A Modified Model for Mixtures of Experts: New Gating Net, EM Algorithm and Piecewise Function Approximations

Lei Xu[1], Michael I. Jordan[1] and Geoffrey E. Hinton[2]
1. Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
and
2. Department of Computer Science
University of Toronto
10 King's College Road
Toronto, M5S 1A4, Canada

Abstract

A mixtures-of-experts model is modified with a new gating net in place of the old one. The modified model is trained by an *EM algorithm* and thus can automatically guarantee the convergence without any external help. The model with the old gating net does not share this property and needs to heuristically select a suitable learning stepsize to guarantee the convergence in its learning by either gradient or IRLS algorithm. Experiments have also shown that the use of EM algorithm can considerably speed up the whole learning process. Furthermore, the modified model with its EM learning is also proposed to tackle the tasks of piecewise nonlinear approximations by using polynomial, trigeometry, or other prespecified basis functions. Finally, the differences of our model to a related model have been elaborated.

1 Mixtures-of-Experts and Its EM Learning

Mixtures of Experts is a modular architecture (Jacobs, Jordan, Nowlan & Hinton, 1991) which outperforms a single multilayer net in tackling a complex task that consists of several simpler problems. It implements the following mixture conditional probabilistic model:

$$\begin{aligned} P(y|x,\Theta) &= \sum_{j=1}^{K} g_j(x,\nu) P(y|x,\theta_j), \\ P(y|x,\theta_j) &= (2\pi det\Gamma_j)^{-\frac{1}{2}} exp\{-\frac{1}{2}[y-f_j(x,w_j)]^T \Gamma_j^{-1}[y-f_j(x,w_j)]\}. \end{aligned} \tag{1}$$

with Θ consisting of $\nu, \{\theta_j\}_1^K$, and θ_j consisting of $\{w_j\}_1^K, \{\Gamma_j\}_1^K$. , $f_j(x,w_j)$ is the output of the j-th expert net. $g_j(x,\nu), j=1,\cdots,K$ are given by the so call *softmax* function

$$g_j(x,\nu) = e^{\beta_j(x,\nu)} / \sum_{i=1}^{K} e^{\beta_i(x,\nu)}, \tag{2}$$

of $\beta_j(x,\nu), j=1,\cdots,K$ — the outputs of a single feedfarward net called *gating* net. The gating net acts as a stochastic gate (switch) which selects the output of the j-th expert net with probability $P(j|x) = g_j(x,\nu)$.

The output of *Mixtures of Experts* can be one of the following three modes:

$$\begin{aligned} o(x) &= E(y|x,\Theta) = \sum_{j=1}^{K} g_j(x,\nu) E(y|x,\theta_j), \quad \textit{regression function}, \\ o(x) &= E(y|x,\theta_{j*}), \ \textit{with } P(j*=j) = g_j(x,\nu), \quad \textit{stochastic switching}, \\ o(x) &= E(y|x,\theta_{j*}), \ \textit{with } j* = arg\ Max_j\ g_j(x,\nu). \end{aligned} \tag{3}$$

The *regression function* mode is commonly used. Recently, Ghahramani & Jordan (1993) have shown that the last two modes are better for implementing one-to-many mappings.

The entire parameter set Θ is estimated by the rule of Maximum Likelihood(ML). That is, given a training set $\{y^{(t)}, x^{(t)}\}_{t=1}^N$, we find a Θ^* which maximizes the following likelihood function:

$$L = \sum_{t=1}^{N} \ln P(y^{(t)}|x^{(t)},\Theta). \tag{4}$$

Jacobs, Jordan, Nowlan & Hinton (1991) proposed a gradient ascent algorithm to maximize L. The Expectation-Maximization (EM) algorithm (Dempster, Lair and Rubin, 1977) is a typical iterative technique for ML estimation. Recently Jordan & Jacobs (1993) have used EM to the problem eq.(4) with a considerably improved convergence speed. The readers are referred to Jordan & Xu(1993) and Xu & Jordan (1993) for a detailed introduction on EM and some new theoretical results.

Given the current estimate $\Theta^{(k)}$, the EM procedure for the problem eq.(4) consists of two steps.

