computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

DIVIDING OUTLIERS INTO VALUABLE AND NOISE POINTS

Vladimir E. Podolskiy

Computer Systems and Networks Department, Bauman Moscow State Technical University,
5, vtoraya Baumanskaya st., 105005, Moscow, RUSSIA, v.e.podolskiy@gmail.com, http://iu6.bmstu.ru

Abstract: *A great number of different clustering algorithms exists in computer science. These algorithms solve the task of dividing data set into clusters. Data points which were not included into one of these clusters are called 'outliers'. But such data points can be used for the discovery of unusual behavior of the analyzed systems. In this article we present a novel fuzzy based optimization approach for division these outliers into two classes: interesting (usable for solving the problem) outliers and noise.*

Keywords: *fuzzy sets, outlier analysis, data classification.*

1. INTRODUCTION

In most of the computer science papers notions of noise and outliers are used interchangeably. But when we talk about some kind of data that is very rare among all the data acquired in some research or observation we must take into consideration that real-life observations are based on physical detectors. Thus we may have a few incorrect data points among all the data acquired. Modern algorithms do not make any difference between these erroneous points (or noise) and valuable data [1]. These algorithms are either used to find all the outliers and throw them out or used to find all the outliers and process them.

In the following paper the problem of distinguishing valuable and rare data points from noise points is considered. In order to solve the task posed in the next section a set of different techniques briefly described in section 3 was used. Sections 4 and 5 describe an approach and an algorithm to solve the problem posed. Section 6 shows the results of algorithm's work on one small example. In section 7 some information on comparison is given. Conclusions about possible further developments in this field are made in section 8.

2. PROBLEM DESCRIPTION

Clusterization is a process of dividing data into groups named 'clusters'. One must distinguish clusterization from classification as classification requires knowledge about classes in which data must be divided (e.g. taxonomy classes, political parties). Each data set can be divided into two parts. The first one is the data that can be clusterized. The second one consists of the data points which cannot be put into any of clusters by the means of methods used. Currently existing algorithms were primarily developed to divide data set into clusters or to detect outliers. In this paper we address the problem of handling such outliers and dividing them into two classes. The following terms are needed for our approach description.

Term 1. Interesting point is an outlier which differs from the rest of the data in clusters and can be useful for the person whose primary goal is to detect the facts that do not fall upon restrictions of the laws which describe data clusters.

Interesting points form a class of interesting points, i.e. one of those two classes briefly mentioned before.

Term 2. Noise point is an outlier which differs very little from the rest of the data in clusters and is of no use to the person analyzing outliers.

Noise just represents some fluctuations and errors. No useful information can be obtained from noise in the analysis conducted. Noise points form noise class, the second of the two classes mentioned in the beginning of this paper.

The difference between interesting points and noise can be easily understood by a human (e.g. in the case of two dimensions), but not a computer. To divide outliers into the classes briefly described above we propose usage of basic instruments of fuzzy logic, clustering algorithms, and optimization methods. It is evident, that such a division cannot be performed without previous clusterization of data because outliers division into the classes of interesting points and noise points requires

knowledge about the data distribution and some of the data parameters.

The problem of dividing outliers into the classes of interesting and noise points arises in different fields of study. This problem's solution can be very useful for many applications, such as: geology (distinguishing minerals from ore), fraud detection (distinguishing fraudulent usage of credit cards from a mere change in a customer behavior), search for unusual patterns in researches (distinguishing exceptions from the noise provided by sensors and detectors), search for critical errors in equipment functionality (distinguishing critical issues from minor problems), distinguishing valuable information from background noise (SETI program, cosmology, telescope images), etc.

It can be seen, that the problem stated in the beginning of the paper has high level of topicality for different issues. Of course, this problem requires efficient and logical way of solving. A possible approach to solving this problem is shown in the paper. Following section covers some basic assumptions needed for understanding of the approach developed.

