathbf{C}_j} \mathbf{z}_p \tag{2}$$

and updated. The step two is repeated, because it may turned out that after recalculation of centroids some objects should be reallocated. If there are no more reassignments, the algorithm is stopped. The stop criterion can also consider the number of iterations or timeout has been exceeded, etc.

Rather than starting with partition of data into initial clusters, the N_C initial centroids can be specified and then all objects are divided into groups.

Algorithm 1 K-Means

1: Partition the vectors into N_C initial clusters
2: **repeat**
3: For each data vector, assign it to the cluster whose centroid is nearest (usually computed distance is Euclidean distance with either standardized or unstandardized vectors (1)
4: Recalculate the cluster centroids using (2)
5: **until** no more reassignments take place

The final result depends largely on the initial means. Poor selection of centroids might cause poor quality of obtained clusters.

2.2 *Gbest* algorithm

In this and the next subsection two methods based on particle swarm optimization (PSO) are introduced. Particle swarm optimization is inspired by social behavior of bird flocking and fish schooling [6]. Each element is called a *particle* and represents a potential solution of optimization problem. Therefore, the whole swarm is a set of potential solutions. During the optimization process each particle changes its position, remembering its best position and best positions of its neighbors. It moves toward the best position of the swarm leader and shares information with its neighbors.

The algorithm based on PSO idea is the *Gbest*, in which i-th particle $\mathbf{x}_i = (\mathbf{m}_{i1}, \mathbf{m}_{i2}, \ldots, \mathbf{m}_{iN_c})$ is initialized as a vector of randomly proposed centroids for N_c clusters. Therefore the swarm represents candidates for clusters obtained from data. The number of particles in a swarm is chosen arbitrarily by the user. Every particle stores three types of information: the current position ($\mathbf{x}_i$), the current velocity ($\mathbf{v}_i$) and its best position ($\mathbf{P}_i$) In the next step, for each particle and each object the distance between the object and centroids is calculated and then the element is allocated into the cluster according to the rule of the nearest centroid. Besides, the value of the fitness function is evaluated. The fitness function calculated as the quantization error has the form

$$J_e = \frac{\sum_{j=1}^{N_c} \left(\sum_{\forall \mathbf{z}_p \in C_{ij}} d(\mathbf{z}_p, \mathbf{m}_j)/|C_{ij}| \right)}{N_C} , \tag{3}$$

where $|C_{ij}|$ denotes the number of objects belonging to ij-th cluster.
Then the best local ($\mathbf{P}_i$) and global particle ($\mathbf{G}_i$) is found and actualized and finally, the centroids in every particle are recalculated using equation for actualization of the velocity of the particle

$$v_i(t+1) = \omega v_i(t) + c_1\phi_1(t)(\mathbf{P}_i(t) - x_i(t)) + c_2\phi_2(t)(\mathbf{G}(t) - x_i(t)) , \tag{4}$$

and new position of the particle

$$x_i(t+1) = x_i(t) + v_i(t+1)\ , \tag{5}$$

where ω – inertia weight, c_1 and c_2 are acceleration constances, $\phi_1, \phi_2 \sim U(0,1)$, P_i is the best position of the *i*-th particle, G is the best global position.
The standard *Gbest* algorithm is presented in the listing 2.

Algorithm 2 Gbest

```
Initialize each particle to contain N_C randomly selected cluster centroids
for t = 1 to t_max do
    for each particle i do
        for each data vector z_p do
            calculate the Euclidean distance (1) d(z_p, m_ij) to all C_ij cluster centroids
            assign z_p to C_ij cluster such that d(z_p, m_ij) = min_{∀c=1,...,N_C}(z_p, m_ij)
            calculate the fitness using (3)
        end for
        Update the best global and best local positions
        Update the cluster centroids in a particle using (4) and (5)
    end for
end for
```

where:
t_{max} – the maximum number of iterations.

2.3 *Hybrid* algorithm

The *Hybrid* method presented in this subsection is similar to the *Gbest* algorithm. The only difference is a placement of centroids, found by *K-means* method, into a set of particles. The aim is to facilitate the task of searching the solution and the swarm leader with the best position. The hybrid algorithm has four steps:

1. Determine the N_c number of clusters and input the data set.
2. Execute *K-means* method and find the centroids of clusters.
3. Put the result of the previous step as initial values of one particle and for the rest particles of the swarm use values initialized randomly.
4. Execute *Gbest* algorithm with initialized swarm.

