Self-adaptive Online Sequential Learning Radial Basis Function Classifier Using Multi-variable Normal Distribution Function

Keming Dong*, Hyoung Joong Kim* and S.Suresh

Abstract

Online or sequential learning is one of the most basic and powerful method to train neuron network, and it has been widely used in disease detection, weather prediction and other realistic classification problem.
At present, there are many algorithms in this area, such as MRAN, GAP-RBFN, OS-ELM, SVM and SMC-RBF. Among them, SMC-RBF has the best performance; it has less number of hidden neurons, and best efficiency. However, all the existing algorithms use signal normal distribution as kernel function, which means the output of the kernel function is same at the different direction. In this paper, we use multi-variable normal distribution as kernel function, and derive EKF learning formulas for multi-variable normal distribution kernel function. From the result of the experience, we can deduct that the proposed method has better efficiency performance, and not sensitive to the data sequence.

Keywords: Online learning, Resource allocation network(RAN), Multi-variable normal distribution function.

I. Introduction

In some real world classification problem such as weather prediction, disease detection, you can't get all the training data at one time. However, to the classic machine learning algorithm, you can start training the network only when you get enough data, and if there is more data, then you should retrain the network using all the data. Because of this, the concept of online learning is proposed. Online learning can be summarized as the model which learns one instance (or one set of instances) at a time. In this case, the leaning network can "grow" one training data by one training data.

Among various classification methods, neural network has showed a very good performance on many classification problems. Because of the simple architecture and good performance, RBFN is very popular among all neural network algorithms.

RBF has three layers: the input layer, hidden layer and output layer. Every input neuron just transmits its input value to all the hidden neurons. In every hidden neuron, there is a Gaussian Function whose input is the input feature vector, and output is the result of Gaussian Function. Then the hidden neurons transmit its value to every output neuron multiplying a weight vector. At last, every output neuron sums up all the outputs of every hidden neuron. The architecture of RBF network is shown in Figure 1

In classical RBF networks, the number of hidden neurons should be input before training, then the network uses K-means clustering algorithm to get the right parameter values of expect values μ and variance σ of every Gaussian Function. Then using Least Square method, it can estimate right weight values of the interconnection between every hidden neuron and output neuron. In the Resource Allocation Network (RAN), you don't have to input the number of hidden neurons before training; it will decide to add neuron according to the error computed using the new training sample with the system. Replacing LS method with Kalman Filter, it changes to RAN-EKF. MRAN just add pruning method to RAN. To make the algorithm more quickly, EMRAN introduces winner neuron concept, i.e., the nearest neuron from the current training sample, only the winner neuron's parameters will be updated by EKF. The Growing And Pruning RBFN (GAP-RBFN) uses the distance between the current sample and the nearest neuron in the same class and error as adding condition, and in learning part, only the nearest neighbor neuron can be updated by EKF (Extend Kalman Flitter). If the significance of nearest neighbor neuron is less than one threshold, then it can be pruned away from network. Recently S.Suresh has developed a sequential algorithm called Sequential Multi-Category Classifier using Radial Basis Function (SMC-RBF). This algorithm is better than other algorithms considerably. The efficiency is higher but the number of hidden neuron is less than the existing algorithms. It predicts class for every example, depending on the prediction and the error the network make the network make the decision to add or to learn.

Although the performance of all the existing RBF algorithms is good enough for application, but they all have some public problems. One problem is that all the existing algorithms use signal normal distribution function as kernel function, so if two different examples have the same distance to the center of one neuron then the output of the neuron is same, which means there is no difference between this two different examples to this neuron, but it is not true. The second problem is that all the existing algorithms have sequence learning problem, which means that the performance is sensitive to the data sequence. The third problem is that you should configure so many parameters before you apply the algorithm to classifica-

* 고려대학교 정보경영공학전문대학원
(kevindong@korea.ac.kr, khj-@korea.ac.kr)
** department of electrical engineering, Indian Institute of Technology – Delhi, India, (Sundaram.Suresh@hotmail.com)

tion problem, in other words, it is not self-adapting.

