International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:5, 2010

TOSOM: A Topic-Oriented Self-Organizing Map for Text Organization

Hsin-Chang Yang, Chung-Hong Lee, and Kuo-Lung Ke

Abstract—The self-organizing map (SOM) model is a well-known neural network model with wide spread of applications. The main characteristics of SOM are two-fold, namely dimension reduction and topology preservation. Using SOM, a high-dimensional data space will be mapped to some low-dimensional space. Meanwhile, the topological relations among data will be preserved. With such characteristics, the SOM was usually applied on data clustering and visualization tasks. However, the SOM has main disadvantage of the need to know the number and structure of neurons prior to training, which are difficult to be determined. Several schemes have been proposed to tackle such deficiency. Examples are growing/expandable SOM, hierarchical SOM, and growing hierarchical SOM. These schemes could dynamically expand the map, even generate hierarchical maps, during training. Encouraging results were reported. Basically, these schemes adapt the size and structure of the map according to the distribution of training data. That is, they are data-driven or data-oriented SOM schemes. In this work, a topic-oriented SOM scheme which is suitable for document clustering and organization will be developed. The proposed SOM will automatically adapt the number as well as the structure of the map according to identified topics. Unlike other data-oriented SOMs, our approach expands the map and generates the hierarchies both according to the topics and their characteristics of the neurons. The preliminary experiments give promising result and demonstrate the plausibility of the method.

Keywords—Self-Organizing Map, Topic Identification, Learning Algorithm, Text Clustering.

I. Introduction

MANY artificial neural network models had been proposed to model the learning mechanism of brain. One of the widely applied models is the self-organizing map (SOM) model [1]. The SOM learns from high dimensional data and maps them on a low, often 2, dimensional map in a topology-preserving manner. That is, data close together in high-dimensional space will also be close in the mapped low-dimensional space. Such abilities make the SOM being widely applied in data visualization and clustering tasks. Up to now, there are more than 7,700 papers published regarding the SOM, according to the bibliography[1] compiled by the original SOM team [2].

Although the SOM is easy and effective in data clustering, there are three main disadvantages in its learning algorithm. First, the training time of the SOM is often long since the input data is often high-dimensional and the SOM requires repeated training over input data. Second, the size of the map is fixed. The number of neurons in the map should be determined a priori. However, there are no gold standard for determining this number. Third, the original SOM can reveal only lateral correlations among data, but not hierarchical ones. To remedy these deficiencies, several schemes have been proposed. To overcome the fixed structure problem, it is often to make the map expandable during training. That is, the map contains few neurons initially and expands in later training stages according to the distributions of data. There are two major approaches to conquer the lack of hierarchical relationships. The first is to apply agglomerative hierarchical clustering or divisive hierarchical clustering processes on the two-dimensional map to generate hierarchies. The second is to retrain each cluster in the map using a lower-level map and obtain the hierarchical structure. These approaches on map expansion or hierarchy generation all relies on the characteristics of data to decide when and how to expand the map. Since the SOM will cluster together similar data, many similar data will be clustered into few neurons. Thus we can decide whether to expand the map or not by detecting the distribution of data on the map. Here we may call such approaches the data-driven or data-oriented approaches. Although data-oriented approaches may effectively generate adequate number of clusters or even hierarchies, they provide no insight into the meaning of data, let along guiding the training process.

The SOM is often used for text document clustering and categorization. Text categorization concerns of classifying documents into some categories according to their contents, characteristics, and properties. When documents are properly categorized, documents in a cluster should have a common theme. Oppositely, we need some well-defined category structures to perform text categorization. Such structures are often manually constructed and are often inconsistent. Automatic schemes for generating categorization structure are needed in categorizing large, dynamic repositories.

