

# Signal Self Organizing Map

Chi Kin CHOW and Shiu Yin YUEN

**Abstract**—The self organizing map (SOM) has been applied to wide ranges of fields including computer vision and image processing. Despite of its simple training algorithm, the vectoral input pattern of SOMs induced a sequence of drawbacks which should not be overlooked. These drawbacks include optimal description length selection problem and inaccurate clustering of scattered point patterns. In this article, an extension of SOM to continuous domain, namely signal SOM (SSOM), is proposed to tackle the drawbacks caused by the vectoral input pattern SOMs. Remarkably, it provides an analytical model expression and involves no model selection problem. The SSOM is evaluated by a simulation about clustering of three signal groups. By comparing with the conventional SOM, a more structural map in term of signal group distribution is obtained by the SSOM. Thus, it indicate the contribution of this article on extending the ability of SOM.

## I. INTRODUCTION

Kohonen in 1990 proposed an unsupervised neural network model [1], namely self-organizing map (SOM), which projects high dimensional pattern (*observation*) into a low dimensional grid. The projected *observation* preserves the topological relation of the original *observation*. Hence, similar *observations* in the input space are assigned to the same neuron or nearby neurons of the trained map. This feature enables classification by SOM. The core process of the projection is first, for each *observation*, determining the corresponding winner neuron from the map. The winner neuron is the neuron most similar to the *observation*. Then the process proceeds to update the winner neuron and its neighborhood in order to reduce the differences between those neurons and the *observation*. Due to the simple training algorithm of the SOMs, they are employed in a wide range of applications in engineering. Examples include vector quantization [4] [12], data fusion [5] and computer vision [13]. Recently, applications to other fields have emerged including biometric analysis [6], video coding [11] and electromagnetism [7].

Several articles reported the variants of SOM which

extend the abilities of SOM. Berglund and Sitte [3] proposed a parameterless SOM. The learning rate and neighborhood size of SOM are adaptively calculated based on the local quadratic fitting error of the map to the input space. This adaptive scheme leads to a more effective SOM training algorithm. Yin and Allinson [8] proposed the Bayesian self-organizing map (BSOM) for learning mixtures of Gaussian distributions. The posterior probabilities of the neurons replace the common Euclidean distance winning rule and define explicitly the neighborhood function. The learning can be retained in a small but fixed neighborhood of the winner. The BSOM in turn provides an insight into the role of neighborhood functions used in the common SOM. Hsu [2] reported a generalized self-organizing map model that offers an intuitive method of specifying the similarity between categorical values via distance hierarchies and, hence, enables the direct process of categorical values during training. Yin and Allinson [9] presented a learning algorithm for enhancing the structure of an already trained SOM without retraining the original training data. Kusumoto and Takefuji [10] proposed a training algorithm without the usage of neighborhood functions. This formulation eliminates the computational cost in learning of neighborhood vectors and adjusting the parameters of neighborhood functions. Benson and Hu [14] extended the stochastic gradient method for optimizing an energy function associated with a SOM to asynchronous situations. Voegtlin [15] and Koskela et al. [16] proposed the recursive SOMs (RSOMs) for time series prediction. They combined SOM and a time-delay feedback system and learned to represent sequences recursively.

In the existing SOMs, the patterns are in the form of feature vectors. Though the vectoral representation of pattern simplifies the training algorithm of SOM, it suffers from the selection problem of optimal description length and an inaccurate representation of scattered point patterns. In this article, an extension of SOM to handle signal patterns, namely signal SOM (SSOM), is proposed. The SSOM accepts continuous signals instead of vectors as a training set, which performs pattern clustering in continuous domain. The structure of the SSOM is similar to that of SOM: a set of interconnected neurons which align as a low dimensional grid. Instead of a vector, the weight of a neuron is represented by a continuous signal, namely neuron signal. Given a set of continuous signal  $\mathbf{T} = \{f(\mathbf{x} \in \mathbb{R}^n)\}$ , SSOM aims at aligning a signal set  $\mathbf{T}$  on a low dimensional grid based on their mutual functional distances (Fig. 1).

Chi Kin CHOW and Shiu Yin YUEN are with the department of electronic engineering, City University of Hong Kong, Hong Kong, PRC (tel: 852-27887717; e-mail: {chowchi, kelvinyu}@cityu.edu.hk).