2.4 Indication of achieved clusters

To measure the correspondence between partitions of the objects several indices might be applied. In this article we concentrate on the adjusted Rand index, mistakes matrix, inter cluster, intra cluster and validity quantities.
The *Adjusted Rand Index* (ARI) [4] is the modification of the *Rand Index* [13]. The aim of ARI is to prove the disagreement or agreement between two divided groups of objects and classes which they belong to. The maximum value of ARI and RI equals 1 and it means that there is complete agreement between obtained clusters and original classes. Since the RI lies between 0 and 1, its expected value must be greater than or equal 0. The expected value of ARI is 0 and this index has wider range of values. Low values indicate poor agreement between clusters and classes.

3. Results

In this article seven data sets were chosen to experiments. First five sets are artificial problems and can be downloaded from [15].

- *Cl-3* – data set consisted of three classes of objects described by 2 attributes. In each class there are 200 objects. Elements of every class can be described by normal distribution with mean vectors $\mu_1 = (0,0)$, $\mu_2 = (2.5,1)$, $\mu_3 = (-0.5,3)$ and the same covariance matrices $\Sigma_1 = \Sigma_2 = \Sigma_3$.
- *Mag* – data set consisted of two classes of normally distributed objects from two dimensional space with mean vectors: $\mu_1 = (0.1342, 0.0229)$, $\mu_2 = (3.1186, 0.0978)$ and covariance matrices:

$$\Sigma_1 = \begin{bmatrix} 2.2617 & 2.1477 \\ 2.1477 & 2.4903 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 2.3649 & 2.0096 \\ 2.0096 & 2.0425 \end{bmatrix}$$

  100 elements for every class were randomly generated from normal distribution with mentioned parameters.
- *Mag2-out* – *Mag* data increased by one outlier $(-1,4)$.
- *Squares4* – data set consisted of four classes with 100 generated objects in each class. The classification rule for this problem was as follows:

$$class = \begin{cases} 1 \text{ if } (z_1 >= 4 \text{ and } z_1 <= 6 \text{ and } z_2 >= 4 \text{ and } z_2 <= 6) \\ 2 \text{ if } (z_1 >= 7 \text{ and } z_1 <= 9 \text{ and } z_2 >= 4 \text{ and } z_2 <= 6) \\ 3 \text{ if } (z_1 >= 4 \text{ and } z_1 <= 6 \text{ and } z_2 >= 8 \text{ and } z_2 <= 10) \\ 4 \text{ if } (z_1 >= 7 \text{ and } z_1 <= 9 \text{ and } z_2 >= 8 \text{ and } z_2 <= 10) \end{cases}$$

– *Squares2* – objects from *Squares4* data set where original classes 1 and 4 were connected to one class with label 1, remaining two classes created new class with label 2. Thus the obtained location of objects in classes is like in XOR truth table problem.
– *Iris* – The well-known and well-understood problem of three species of irises, where each of 150 flowers is described by four attributes and belongs to one of three classes.
– *Wine* – The well known classification problem with three classes, 178 objects and 13 attributes.

The results presented in the tables below are the mean values for twenty starts of the algorithms.

3.1 Experiment 1

The first experiment concerned selection of particle swarm optimization algorithm parameters to achieve the best numerical values and to use them in second experiment.
At first, the relationship between ϕ_1 and ϕ_2 in (4) was examined. If $\phi_1 < \phi_2$ occurs it means that the centroids of the particle consider more their best local position. In the case when $\phi_1 > \phi_2$ occurs, the centroids consider more the best global position. If there is equality between ϕ_1 and ϕ_2 the centroids are influenced by the best local and global positions to the same extent.
As the example we present selected results for *Cl-3* data set with three separate clusters in the Table 1. The best results were obtained for $\phi_1 = 0.8$ and $\phi_2 = 0.2$. The similar results, when the relationship $\phi_1 > \phi_2$ was observed for all presented data sets.

