OUTLIER DETECTION IN SELF-ORGANIZING MAPS AND THEIR QUALITY ESTIMATION

P. Stefanovič[*], *O. Kurasova*[†]

Abstract: In the paper, an algorithm that allows to detect and reject outliers in a self-organizing map (SOM) has been proposed. SOM is used for data clustering as well as dimensionality reduction and the results obtained are presented in a special graphical form. To detect outliers in SOM, a genetic algorithm-based travelling salesman approach has been applied. After outliers are detected and removed, the SOM quality has to be estimated. A measure has been proposed to evaluate the coincidence of data classes and clusters obtained in SOM. A larger value of the measure means that the distance between centers of different classes in SOM is longer and the clusters corresponding to the data classes separate better. With a view to illustrate the proposed algorithm, two datasets (numerical and textual) are used in this investigation.

Key words: *outlier detection and rejection, self-organizing map, SOM quality estimation*

Received: April 6, 2016
Revised and accepted: April 16, 2018

DOI: 10.14311/NNW.2018.28.006

1. Introduction

Recently, data mining remains one of the most relevant domains due to a huge amount of data generated by various devices, sensors, etc. A lot of data mining methods have been developed to solve problems of classification, clustering, association rule generation, anomaly detection. The most known and widely used methods are: k-means, neural network, genetic algorithms, fuzzy logic, Bayesian networks, hierarchical clustering, etc. [17]. One of data mining methods is a self-organizing map (SOM). Sometimes it is called by the name of the creator – a Kohonen map [20]. SOM can be used to cluster and visualize multidimensional data as well as to and a multidimensional data projection into a space of a smaller number of dimensions. The main SOM advantage, as compared with other data mining methods, is that, as a result, SOM gives not only some numerical estimates,

[*]Pavel Stefanovič – Corresponding author, Vilnius Gediminas Technical University, Faculty of Fundamental Science, Saulėtekio al. 11, LT-10223 Vilnius, Lithuania, E-mail: pavel.stefanovic@vgtu.lt

[†]Olga Kurasova, Vilnius University, Institute of Data Science and Digital Technologies, Akademijos str. 4, LT–08663 Vilnius, Lithuania, E-mail: olga.kurasova@mii.vu.lt

like most data mining methods do, but also the result is presented in a visual form [29]. Visualization allows researchers to see clusters and relations between the data analyzed. Comparing SOM with other clustering methods, there are no precisely defined clusters in SOM, i.e. the data are not unambiguously assigned to one or other cluster. Clustering results can be variously interpreted by researchers when exploring a visual representation of SOM [35]. SOM can also be applied to data, assigned to classes. In this case, a researcher can investigate whether classes are coincident with the clusters, obtained in SOM, and explain the reasons for not being coincident, one of which may be related to the fact that the data were incorrectly assigned to the classes. The coincidence can be explored in the visual SOM representation, however, it is purposeful to have numerical estimates that show overlaps of data classes and clusters in SOM. Moreover, it does not matter which kind of data (numerical, textual, images, etc.) will be analyzed, it is often possible to find some data items, called outliers that differ from other items [22]. It is necessary to take into consideration the existing outliers, when estimating the coincidence of data classes and clusters in SOM, as outliers can distort the values of estimates. There are some reasons for existing outliers: data errors, intentional or motivated reporting, sampling error, standardization failure, etc. [26]. Usually, outliers can be considered as corrupted (faulty) data. For example, the data are described and coded in a wrong way and an experiment, generating data is carried out incorrectly. If it is possible to determine that an outlier is certainly erroneous, then it has to be removed from the dataset because it can influence data mining results. Otherwise, in some cases, it is difficult to find the reasons why one or other data object is distinct from other data items. In any way, the outlier cannot be simply removed, it depends on reasons of outlier existing and types of the dataset analyzed, etc. Firstly, it is necessary to analyze these phenomena deeply, because outliers can indicate something scientifically interesting. The main goal of the research is to find and reject outliers in SOM with a view to precisely estimate the coincidence of the data classes and clusters obtained in SOM.

2. Related works

2.1 Self-organizing map

The self-organizing map is one of the most popular artificial neural network (ANN) models, developed by Professor T. Kohonen [20]. SOM is trained by unsupervised learning. The main aim of SOM is to preserve the topology of multidimensional data when they are transformed into a lower dimensional space (usually two-dimensional). The set of weights forms a vector $M_{\mathrm{ij}}, i = 1, \ldots, k_{\mathrm{x}}, j = 1, \ldots, k_{\mathrm{y}}$ that is usually called a neuron or codebook vector, where k_{x} is the number of rows, and k_{y} is the number of columns of SOM (in the case of a rectangular topology). SOMs can be used to cluster, classify, and visualize different kinds of datasets, so first of all, the datasets have to be transformed to numerical expressions. Suppose a dataset $X = \{X_1, X_2, \ldots, X_{\mathrm{N}}\}$ is analyzed, where each data item is described by the features $x_1, x_2, \ldots, x_{\mathrm{N}}$, i.e. a data item $X_{\mathrm{p}} = (x_{\mathrm{p1}}, x_{\mathrm{p2}}, \ldots, x_{\mathrm{pn}})$. So, X_{p} is a point (or a vector) in the n-dimensional space, $X_{\mathrm{p}} \in R^n$. All data are presented to SOM as a matrix:

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{pmatrix}. \tag{1}$$

Here x_{pl} is the value of the lth component of the vector $X_p, p = 1, \ldots, N, l = 1, \ldots, n$. N is the number of analyzed input vectors, and n is the number of components. The learning process of the SOM algorithm starts from initialization of components of the vectors (neurons) M_{ij}. They can be initialized at random (usually these values are random numbers from the interval $(0, 1)$) or by the principal components. At each learning step, an input vector $X_p \in \{X_1, X_2, \ldots, X_N\}$ is passed to SOM. The vector X_p is compared with all the neurons M_{ij}. Usually the Euclidean distance between this input vector X_p and each neuron M_{ij} are calculated. The vector (neuron) M_w with the minimal Euclidean distance to X_p is designated as a neuron-winner (best match unit). Components of all neurons are adapted according to the learning rule:

$$M_{ij}(t+1) = M_{ij}(t) + h_{ij}^{w}(X_p - M_{ij}(t)). \tag{2}$$

Here t is the number of a learning step, h_{ij}^{w} is a neighboring function [28], w is a pair of indices of the neuron-winner of vector X_p. The learning is repeated until the maximum number of learning steps is attained. After SOM learning, the data are presented to SOM, and the neurons-winners for each X_p are found. In such a way, the data items are distributed in SOM, and some data clusters can be observed (see Fig. 1). In the top left side of the map (Fig. 1a), the cluster from I and II classes members are located, and in the opposite corner of the map are placed the members from III and IV classes. The elements from same classes are near to each other or are in the same cell, so it means that it can be considered as a strong cluster. In the other map (Fig. 1b) the elements are more distracted, so in this case, clusters are not so strong and can be defined differently.

More than 30 years have passed since SOM has been introduced, thus the new extensions and modifications of the original SOM are constantly developed. Some of the most known extensions are listed here: a merge self-organizing map (MSOM) [32], a recursive self-organizing map (RecSOM) [34], WEBSOM [19], etc. Mostly all of them were created to speed-up the learning algorithm or to perform specific data mining tasks. For example, WEBSOM is the first SOM extension developed for the textual document analysis. By using the WEBSOM method, a textual documents collections are presented on the map, where similar documents always are placed nearby. This method helps to observe obtained clusters of text documents data, and to find similarities in the contexts of the words. MSOM model is based on ordinary self-organizing map algorithm, but differently, than usual SOM the contextual vector is included. The contextual vector is the same size as an input vector. This vector consists of previous training neuron-winner data and the weights of input vectors, which are merged to the one vector. MSOM can be used in models, which do not have a specific topology, and also for structured data analysis. The main difference of RecSOM from other SOM methods is that simplest recursive self-organizing maps are using neurons as leaky integrators and the most

difficult are doing the copies of all neurons from previous learning step. One of the SOM modifications is the batch-learning SOM (BLSOM), used in the bioinformatics area [15]. In this approach, SOM has been modified for gene informatics to make the learning process and the resulting map independent of the data analyzed. BLSOM is a powerful tool for big data analysis. It allows us to visualize and classify big sequences, obtained from genomes (millions of metagenomics sequences). Another SOM modification for a large data set is an environment self-organizing map (EnvSOM) [4]. The EnvSOM algorithm consists of two phases. In the first phase, a SOM is trained using all the data features, but only environment features of the data are used to find a neuron-winner. In the second phase, a new SOM is created appropriately with information from the reference vectors of the first phase SOM. In this phase, SOM uses all the data set features for neuron-winner computation. There are also researches related to combinations of SOM with dimensionality reduction methods, where vectors of neurons-winners are mapped onto a space of lower-dimensionality by multidimensional scaling [21]. Some researchers combine self-organizing maps with the modified k-means algorithms to solve high dimensional data problems [25]. The main steps of the method are as follows: (1) SOM is used to reduce the dimensionality of the data and to determine the number of clusters; (2) the genetic algorithm is applied to the reduced dimensionality data in order to obtain the initial centers of the clusters; (3) the k-means algorithm is used to get the resultant clusters. Thus, in this method, SOM is used for dimensionality reduction and visualization.

2.2 Outlier detection

One of the challenges in data mining is dealing with outliers. In general, there are five major categories of approaches for outlier detection in the literature [6, 9, 10, 33]: distribution-based, clustering-based, distance-based, density-based, and depth-based methods. The distribution-based techniques use standard methods for estimating statistical distribution. The depth-based methods aim to detect outliers by computing a distance measure of a particular data item to the centroid of data. The distance-based methods aim to measure the distance between a data item and its neighbor. The density-based approaches compare the density around a data item with the density around its local neighbors. There are also other classifications of outlier methods, such as supervised, semi-supervised, and unsupervised detection. Selection of these approaches depends on various situations and data mining tasks [3, 12].