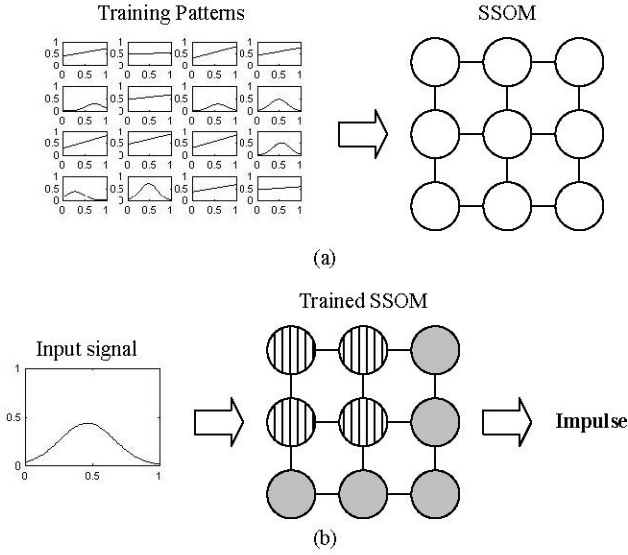


Fig. 1. Objective of the SSOM (a) Training: the 16 signal patterns forms a training set  $T$  of the 3 by 3 SSOM. The patterns in  $T$  belong to either ‘impulse’ or ‘planar’. (b) Testing: the neurons of the trained SSOM can be divided into two groups according to their neuron signals: the vertical-line circles indicate the ‘impulse’ neurons and the gray-filled circles indicate the ‘planar’ neurons. Given an input signal, a winner neuron (see definition 1) is excited and the corresponding group is assigned as the group of the input signal.

Thus, patterns are no longer represented by feature vectors and the drawbacks of vectoral SOM are overcome. Though the RSOMs [15] [16] deals with continuous signal; their objectives are different from the SSOM. The RSOMs accept one continuous signal and the corresponding trained SOM is an alternative representation of this signal. On the other hand, the SSOM considers a set of continuous signals as a training set, and it aims at describing the relations among the signal set in sense of their similarities.

The basic idea of the SSOM is similar to the Volterra series expansion model which was firstly introduced by Vito Volterra [18] in 1959. Volterra series describe the output  $y(t)$  of a nonlinear system of  $x(t)$  as the sum of the responses of a 1<sup>st</sup> order, 2<sup>nd</sup> order, 3<sup>rd</sup> order operators and so on:

$$y(t) = \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \dots, u_n) \prod_{r=1}^n x(t - u_r)$$

Every operator is described in time or frequency domain with a transfer function called Volterra kernel. A Volterra representation can be regarded either as a black-box or a circuit-level description. Black-box Volterra presentation with memory effects is described by Le Gallou et al. [27]. Though a reasonably good correlation between modeled and measured memory is reported, the Volterra description is empirical and must be characterized at the desired operating point.

Zyla and Figueiredo [19] extended the idea of Volterra series to predict the output of a given continuous function. Since [19] does not specify the form of input function, the

formulation is an application-dependent framework. In addition, it may lead to an undetermined model formulation as the integrations of certain functions have no analytic solution. Even the model of Zyla and Figueiredo is derived in continuous domain, Panagiotopoulos et al. in [20] rewrote it in discrete time so as to simplify the complicated expression of functional weights.

By comparing with the similar works of the model of Zyla and Figueiredo, the proposition of SSOM is motivated by different reasons. Firstly, [19] performs regression while the proposed model concerns on clustering. Secondly, the model of Zyla and Figueiredo is derived from the Volterra series mainly works on nonlinear circuit analysis. Though the concept of continuous input was proposed in [19], the actual implementation in [20] still keeps in discrete domain for the simplicity.

The rest of this article is organized as follows: In Section II, we present the essentiality of clustering in continuous domain. Section III reports the details of SSOM. Section IV demonstrates the performances of SSOM by a simulation on distribution clustering. A conclusion is drawn in Section V.