(1) E-step. First, for each pair $\{x^{(t)}, y^{(t)}\}$, we compute

$$h_j^{(k)}(y^{(t)}|x^{(t)}) = P(j|x^{(t)}, y^{(t)}) = \frac{g_j(x^{(t)}, \nu^{(k)})P(y^{(t)}|x^{(t)}, \theta_j^{(k)})}{\sum_{i=1}^{K} g_i(x^{(t)}, \nu^{(k)})P(y^{(t)}|x^{(t)}, \theta_j^{(k)})}; \tag{5}$$

Then, we form a set of new objective functions

$$\begin{aligned} Q_j^e(\theta_j) &= \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}) \ln P(y^{(t)}|x^{(t)}, \theta_j), \ j = 1, \cdots, K; \\ Q^g(\nu) &= \sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(y^{(t)}|x^{(t)}) \ln g_j^{(k)}(x^{(t)}, \nu^{(k)}). \end{aligned} \tag{6}$$

(2). M-step. Find a new estimate $\Theta^{(k+1)} = \{\{\theta_j^{(k+1)}\}_{j=1}^{K}, \nu^{(k+1)}\}$ with

$$\theta_j^{(k+1)} = arg\ Max_{\theta_j}\ Q_j^e(\theta_j), j = 1, \cdots, K; \quad \nu^{(k+1)} = arg\ Max_{\nu}\ Q^g(\nu). \tag{7}$$

Each maximization in eq.(7) will encounter two possibilities: able or unable to be solved analytically. When $f_j(x, w_j)$ is linear with respect to θ_j (e.g., $f_j(x, w_j) = w_j^T[x, 1]$), $Max_{\theta_j}\ Q_j^e(\theta_j)$ is solved by analytically solving $\frac{\partial Q_j^e}{\partial \theta_j} = 0$. When $f_j(x, w_j)$ is nonlinear with respect to w_j, however, this maximization can not be solved analytically. Moreover, due to the nonlinearity of softmax eq.(2), the maximization $Max_{\nu}\ Q^g(\nu)$ is insolvable analytically in any cases. For an analytically insolvable maximization, two extensions can be made. One is to use one of the conventional iterative optimization techniques (e.g., gradient ascent) to run an inner- loop iteration to solve this maximization. The other is , by some way, to find a new estimate such that

$$Q_j^e(\theta_j^{(k+1)}) \geq Q_j^e(\theta_j^{(k)}), j = 1, \cdots, K; \quad Q^g(\nu^{(k+1)}) \geq Q^g(\nu^{(k)}). \tag{8}$$

without requiring the satisfaction of eq.(7). According to Dempster, Lair and Rubin (1977), the algorithms that keep eq.(7) satisfied are called as EM, and the algorithms that keep only eq.(8) but not eq.(7) satisfied as the Generalized EM (GEM). Considering the difference between EM algorithms with and without inner-loop iterations made in the M step, we call the algorithm with all the maximization in eq.(7) being analytically solvable as *single-loop EM*, and the algorithms with inner-loop iteration as *double-loop EM*.

Jordan & Jacobs (1993) considered the case of linear $\beta_j(x, \nu) = \nu_j^T[x, 1]$ with $\nu = [\nu_1, \cdots, \nu_K]$ and semi-linear $f_j(w_j^T[x, 1])$ with nonlinear $f_j(.)$. They proposed a double-loop EM algorithm by using the *Iterative Recursive Least Square (IRLS)* method to implement the inner-loop iteration. For the more general cases of $\beta_j(x, \nu)$ and $f_j(x, \theta_j)$, Jordan & Xu (1993) have further showed that

an extended IRLS can be used for this inner loop. Actually, it can also be shown that IRLS and the extension are equivalent to solving eq.(6) by the so called *Fisher Scoring* method.

2 A New Gating Net and An Alternative EM Learning

For the original model discussed above, its gating net eq.(2) will cause one disadvantage. The nonlinearity of *softmax* makes the analytical solution of $Max_\nu\ Q^g(\nu)$ impossible even for the simple and useful cases that $\beta_j(x,\nu) = \nu_j^T[x,1]$ and $f_j(x^{(t)}, w_j) = w_j^T[x,1]$. That is, we do not have a single-loop EM algorithm for training this model even when all the maximizations related to expert nets are analytical solvable. We need to use either double-loop EM or GEM. Although it was shown by Dempster, Lair and Rubin (1977) that both EM (including single and double loop ones) and GEM will let the likelihood L keep satisfying $L(\Theta^{(k+1)}) \geq L(\Theta^{(k)})$ [1], their convergence properties and computing costs are quite different. For a single-loop EM, the convergence is guaranteed automatically without any external helps and regardless any initials. For a double-loop EM, e.g., the IRLS loop of Jordan & Jacobs (1993), the replacement of a single M-step by a whole convergence process of an inner-loop iteration will increase the computational costs considerably. Moreover, in order to guarantee the convergence of the inner loop, some safeguard measures (e.g., appropriately choosing the learning stepsize) are needed to ensure the inner-loop convergence, which will further increase computing costs. For a GEM, in its M step, a new estimate that satisfies eq.(7) is actually made by a nonlinear optimization technique. In general, the use of any existing optimization techniques needs some external and heuristic control or extensive extra searching to guarantee the satisfaction of eq.(7); and also the convergence speed of the whole outer-loop iteration is usually quite slow.

