16th World Congress of the International Fuzzy Systems Association (IFSA)
9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)

Generalized SOMs with Splitting-Merging Tree-Like Structures for WWW-Document Clustering

Marian B. Gorzałczany[1] Filip Rudziński[2] Jakub Piekoszewski[3]

[1,2,3]Department of Electrical and Computer Engineering, Kielce University of Technology,
Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland

Abstract

This paper presents our clustering technique based on generalized SOMs with evolving splitting-merging tree-like structures and its application to complex clustering problems including some benchmark data sets and, first of all, WWW-document clustering. Our approach that works in a fully unsupervised way (i.e., without the pre-defined cluster number and using unlabelled data), automatically detects the number of clusters and generates multi-prototypes for them. The collection of 548 abstracts of technical reports as well as its 476-element subset, both available at WWW server of the Department of Computer Science, University of Rochester, USA (www.cs.rochester.edu/trs) are the subjects of clustering. A comparative analysis with five alternative clustering techniques is also carried out. The reported results prove that our approach is a powerful tool (that outperforms several alternative approaches) for complex cluster-analysis tasks including the problems of WWW-document clustering.

Keywords: WWW-document clustering, generalized SOMs with tree-like structures, cluster analysis, unsupervised learning

1. Introduction

Significant advances in information and communication technologies and the dynamic growth of World-Wide-Web resources make more and more important the problems of helping users to efficiently access relevant information and to organize it in intelligible way. Among the most widely available WWW resources are text and hypertext documents. For this reason, WWW-text-document processing techniques, including thematic WWW-document clustering methods, play an important role in mining the Web [1]. For a collection of WWW documents, the task of document clustering, in general, is to group particular documents together in such a way that the items within each cluster are as "similar" as possible to each other and as "dissimilar" as possible from those of the other clusters.

This paper presents a clustering method that employs generalized self-organizing maps (SOMs) with evolving splitting-merging tree-like structures (cf. [2]) and its application to clustering of selected collections of WWW-documents. In general, original SOMs [3] are used to visually display topological structures of high dimensional data in lower (usually two-dimensional) space rather than for clustering, i.e., partitioning of these data into groups [4]. However, the proposed generalized SOMs with structure splitting and merging mechanism are equipped with both data-dimensionality reduction and data-segmentation abilities. It is worth emphasizing that our approach works in a fully unsupervised way, i.e., without a predefined number of clusters and using unlabelled data. First, the clustering process using the proposed generalized SOMs is presented and illustrated by means of two benchmark data sets. Then, a Vector-Space-Model representation of WWW documents and some approaches to its dimensionality reduction are outlined. In turn, the application of our approach to clustering of the collection of 548 abstracts of technical reports available at the WWW site of the Department of Computer Science, University of Rochester, USA (www.cs.rochester.edu/trs) is presented. Finally, a comparative analysis with several alternative text-clustering techniques is also carried out (for this purpose, a subset of 476 abstracts of the aforementioned original collection of abstracts is also considered).

2. Generalized SOMs with Evolving Splitting-Merging Tree-Like Structures for Data Clustering

Consider, first, the conventional SOM with one-dimensional neighborhood (SOM with 1DN), i.e., the neuron chain. Assume that the network has n inputs $x_1, x_2, \ldots, x_n$ and consists of m neurons; their outputs are $y_1, y_2, \ldots, y_m$, where $y_j = \sum_{i=1}^{n} w_{ji} x_i$, $j = 1, 2, \ldots, m$ and w_{ji} are weights connecting the i-th input of the network with the output of the j-th neuron. Using vector notation ($\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, $\boldsymbol{w}_j = (w_{j1}, w_{j2}, \ldots, w_{jn})^T$), $y_j = \boldsymbol{w}_j^T \boldsymbol{x}$. The learning data consists of L input vectors $\boldsymbol{x}_l$ ($l = 1, 2, \ldots, L$). In the first stage of any Winner-Takes-Most (WTM) learning algorithm that can be used in the learning process of the con-

sidered network, the neuron $j_{\boldsymbol{x}}$, which wins in competition of neurons when the learning vector $\boldsymbol{x}_l$ is presented to the network must be determined. Assuming that the normalization of learning vectors is performed, the winning neuron $j_{\boldsymbol{x}}$ is selected in the following way:

$$d(\boldsymbol{x}_l, \boldsymbol{w}_{j_{\boldsymbol{x}}}) = \min_{j=1,2,\ldots,m} d(\boldsymbol{x}_l, \boldsymbol{w}_j), \tag{1}$$

where $d(\boldsymbol{x}_l, \boldsymbol{w}_{j_{\boldsymbol{x}}})$ is a distance measure between $\boldsymbol{x}_l$ and $\boldsymbol{w}_j$. Different measures are more or less suitable in different clustering tasks [5], [6]. As far as text-document is concerned, most often a distance measure d_{cos} based on the cosine similarity function S_{cos} (frequently used for determining the similarity of text documents) is applied:

$$\begin{aligned} d_{cos}(\boldsymbol{x}_l, \boldsymbol{w}_j) &= 1 - S_{cos}(\boldsymbol{x}_l, \boldsymbol{w}_j) = \\ &= 1 - \frac{\boldsymbol{x}_l^T \boldsymbol{w}_j}{\|\boldsymbol{x}_l\|_E \|\boldsymbol{w}_j\|_E} = \\ &= 1 - \frac{\sum_{i=1}^{n} (x_{li} w_{ji})}{\sqrt{\sum_{i=1}^{n} x_{li}^2 \sum_{i=1}^{n} w_{ji}^2}}, \end{aligned} \tag{2}$$

where $\|.\|_E$ is the Euclidean norm. In our experiments presented in Section 4 and regarding WWW document clustering, d_{cos} will be used. However, in many other applications (including two benchmark data clustering presented at the end of this section), the Euclidean distance measure d_E:

$$d_E(\boldsymbol{x}_l, \boldsymbol{w}_j) = \|\boldsymbol{x}_l - \boldsymbol{w}_j\|_E \tag{3}$$

is used. The WTM learning rule has the following form:

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \eta_j(k) N(j, j_{\boldsymbol{x}}, k)[\boldsymbol{x}(k) - \boldsymbol{w}_j(k)], \tag{4}$$

where k is the iteration number, $\eta_j(k)$ is the learning coefficient, and $N(j, j_{\boldsymbol{x}}, k)$ is the neighborhood function of the $j_{\boldsymbol{x}}$-th winning neuron. Most often the Gaussian-type neighborhood functions are used, i.e.:

$$N(j, j_{\boldsymbol{x}}, k) = e^{-\frac{d_{tpl}^2(j, j_{\boldsymbol{x}})}{2\lambda^2(k)}} \tag{5}$$

where $\lambda(k)$ is the neighborhood radius and $d_{tpl}(j, j_{\boldsymbol{x}})$ - the topological distance between the $j_{\boldsymbol{x}}$-th and j-th neurons. In the case of the conventional SOM with 1DN, $d_{tpl}(j, j_{\boldsymbol{x}}) = |j - j_{\boldsymbol{x}}|$. However, when our mechanisms (presented below) for splitting and merging of the network structure are implemented, the conventional SOM with 1DN evolves toward a tree-like structure. As a result of that, the neighborhood of a given neuron in such a tree-like topology of our generalized SOMs is defined along all the arcs emanating from that neuron as shown in Fig. 1. Those arcs are the pieces of the conventional SOM with 1DN. Therefore, $d_{tpl}(j, j_{\boldsymbol{x}}) = 1$ for all j-th neurons being direct neighbors of the $j_{\boldsymbol{x}}$-th one as illustrated in Fig. 1. In turn, $d_{tpl}(j, j_{\boldsymbol{x}}) = 2$ for all j-th neurons being second along all paths starting at the $j_{\boldsymbol{x}}$-th one (see Fig. 1), etc.

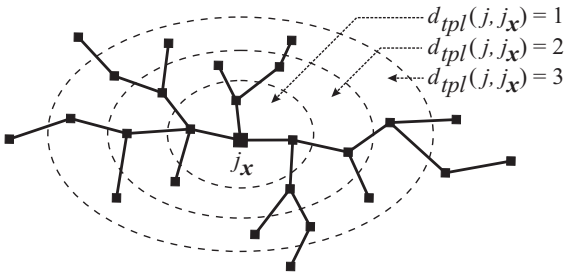

Figure 1: Illustration of neighborhood of the $j_{\boldsymbol{x}}$-th neuron [2]

The essence of the proposed generalization consists in introducing, in the learning phase, three mechanisms: (i) automatic adjustment of the number of neurons in the network by removing low-active neurons and adding new neurons in areas of high neuronal activity, starting from arbitrarily selected small (e.g., equal to 2) number of neurons, (ii) automatic disconnection of the tree-like structure into subnetworks, and (iii) automatic reconnection of some of the subnetworks preserving the no-loop spanning-tree properties. Such a generalized SOM is able to detect data clusters of various shapes and densities by assigning a single disconnected subnetwork to each cluster. Thus, the number of automatically generated subnetworks is equal to the number of clusters. Additionally, the collection of neurons in a given subnetwork is a multi-prototype of the corresponding cluster. Such prototypes can be directly used in clustering and classification tasks by employing the well-known nearest multi-prototype method [7], [8]. The proposed approach is a generalization of our earlier solutions to automatic determination of the cluster numbers and cluster prototypes in data sets [9], [10], [11], [12].

The implementation of the above-mentioned mechanisms is carried out by the activation of four operations after each learning epoch (provided that the required conditions are fulfilled).

Operation 1 (the removal of single low-active neurons): The neuron no. j_r is removed from the network (preserving the network continuity - see [2] for details) if its activity - measured by the number of its wins win_{j_r} - is below an assumed level win_{min}, i.e., $win_{j_r} < win_{min}$. win_{min} is experimentally selected parameter (usually, $win_{min} \in \{2, 3, 4\}$).