## II. FROM VECTOR TO SIGNAL

In the vectoral SOMs [2] - [14], the description length of *observation* is predefined and non-adjustable during training and prediction phases. Since the size of an accurate description changes from application to application and from *observation* to *observation*, fixed-length description is always insufficient to represent all possible *observation* of an application. A long description provides a detailed description of an *observation* but consumes longer processing time. On the other hand, a short description gives an inaccurate *observation* description but a more generalized network. The determination of the optimal description length remains a challenging problem. Despite that Minimum Description Length (MDL) [21] [22] is a promising guideline to select the size of features; MDL can only return the optimal length among the training set. The length may not be true to the testing set and the problem of description length is not solved.

Sometimes *observation* is described as a parametric distribution or simply a set of scattered points. Unlike the grid-form *observation* that its order can be represented by the raster scanning order, the point-set *observation* cannot be converted to a vector directly. Meanwhile, neither feature extraction nor down sampling scheme is applicable to it. Though any distribution can be modeled as a Gaussian mixture model (GMM) by the EM algorithm [23], the problem of description length exists in the parametric distribution *observation*.

Motivated by the limitations of vectoral *observation*, this article extends the SOM to the continuous domain, namely signal self-organizing map (SSOM), which performs clustering of signal patterns  $\{f(\mathbf{x})\}$ . In SSOM, *observation* is represented as a continuous function  $f(\mathbf{x})$  where  $\mathbf{x} \in \mathbb{R}^n$ . It

is proven in [ 24 ] [ 25 ] that any real function can be approximated as a GMM by giving sufficient number of samples. This leads to a unified expression of SSOM when all functional *observations* are modeled as GMMs. Continuous *observation* avoids the description length selection problem as no quantization is involved. Since the aims of GMM in SSOM is to unify the representation of input signal rather than an information compression, all lossless transformations (i.e. GMM) should return same performance of SSOM. The selection of feature extraction needs not be considered in the performance measurement of SSOM. The EM algorithm makes SSOM valid in clustering of distribution type *observation*.

#### A. Functional distance

Similarity plays an important role in the determination of the winner neuron in SOM, and the Euclidean distance is a common measurement. In this section, Euclidean distance is extended to continuous domain, namely *functional distance*, which is a methodology for measuring the similarity between functions. The two functions  $g(\mathbf{x})$  and  $f(\mathbf{x})$  are represented as GMM (eq. ( 1 ) and eq. ( 2 ) respectively) in the following derivation.

$$g(\mathbf{x}) = \sum_j m_j \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\alpha_j^2}\right) + A \quad (1)$$

and

$$f(\mathbf{x}) = \sum_i w_i \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|^2}{2\sigma_i^2}\right) + B \quad (2)$$

where  $\{m_j\}$  and  $\{w_i\}$  are the kernel weights,  $\{\boldsymbol{\mu}_j\}$  and  $\{\boldsymbol{\mu}_i\}$  are the kernel means,  $\{\alpha_j^2\}$  and  $\{\sigma_i^2\}$  are the kernel variances,  $A$  and  $B$  are the kernel biases and  $\mathbf{x} \in X = \Pi_{k=1}^n [X_k, X_k^+] \subset \mathbb{R}^n$ . The similarity between two functions  $f(\mathbf{x})$  and  $g(\mathbf{x})$ , namely functional distance  $\|f(\mathbf{x}) - g(\mathbf{x})\|_X$ , is defined as the integration of the squared difference over the range  $X$ :

$$\|f(\mathbf{x}) - g(\mathbf{x})\|_X = \int_X (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x} \quad (3)$$

Since we represent any function as a Gaussian mixture model together with the fact that the product of two Gaussian functions remains Gaussian, the *functional distance* can be generalized as:

$$\sum_a C_a \prod_k \int_{X_k^-}^{X_k^+} \exp\left(-\frac{(x_k - l_{a,k})^2}{2q_a^2}\right) dx_k + (A - B)^2 \prod_k (X_k^+ - X_k^-) \quad (4)$$

where  $\{C_a\}$ ,  $\{\mathbf{l}_a = [l_{a,1}, l_{a,2}, \dots, l_{a,n}]\}$  and  $\{q_a\}$  are functions of  $\{m_j\}$ ,  $\{w_i\}$ ,  $\{\boldsymbol{\mu}_j\}$ ,  $\{\boldsymbol{\mu}_i\}$ ,  $\{\alpha_j^2\}$  and  $\{\sigma_i^2\}$ .

