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]:

\[
f_k(x) = \sum_{j=1}^{k} w_{j} \phi([x - m_j]^T \Sigma_j^{-1} [x - m_j]),
\]

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. \( \Sigma_j \) 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 \( \mathcal{D}_u = \{x_i\}_{i=1}^N \) 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 \( \Sigma_j \) 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 \( \mathcal{D}_{u,z} = \{x_i, z_i\}_{i=1}^N \). 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 financial 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

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

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
\[
\alpha_j = \sqrt{\frac{E_x}{N}} p(j|x_i, \nu) = \frac{e^{-0.5(z_i-m_j)^T\Sigma_j^{-1}(z_i-m_j)}}{\sum_{j=1}^k e^{-0.5(z_i-m_j)^T\Sigma_j^{-1}(z_i-m_j)}},
\]
where \(w_j, W_j, \sigma_j, \Sigma_j, \) 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 NRB and ENRBF nets. Moreover, via showing that the AMME is a special case of the recently 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 \(\Theta^{\text{old}},\) get
\[
h_j(x_i) = \frac{\phi(\|x_i-m_j^{\text{old}}\|)G(z_i, x_i^{\text{old}}, \Gamma^{\text{old}})}{\sum_{j=1}^k \phi(\|x_i-m_j^{\text{old}}\|)G(z_i, x_i^{\text{old}}, \Gamma^{\text{old}})},
\]
\[
\phi(\|x_i-m_j^{\text{old}}\|) = e^{-0.5(z_i-m_j^{\text{old}})^T\Sigma_j^{-1}(z_i-m_j^{\text{old}})},
\]
\[
r_j^{\text{old}} = \left(\frac{W_j^{\text{old}}}{\sqrt{\Gamma_j^{\text{old}}}}\right)x_i + e_j^{\text{old}}, \quad \text{for an NRB net},
\]
\[
r_j^{\text{old}} = \left(\frac{W_j^{\text{old}}}{\sqrt{\Gamma_j^{\text{old}}}}\right)x_i + e_j^{\text{old}}, \quad \text{for an ENRBF net}.
\]

M Step: First, get \(N_j^{\text{eff}} = \sum_{i=1}^N h_j(x_i)x_i\) and update
\[
m_j^{\text{new}} = \frac{1}{N_j^{\text{eff}}} \sum_{i=1}^N h_j(x_i)x_i,
\]
\[
\Sigma_j^{\text{new}} = h_j(x_i) + (1/N_j^{\text{eff}}) \sum_{i=1}^N h_j(x_i)\Sigma_j^{\text{old}},
\]
\[
\Sigma^{\text{old}} = [x_i-m_j^{\text{new}}][x_i-m_j^{\text{new}}]^T,
\]
\[
h_j(x_i) = (0.25d_x C_x/N_j^{\text{eff}}) \frac{1}{1+m_j^{\text{new}}}, \quad \text{for an NRB net},
\]
\[
\text{and then (a) for an NRB net, update}
\]
\[
w_j^{\text{new}} = \frac{1}{N_j^{\text{eff}}} \sum_{i=1}^N h_j(x_i)x_i,
\]
\[
\Gamma_j^{\text{new}} = h_j(x_i) + (1/N_j^{\text{eff}}) \sum_{i=1}^N h_j(x_i)\Gamma_{i,j},
\]
\[
\Gamma_{i,j} = [x_i-w_j^{\text{new}}][x_i-w_j^{\text{new}}]^T,
\]
\[
h_i(j) = (0.25d_x C_x/N_j^{\text{eff}}) \frac{1}{1+m_i^{\text{new}}}, \quad \text{for an ENRBF net},
\]
\[
\text{and (b) for an ENRBF net, update}
\]
\[
E_{x_i} = \frac{1}{N_j^{\text{eff}}} \sum_{i=1}^N h_j(x_i)x_i,
\]
\[
R_{x_i} = \frac{1}{N_j^{\text{eff}}} \sum_{i=1}^N h_j(x_i)[x_i-m_j^{\text{new}}][x_i-E_{x_i}]^T,
\]
\[
W_j^{\text{new}} = \left[\Sigma_j^{\text{new}}\right]^{-1}R_{x_i},
\]
\[
\Gamma_{i,j} = [x_i-w_j^{\text{new}}][x_i-w_j^{\text{new}}]^T,
\]
\[
h_i(j) = \frac{1}{N_j^{\text{eff}}} \sum_{i=1}^N h_i(j)\Gamma_{i,j}. \quad \text{for an ENRBF net},
\]

