Twenty Years of Mixture of Experts

Seniha Esen Yuksel, *Member, IEEE*, Joseph N. Wilson, *Member, IEEE*, and Paul D. Gader, *Fellow, IEEE*

***Abstract*— In this paper, we provide a comprehensive survey of the mixture of experts (ME). We discuss the fundamental models for regression and classification and also their training with the expectation-maximization algorithm. We follow the discussion with improvements to the ME model and focus particularly on the mixtures of Gaussian process experts. We provide a review of the literature for other training methods, such as the alternative localized ME training, and cover the variational learning of ME in detail. In addition, we describe the model selection literature which encompasses finding the optimum number of experts, as well as the depth of the tree. We present the advances in ME in the classification area and present some issues concerning the classification model. We list the statistical properties of ME, discuss how the model has been modified over the years, compare ME to some popular algorithms, and list several applications. We conclude our survey with future directions and provide a list of publicly available datasets and a list of publicly available software that implement ME. Finally, we provide examples for regression and classification. We believe that the study described in this paper will provide quick access to the relevant literature for researchers and practitioners who would like to improve or use ME, and that it will stimulate further studies in ME.**

***Index Terms*— Applications, Bayesian, classification, comparison, hierarchical mixture of experts (HME), mixture of Gaussian process experts, regression, statistical properties, survey, variational.**

I. Introduction

SINCE its introduction 20 years ago, the mixture of experts (ME) model has been used in numerous regression, classification, and fusion applications in healthcare, finance, surveillance, and recognition. Although some consider ME modeling to be a solved problem, the significant number of ME studies published in the last few years suggests otherwise. These studies incorporate experts based on many different regression and classification models such as support vector machines (SVMs), Gaussian processes (GPs), and hidden Markov models (HMMs), to name just a few. Combining these models with ME has consistently yielded improved performance. The ME model is competitive for regression problems with nonstationary and piecewise continuous data, and for nonlinear classification problems with data that contain natural distinctive subsets of patterns. ME has a well-studied statistical basis, and models can be easily trained with well-known techniques such as expectation-maximization (EM), variational learning, and Markov chain Monte Carlo (MCMC) techniques including Gibbs sampling. We believe that further research can still move ME forward and, to this end, we provide a comprehensive survey of the past 20 years of the ME model. This comprehensive survey can stimulate further ME research, demonstrate the latest research in ME, and provide quick access to relevant literature for researchers and practitioners who would like to use or improve the ME model.

The original ME model introduced by Jacobs *et al.* [1] can be viewed as a tree-structured architecture, based on the principle of divide and conquer, having three main components: several experts that are either regression functions or classifiers; a gate that makes soft partitions of the input space and defines those regions where the individual expert opinions are trustworthy; and a probabilistic model to combine the experts and the gate. The model is a weighted sum of experts, where the weights are the input-dependent gates. In this simplified form, the original ME model has three important properties: 1) it allows the individual experts to specialize on smaller parts of a larger problem; 2) it uses soft partitions of the data; and 3) it allows the splits to be formed along hyperplanes at arbitrary orientations in the input space [2]. These properties support the representation of nonstationary or piecewise continuous data in a complex regression process, and identification of the nonlinearities in a classification problem. Therefore, to understand systems that produce such nonstationary data, ME has been revisited and revived over the years in many publications. The linear experts and the gate of the original ME model have been improved upon with more complicated regression or classification functions, the learning algorithm has been changed, and the mixture model has been modified for density estimation and for time-series data representation.

In the past 20 years, there have been solid statistical and experimental analyses of ME, and a considerable number of studies have been published in the areas of regression, classification, and fusion. ME models have been found useful in combination with many current classification and regression algorithms because of their modular and flexible structure. In the late 2000s, numerous ME studies have been published, including [3]–[30]. Although many researchers think of ME only in terms of the original model, it is clear that the ME model is now much more varied and nuanced than when it was introduced 20 years ago. In this paper, we attempt to address all these changes and provide a unifying view that covers all these improvements showing how the ME model has progressed over the years. To this end, we divide the literature into distinct areas of study and keep a semichronological order within each area.

