Supervised Kernel Self-Organizing Map

Dongjun Yu[1,2], Jun Hu[1], Xiaoning Song[3], Yong Qi[1,2], and Zhenmin Tang[1]

[1] School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing 210094, China
[2] Changshu Institute, Nanjing University of Science and Technology,
Changshu 215500, China
[3] School of Computer Science and Engineering,
Jiangsu University of Science and Technology, Zhenjiang 212003, China
njyudj@njust.edu.cn

Abstract. We generalize the traditional supervised self-organizing map to supervised kernel self-organizing map by incorporating the kernel function to further improve its capability of solving non-linear problems. The kernel function maps the low-dimensional input space to high-dimensional feature space thus potentially makes the complex non-linear structure in the input space much easier in the mapped feature space. Qualitative and quantitative analysis of the experimental results on the two benchmark datasets illustrate the effectiveness of the proposed method.

Keywords: Self-organizing Map, Kernel Function, Supervised Learning, Non-linear System.

1 Introduction

The Self-Organizing Map (SOM) [1] proposed by Kohonen has been widely used in many fields such as pattern recognition [2, 3], data mining [4] and clustering analysis [5]. The traditional SOM is a typical unsupervised learning model and much effort has been made to facilitate the SOM to suit for supervised learning tasks. One of the most straightforward methods to construct supervised SOM (SSOM) was proposed by Barreto et al. in [6].

In SSOM [6], each training sample consists of two parts, which correspond to the input and the desired output of a input-output mapping, respectively. Accordingly, the weight of each output node of SSOM is divided into two parts. During the training stage, the first parts of training samples and weights are used to locate the best matching unit (BMU), while the second parts of training samples are used as teachers to guide the learning of the second parts of weights.

However, in SSOM [6], the Euclidean distance metric is taken for locating the BMU, thus the effectiveness will be significantly affected by the nonlinearity of the underlying data. In this paper, we generalize the SSOM to supervised kernel self-organizing map (SKSOM) by incorporating the kernel function into the SSOM. The essence of SKSOM is to map the low-dimensional input space to a high-dimensional feature space and potentially convert the complex nonlinear problem much easier in

J. Yang, F. Fang, and C. Sun (Eds.): IScIDE 2012, LNCS 7751, pp. 246–253, 2013.
© Springer-Verlag Berlin Heidelberg 2013

the mapped space. Experimental results on two benchmark datasets demonstrate the effectiveness of the proposed method.

2 Supervised Kernel Self-Organizing Map (SKSOM)

Let $\Phi : \mathbf{a} \in L \to \Phi(\mathbf{a}) \in F$ be a non-linear mapping, L be the original input space, F be the mapped high-dimensional space. We defined the distance metric in the mapped space as:

$$KD(\mathbf{a},\mathbf{b}) = \left\| \Phi(\mathbf{a}) - \Phi(\mathbf{b}) \right\|^2 \tag{1}$$

The kernel function satisfies the Mercer condition is defined as:

$$K(\mathbf{a},\mathbf{b}) = \Phi(\mathbf{a})^T \Phi(\mathbf{b}) \tag{2}$$

Thus, Eq. (1) can be further formularized as:

$$KD(\mathbf{a},\mathbf{b}) = \left\| \Phi(\mathbf{a}) - \Phi(\mathbf{b}) \right\|^2 = K(\mathbf{a},\mathbf{a}) + K(\mathbf{b},\mathbf{b}) - 2K(\mathbf{a},\mathbf{b}) \tag{3}$$

Let $X = \{\mathbf{x}(i)\}_{i=1}^{N}$ be the training dataset, and each training sample $\mathbf{x}(i)$ consists of two parts as follows:

$$\mathbf{x}(i) = \begin{pmatrix} \mathbf{x}^{in}(i) \\ \mathbf{x}^{out}(i) \end{pmatrix} \tag{4}$$

where $\mathbf{x}^{in}(i)$ be the input part, $\mathbf{x}^{out}(i)$ be the desired output.

