Evaluation of the Performance of $O(\log_2 M)$ Self-Organizing Map Algorithm without Neighborhood Learning

Hiroki Kusumoto and Yoshiyasu Takefuj,

Graduate School of Media and Governance, Keio University, Kanagawa, Japan

Summary
In this paper, we evaluate the performance of $O(\log_2 M)$ self organizing map (SOM) algorithm previously presented by us. Our algorithm was developed for reduction of the computational costs and is the fastest one in SOM algorithm. The order of the computational costs is $\log_2 M$ where the size of a feature map is M^2. Our algorithm does not require neighborhood learning and thus tuning of parameters is relatively simple. The performances of the basic SOM developed by Kohonen and our algorithm were tested using the benchmark of a central nervous system (CNS) tumor patient dataset which comprises five groups. The simulation results show our algorithm can map the input dataset more appropriately than the basic SOM with constant or changing parameters of the neighborhood function. Our algorithm is able to contribute to various research fields using the SOM algorithm.

Key words: *Performance of Self-Organizing map (SOM), central nervous system tumor (CNS), Computational reduction, Neighborhood learning, Subdividing method, and Binary search*

Introduction

Self organizing map (SOM) developed by Kohonen [1] is one of the unsupervised learning algorithms that maps multi-dimensional input datasets onto a two-dimensional lattice or hexagonal space in such way that the similar input data are mapped closely and the different input data distantly. SOM is applied in various research fields including speech or speaker recognition [2], image segmentation [3], and bioinformatics [4].

One of SOM problems lies in computational costs. The computational costs are largely depending on comparing weight vectors for searching winner vectors where the order of the basic SOM is M^2 when the size of the map is M^2. Tree-structured SOM with $O(M \log M)$ was proposed by Koikkalainen and Oja [5] and Truong [6], a new method with $O(M)$ by Kohonen, and $O(\log M)$ by Xu and Chang [7]. Our algorithm requires $O(\log_2 M)$ times of comparison using the subdividing method with the binary search method, and is currently the fastest SOM algorithm [8].

The other SOM problem is in difficulty of parameters tuning. Especially, the neighborhood function, an essential part of the SOM algorithm, has several parameters involved each other. Neighborhood size and variations of learning rates according to the distance from winner vectors are required to be set adequately. And reductions of neighborhood size and learning rates through iteration learning are important for high performance mapping. Our algorithm tunes just learning rates of winner vectors without any neighborhood learning function where tuning of parameters is relatively simple [8].

In this paper, the performances of the basic SOM and our algorithm were tested using the benchmark of a central nervous system (CNS) tumor patient dataset [9] which comprises five groups. Similar data belonging to the same group should be mapped to cluster in a tight area but are sometimes mapped on long-distant areas unsuccessfully making two or more clusters when the size of the map is large. The number of clusters made on maps was counted under various settings of parameters. $O(\log_2 M)$ SOM algorithm is detailed in section 2 and CNS tumors dataset in section 3.

2. $O(\log_2 M)$ SOM algorithm

In this section, our previous work [8] is quoted to illustrate $O(\log_2 M)$ SOM algorithm precisely.

The proposed algorithm is composed of two methods. One is a binary search for searching winner vectors and the other is a method of subdividing feature map gradually. At the initialization in the proposed algorithm, there is a 2×2 weight vector on SOM as shown in Fig.1-a. As the process of the proposed algorithm proceeds, the feature map is subdivided by the subdividing method. At any subdivision stage in the proposed algorithm, winner vectors are searched roughly at the beginning and accurately at the end by the binary search method. The proposed algorithm does not have the neighborhood function, because one and only one winner-vector set is trained every stage. The procedure is detailed in this section.

A. Initialization

Input is a dataset of k-dimensional vectors X_i(i=1,2,3,…,n). A feature map is two-dimensional layer of $M \times M$ nodes

Manuscript received October 5, 2006.
Manuscript revised October 25, 2006.