Operation 2 (the disconnection of the network (subnetwork) into two subnetworks): The disconnection of two neighboring neurons j_1 and j_2 takes place if the following condition is fulfilled: $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) > d_{coef} d_{avr}$ where $d_{avr} = \frac{1}{P} \sum_{p=1}^{P} d_p$ is the average distance between two neighboring neurons for all pairs p, $p = 1, 2, \ldots, P$, of such neurons (d, d_{avr}, and d_p are either cosine or Euclidean distance measures according to the area of applications). d_{coef} is experimentally selected parameter (a distance coefficient) governing the disconnection operation (usually, $d_{coef} \in [3, 4]$). Possible

very short (single- or two-neuron) subnetworks are removed from the system since they cannot be reconnected by *Operation 4* (see below).

Operation 3 (the insertion of additional neurons into the neighborhood of high-active neurons in order to take over some of their activities). *Case 3a*: A new neuron, labelled as (*new*), is inserted between two neighboring and high-active neurons j_1 and j_2 (i.e., their numbers of wins win_{j_1} and win_{j_2} are above an assumed level win_{max}: $win_{j_1}, win_{j_2} > win_{max}$). win_{max} is experimentally selected parameter (usually $win_{max} \in \{2, \ldots, 5\}$ and $win_{max} \geq win_{min}$, where win_{min} is defined in *Operation 1*). The weight vector $\boldsymbol{w}_{(new)}$ of the new neuron is calculated as follows: $\boldsymbol{w}_{(new)} = \frac{\boldsymbol{w}_{j_1} + \boldsymbol{w}_{j_2}}{2}$. *Case 3b*: A new neuron (*new*) is inserted in the neighborhood of high-active neuron j_1 surrounded by less-active neighbors (i.e., $win_{j_1} > win_{max}$ and $win_j \leqslant win_{max}$ for j such that $d_{tpl}(j, j_1) = 1$). The weight vector $\boldsymbol{w}_{(new)} = [w_{(new)1}, w_{(new)2}, \ldots, w_{(new)n}]^T$ is calculated as follows: $w_{(new)i} = w_{j_1 i}(1 + \xi_i)$, $i = 1, 2, \ldots, n$, where ξ_i is a random number from the interval $[-0.01, 0.01]$ (see [2] for details).

Operation 4 (the reconnection of two selected subnetworks): Two subnetworks S_1 and S_2 are reconnected by introducing topological connection between neurons j_1 and j_2 ($j_1 \in S_1$, $j_2 \in S_2$) after fulfilling condition $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) < d_{coef} \frac{d_{avr_{S_1}} + d_{avr_{S_2}}}{2}$. $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2})$ and d_{coef} are the same as in *Operation 2*. $d_{avr_{S_1}}$ and $d_{avr_{S_2}}$ are calculated for subnetworks S_1 and S_2, respectively, in the same way as d_{avr} is calculated in *Operation 2* for the considered network.

Following Kohonen's comments [3], the learning parameters are selected mainly in an experimental way. The learning coefficient $\eta(k)$ and the neighborhood radius $\lambda(k)$ should be some monotonically decreasing functions of time ($\lambda(k)$ can also be constant in time) [3]. Taking that into account, the learning parameters in our experiments are defined as follows: $\eta_j(k) = \eta(k)$ of (4) is linearly decreasing over the learning horizon (which includes 1000 epochs) from $7 \cdot 10^{-4}$ to 10^{-6}, $\lambda(k) = \lambda$ of (5) is equal to 2, the initial number of neurons in the network is equal to 2, $win_{min} = 3$, $win_{max} = 5$, and $d_{coef} = 4$. It is worth emphasing that the same set of experimentally selected parameters that govern the structure splitting and merging mechanisms (i.e., win_{min}, win_{max}, and d_{coef}) gives exellent results in quite different (in terms of data dimensionality and the distance-measure definition) applications such as some benchmark data sets (see below in this section) and document sets (see Section 4). Thus, the sensitivity of our approach to the changes of those parameters is low.

In order to illustrate the operation of our clustering technique and to evaluate its performance, the clustering of two benchmark data sets from the so-called Fundamental Clustering Problem Suite (FCPS) [13] will now be carried out. The first benchmark set, referred to as *GolfBall* data, consists of points that are equidistant on the surface of a sphere. Thus, no cluster structure exists in that set or, equivalently, one big cluster covering all the data can be considered. It is a hard-to-pass test for many clustering algorithms, especially those generating a pre-defined number of cluster. The second benchmark set (*Chainlink* data) contains two clusters that are not separable by hyperplanes.

Figs. 2 and 3 illustrate the operation of our clustering approach for both benchmark data sets. Parts a) of both figures represent the data, parts b), c), d), and e) show the evolution of the tree-like structures of our generalized SOMs in data sets at different stages of the learning process, and parts f) and g) present the plots of the number of neurons (f) and the number of subnetworks (equal to the number of detected clusters)(g) vs. learning epoch number. Our approach automatically increases the number of neurons in particular networks (starting from the initial numbers of two neurons) and detects the correct numbers of data clusters in both sets by disconnecting the tree-like structures of the generalized SOMs into appropriate number of subnetworks. In particular, it detects one big cluster covering all the data in *GolfBall* set. It confirms that no cluster structure exists in that data set.