Table 1. Fitness function results for parameters ϕ_1 and ϕ_2 for *Cl-3* data set

ϕ_1	ϕ_2	*Gbest* method	*Hybrid* method
0.1	0.9	1.0160	0.6867
0.2	0.8	0.9112	0.6866
0.3	0.7	0.9638	0.6866
0.4	0.6	0.7752	0.6866
0.5	0.5	0.7866	0.6866
0.6	0.4	0.7423	0.6866
0.7	0.3	0.7338	0.6866
0.8	0.2	0.7012	0.6865
0.9	0.1	0.7225	0.6866

Next, the relationship between parameters c_1 and c_2 for the same data set was examined. Results are presented in the Table 2. Both parameters are the acceleration constants with which the centroids in a particle move. Too high value may cause the falling out of the swarm territory, while too small may cause very slow relocation of the particles and finally not getting the optimal solution. To assess the best set of c_1 and c_2 values the fitness function of the best particle was used. The best results as the quantization error equaled to the fitness function value was achieved for $c_1 = c_2 = 0.5$ for examined data set.

Table 2. Fitness function results for parameters c_1 and c_2 for *Cl-3* data set

c_1	c_2	*Gbest* method	*Hybrid* method
0.1	0.9	0.8909	0.6866
0.2	0.8	0.7658	0.6861
0.3	0.7	0.9503	0.6867
0.4	0.6	0.7216	0.6849
0.5	0.5	0.7125	0.6834
0.6	0.4	0.7989	0.6855
0.7	0.3	0.8231	0.6866
0.8	0.2	0.7597	0.6865
0.9	0.1	0.9548	0.6867

For the selected data set we examined the influence on the results by different number of particles in the swarm. The *Gbest* and *Hybrid* methods belong to meta-heuristic optimization algorithm and wrong number of particles may be crucial to obtain the optimal solution. In the Table 3 the average time with standard deviation of three executions of two algorithms are shown for 3, 10, 50 and 100 particles. Moreover the values of classification error and the Rand index values were tested. The classification error for 3 particles equals 3.6% indicates that in the *Gbest* method 22 objects were located to the wrong cluster comparing with their original class. In other cases all objects were classified correctly.

Despite the fact that times are smaller for the *Hybrid* method, it should be remembered that as the first step the *K-means* algorithm is performed. Other parameters, like inertia coefficient or number of particles in the swarm, were also tested, but due to limited space results are not presented in this article.

3.2 Experiment 2

In the second experiment we tested three clustering methods on the data sets described at the beginning of this section and the quality of obtained clusters. Analyz-

Table 3. Average execution time, classification error (B) and Adjusted Rand index (ARI) for *Cl-3* data set

No of particles	*Gbest*			*Hybrid*		
	Time	B	ARI	Time	B	ARI
3	91.38 ± 0.778	0.036	0.890	89.29 ± 1.396	0.0	1.000
10	193.53 ± 1.388	0.0	1.000	131.99 ± 0.522	0.0	1.000
50	369.33 ± 6.698	0.0	1.000	276.37 ± 10.494	0.0	1.000
100	982.85 ± 23.160	0.0	1.000	761.64 ± 7.883	0.0	1.000

ing the values of indices in the Table 4 it can be concluded that we have to deal with different sets of data. Some of them, like *Cl-3* and *Squares4*, have distinct clusters of data, which during clustering process are completely correctly separated. In this case Adjusted Rand Index equals 1 and it means that there is a complete agreement between obtained clusters and original classes. The classification error equaled 0 indicates that there are no objects not correctly classified (located in a wrong cluster).

Table 4. Indices for quality of clusters obtained by 3 grouping methods – classification error (B) and Adjusted Rand Index (ARI)

Data set	Method					
	K-means		*Gbest*		*Hybrid*	
	ARI	B	ARI	B	ARI	B
Cl-3	1.000	0.000	1.000	0.000	1.00	0.000
Mag2	0.129	0.245	0.211	0.215	0.170	0.230
Mag2-out	-0.108	0.457	0.481	0.209	-0.094	0.452
Squares4	1.000	0.000	1.000	0.000	1.000	0.000
Squares2-2c	-0.499	0.500	-0.499	0.500	-0.499	0.500
Squares2-4c	0.001	0.500	0.001	0.500	0.001	0.500
Iris	0.693	0.120	0.806	0.067	0.693	0.120
Wine	0.427	0.197	0.654	0.118	0.613	0.129

The *Mag* and *Mag-out* sets consist of objects belonging to two classes described by normal distribution. The scatter plot of the classes is presented in the Figure 1 in the top-left corner. Each class has the long narrow ellipse shape. In this case all the methods yield rather poor values of indices. The best values of ARI have been achieved for *Gbest* algorithm – *Mag*: 0.211 and *Mag-out*: 0.481. The classification error has the smallest value also for the same method. In the case of *Mag* data 21.5% objects have been classified incorrectly (Fig. 1), while for the *Mag-out* set this value

was 20.9% (Fig 2). The problem of the second data set is the outlier, for which the separate cluster is suggested by the *Gbest* method. For these data sets the process of variables standarization was made, but it has almost no influence on the results. The *K-means* method prefers round-shape clusters. Instead of using Euclidean distance, the Mahalanobis distance could be applied to this kind of data. The alternative is to apply *whitening process* described in [2].