3. BASICS OF THE APPROACH

In order to make the approach proposed work all the data must be clusterized and divided into clusters and outliers. DBSCAN is one of the algorithms for solving this subtask. Let us briefly describe DBSCAN.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm. This algorithm grows regions with high density into clusters and discovers clusters of arbitrary shape in spatial database with noise. DBSCAN defines a cluster as a maximal set of density-connected points.

In order to provide understandable description of the algorithm some definitions must be presented.

The neighborhood within a radius ε of a given object is called the ε-neighborhood of the object.

If the ε-neighborhood of an object contains at least a minimum number, *MinPts*, of objects, then the object is called a core object.

Given a set of objects, *D*, we say that an object *p* is directly density-reachable from object *q* if *p* is within the ε-neighborhood of *q*, and *q* is a core object.

An object *p* is density-reachable from object *q* with respect to ε and *MinPts* in a set of objects, *D*, if there is a chain of objects $p_1,...p_n$, where $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable from p_i with respect to ε and *MinPts*, for $1 \le i \le n$, $p_i \in D$.

An object *p* is density-connected to object *q* with respect to ε and *MinPts* in a set of objects, *D*, if there is an object $o \in D$ such that both *p* and *q* are density-reachable from *o* with respect to ε and *MinPts*.

Density reachability is the transitive closure of direct density reachability, and this relationship is asymmetric. Only core objects are mutually density reachable. Density connectivity is a symmetric relation.

A density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. Every object not contained in any cluster is considered to be noise.

The concept that underlies DBSCAN is rather simple. This algorithm searches for clusters by checking the ε- neighborhood of each point in the database. If the ε-neighborhood of a point *p* contains more than *MinPts*, a new cluster with *p* as a core object is created. DBSCAN then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster.

Figure 1 illustrates DBSCAN's work.

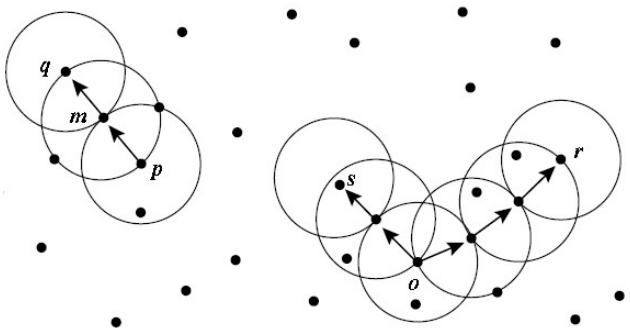

Fig. 1 – Density reachability and density connectivity in density-based clustering

In order to find values of the membership function for each of the outlier points we need to use one of the optimization approaches. Here a brief description of the interior-point method is provided.

Actually, **Interior point methods** (or barrier methods) are a certain class of algorithms to

solve linear and nonlinear convex optimization problems.

The basic elements of the method consist of a self-concordant barrier function used to encode the convex set. Contrary to the simplex method, it reaches an optimal solution by traversing the interior of the feasible region.

In constrained optimization, a field of mathematics, a barrier function is a continuous function whose value on a point increases to infinity as the point approaches the boundary of the feasible region. It is used as a penalizing term for violations of constraints. The two most common types of barrier functions are inverse barrier functions and logarithmic barrier functions.

The class of primal-dual path-following interior point methods is considered the most successful. The primal-dual method's idea is easy to demonstrate for constrained nonlinear optimization. For simplicity consider all-inequality version of a nonlinear optimization problem:

$$\text{Minimize } f(x) \text{ subject to } c(x) \geq 0 \quad x \in R^n, c(x) \in R^m \tag{1}$$

The logarithmic barrier function associated with (1) is

$$B(x, \mu) = f(x) - \mu \ln(c(x)) \tag{2}$$

Here μ is a small positive scalar, sometimes called the "barrier parameter". As μ converges to zero the minimum of $B(x, \mu)$ should converge to a solution of (1).

