Variational Mixture of Experts For Classification with Applications to Landmine Detection

Seniha Esen Yuksel, Paul Gader
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, USA
{*seyuksel, pgader*}*@cise.ufl.edu*

***Abstract*—In this paper, we (1) provide a complete framework for classification using Variational Mixture of Experts (VME); (2) derive the variational lower bound; and (3) apply the method to landmine, or simply mine, detection and compare the results to the Mixtures of Experts trained with Expectation Maximization (EMME). VME has previously been used for regression and Waterhouse explained how to apply VME to classification (which we will call as VMEC). However, the steps to train the model were not made clear since the equations were applicable to vector valued parameters as opposed to matrices for each expert. Also, a variational lower bound was not provided. The variational lower bound provides an excellent stopping criterion that resists over-training. We demonstrate the efficacy of the method on real-world mine classification; in which, training robust mine classification algorithms is difficult because of the small number of samples per class. In our experiments VMEC consistently improved performance over EMME.**

***Keywords*-Variational Mixture of Experts; Landmine Detection; Classification; Lower Bound; Ensemble Learning**

I. Introduction

Hierarchical mixture of experts (HME) introduced by Jordan et *al*.[1] is a tree-like architecture that makes soft splits at both the experts and the gates. Training is accomplished through the Expectation Maximization (EM) algorithm which decouples the learning at the experts and gates. The EM algorithm for HME (EM-HME) converges linearly [2]; and provides a probabilistic form with easily interpreted parameters. However, EM-HME suffers from sensitivity to initialization, does not regularize the parameters, or use any prior information; and hence it is prone to over-training. To address some of these problems, VME (a Bayesian approach) was proposed in [3], [4], [5], [6] for regression. Although the similarities of the classification and regression algorithms were discussed in [6]; a clear framework was not given for classification. More explicitly; in a K class classification, there are K weight vectors for each expert instead of a single weight vector. Hence, instead of a single hyper-parameter, we assumed K hyper-parameters for each expert. Similarly, instead of assuming a single distribution per expert, we used K distributions per expert. In addition, a distribution over the hyper-parameters was not assumed in [6]; but we found such an assumption to be necessary to calculate the lower bound. With these assumptions, the joint distribution was modified to be a product of the aforementioned distributions; and the lower bound was derived using this modified joint distribution.

Our interest in this problem was motivated by mine detection. In previous work [7], EM-HME was used in decision fusion for mine detection and instead of a traditional mine/non-mine decision, the experts were trained on specific classes of mines and non-mines, i.e. High Metal Anti Tank (HMAT), Low Metal Anti-personnel (LMAP), LMAT, HMAP, metallic and non-metallic clutter. However, since features still overlap, the EM-HME algorithm leads to over-fitting. In this paper, we provide a complete VMEC framework, and compare the results of VMEC to EM-HME.

II. Mixture of Experts for Classification

For a K-class problem, let $D = \{X, Y\}$ denote the data with $X = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ and $Y = \{\mathbf{y}^{(n)}\}_{n=1}^{N}$ where $\mathbf{y}^{(n)}$ is of length K and $y_k^{(n)} = 1$ if $x^{(n)}$ belongs to class k and 0 otherwise. Let I be the number of experts and $k : 1...K$ be the class index.

In a mixture of experts architecture composed of experts and a gate as illustrated in Fig.1 ; $\widehat{y}_{ik}$ is the output of expert i for class k; and $\mathbf{w}_{ik}$ is the corresponding weight vector. On the upper level, g_i is one of the outputs of the gating network; and $\mathbf{v}_i$ is the related weight vector. For a given $\mathbf{x}^{(n)}$; the expert network i produces a prediction with probability $P_i(\mathbf{y}^{(n)})$ following a multinomial distribution with mean $\widehat{y}_{ik}$ such that

$$\widehat{y}_{ik} = \frac{\exp(\mathbf{w}_{ik}^T \mathbf{x})}{\sum_{r=1}^{K} \exp(\mathbf{w}_{ir}^T \mathbf{x})}$$

and

$$P_i(\mathbf{y}) = \prod_k \widehat{y}_{ik}^{y_k}$$

The gate estimates the probability of each expert; and its i^{th} output g_i is a softmax nonlinearity given as:

$$g_i = \frac{\exp(\mathbf{v}_i^T \mathbf{x})}{\sum_{m=1}^{I} (\exp \mathbf{v}_m^T \mathbf{x})}$$