Particularly, when \(\Gamma_j = \gamma_j^2 I,\) which is widely assumed in the literature, the updating on \(\Gamma_j\) can be simply
\[
\gamma_j^2 = h_i(j) + N^{-1} \sum_{i=1}^N \sum_{j=1}^N h_j(x_i)[z_i - r_j^{\text{new}}]_i^2,
\]
\[
\text{In the above equations, } h_i(j), h_j(x_i) \text{ are the smooth parameters used in the kernel density estimate, roughly according to Theorem 20 in [9]. } C_x, C_{\gamma} \text{ are two heuristic constants. In the special cases of } h_i(j) = 0, h_j(x_i) = 0, \text{ the above EM algorithm returns exactly to the EM algorithm given in [12, 14].}
\]

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 also be used for determining the number of basis functions by \(k^* = \arg \min_k J(k) \text{ with } J(k) \text{ being either } J_1(k) \text{ or } J_2(k):\)
\[
J_1(k) = \frac{1}{N} \sum_{i=1}^N \ln \left(\frac{1}{N} \sum_{i=1}^N h_j(x_i) \ln h_j(x_i)\right) +
\]
\[
\frac{1}{k} \sum_{j=1}^k \ln \sqrt{\sum_{i=1}^N \left(\frac{1}{N} \sum_{i=1}^N \ln \left(\frac{1}{N} \sum_{i=1}^N h_j(x_i) \ln h_j(x_i)\right)\right)} +
\]
\[
\frac{1}{2} \sum_{j=1}^k \ln \sqrt{\sum_{i=1}^N \left(\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^k h_j(x_i) \ln h_j(x_i)\right)} +
\]
\[
\frac{1}{k} \ln \sum_{j=1}^k N_j^{\text{eff}} + \ln \sum_{j=1}^k N_j^{\text{eff}}, \quad \text{for an ENRBF net,}
\]
\[
J_2(k) = \frac{1}{k} \sum_{j=1}^k \ln \sqrt{\sum_{i=1}^N \left(\frac{1}{N} \sum_{i=1}^N \ln \left(\frac{1}{N} \sum_{i=1}^N h_j(x_i) \ln h_j(x_i)\right)\right)} +
\]
\[
\frac{1}{2} \sum_{j=1}^k \ln \sum_{i=1}^N \left(\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^k h_j(x_i) \ln h_j(x_i)\right) +
\]
\[
\frac{1}{2} \ln \sum_{j=1}^k \left(\frac{1}{N} \sum_{i=1}^N \left(\frac{1}{N} \sum_{i=1}^N h_j(x_i) \ln h_j(x_i)\right)\right) + \ln k.
\]

Obviously, \(\ln k \text{ increases as } k \text{ increases, and } \left[J_1^*\right], \left[J_2^*\right] \text{ decreases as } k \text{ increases for a given number } N \text{ of samples.}

From eq.(10), we can also see that in different from their unsmoothed versions, the effect of the smoothed estimates \(\left[J_1^*\right], \left[J_2^*\right]\) by the above smoothed EM algorithm is to reduce the decreasing speeds of \(\left[J_1^*\right], \left[J_2^*\right]\) 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 \(x_i \rightarrow z_i\) 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:
\[
I(j|x_i) = \begin{cases} 1, & \text{if } j = \arg \min_l \{- \log h_l(x_i)\}, \\ 0, & \text{otherwise}. \end{cases}
\]

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 \( \Sigma j \) is \( 0 \) 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 be-up the convergence. After running for a certain period, the EM by omitting those irrelevant computations.