A lot of techniques and approaches have been developed in the recent years to solve outlier detection problems. Various techniques have been proposed and applied in a series of papers [2, 13, 23, 27, 37] including SOMs as well [1, 7, 8]. In the paper [23], the proposed algorithm consists of two stages. In the first stage, an improved genetic k-means algorithm is used, and in the second stage, the vectors (outliers) which are far from their cluster centroids are removed. In the papers [7,8], a SOM-based algorithm for spatial outliers with multiple spatial and non-spatial attributes has been proposed. The Mahalanobis distance concept was used to determine a threshold for identifying spatial outliers. With an iterative utilization of SOM, the neighbor set can be effectively updated to eliminate the influence of

potential local outliers for more robust detection. There are various fields where detection of outliers is a relevant issue: clinical trials [11], meteorology [36], finance [18], etc.

3. Estimation of the SOM quality

After training SOM, usually, quantization and topographic errors are calculated [20]. A quantization error shows how well neurons of the trained network adapt to the input vectors. It is an average distance between the data vectors $X_{\mathrm{p}}, p = 1, \ldots, N$ and their neuron-winners. A topographic error shows how well the trained network keeps the topography of the data analyzed. When the classified data are analyzed by the clustering methods, there is a need to evaluate the coincidence between data classes and the obtained clusters. The coincidence indicates that the data are assigned to appropriate classes. In a mismatched case, a researcher must seek causes of the mismatch. One of the possible reasons is that the data are assigned to unsuitable classes. However, neither quantization nor topographic errors show whether the analyzed data classes correspond to the clusters formed in SOM. There are some other errors that help evaluate the coincidences between the classes and the obtained clusters, obtained not by SOM, but other clustering methods [24]. However, in those cases, the data must be unambiguously assigned to one of the clusters. The uniqueness of SOM, compared to other clustering methods, is that in the SOM results there are no strictly expressed clusters, i.e., it is not specified which data item is assigned to which cluster, a researcher can only observe clusters in the visual representation of SOM.

In our previous work, two measures have been proposed [30]. The measures can be applied to compare several SOMs when analyzing the same dataset and the SOM sizes are the same. Both measures are helpful to estimate the SOM quality. The first measure shows how close the same class members are in SOM. The smaller value of the measure indicates better results. It means that all the same class members are closer to each other, the clusters are "stronger". The second measure shows how far the centers of different classes are in SOM. The higher value of the measure means better results, i.e., all the different class centers are far from each other, so they are separated in the map. However, deeper investigations identify that one of the proposed measures (E'_{center}) has a disadvantage. There are some cases where the measure does not show accurate results. Here, the measure has been modified to eliminate shortages [31].

The procedure of measure computation is described below. First of all, the indices of centers Y^c of each class in SOM are computed:

$$Y^c = \frac{1}{N_c} \sum_{i=1}^{N_c} Z_i^c. \qquad (3)$$

Here N_{c} is the number of data items from the c-th class, $c \in \{1, \ldots, k\}$, k is the number of data classes; Z_{i}^c is a vector that consists of the indices of SOM cells, corresponding to the data from the cth class, $Z_{\mathrm{i}}^c \in R^2$. In our previous work [30]

the value of the measure E'_{center} was calculated by the formula:

$$E'_{\text{center}} = \frac{1}{m} \sum_{c=1}^{m-1} \sum_{d=c+1}^{m} \|Y^c - Y^d\|. \tag{4}$$

Here $m = k(k-1)/2$, A problem appears if different class members fall into the same SOM cell (Fig. 1) and the distances between clusters are longer. So, to make the measure E'_{center} more accurate, it should be modified by adding weights $w_c = (n'_c)/n_c$ (5), where n_c is the total number of neurons (cells) corresponding to the data from the c-th class, n'_c is the number of neurons corresponding only to the data from the cth class. The weights reduce the distance value between clusters of the different classes. If SOM where the members of different classes fall into the different SOM cells is analyzed, the value of the measure E_{center} will be higher than in the case when the classes overlap in SOM.

$$E_{\text{center}} = \frac{1}{m} \sum_{c=1}^{k-1} \sum_{d=c+1}^{k} \|Y^c - Y^d\| w_c w_d \tag{5}$$

The motivation of necessity to modify the measure E_{center} is illustrated by a simple example. Suppose two SOMs of the same size and the same dataset with four classes is analyzed, but differently mapped in SOMs (Fig. 1). Here, the highlighted numbers denote the analyzed data class labels ($c = 1, 2, 3, 4$). The pairs of numbers in the corner of each cell indicate the cell indices.