Accordingly, the weight of the output node j $(1 \le j \le K)$ also consists of two parts:

$$\mathbf{w}_j = \begin{pmatrix} \mathbf{w}_j^{in} \\ \mathbf{w}_j^{out} \end{pmatrix} \tag{5}$$

where K is the total number of output nodes.

Let

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{x}_t^{in} \\ \mathbf{x}_t^{out} \end{pmatrix} \in X \tag{6}$$

be the chosen training sample at the learning step t.

Then, the BMU is located by the following formula:

$$j^* = \arg\min_{1 \le j \le K} \{KD(\mathbf{x}_t^{in}, \mathbf{w}_j^{in}(t))\} \tag{7}$$

where $KD(\mathbf{x}_t^{in}, \mathbf{w}_j^{in}(t))$ denotes the kernel distance between $\mathbf{x}_t^{in}$ and $\mathbf{w}_j^{in}(t)$ as defined in Eq.(3).

After locating the BMU, the input and output parts of the weights of the BMU and its neighboring nodes are updated as follows:

$$\mathbf{w}_i^{in}(t+1) = \mathbf{w}_i^{in}(t) + \alpha(t)h(j^*, i; t)\frac{\partial KD(\mathbf{x}_t^{in}, \mathbf{w}_i^{in}(t))}{\partial \mathbf{w}_i^{in}(t)} \tag{8}$$

$$\mathbf{w}_i^{out}(t+1) = \mathbf{w}_i^{out}(t) + \alpha(t)h(j^*, i; t)\frac{\partial KD(\mathbf{x}_t^{out}, \mathbf{w}_i^{out}(t))}{\partial \mathbf{w}_i^{out}(t)} \tag{9}$$

where $\alpha(t)$ is the learning rate, and $h(j^*, i; t)$ is the neighborhood function. Different neighborhood function can be applied. For example, a Gaussian neighborhood function is formularized as:

$$h(j^*, i; t) = \exp\left(-\frac{\| r_i(t) - r_{j^*}(t) \|^2}{2\sigma(t)^2}\right) \tag{10}$$

where $r_i(t)$ and $r_{j^*}(t)$ are the coordinates of the output node i and j^* on the output layer, respectively.

Obviously, the updating equations (8) and (9) will be different when applying different kernel function $K(\cdot,\cdot)$. Taking radial basis kernel function as an example:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2} \tag{11}$$

Then, Eq. (8) can be further formularized as follows：

$$\begin{aligned}
\mathbf{w}_i^{in}(t+1) &= \mathbf{w}_i^{in}(t) + \alpha(t)h(j^*, i; t)\frac{\partial KD(\mathbf{x}_t^{in}, \mathbf{w}_i^{in}(t))}{\partial \mathbf{w}_i^{in}(t)} \\
&= \mathbf{w}_i^{in}(t) + \alpha(t)h(j^*, i; t)\frac{\partial\left(K(\mathbf{x}_t^{in}, \mathbf{x}_t^{in}) + K(\mathbf{w}_i^{in}(t), \mathbf{w}_i^{in}(t)) - 2K(\mathbf{x}_t^{in}, \mathbf{w}_i^{in}(t))\right)}{\partial \mathbf{w}_i^{in}(t)} \\
&= \mathbf{w}_i^{in}(t) + \alpha(t)h(j^*, i; t)\frac{\partial\left(-2K(\mathbf{x}_t^{in}, \mathbf{w}_i^{in}(t))\right)}{\partial \mathbf{w}_i^{in}(t)} \\
&= \mathbf{w}_i^{in}(t) - 2\alpha(t)h(j^*, i; t)\frac{\partial e^{-\|\mathbf{x}_t^{in} - \mathbf{w}_i^{in}(t)\|^2/2\sigma^2}}{\partial \mathbf{w}_i^{in}(t)} \\
&= \mathbf{w}_i^{in}(t) - \frac{2\alpha(t)h(j^*, i; t)e^{-\|\mathbf{x}_t^{in} - \mathbf{w}_i^{in}(t)\|^2/2\sigma^2}}{\sigma^2}\left(\mathbf{x}_t^{in} - \mathbf{w}_i^{in}(t)\right)
\end{aligned} \tag{12}$$

