ADAPTIVE RBF NET ALGORITHMS FOR NONLINEAR SIGNAL LEARNING WITH APPLICATIONS TO FINANCIAL PREDICTION AND INVESTMENT

Lei Xu

Computer Science and Engineering Department
The Chinese University of Hong Kong, Shatin, NT, Hong Kong, P.R.China
lxu@cse.cuhk.edu.hk

ABSTRACT

A smoothed variant of the EM algorithm is given for simultaneous training the first layer and the output layer globally in the *Normalized Radial Basis Function (NRBF)* nets and *Extended Normalized RBF nets (ENRBF)*, together with a BYY learning criterion for the selection of number of basis function. Moreover, a hard-cut fast implementation and an adaptive algorithm have also been proposed for speeding up the training and to handling time varying in the real time nonlinear signal learning and processing. A number of experiments are made on foreign exchange prediction and trading investment.

1. INTRODUCTION: NRBF AND ENRBF NETS

Normalized RBF (NRBF) nets have been used in nonlinear signal learning and processing with quite promising results [1, 4, 5]. In this paper, we propose a new learning technique for NRBF nets with adaptive algorithm.

The NRBF nets and its Extended NRBF (ENRBF) nets can be summarized in the general form [6, 4]:

$$f_k(x) = \frac{\sum_{j=1}^{k} r_j \phi([x-m_j]^T \Sigma_j^{-1}[x-m_j])}{\sum_{j=1}^{k} \phi([x-m_j]^T \Sigma_j^{-1}[x-m_j])},$$
$$r_j = \begin{cases} w_j, & \text{for an NRBF net,} \\ W_j^T x + c_j, & \text{for an ENRBF net;} \end{cases} \quad (1)$$

where $\phi(r^2)$ is a prespecified basis function satisfying certain weak conditions. The most common choice is the Gaussian $\phi(r^2) = e^{-r^2}$, several choices are listed in [7]. m_j is called the center vector and w_j is a weight vector. Σ_j is a $d \times d$ covariance matrix, and W_j is a parameter matrix.

For the existing approaches [6, 2, 4], the learning on the parameters in eq.(1) is separated into two steps:

(1) *Determining the parameters in the input layer*. The centers $m_j, j = 1, \cdots, k$ are determined only based on the input samples $\mathcal{D}_x = \{x_i\}_{i=1}^N$ via some clustering technique, for example, the K-means algorithm [6]. The cluster centers are usually used as $m_j, j = 1, \cdots, k$, with the k heuristically pre-fixed. While Σ_j is either externally prespecified at some value or computed from the resulted cluster centered at m_j.

(2) *Determining the parameters in the output layer*. After the parameters are settled, the parameter vector w_j or W_j, c_j can be determined by the least squares method based on the paired data set $\mathcal{D}_{x,z} = \{x_i, z_i\}_{i=1}^N$. The procedure can be implemented either in the batch way or by the adaptive least squares algorithm.

The above two step learning usually results in a suboptimal result. This is one problem that needs to be further solved. The second problem for learning in RBF nets is how to select the number of basis functions, which will also affect the performance considerably. In the paper [9], a number of theoretical results are given for the upper bounds of convergence rate of the approximation error with respect to the number of basis functions. Rival Penalized Competitive Learning (RPCL) is able to automatically select the number of clusters and thus suggested for RBF nets [8]. However, although it experimentally works well, RPCL is a heuristic approach and still in lack of theoretical justification.

In [12], the NRBF and ENRBF nets are shown to be a special case of the *Alternative Model for Mixture of Experts* [11] and thus the well known Expectation-Maximization (EM) algorithm for maximum likelihood learning is suggested to the two types of RBF nets for determining the parameters in the input and output layers globally. Moreover, the Alternative Model for Mixture of Experts (AMME) is shown to be a special case of the recent proposed *Bayesian Ying-Yang (BYY)* learning system and theory [13, 14, 10] such that an interesting model selection criterion has been obtained to determine the number of experts and basis functions. This paper further considers how to extend these previous results into adaptive learning to tackle practical problems in nonlinear signal learning and processing, especially for time-varying finaicial time series.

In Sec. 2, we introduce a smoothed variant of the batch way EM algorithm for NRBF and ENRBF nets, as well as the BYY learning criterion for the selection of number of basis function. In Sec.3, we propose to approximate the EM algorithm by hard-cut implementation and adaptive algorithms for considerably speeding up and for on-line processing. Furthermore, in Sec.4, a number of experiments are made on foreign exchange prediction and trading investment, which demonstrate that the proposed algorithms and criterion work well. We conclude in Sec. 5.