(M=2^m+1, m=1, 2, 3,…). At the beginning, only four nodes on the coordinates (1,1), (1,M), (M,1) and (M,M) have k-dimensional weight vectors W(x,y) whose values are arbitrary in the extent of the distribution of input data and other nodes do not appear as shown in Fig. 1-a. These four weight vectors are trained by the same method as the basic SOM with total O(1) computation.

B. Subdividing method

Subdividing method draws median lines between all neighboring two lines on feature maps, so that subdivides an $M'\times M'$ feature map into a $(2M'-1)\times(2M'-1)$ feature map. Fig.1-a shows the process of the subdivisions from a 2×2 map to a $M\times M$ map. Every new node is assigned a weight vector, whose value is the average of the values of weight vectors of the closest nodes to the new node. The values of the new gray nodes in Fig.1-b are defined by

$$W(X,Y+s)=\frac{W(X,Y)+W(X,Y+2s)}{2} \quad (1)$$

$$W(X+s,Y)=\frac{W(X,Y)+W(X+2s,Y)}{2} \quad (2)$$

$$W(X+s,Y+2s)=\frac{W(X,Y+2s)+W(X+2s,Y+2s)}{2} \quad (3)$$

$$W(X+2s,Y+s)=\frac{W(X+2s,Y)+W(X+2s,Y+2s)}{2} \quad (4)$$

$$W(X+s,Y+s)=\frac{W(X,Y)+W(X,Y+2s)}{4}+\frac{W(X+2s,Y)+W(X+2s,Y+2s)}{4} \quad (5)$$

After each subdivision, winner vectors searched by the binary search are trained with the total $T_{(p)}$ times. T is an overall total learning times and p is the number of subdivision stages when the feature map size is $(2^{p-1}+1)\times(2^{p-1}+1)$. The final size of the map is determined at the initialization and the subdivision stops at the size.

C. Binary Search

When an $M''\times M''$ map, that means that $M''\times M''$ nodes on a map have weight vectors and other nodes do not appear, and an input data X(t) are given, an winner vector is searched by following procedures.

At the initialization, the search space of the map is extended by

$0\le x\le M$ and $0\le y\le M$ (6)

All space of the map is the subject of the search. After repeating Step A and Step B $\log_2(M''-1)+1$ times reflexively, the final closest vector $W(x_c,y_c)$ is the winner vector.

Step A: When search space is extended by

$x_1\le x\le x_2$ and $y_1\le y\le y_2$ (7)

the closest vector to X(t) is searched from four weight vectors on the vertexes of the search space.

$$\| X(t)-W(x_c,y_c)\| = \min\{\| X(t)-W_i \| \mid W_i = W(x_1,y_1), W(x_1,y_2), W(x_2,y_1), W(x_2,y_2)\} \quad (8)$$

Step B: The search extent is divided into four quarters and the proposed algorithm assumes that a winner vector is on a quarter space where the closest vector $W(x_c,y_c)$ exits. The extent of search space is changed into

$x'_1\le x\le x'_2$ and $y'_1\le y\le y'_2$ (9)

$$x'_1=\min(x_c,\frac{x_1+x_2}{2}) \text{ and}$$

$$x'_2=\max(x_c,\frac{x_1+x_2}{2}) \quad (10)$$

$$y'_1=\min(y_c,\frac{y_1+y_2}{2}) \text{ and}$$

$$y'_2=\max(y_c,\frac{y_1+y_2}{2}) \quad (11)$$

Fig.1-c shows the process of the binary search when M'=5. The solid lines mean the searched space, the weight vectors of the black and gray nodes are compared. The three weight vectors of the black nodes are the closest vectors to the input vector X(t) at each step and the final closest vector W_{win} is a winner.

D. Learning

The proposed algorithm only trains winner vectors by

$$W_{win}(t+1)=W_{win}(t)+\alpha(t)(X(t)-W_{win}(t)). \quad (12)$$

Because neighborhood vectors do not require training, learning rate $\alpha(0<\alpha<1)$ dose not have arguments of coordinates of the weight vectors. Value of α can decrease in inverse proportion to time argument t meaningfully and constant α can also work valuably.