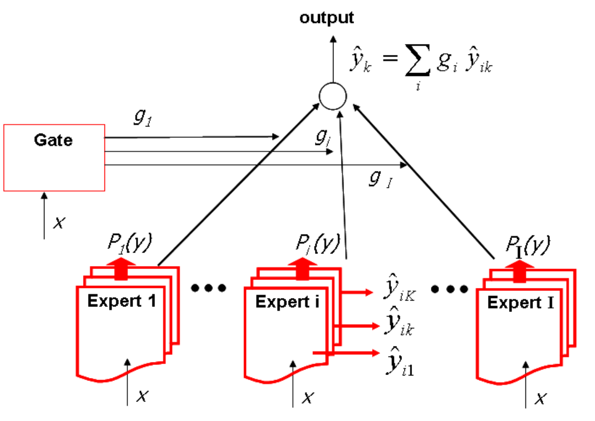


Figure 1. Mixture of Experts Architecture

And the total output of the architecture is:

$$\widehat{y}_k = \sum_i g_i \ \widehat{y}_{ik}$$

Hence, total probability of the output Y is:

$$P(Y|\mathbf{w},\mathbf{v}) = \prod_{n=1}^{N} \sum_{i=1}^{I} g_i^{(n)} P_i(\mathbf{y}^{(n)})$$

where $\mathbf{w} = \{\{\mathbf{w}_{ik}\}_{i=1}^{I}\}_{k=1}^{K}$ and $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^{I}$. These parameters $\{\mathbf{w},\mathbf{v}\}$ can be estimated using EM [1], [2] with hidden variables $Z = \{\{z_i^{(n)}\}_{n=1}^{N}\}_{i=1}^{I}$:

$$z_i^{(n)} = \begin{cases} 1 & \text{if } \mathbf{x}^{(n)} \in R_i \ ; \\ 0 & \text{otherwise} \end{cases}$$

where R_i is the region specified by expert i. The expectations of these missing variables are [1]:

$$h_i^{(n)} = \frac{g_i^{(n)} P_i(\mathbf{y}^{(n)})}{\sum_j g_j^{(n)} P_j(\mathbf{y}^{(n)})}$$

Since $z_i^{(n)}$ represents using expert i for data point n; we can write $P(Z|\mathbf{v}_i) = g_i$. Therefore, the complete data distribution becomes:

$$P(Y,Z|\mathbf{w},\mathbf{v}) = \prod_n \prod_i \left(g_i^{(n)} P_i(y^{(n)}) \right)^{z_i^{(n)}}$$

III. VME FOR CLASSIFICATION (VMEC)

Let $\Theta = \{\{\mathbf{v}_i, \mathbf{w}_{ik}\}_{i=1}^{I}\}_{k=1}^{K}$ denote the gate and expert parameters. We place Gaussian priors $\Phi = \{\{\mu_i, \alpha_{ik}\}_{i=1}^{I}\}_{k=1}^{K}$ on the gate and the experts:

$$\begin{aligned} P(\mathbf{w}|\alpha) &= \prod_{i,k} P(\mathbf{w}_{ik}|\alpha_{ik}) = \prod_{i,k} N(\mathbf{w}_{ik}|0, \alpha_{ik}^{-1}\mathbf{I}) \\ P(\mathbf{v}|\mu) &= \prod_i \quad P(\mathbf{v}_i|\mu_i) \quad = \prod_i N(\mathbf{v}_i|0, \mu_i^{-1}\mathbf{I}) \end{aligned}$$

We also assume that the hyperparameters are Gamma distributed:

$$\begin{aligned} P(\mu) &= \prod_i \quad P(\mu_i) \quad = \prod_i \ Gam(\mu_i|c_0, d_0) \\ P(\alpha) &= \prod_{i,k} P(\alpha_{ik}) = \prod_{i,k} Gam(\alpha_{ik}|a_0, b_0) \end{aligned}$$