2. SMOOTHED EM ALGORITHM AND SELECTION OF BASIS FUNCTIONS

It can be shown [12, 14] that the above NRBF and ENRBF nets with gaussian $\phi(r^2) = e^{-r^2}$ can be obtained from the *Alternative Model for Mixture of Experts (AMME)* [11] at the special cases that each expert is

$$p(z|x,\theta_j) = \begin{cases} G(z, w_j, \Gamma_j), & \text{for NRBF,} \\ G(z, W_j^T x + c_j, \Gamma_j), & \text{for ENRBF;} \end{cases} \quad (2)$$

This work was supported by HK RGC Earmarked Grants CUHK484/95E and CUHK 339/96E and by Ho Sin-Hang Education Endowment Fund for Project HSH 95/02.

and the gating network is

$$\alpha_j = \frac{\sqrt{|\Sigma_j|}}{\sum_{j=1}^k \sqrt{|\Sigma_j|}},\ p(j|x,\nu) = \frac{e^{-0.5(x-m_j)^T\Sigma_j^{-1}(x-m_j)}}{\sum_{j=1}^k e^{-0.5(x-m_j)^T\Sigma_j^{-1}(x-m_j)}}. \tag{3}$$

where $w_j, W_j, c_j, m_j, \Sigma_j$ are exactly the same as in eq.(1). Moreover, by changing the gating net, we may also obtain their variants with different basis functions $\phi(r)$.

The *EM Algorithm* for training the AMME [11] can be directly used for learning in NRBF and ENRBF nets. Moreover, via showing that the AMME is a special case of the recent proposed *Bayesian Ying-Yang (BYY)* learning system and theory [13, 14, 10], we can get a smoothed variant of EM algorithm for the cases of finite number N of samples by using the kernel estimate to replace empirical estimate of densities.

The Smoothed EM Algorithm-RBF

E Step: Fix Θ^{old}, get

$$\begin{aligned}
&h(j|x_i) = \frac{\phi(||x_i - m_j^{old}||)G(z_i, r_j^{old}, \Gamma_j^{old})}{\sum_{j=1}^k \phi(||x_i - m_j^{old}||)G(z_i, r_j^{old}, \Gamma_j^{old})},\\
&\phi(||x_i - m_j^{old}||) = e^{-0.5(x_i-m_j^{old})^T\Sigma_j^{-1\,old}(x_i-m_j^{old})},\\
&r_j^{old} = \begin{cases} w_j^{old}, & \text{for an NRBF net,}\\ (W_j^{old})^T x_i + c_j^{old}, & \text{for an ENRBF net.}\end{cases}.
\end{aligned} \tag{4}$$

M Step: First, get $N_j^{eff} = \sum_{i=1}^N h(j|x_i)$ and update

$$\begin{aligned}
&m_j^{new} = (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)x_i,\\
&\Sigma_j^{new} = h_x(j) + (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)\Sigma_{i,j},\\
&\Sigma_{i,j} = [x_i - m_j^{new}][x_i - m_j^{new}]^T,\\
&h_x(j) = (0.25 d_x C_x / N_j^{eff})^{\frac{1}{4+d_x}},
\end{aligned} \tag{5}$$

and then (a) for an NRBF net, update

$$\begin{aligned}
&w_j^{new} = (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)z_i,\\
&\Gamma_j^{new} = h_z(j) + (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)\Gamma_{i,j},\\
&\Gamma_{i,j} = (z_i - w_j^{new})(z_i - w_j^{new})^T,\\
&h_z(j) = (0.25 d_z C_z / N_j^{eff})^{\frac{1}{4+d_z}},
\end{aligned} \tag{6}$$

and (b) for an ENRBF net, update

$$\begin{aligned}
&Ez_j = (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)z_i,\\
&R_{xz} = (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)[x_i - m_j^{new}][z_i - Ez_j]^T,\\
&W_j^{new} = [\Sigma_j^{new}]^{-1}R_{xz},\ c_j^{new} = Ez_j - (W_j^{new})^T m_j^{new},\\
&\Gamma_{i,j} = [z_i - W_j^{new\,T}x_i - c_j^{new}][z_i - W_j^{new\,T}x_i - c_j^{new}]^T,\\
&\Gamma_j^{new} = h_z(j) + (1/N_j^{eff})\sum_{i=1}^N h(j|x_i)\Gamma_{i,j}.
\end{aligned} \tag{7}$$