3. CNS tumors dataset

The basic SOM and $O(\log_2 M)$ SOM were tested with 42 patient samples of the central nervous system tumors [9] that comprise five groups: 10 medulloblastomas, 10 malignant gliomas, 10 rhabdoid tumors, 8 supratentorial primitive neruroectodermal tumors, and 4 normal human

cerebella. Each sample is described by expression levels of 50 genes comparing these tumor types. Among each gene, the values of expression levels were normalized by the standard deviation from the mean. The dataset is available at http://www.broad.mit.edu/cancer/.

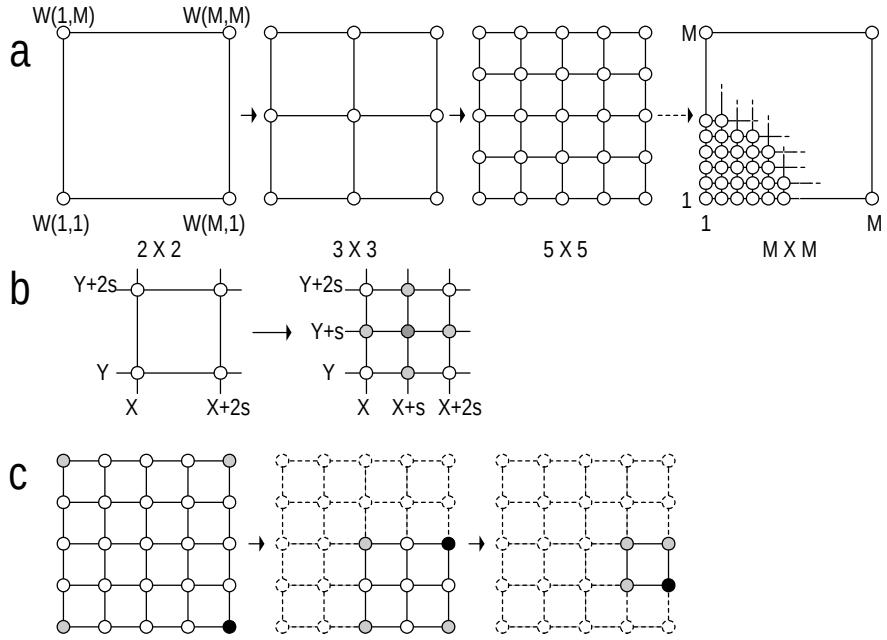

Fig.1. Process of $O(\log_2 M)$ SOM algorithm. Circles denote weight vectors. (a) Process of the subdividing method. (b) New node by subdivision. (a part of a map) (c) Process of the binary search method. [8]

4. Evaluation SOM performance

The optimal mapping is that the differences between the features of input data are represented by the distances on the two-dimensional map. The essential part of SOM algorithm is the neighborhood function that enables the optimal mapping. The neighborhood function tunes the close-range weight vectors with the same input vectors, and the values of close-range weight vectors become close. Consequently, the input data of the same group would be mapped on the close-range weight vectors.

But the neighborhood function sometimes allows that long-distance weight vectors become close. The neighborhood function can work the close-range weight vectors from winner vectors to be close, but cannot be effective in training the long-distant weight vectors to be different. As a result, input data of the same group are mapped separately to make two or more clusters.

In this paper, we have evaluated mapping results based on how small number of clusters the input data of the same group forms on the map. The definition of the cluster is

- **a.** The number of clusters is counted according to each group of the input dataset and finally added.
- **b.** An input data that the Manhattan distances on the map between itself and at least one input vector belonging to a cluster C_u are less than or equal to d_z belongs to C_u.
- **c.** The Manhattan distance on the map between at least one pair of elements in an arbitrary set of C_u and in the difference set must be less than or equal to d_z.
- **d.** When an input data X_a belonging to C_v (v is not equal to u) is mapped between two input data X_b and X_c belonging to C_u, the Manhattan distances between X_b and X_c must be measured circumventing the lattice on which X_a is mapped.