Because of these problems, we propose a new algorithm called SOSM-RBF, which is the multi-variable normal distribution version of SOS-RBF. In this method, we replace the signal normal distribution with multi-variable normal distribution function as kernel function, and find the corresponding extend Kalman Filter formula for the multi-variable distribution kernel function. We also find a new leaning and adding mechanism in order to solve the sequence problem and self-adapting problem. From the result, we can conclude that the proposed method successfully solve the self-adapting problem, and is better than the other algorithms on sequence problem.

The organization of this paper is as follows: Section 2 describes the new Self-adaptive Online Sequential learning RBF Classifier using Multi-variable normal distribution(SOSM-RBF). Section 3 performance comparison results for SAG-RBF along with RAN, MRAN, OS-ELM, SVM and SMC-RBF on the criterions of efficiency, number of hidden neurons, Self-Adapting Growing ability and performance on random sequential training samples. Section 4 summarizes the conclusions from this study.

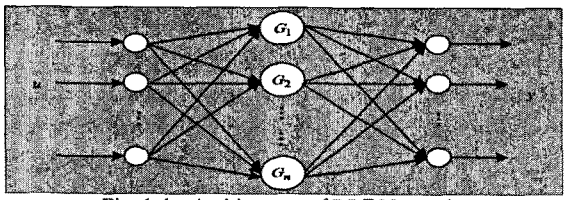

Fig. 1 the Architecture of RBF Network

II. SOSM-RBF algorithm

In this section, we firstly describe some concepts in multi-category classification problem, and then provide the mathematical model of RBF. After that, the main ideas behind SOSM-RBF algorithm are described.

The learning mechanism of human can be briefly reduced to reduction and inference, and multi-classification problem is a very common form of reduction, so multi-classification makes great sense in machine learning.

In order to divide the samples to different classes, we should extract some features which have relationship with criterion for classification. So to classier in machine learning, we use some features to stand for every sample. Let the training sample is$(\mathbf{x_i}, \mathbf{y_i})$, i=1, 2... N. $\mathbf{x_i}, \mathbf{y_i}$ stands for the input features and class label of i^{th} samples. If the order is not important, we just use **x**,**y**..

Before using training data, we firstly normalize value range of **x** to [-1, 1]. If the sample belongs to j^{th} class, then the j^{th} element of **y** is one and all others are -1.

Generally speaking, the model of RBF network can be generalized as follows:

$$f(x) = \alpha_0 + \sum_{k=1}^{K} \alpha_k \Phi_k(x), \quad (1)$$

Where $\Phi_k(x)$ is the response of the k^{th} hidden neuron to the input x and α_k is the weight connecting the k^{th} hidden unit to the output unit. $\Phi_k(x)$ is a gaussian function given by

$$\Phi_k(x) = \exp\left(-\frac{1}{\sigma_k^2}\|x - \mu_k\|\right) \quad (2)$$

Where μ_k is the k^{th} neuron center and σ_k is the covariance matrix of the k^{th} neuron.

The predicted class label \^c for the new training sample is given by

$$\hat{c} = \arg\max_{i \in 1,\ldots,n_c} \hat{y}_i \quad (3)$$

1. The Idea of the Algorithm

Dropping Condition: if $\hat{c} \neq c$ and $E \leq \varepsilon_a$, then drop this sample, i.e., do not use this example. If new sample satisfies such a condition, then it shows that the current system is efficient enough, so we don't have to use this sample. Adding the dropping mechanism, we can prevent overtraining problem and also because we don't use this sample, so the time for training is less.

Adding Condition: The one of the following conditions must be satisfied for an observation (x_i, y_i) to be used to add a new hidden unit to the network:

$$\begin{cases} \hat{c} \neq c \\ E \geq \varepsilon_a \end{cases} \quad (4)$$

Where E is the maximum error in the current sample and ε_a is the threshold for adding neuron. Since, the coded class label y and its predicted value ($\hat{y}$) are within the limit of +1 or -1, the threshold is initialized at maximum value of 1.5. Here, the threshold is adapted based on the in incremental knowledge added to the network, i.e.,

$$\varepsilon_a = \sigma\varepsilon_a - (1 - \sigma)E \quad (5)$$

Where σ the control parameter, usually kept very close to one.

