High-Speed Self-Organzing Map for Document Clustering

*Ponthap Rojanavasu and **Ouen Pinngern

Department of Computer Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
(Tel : +66-2-739-2002; E-mail: s4061612@kmitl.ac.th*, kpouen@kmitl.ac.th **)

Abstract: Self-Oranizing Map(SOM) is an unsupervised neural network providing cluster analysis of high dimensional input data. The output from the SOM is represented in map that help us to explore data. The weak point of conventional SOM is when the map is large, it take a long time to train the data. The computing time is known to be O(MN) for trainning to find the winning node (M,N are the number of nodes in width and height of the map). This paper presents a new method to reduce the computing time by creating new map. Each node in a new map is the centroid of nodes' group that are in the original map. After create a new map, we find the winning node of this map, then find the winning node in original map only in nodes that are represented by the winning node from the new map. This new method is called "High Speed Self-Oranizing Map"(HS-SOM). Our experiment use HS-SOM to cluster documents and compare with SOM. The results from the experiment shows that HS-SOM can reduce computing time by 30%-50% over conventional SOM.

Keywords: Self-organizing map, SOM, HS-SOM, Document clustering, Unsupervised neural network

1. INTRODUCTION

Self-Oranizing Map(SOM), unsupervised neural network is an providing cluster analysis of high dimensional input data[2,4,9]. The advantage of this approch is that its result is represented by using two dimensions mapping. This mapping show data's relationship including distribution and density of data. Unfortunately, high desity data affect the map to increase its nodes and take longer time for computation. Most time consuming for computation are finding the winning nodes. If there are MxN dimension in the map, finding the winning node will take O(MN) steps . Each step is taken for computing distance of vectors. For document clustering, vectors have a very high dimension so it take a long time to compute. In this paper, we propose new method to reduce computing time for finding the winning node. We separate nodes in map into small group and use centroid vector to find the winning node of each group. Then we find winning node from them again.

2. SELF-ORGANIZING MAP(SOM)

The self-organizing map (SOM) is one of the most prominent unsupervised artificial neural network models[1,3]. The model consists of neural elements called nodes. Each node i is assigned an n-dimensional weight vector m_i. That is $m_i \in \Re^n$, where $\Re^n$ is an n-dimensional space. It is necessary to note that the weight vectors have the same dimensionality as the input patterns.

The learning process of SOM may be described in terms of adaptive node for input vectors[5-7]. In each t learning-iteration, input vector x(t) is randomed to compare with every node in the map for finding the output vector's winning node. Generally, function for comparision is Euclidean distance. The winning node c can thus be defined the smallest distance between input vector and nodes by

$$c : m_c(t) = \min_i \| x(t) - m_i(t) \| \tag{1}$$

The weight of winning node, c , is tuned by consider the differentiation of input vector and weight vector. Each learning-iteration is gradually reduced. Not only winning node is learning but also its neighborhood node, and thus the weight vectors become more similar to the input pattern. The respective node is more likely to win at future presentations of this input pattern shown in :

$$m_i(t+1) = m_i(t) + \alpha(t) \times h_{ci}(t) \times [x(t) - m_i(t)] \tag{2}$$

where t is the number of learning-iteration, $x(t)$ is current input vector, $m_i(t)$ is weight vector, $\alpha(t)$ is learning rate depended on number of iteration shown in:

$$\alpha(t) = \alpha(0) \times \frac{T - t}{T} \tag{3}$$

when T is the total number of iteration, t is the current iteration

$h_{ci}(t)$ is the neighborhood function. For convergence it is necessary that $h_{ci}(t)$ 0 when t 0. This means that with learning increasing, the neighborhood within which the nodes are activated will shrink and the same time the modifying rate of reference vectors will decrease. Usually we use Guassian function,

$$h_{ci}(t) = \exp(-\frac{\| r_c - r_i \|^2}{2\sigma^2(t)}) \tag{4}$$

when $\|r_c - r_i\|$ is the father node i from the winning node c, $\sigma(t)$ is the radius of neigborhood node as

$$\sigma(t+1) = 1 + (\sigma(t) - 1) \times \frac{T - t}{T} \tag{5}$$

Fig 1. shows 5x5 SOM.Firstly, the input vector x(t) is randomly selected from input domain, then find winning node as the darkest node. Secondly, the weight vector $m_c(t)$ is tuned to $m_c(t+1)$ which will get close to the input vector. Finally the neighborhood nodes of node c (lighter shading) are turned according to the previous equations.