To overcome this disadvantage of the original gating net eq.(2), we propose a new gating net model

$$g_j(x,\nu) = \alpha_j P(x|\nu_j) / \textstyle\sum_{i=1}^K \alpha_i P(x|\nu_i),\ \sum_{j=1}^K \alpha_j = 1, \alpha_j \geq 0 \quad ,$$
$$P(x|\nu_j) = a_j(\nu_j)^{-1} b_j(x) exp\{c_j(\nu_j)^T t_j(x)\},\ \ b_j(x) \geq 0,\ a_j(\nu_j) = \textstyle\int b_j(x) exp\{c_j(\nu_j)^T t_j(x)\} dx \quad (9)$$

where $\nu = \{\alpha_j, \nu_j, j = 1, \cdots, K\}$. $P(x,\nu_j)$'s are density functions from the exponential family. $a_j(.), b_j(.), c_j(.), t_j(.)$ are prespecified functions. $t_j(x)$ is a sufficient statistics. The exponential family covers most useful density functions in practice. The commonly used one is Gaussian density

$$P(x|\nu_j) = (2\pi \det \Sigma_j)^{-\frac{1}{2}} \exp\{-\frac{1}{2}(x-m_j)^T \Sigma_j^{-1}(x-m_j)\},\ \Sigma_j \textit{ is positive definite} \quad (10)$$

[1] I.e., the convergence is guaranteed

with ν_j consisting of m_j, Σ_j.

In eq.(9), $g_j(x,\nu)$ is actually the posteriori probability $P(j|x)$ that x is assigned to the partition corresponding to the j-th expert net, obtained from Bayesian law

$$g_j(x,\nu) = P(j|x) = \alpha_j P(x|\nu_j)/P(x,\nu), \quad P(x,\nu) = \sum_{i=1}^{K} \alpha_i P(x|\nu_i), \tag{11}$$

Inserting this $g_j(x,\nu)$ into the model eq.(1), we get

$$P(y|x,\Theta) = \sum_{j=1}^{K} \frac{\alpha_j P(x|\nu_j)}{P(x,\nu)} P(y|x,\theta_j), \tag{12}$$

If we directly put this $P(y|x,\Theta)$ into eq.(4) and use the EM technique to this ML problem, we will again find that the maximization $Max_\nu\ Q^g(\nu)$ is analytically insolvable too. To avoid this difficulty, we rewrite eq.(12) into an equivalent form

$$P(y,x) = P(y|x,\Theta)P(x,\nu) = \sum_{j=1}^{K} \alpha_j P(x|\nu_j) P(y|x,\theta_j), \tag{13}$$

Assume that we have already known the parameters $\{\alpha_j\}_1^K, \{\nu_j\}_1^K, \{\theta_j\}_1^K$, then by taking integral over y on the both sides of eq.(13), we have $P(x,\nu) = \sum_{j=1}^{K} \alpha_j P(x|\nu_j)$. This suggests that we can easily obtain the model eq.(12) from eq.(13)—an asymmetrical representation of joint density.

Therefore, we cna accordingly modify eq.(4) into the following likelihood function:

$$L' = \sum_{t=1}^{N} \ln P(y^{(t)}, x^{(t)}) = \sum_{t=1}^{N} \ln\{\sum_{j=1}^{K} \alpha_j P(x^{(t)}|\nu_j) P(y^{(t)}|x^{(t)},\theta_j)\}. \tag{14}$$

With some derivations, we can obtain the two steps of the EM procedure for this ML problem.