3. An Outline of WWW-Document Vector Space Model

The Vector Space Model (*VSM*) [1], [14], [15], [16] of the collection of L WWW documents consists of L vectors $\boldsymbol{x}_l = (x_{l1}, x_{l2}, \ldots, x_{ln})^T$, $l = 1, 2, \ldots, L$ that describe particular documents. The component x_{li} ($i = 1, 2, \ldots, n$) of $\boldsymbol{x}_l$ represents a relationship between the i-th key word or term and the l-th document from the collection. There are various schemes for measuring that relationship (often referred to as term weighting). Among them three approaches are the most popular: a) binary term weighting: $x_{li} = 1$ when the i-th term occurs in the l-th document and $x_{li} = 0$ otherwise, b) term frequency weighting (or, *tf*-weighting, for short): $x_{li} = tf_{li}$ where tf_{li} is equal to the number of occurrences of the i-th term in the l-th document, and c) term frequency - inverse document frequency weighting (or, *tf-idf*-weighting, for short): $x_{li} = tf_{li} \log(L/df_i)$, where tf_{li} is the term frequency as in *tf*-weighting, df_i denotes the number of documents in which the i-th term appears, and L is the total number of documents in the collection. In our experiments *tf*-weighting will be used. Once the way of determining x_{li} is selected, the Vector Space Model can be formulated in a matrix form:

$$VSM_{(n \times L)} = \boldsymbol{X}_{(n \times L)} = [\boldsymbol{x}_l]_{l=1,2,\ldots,L} = \\ = [(x_{l1}, x_{l2}, \ldots, x_{ln})^T]_{l=1,2,\ldots,L} \quad (6)$$

where the $(n \times L)$ index represents *VSM*'s dimensionality.

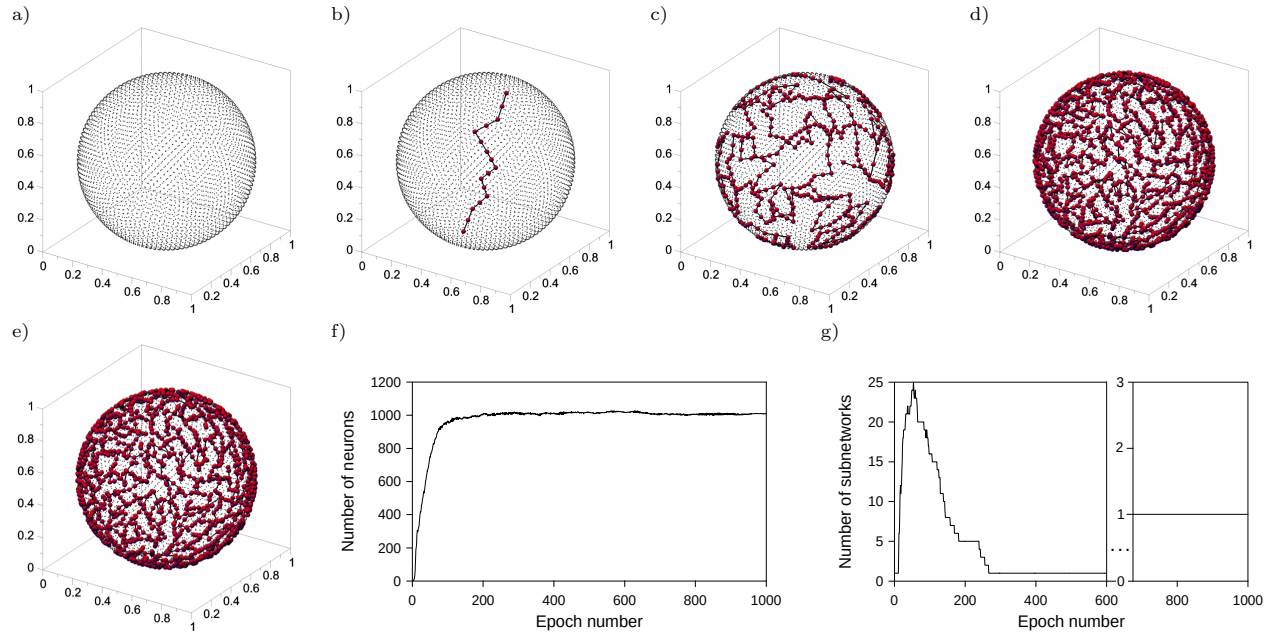

Figure 2: *GolfBall* data set (a) and the evolution of our generalized SOM's structure in it in learning epochs: b) no. 5, c) no. 10, d) no. 50, and e) no. 1000 (end of learning), as well as plots of the number of neurons (f) and the number of subnetworks (clusters) (g) vs. epoch number