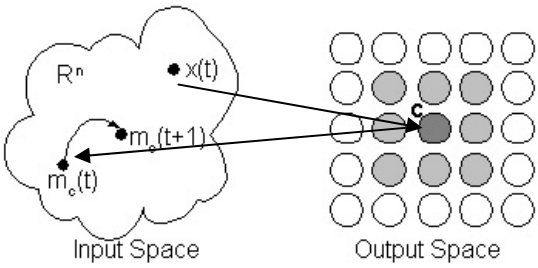


Fig 1. Structure of SOM and its learning

3. HIGH SPEED SELF-ORGANIZING MAP(HS-SOM)

Main problem for conventional SOM is that if the map has large size and too many dimensional data, it takes a long time to train the data, e.g. if there are MxN map, the time taken for finding the winning node is O(MN). This paper proposes the method to reduce this computation time by create new SOM map between input layer and SOM layer as shown in Fig 2

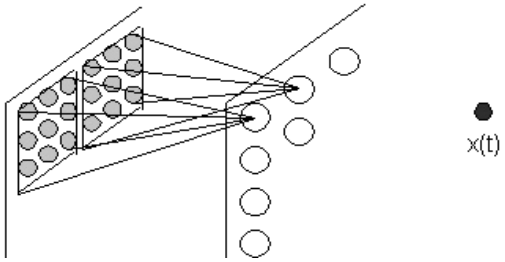


Fig 2. New layer is represented for reducing computation time

Weight vectors in the second layer are computed from nodes' centroid in first layer as equation (6)

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (6)$$

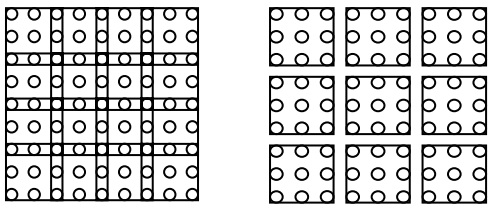
when x_i is vector component, n is total number of nodes in group

The new algorithm is used to find the winning node and adjust the neighborhood nodes according to the following steps:

New Algorithm

1. Find the winning node in the second layer map(Fig. 1)
2. Find the winning node in the first layer only in nodes that are represented by the winning node from the second layer.
3. Adjust neighborhood nodes in the first layer
4. Adjust nodes in the second layer by findding new centroids for the changed nodes in the first layer.

The computation for new winning node depend on the design of grouping in first layer which we can design both the number of member and its shape. We also can design the overlapping of groups(Fig 3). We will discuss the effect of the design in next section



a.) The overlapping design b.) The non-overlapping design

Fig 3. The design of node grouping

Computing time for non-overlap square group of HS-SOM(Fig. 3 (b)) is :

time = (axb)+(i x j) (7)

where

i – number of groups' members in width of first layer,
j – number of groups' members in height of first layer,
a – number of nodes in width of the second layer,
b – number of nodes in height of the second layer,
and (ixj) is computation time for winning node in first layer. (axb) is computation time for winning node in second layer. For example If M=9,N=9, Fig 3 a) is 4x4 overlapping design that has i=j=3, a=b=4 Fig 3 b) is 3x3 non-overlapping design that has a=b=3 and i=j=3. The computing time to find the winning node in conventional SOM is 9x9(MxN)=81, while computing time in HS-SOM Fig 3 a) is (4x4)+(3x3) = 25 and Fig 3 b) is (3x3)+(3x3) = 18. If i=j=1 then a=M, b=N and the computation time is in order of O(MN) that is the conventional computation time. If the map has large size we can increase some more layer to reduce computation time.

4. THE IMPLEMENTATION OF HS-SOM IN DOCUMENT CLUSTERING

4.1 Characteristic extraction of document

There are 1111 documents for clustering. We use document title and abstract to represent each document[1,2]. Before creating index term of document, we eliminate the stop words such as "a", "and", "the" and we use Porter stemming [8] to find the stemming word such as "classification", "classify" , "classified" which we will replace with "classif"

After reducing the number of word in documents, the template vector is created for all documents. Its component compose of 1230 words. Each document uses this template vector to create its own vector by compute the term weight using equation (8)

$$w_{i,j} = f_{i,j} \times idf_i \qquad (8)$$

when $f_{i,j} = \frac{freq_{i,j}}{\max(freq_{i,j})}$ (9)

and $idf_i = \log\left[\frac{N}{n_i}\right]$ (10)