And the new hidden neuron will be as follows:

$$\begin{cases} \omega_{new} = e \\ \mu_{new} = x \\ \Sigma_{new} = \kappa\|x_i - \mu_{nr}^c\| * I_{nx*nx} \end{cases} \quad (6)$$

Where κ is a positive constant which controls the overlap between the hidden neurons, and μ_{nr}^c is the nearest neuron in the same class.

Learning Condition: The following conditions must be satisfied to make the network learn.

$$\begin{cases} \hat{c} = c \\ E \geq \varepsilon_l \end{cases} \quad (7)$$

Where ε_l is the threshold for adapting the network parameters. Hence, ε_l is restricted between [0.75 0.05]. Similar to ε_a, ε_l is also adapted based on the knowledge present in the current sample.

$$\varepsilon_l = \sigma\varepsilon_l - (1 - \sigma)E \quad (8)$$

However, our method uses multi-variable normal distribution as kernel function, so we should derive a new set of EKF formula.

The mathematic model of SOSM-RBF is as follows:

$$f(x) = \alpha_0 + \sum_{k=1}^{K} \alpha_k \Phi_k(x) \quad (9)$$

and $\Phi_k(x)$ is as bellows:

$$\Phi_k(x) = \exp\left(-\frac{1}{2}(x - \mu_k)'\Sigma^{-1}(x - \mu_k)\right) \quad (10)$$

Where μ_k is the k^{th} neuron center and Σ is the covariance matrix of the k^{th} neuron.

In order to accelerate the learning speed, we choose two neurons which are the nearest to the current example to learn.

The parameter of the network is

$$w = [\alpha_0, \alpha_1, \mu_1, \overrightarrow{\Sigma_1}, \alpha_2, \mu_2, \overrightarrow{\Sigma_2}]$$

Where $*_1$ are parameters of the first nearest neuron, $*_2$ are the second nearest neuron. $\overrightarrow{\Sigma_1}$ is the lowest triangular elements vector in column sequence, and so is $\overrightarrow{\Sigma_2}$.

The parameter **w** is adapted using EKF as follows:

$$w_i = w_{i-1} + K_i e_i \quad (11)$$

Where K_i is the Kalman gain matrix given by:

$$K_i = P_{i-1}B_i[R_i + B_i^T P_{i-1} B_i]^{-1} \quad (12)$$

B_i is the gradient matrix and has the following form:

$$B_i = [I, \sigma_1(x_i)I, \Sigma^{-1}(x_i - \mu_1)[\alpha_1 * \sigma_1(x_i)]^T, (x_i - \mu_1)'(x_i - \mu_1)[\alpha_1 * \sigma_1(x_i)]^T, \ldots, (x_i - \mu_2)'(x_i - \mu_2)[\alpha_2 * \sigma_2(x_i)]^T] \quad (13)$$

R_i is the variance of the measurement noise. P_i is the error covariance matrix which is updated by,

$$P_i = [I_{z*z} - K_i B_i^T]P_{i-1} + qI_{z*z} \quad (14)$$

q is a scalar that determines the allowed random step in the direction of the gradient matrix. If the number of parameters to be adjusted is z, P_i is a z*z positive definite symmetric matrix. When a new hidden neuron is allocated, the dimensionality of P_i increases to,

$$\begin{pmatrix} P_{i-1} & 0 \\ 0 & p_0 I_{z*z} \end{pmatrix} \quad (15)$$

Where the new rows and columns are initialized by P_0. P_0 .is an estimate of the uncertainty when the initial values assigned to the parameters. The dimension z of the identity matrix *I* is equal to the number of new parameters introduced by the new hidden neuron.