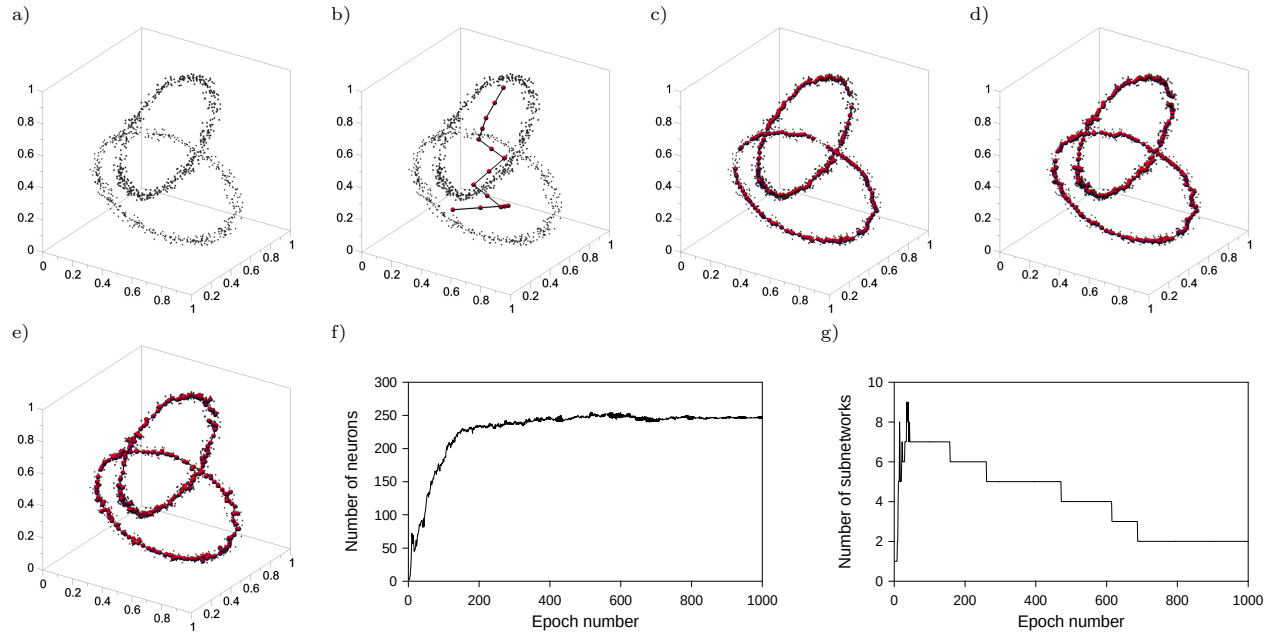

Figure 3: *Chainlink* data set (a) and the evolution of our generalized SOM's structure in it in learning epochs: b) no. 5, c) no. 10, d) no. 50, and e) no. 1000 (end of learning), as well as plots of the number of neurons (f) and the number of subnetworks (clusters) (g) vs. epoch number

The reduction of the *VSM* dimensionality is an important problem from the point of view of practical usage of *VSM*s. Two main categories of *VSM*-dimensionality-reduction techniques are considered [17]: a) feature selection methods and b) feature transformation methods. Feature selection consists in sorting terms and then eliminating some of them on the basis of some numerical measures computed from the considered collection of documents. Such a process is preceded by filtering, stemming, and stop-word removal preprocessing operations that also contribute to *VSM*-dimensionality reduction.

The filtering (and tokenization) operation removes special characters, such as %, #, $, etc., from the original text as well as identifies word- and sentence boundaries in it. As a result of that, initial $VSM_{(n_{ini} \times L)}$ is obtained, where n_{ini} is the number of different words isolated from all documents.

The stemming operation replaces all the words in initial model by their respective stems (a stem is

Table 1: The dimensionality reduction of the initial *VSM*s for CSTR-548 and CSTR-476 collections of abstracts

VSM	Dimensionality of *VSM* for abstract collection:			
	CSTR-548		CSTR-476	
$VSM_{(n_{ini}\times L)}$	$(n_{ini}\times L) = (7438\times 548)$ [47382]		$(n_{ini}\times L) = (6752\times 476)$ [41072]	
$VSM_{(n_{stem}\times L)}$	$(n_{stem}\times L) = (4896\times 548)$ [44201]		$(n_{stem}\times L) = (4438\times 476)$ [38307]	
$VSM_{(n_{stpl}\times L)}$	$(n_{stpl}\times L) = (4574\times 548)$ [30644]		$(n_{stpl}\times L) = (4119\times 476)$ [26525]	
$VSM_{(n_{fin}\times L)}$	CSTR-548"Small" ($q_{tres} = 20$)	CSTR-548"Large" ($q_{tres} = 2$)	CSTR-476"Small" ($q_{tres} = 20$)	CSTR-476"Large" ($q_{tres} = 2$)
	$(n_{fin}\times L) =$ (405×548) [18096]	$(n_{fin}\times L) =$ (2396×548) [28466]	$(n_{fin}\times L) =$ (342×476) [14805]	$(n_{fin}\times L) =$ (2217×476) [24623]

a portion of a word left after removing its suffixes and prefixes). As a result of that, $VSM_{(n_{stem}\times L)}$ is obtained, where $n_{stem} < n_{ini}$.

The stop-word removal operation (i.e., removing words from a so-called stop list) eliminates from the model words that on their own do not have identifiable meanings and therefore are of little use in various text processing tasks. As a result of that, $VSM_{(n_{stpl}\times L)}$ is obtained, where $n_{stpl} < n_{stem}$.

Feature selection methods usually operate on term quality q_i, $i = 1, 2, \ldots, n_{stpl}$ defined for each term occurring in the latest *VSM*. Terms characterized by $q_i < q_{tres}$, where q_{tres} is a pre-defined threshold value are removed from the model. In our experiments, the document-frequency-based is used to determine q_i, i.e., $q_i = df_i$, where df_i is the number of documents in which the i-th term occurs. As a result of that, final $VSM_{(n_{fin}\times L)}$ is obtained, where $n_{fin} < n_{stpl}$.