In this work, we propose a novel self-organizing map algorithm based on topics of documents rather than documents themselves. The core difference between our method and traditional data-oriented SOMs lies on the intervention of topics in the learning process. The topics of each cluster are identified continuously during training process. These topics are then served as bases of expanding maps and hierarchies. Using such higher-level knowledge during training should provide deeper insight of the underlying documents and better guidance of the training. Besides, our method can naturally generate a categorization structure with identified themes and

H.-C. Yang is with the Department of Information Management, National University of Kaohsiung, Kaohsiung 811, Taiwan (corresponding author, phone: +886-7-5919512; e-mail: yanghc@nuk.edu.tw).

C.-H. Lee is with the Department of Electrical Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan (e-mail: leechung@mail.ee.kuas.edu.tw)

K.-L. Ke is with the Institute of Information Management, National University of Kaohsiung, Kaohsiung 811, Taiwan

[1]Accessible from http://www.cis.hut.fi/research/refs/

hierarchical structure. Thus it is also well fit for tasks regarding text documents such as text categorization, text clustering, and text hierarchy generation, etc.

The following text is divided into four sections. Sec. II describes some works related to our research. In Sec. III we will introduce the proposed topic-oriented self-organizing map algorithm. Sec. IV gives the experimental result. Finally, we give conclusions and discussions in the last section.

II. Related Work

We briefly review some related works here.

A. Adaptable SOM

To overcome the two major disadvantages, namely static map structure and lack of hierarchical relationships, of traditional SOM [3], [1], many models have been proposed. Some models use adaptable map size, e.g. the growing grid model [4]. Another approach is to use hierarchical arranged maps, such as hierarchical feature map [5] and tree-structured self-organizing map [6]. Hybrid approaches were also developed, e.g. growing hierarchical self-organizing map (GHSOM) [7], [8], [9].

B. Text Organization Based on SOM

Here we refer text organization to the efforts involved in organizing a corpse of texts into some predefined structures. Typical text organization processes include text clustering, text categorization, and text structure generation. Text organization research has been active for several decades. Many methods have been proposed to conquer this problem. Here we mention some works that make use of SOM. SOM was widely used in text clustering. A famous example is the WEBSOM project [10]. Liu et al. [11] proposed ConSOM which use concepts along with traditional features to guide the learning process. Their method demonstrated better result compared to traditional SOM due to its semantic sensibility.

III. Topic-Oriented SOM Algorithm

In this work we try to develop a new training method for SOM. The core of the proposed approach is the identification of topics which are used to guide the training process as well as to change the structure of the map. Fig. 1 depicts the flowchart of our method. The detailed explanations of the steps are described as follows.

A. Preprocessing

We need to transform the text documents into vectors before training since the input to SOM need to be vector-form. In this step common procedures for processing text documents, such as word segmentation, stopword elimination, and stemming, were first applied to obtain a set of keywords that can describe the contents of a document. All keywords were collected into the vocabulary of the corpse. A document is then transformed into a vector according to the keywords it contains. Let document $d_j = \{k_{ij} | 1 \leq i \leq n_j\}, 1 \leq j \leq N$, where N is the number of documents, n_j is the number of distinct keywords in d_j, and k_{ij} represents the ith keyword in d_j. The vocabulary, denoted by V, is just the union of all d_j, i.e. $V = \bigcup_j d_j = \{k_i | 1 \leq i \leq |V|\}$. A document is encoded into a binary vector according to those keywords that occurred in it. When a keyword k_i occurs in this document, the ith element of the vector will have value 1; otherwise, the element will have value 0. With this scheme, a document d_j will be encoded into a binary vector $\mathbf{d}_j$.