The barrier function gradient

$$g_b = g - A^T \mu / c(x) \tag{3}$$

Where g is the gradient of the original function $f(x)$ and the matrix A is the constraint $c(x)$ Jacobian.

In addition to the original ("primal") variable x we introduce a Lagrange multiplier inspired dual variable λ

$$c(x)\lambda = \mu \tag{4}$$

(4) is sometimes called the "perturbed complementarity" condition, for its resemblance to "complementary slackness" in KKT conditions.

We try to find those (x_μ, λ_μ) which turn gradient of barrier function to zero. Applying (4) to (3):

$$g - A^T \lambda = 0 \tag{5}$$

Applying Newton's method to (4) and (5) we get an equation for (x, λ) update (p_x, p_λ):

$$\begin{pmatrix} W & -A^T \\ \Lambda A & C \end{pmatrix} \begin{pmatrix} p_x \\ p_\lambda \end{pmatrix} = \begin{pmatrix} -g + A^T \lambda \\ \mu l - C\lambda \end{pmatrix} \tag{6}$$

W is the Hessian matrix of $f(x)$ and Λ is a diagonal matrix of λ.

Because of (1), (4) the condition $\lambda \geq 0$ should be enforced at each step. This can be done by choosing appropriate α:

$$(x, \lambda) \to (x + \alpha p_x, \lambda + \alpha p_\lambda) \tag{7}$$

So, it seems that interior point method is good-enough for our approach.

4. OUTLIERS DIVIDING ALGORITHM

The main part of our approach is the algorithm that divides all the outliers into two classes described in section 1 of this paper. We need to make a few assumptions about data set in order to simplify our task.

Some basic assumptions about the data for the algorithm developed:

1. Data has numerical type, i.e. it can be presented as a point in a linear space;
2. Data has been divided into two sets by the means of the data clustering technique (e.g. DBSCAN, OPTICS, DENCLUE, etc [2]). Results of the division are as follows:
   - C data clusters;
   - M outliers.

The first assumption has been included for the simplicity of experiments with the algorithm and can be removed. The second assumption is necessary because developed algorithm needs information about clusters of data in order to clearly specify for which cluster current point is a noise or an interesting point. Simple but good fuzzy based algorithm for purpose of outlier detection can also be found in [3]. Unfortunately, this algorithm can only be used to detect outliers but not to divide them.

As it is seen, outliers can be both interesting points and noise simultaneously, so we need to use some basic methods of fuzzy sets theory in

order to conclude in which class current outlier must be included. This is a task for the classification algorithm described further in this section. We have decided to use fuzzy sets and optimization methods to classify outliers into two described classes. We need a few more definitions to present our algorithm.

Let $Cl_i = \{X_1, X_2, ..., X_{num(i)}\}$ be the i^{th} ($i = \overline{1,C}$ as was defined in the assumptions above) cluster found by a clusterization algorithm (e.g. DBSCAN). *num(i)* is a number of data points in the i^{th} cluster. Let us also assume that we are working with *D*-dimensional data, i.e. each point can be presented as a point in a linear space with *D* coordinates: $X_j = \left(x_1^{(j)}, x_2^{(j)}, ...x_D^{(j)}\right)$ where $j = \overline{1,N}$ X_j is a j^{th} point, and *N* is a total number of data points (including outliers).

Definition 1. Let us assume that point X_k is an outlier, and Cl_i is a cluster closest to this point (i.e. Euclidean distance from this point to the center of this cluster is the smallest one compared to the distances from this point to other clusters). Then the point X_j in the cluster Cl_i is called a projection of the point X_k if the distance between $X_j^{(i)}$ and X_k is lesser than any distance between X_k and every other point of this cluster, i.e.

$$p^{(i)}(X_k) = \left\{X_j^{(i)} : dist\left(X_j^{(i)}, X_k\right) = \min_{j=\overline{1,num(i)}} dist\left(X_j^{(i)}, X_k\right)\right\}$$

The *dist* function can be defined as Euclidean distance.