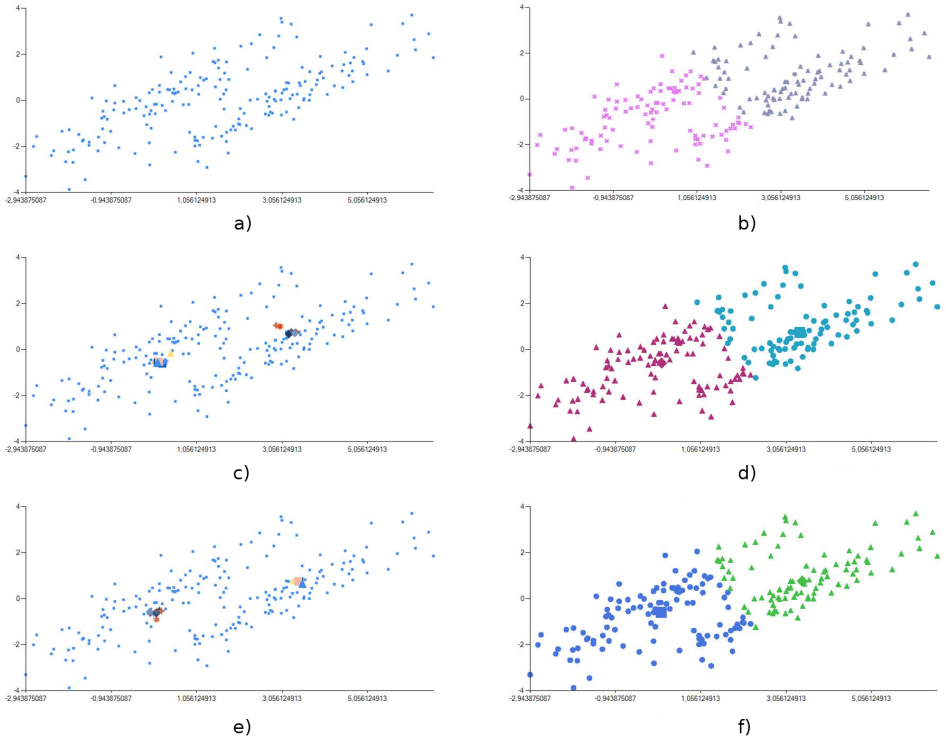

Fig. 1. *Mag* data set: a) scatter plot; particles and centroids for two algorithms: c) *Hybrid* , e) *Gbest*; final clusters obtained for three algorithms: b) *K-means*, d) *Hybrid*, f) *Gbest*

Analyzing *Squares2-2c* and *Squres2-4c* data in all cases there was 50% incorrectly classified objects. The agreement of achieved clusters and original classes was poor. For the first data set there were only two centroids and original classes placed like in the XOR problem. The obtained clusters merged two groups of objects placed