Similarly, Eq. (9) can be further formularized as:

$$\mathbf{w}_i^{out}(t+1) = \mathbf{w}_i^{out}(t) - \frac{2\alpha(t)h(j^*, i; t)e^{-\|\mathbf{x}_t^{out} - \mathbf{w}_i^{out}(t)\|^2/2\sigma^2}}{\sigma^2}\left(\mathbf{x}_t^{out} - \mathbf{w}_i^{out}(t)\right) \tag{13}$$

3 Experimental Results and Analysis

3.1 Sunspot Dataset

Sunspots are temporary phenomena on the photosphere of the Sun that appear visibly as dark spots compared to surrounding regions. The frequency and intensity will

significantly affect the earth environment. Thus, accurately predicting the trend of sunspot is of great significance for human life and activities. In this study, we obtained the sunspot dataset from SIDC-Solar Influences Data Analysis Center (http://sidc.oma.be/sunspot-data/) [7]. Note that the yearly sunspot dataset (yearssn.dat) was taken as the benchmark dataset.

3.2 Nonlinear System Identification

The nonlinear system [8] defined as follows was taken to generate the second benchmark dataset:

$$y(t+1) = \frac{y(t)y(t-1)(y(t)+2.5)}{1+y^2(t)+y^2(t-1)} + u(t) \tag{14}$$

where $u(t) = \sin(2\pi t / 25)$ is the activation function, and the initial values of the nonlinear system are as follows:

$$y(-1) = 0.5, y(0) = 0.9 \tag{15}$$

3.3 Results and Analysis

Experimental Parameters Configuration

The maximal iteration is set to be $T_{\max} = 20000$, the width of Gaussian kernel function $\sigma = \sqrt{2}$. The initial and end values of learning rate are $\alpha_0 = 0.8$ and $\alpha_T = 0.03$, respectively. In addition, the learning rate is monotonically decreasing with t according to Eq. (16):

$$\alpha(t) = \alpha_0(\alpha_T / \alpha_0)^{t/T_{\max}} \tag{16}$$

Square neighborhood function is applied. Let *outputwidth* be the width of the output layer, then $h(t)$, the width of square neighborhood at learning step t, is defined as:

$$h(t) = \left\lfloor \frac{1}{2} \times outputwidth \times (1 - t/T_{\max}) \right\rfloor \tag{17}$$

The weights of the output nodes are randomly initialized with values among interval (-1, 1).

Data Pre-processing and Sample Generation

As the values of sunspot dataset are much bigger and there may exist noises, we pre-processed the sunspot dataset as follows:

For a sunspot time series $x_1, x_2, \cdots,$ and x_n, we first normalize the time series using the following equation:

$$y_i = \frac{x_i - \min\limits_{1 \le j \le n}\{x_j\}}{\max\limits_{1 \le j \le n}\{x_j\} - \min\limits_{1 \le j \le n}\{x_j\}} \tag{18}$$

Then, we perform the median filter on the normalized time series $\{y_i\}_{i=1}^{n}$ and the obtained time series are used.

On both the two benchmark datasets, 100 time series values are taken to generate training and testing samples. The first 50 values are used to generate training samples and the remaining 50 values are used to generate testing samples. Note that, we use 3 previous values to predict the next value, i.e., the dimensionality of the input is 3 while the dimensionality of the output is 1.

To comprehensively investigate the influence of output size on the performance of SKSOM, we also carried out experiments on both benchmark datasets with different output sizes, i.e., 8×8 and 16×16. Fig. 1 and 2 illustrate the performance of SKSOM on sunspot dataset with output sizes 8×8 and 16×16, respectively; while Fig. 3 and 4 show the performance of SKSOM on the nonlinear system as described in (14) with output sizes 8×8 and 16×16, respectively.

Fig.1 ~ 4 qualitatively illustrates the good performance of SKSOM. From Fig.1~4, we can find that the proposed SKSOM not only performs well on the training dataset, but also has good generalization ability on testing dataset. In addition, we also quantitatively investigate the performance of the proposed SKSOM by comparing it with the SSOM with different output sizes. Each experiment was independently performed 10 times and the averaged results were then reported in Table 1 and 2.