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 adaptively for such a purpose:

The Adaptive EM Algorithm-RBF

E Step: Fix \( \Theta^{old} \), 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

\[
\text{Either} \eta_{j,i} = \gamma_{0} h(j|x_i)/\alpha, \quad \text{or} \eta_{j,i} = \gamma_{0} I(j|x_i)/\alpha, \tag{12}
\]

with \( \gamma_{0} \) being a prefixed learning rate and \( h(j|x_i) \) or \( I(j|x_i) \) modifying this rate adaptively.

M Step: First, update

\[
\begin{align*}
\hat{m}_{j}^{new} &= m_{j}^{old} + c_{j}^{new} (x_{i} - m_{j}^{old}), \\
\hat{\Sigma}_{j}^{new} &= (1 - \eta_{j,i} \Sigma_{j}^{old} + \eta_{j,i} (h_{i,j} + \Sigma_{i,j})), \\
\Sigma_{i,j} &= (x_{i} - m_{j}^{new}) (x_{i} - m_{j}^{new})^{T}. 
\end{align*}
\tag{13}
\]

Then, (a) for a NRBF net, update

\[
\begin{align*}
\hat{w}_{j}^{new} &= \hat{w}_{j}^{old} + \eta_{j,i} (c_{j}^{new}), \\
\hat{\Gamma}_{j}^{new} &= (1 - \eta_{j,i} \Gamma_{j}^{old} + \eta_{j,i} (h_{i,j} + \Gamma_{i,j})), \\
\Gamma_{i,j} &= (c_{i} - c_{j}^{new}) (c_{i} - c_{j}^{new})^{T}; 
\end{align*}
\tag{14}
\]

(b) for an ENRBF net, update

\[
\begin{align*}
E_{z_{j}}^{new} &= E_{z_{j}}^{old} + \eta_{j,i} (z_{i} - E_{z_{j}}^{old}), \\
\gamma_{j}^{new} &= E_{z_{j}}^{old} - w_{z_{j}}^{old} T m_{j}^{old}; \\
\Gamma_{j}^{new} &= (1 - \eta_{j,i}) \gamma_{j}^{old} + \eta_{j,i} (h_{i,j} + \Gamma_{i,j}), \\
\Gamma_{i,j} &= (z_{i} - w_{z_{j}}^{old} T x_{i} - \gamma_{j}^{new} (z_{i} - w_{z_{j}}^{old} T x_{i} - \gamma_{j}^{old})^{T}, \\
w_{j}^{new} &= w_{j}^{old} + \eta_{j,i} (z_{i} - w_{z_{j}}^{old} T x_{i} - \gamma_{j}^{new})^{T}. \tag{15}
\] (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., \( \Sigma_{j} \) 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.

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;
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.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Flops*</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRBF 2-stage</td>
<td>4.81 × 10^4</td>
<td>0.3965</td>
<td>1.7035</td>
</tr>
<tr>
<td>EM-NRBF (HC) II</td>
<td>5.94 × 10^4</td>
<td>0.2387</td>
<td>0.7680</td>
</tr>
<tr>
<td>ENRBF 2-stage</td>
<td>3.91 × 10^4</td>
<td>0.1737</td>
<td>0.4520</td>
</tr>
<tr>
<td>EM-ENRBF (HC) II</td>
<td>3.96 × 10^4</td>
<td>0.1517</td>
<td>0.4450</td>
</tr>
</tbody>
</table>

* 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](image1)

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](image2)

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.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Net profit point</th>
<th>Profit in USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM-NRBF</td>
<td>14.25</td>
<td>9262.5</td>
</tr>
<tr>
<td>Adaptive EM-NRBF</td>
<td>39.66</td>
<td>257.75</td>
</tr>
<tr>
<td>EM-ENRBF</td>
<td>206.36</td>
<td>1340.65</td>
</tr>
<tr>
<td>Adaptive EM-ENRBF</td>
<td>291.16</td>
<td>1898.44</td>
</tr>
</tbody>
</table>

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

6. REFERENCES