Skipping Condition: if the situation doesn't satisfied any condition, the current sample does not contain any significant information at this moment, so skip this sample, i.e., move this sample to unused data set, and use it in the next circle.

Then repeat these steps until the unused data set is empty. To summarize, the SOSM-RBF algorithm in a pseudo code form is given below.

2. SOSM-RBF Algorithm

Given a input feature *x* and its corresponding coded label *y*, and we assume that (**x**, **y**) is the current training sample:

1. **Compute** the network output:

$$\hat{y} = \alpha_0 + \sum_{k=1}^{K} \alpha_k \Phi_k(x) \quad (16)$$

2. **Calculate** the sample error **e**:

$$e_i = \begin{cases} y_i - \hat{y_i}, & \text{if } y_i \hat{y_i} < 1 \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

and predicted class label

$$\hat{c} = \arg\max_{i \in 1.2 \ldots, c} \hat{y_i} \quad (18)$$

3. **Skip Condition:** If E is less than 0.05, the current sample is discarded from the learning process.
4. **Growing Condition:** If the predicted class label is different from the original class label or maximum error is greater then self-adaptive threshold, add a new neuron($\hat{c} \neq c$ OR $E \geq \varepsilon_a$)

$$\omega_{new} = e; \ \mu_{new} = x;$$
$$\Sigma_{new} = \kappa \|x_i - \mu_{nr}^c\| * I_{nx*nx}$$

5. **Learning Condition:** If the maximum error is greater than threshold and predicted class label is same as the priginal class label, the network parameters using EKF.
6. Otherwise, the current observation does not contain significant information at this stage. Hence the samples is not used and kept in **reserve** for future fine tuning the network parameters.
7. Repeat the above steps until all the samples are utilized. Now, use the sample in reserve fr fine tuning the RBF classifier using the same learning process.

III. Performance

In this section, we present performance evaluation of SOSM-RBF classifier using three real-world classification problems from UCI Machine Learning Repository, namely: image segmentation, vehicle classification and glass identification problem. The detailed specification on number of inputs, number of classes and number of training/testing samples are given in Table 1

Table 1. Specification of UCI classification data set

Date set	Features	classes	Samples	
			Training	Testing
Image	19	7	210	2100
Vehicle	18	4	424	422
Glass	9	7	109	105

The performance of the SOSM-RBF is compared with MRAN, GAP-RBFN, OS-ELM and support vector machine. The control parameters used in the sequential learning algorithms MRAN, GAP-RBFN and SMC-RBF are optimized as highlighted in[8]. The average and overall classification accuracies as described in Table 2

Table 2. Parameters for all algorithms

Data Set	Methods	NO of neurons	Samples used	Testing	
				Average	Overall
Image Segmentation	SVM	96	210	90.62	90.62
	OS-ELM	100	210	90.67	90.67
	MRAN	76	210	86.52	86.52
	GAP-RBF	83	210	87.19	87.19
	SMC-RBF	43	210	91.00	91.00
	SOSM-RBF	46	198	**91.55**	**91.55**
Vehicle	SVM	96	424	68.72	97.99
	OS-ELM	100	424	68.95	67.56
	MRAN	76	424	59.54	59.83
	GAP-RBF	83	424	59.24	58.23
	SMC-RBF	43	424	74.18	73.52
	SOSM-RBF	123	400	**75.00**	**76.25**
Glass	SVM	102	109	64.23	60.01
	OS-ELM	60	109	67.62	70.12
	MRAN	51	109	63.81	70.24
	GAP-RBF	75	109	58.29	52.24
	SMC-RBF	58	109	78.09	77.96
	SOSM-RBF	83	103	**85.25**	**80.01**

From Table 2, we can say that comparing to existing

methods; our method has almost best efficiency. In addition, the performance of our method is not sensitive to the sequence of training samples. To Self-Adapting property, you don't have to input any initial parameter value using our method. Please also remember that, our method just uses part of training data for training, but gets the best results both on efficiency.

IV. Conclusion

In this paper, we present a new un-sequential and self-adapting Growing algorithm. In this method, we use multi-variable normal distribution function as kernel function, and deduct corresponding learning equation. In order to prevent the overlap problems, we introduce drop part to algorithm.