#### B. Approximation of Gaussian function integration

As the functional distance involves Gaussian function integration which has no analytical solution, we propose to approximate it as a sigmoid function:

$$\int_a^b \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \approx V(\mu, \sigma, [a, b]) = \frac{\frac{\sqrt{2\pi}}{\sigma}}{1 + \exp\left(-\frac{1.72254(x-\mu)}{0.9772}\right)}$$

Thus, the functional distance between  $f(\mathbf{x})$  and  $g(\mathbf{x})$  can be further rewritten as:

$$\sum_a C_a \prod_k V(l_{a,k}, q_a, [X_k^-, X_k^+]) + (A - B)^2 \prod_k (X_k^+ - X_k^-) \quad (5)$$

To illustrate the accuracy of the approximation, we study the difference between the approximation and the integration derived by numerical analysis (the trapezoidal rule is employed in this section). As the interval is divided into  $10^5$  divisions, the precision of the trapezoidal rule is sufficient to represent the actual integration. Fig. 2(a) shows the approximation of 1D Gaussian integration by the proposed expression where  $\sigma \in [0.1, 5]$ ,  $M \in [-4, 4]$ ,  $a = M\sigma$  and  $b = -4\sigma$ . Fig. 2(b) shows the absolute differences between the approximations of the proposed expression and the trapezoidal rule. The differences keep at a low level (the max. absolute difference is smaller than 0.25).

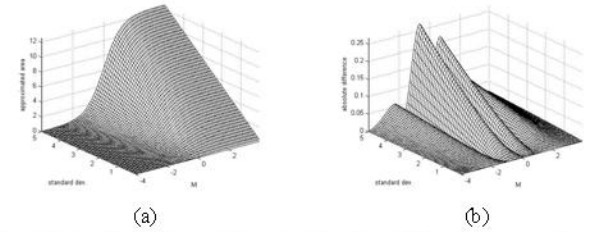


Fig. 2. The integration of Gaussian function (a) by the proposed method and (c) the corresponding error.

### III. SIGNAL SELF-ORGANIZING MAP (SSOM)

SOM is able to project high dimensional *observation* into a low dimensional grid. The projected *observation* preserves the topological relation of the original data. In this article, the SSOM  $\Omega$  is a set of interconnected neurons  $\{N_i\}$ , i.e.  $\Omega = \{N_i\}$ , which partitions a set of continuous signals  $T = \{f(\mathbf{x})\}$  at a low dimensional space given that the number of clusters is not known. The basic mechanism of the SSOM training procedures is similar to that of the conventional SOM (SOM). The two key differences between the SSOM and SOM are: 1) the neuron of SOM is represented as a weight vector  $\mathbf{m}$  on the feature domain. In contrast, the neuron of SSOM is considered as a continuous signal  $s(\mathbf{x})$ , namely neuron signal. The neuron signal of a neuron in SSOM is represented as a weighted sum of the signal input patterns in  $T$ , i.e.  $s(\mathbf{x}) = \sum_j w_j f_j(\mathbf{x})$  where  $w_{j,i}$  is the weight of  $f_j(\mathbf{x})$  on the  $i^{\text{th}}$  neuron. Thus, the functional distance between  $s(\mathbf{x})$  and a signal pattern  $g(\mathbf{x})$  can be rewritten as  $\|\sum_j w_j f_j(\mathbf{x}) - g(\mathbf{x})\|_X$ . As a result, a neuron  $N$  of a SSOM  $\Omega$  is represented by two quantities: neuron signal  $s(\mathbf{x})$  and neuron topology  $\mathbf{r}$ , i.e.  $N = [\{w_j\}, \mathbf{r}]$ . 2) The winner neuron of SOM is defined as the neuron where its weight is closest to the current sample in term of Euclidean distance. Since of the different representation, the functional distance discussed in Section II is employed to measure the similarities between  $N$  and the sample pattern  $g(\mathbf{x})$ .

In this section, we first introduce the initialization of the signal weight. Afterwards, the update rule of neuron signal is presented.

#### A. Initialization of SSOM

The signal weight  $w_j$  of a neuron  $\mathbf{N}$  is initialized as a small number ranged from 0 to 1 for all  $j$ .