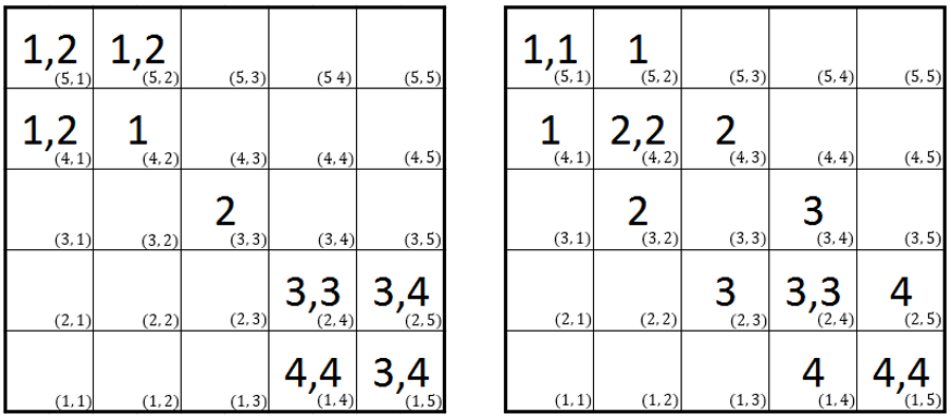

Fig. 1 *Example of two SOMs: a)* $E'_{center} = 2.86$, $E_{center} = 0.36$*; b)* $E'_{center} = 2.82$, $E_{center} = 2.82$.

In picture Fig. 1a, there are SOM cells, into which not only the same class members fall, i.e., some classes overlap. Estimating the level of overlapping is necessary. If both maps will be observed and compared visually, it is possible to see that the classes are far away from one another in Fig. 1b, but the value of

the measure E'_{center} is larger in the case of Fig. 1a. If the measure is evaluated by Eq. (5), the weights for each class should be computed: $w_1 = 1/4$, $w_2 = 1/4$, $w_3 = 1/3$, and $w_4 = 1/3$. These values of the weights decrease the measure from 2.86 to 0.36. In the case of Fig. 1b, the value of the measure does not change, because there are no cells, into which the different class member would be fallen. Thus, when SOMs are compared by the modified measure, the higher value indicates that the centers of different classes are farther and clusters are separated better.

4. Outlier detection and rejection

The definitions of outliers are quite similar in various literature, and the general intent is described by definition given by Hawkins [14]: "an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". Depending on the aim of application or used method, the term of outlier always can be named such as anomaly, noise, deviation, exception or novelty. Outliers in a dataset correspond to a very small percentage of the data objects. It always has something unordinary.

As it is mentioned before, the advantage of SOM, as compared to other data mining methods, is a capability of dataset visualization. When SOM has been trained, obtained clusters in SOM can be observed. There is a possibility to see some data items that can be far away from other members of the same class on the map. Such data items can be considered as a SOM-based outlier. First of all, it is necessary to confirm or deny outlier presence. If the outliers found are treated as bad data, they should be rejected before estimating the SOM quality.

In this paper, an algorithm that allows to detect and reject outliers in the trained SOM have been proposed. In the procedure of outlier detection, the Travelling Salesman (TS) approach [16], based on genetic algorithm, has been used. A particularity of the approach is that it searches for the shortest path from the fixed starting point without limit to the last point. If there is an outlier in SOM, it has to be removed before estimating SOM by the proposed measure (5). A general algorithm for outlier detection, rejection, and SOM quality estimation is described below step-by-step (Algorithm 1).

5. Simulation results and analysis

The SOM system, designed and described in [29] is used in the simulation analysis. The system has been improved by implementing the modified measure (5) as well as the algorithm proposed for outlier detection and rejection (Algorithm 1). Two datasets are used to simulate results: numerical and textual. If the data features gain the numerical values, such data will be called as numerical ones. The data obtained from the text documents are called textual data. The following values of parameters are selected in the simulation: $q = (k_x + k_y)/4, p = 10\%$.

A wine dataset has been chosen as numerical data [5]. The dataset consists of the results of chemical analysis of wines grown in the same region in Italy, but derived from three different cultivators. The analysis has determined the quantities of

13 constituents found in each of the three types of wines. Thirteen-dimensional vectors $X_1, X_2, \ldots, X_{178}$ are formed, where $X_\mathrm{p} = (x_{\mathrm{p1}}, x_{\mathrm{p2}}, \ldots, x_{\mathrm{p13}}), p = 1, \ldots, 178$. The dataset is divided into three classes: Class 1 – wines of the first farmer, Class 2 – wines of the second farmer, and Class 3 – wines of the third farmer.

The SOM result of the wine dataset is depicted in Fig. 2a. The pie charts in SOM shows a proportion between the data items that are assigned to the different

Algorithm 1 Outlier detection in self-organizing map: Part I.

Step 1. The dataset $X = \{X_1, X_2, \ldots, X_\mathrm{N}\}$ is chosen, where N is the number of data items. Each item $X_\mathrm{p}, p = 1, \ldots, N$ has to be assigned to a class $c \in \{1, \ldots, k\}$, where k is the number of data classes.

Step 2. The SOM learning parameters are selected (neighboring function, learning rate, SOM size $k_\mathrm{x} \times k_\mathrm{y}$).