From the results, we can find that the proposed algorithm is not sensitive with the sequence of training samples. Unlike other methods, before using to a concrete problems, you should input many parameter initial values, you just have to configure one value for SOSM-RBF. At last, the efficiency performance of SOSM-RBF is better than other methods.

V. Acknowledgements

Dong Keming graduated from Harbin Institute of Technology, Department of Computer Science and Technology, China, 2004, and is currently pursuing a Ph.D. degree in Korea University. His research interests focus on Multimedia, and Information Management and Security and Machine Learning.

Hyoung Joong Kim received his B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1978, 1986, 1989, respectively. He joined the faculty of the Department of Control and Instrumentation Engineering, Kangwon National University, Korea, in 1989. He is currently a Professor of the Graduate School of Information Management and Security, Korea University, Korea since 2006. His research interests include parallel and distributed computing, multimedia computing, and multimedia security. He contributed to MPEG standardization for Digital Item Adaptation, File Format, Symbolic Music Representation, and Multimedia Application Format with more than 10 contributions and the same number of patents. In addition, he filed many patents and published more than 30 reviewed papers to international journals including IEEE and ACM, and 2 peer-reviewed book chapters. He served as Guest Editor of the IEEE Transactions on Circuits and Systems for Video Technology, EURASIP Journal of Advances in Signal Processing, and Technical Program Chair of many international conferences including International Workshop on Digital Watermarking (IWDW), and so on. He is a Vice Editor-in-Chief of the LNCS Transactions on Data Hiding and Multimedia Security, Associate Editors of well-known international journals, and Editors of many Lecture Notes in Computer Science series. He was the prime investigator of the national projects during 1997- 2005 developing interactive and personalized digital television. He is a member of ACM, IEEE and a couple of Korean academic societies.

References

[1] L. Yingwei, N. Sundararajan, and P.Saratchandran, "Performance evaluation of a sequential minimal radial basis function neural network learning algorithm," *IEEE Trans Neural Networks*, vol. 9, no. 2, pp. 185-189, 1982.

[2] G.-B.Huang, P.Saratchandran, and N.Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF network," *IEEE Trans*, vol. 34, no.6, pp. 22842292, 2004.

[3] G.-B.Huang, P.Saratchandran, and N.Sundararajan, "A generalized growing and pruning RBF neural network for function approximation," *IEEE Trans Neural Networks*,,Vol. 16,no. 1, pp.5767, 2005 .

[4] N. Y. Liang, G-B., Huang, P.saratchandran, and N.Sundararajan, "A fast and accurate online sequential learning algorithm for feedforword networks," *IEEE Trans Neural Networks*, vol. 147 No. 16, pp.1411-1423 2006.

[5] S. Suresh, R. V. Babu, and H.J. Kim, " No-reference image quality assessment using modified extreme learning machine classifier," *Applied Soft Computing*, Vol. **9**, no. 2 pp. 541-552 (2008).

[6] C.Blake, and C.Merz, UCI Repository of Machine Learning Database, Department of Information and Computer Science, University of California, Irvine, URL:(www.ics.uci.edu/mlearn/MLRopository.HTML) 1998.

[7] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines. Taiwan. Department of Computer Science and Information Engineering, National Taiwan University, 2003.

[8] S.Suresh, N.Sundararajan, and P. saratchandran, "A sequential multi-category classifier using radial basis function networks," *Neurocomputing*,,Vol. 71, pp1345-1358, 2008 .

[9]http://books.google.com/books?hl=en&lr=&id=7kp0AyXyq-0C&oi=fnd&pg=PR11&dq=MRAN&ots=yHoN4QSYHe&sig=vccnOhfwWt4Depe2iAktb4yBz7Y
[10]http://bioinformatics.oxfordjournals.org/cgi/content/abstract/16/10/906
[11]http://bioinformatics.oxfordjournals.org/cgi/content/abstract/16/10/906