#### B. Update rule of SSOM

Similar to the training algorithm of SOM, the elements of SSOM  $\mathbf{\Omega} = \{\mathbf{N}_i = [\{w_{ij}\}, \mathbf{r}_i]\}$  is constructed by competitive learning. Given an arbitrary chosen signal sample  $f_k(\mathbf{x}) \in \mathbf{T}$ , the functional distances between  $f_k(\mathbf{x})$  and the neuron signals  $\{s_i(\mathbf{x})\}$  are computed.

##### Definition 1: Winner neuron of SSOM

The neuron  $\mathbf{N}_c = [\{w_{cj}\}, \mathbf{r}_c] \in \mathbf{\Omega} = \{[\{w_{ij}\}, \mathbf{r}_i]\}$  with the shortest functional distance related to the current signal sample  $f_k(\mathbf{x}) \in \mathbf{T}$  is denoted as the winner neuron, i.e.  $\|\sum_j w_{cj} f_j(\mathbf{x}) - g(\mathbf{x})\|_x \leq \|\sum_j w_{ij} f_j(\mathbf{x}) - g(\mathbf{x})\|_x$ .  $\square$

As the winner neuron is computed, the neuron signals of  $\mathbf{N}_c$  and its neighbors are adjusted towards  $f_k(\mathbf{x})$  by adjusting their signal weights. During the adjustment of neuron signals, the magnitude of the change of  $s_i(\mathbf{x})$  decreases along with time  $t$  and the topological distance related to the winner neuron. The update formula of  $s_i(\mathbf{x})$  is defined as:

$$w_{i,j} \leftarrow \begin{cases} w_{i,j} + h_c \eta_i (1 - w_{i,j}) & \text{if } j = k \\ w_{i,j} & \text{otherwise} \end{cases} \quad (6)$$

where  $h_c$  is the learning rate and  $\eta_i$  is the neighborhood function related to the winner neuron. These two variables monotonically decrease along with  $t$ , and the Gaussian function is employed to model this nature in this article. The learning rate  $h_c$  at time  $t$  is defined as:

$$h_c = H_0 \exp(-t^2 / (2\sigma_h^2)) \quad (7)$$

where  $H_0$  is an arbitrary initial learning rate. The current neighborhood  $\eta_i$  between  $\mathbf{N}_i = [s_i(\mathbf{x}), \mathbf{r}_i] \in \mathbf{\Omega}$  and the winner neuron  $\mathbf{N}_c = [s_c(\mathbf{x}), \mathbf{r}_c]$  is defined as:

$$\eta_i = \exp(-\|\mathbf{r}_c - \mathbf{r}_i\| / (2\sigma_c^2)) \quad (8)$$

$\sigma_c$  is also a monotonically decreasing function:

$$\sigma_c = \sigma_0 \exp(-t^2 / (2\alpha^2)) \quad (9)$$

where  $\sigma_0$  is the degree of neighborhood and  $\alpha$  is the decay constant of  $\sigma_0$ . In this article, the values of  $\sigma_h$  and  $\alpha$  are suggested as a function of the maximum number of training iteration  $t_{max}$ :  $\sigma_h = \alpha = 0.333t_{max}$ , such that the influence of the sample tends to zero at the end of training process. The training process is repeated by arbitrary selecting another pattern from  $\mathbf{T}$  and then updating the SSOM until  $t_{max}$  iterations is completed. The procedures of the algorithm are summarized in the following:

#### SSOM training algorithm

**Input:** 1) a training set  $\mathbf{T} = \{f_i(\mathbf{x} \in \mathbb{R}^n)\}_{i \in [1, k]}$  consists of  $k$  signal patterns and 2) maximum number of iteration  $t_{max}$ .

1. Initialize neuron signals.
2. Accumulate the iteration index  $t \leftarrow t + 1$
3. Randomly select the current signal pattern  $g(\mathbf{x}) \in \mathbf{T}$ .
4. Determine the winner neuron  $\mathbf{N}_c \in \mathbf{\Omega}$
5. Update all neuron signals of  $\mathbf{\Omega}$ .
6. Repeat from step 2 until  $t = t_{max}$ .