Hence the joint distribution is:

$$P(\Theta,\Phi,Z,D) = P(Y,Z|\mathbf{w},\mathbf{v})P(\mathbf{w}|\alpha)P(\alpha)P(\mathbf{v}|\mu)P(\mu)$$

In the variational approach, the goal is to find the distribution Q that best approximates the posterior distribution, so the evidence $P(D)$ is decomposed using

$$\log P(D) = F(Q) + KL(Q||P)$$

where

$$F(Q) = \int Q(\Theta,\Phi,Z) \log \frac{P(\Theta,\Phi,Z,D)}{Q(\Theta,\Phi,Z)} d\Theta d\Phi dZ$$

and

$$KL(Q||P) = -\int Q(\Theta,\Phi,Z) \log \frac{P(\Theta,\Phi,Z|D)}{Q(\Theta,\Phi,Z)} d\Theta d\Phi dZ$$

The Q distribution minimizes the KL-divergence; however, working on the KL-divergence would be intractable, so we maximize the free energy F instead [8]. We assume the approximating distribution factorizes as:

$$Q(\Theta,\Phi,Z) = Q(Z) \prod_i Q(\mathbf{v}_i) Q(\mu_i) \prod_k Q(\mathbf{w}_{ik}) Q(\alpha_{ik})$$

Plugging the joint distribution and the Q distribution into the lower bound equation, and taking the expectations wrt. to all the other variables, we obtain the Q distributions as:

$$\begin{aligned} Q^*(\mathbf{w}_{ik}) &= N(\mathbf{w}_{ik}|\overline{\mathbf{w}}_{ik}, A_{\mathbf{w}_{ik}}) \\ Q^*(\alpha_{ik}) &= Gam(\alpha_{ik}|a_p, b_p) \\ Q^*(\mathbf{v}_i) &= N(\mathbf{v}_i|\overline{\mathbf{v}}_i, A_{\mathbf{v}_i}) \\ Q^*(\mu_i) &= Gam(\mu_i|c_p, d_p) \\ Q^*(Z) &= \prod_{n=1}^{N} \prod_{i=1}^{I} {h_i^{(n)}}^{z_i^{(n)}} \end{aligned}$$

Here $\overline{\mathbf{w}}_{ik}$ and $\overline{\mathbf{v}}_i$ are the means of the Gaussians and they are found using Newton-Raphson updates. The covariance matrices $A_{\mathbf{w}_{ik}}$ and $A_{\mathbf{v}_i}$ are the inverse of the negative Hessian matrices; $A_{\mathbf{w}_{ik}} = -H_w^{-1}$ and $A_{\mathbf{v}_i} = -H_v^{-1}$.

For a learning rate η, expert parameters are found by:

$$\mathbf{w}_{ik}^{(p+1)} = \mathbf{w}_{ik}^{(p)} - \eta H_w^{-1} G_w$$

where

$$\begin{aligned} G_w &= \sum_n \overline{h}_i^{(n)} (y_k^{(n)} - \widehat{y}_{ik}^{(n)}) \mathbf{x}^{(n)} - \overline{\alpha}_{ik} \mathbf{w}_{ik} \\ H_w &= -\sum_n \overline{h}_i^{(n)} \widehat{y}_{ik}^{(n)} (1 - \widehat{y}_{ik}^{(n)}) (\mathbf{x}^{(n)}) (\mathbf{x}^{(n)})^T - \overline{\alpha}_{ik}.I \end{aligned}$$

Similarly, the updates to the gating parameters follow

$$\mathbf{v}_i^{(p+1)} = \mathbf{v}_i^{(p)} - \eta H_v^{-1} G_v$$

where

$$\begin{aligned} G_v &= \sum_n (h_i^{(n)} - g_i^{(n)}) \mathbf{x}^{(n)} - \overline{\mu}_i \mathbf{v}_i \\ H_v &= -\sum_n g_i^{(n)} (1 - g_i^{(n)}) (\mathbf{x}^{(n)}) (\mathbf{x}^{(n)})^T - \overline{\mu}_i.I \end{aligned}$$

Newton-Raphson updates are continued in a loop until $P(Y,Z|\mathbf{w},\mathbf{v})$ converges; and the parameters found at the last iteration are taken to be $\overline{\mathbf{w}}_{ik}$ and $\overline{\mathbf{v}}_i$; where $A_{\mathbf{w}_{ik}}$ and $A_{\mathbf{v}_i}$ are their covariance matrices.