(1) E-step.

$$h_j^{(k)}(y^{(t)}|x^{(t)}) = \frac{\alpha_j^{(k)} P(x^{(t)}|\nu_j^{(k)}) P(y^{(t)}|x^{(t)},\theta_j^{(k)})}{\sum_{i=1}^{K} \alpha_i^{(k)} P(x^{(t)}|\nu_i^{(k)}) P(y^{(t)}|x^{(t)},\theta_j^{(k)})}; \tag{15}$$

By letting both the numerator and denominator of eq.(15) be divided by $P(x)$ and noticing eq.(11), we can find that eq.(15) is identical to eq.(5). In addition, the objective functions $Q_j^e(\theta_j), j = 1,\cdots,K$ are the same as given in eq.(6). While the objective function $Q^g(\nu)$ can be further decomposed into

$$\begin{aligned} Q_j^g(\nu_j) &= \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}) \ln P(x^{(t)}|\nu_j),\ j = 1,\cdots,K; \\ Q^\alpha &= \sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(y^{(t)}|x^{(t)}) \ln \alpha_j, \quad with\ \alpha = \{\alpha_1,\cdots,\alpha_K\}. \end{aligned} \tag{16}$$

(2). M-step. Find a new estimate with

$$\begin{aligned}
\theta_j^{(k+1)} &= arg\ Max_{\theta_j}\ Q_j^e(\theta_j), j = 1, \cdots, K; \\
\nu_j^{(k+1)} &= arg\ Max_{\nu_j}\ Q_j^g(\nu_j), j = 1, \cdots, K; \\
\alpha^{(k+1)} &= arg\ Max_{\alpha}\ Q^{\alpha},\ s.t.\ \sum_{j=1}^{K} \alpha_j = 1.
\end{aligned} \tag{17}$$

The maximization for the expert nets keep the same as in eq.(7). However, For the gating net all the maximizations now become analytically solvable as long as $P(x|\nu_j)$ is from the exponential family eq.(9). That is, we have

$$\begin{aligned}
\nu_j^{(k+1)} &= \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)})} \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}) t_j(x^{(t)}), \\
\alpha_j^{(k+1)} &= \frac{1}{N} \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}),
\end{aligned} \tag{18}$$

Particularly, when $P(x|\nu_j)$ is Gaussian density eq.(10), the above first formula become

$$\begin{aligned}
m_j^{(k+1)} &= \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)})} \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}) x^{(t)}, \\
\Sigma_j^{(k+1)} &= \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)})} \sum_{t=1}^{N} h_j^{(k)}(y^{(t)}|x^{(t)}) [x^{(t)} - m_j^{(k)}][x^{(t)} - m_j^{(k)}]^T.
\end{aligned} \tag{19}$$

We show the results of a computer experiment by the above EM algorithm on the modified model in comparison with the original model with its EM learning. As shown in Fig.1(a), we have 1000 training points for learning an univariate mapping function which consists of two pieces of segments of linear regression functions. We consider such a mixture-of-experts model with $K = 2$. For expert nets, each $P(y|x, \theta_j)$ is Gaussian given by eq.(1) with linear $f_j(x, w_j) = w_j^T[x, 1]$. For the new gating net, each $P(x, \nu_j)$ in eq.(9) is Gaussian given by eq.(10). For the old gating net eq.(2), $\beta_1(x, \nu) = 0$ and $\beta_2(x, \nu) = \nu^T[x, 1]$.

The two lines through the clouds in Fig.1(a) are the estimated models of two expert nets. The ones obtained by the new and old learnings are almost the same. However, the learning speeds are considerably different. As shown in Fig.1(b), the new learning takes k=15 iterations when the log-likelihood converges to the value of -1271.8. It takes about 1351383 *flops* before getting converged[2]. For the old learning, we use the IRLS algorithm given in Jordan & Jacobs (1993) for the inner loop iteration. In experiments, we found that it usually took a large number of iterations

[2]Each operation of real addition, subtraction, multiplication and division is one *flop*, counted by the software PRO-MATLAB of MathWorks, Inc.

for the inner loop to converge. To save computations, we limit the maximum number of iterations by $\tau_{max} = 10$. We found that this did not obviously influence the total performance, but can save a lot of computations. From Fig.1(b), we see that its outer-loop converges around 16 iterations. Each inner-loop takes 290498 *flops* and the entire process takes 5312695 *flops*. So, we see that the new learning get about $4648608/1441475 = 3.9$ times speeding up[3]

Moreover, there is no need for any external adjustment to ensure the convergence of the new learning. However, for the old learning we found the direct use of IRLS will make the inner- loop diverge. A rescaling of the stepsize along the updating direction obtained by IRLS should be selected appropriately. If the scaling factor is not small enough, the inner loop is still unstable and will not converge. In our experiments, we choose the factor as 0.01. Usually, one needs a number of tries to get it appropriate. It certainly will cost more computations.