Step 3. SOM is trained. As a result matrices Z^c are formed, $c = 1, \ldots, k$, each row of the matrix represents the indices of SOM cells, corresponding to the data from the c-th class, $Z^c \in R^2$.

Step 4. The rows $Z^c_\mathrm{i}, i = 1, \ldots, N_\mathrm{c}$ of each matrix Z^c are sorted in descending order according to the first column values, where n_c is the total number of SOM cells, corresponding to the data from the cth class. The sorted matrices Z^c are obtained, where the indices of their rows are renewed and denoted by $Z^c_\mathrm{i}, i = 1, \ldots, N_\mathrm{c}$.

Step 5. Starting from the first row of each matrix Z^c, the shortest path between all the points, corresponding to the rows of the matrix Z^c, is found using the traveling salesman algorithm.

Step 6. The rows $Z^c_\mathrm{i}, i = 1, \ldots, N_\mathrm{c}$, of matrices Z^c are sorted according to the order in the found path. The sorted matrices Z^c are obtained, where the indices of their rows are renewed and denoted by $Z^c_\mathrm{i}, i = 1, \ldots, N_\mathrm{c}$.

Step 7. The Euclidean distances between two adjacent rows Z^c_i are calculated and saved as a vector $d^c = \{d^c_1, d^c_2, \ldots, d^c_{N_\mathrm{c}-1}\}$, where $d^c_\mathrm{i} = \|Z^c_\mathrm{i} - Z^c_{\mathrm{i}+1}\|, i = 1, \ldots, N_\mathrm{c} - 1$.

Step 8. The vectors r^c_j consisting of d^c_i indices have to be created. The value of the threshold q is selected, which shows how far a data item should be from the nearest item in the shortest path for the item to be considered as an outlier:

$j = 2; r^c_1 = 0;$
FOR $i = 1$ **TO** $N_\mathrm{c} - 1$
  IF $(d^c_\mathrm{i} > q)$
    $r^c_j = i;\ j = j + 1;$
  END
END
$r^c_\mathrm{j} = N_\mathrm{c};$
  IF $(length(r^c) = 2)$
    Go to Step 11
  END

Step 9. The percentage p is selected which shows how many members of the c-th class can be distant from the other members of the same class so that they were not considered as outliers. The bound b^c is calculated: $b^c = (p \times N_\mathrm{c})/100$.

Algorithm 2 Outlier detection in self-organizing map: Part II.

Step 10. Outlier data rejection is realized:
FOR $i = 1$ **TO** $j - 1$
  IF $(r^c_{i+1} - r^c_i \leq b^c)$
    Remove rows from $r^c_i + 1$ to r^c_{i+1} from matrices Z^c
  END
END
Z^c matrices remain without outliers. The total number n_c of the SOM cells, corresponding to the data from the c-th class, is renewed for each matrix Z^c.
Step 11. The indices of data centers Y^c of each class in SOM are found by Eq. (3).
Step 12. The value of the measure E_{center} is calculated by Eq. (5).

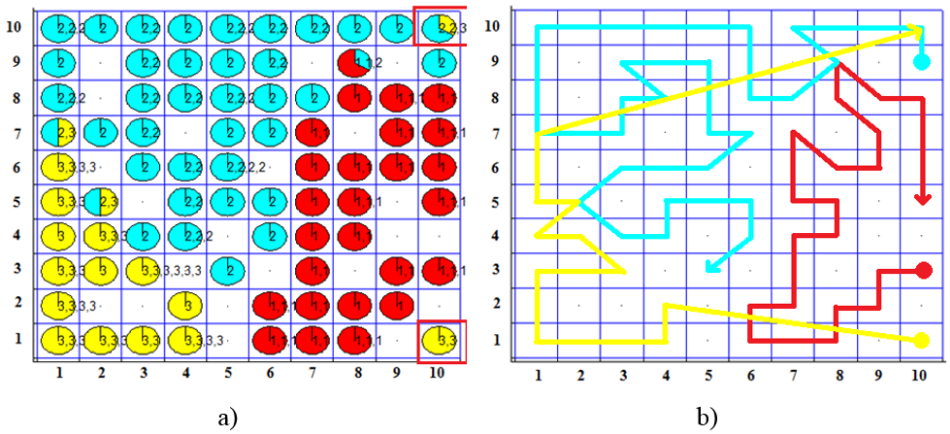

Fig. 2 *a) wine dataset in* 10×10 *SOM; b) shortest path between all the neurons in SOM of Classes 1 (red), 2 (light blue), and 3 (yellow).*