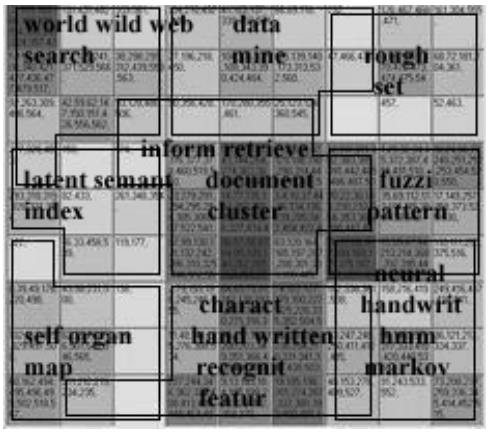
where $f_{i,j}$ is normalized frequency words, $freq_{i,j}$ is frequency of term k_i in document d_j , N is the total number of documents and n_i is the number of document appeared in term k_i

Thus , the documents are represented in term of $\vec{d}_j = \{w_{1,j}, w_{2,j}, w_{3,j}, ..., w_{t,j}\}$ where $w_{i,j} \geq 0$.

4.2 Document clustering by HS-SOM

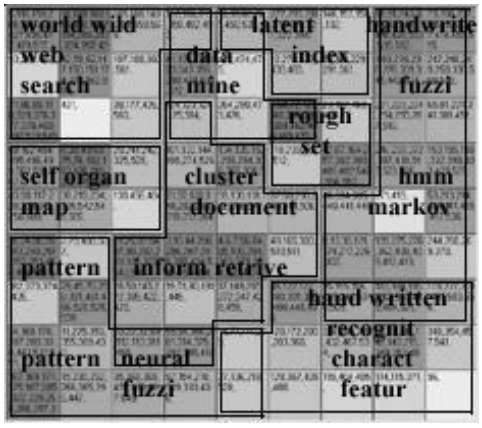
From the experiment of HS-SOM in document clustering, we design first layer size in 9x9, i=j=3. For the second layer, we design two size in 3x3 and 4x4 as in Fig 4 a) and 4 b)

In learning process, we use number of iteration T=5000, the initial learning rate $\alpha(0)=0.1$ and the initial radius of neighborhood nodes $\sigma(0)=5$. For weight vector we random the initial value between 0-0.1



a.) The 3x3 second layer document clustering.

b.) The 4x4 second layer document clustering.

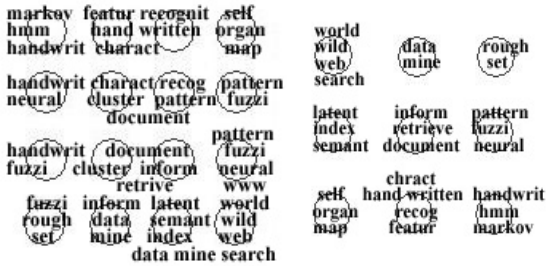


c.) The conventional SOM document clustering by KOHONEN

Fig 4. The order maps of HS-SOM and conventional SOM

It is necessary to consider the data characteristic when we design of HS-SOM . If the domain of documents for clustering are not related to each other, the non-overlapping map is used. For example, Fig 4 a) has show the domain of "inform retrive" is clearly not related to the domain of "hand written". For the documents that their domains are unclear, we propose the overlapping map in second layer.

The advantages of HS-SOM for document clustering, first, from Fig 5 a) and 5 b) show that it is easier to browse document in HS-SOM than in the conventional SOM because we can browse the group of document in second layer to guide the group of documents in first layer.



a.) 4×4 Second layer b.) 3×3 Second layer

Fig 5. The sample of second layer in different size.

From Fig 5 a), some nodes have two domains so there are overlapping data, for example in lower left node has two domain of "rough set" and "fuzzi set". But for Fig 5 b) the upper right has domain of "rough set" separated from "fuzzi set".

Second, as a result, we can reduce time for finding winning node, the speed of traning in HS-SOM is better than convensional SOM. The winning node from HS-SOM in the winning node of sub-domain which may not be a winning node of the whole domain of document. Considerately, the winning node is the significant node for the associated documents that will be the neighborhood nodes. As a consequence, we can group nodes and use the centroid to find the represent node. We will choose the shortest centroid to find winning node in that group.

Fig 6 is representing the time taken to train the learning of different SOM. The iteration T is 5000. The graph show, that HS-SOM 4x4 and 3x3 take less time, depend on the pattern of grouping, than conventinal SOM about 50%.