**Output:** a trained SSOM  $\mathbf{\Omega}$

#### IV. SIMULATION RESULTS

In this section, the performance of the SSOM is presented for a problem in signal clustering. The signal patterns are from three function groups  $\mathbf{F} = \{\mathbf{P}_i\}_{i=1,2,3}$  including planar functions, oscillated functions and impulse functions:

$$\begin{aligned} \mathbf{P}_1 &= \left\{ p_1(x) = w \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \right\} \\ \mathbf{P}_2 &= \{ p_2(x) = 0.25 \sin(2\pi(x+\theta)) \} \\ \mathbf{P}_3 &= \{ p_3(x) = x(D_h - D_l) + D_l \} \end{aligned}$$

where  $x \in [0, 1]$ ,  $w \in [0.5, 1]$ ,  $\sigma \in [0.1, 0.2]$ ,  $\mu \in [0.35, 0.75]$ ,  $\theta \in [0, 1]$ ,  $D_l \in [0.5, 2]$  and  $D_h \in [0, 0.5]$ . In this simulation, the training set  $\mathbf{T}$  consists of 300 signal patterns. The group and the corresponding parameters of a signal sample in  $\mathbf{T}$  are chosen randomly.

A SSOM of 5 by 5 neurons are employed in this simulation. The values of  $H_0$  and  $\sigma_0$  are chosen as 0.05 and 0.72 respectively. The training process is terminated after 5000 iterations. Thus, the values of  $\sigma_h$  and  $\alpha$  are assigned as 1666. In order to illustrate the essentiality of the SSOM, SOM is also applied to the same training set  $\mathbf{T}$ . Since SOM accepts only fixed length vectoral pattern,  $\mathbf{T}$  must be parameterized as  $\mathbf{T}_c = \{Q(g(x) \in \mathbf{T}, l)\}$  where  $l$  is the description length and the operator  $Q(\cdot)$  denotes the parameterization from a GMM to its feature vector. Given a GMM:  $f(\mathbf{x}) = \sum_j w_j \exp(-\|\mathbf{x} - \boldsymbol{\mu}_j\| / (2\sigma_j^2))$ , the feature vector of  $f(\mathbf{x})$ ,  $G(f(\mathbf{x}))$ , is defined as  $[\{w_j\}, \{\boldsymbol{\mu}_j\}, \{\sigma_j\}]_{j \in [1, J]}$ . We either insert redundant kernel(s) or prune the least significant kernel(s) (the kernel with the smallest absolute volume) in order to maintain fixed length feature vector. The value of  $l$  in this simulation is chose as 7.

Fig. 3(a) shows the neuron signals of the trained SSOM. The alignment of the neuron signals is in the same order of the neuron topology. Seen from the figure, the forms of the neuron signals are similar to their neighborhood such that similar signals fall into the nearby neurons. For the case of SOM, the neuron signal is equivalent to the GMM represented by the corresponding weight vector. Fig. 3(b) shows the trained SOM in the same illustration of Fig. 3(a).

The generalization ability of the SSOM is shown by a testing set  $\mathbf{E}$  consisting of 9000 signal patterns randomly selected from  $\mathbf{F}$ . A vote is added to the neuron  $\mathbf{N}_i$  if  $\mathbf{N}_i$  is the winner neuron of a testing signal from  $\mathbf{E}$ . Thus, 9000 votes

are added to the trained SSOM and each neuron consists of three voting options. Fig. 4(a) shows the vote result of the trained SSOM. Each pie chart represents the relative vote fractions of a neuron. The dark grey, light grey and white regions of a pie chart represent the relative vote fractions of  $P_1$ ,  $P_2$  and  $P_3$  respectively. The circles with dotted outline represent the zero-vote neurons. Fig. 4(b) shows the voting result of the trained SOM. Seen from fig. 4(a), most neurons consist of unique or dominated votes that the votes of a neuron are come from the testing patterns of same function group. In addition, the neurons with the same dominated votes assemble and form signal clusters. Empirically, these two observations infer the successful signal clustering by the SSOM. On the other hand, SPM consists of more impure-voting neurons that the clusters of  $P_1$  and  $P_2$  have more overlapping area.

We define  $n_k$  as the number of Gaussian kernels of a signal pattern. Table I lists the distributions of  $n_k$  of the training set  $T$  and the testing set  $E$  respectively. Seen from the table, the values of  $n_k$  are different from patterns to patterns. Especially there exist some patterns in  $E$  that the corresponding  $n_k$  are larger than the maximum  $n_k$  in  $T$ . Thus, it infers the two key issues pointed out in this article: 1) the problem of optimal description length selection in vectoral SOM and 2) the essentiality of clustering patterns in the continuous domain.