The updates for expert hyper-hyperparameters are:

$$a_p = a_0 + \frac{d}{2}$$
$$b_p = b_0 + \frac{1}{2}(\mathbf{w}_{ik}^T \mathbf{w}_{ik} + Trace(A_{\mathbf{w}_{ik}}))$$

and similar equations apply for the gate hyper-hyperparameters c_p and d_p. As a result, hyperparameter updates become $\alpha_{ik}^{(p+1)} = a_p/b_p$ and $\mu_i^{(p+1)} = c_p/d_p$.

A. VMEC Lower Bound:

Parameter updates are continued until the lower bound converges; and the lower bound provides a test of correctness as it is supposed to be nondecreasing at each re-estimation of the parameters. Expanding the integral and evaluating the expectations, we arrive at the closed form solution for the lower bound as:

$$\begin{aligned}
F(Q) &= \int Q(\Theta, \Phi, Z) \log \frac{P(\Theta, \Phi, Z, D)}{Q(\Theta, \Phi, Z)} d\Theta d\Phi dZ \\
&= \sum_{n,i} E_{Z,\mathbf{v},\mathbf{w}}[\log P(Y, Z|X, \mathbf{v}, \mathbf{w})] \\
&+ \sum_i E[\log P(\mathbf{v}_i|\mu_i)] + \sum_i E[\log P(\mu_i)] \\
&+ \sum_{i,k} E[\log P(\mathbf{w}_{ik}|\alpha_{ik}) + \sum_{i,k} E[\log P(\alpha_{ik})] \\
&- \sum_i E[\log Q(\mathbf{v}_i)] - \sum_i E[\log Q(\mu_i)] \\
&- \sum_{i,k} E[\log Q(\mathbf{w}_{ik})] - \sum_{i,k} E[\log Q(\alpha_{ik})] - E[\log Q(Z)]
\end{aligned}$$

where

$$\begin{aligned}
E_{\mathbf{w},\alpha}[\log P(\mathbf{w}|\alpha)] &= \frac{d}{2}[\psi(a_p) - \log b_p] - \frac{d}{2}\log(2\pi) \\
&\quad - \frac{a_p}{2b_p}\left(\overline{\mathbf{w}}_{ik}^T \overline{\mathbf{w}}_{ik} + Trace(A_{\mathbf{w}_{ik}})\right) \\
E_\alpha[\log P(\alpha)] &= a_0 \log b_0 - b_0(a_p/b_p) - \log\Gamma(a_0) \\
&\quad + (a_0 - 1)[\psi(a_p) - \log b_p]
\end{aligned}$$

$$\begin{aligned}
E_\alpha[\log Q(\alpha)] &= -\log\Gamma(a_p) + (a_p - 1)\psi(a_p) \\
&\quad + \log b_p - a_p
\end{aligned}$$

$$\begin{aligned}
E_{\mathbf{w}}[\log Q(\mathbf{w}_{ik})] &= \frac{1}{2}\log|A_{\mathbf{w}_{ik}}| + \frac{d}{2}(\log(2\pi) + 1) \\
E_Z[\log Q(Z)] &= \textstyle\sum_n h_i^{(n)} \log h_i^{(n)} \\
E_{Z,\mathbf{v}}[\log P(Z|\mathbf{v}_i)] &= \textstyle\sum_n h_i^{(n)} \log \overline{g}_i^{(n)} \\
E_{Z,\mathbf{w}}[\log P(Y|Z, \mathbf{w}_{ik})] &= \textstyle\sum_n h_i^{(n)} \log \overline{P}_i(\mathbf{y}^{(n)})
\end{aligned}$$

Expressions for the gate $E_\mu[\log P(\mu_i)]$), $E_\mu[\log Q(\mu_i)]$), $E_{\mathbf{v},\mu}[\log P(\mathbf{v}_i|\mu_i)]$), $E_{\mathbf{v}}[\log Q(\mathbf{v}_i)]$ are similar to those of the experts. VMEC algorithm updates the parameters through the Expectation (E) and Maximization (M) steps until the change in the lower bound becomes less than a threshold (e^{-5} in our case).