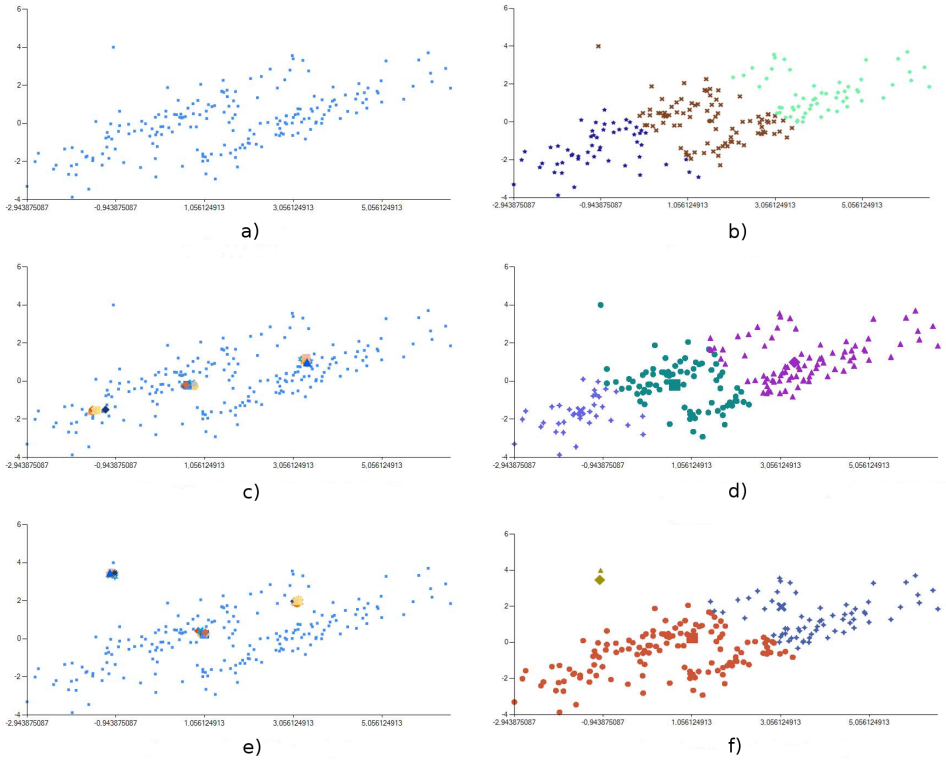

Fig. 2. *Mag-out* data set: a) scatter plot; particles and centroids for two algorithms: c) *Hybrid* , e) *Gbest*; final clusters obtained for three algorithms: b) *K-means*, d) *Hybrid*, f) *Gbest*

at the top into one cluster, and two groups below to the other. Therefore the classification error in each case equaled 50%. In the case of *Squres2-4c* set, there were four clusters – each distinct group of objects constituted distinct cluster. Instead of two original groups, we achieved for clusters. The ARI equaled 0.001 for all methods.
The best value of the ARI in the case of *Iris* set was obtained for the *Gbest* method – ARI=0.806, that means that there is good agreement between clusters and original classes. There were only 6.7% of incorrectly classified objects. For the *Wine* set the *Gbest* was the best algorithm – ARI equaled 0.654 and 11.8% of objects were located incorrectly to the clusters.
To compare algorithms based on the PSO in more detail, the indices like inter-cluster, intra-cluster distances and validity are presented in the Table 5.

Table 5. Indices assessing the quality of clusters obtained by two clustering methods based on PSO – Inter-cluster distance, Intra-cluster distance and Validity

Method	Inter-cluster	Intra-cluster	Validity
Mag			
Gbest	3.5600 ± 1.7601	0.0184 ± 0.0017	0.0148 ± 0.0015
Hybrid	4.2829 ± 1.3883	0.0152 ± 0.0004	0.0141 ± 0.0280
Mag-out			
Gbest	1.6997 ± 1.1968	0.0184 ± 0.0017	0.0245 ± 0.0294
Hybrid	3.0887 ± 1.0358	0.0187 ± 0.0013	0.0099 ± 0.0146
Squares4			
Gbest	0.7333 ± 0.4669	0.0091 ± 0.0006	0.0170 ± 0.0098
Hybrid	0.9879 ± 0.0183	$0.0074 \pm 1.4E-6$	0.0075 ± 0.0001
Squares2-2c			
Gbest	2.9201 ± 1.5894	0.0083 ± 0.0002	0.0119 ± 0.0263
Hybrid	4.0120 ± 0.0139	$0.0081 \pm 3.8E-8$	$0.0020 \pm 6.9E-6$
Squares2-4c			
Gbest	1.2531 ± 1.0588	0.0079 ± 0.0006	0.0489 ± 0.1211
Hybrid	1.0168 ± 0.0196	$0.0074 \pm 9.5E-7$	0.0073 ± 0.0001
Iris			
Gbest	1.8501 ± 1.3326	0.0141 ± 0.0024	0.0435 ± 0.1331
Hybrid	2.2151 ± 1.0719	0.0120 ± 0.0012	0.0083 ± 0.0067
Wine			
Gbest	5.1038 ± 4.3797	0.0390 ± 0.0096	0.0258 ± 0.0314
Hybrid	1.2613 ± 0.8971	0.0476 ± 0.0029	0.0484 ± 0.0190

Considering inter-cluster and intra-cluster distances, the smaller values for the former ensures larger separation between groups of objects, while the latter ensures more compact clusters with smaller variance. In general, for five out of seven examined data sets the inter-cluster distances were larger for the *Hybrid* method than for the *Gbest*.