Manuscript received June 30, 2011; revised March 16, 2012; accepted May 10, 2012. Date of publication June 11, 2012; date of current version July 16, 2012. This work was supported in part by the National Science Foundation under Grant 0730484.

The authors are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: seyuksel@cise.ufl.edu; jnw@cise.ufl.edu; pgader@cise.ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2200299

To see the benefit that ME models have provided, we can look at a 2008 survey that identified the top 10 most influential algorithms in the data mining area. It cited C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, k-nearest neighborhood, naive Bayes, and classification and regression trees (CART) [31]. Although ME is not explicitly listed here, it is closely related to most of these algorithms, and has been shown to perform better than some of them and combined with many of them to improve their performance. Specifically, MEs have often been trained with EM [32] and have been initialized using k-Means [15], [33]. It has been found that decision trees have the potential advantage of computational scalability, handling data of mixed types, handling missing values, and dealing with irrelevant inputs [34]. However, decision trees has the limitations of low prediction accuracy and high variance [35]. ME can be regarded as a statistical approach to decision tree modeling where the decisions are treated as hidden multinomial random variables. Therefore, ME has the advantages of decision trees, but improves on them with its soft boundaries, its lower variance, and its probabilistic framework to allow for inference procedures, measures of uncertainty, and Bayesian approaches [36]. On the other hand, decision trees have been combined in ensembles, forming random forests, to increase the performance of a single ensemble and increase prediction accuracy while keeping other decision tree advantages [37]. Similarly, ME has been combined with boosting and, with a gating function that is a function of confidence, ME has been shown to provide an effective dynamic combination for the outputs of the experts [38].

One of the major advantages of ME is that it is flexible enough to be combined with a variety of different models. It has been combined with SVM [12]–[14] to partition the input space and to allocate different kernel functions for different input regions, which would not be possible with a single SVM. Recently, ME has been combined with GPs to make them accommodate nonstationary covariance and noise. A single GP has a fixed covariance matrix, and its solution typically requires the inversion of a large matrix. With the mixture of GP experts model [6], [7], the computational complexity of inverting a large matrix can be replaced with several inversions of smaller matrices, providing the ability to solve larger scale datasets. ME has also been used to make an HMM with time-varying transition probabilities that are conditioned on the input [4], [5].

A significant number of studies have been published on the statistical properties and the training of ME to date. ME has been regarded as a mixture model for estimating conditional probability distributions and, with this interpretation, ME statistical properties have been investigated during the period from 1995 to 2011 (e.g., [39]–[42]). These statistical properties have led to the development of various Bayesian training methods between 1996 and 2010 (e.g., [23], [43]–[45]), and ME has been trained with EM [46], variational learning [47], and MCMC methods [48]. The Bayesian training methods have introduced prior knowledge into the training, helped avoid overtraining, and opened the search for the best model (the number of experts and the depth of the tree) during 1994 to 2007 (e.g., [2], [10], [18], [49]). In the meantime, the model has been used in a very wide range of applications, and has been extended to handle time-series data.

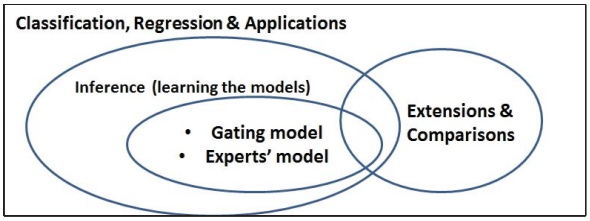


Fig. 1. Outline of the survey.

In this paper, we survey each of the aforementioned areas under three main groups: regression studies, classification studies, and applications. For each one of these groups, we describe how the models have progressed over the years, how they have been extended to cover a wide range of applications, as well as how they compare with other models. An outline is shown in Fig. 1.

Within the regression and classification studies, we group the core of the studies into three main groups; models for the gate, models for the experts, and the inference techniques to learn the parameters of these models. Some of the representative papers can be summarized as follows.