B. Training for VMEC:

1) For a 1-of-K class problem, initialize the number of experts I, parameters and the hyperparameters.
2) E-step: Compute the expert and gating outputs $\widehat{y}_{ik}^{(n)}$, $g_i^{(n)}$ as well as the expert probabilities $P_i(\mathbf{y}^{(n)})$ and the posterior probabilities $h_i^{(n)}$.
3) M-step: Compute the new expert parameters $\mathbf{w}_{ik}^{(p+1)}$ and the new gating parameters $\mathbf{v}_i^{(p+1)}$ using Newton-Raphson updates.
4) Update the hyperparameters $\alpha_{ik}^{(p+1)}$ and $\mu_i^{(p+1)}$.
5) Check the convergence of the lower bound. Go to Step 2 if $F(p+1) - F(p) > 1e-5$; else terminate.

IV. Experimental Results

Synthetic data was generated by sampling from two Gaussian distributions with standard deviation 0.1 at means $(0.7, 0.7)$ and $(0.5, 0.7)$. For testing, 200 points were generated from each class. For training, the number of points were increased at each iteration, from 10 points to 60 points per class. Classification performance on test data is displayed in Fig.2 where VME consistently performs better. VME gives better results even when we increase the number of points in the training set because of the fact that (1) the results y_k sum to 1, and (2) the VME gate prefers fewer experts.

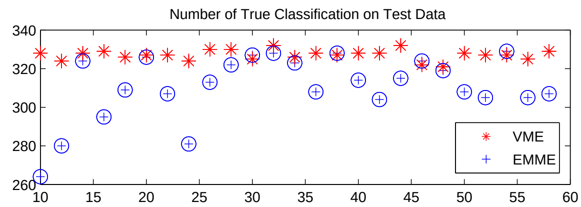
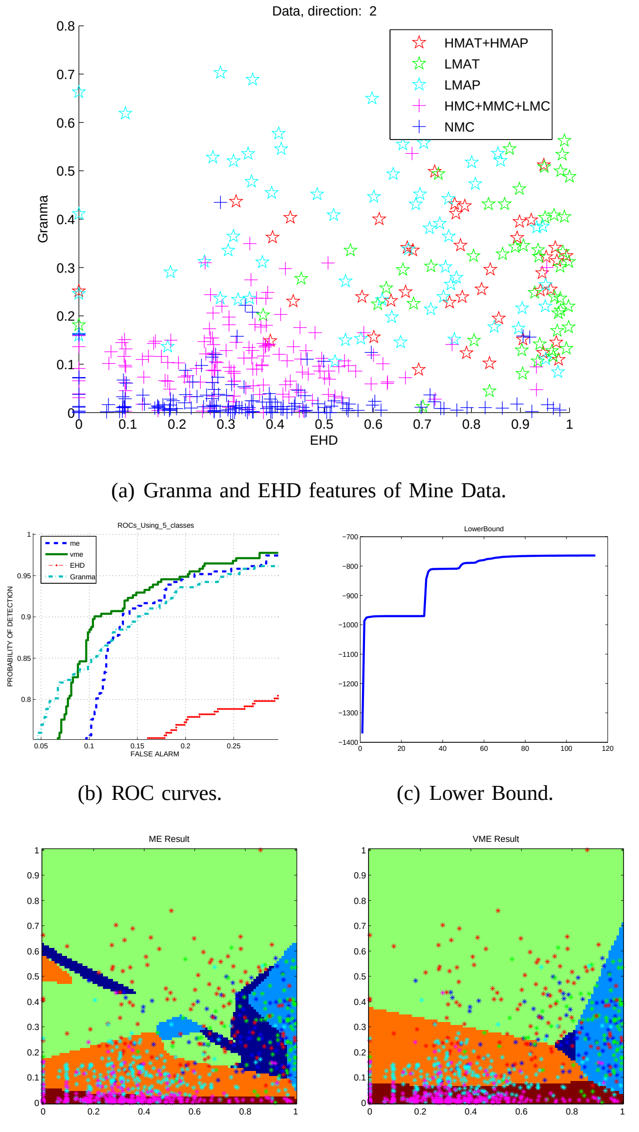


Figure 2. Classification for varying numbers of training data. VME performs better and shows consistent behaviour.