4. Conclusions

The paper concerns three algorithms of clustering data. Two are based on the swarm intelligence idea [14] and one is classical, well-known method. Two experiments were performed to present the comparison between considered methods. The aim of the first experiment was to check what is the influence of the different values of parameters on the results obtained with *Gbest* and *Hybrid* algorithms. The second experiment described the clusters found by three methods and their quality calculated by using various indices.

It was found that for two data sets *Gbest* method gives larger inter-cluster distances and smaller intra-cluster distances than the *Hybrid* algorithm.
In the future the larger data sets will be tested and selection of the best parameters in these cases will be performed.

References

[1] X. Cui, T.E. Potok and P. Palathingal, Document Clustering using Particle Swarm Optimization *IEEE Swarm Intelligence Symposium*, The Westin , 2005.

[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Morgan Kaufmann, San Francisco, 1990.

[3] T. Hardin, X. Cui, R.K. Ragade, J.H. Graham and A.S. Elmaghraby, A Modified Particle Swarm Algorithm for Robotic Mapping of Hazardous Environments, *The 2004 World Automation Congress*, Seville, Spain, 2004.

[4] L. Hubert and P. Arabie, Comparing partitions., *Journal of Classification*, 193-218, 1985.

[5] R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall International, Inc., 1992.

[6] J. Kennedy and R. Eberhart, Particle swarm optimization., *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, Piscataway, NJ, USA, 1942-1948, 1995.

[7] R.J. Kuo, L.M. Ho and C.M. Hu, Integration of self-organizing feature map and K-means algorithm for market segmentation, *Computers & Operations Research*, Vol. 29, Issue 11, 1475–1493, 2002.

[8] H. Liu and X. Yu, Application Research of k-means Clustering Algorithm in Image Retrieval System, Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT '09), Huangshan, P.R. China, 274-277, 2009

[9] J.B. McQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1, Berkeley, Calif.: University of California Press, 281-297, 1967.

[10] M. Omran, A. Salman and A.P. Engelbrecht, Image classification using particle swarm optimization, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning* (SEAL 2002), Singapore, 370-374, 2002.

[11] O.J. Oyelade, O.O. Oladipupo, I.C. Obagbuwa, Application of k-Means Clustering algorithm for prediction of Students' Academic Performance, *International Journal of Computer Science and Information Security*, Vol. 7, No. 1, pp. 292-295, 2010.

[12] R. Pietrzykowski and W. Zieliński and D. Kozioł, Application of k-means method for a portfolio of shares taxonomy (in Polish), Wyd. Wyższej Szkoły Ekonomiczno–Informatycznej, Warszawa, s.3, 74-76, 2005.
[13] W.M. Rand, Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 66, 846-850, 1971.
[14] D.W. van der Merwe and A.P. Engelbrecht, Data clustering using particle swarm optimization, The 2003 Congress on Evolutionary Computation (CEC'03), vol. 1, 215- 220, Canbella, Australia, 2003.
[15] http:/ /aragorn.pb.bialystok.pl/~magda/data/dane.zip

KLASYFIKACJA NIENADZOROWANA I OPTYMALIZACJA ROJEM CZĄSTEK

Streszczenie: W niniejszym artykule porównywane są trzy algorytmy analizy skupień - metoda k-średnich, algorytm gbest oraz metoda hybrydowa. Algorytmy gbest oraz hybrydowy zostały zaproponowane w publikacji [14]. Wszystkie trzy metody należą do rodziny metod niehierarchicznych, w których na początku tworzony jest podział obiektów na znaną z góry liczbę klastrów. Następnie, niektóre obiekty przenoszone są pomiędzy klastrami, by uzyskać jak najlepszy podział - wariancja pomiędzy skupieniami powinna być znacznie większa niż wariancja wewnątrz skupień. Pierwszy algorytm (k-means) jest znaną, klasyczną metodą. Drugi oparty jest na idei inteligencji roju cząstek. Natomiast trzeci jest metodą hybrydową łączącą dwa wymienione wcześniej algorytmy. Do porównania uzyskanych skupień wykorzystano kilka różnych indeksów szacujących jakość otrzymanych skupień.

Słowa kluczowe: klasyfikacja nienadzorowana, analiza skupień, optymalizacja rojem cząstek