Particularly, when $\Gamma_j = \gamma_j^2 I$, which is widely assumed in the literature, the updating on Γ_j can be simply

$$\gamma_j^2 = h_z(j) + N^{-1}\sum_{j=1}^k\sum_{i=1}^N h(j|x_i)||z_i - r_j^{new}||^2, \tag{8}$$

In the above equations, $h_x(j), h_z(j)$ are the smooth parameters used in the kernel density estimate, roughly according to Theorem 20 in [9]. C_x, C_z are two heuristic constants. In the special cases of $h_x(j) = 0, h_z(j) = 0$, the above EM algorithm returns exactly to the EM algorithm given in [12, 14].

Furthermore, from the fact that the AMME is a special case of the BYY learning [13, 14, 10], we can get a model selection criterion for determining the number of experts. This criterion can be also used for determining the number of basis functions by $k^* = arg\min_k J(k)$ with $J(k)$ being either $J_1(k)$ or $J_2(k)$:

$$\begin{aligned}
&J_1(k) = \frac{1}{N}\sum_{i=1}^N\sum_{j=1}^k h^*(j|x_i)\ln h^*(j|x_i) +\\
&\sum_{j=1}^k \frac{\sqrt{|\Sigma_j^*|}}{\sum_{j=1}^k\sqrt{|\Sigma_j^*|}}\ln\sqrt{|\Gamma_j^*|} + \ln\sum_{j=1}^k\sqrt{|\Sigma_j^*|},\\
&h(j|x_i) = \frac{\phi(||x_i - m_j^*||)G(z_i, r_j^*, \Gamma_j^*)}{\sum_{j=1}^k \phi(||x_i - m_j^*||)G(z_i, r_j^*, \Gamma_j^*)},\\
&\phi(||x_i - m_j^*||) = e^{-0.5(x_i-m_j^*)^T(\Sigma_j^*)^{-1}(x_i-m_j^*)},\\
&r_j^* = \begin{cases} c_j^*, & \text{for an ENRBF net,}\\ W_j^{*T}x_i + c_j^*, & \text{for an ENRBF net;}\end{cases}\\
&J_2(k) = \sum_{j=1}^k \frac{\sqrt{|\Sigma_j^*|}}{\sum_{j=1}^k\sqrt{|\Sigma_j^*|}}\ln\sqrt{|\Gamma_j^*|} + \ln\sum_{j=1}^k\sqrt{|\Sigma_j^*|}.
\end{aligned} \tag{9}$$

where "$*$" denotes the estimated values for the parameters after a learning algorithm, e.g., the above smoothed EM algorithm eq.(7).

Taking $J_2(k)$ as an example, we can intuitively understand how the above criteria work. Its first term will decrease as k increases and its second term will increase as k increases and thus trades off the best k^*. This point can be even more clearly observed by letting $\Gamma_j^* = \Gamma^*, \Sigma_j^* = \Sigma^*$ and $\alpha_j^* = 1/k$ such that

$$J_2(k) = 0.5(\ln|\Gamma^*| + \ln|\Sigma^*|) + \ln k. \tag{10}$$

Obviously, $\ln k$ increases as k increases, and $|\Gamma^*|, |\Sigma^*|$ decreases as k increases for a given number N of samples.

From eq.(10), we can also see that in different from their unsmoothed versions , the effect of the smoothed estimates $|\Gamma^*|, |\Sigma^*|$ by the above smoothed EM algorithm is to reduce the decreasing speeds of $|\Gamma^*|, |\Sigma^*|$ as k increases in the cases of finite number samples so that the over-fitting can be penalized.

3. HARD-CUT AND ADAPTIVE ALGORITHMS

Actually, $h(j|x_i)$ is a posterior probability of assigning the mapping task of the pair $x_i \to z_i$ to the j-th basis function. Alternatively, this soft assignment can be approximated by a Winner-Take-All (WTA) competition according to Bayesian decision via replacing $h(j|x_i)$ in the M-step of *EM algorithm* with the following hard-cut indicator:

$$I(j|x_i) = \begin{cases} 1, & \text{if } j = arg\min_r\{-\log h(r|x_i)\},\\ 0, & \text{otherwsie.}\end{cases} \tag{11}$$

Although some accuracy may be lost in performance (as will be shown later by experiments, this loss can usually be ignored), it

may bring two advantages. **First**, it can speed up considerably, not only because the multiplication with $h(j|x_i)$ is not needed and the summation of the terms with $I(j|x_i) = 0$ can be saved, but also because the computing cost for $h(j|x_i)$ can be considerably reduced since the computing for $I(j|x_i)$ is significantly simplified by omitting those irrelevant computations. **Second**, it can speed up the convergence. After running for a certain period, the EM learning usually slows down considerably, sometimes it will become very slow. In this case, we can switch $h(j|x_i)$ into $I(j|x_i)$, which can obviously speed up the convergence, without affecting the performance too much.