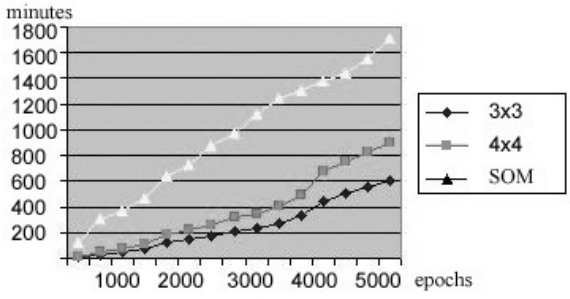


Fig 6. Show time taking to train HS-SOM 3x3, 4x4, and conventional SOM

The measurment of efficient for SOM use the entropy to indicate the document clustering. If there is one domain in the cluster, its entropy is zero. If there are many domains in the cluster, its entropy is going to be high as equation (11)

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \quad (11)$$

The summation of the entropy is in equation (12)

$$E_{cs} = \sum_{j-1}^{m} \frac{n_j \times E_j}{n} \quad (12)$$

where p_{ij} is the probability of the members in node j belong to group i, n_j is the number of document in node j, m is the number of clusters, that is the number of nodes and n is the number of document.

Table 1. The entropy of the different clustering.

Type	Summation of Entropy
Conventional SOM	0.295
HS-SOM 3x3	0.283
HS-SOM 4x4	0.315

From Table 1, the entropy of three SOM is similar. The entropy of HS-SOM 3x3 is less than HS-SOM 4x4 that mean the domain of sample documents for our experiment is almost not relate. The implicit documents are located at the overlapping area between groups.

We created the interface to browse the relevant documents. We assigned index terms for each node and compared with keyword. In Fig 7 we searched by using keyword "self organizing map". Before searching, the keyword was taken to find its stem words and displayed the most relevant node. We can move the arrows to browse the neighborhood node for related document.

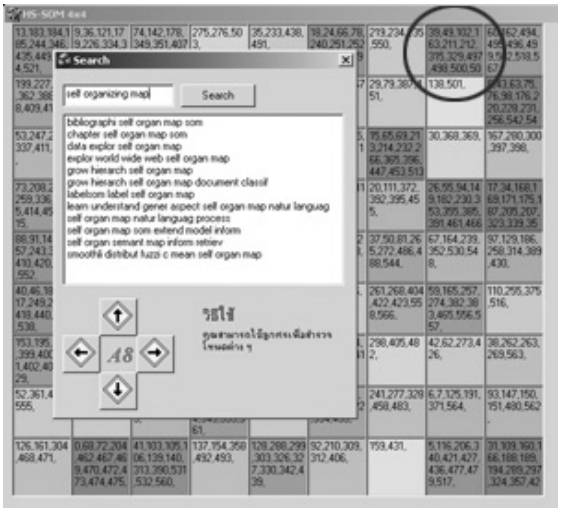


Fig 7. The interface to browse the relevant documents in HS-SOM4x4

5. CONCLUSION AND FUTURE

Our research focus on technique for reducing computing time in SOM by findding the winning node in the original map only in nodes that were represented by the winning node from the new map. We can increase the layer if there are a large number of data. Our experiments show that HS-SOM outperforms conventional SOM by 30%-50%. One of special characteristic of HS-SOM is the ability to initialize the direction the user's browsing by using the upper layer map. For this approch we can reduce time for browsing.

We can apply the HS-SOM to various applications such as image clustering, signal classification and pattern recognition.

REFERENCES

[1] 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, November 2002.

[2] A.Rauber, D. Merkl, "The SOMLib Digital Library System," *Proceedings of the 3rd Europ. Conf. on Research and Advanced Technology for Digital Libraries (ECDL'99)*, Paris, France, September 22. - 24. 1999, Springer, 1999.

[3] Helge Ritter, Thomas Martinetz, Klaus Schulten, *Neural computation and self-organizing maps : an introduction*, Imprint Massachusetts : Addison-Wesley, 1992.

[4] Kohonen, T. (1998). "Self-organization of very large document collections: State of the art," *ICANN98*, pages 65-74. Springer, London

[5] Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. "WEBSOM for textual data mining," *Artificial Intelligence Review*, volume 13, pages 345-364, Kluwer Academic Publishers, 1999

[6] Merkl and A. Rauber, "Document Classification with Unsupervised Neural Networks", *Soft Computing in Information Retrieval*,pp. 102 - 121, 2000.

[7] Qing Ma, Min Zhang, Ming Zhou. "Self-Organization of Chinese Semantic Maps Using TFIDF Term Weighting," *The Second Workshop on Natural Language Processing and Neural Networks*, Tokyo, Japan, November, 2001.

[8] R.Baeza-Yates B. Ribeiro-Neto, *Modern Information Retrieval*, ACM- Press, Addison-Wesley, 1999.

[9] Xia Lin, Dagobert Soergal, Gary Marchioninl. "A Self-Organizing Semantic Map," *ACM*,1991.