V. Landmine Detection

The mine dataset and the two mine features, Granma and EHD are completely described in [7], [9], [10]. In Fig.3(a), the two mines features (Granma and EHD) are displayed. Different colors represent the five different classes. The first three classes are High metal mines, LMAT, and LMAP. The last two classes are the metallic and non-metallic clutter. In Fig.3(b), Receiver operating characteristic (ROC) curves zoomed around 90%PD are displayed for 1 experiment with 5 classes, 5 experts, and 10 fold cross-validation. In Fig.3(c), the increase in the lower bound in one of the ten folds is displayed. The sharp increase in the lower bound corresponds to one of the hyper-parameter updates. In Fig.3(d), on the same fold, classification using EMME is given; where over-training can be observed. In Fig.3(e), over-training is avoided using VME. The solid colors represent the class of the maximum decision in that region. In Table I, for 5 experiments on mine data, VME algorithm gives an average of 11.6% PFA at 90%PD, whereas EMME stays around 16.20%PFA at 90%PD.



(a) Granma and EHD features of Mine Data.

(b) ROC curves.

(c) Lower Bound.

(d) EMME Classification results.

(e) VME Classification results.

Figure 3. Landmine Results. In a setting of 5 classes and 5 experts, for a 10 fold cross-validation, VME algorithm consistently increases detection rates to around 90/11.6 percent from 90/16.2 of EMME.

VI. Future Work

VME improved the performance in these experiments with real-world data. Our future work will involve sensitivity analysis, finding the optimal number of experts, and investigating more suitable experts.

Table I
PFAs at 90% PD for 5 cross-validation runs

Experiment	1	2	3	4	5
EMME	0.17	0.16	0.15	0.18	0.15
VME	0.12	0.11	0.10	0.13	0.12

Acknowledgment

This research was partially supported by NSF Grant No. 0730484.

References

[1] M. I. Jordan, “Hierarchical mixtures of experts and the EM algorithm,” *Neural Computation*, vol. 6, pp. 181–214, 1994.

[2] M. I. Jordan and L. Xu, “Convergence results for the EM approach to mixtures of experts architectures,” *Neural Networks*, vol. 8, pp. 1409–1431, 1995.

[3] C. M. Bishop and M. Svensen, “Bayesian hierarchical mixtures of experts,” in *Proceedings Nineteenth Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 57 – 64.

[4] N. Ueda and Z. Ghahramani, “Optimal model inference for Bayesian mixture of experts,” in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, vol. 1, 2000, pp. 145–154.

[5] S. Waterhouse, D. Mackay, and T. Robinson, “Bayesian methods for mixtures of experts,” in *Adv. Neur. Inf. Proc. Sys. 7*. MIT Press, 1996, pp. 351–357.

[6] S. R. Waterhouse, “Classification and regression using mixtures of experts,” Ph.D. dissertation, Department of Engineering, University of Cambridge, 1997.

[7] S. Yuksel, G. Ramachandran, P. Gader, J. Wilson, D. Ho, and G. Heo, “Hierarchical methods for landmine detection with wideband electro-magnetic induction and ground penetrating radar multi-sensor systems,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, July 2008, pp. II–177–II–180.

[8] C. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, August 2006.

[9] H. Frigui and P. Gader, “Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic K-nearest neighbor classifier,” *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 1, pp. 185–199, Feb. 2009.

[10] G. Ramachandran, P. Gader, and J. Wilson, “Fast physics-based mine detection algorihtms for wide-band electromagnetic induction sensors,” in *SPIE Defense, Security and Sensing*, April 2009, pp. 7303–77.