When applied to nonlinear signal learning and processing, such as prediction of foreign exchange rate, we need that the learning is made adaptively on line. In the following, we turn the batch way EM algorithm an adaptive for such a purpose:

The Adaptive EM Algorithm-RBF

E Step: Fix Θ^{old}, to get $h(j|x_i)$ in the same way as in the batch *EM Algorithm-RBF* in Sec.2 or to get $I(j|x_i)$ by eq.(11). Then let $\eta_{j,i}$ be given by

$$\text{Either} \eta_{j,i} = \eta_0 h(j|x_i)/\alpha_j, \quad \text{or} \eta_{j,i} = \eta_0 I(j|x_i)/\alpha_j, \tag{12}$$

with η_0 being a prefixed learning rate and $h(j|x_i)$ or $I(j|x_i)$ modifying this rate adaptively.

M Step: First, update

$$\begin{aligned} m_j^{new} &= m_j^{old} + \eta_{j,i}(x_i - m_j^{old}), \\ \Sigma_j^{new} &= (1 - \eta_{j,i})\Sigma_j^{old} + \eta_{j,i}(h_x(j) + \Sigma_{i,j}), \\ \Sigma_{i,j} &= (x_i - m_j^{old})(x_i - m_j^{old})^T. \end{aligned} \tag{13}$$

Then, (a) for a NRBF net, update

$$\begin{aligned} c_j^{new} &= c_j^{old} + \eta_{j,i}(z_i - c_j^{old}), \\ \Gamma_j^{new} &= (1 - \eta_{j,i})\Gamma_j^{old} + \eta_{j,i}(h_z(j) + \Gamma_{i,j}), \\ \Gamma_{i,j}) &= (z_i - c_j^{new})(z_i - c_j^{new})^T; \end{aligned} \tag{14}$$

(b) for an ENRBF net, update

$$\begin{aligned} Ez_j^{new} &= Ez_j^{old} + \eta_{j,i}(z_i - Ez_j^{old}), \\ c_j^{new} &= Ez_j^{new} - w_j^{old\,T} m_j^{old}; \\ \Gamma_j^{new} &= (1 - \eta_{j,i})\Gamma_j^{old} + \eta_{j,i}(h_z(j) + \Gamma_{i,j}), \\ \Gamma_{i,j} &= (z_i - w_j^{old\,T} x_i - c_j^{new})(z_i - w_j^{old\,T} x_i - c_j^{new})^T, \\ w_j^{new} &= w_j^{old} + \eta_{j,i}(z_i - w_j^{new\,T} x_i - c_j^{new}) x_i^T. \end{aligned} \tag{15}$$

The initialization of parameters will affect the performance of adaptive algorithms. For the practical problems like financial data prediction, we usually have quite limited number of sample points, thus using an adaptive algorithm alone can not bring any real advantage. *In this case, we suggest to first use the batch algorithm on a training set to get a solution as an initialization, and then to use the above adaptive algorithm to keep tracing the changes of data on the testing data via adaptation once a new data point is available.*

4. EXPERIMENTS

Experiments are made on comparing the following algorithms:

(a) The conventional two-stage training algorithms [6][4] for NRBF and ENRBF nets, denoted by NRBF 2-stage and ENRBF 2-stage, respectively;

(b) The batch *EM Algorithm-RBF* , denoted by EM-NRBF and EM-ENRBF for NRBF net and Extended NRBF net respectively;

(c) The batch *EM Algorithm-RBF* with hardcut technique, denoted by EM-NRBF (HC) and EM-ENRBF (HC) respectively;

(d) The *Adaptive EM Algorithm-RBF*, denoted by Adaptive EM-NRBF and Adaptive EM-ENRBF respectively;

A FOREX rate data for USD-DEM with 1112 samples (Nov. 25,1991- Aug 30,1995) and a real stock price data of 380 samples from Shanghai stock market are used. On the USD-DEM Forex data, two type of partitions of the training and testing sets are used. For the Type A, the training size is the first 1000 samples and the testing size is the subsequent 112 samples. For Type B, the training size is first 100 samples and the testing size is the subsequent 1012 samples. For the real stock price data, the first 350 samples used as the training set and the subsequent 30 samples as the testing set.