4. Application to WWW-Document Clustering and Comparative Analysis

The performance of our generalized SOMs will now be verified in the real-life WWW-document clustering problem, i.e., the clustering of the collection of 548 abstracts of technical reports available at WWW server of the Department of Computer Science, University of Rochester, USA (www.cs.rochester.edu/trs); henceforward, the collection will be referred to as CSTR-548 (CSTR stands for Computer Science Technical Reports). The number of classes (equal to 4: *AI* (*Natural Language Processing*), *RV* (*Robotics-Vision*), *Systems*, and *Theory*) and the class assignments are known in the considered document set. Therefore, a direct verification of the obtained results is possible. However, the knowledge on the class number and the class assignments by no means will be used by our clustering algorithm (it works in a fully unsupervised way).

In order to extend a comparative analysis, also a subset of 476 abstracts (referred to as CSTR-476) of the aforementioned original collection CSTR-548 is considered. Collection CSTR-476 contains the abstracts of technical reports that were published until the end of 2002, whereas CSTR-548 covers the abstracts published until June 2005. The clustering of CSTR-476 by means of some alternative techniques is presented in [18].

The results of the dimensionality reduction of the initial *VSM*s are collected in Table 1 using the notation introduced in Section 3 (additionally, in square brackets, the overall numbers of occurrences of all terms in all documents of a given collection are presented). Two final numerical models (identified in Table 1 as "Small" and "Large" sets) are generated for CSTR-548 and CSTR-476. For this purpose, two values of threshold parameter q_{tres} are considered: $q_{tres} = 20$ - to get models of more reduced dimensionalities ("Small"-type sets) and $q_{tres} = 2$ - to get models of higher dimensionalities but also of higher accuracies ("Large"-type sets). It is worth noticing that $q_{tres} = 2$ results in removing from the model all the terms that occur in only one document of the collection; therefore, they do not contribute to the clustering process.

Figs. 4 and 5 illustrate the progress of the learning and, thus, the clustering process for both numerical models of CSTR-548 abstract collection. Both systems adjust the overall numbers of neurons in their networks (Figs. 4a and 5a) and the number of disconnected subnetworks (equal to the number of detected clusters; Figs. 4b and 5b). Finaly, four clusters are found in both sets.

Since the class number and the class asignments in the original collection of abstracts are known, a direct verification of the obtained results is also possible as shown in Table 2. It can be seen, in general, that *Systems* and *Theory* groups are much different from each other and different from *AI* and *VR* groups. In turn, *AI* and *VR* are relatively similar to each other, however, according to more accurate model (CSTR-548 "Large") many more *AI*-abstracts are misclassified as *VR*s than vice versa, etc. Moreover, in Table 3 the results of comparative analysis with three alterna-

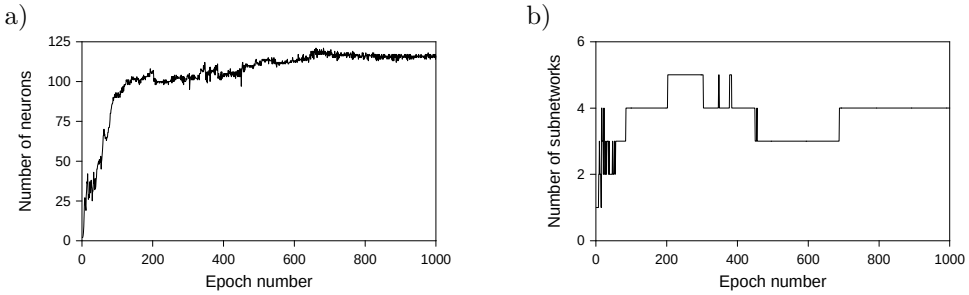

Figure 4: Plots of the number of neurons (a) and the number of subnetworks (b) vs. epoch number for CSTR-548"Small"

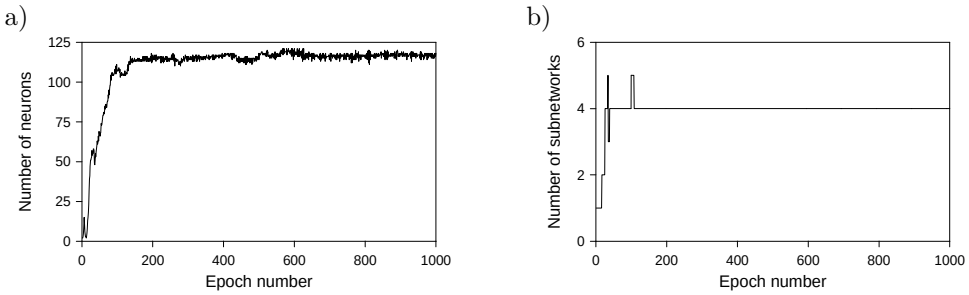

Figure 5: Plots of the number of neurons (a) and the number of subnetworks (b) vs. epoch number for CSTR-548"Large"