One disadvantage of the new gating net is that its number of parameters is $K(1 + d/2 + d^2/2)$ in comparison with $(K-1)d$ required by the old gating net, where d is the dimension of x.

3 Piecewise Nonlinear Function Approximation

In the EM learning for either the modified model or the original model, in order to make the learning problem can be solved by a single-loop EM algorithm instead of a double-loop EM or GEM, the maximization $Max_{\theta_j}\ Q_j^e(\theta_j), j = 1, \cdots, K$ given in eq.(6) should be analytically solvable. Therefore, the output of each expert net is usually assumed to be the simple form $f_j(x, w_j) = w_j^T[x, 1]$. Of course, this is a very useful case, in which the mixture-of-experts as a whole implements a soft version of piecewise linear approximation of nonlinear function. However, this is not the only case that single-loop EM applies. Here, we show that the modified model with a single loop EM can actually apply to a wide class of piecewise nonlinear function approximation problem.

Assume we have

$$f_j(x, w_j) = \sum_i w_{i,j}\phi_{i,j}(x) + w_{0,j} = w_j^T[\phi_j(x), 1], \tag{20}$$

with $\phi_{i,j}(x)$ being prespecified functions of x as a set of basis. It is not difficult to see that the resulted maximization problems $Max_{\theta_j}\ Q_j^e(\theta_j), j = 1, \cdots, K$ in eq.(6) will still become weighted least square problems which can be solved analytically. In fact, this case is equivalent to first transforming input x into x' by $x'_i = \phi_{i,j}(x)$ and then inputing x' into expert nets with $f_j(x', w_j) = w_j^T[x', 1]$.

[3]In fig.1(b), one can observe that the converged value of the likelihood the original model is slightly better than the new model, the reason is that eq.(14) is not exactly equivalent to eq.(13) for a finite training set.

However, this transformation can considerably generalize the model's function approximation ability.

One very useful special case is that $\phi_{i,j}(x)$ is canonical polynomial terms $x_1^{r_1} \cdots x_d^{r_d}$, $r_i \geq 0$. In this case, we can expect that the mixture-of-experts model will implement piecewise polynomial approximations. Another useful case is that $\phi_{i,j}(x)$ is $\prod_i \sin_i^r(j\pi x_1) \cos_i^r(j\pi x_1), r_i \geq 0$. In this case, the mixture-of- experts will implement piecewise trigeometric approximations. Furthermore, $\phi_{i,j}(x)$ may also be some more complicated basis functions. In addition, we can also mixedly use polynomial, trigeometric, and other basis functions in different expert nets or even in a same expert net to set up a hybrid model. As a result, we can expect that the model's approximation ability will greatly increased.

Figs.2(a)&(b) show the results of an computer experiment for piecewise polynomial approximation. The modified model with its EM learning proposed in the previous section is used. As shown in Fig.2(a), we have 1000 training points for learning an univariate mapping function which consists of two pieces of 3-rd polynomial functions. We consider such a mixture-of-experts model with $K = 2$. For expert nets, each $P(y|x, \theta_j)$ is Gaussian given by eq.(1) with $f_j(x, w_j) = w_{3,j}x^3 + w_{2,j}x^2 + w_{1,j}x + w_{0,j}$. In the new gating net eq.(9), each $P(x, \nu_j)$ is again Gaussian given by eq.(10).

The two curves through the clouds in Fig.2(a) are the estimated models of two expert nets. As shown in Fig.2(b), the log-likelihood converges to the value of -608.3 after about $k = 5$ iterations. The converged parameters for two experts nets are $w_1 = [w_{3,1}, w_{2,1}, w_{1,1}, w_{0,1}] = [0.7639, 0.0422, 0.4321, 0.1932]$ and $w_2 = [w_{3,2}, w_{2,2}, w_{1,2}, w_{0,2}] = [-0.014, -0.6751, 0.4321, 0.1932]$. we see from these parameters and Fig.2(b) that the higher order nonlinear regression has been made quite well.

Figs.3(a)& (b) show that results of an experiments on the same data in Fig.2(a) by using the modified model with linear $f_j(x, w_j) = w_j^T[x, 1]$, i.e., the same model as used in the previous section on data in Fig.1(a). From either the two estimated lines in Fig.3(a) or the converged likelihood value, we can see that the approximation is obviously worse than the results obtained in Figs.2(a) & 2(b).