The CNS tumor dataset comprises five groups and when the summation of the number of clusters is five we presume that the mapping is optimal.

5. Simulations and results

We adopted three kinds of constant learning rates: 0.01, 0.005, and 0.001 and two kinds of map sizes: 9^2 and 17^2. At each simulation, total 10,000 input data was chosen from 42 patient samples randomly for learning of weight vectors. We set learning rate of winners and the

neighborhood of the basic SOM as $\alpha \times \frac{1}{d+1}$ where d denotes Manhattan distance from the winner vector and the constant neighborhood size as 2(M-1) that means all of the weight vectors are always objects of neighborhood learning. Our algorithm trained only winner vectors.

And next we adopted variable learning rates and the size of neighborhood for the basic SOM. The learning rates and the sizes of the neighborhood should be gradually reduced for the optimal training. We used two kinds of variable learning rates: one is 0.03 at the start of SOM

Table 1 The number of clusters after mapping by the basic SOM and $O(\log_2 M)$ SOM using CNS tumors dataset that comprises five groups. The breakdown of the 100 times simulations and the average under each condition are listed.

Method	Map size	Learning rate	Neighborhood size	The number of clusters								Average
				5	6	7	8	9	10	11	12	
The basic SOM	17^2	0.01	32	6	25	33	27	4	5	-	-	7.13
		0.005	32	5	19	34	26	9	2	3	2	7.43
		0.001	32	3	18	26	18	16	11	6	2	7.93
	9^2	0.01	16	8	35	34	18	5	-	-	-	6.77
		0.005	16	8	20	44	21	5	-	2	-	7.03
		0.001	16	3	23	20	33	13	8	-	-	7.54
	17^2	0.03 to 0.003	32 to 1	12	26	37	15	7	3	-	-	6.88
		0.01 to 0.001	32 to 1	9	22	39	19	10	1	-	-	7.02
	9^2	0.03 to 0.003	16 to 1	16	24	32	22	4	2	-	-	6.8
		0.01 to 0.001	16 to 1	8	32	26	27	4	3	-	-	7.1
$O(\log_2 M)$ SOM	17^2	0.01	0	14	67	14	3	-	2	-	-	6.14
		0.005	0	18	40	30	11	1	-	-	-	6.37
		0.001	0	14	43	29	13	1	-	-	-	6.44
	9^2	0.01	0	20	63	13	2	-	-	2	-	6.07
		0.005	0	9	45	29	9	8	-	-	-	6.62
		0.001	0	8	47	31	13	1	-	-	-	6.52

algorithm and gradually reduced to 0.003 at the end, the other is from 0.01 to 0.001 and a variable neighborhood size: 32 Manhattan distance with 17^2 map at the start and 16 with 9^2 map and gradually reduced to be one at the end. Each simulation was repeated 100 times with different seeds of the random function that defined the initialization of the weight vectors and counted the number of clusters according to the definition described in section 4. We set d_z=5 and 9 respectively when map size was 9^2 and 17^2. The breakdown of the 100 times of simulations under each condition is shown in Table 1. The average computational time for one simulation was 74.9 sec. by the basic SOM and 3.6 sec. by our algorithm when map size was 17^2.

6. Discussion

The average number of clusters by our algorithm is lower than that by the basic SOM under six kinds of conditions (Table 1). The basic algorithm could upgrade with gradually reduced learning rates and neighborhood sizes. On the other hand, our algorithm with any of three constant learning rates and no learning of neighborhood weight vectors worked effectively.

In the process of the SOM iteration learning, once both of long-distant two weight vectors become close to input data belonging to a group, the possibility that the weight vectors differ is low unless many winner vectors for the input data belonging to other groups are found very near one of the weight vectors fortunately. Because neighborhood learning, the essential part of the SOM algorithm, affects the long-distant weight vectors from winner vectors very weakly.