Table 2: Clustering results for CSTR-548"Small" (a) and CSTR-548"Large" (b)

Class label		Number of decisions for subnetwork labelled:				Number of correct decisions	Number of wrong decisions	Percentage of correct decisions
		AI	*RV*	*Systems*	*Theory*			
a)	*AI*	82	11	18	1	82	30	73.21%
	RV	26	51	10	2	51	38	57.30%
	Systems	0	3	194	0	194	3	98.48%
	Theory	2	2	14	132	132	18	88.00%
	ALL	110	67	236	135	459	89	**83.76%**
b)	*AI*	61	47	3	1	61	51	54.46%
	RV	6	82	1	0	82	7	92.13%
	Systems	0	1	195	1	195	2	98.98%
	Theory	1	4	5	140	140	10	93.33%
	ALL	68	134	204	142	478	70	**87.23%**

tive approaches: the EM (Expectation Maximization) method, the FFTA (Farthest First Traversal Algorithm), and the well-known k-means algorithm, applied to both considered data sets are presented. The WEKA (Waikato Environment for Knowledge Analysis) application that implements the EM, FFTA, and k-means algorithms has been used for that purpose. The WEKA application as well as details on the clustering techniques can be found on WWW site of the University of Waikato, New Zealand (www.cs.waikato.ac.nz/ml/weka).

As already mentioned, in order to extend the comparative-analysis aspects of this paper, additionally, the clustering of CSTR-476 subset of original abstract collection CSTR-548 is carried out - see Figs. 6 and 7 as well as Table 4. Two numerical models of CSTR-476, i.e., "Small" and "Large" data

Table 3: Results of comparative analysis for CSTR-548 numerical models

Clustering method	Percentage of correct decisions	
	CSTR-548"Small"	CSTR-548"Large"
Our	83.76%	87.23%
EM	62.23%	51.09%
FFTA	37.77%	36.68%
k-means	65.33%	36.68%

set (see Table 1) are subject to clustering. As shown in Table 5, this time the operation of our clustering technique is compared with five alternative approaches, including additionally the EB (Entropy-Based clustering) method and hierarchical clustering technique available from CLUTO software package (www-users.cs.umn.edu/~karypis/cluto). The

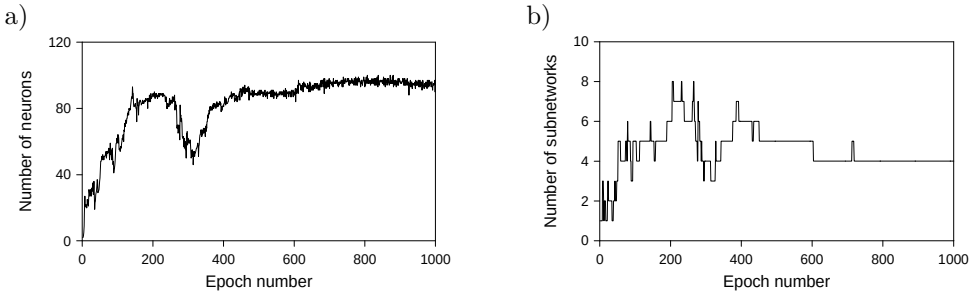

Figure 6: Plots of the number of neurons (a) and the number of subnetworks (b) vs. epoch number for CSTR-476"Small"

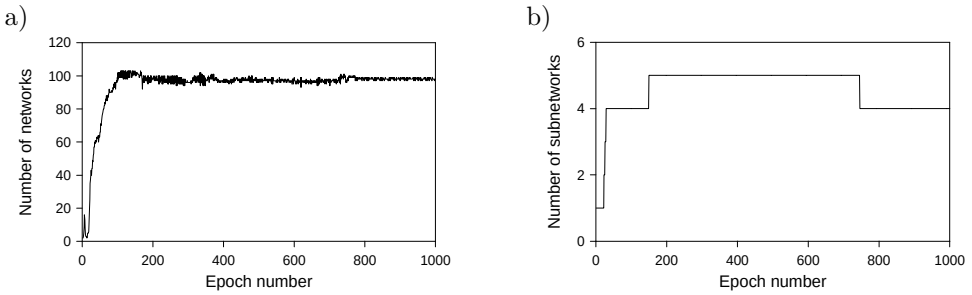

Figure 7: Plots of the number of neurons (a) and the number of subnetworks (b) vs. epoch number for CSTR-476"Large"

Table 4: Clustering results for CSTR-476"Small" (a) and CSTR-476"Large" (b)

Class label		Number of decisions for subnetwork labelled:				Number of correct decisions	Number of wrong decisions	Percentage of correct decisions
		AI	*RV*	*Systems*	*Theory*			
a)	*AI*	43	50	5	3	43	58	42.57%
	RV	2	65	1	3	65	6	91.55%
	Systems	4	5	168	1	168	10	94.38%
	Theory	0	4	11	111	111	15	88.10%
	ALL	49	124	185	118	387	89	**81.30%**
b)	*AI*	59	35	6	1	59	42	58.42%
	RV	0	65	4	2	65	6	91.55%
	Systems	2	10	166	0	166	12	93.26%
	Theory	0	8	4	114	114	12	90.48%
	ALL	61	118	180	117	404	72	**84.87%**

results of applying EB and CLUTO-based clustering techniques to CSTR-476 abstract collection are reported in [18] and repeated in Part II of Table 5. The results presented in Tables 3 and 5 prove that our approach significantly outperforms the considered alternative clustering techniques.