classes and fall into a cell. In addition, the class labels are given. Let us make sure whether the proposed algorithm identifies the outliers. In Step 5 of the algorithm, the shortest paths between all the same class members in SOM are found and illustrated in Fig. 2b ($q = (10 + 10)/4 = 5$). The starting points marked as circles. In Fig. 2a, the members of Class 1 and Class 2 are close from one to another, so all the distances between these members in the shortest path (Fig. 2b) are smaller than the value of the parameter q. Thus, according to the proposed algorithm (Step 8), there are no outliers among the members of Class 1 and Class 2. The majority of the members of Class 3 fall into the left bottom corner of SOM. However, one member falls into the top right corner, and two members fall into the bottom right corner (Fig. 2a). These three members can be considered as outliers. The distances between some members of Class 3 ($d^3_2 = 6.08, d^3_{43} = 9.49$) are longer than the value of the parameter q, therefore in this case, the vector $r^3 = (0, 2, 43, 44)$. According to the algorithm (Step 9), first of all, the bound for this class has to be calculated ($b^3 = 4.4$). Later (Step 10), according to the indices of neurons (cells) in which

possible outliers are determinate, the number of possible outliers are calculated. The obtained values ($r_2^3 - r_1^3 = 2, r_4^3 - r_3^3 = 1$) are smaller than the bound b^3, so the data items in these cells are outliers and must be removed before the measure E_{center} computation. It is obvious that removal of possible outliers makes the results more accurate due to that fact that the same class members form stronger clusters. Before the outlier rejection $E_{\text{center}} = 4.26$ and after rejection $E_{\text{center}} = 4.64$, which means that after outlier rejection the results improve.

A dataset of scientific papers has been chosen as textual data to illustrate the algorithm proposed for outlier detection. 39 scientific papers have been taken randomly from the Internet freely accessible databases (SpringerLink, ScienceDirect, etc.). The dataset is divided into three classes: Class 1 – papers on artificial neural networks (ANN), Class 2 – papers on optimization, and Class 3 – papers on self-organizing maps. The text documents should be converted to numerical expressions, a so-called text document matrix should be created [30]. Firstly, all the document files are converted to text files, – only the text and digits remain, figures and formulas are rejected. Afterwards, the following control factors have been selected: removing the digits from the text files, the minimal word length limit is equal to 3, the minimal word frequency is equal to 3, usage of the common word list and stemming algorithm. According to the control factors, a so-called text document dictionary is created. The document dictionary is a list of words from text files excluding the words that do not satisfy the conditions, defined by the control factors. According to the frequency of the document dictionary words in the text documents, a so-called text document matrix is created. 2487-dimensional vectors $X_1, X_2, \ldots, X_{45}$ are obtained. They have been analyzed by SOM and the results have been presented in Fig. 3.

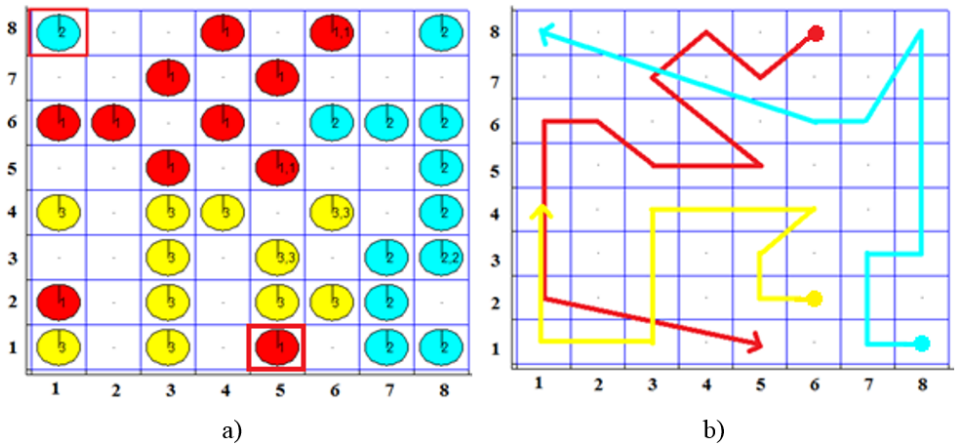

Fig. 3 *a) Dataset pf scientific papers in* 8×8 *SOM; b) shortest path between all the neurons in SOM of Classes 1 (red), 2 (light blue), and 3 (yellow)*

In Fig. 3a, there is one possible outlier of Class 2 in the top left corner of SOM. Most of Class 1 members (the scientific papers on ANN) are placed at the top of SOM, just a few members are at the bottom of the map which can also be

treated as outliers. It is necessary to estimate whether they can be considered as outliers. The highest distance ($d^1_{12} = 4.12$) is between two members of Class 1 at the bottom of SOM. This distance is longer than the parameter $q = (8 + 8)/4 = 4$ ($r^1 = (0, 12, 13)$). According to the algorithm (Step 10), the number of possible outliers in this cell ($r^1_3 - r^1_2 = 1$) is lower than the bound ($b^1 = 1.3$), so this data item is also confirmed as an outlier and has to be removed. The members of Class 3 form a strong cluster at the bottom of SOM, all the members are close to each other. The shortest path between all the same class members shows that there is no outlier in Class 3 (Fig. 3b), because all the distances are lower than $q = 4$. The calculated distance between members of Class 2 confirms that one member is an outlier, because the distance ($d^2_{12} = 5.38$) is higher than q ($r^2 = (0, 12, 13)$), and the bound is equal to 1.3 ($b^2 = 1.3$). It means that, if there is at least one member far away from the whole cluster ($r^2_3 - r^2_2 = 1$), it is considered as an outlier.