1) Inference.
   a) EM-based methods: IRLS [2], generalized EM [50], Newton–Raphson [51], ECM [52], single-loop EM [53].
   b) Variational [7], [43], [45], [54].
   c) Sampling [6], [18], genetic training [25].
2) Models for the gate.
   a) Gaussian mixture model GMM [7], softmax of GPs [55], Dirichlet distribution [56], Dirichlet process (DP) [18], neural networks (NNs) [12], max/min networks [57], probit function [58].
3) Models for the experts.
   a) Gaussian [2], [59], multinomial [23], [60], generalized Bernoulli [51], GP [55], [61], SVM [12], [14].

II. Fundamentals of ME

In this section, we describe the original ME regression and classification models. In the ME architecture, a set of experts and a gate cooperate with each other to solve a nonlinear supervised learning problem by dividing the input space into a nested set of regions as shown in Fig. 2. The gate makes a soft split of the whole input space, and the experts learn the simple parameterized surfaces in these partitions of the regions. The parameters of these surfaces in both the gate and the experts can be learned using the EM algorithm.

Let $D = \{X, Y\}$ denote the data where $X = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ is the input, $Y = \{\mathbf{y}^{(n)}\}_{n=1}^{N}$ is the target, and N is the number of training points. Also, let $\Theta = \{\Theta_g, \Theta_e\}$ denote the set of all parameters where Θ_g is set of the gate parameters and Θ_e is the set of the expert parameters. Unless necessary, we will

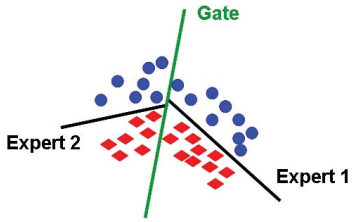


Fig. 2. Simplified classification example for ME. The blue circles and the red diamonds belong to classes 1 and 2, respectively, and they present a nonlinear classification example. The gate makes a soft partition and defines the regions where the individual expert opinions are trustworthy, such that, to the right of the gating line, the first expert is responsible, and to the left of the gating line, the second expert is responsible. With this divide-and-conquer approach, the nonlinear classification problem has been simplified to two linear classification problems. Modified with permission [62].

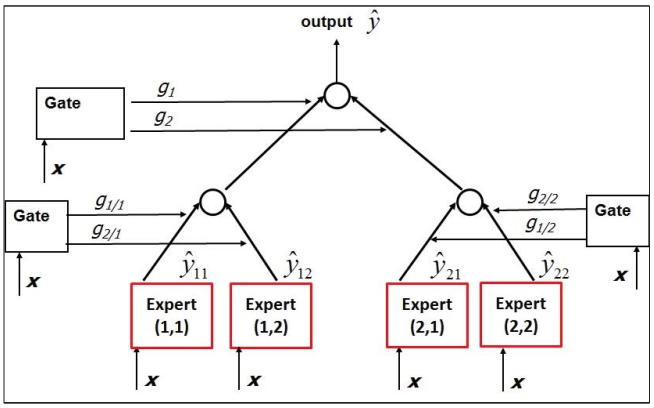


Fig. 3. Two-level HME architecture for regression. In this example, two ME components at the bottom of the figure are combined with a gate at the top to produce the hierarchical ME. Each ME model at the bottom is composed of a single gate and two experts. At the experts, the notation ij is used such that the letter i indexes the branches at the top level and the letter j indexes the branches at the bottom level. For example, $\text{expert}_{1,2}$ corresponds to the first branch at the top level and the second branch at the bottom level. The outputs of the gate are a set of scalar coefficients denoted by $g_{j|i}$. The outputs of the experts, $\hat{\mathbf{y}}_{\mathbf{ij}}$, are weighted by these gating outputs.

denote an input vector $\mathbf{x}^{(n)}$ with $\mathbf{x}$, and a target vector $\mathbf{y}^{(n)}$ with $\mathbf{y}$ from now on. A superscript (n) will be used to indicate that a variable depends on an input $\mathbf{x}^{(n)}$.