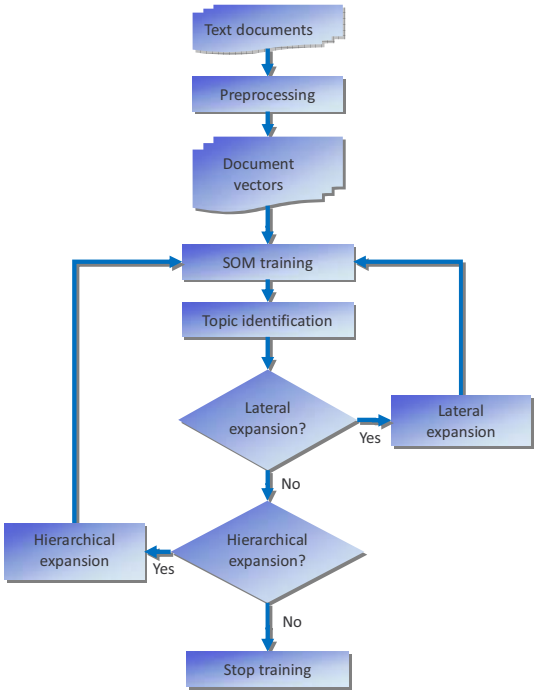

Fig. 1. Flowchart of the proposed method.

B. SOM Training

The document vectors were trained by classical SOM algorithm [3]. The initial training was performed on a small map, says a 2×2 map, named initial SOM. The initial SOM will be expanded when training proceeds. When the training process is accomplished, we then perform a labeling process on the trained map to establish the association between each document and one of the neurons. The labeling process is described as follows. The feature vector of document d_j, $\mathbf{d}_j = (x_{ji}), 1 \leq i \leq |V|, 1 \leq j \leq N$, is compared to the synaptic weight vectors of every neuron in the map. We then label d_j to the lth neuron if its synaptic weight vector is closest to $\mathbf{d}_j$, i.e. $||\mathbf{d}_j - \mathbf{w}_l|| = \text{argmin}_m ||\mathbf{d}_j - \mathbf{w}_m||$, where $\mathbf{w}_m$ is the synaptic weight vector of neuron m, $\mathbf{w}_m = (w_{mi}), 1 \leq i \leq |V|$. After the labeling process, each document is labeled to some neuron or, from a different point of view, each neuron is labeled by a set of documents. We record the labeling result and obtain the document cluster map (DCM). In the DCM, each neuron is labeled by a list of documents which are considered similar and are in the same cluster.

We will also construct the keyword cluster map (KCM)

by labeling each neuron in the trained network with certain keywords. Such labeling is achieved by examining the neurons' synaptic weight vectors. For the weight vector of the kth neuron $\mathbf{w}_k$, if its ith component exceeds a predetermined threshold, the corresponding word of that component, i.e. k_i, is then labeled to this neuron. By virtue of the SOM algorithm, a neuron may be labeled by several keywords which often co-occur in a set of documents, making a neuron a keyword cluster.

C. Topic Identification

A topic identification process is applied to the trained map to identify topics of neurons. We identify topics by examining the weight vectors of neurons. The underlying idea is that SOM will cluster similar documents together. Such documents should have similar topics and share many keywords. Therefore, a neuron's synaptic weight vector should have some elements with larger values which reflect the importance of these keywords shared by those documents labeled to this neuron. Finding such keywords should also identify topic of the neuron (or cluster).

According to above realization, we design the following topic identification schemes. The topics of the cluster on neuron l is a set of keywords which corresponding elements in $\mathbf{w}_l$ have values higher than some threshold. That is,

$$k_i \in C_l \iff w_{li} > \tau, \tag{1}$$

where k_i is the ith keyword in V, C_l is the set of topics for neuron l, w_{li} is the ith element of $\mathbf{w}_l$, and τ is a threshold.

Eq. 1 only consider weight vector of single neuron. By virtue of SOM, neighboring neurons should represent similar clusters. Therefore we can obtain more reliable topics by accumulating the weights from neighboring neurons as follow:

$$k_i \in C_l \iff \frac{1}{N_c(l)} \sum_{m \in N_c(l)} w_{mi} > \tau, \tag{2}$$

where $N_c(l)$ is the set of neighboring neurons of neuron l. Using these schemes, we can obtain the set of topics for each neuron. Note that we will not identify topic for neurons that were not labeled by any documents.