In all the experiments, the initialization of the parameters and its variants are made randomly in their corresponding domains, e.g., Σ_j is initialized at a positive definite matrix. According to our experience, we use $x = [x(t-1), x(t-2), x(t-3)]$ as our input vector at the time t.

Shown in Tab.1 are the results of using 20 basis functions. We observe that EM-NRBF indeed improves the two stage algorithm considerably. We can also see that the computing cost of EM-NRBF (HC) is almost comparable to the conventional algorithm, which is a significant speeding up from EM-NRBF before hardcut that is about one or two magnitudes slower. Moreover, further increasing the number of basis functions will not obviously improve the results by the two stage algorithm, but its computing cost will increase fastly and become worse than EM-NRBF (HC).

In Fig.1, the comparison are made on Forex data of USD-DEM-SET Type B by EM-NRBF (HC) and Adaptive EM-NRBF. The adaptive algorithm is used to track time series in such a way that the sample point at t is used to modified the network once this point is known already (i.e., once the current time t is passed into $t+1$). As shown in Fig.1, the adaptive algorithm indeed can track the temporal change very well and outperform its corresponding batch way algorithm significantly.

Shown in Fig.2, are the comparison results on the real stock data by EM-NRBF (HC) and Adaptive EM-NRBF. Again, the adaptive algorithm can outperform its corresponding batch way algorithm significantly.

We use the simple trading rule proposed in [3] for trading investment based on the obtained prediction. Forex data of USD-DEM Type A is used again such that we can compare the result made in our previously results by the different approaches [3]. We assume that a trader can hold at most a long or short contract of foreign dollars or stock at one time. The deposit amount is fixed to be US$ 6500 and the transaction cost rate is 0.5% of the deposit.

The results are shown in Tab.2. The adaptive algorithms can bring significant profit. Especially, Adaptive EM-NRBF on NRBF net improves its non-adaptive version with the profits being as large as nearly 3 times. Also, Adaptive EM-NRBF on NRBF net gives the best result which is a considerable improvement over the one made on ENRBF net. Moreover, the batch way algorithm on ENRBF net got an obvious better result over that on NRBF net. The result given in Tab.2 also provides a considerable improvement over the result made in [3], which was compared with Random walk and AMAR model with significant improvements already.

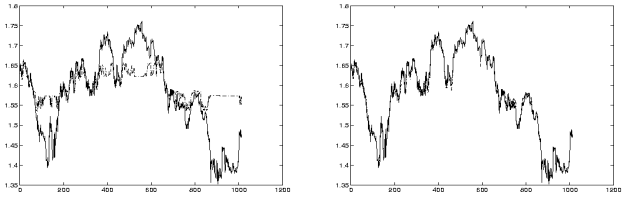
5. CONCLUSIONS

The EM algorithm improves the conventional two stage algorithm considerably for learning in NRBF and ENRBF nets. The hardcut technique can significantly speed up convergence while still keep a very good performance. By the adaptive algorithm, we can get significant improvements on financial predication and trading investment. For the two stage algorithm, ENRBF net is much better than NRBF net. However, it is not the case for the EM algorithm as well as its hardcut and adaptive variants.

Algorithms	Flops*	Training	$Testing$
NRBF 2-stage	4.81×10^5	0.396	1.703
EM-NRBF (HC) II	5.94×10^5	0.238	0.768
ENRBF 2-stage	3.91×10^5	0.173	0.452
EM-ENRBF (HC) II	3.96×10^6	0.151	0.445

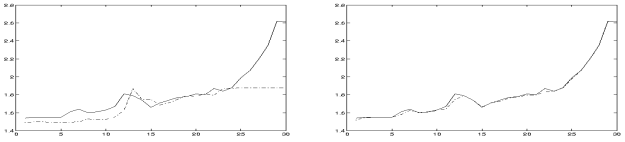
* They are training Flops with one flop counted by MATLAB as an addition or multiplication operation.

Tab. 1 The results of prediction on FOREX rate of USD-DEM-SET Type A (No. of units = 20), with NMSE error used.



Normalized Testing Error = 0.0840 (a) by Batch way EM-NRBF. Normalized Testing Error = 0.0083 (b) by Adaptive EM-NRBF

Fig. 1 The Results of prediction on Forex data of USD-DEM-SET Type B (No. of units = 20). In (b), the prediction and the real data are almost overlapped.