TABLE I

DISTRIBUTIONS OF $n_k$ : (A) TRAINING SET $T$ AND (B) TESTING SET $E$				
	4-6	7-9	10-12	> 12
$T$	13	20	17	0
$E$	147	623	180	50

## V. CONCLUSION

Self-organizing map (SOM) is a neural network model which projects high dimensional pattern into a low dimensional grid. The projected data preserves the topological relation of the original data. To simplify the training process, *observation* is commonly represented as a vector after quantization or feature extraction. However, these simplifications lead to a sequence of limitations on *observations*. During the quantization, the size of quantized vector (the description length) plays an important role to the network complexity and accuracy. A long description *observation* forms a complicated and accurate network while a short description *observation* provides a simple but inaccurate network. Furthermore, as the distribution form *observation* involves no ordering, none of the quantization methods is applicable. Though point distribution can be represented by GMM through EM algorithm, the problem of description length selection still exists. Furthermore, vector is an inaccurate representation of distribution-form *observation*.

Driven by the limitations of vectoral *observations*, we propose an extension of SOM called signal SOM (SSOM)

which performs clustering of a continuous signal set. This feature provides a more accurate description to the relations among *observations*. As a result, SSOM avoids the following problems: 1) selection of optimal description length and 2) inaccurate clustering of scattered points (distribution) patterns. The simulation results empirically verify the contributions of the SSOM on overcoming the limitations of vectoral input patterns.

## ACKNOWLEDGMENT

The work described in this article was fully supported by a grant from CityU (7001707).

## REFERENCES

- [1] T. Kohonen, "The self-organizing map", *Proc. IEEE*, vol. 78, no. 9, pp. 1464-1480, Sept. 1990.
- [2] Chung-Chian Hsu, "Generalizing self-organizing map for categorical data", *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 294 - 304, March 2006.
- [3] Berglund E. and Sitte J., "The parameterless self-organizing map algorithm", *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 305 - 316, March 2006.
- [4] Chip-Hong Chang, Pengfei Xu, Rui Xiao and Srikanthan T., "New adaptive color quantization method based on self-organizing maps", *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 237 - 249, Jan. 2005.
- [5] Weijian Wan and Fraser D., "Multisource data fusion with multiple self-organizing maps", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, part 1, pp. 1344 - 1349, May 1999.
- [6] Matsopoulos G. K., Asvestas P. A., Mouravliansky N. A. and Delibasis K. K., "Multimodal registration of retinal images using self organizing maps", *IEEE Transactions on Medical Imaging*, vol. 23, no. 12, pp. 1557 - 1563, Dec. 2004.
- [7] Igarashi H., "Visualization of optimal solutions using self-organizing maps in computational electromagnetism", *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1816 - 1819, May 2005.
- [8] Yin H. and Allinson N. M., "Bayesian self-organising map for Gaussian mixtures", *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 4, pp. 234 - 240, Aug. 2001.
- [9] Yin H. and Allinson N. M., "Interpolating self-organising map (iSOM)", *Electronics Letters*, vol. 35, no. 19, pp. 1649 - 1650, Sept. 1999.
- [10] Kusumoto H. and Takefuji Y., "O(log<sub>2</sub>M) Self-Organizing Map Algorithm Without Learning of Neighborhood Vectors", *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1656 - 1661, Nov. 2006.
- [11] Yu-Len Huang and Hsiu-Yi Lien, "Temporal error concealment for MPEG coded video using a self-organizing map", *IEEE Transactions on Consumer Electronics*, vol. 52, no. 26, pp. 676 - 681, May 2000.
- [12] Kim Y. K. and Ra J. B., "Adaptive learning method in self-organizing map for edge preserving vector quantization", *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 278 - 280, Jan. 1995.