Eq. 1 and 2 are called naive and aggregated neuron weight thresholding schemes, abbreviated NNWT and ANWT schemes, respectively. Another scheme, namely average document weight thresholding (ADWT) scheme, identifies topics using the documents labeled on a neuron rather than weight vector of the neuron itself:

$$k_i \in C_l \iff \frac{1}{|A_l|} \sum_{j \in A_l} x_{ji} > \tau, \tag{3}$$

where A_l denotes the set of documents labeled to neuron l. Note that the threshold τ in Eq. 1, 2, and 3 are not necessary the same.

D. Lateral Expansion

We need to decide if lateral expansion is necessary before we actually doing it. The need of lateral expansion for a neuron relies on the diversity of topics on it. When the neuron has diverse topics, the documents labeled on it may have different topics. It is necessary to expand this neuron, i.e. giving more neurons, to allow these documents forming much precise clusters. The map needs lateral expansion when the topics on a neuron are incompatible. Two possible approaches could be used to measure the incompatibility between topics. The first is to use some predefined ontology such as WordNet [12]. When topics on a neuron are dissimilar enough, we could then expand the neuron. To measure the similarity, we could adopt the common measurements, such as those compiled in [13], defined via WordNet. The other approach is to examine the distribution of weight for a topic keyword across the neurons on the map. The distribution is defined as the histogram of the weight corresponding to the topic keyword in each neuron. Let $\mathbf{H}_i$ denote the histogram of keyword k_i. We can define $\mathbf{H}_i = (w_{ji}), 1 \le j \le N$. The topic incompatibility is measured as the average pairwise differences among topics:

$$I_l = \frac{1}{\binom{|B_l|}{2}|V|} \sum_{p,q \in B_l} ||\mathbf{H}_p - \mathbf{H}_q||, \tag{4}$$

where I_l and B_l denote the topic incompatibility and the set of topic keywords on neuron l, respectively. Note that the histograms should be normalized to obtain correct result. We only expand one neuron with the largest topic incompatibility after each SOM training.

When a neuron needs lateral expansion, we can expand the neuron in two ways. The first is the omnidirectional approach which inserts two rows and two columns of neurons around this neuron, as shown in Fig. 2. The synaptic weight vector of an added neuron is set as the average of its two neighboring neurons in the original map. When the newly added neuron is on the boundary of the map, the weight vector of the closest neuron in the original map is copied. The second approach is to insert a row or a column of neurons between the neuron and its worst neighbor, as shown in Fig. 3. This approach is the same as GHSOM [8]. The worst neighbor of the expanded neuron is the one, among all neurons surrounding this neuron, with the largest distance between their weight vectors. Note that we only show the case of adding a column of neurons in Fig. 3. In the case that the worst neighbor locates on the same column of the neuron, a row of neurons will be added between the expanded neuron and its worst neighbor. The weights of the added column or row of neurons are set as the average of their closest neighbors in the original map.

E. Hierarchical Expansion

The need for hierarchical expansion is decided after lateral expansion. Here we will expand a neuron with a lower-level map when the neuron contains a large or inconsistent document cluster which satisfies one of the following criteria:

1) The number of documents labeled to this neuron exceeds a multiple of the average number.
2) The document cluster has a high document inconsistency. The document inconsistency is defined as :

$$\frac{1}{|A_l||V|} \sum_{j \in A_l} ||\overline{\mathbf{d}_l} - \mathbf{d}_j||, \tag{5}$$

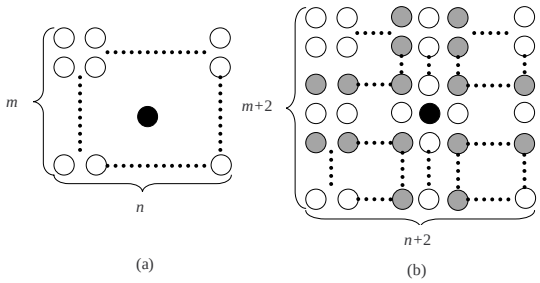

Fig. 2. The omnidirectional lateral expansion approach. (a) The size of the original map is $m \times n$. The black disk depicts the neuron to be expanded. (b) After expansion, the size of the map becomes $(m+2) \times (n+2)$. The grey disks depict the added neurons.