Let us also assume that each outlier (X_k) belongs to noise class of one of the clusters (Cl_i) with some degree $\mu_i(X_k) \in [0;1]$ and to the class of interesting data points with degree $\theta_i(X_k) = 1 - \mu_i(X_k)$. This is the only element (also called 'membership function' [4]) of fuzzy sets theory that we use in the paper for dividing outliers into two classes. A criterion based on a value of membership function for dividing outliers into two classes will be described further in the paper.

Definition 2. Minimal distance between the outlier X_k and the cluster Cl_i is a distance between outlier and its projection on this cluster, i.e.: $d_{\min}\left(X_k, Cl_i\right) = dist\left(X_k, p^{(i)}\left(X_k\right)\right)$.

Let us assume that we have found values of membership functions for every outlier X_k and its closest cluster Cl_i, i.e. we know values of $\mu_i(X_k)$ and $\theta_i(X_k)$ for the point X_k, and we also know the minimal distance $d_{\min}\left(X_k, Cl_i\right)$ between outlier point X_k and its closest cluster Cl_i. Let us also define a decision boundary for the i^{th} cluster as:

$$bnd_i = \frac{\sum_{k=1}^{q} \mu_i(X_k) \cdot d_{\min}(X_k, Cl_i)}{\sum_{k=1}^{q} d_{\min}(X_k, Cl_i)}, \qquad (8)$$

where *q* is a number of outliers closest to the i^{th} cluster.

Then we can use following criterion to decide which class current outlier belongs to:

$$dec(X_k) = \begin{cases} \text{noise, if } \mu_i(X_k) > bnd_i, \\ \text{interest. point, otherwise.} \end{cases} \qquad (9)$$

This criterion is the most suitable for the current task.

Now let us describe a general idea behind the algorithm and then provide a full description of our approach. For simplicity we will use this notation: $\mu_k^{(i)} = \mu_i(X_k)$.

The idea that was partially taken from [5] is as follows. We can substitute classification task with a task of maximizing *C* linear functions of q_i variables (q_i is a number of outliers closest to the i^{th} cluster, such that $\sum_{i=1}^{C} q_i = M$), i.e. we must find global maximum for each of the following functions:

$$f_i(\mu_1^{(i)}, \mu_2^{(i)}, ..., \mu_{q_i}^{(i)}) = \sum_{k=1}^{q_i} \frac{\mu_k^{(i)}}{d_{\min}(X_k, Cl_i)}, \qquad (10)$$

$i = \overline{1,C}$.

We characterize structure of the i^{th} cluster with a parameter R_i that can be computed as a radius of the i^{th} cluster, i.e. it equals half of a maximum distance between two points in this cluster. In order to take into consideration structure of the cluster, we must also define constraints on production of value of membership function and minimal distance between outlier and the closest cluster:

$\mu_k^{(i)} d_{\min}(X_k, Cl_i) \le R_i, k = \overline{1, q_i}, i = \overline{1, C}$

The system is written for i^{th} cluster where $i = \overline{1, C}$:

$$\begin{cases} f_i(\mu_1^{(i)}, \mu_2^{(i)}, ..., \mu_{q_i}^{(i)}) \to \max \\ \mu_k^{(i)} \cdot d_{\min}(X_k, Cl_i) \le R_i, k = \overline{1, q_i}. \\ 0 \le \mu_k^{(i)} \le 1, k = \overline{1, q_i} \end{cases} \quad (11)$$

We have C such systems. It is evident, that we must use optimization methods that take constraints into consideration (e.g. method of interior point or Lagrange multipliers method). For numerical solution we decided to use method of interior point [6] (penalty functions method) described earlier in this paper.

Interesting situation occurs when outlier is a valuable point for all the data clusters. It is then can be considered as a so-called *global valuable point*. It can be also shown that these global valuable points require special treatment because they can belong to the class of error as well. Let us briefly discuss this problem.