6. Conclusions

The research deals with a capability of self-organizing maps to detect data outliers. A new algorithm for detection of outliers in SOM has been proposed. The data item is considered as an outlier, if it is far away from the cluster in SOM containing the majority of data assigned to the same class as the outlier. The proposed algorithm identifies the shortest path between data items in SOM. Taking into consideration the distances in the path and some parameters, outliers are detected.

In this paper, the algorithm has been applied to the classified data and outliers are searched for each class. However, such an idea could also be used for unclassified data. In that case, all the data items would be considered as assigned to one class. The paper has investigated another aspect, – a measure that estimates the coincidence of data classes and clusters in SOM. The measure has been modified, which was proposed in our previous work that estimates a distance between the centers of different classes in SOM, by introducing some weights. The modified measure allows us to precisely estimate the coincidence of the data classes and clusters, formed in SOM, when some SOMs are compared. The simulation results and analysis have shown that the proposed algorithm allows us to find and reject outliers in SOM. Of course, before an outlier is removed it has to be confirmed as a bad data item. If an outlier is confirmed as a bad data item, it is important to remove this data item before calculating the quality of SOM. In this case, SOM results will be obtained more accurate according to the measure E_{center}. Also, the clusters of different classes will be more separated from each other and easier to be observed in SOM.

The outlier rejection depends on two parameters: 1) a bound that shows how many members of the same class can be far away from their cluster that they could be considered as outliers; 2) the distance which shows how far away some data should be from the other data of the same class that they could be considered as outliers as well. It is obvious that these two parameters could be different chosen for different datasets, so it is purposeful to analyze them more detail in the future.

References

[1] ABIDOGUN O.A., OMLIN C.W. A self-organizing maps model for outlier detection in call data from mobile telecommunication networks. *In Proceedings of the 8th Southern African Telecommunication Networks and Applications Conference (SATNAC 2004)*, SouthWestern Cape, South Africa, September, 2004, pp. 4.

[2] AGGARWAL C.C., YU P.S. Outlier detection for high dimensional data. *ACM Sig-mod Record*. 2001, 30(2), pp. 37–46, doi: 10.1145/375663.375668.

[3] AGGARWAL C.C. Supervised Outlier Detection. *Outlier Analysis*. 2012, pp. 169–198, doi: 10.1007/978-1-4614-6396-2_6.

[4] ALONSO S., SULKAVA M., PRADA M.A., DOMÍNGUEZ M., HOLLMÉN J. EnvSOM: A SOM Algorithm Conditioned on the Environment for Clustering and Visualization. In: J. Laaksonen, T. Honkela (Eds.). *Advances in Self-Organizing Maps: 8th International Workshop, WSOM 2011*, Espoo, Finland, June 13–15, 2011, Proceedings. Book Series: Lecture Notes in Computer Science. Vol. 6731. ISBN 9783642215, 2011, pp. 61–70.

[5] ASUNCION A., NEWMAN D.J. UCI Machine Learning Repository. *Irvine, CA: University of California, School of Information and Computer*, 2007. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[6] BREUNIG M. M., KRIEGEL H. P., NG R.T., SANDER J. LOF: identifying density-based local outliers. *In ACM SIGMOD on Management of Data (SIGMOD)*, Dallas, 70 Proc. Int. Conf. TX. 2000, pp. 93–104, doi: 10.1145/335191.335388.

[7] CAI Q., HE H., MAN H., QIU J. IterativeSOMSO: An iterative self-organizing map for spatial outlier detection. In: L. Zhang, J. Kwok, and B.-L. Lu (Eds.), ISNN 2010, Part I, LNCS 6063, Springer-Verlag Berlin Heidelberg 2010, pp. 325–330.

[8] CAI Q., HE H., MAN H. Spatial outlier detection based on iterative self-organizing learning model. *Neurocomputing*, 2013, vol. 117, pp. 161–172.

[9] CÁRDENAS-MONTES M. Depth-Based Outlier Detection Algorithm. In: Polycarpou M., de Carvalho A.C.P.L.F., Pan JS., Woźniak M., Quintian H., Corchado E. (Eds). *Hybrid Artificial Intelligence Systems. HAIS 2014*, vol 8480. Springer, Cham, 2014.

[10] DUAN L., XU L., LIU Y., LEE J. Cluster-based outlier detection. *Annals of Operations Research*, 2009, vol. 168, pp. 151–168.

[11] DENESHKUMAR V., SENTHAMARAIKANNAN K., MANIKANDAN M. Identification of outliers in medical diagnostic system using data mining techniques. *International Journal of Statistics and Applications*, 2014, 4(6), pp. 241–248.