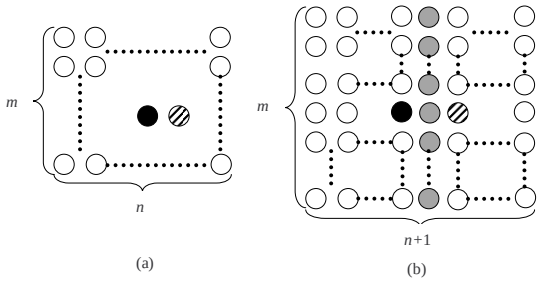

Fig. 3. The worst-neighbor lateral expansion approach. (a) The size of the original map is $m \times n$. The black disk depicts the neuron to be expanded. The slashed disk is its worst neighbor. (b) After expansion, the size of the map becomes $m \times (n+1)$ since the worst neighbor is in the same row of the neuron. The grey disks depict the added neurons.

where $\overline{\mathbf{d}_l}$ denotes the average of document vectors in A_l, i.e. $\overline{\mathbf{d}_l} = \frac{1}{|A_l|} \sum_{j \in A_l} \mathbf{d}_j$.

When a neuron l needs hierarchical expansion, we should create an initial SOM as described in Sec. III-B. This SOM is then trained by the documents labeled to this neuron, i.e. A_l.

IV. Experimental Result

We performed the experiments on two corpora which were used in our previous research [14], [15] for comparison purpose. The first corpus (C1) contains 100 Chinese news articles posted in Aug. 1, 2, and 3, 1996 by Central News Agency[2]. The second corpus (C2) contains 3268 documents posted during Oct. 1 to Oct. 9, 1996. A word extraction process is applied to the corpora to extract Chinese words. There are 1475 and 10937 words extracted from C1 and C2, respectively. To reduce the dimensionality of the feature vectors we discarded those words which occur only once in a document. We also discarded the words appeared in a manually constructed stoplist. This reduces the number of words to 563 and 1976 for C1 and C2, respectively. A reduction rate of 62% and 82% are achieved for the two corpora, respectively.

We trained the corpora with the proposed algorithm. The sizes of the initial SOMs for C1 and C2 are 2×2 and 4×4, respectively. Table I shows the parameters used in the experiments.

[2]http://www.cna.com.tw/

TABLE I
Training parameters used in our experiments.

Corpus	C1	C2
Size of Initial SOM	2×2	4×4
Number of synapses in a neuron	563	1976
Initial training gain for SOM training	0.4	0.4
Maximal training time for SOM training	200	500

TABLE II
The result of training process

Corpus C1				
Topic identification scheme	NNWT	ANWT	ADWT	LabelSOM
Average number of topic in each neuron	2.37	2.78	3.67	3.23
Average number of neurons in each map	5.75	6.24	5.47	6.45
Average number of map in each training	4.32	3.67	5.27	6.31
Average depth of hierarchies	2.3	3.21	3.45	5.76
Corpus C2				
Topic identification scheme	NNWT	ANWT	ADWT	LabelSOM
Average number of topic in each neuron	4.32	5.68	6.57	6.54
Average number of neurons in each map	8.81	9.27	9.31	11.46
Average number of map in each training	7.37	8.26	9.31	12.29
Average depth of hierarchies	3.25	3.94	4.37	7.58