Normalized Testing Error = 0.2320 (a) by Batch way EM-NRBF. Normalized Testing Error = 0.0169 (b) by Adaptive EM-NRBF

Fig. 3 The Results of prediction on the stock price data (No. of units = 20). In (b), the prediction and the real data are almost overlapped.

Algorithms	Net profit point	Profit in US$
EM-NRBF	1425	9262.5
Adaptive EM-NRBF	3966	25779.0
EM-ENRBF	2063	13406.5
Adaptive EM-ENRBF	2916	18954.0

Tab. 2 The results of trading investment based on the prediction on USD-DEM-SET Type A (in 112 days)

6. REFERENCES

[1] S.M.Botros and C.G.Atkeson, Generalization properties of radial basis function, *Advances in Neural Information Processing Systems 3*, eds. R.P.Lippmann et al, (Morgan Kaufmann Pub., 1991), 707-713.

[2] S.Chen, C.F.N.Cowan and P.M.Grant, Orthogonal least squares learning algorithm for Radial basis function networks, *IEEE Trans. Neural Networks 2*(1991), 302-309.

[3] Y.M. Cheung, Helen Z.H. Lai and L. Xu, Adaptive rival penalized competitive learning and combined linear predictor with application to financial investment, *Proc. of 1996 IEEE/IAFE Conf Computational Intelligence for Financial Engineering* , New York City, (1996), 141-147

[4] R.D.Jones et al, Information theoretic derivation of network architecture and learning algorithms, *Proc. of IJCNN91-Seattle*, Vol.II(1991), 473-478.

[5] V. Kardirkamanathan, M. Niranjan and F.Fallside, Sequential adaptation of Radial basis function neural networks and its application to time-series prediction, *Advances in Neural Information Processing System 3*, eds., R.P.Lippmann, et al , (San Mateo: Morgan Kaufmann Pub, 1991), 721-727.

[6] J.Moody & J.Darken, Fast learning in networks of locally-tuned processing units, *Neural Computation 1*(1989), 281-294.

[7] T.Poggio and F.Girosi, Networks for approximation and learning, *Proc. of IEEE 78*(1990), 1481-1497.

[8] L. Xu, A.Krzyzak, and E.Oja, Rival Penalized Competitive Learning for Clustering Analysis, RBF net and Curve Detection, *IEEE Trans. on Neural Networks 4*(4)(1993), 636-649. Its preliminary version was first published in Proc. 1992 IJCNN, Beijing, P.R.China, Nov.3-6, Vol.2 (1992), 665-670.

[9] L. Xu, A.Krzyzak, & A.L.Yuille, On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Field Size, *Neural Networks*, *7*(4) (1994), 609-628.

[10] L. Xu, YING-YANG Machine: a Bayesian-Kullback scheme for unified learnings and new results on vector quantization, Keynote talk, *Proc. Intl Conf. on Neural Information Processing (ICONIP95)*, Oct 30 - Nov. 3, (1995), 977-988.

[11] L. Xu, M.I.Jordan, & G.E.Hinton, An Alternative Model for Mixtures of Experts, *Advances in Neural Information Processing Systems 7*, eds., Cowan, J.D., Tesauro, G., and Alspector, J., (MIT Press, 1996), 633-640. A preliminary version was first published in *Proc. of WCNN'94*, San Diego, Vol. 2(1994), 405-410.

[12] L. Xu, Bayesian-Kullback YING-YANG Learning Scheme: Reviews and New Results, *Proc. Intl Conf. on Neural Information Processing (ICONIP96)*, Sept. 24-27, (Springer-Verlag, 1996), 59-67.

[13] Xu, L., "Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (I) Unsupervised and Semi-Unsupervised Learning", Invited paper, ", S. Amari and N. Kassabov eds., *Brain-like Computing and Intelligent Information Systems*, 1997, Springer-Verlag, pp241-274.

[14] Xu, L., " Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach (II): From Unsupervised Learning to Supervised Learning and Temporal Modeling & (III) Models and Algorithms for Dependence Reduction, Data Dimension Reduction, ICA and Supervised Learning ", Invited paper, **Lecture Notes in Computer Science***: Proc. of Intl Workshop on Theoretical Aspects of Neural Computation*, May 26-28, 1997, Hong Kong, Springer- Verlag, pp25-42 and pp43-60.