[12] JING G., HAIBIN C., PANG-NING T. Semi-supervised outlier detection. *Proceedings of the 2006 ACM Symposium on Applied Computing*, 10.1145/1141277.1141421, 2006, 1. pp. 635–636.

[13] HAWKINS S., HE H., WILLIAMS G.J., BAXTER R.A. Outlier detection using replicator neural networks. Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) *DaWaK 2002. LNCS*, vol. 2454, Springer, Heidelberg, 2002, pp. 170–180.

[14] HAWKINS D.M. Identification of Outliers. *Chapman and Hall*, London – New York, 188 S, 1980.

[15] IWASAKI Y., ABE T., WADA Y., WADA K., IKEMURA T. Novel bioinformatics strategies for prediction of directional sequence changes in influenza virus genomes and for surveillance of potentially hazardous strains. *BMC Infectious Diseases*, 13(386), 2013.

[16] JOSEPH K. Traveling salesman problem – genetic algorithm. *MathWorks*, MATLAB 7, 2014. http://www.mathworks.com/matlabcentral/fileexchange/13680.

[17] KAMBER H., JAIWEI P., JIAN M. Data Mining: Concepts and Techniques (3rd ed.). *The Morgan Kaufmann Series in Data Management Systems*, ISBN 978-0-12-381479-1, 2011.

[18] KANHERE P., KHANUJA H.K. A survey on outlier detection in financial transactions. *International Journal of Computer Applications*, vol. 108 – No 17, December 2014.

[19] KASKI S., HONKELA T., LAGUS K., KOHONEN T. WEBSOM – self-organizing maps of document collections. *Neurocomputing*, 1998, 21, pp. 101–117.

[20] KOHONEN T. Self-Organizing Maps. 3rd ed., *Springer Series in Information Sciences*. Berlin: Springer-Verlag, 2001.

[21] KURASOVA O., MOLYTĖ A. Quality of quantization and visualization of vectors obtained by neural gas and self-organizing map. *Informatica*, 2011, 22(1), pp. 115–134.

[22] MADDALA G.S. Outliers. *Introduction to Econometrics (2nd ed.)*, New York: MacMillan. 1992, pp. 88–96.

[23] MARGHNY M.H., TALOBA A.I. Outlier detection using improved genetic k-means. *International Journal of Computer Applications*, 28(11), 2011.

[24] MANNING D.C., RAGHAVAN P., SCHÜTZE H. Introduction to information retrieval. *Cambridge University Press*, 2008.

[25] MISHRA M., BEHERA H. Kohonen self-organizing map with modified k-means clustering for high dimensional data set. *International Journal of Applied Information Systems*, 2(3), 2012, pp. 34–39.

[26] OSBORNE J.W., OVERBAY M. The power of outliers (and why researchers should always check for them). *Practical Assessment, Research & Evaluation*, 9(6), 2004.

[27] SCHUBERT E., ZIMEK A., KRIEGEL H.P. Fast and scalable outlier detection with approximate nearest neighbor ensembles. *DASFAA*, 2015, pp. 19–36.

[28] STEFANOVIČ P., KURASOVA O. Influence of learning rates and neighboring functions on self-organizing maps. In: J. Laaksonen, T. Honkela (Eds.). *Advances in Self-Organizing Maps: 8th International Workshop, WSOM 2011*, Espoo, Finland, June 13–15, 2011, Proceedings. Book Series: Lecture Notes in Computer Science. Vol. 6731. ISBN 9783642215, 2011, pp. 141–150.

[29] STEFANOVIČ P., KURASOVA O. Visual analysis of self-organizing maps. *Nonlinear Analysis: Modeling and Control*, 16(4), 2011, pp. 488–504.

[30] STEFANOVIČ P., KURASOVA O. Creation of text document matrices and visualization by SOM. *Information Technology and Control*, 43(1), ISSN 1392-124, 2014, pp. 37–46.

[31] STEFANOVIČ P., KURASOVA O. Investigation on learning parameters of self-organizing maps. *Baltic Journal of Modern Computing* 2(2), ISSN 2255-8942, 2014, pp. 45–55.

[32] STRICKERT M., HAMMER B. Merge SOM for temporal data. *Neurocomputing*, 2005, vol. 64, pp. 39–72.

[33] TAO Y., XIAO X., ZHOU S. Mining distance-based outliers from large databases in any metric space. *In Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, New York, NY, 2006.

[34] VOEGTLIN T. Recursive self-organizing maps. *Neural Networks*, 2002, 15, pp. 979–992.

[35] ZHANG J., FANG H. Using Self-Organizing Maps to Visualize, Filter and Cluster Multidimensional Bio-Omics Data. *Applications of Self-Organizing Maps, Dr. Magnus Johnsson (Ed.) InTech*. 2012, doi: 10.5772/51702.

[36] ZHAO J., LU C.T., KOU Y. Detecting region outliers in meteorological data. In: *Proc. of the 11th ACM-GIS*, 2003, pp. 49–55.

[37] ZIMEK A., SCHUBERT E., KRIEGEL H.P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5), 2012, pp. 363–387.