After SOM training, we performed topic identification using three topic identification schemes described in Sec. III-C. We then find neurons which need lateral expansion according to the topic incompatibility defined in Sec. III-D. The thresholds for topic identification were set to 0.8, 0.85, and 0.8 for NNWT, ANWT, and ADWT schemes, respectively. Besides, the threshold for topic incompatibility was set to 0.7. Both omnidirectional and worst neighbor schemes were adopted to expand a neuron. The hierarchical expansion was then performed with document inconsistency threshold been set to 0.65. A neuron was expand to a lower SOM with sizes initialized as those in Table I. We retrained the two corpora 100 times and obtained statistical information about the proposed algorithm. Table II shows the statistics of the training process. Note that the values are averaged over 100 times of training. We also conducted the same experiments using GHSOM, which is denoted as LabelSOM in Table II.

V. Conclusions and Discussions

The self-organizing map model has been widely and successfully used in data clustering and visualization, as well as other wide scope of applications. Traditional SOM models cluster data according to their similarity. Besides, the structure of the map is always fixed. Various models have been proposed to overcome such insufficiency. In this work, we try to develop a novel learning algorithm which is suitable for text organization. When a set of text documents are being trained, we will identify the topics of a neuron which represents a document cluster after SOM training. We then use such topics to decide if a neuron needs to be lateral expanded or

hierarchical expanded. Since our method incorporates various text mining approaches in training, it is feasible to use our method on text documents. The experimental results suggest that our method is adequate for developing structure which can be used for text categorization and organization.

ACKNOWLEDGMENT

This research was funded by National Science Council under grant NSC 98-2221-E-390-040.

REFERENCES

[1] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer-Verlag, 1997.
[2] M. Pöllä, T. Honkela, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 2002-2005 addendum," Information and Computer Science, Helsinki University of Technology, Espoo, Finland, Tech. Rep. TKK-ICS-R24, 2009.
[3] T. Kohonen, "Self-organizing formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.
[4] B. Fritzke, "Growing grid - a self-organizing network with constant neighborhood range and adaption strength," *Neural Processing Letter*, vol. 2, no. 5, pp. 9–13, 1995.
[5] R. Miikkulainen, "Script recognition with hierarchical feature maps," *Connection Science*, vol. 2, pp. 83–101, 1990.
[6] P. Koikkalainen, "Tree structured self-organizing maps," in *Kohonen Maps*, E. Oja and S. Kaski, Eds. Amsterdam, Netherlands: Elsevier, 1999, pp. 121–130.
[7] A. Rauber, M. Dittenbach, and D. Merkl, "Towards automatic content-based organization of multilingual digital libraries: An English, French and German view of the Russian information agency Nowosti news," in *Proceedings of the Third All-Russian Scientific Conference on Digital Libraries: Advanced Methods And Technologies, Digital Collections*, September 11-13 2001, pp. 11–13.
[8] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1331–1341, 2002.
[9] M. Dittenbach, A. Rauber, and D. Merkl, "Recent advances with the growing hierarchical self-organizing map," in *Advances in Self-Organizing Maps*, N. Allinson, Y. Ahujun, L. Allinson, and J. Slack, Eds. Lincoln, England: Springer, 2001, pp. 140–145.
[10] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "WEBSOM–Self-organizing maps of document collections," *Neurocomputing*, vol. 21, pp. 101–117, 1998.
[11] Y. Liu, X. Wang, and C. Wu, "ConSOM: A conceptional self-organizing map model for text clustering," *Neurocomputing*, vol. 71, no. 4-6, pp. 857–862, 2008.
[12] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[13] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity - measuring the relatedness of concepts," in *HLT-NAACL 2004: Demonstration Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 38–41.
[14] C. H. Lee and H. C. Yang, "A Web text mining approach based on self-organizing map," in *Proceedings of the ACM CIKM'99 2nd Workshop on Web Information and Data Management*, Kansas City, Missouri, 1999, pp. 59–62.
[15] H. C. Yang and C. H. Lee, "A text mining approach on automatic generation of Web directories and hierarchies," *Expert Systems with Applications*, vol. 27, no. 4, pp. 645–663, 2004.