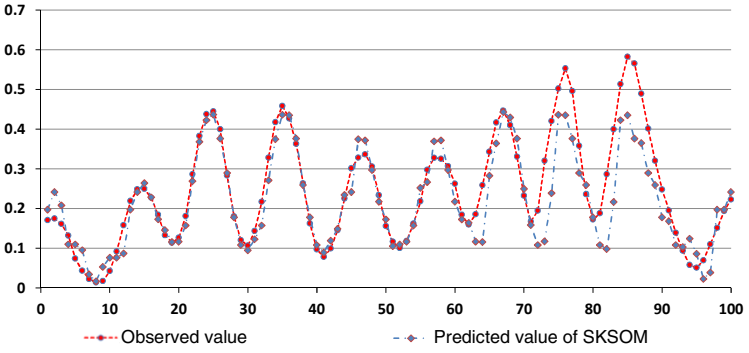

Fig. 1. Performance of SKSOM on sunspot dataset with output size 8×8

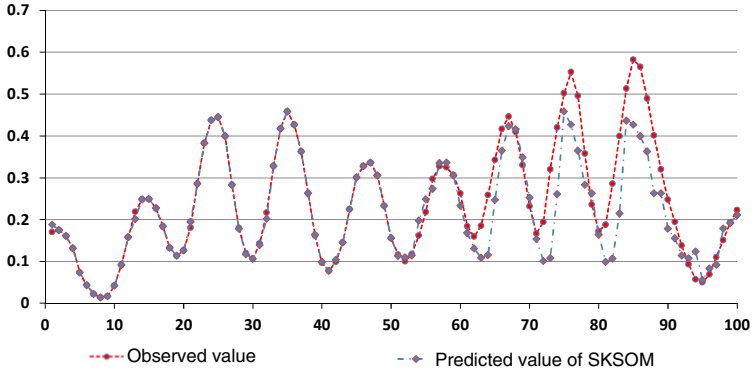

Fig. 2. Performance of SKSOM on sunspot dataset with output size 16×16

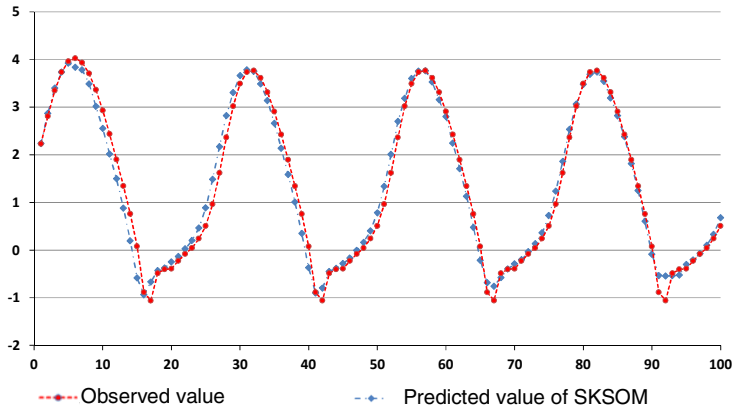

Fig. 3. Performance of SKSOM on nonlinear system with output size 8×8

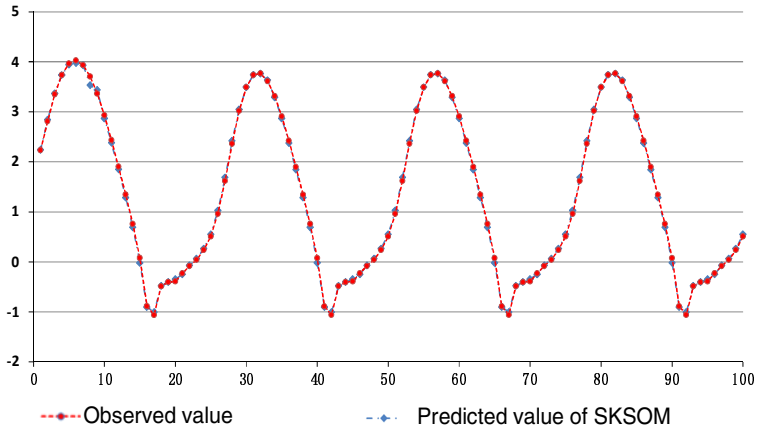

Fig. 4. Performance of SKSOM on nonlinear system with output size (16×16)