There are conceivably two reasons why the values of two long-distant weight vectors are close in the process of iteration learning. One is that both of the two long-distant weight vectors are assigned close values at the initialization of the SOM algorithm and the other is that the winner vectors for input data belonging to the same group happen to be found in two different areas many times. The both of the two situations are common when the number of the weight vectors is large. We think the high performance of our algorithm is on account that there are only four weight vectors at the initialization and that our algorithm always compares only four weight vectors at

a time in searching winner vectors. The subdividing method and the binary search developed for reduction of the computational costs could remove neighborhood learning and tuning of parameters of neighborhood functions and heighten the performance of mapping.

7. Conclusion

In this paper, we evaluated the basic SOM and $O(\log_2 M)$ SOM algorithm using CNS tumor patients dataset comprises five groups. Three kinds of learning rates and two kinds of map sizes are adopted for both algorithm and variable learning rates and neighborhood sizes for the basic SOM algorithm. The simulation results show that our algorithm could map CNS tumor patient dataset more appropriately than the basic algorithm under all kinds of parameters settings. $O(\log_2 M)$ SOM algorithm is able to contribute to various research fields using SOM algorithm.

References

[1] T.Kohonen, Self-Organizing Maps, 2nd ed. New York: Springer Verlag, 1997.
[2] I. Lapidot (Voitovetsky), H. Guterman, and A. Cohen, "Unsupervised speaker recognition based on competition between self-organizing maps", IEEE Trans. Neural Netw, vol. 13, no. 4, pp. 877-887, Jul. 2002.
[3] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks", IEEE Trans. Neural Netw., vol. 16, no. 4, pp.925-936, Jul. 2005.
[4] H.-C. Wang, J. Badger, P. Kearney, and M. Li, "Analysis of codon usage patterns of bacterial genomes using the self-organizing map", Mol. Biol. Evol., pp. 792-800, May 2001.
[5] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps", in Proc. Int. Joint Conf. Neural Networks (IJCNN), 1990, pp. 279-284.
[6] K. K. Truong, "Multilayer Kohonen image codebooks with a logarithmic search complexity", in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process., May 1991, pp. 2789-2792.
[7] P. Xu and C.-H. Chang, "Self-organizing topological tree", in Proc. Int. Symp. Circuits Syst. (ISCAS), 2004, pp.732-735.
[8] H. Kusumot and Y. Takefuji, "$O(\log_2 M)$ self-organizing map algorithm without learning of neighborhood vectors", IEEE Trans. Neural Netw., IN PRESS.
[9] S. L. Pnomeroy, et al., "Prediction of central nervous system embryonal tumour outcome based on gene expression", Nature, vol. 415, pp. 436-442, Jan. 2002.

Hiroki Kusumoto received the B.S. (2001) in Environmental Information and the M.S. (2003) in Media and Governance from Keio University. He is currently a Ph.D. student at Keio University. His research interests include neural networks, clustering algorithms, and hyper-spectral image processing.

Yoshiyasu Takefuji is a tenured professor on faculty of environmental information at Keio University since April 1992 and was on tenured faculty of Electrical Engineering at Case Western Reserve University since 1988. Before joining Case, he taught at the University of South Florida for 2 years and the University of South Carolina for 3 years. He received his BS (1978), MS (1980), and Ph.D. (1983) in Electrical Engineering from Keio University. His research interests focus on neural computing, security, electronic toys. He received the National Science Foundation Research Initiation Award in 1989, the distinct service award from IEEE Trans. on Neural Networks in 1992, the TEPCO research award in 1993, the Takayanagi research award in 1995, the Kanagawa Academy of Science and Technology research award in 1993, the best courseware award from Asia multimedia forum in 1999, the best paper award of Information Processing Society of Japan in 1980, special research award from the US air force office of scientific research in 2003, chairman award from JICA in 2004. He authors 25 books including neural network parallel computing in 1992, and has published more than 200 papers.