4 Differences from A Related Model

Recently, Ghahramani & Jordan (1993) propose to solve function approximation via estimating joint density based on the mixture Gaussians

$$P(y,x,\Theta)=\sum_{j=1}^{K}\alpha_j P(y,x,\theta_j),\quad \textit{where } P(y,x,\theta_j) \textit{ is Gaussian density } N(m_j,\Sigma_j).$$

First, they use EM algorithm for ML estimation of the model's parameters, which are further partitioned into $m_j=\begin{pmatrix} m_j^x \\ m_j^y \end{pmatrix}$ and $\Sigma_j=\begin{bmatrix} \Sigma_j^x & \Sigma_j^{xy} \\ \Sigma_j^{yx} & \Sigma_j^y \end{bmatrix}$. Then, they get the linear regressions and weights

$$\begin{aligned} E(y|x,\theta_j) &= m_j^y+\Sigma_j^{xy}(\Sigma_j^x)^{-1}(x-m_j^x),\ j=1,\cdots,K \\ \omega_{j,x} &= \frac{P(x|m_j^x,\Sigma_j^x)}{\sum_{i=1}^{K}P(x|m_i^x,\Sigma_i^x)},\ j=1,\cdots,K,\ P(x|m_j^x,\ \Sigma_j^x) \textit{ is Gaussian density } N(m_j^x,\Sigma_j^x) \end{aligned} \tag{21}$$

Third, they put $E(y|x,\theta_j)$ into eq.(3) with $\omega_{j,x}$ replacing $g_j(x,\nu)$ in order to get the output for function approximation in one of the three modes.

In the special case that (a) $P(x|\nu_j)$ in eq.(9) is Gaussian, (b) $\alpha_1=\cdots=\alpha_k$, (c) $P(y|x,\theta_j)$ is Gaussian, and (d) $f_j(x,w_j)=w_j^T[x,1]$, then our method given in sec.2 provides the same result as Ghahramani & Jordan (1993) although the parameterizations of the two methods are different. Furthermore, it is also possible to replace $\omega_{j,x}$ in eq.(21) by $\alpha_j P(x|m_j^x,\Sigma_j^x)/\sum_{i=1}^{K}\alpha_i P(x|m_i^x,\Sigma_i^x)$ to let the two methods keep equivalent for the cases of unequal priori α_j.

However, the Gaussian Mixtures method of Ghahramani & Jordan (1993) does not apply the following three types of more general cases that our method are able to solve:

(1) The cases that $f_j(x,w_j)=w_j^T[\phi_j(x),1]$, i.e., the cases that make piecewise nonlinear function approximation discussed in the previous section. In these cases, even when both $P(x|\nu_j)$ and $P(y|x,\theta_j)$ are Gaussians, the joint density $P(y,x|\nu_j,\theta_j)$ is no longer Gaussian. So, eq.(13) is not Gaussian Mixtures.

(2) The cases that $P(x|\nu_j)$ is from exponential family but not Gaussian. As we discussed in sec.2, our EM learning works still in the cases. However, the joint density $P(y,x|\nu_j,\theta_j)$ is now no longer Gaussian too.

(3) The general cases that $f_j(x,w_j)$ is nonlinear with respect to w_j or that $P(y,x|\nu_j,\theta_j)$ is not Gaussian. In these cases, eq.(13) is certainly not Gaussian Mixtures. Although we have no a single-loop EM to tackle the tasks, we can still use a double-loop EM with IRLS (or extended IRLS) to solve the problems.

Finally, we like to point out that the method proposed in sec.2 can also be extended to the hierarchical architecture for mixtures-of-experts of Jacobs&Jordan (1993) so that single-loop EM can be used to facilitate its training. In addition, by using $f_j(x, w_j) = w_j^T[\phi_j(x), 1]$, we can expect to implement tree structure nonlinear approximation by polynomial or trigeometric basis functions piecewisely.

Acknowledgements

This project was supported in part by a grant from the Mcdonnell-Pew Foundation, by a grant from ATR Auditory and Visual Perception Research Laboratories, by a grant from Siemens Corporation, by grant IRI-9013991 from the National Science Foundation, an by grant N00014-90-J-1942 from the Office of Naval Research. Jordan is NSF Presidential Young Investigator. Hinton is a fellow of the Canadian Institute for Advanced Research.

REFERENCES

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977), Maximum- likelihood from incomplete data via the EM algorithm, *J. of Royal Statistical Society, B39*, pp1-38.

Ghahramani, Z, and Jordan, M.I.(1993), Function approximation via density estimation using the EM approach, Technical Report 9304, Dept. of Brain and Cognitive Science, MIT, Cambridge, MA.