If we have many global valuable points (i.e. their frequency of appearance in dataset is large) then it is highly possible that detector (device that provides us with the data) is broken. If the frequency is not so large then it occurs that there are some super-exceptional data points that are to be considered first. The formal criterion for deciding whether the device providing data is broken or not is as follows:

$$\begin{cases} num_{cl\_interest}(X_j) = C, \\ \dfrac{w_g}{M} \ge 0.5, \ j = \overline{1, w} \end{cases}, \quad (12)$$

where C is a total number of data clusters, $num_{cl\_interest}(X_j)$ is an amount of classes for which the data point in parenthesis is valuable point, w_g is an amount of global valuable points, M is a total amount of outliers, w is a total amount of valuable data points.

We can also consider the following value:

$$P(X_j) = \frac{num_{cl\_interest}(X_j)}{C}, \quad (13)$$

Due to discrete nature of valuable (interesting) points we can only consider discrete values like (13) for $j = \overline{1, w}$. But we can also construct some approximation based on the values $P(X_j)$ and X_j. This can be done by means of interpolation techniques. Construction of such hyperplane can help us to predict amount of classes for which the point given is a valuable point. This can be used to consider the problem discussed earlier (see eq. (12)).

5. OUTLIERS PARTITIONING APPROACH

Now we can fully and consequently describe our simple approach:

1. Divide data set into C clusters by one of the clustering techniques (e.g. DBSCAN that can handle clusters of arbitrary form).
2. For every outlier find closest cluster.
3. For every cluster and every outlier bound to this cluster (closest cluster, see step 2) find projection of outlier on this cluster.
4. For every outlier and its projection find outlier's minimal distance to the closest cluster (steps 3 and 4 can be unified based on the definition of projection given in this paper).
5. For every i^{th} cluster define starting point for optimization method (e.g. $\mu_1^{(i)} = 0, \mu_2^{(i)} = 0, ..., \mu_{q_i}^{(i)} = 0$) that satisfies initial conditions of system (11).
6. For every cluster find solution for system (4).
7. For every cluster compute decision boundary using formula (8).
8. For every cluster and outlier closest to cluster decide (based on the criterion (9)), whether outlier under consideration belongs to the noise class or to the interesting points.

This approach is rather simple and modulate. For example, we can use another data clustering technique or another decision criterion.

6. ILLUSTRATIVE EXAMPLES

In this section we apply the approach described above to sets of data points defined further in this paragraph. These data sets are distributions of points on a two-dimensional plane. First random distribution can be easily divided into two groups by clustering algorithm (in our realization we have used DBSCAN [7]). The second one can be divided in three classes. For test purposes application MATLAB® was used. MATLAB code for the first distribution described above is as follows:

```
X=[randn(30,2)*.4;randn(40,2)*.5
                +
        ones(40,1)*[4 4]];
```

We have changed coordinates of nine points in this set in order to convert them to outliers. Coordinates of these points are provided in table 1. Third column presents the class of each outlier defined by approach from section 5.

Table 1. Outliers and classes.

Point №	X coord.	Y coord.	Class
1	11.00	10.00	Interesting
2	2.00	4.00	Noise
3	0.43	4.00	Noise
4	-1.00	-1.33	Noise
5	15.00	17.00	Interesting
6	33.00	4.00	Interesting
7	-3.00	-5.00	Noise
8	-3.55	4.98	Noise
9	9.00	4.30	Noise

Figure 2 illustrates our example. Symbol '*' denotes point from 1st cluster, '+' – from 2nd. Symbol '°' is reserved for noise, and symbol 'X' denotes interesting points (number 1, 5, and 6 in table 1).