Table 1. Performance comparison between SSOM and SKSOM on sunspot dataset

		RMSE	
	Output size	8×8	16×16
SSOM	Error of training samples	0.001619	0.000057
	Error of testing samples	0.007872	0.007685
	Averaged error	0.004745	0.003871
SKSOM	Error of training samples	0.001318	0.000033
	Error of testing samples	0.007681	0.007309
	Average error	0.004500	0.003671

Table 2. Performance comparison between SSOM and SKSOM on non-linear system dataset

		RMSE	
	Output size	8×8	16×16
SSOM	Error of training samples	0.011752	0.000198
	Error of testing samples	0.017578	0.000203
	Averaged error	0.014665	0.000201
SKSOM	Error of training samples	0.011324	0.000070
	Error of testing samples	0.016563	0.000067
	Average error	0.013944	0.000069

By carefully observing Table 1 and 2, the performances of both SSOM and SKSOM increase when the output size is enlarged from 8×8 to 16×16. We explain this phenomenon as follows: when the output size increases, the number of weights connecting the input layer and the output layer will increase accordingly thus can more effectively learn the data distribution of the training samples. While the less number of weights will results in a much rough insight of the data distribution of the training samples thus leading to a poor prediction performance. In addition, we also found that the SKSOM consistently outperforms SSOM on both benchmark datasets. We speculate that the major reason is that in SKSOM the kernel function effectively maps the non-linear structure in the low-dimensional input space to an easier solving structure in the high-dimensional feature space.

4 Conclusion

By incorporating kernel function, we have generalized the supervised self-organizing map (SSOM) to the supervised kernel self-organizing map (SKSOM). On the one hand, the SKSOM inherits the good advantages of SSOM; on the other hand, its capability of solving non-linear problems has been improved by introducing kernel function into SSOM. Experimental results on sunspot and non-linear system datasets demonstrated the effectiveness of the proposed method.

Acknowledgments. This work was supported by the Natural Science Foundation of Jiangsu (Grant No. BK2011371), the Jiangsu Postdoctoral Science Foundation (Grant No. 1201027C), and the Industry-Academia Cooperation Innovation Fund Projects of Jiangsu Province (Grant No. BY2012022).

References

1. Kohonen, T.: Self-organizing map. Proc. IEEE 78, 1464–1480 (1990)
2. Kohler, A., Ohrnberger, M., Scherbaum, F.: Unsupervised pattern recognition in continuous seismic wavefield records using Self-Organizing Maps. Geophysical Journal International 182(3), 1619–1630 (2010)

3. Yu, D.J., Shen, H.B., Yang, J.Y.: SOMRuler: A Novel Interpretable Transmembrane Helices Predictor. IEEE Transactions on Nanobiosci. 10(2), 121–129 (2011)
4. Fan, C.Y., Fan, P.S., Chan, T.Y., Chang, S.H.: Using hybrid data mining and machine learning clustering analysis to predict the turnover rate for technology professionals. Expert Systems with Applications 39(10), 8844–8851 (2012)
5. Liu, Y.C., Wu, C., Liu, M.: Research of fast SOM clustering for text information. Expert Systems with Application 38(8), 9325–9333 (2011)
6. Barreto, G.A., Aluizio, A.F.R.: Identification and control of dynamical systems using the self-organizing map. IEEE Transactions on Neural Networks 15(5), 1244–1259 (2004)
7. Podladchikova, T., Van der Linden, R.: A Kalman Filter Technique for Improving Medium-Term Predictions of the Sunspot Number. Solar Physics 277(2), 397–416 (2012)
8. Lin, C.T. (ed.): Neural Fuzzy Systems. Prentice-Hall Press, New York (1997)