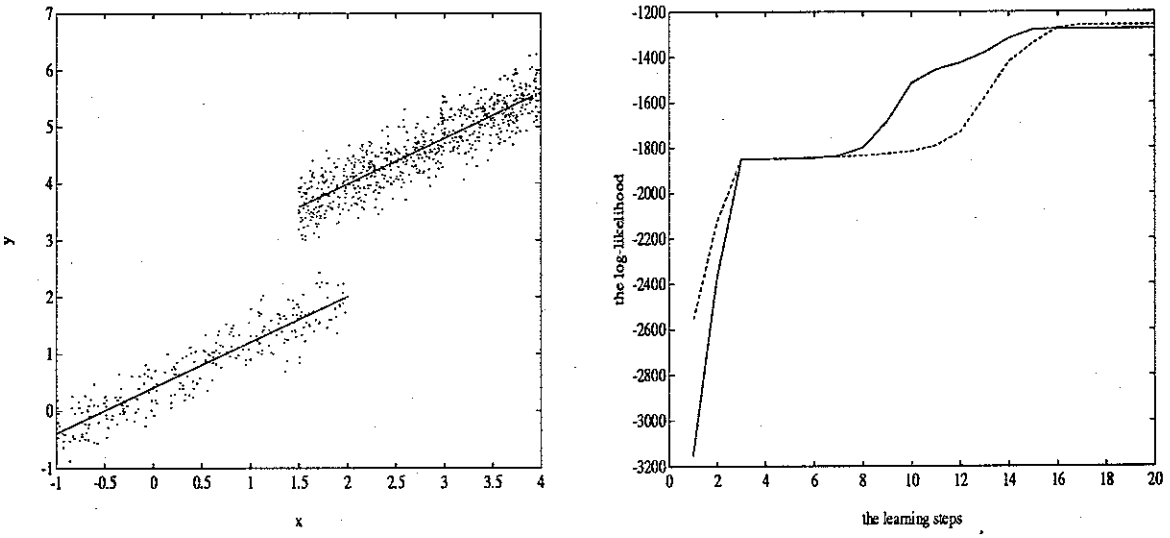
Jacobs, R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E., (1991), Adaptive mixtures of local experts, *Neural Computation, 3*, pp 79-87.

Jordan, M.I. and Jacobs, R.A. (1992a), Hierarchies of adaptive experts, *Advances in Neural Information Processing System 4*, eds., J.E.Moody, S.Hanson and R.P.Lippmann, San Mateo: Morgan Kaufmann Pub., pp 985-992.

Jordan, M.I. and Jacobs, R.A. (1992b), Hierarchies mixtures of experts and the EM algorithm, MIT Computational Cognitive Science, Technical Report 9203, Dept. of Brain and Cognitive Science, MIT, Cambridge, MA.

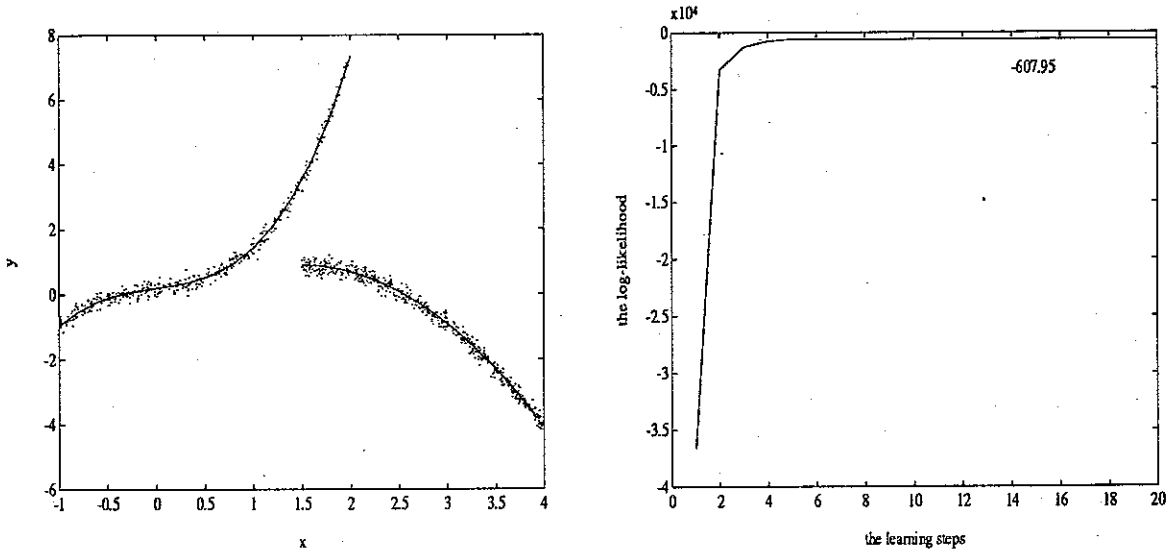
Jordan, M.I. and Xu, L. (1993), Convergence properties of the EM approach to learning in mixture-of-experts architectures, MIT Computational Cognitive Science, Technical Report 9302, Dept. of Brain and Cognitive Science, MIT, Cambridge, MA.