5. Conclusions

In this paper our clustering technique based on the generalized SOMs with evolving splitting-merging tree-like structures is presented and applied to complex clustering problems including WWW-document clustering. Our approach that works in a fully unsupervised way (i.e., without the pre-defined cluster number and using unlabelled data), automatically detects the number of clusters (equal to the number of disconnected subnetworks) and generates multi-prototypes for them (represented by neurons in particular subnetworks). It is achieved by the implementation of automatic adjustment of the number of neurons in the network and the disconnection and reconnection mechanisms of the network tree-like structures during the learning process.

Two benchmark data sets coming the Fundamental Clustering Problem Suite [13] and the collection of 548 abstracts of technical reports as well as its 476-element subset, both available at WWW server of the Department of Computer Science, University of Rochester, USA (www.cs.rochester.edu/trs) were the subjects of clustering.

The results reported in this paper prove that our approach is a powerful tool for complex cluster-analysis tasks including high-dimensional problems

Table 5: Results of comparative analysis for CSTR-476 numerical models
Part I:

Clustering method	Percentage of correct decisions	
	CSTR-476"Small"	CSTR-476"Large"
Our	81.30%	84.87%
EM	69.96%	50.00%
FFTA	37.82%	38.03%
k-means	69.75%	38.45%

Part II:

Clustering method	CSTR-476 abstract collection	
	Dimensionality of *VSM*	Percentage of correct decisions
EB	not available	∼73.9%
CLUTO	not available	∼68.8%

of WWW-document clustering and provides much better results that many alternative techniques in this field.

References

[1] S. Chakrabarti. *Mining the Web: Analysis of Hypertext and Semi Structured Data*. Morgan Kaufmann Publishers, San Francisco, August 2002.

[2] M. B. Gorzałczany, J. Piekoszewski, and F. Rudziński. Generalized tree-like self-organizing neural networks with dynamically defined neighborhood for cluster analysis. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, editors, *Artificial Intelligence and Soft Computing - ICAISC 2014*, volume 8468 of *Lecture Notes in Computer Science*, pages 725–737. Springer-Verlag, Berlin, 2014.

[3] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3 edition, 2001.

[4] N. R. Pal, J. C. Bezdek, and E. C.-K. Tsao. Generalized clustering networks and Kohonen's self-organizing scheme. *IEEE Transactions on Neural Networks*, 4(4):549–557, 1993.

[5] W. Pedrycz. *Knowledge-Based Clustering, From Data to Information Granules*. J. Wiley, Hoboken, 2005.

[6] M. W. Berry, editor. *Survey of Text Mining*. Springer, New York, 2004.

[7] J. C. Bezdek, T. R. Reichherzer, G. S. Lim, and Y. Attikiouzel. Multiple-prototype classifier design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 28(1):67–79, 1998.

[8] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer Science & Business Media, New York, 2005.

[9] M. B. Gorzałczany and F. Rudziński. Cluster analysis via dynamic self-organizing neural networks. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, editors, *Artificial Intelligence and Soft Computing - ICAISC 2006*, volume 4029 of *Lecture Notes in Computer Science*, pages 593–602. Springer-Verlag, Berlin, 2006.

[10] M. B. Gorzałczany and F. Rudziński. WWW-newsgroup-document clustering by means of dynamic self-organizing neural networks. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, editors, *Artificial Intelligence and Soft Computing - ICAISC 2008*, volume 5097 of *Lecture Notes in Computer Science*, pages 40–51. Springer-Verlag, Berlin, 2008.

[11] M. B. Gorzałczany and F. Rudziński. Application of genetic algorithms and Kohonen networks to cluster analysis. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. H. Siekmann, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 556–561. Springer-Verlag, Berlin, 2004.

[12] M. B. Gorzałczany and F. Rudziński. Modified Kohonen networks for complex cluster-analysis problems. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. H. Siekmann, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 562–567. Springer-Verlag, Berlin, 2004.

[13] A. Ultsch. Clustering with SOM: U*C. In *Proceedings of the Workshop on Self-Organizing Maps*, pages 75–82, Paris, France, 2005.

[14] J. Franke, G. Nakhaeizadeh, and I. Renz, editors. *Text Mining: Theoretical Aspects and Applications*. Physica-Verlag, Heidelberg, New York, 2003.

[15] S. M. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, New York, 2004.

[16] A Zanasi, editor. *Text Mining and Its Applications to Intelligence, CRM and Knowledge Management*. WIT Press, Southampton, 2005.

[17] B. Tang, M. Shepherd, E. Milos, and M. I. Heywood. Comparing and combining dimension reduction techniques for efficient text clustering. In *Proceedings of the International Workshop on Feature Selection and Data Mining*, pages 292–296, Newport Beach, 2005.

[18] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In *Proceedings of the 21st IEEE International Conference on Machine Learning*, pages 536–543, Banff, Alberta, 2004.