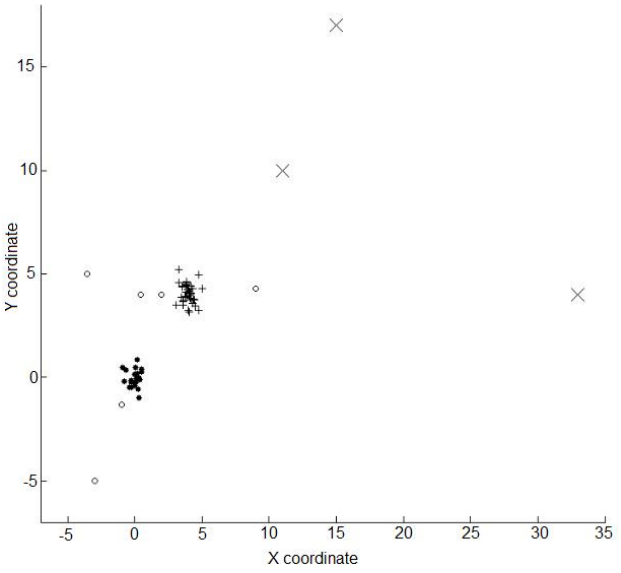

Fig. 2 – Results of applying approach to the test data set #1

The second example shows us that form of clusters has no effect on results of application of approach developed. Repetitive experiments on the data show that slight change in placement of outliers can result in changing classes for all outliers (e.g. from noise to interesting). This problem occurs only for small data sets. For larger data sets with many more outliers this problem with robustness influences results very little. In table 2 we present data for second example.

Table 2. Data for second example.

X	Y	Class	X	Y	Class
0.500	0.500	1	2.375	1.750	2
0.625	0.625	1	2.500	1.750	2
0.625	0.500	1	2.375	1.625	2
0.500	0.625	1	2.500	1.625	2
0.750	0.750	1	2.500	1.500	2
0.500	0.750	1	0.250	3.500	3
0.750	0.500	1	0.375	3.500	3
0.750	0.625	1	0.250	3.625	3
0.625	0.750	1	0.375	3.625	3
0.875	0.875	1	0.500	3.625	3
0.750	0.875	1	0.625	3.625	3
0.875	0.750	1	0.750	3.625	3
0.875	0.625	1	0.875	3.625	3
0.625	0.875	1	1.000	3.625	3
0.500	0.875	1	0.500	3.750	3
0.875	0.500	1	0.625	3.750	3
1.000	1.000	1	0.750	3.750	3
0.875	1.000	1	0.875	3.750	3
0.750	1.000	1	1.000	3.750	3
0.625	1.000	1	0.875	3.875	3
0.500	1.000	1	1.000	3.875	3
1.000	0.500	1	1.125	3.875	3
1.000	0.625	1	1.250	3.875	3
1.000	0.750	1	1.125	4.000	3
1.000	0.875	1	1.250	4.000	3
2.000	2.000	2	1.500	4.500	Noise
2.125	2.000	2	0.000	2.000	Int. p.
2.250	2.000	2	1.750	0.750	Noise
2.375	2.000	2	1.875	5.000	Noise
2.500	2.000	2	1.750	3.000	Noise
2.125	1.875	2	2.750	0.500	Noise
2.250	1.875	2	3.000	0.000	Noise
2.375	1.875	2	1.000	8.000	Int. p.
2.500	1.875	2	0.000	5.000	Int. p.
2.250	1.750	2	1.500	1.500	Noise

Figure 3 illustrates this example.