- [13] Singh R., Cherkassky V. and Papanikolopoulos N., "Self-organizing maps for the skeletonization of sparse shapes", *IEEE Transactions on Neural Networks*, vol. 11, no. 1 pp. 241 – 248, Jan. 2000.
- [14] Benson M. W. and Jie Hu, "Asynchronous self-organizing maps", *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1315 – 1322, Nov. 2000.
- [15] Thomas Voegtlin, "Recursive self-organizing maps", *Neural Networks*, vol. 15, pp. 979 – 991, 2002.
- [16] Koskela, T., Varsta, M., Heikkonen, J., and Kaski, K., "Time series prediction using recurrent SOM with local linear models", *International Journal of Knowledge-based Intelligent Engineering System*, 2(1), pp. 60 – 68, 1998
- [17] V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.
- [18] Volterra V., *Theory of Functionals and of Integral and Integro-Differential equations*, New York: Dover.
- [19] L. V. Zyla and R. J. P. de Figueiredo, "Nonlinear system identification based on a Fock space framework", *SIAM J. Contr. Optimization*, vol. 21, no. 6, pp. 931 – 939, Nov. 1983.
- [20] A. Panagiotopoulos, R. W. Newcomb and S. K. Singh, "Planning with a Functional Neural-Network Architecture", *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 115- 127, Jan. 1999.
- [21] M. Small and C. K. Tse, "Minimum description length neural networks for time series prediction", *Phys. Rev. E, American Physical Soc.*, College PK, USA, Dec. 2002, 066701.
- [22] K. Judd and A. Mees, "On selecting models for nonlinear time series", *Physica D*, Elsevier Science BV, Amsterdam, Netherlands, May 1995, pp. 426 - 444.
- [23] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm", *J. Royal Statistic Soc.*, vol. 30, no. B, pp. 1 - 38, 1977.
- [24] B. Liu and J. Si, "The best approximation to C functions and its error bounds using regular-center Gaussian networks", *IEEE Trans. Neural Networks*, vol. 5, pp. 845 – 847, 1994.
- [25] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations", *Neural Computation*, vol. 2, no. 2, pp. 210–215, 1990.
- [26] Kai Tik Chow D. and Tong Lee, "Image approximation and smoothing by support vector regression", *Proceedings of International Joint Conference on Neural Networks 2001*, vol. 4, pp. 2427 - 2432.
- [27] Le Gallou N., Ngoya E., Buret H., Barataud D. and Nebus J. M., "An improvement behavioral modeling technique for high power amplifiers with memory", *Proceeding of the IEEE International Microwave Symposium*, Phoenix, Arizona, 2000.
- [28] W. Lam and F. Bacchus, "Learning Bayesian belief networks: an algorithm based on the MDL principle", *Computational Intelligence*, vol.10, no.4, pp. 269 – 293, 1994.

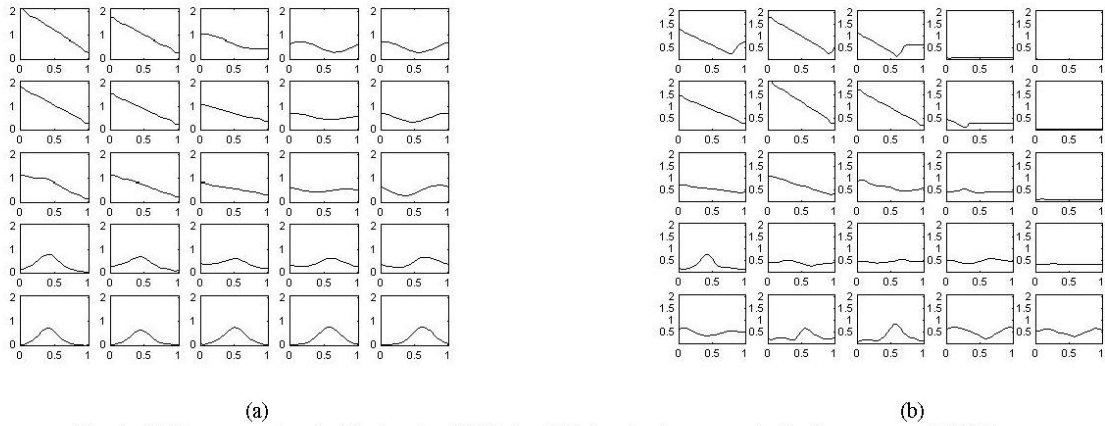


Fig. 2. (a) The neuron signal of the trained SSOM and (b) the signals represented by the neurons of SOM



Fig. 3. Voting results of Section IV: (a) SSOM and (b) conventional SOM.