Xu, L., and Jordan, M.I. (1993), Theoretical and Experimental Studies of The EM Algorithm for Unsupervised Learning Based on Finite Gaussian Mixtures, MIT Computational Cognitive Science, Technical Report 9301, Dept. of Brain and Cognitive Science, MIT, Cambridge, MA.



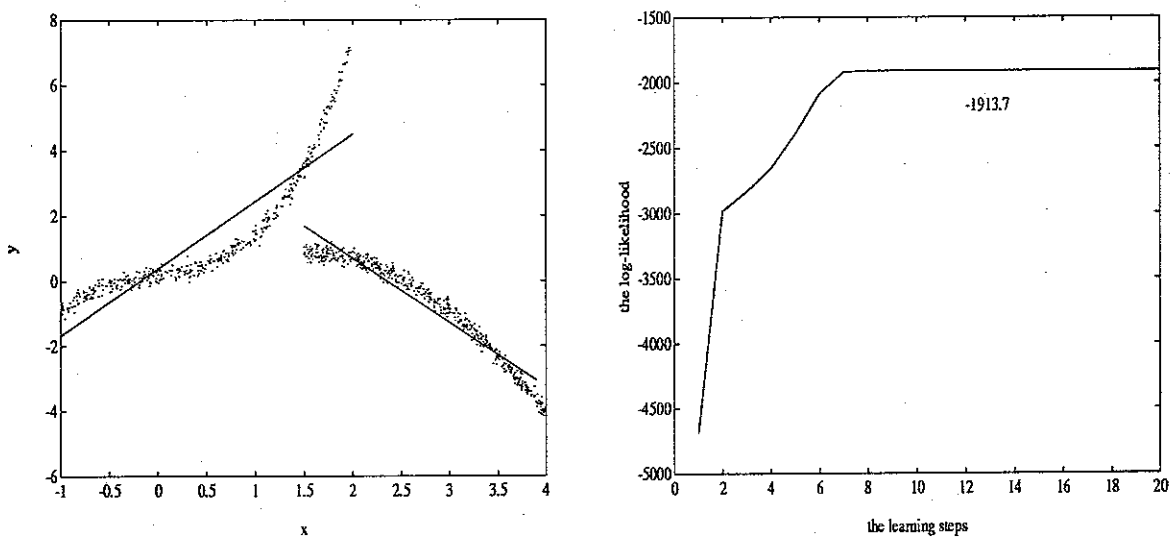
(a) (b)

Figure 1: (a) 1000 samples from $y = a_1 x + a_2 + \varepsilon, a_1 = 0.8, a_2 = 0.4, x \in [-1, 1.5]$ with prior $\alpha_1 = 0.25$ and $y = a_1' x + a_2' + \varepsilon, a_1' = 0.8, a_2 =' 2.4, x \in [1, 4]$ with prior $\alpha_2 = 0.75$, where x is uniform random variable and z is from Gaussian $N(0, 0.3)$. The two lines through the clouds are the estimated models of two expert nets. The ones obtained by the two learning are almost the same. (b) The changes of the log-likelihood as the iteration goes. The solid line is for the modified learning. The dotted line is for the original learning (the outer-loop iteration



(a) (b)

Figure 2: Piecewise 3rd polynomial approximation. (a) 1000 samples from $y = a_1x^3 + a_3x + a_4 + \varepsilon, x \in [-1, 1.5]$ with prior $\alpha_1 = 0.4$ and $y = a_2'x^2 + a_3'x^2 + a_4' + \varepsilon, x \in [1, 4]$ with prior $\alpha_2 = 0.6$, where x is uniform random variable and z is from Gaussian $N(0, 0.15)$. The two curves through the clouds are the estimated models of two expert nets. (b) The changes of the log-likelihood as the iteration goes.



(a) (b)

Figure 3: Piecewise linear approximation. (a) The two lines estimated for two expert nets. (b) The changes of the log-likelihood as the iteration goes.