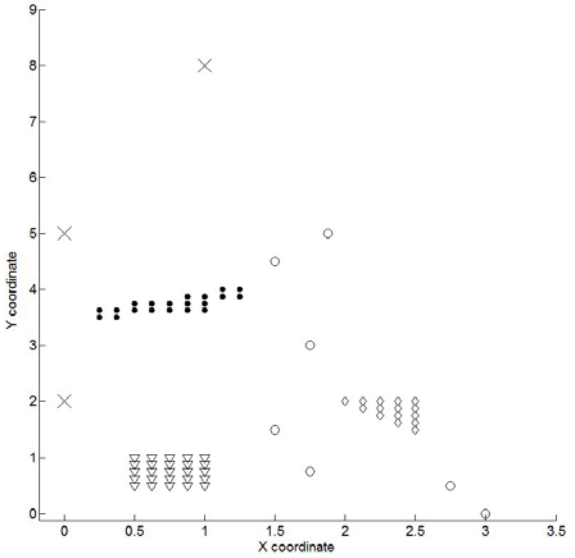

Fig. 3 – Results of applying approach to the test data set #2

For these simple examples we can imagine membership function as a mesa in three-dimensional space, where third coordinate is a value of the membership function. Points of the clusters are on the plain part of the mesa. Noise points lie higher on mesa's slopes than the boundary value. Interesting points are under the boundary value on the slopes. These illustrations give us some basic insights on this approach.

7. COMPARISON

Unfortunately, it seems that the problem addressed in this paper almost hasn't been considered in other papers. So, no other similar approaches have been found because almost all of the papers in the field of AI consider only the problem of clusterization at large.

The approach provided has time complexity $O(n^2)$. This approach is rather unique because we do not use clusterization technique to extract outliers from data and then two-class classifiers on the outliers extracted. Such method will fail because it does not consider clusters at all. But the approach provided in this paper is based on the cluster characteristics. And therefore we can easily divide outliers into noise and valuable points.

8. CONCLUSION

The problem addressed in the paper is interesting for many reasons and can be met in such fields as: handling research data, fraud detection, search for minerals, etc. The approach proposed is mathematically convenient, simple and modulate. Developed approach can be used in its initial form or it can be modified for the sake of optimization and simplicity. There are a great number of algorithms that can be used for solving such a problem, but described approach can lead to simple mathematical expressions. The approach presented is also very intuitive, simple and flexible in comparison to other approaches (neural networks classification, SVM-based classification, decision trees classifiers, some fuzzy based algorithms, etc). So, this approach also briefly introduced in [8] is important and can be used for variety of tasks. Further researches in this field may be conducted in order to obtain sustainable and useful solutions for the problem stated in the beginning of the paper.

9. REFERENCES

[1] F. Rehm, F. Klawonn, R. Kruse, A novel approach to noise clustering for outliers detection, *Soft Computing,* (11) 5 (2007), pp. 489-494.

[2] J. Han, M. Kamber, *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, USA, 2006, 743 p.

[3] D. Viattchenin, Direct algorithms of fuzzy clustering based on the transitive closure operation and their application to outliers detection, *Artificial Intelligence,* (3) (2007), pp. 205-216. (in Russian)

[4] A. Kaufmann, *Introduction to Fuzzy Sets Theory,* Radio and Contact, Moscow, 1982, 432 p. (in Russian)

[5] D. Viattchenin, *Fuzzy Methods of Automatic Classification,* Minsk, Technoprint, 2004, 219 p. (in Russian)

[6] M. Gautam, R. Levkovitz, *Interior Point Methods for Linear Programming Optimization: Theory and Practice,* John Wiley & Sons, USA, 1996, 300 p.

[7] M. Daszykowski, B. Walczak, D.L. Massart, Looking for Natural Patterns in Data. Part 1: Density Based Approach. Chemom. Intell. Lab. Syst. (56) (2001), pp. 83-92.

[8] V.E. Podolskiy, Simple fuzzy based optimization approach to the problem of dividing outliers into classes of valuable and noise points, *Pattern Recognition and Information Processing (PRIP'2011): proceedings of the 11th International Conference (18-20 May, Minsk, Republic of Belarus),* Minsk: BSUIR, 2011, pp. 176-179.

Vladimir Eduardovich Podolskiy, *student at Computer Systems and Networks Department of Informatics and Control Systems faculty at Bauman Moscow State Technical University. Areas of scientific interest include: fuzzy systems, fuzzy methods in sociology, data mining, clusterization, parallel computations, global politics, and human psychology.*