Given an input vector $\mathbf{x}$ and a target vector $\mathbf{y}$, the total probability of observing $\mathbf{y}$ can be written in terms of the experts, as

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}, \Theta) &= \sum_{i=1}^{I} P(\mathbf{y}, i|\mathbf{x}, \Theta) \\ &= \sum_{i=1}^{I} P(i|\mathbf{x}, \Theta_g) P(\mathbf{y}|i, \mathbf{x}, \Theta_e) \\ &= \sum_{i=1}^{I} g_i(\mathbf{x}, \Theta_g) P(\mathbf{y}|i, \mathbf{x}\Theta_e) \end{aligned} \tag{1}$$

where I is the number of experts, the function $g_i(\mathbf{x}, \Theta_g) = P(i|\mathbf{x}, \Theta_g)$ represents the gate's rating, i.e., the probability of the ith expert given $\mathbf{x}$, and $P(\mathbf{y}|i, \mathbf{x}$, and $\Theta_e)$ is the probability of the ith expert generating $\mathbf{y}$ given $\mathbf{x}$. The latter will be denoted by $P_i(\mathbf{y})$ from now on.

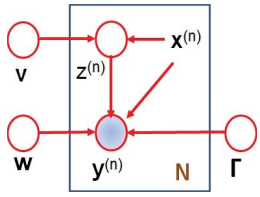


Fig. 4. Graphical representation of the ME regression model. The box denotes a set of N independent identically distributed (i.i.d.) observations where $\mathbf{x}^{(n)}$ is the input, $\mathbf{y}^{(n)}$ is the output, and $z^{(n)}$ is the latent variable. The output node $\mathbf{y}^{(n)}$ is shaded, indicating that these variables are observed [45], [63]. The variables whose size does not change with the size of the dataset are regarded as parameters. The parameters $\mathbf{w}, \Gamma$ of the experts have a direct link to the output since the target vector $\mathbf{y}^{(n)}$ is a function of $\mathbf{w}$ and Γ. The parameter $\mathbf{v}$ of the gate is linked to the output through the indicator variable $z^{(n)}$.

The ME training algorithm maximizes the log-likelihood of the probability in (1) to learn the parameters of the experts and the gate. During the training of ME, the gate and experts get decoupled, so the architecture attains a modular structure. Using this property, the ME model was later extended into a hierarchical mixture of experts (HME) [2], [39], as shown in Fig. 3, for which the probability model is:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{i=1}^{I} g_i(\mathbf{x}, \Theta_{g_i}) \sum_{j=1}^{J_i} g_{j|i}(\mathbf{x}, \Theta_{g_{j|i}}) P_{ij}(\mathbf{y}, \Theta_e) \tag{2}$$

where I is the number of nodes connected to the gate at the top layer, and J_i is the number of nodes connected to the ith lower-level gating network, g_i is the output of the gate in the top layer, $g_{j|i}$ is the output of the jth gate connected to the ith gate of the top layer, and Θ_{g_i} and $\Theta_{g_{j|i}}$ are their parameters, respectively.

For both classification and regression, the gate is defined by the softmax function

$$g_i(\mathbf{x}, \mathbf{v}) = \frac{\exp(\beta_i(\mathbf{x}, \mathbf{v}))}{\sum_{j=1}^{I} \exp(\beta_j(\mathbf{x}, \mathbf{v}))} \tag{3}$$

where the gate parameter $\Theta_g = \mathbf{v}$, and the functions of the gate parameter $\beta_i(\mathbf{x}, \mathbf{v})$ are linear given by $\beta_i(\mathbf{x}, \mathbf{v}) = \mathbf{v}_i^T[\mathbf{x}, 1]$. The softmax function is a smooth version of the winner-take-all model. The experts, on the other hand, have different functions for regression and classification, as explained below.

A. ME Regression Model

Let $\Theta_e = \{\theta_i\}_{i=1}^{I} = \{\mathbf{w}_i, \Gamma_i\}_{i=1}^{I}$ be the parameters of the experts with the graphical model as shown in Fig. 4. In the original ME regression model, the experts follow the Gaussian model:

$$P(\mathbf{y}|\mathbf{x}, \theta_i) = N(\mathbf{y}|\hat{\mathbf{y}}_\mathbf{i}(\mathbf{x}, \mathbf{w}_i), \Gamma_i) \tag{4}$$

where $\mathbf{y} \in R^S$, $\hat{\mathbf{y}}_\mathbf{i}(\mathbf{x}, \mathbf{w}_i)$ is the mean, and Γ_i is the covariance. The vector $\hat{\mathbf{y}}_\mathbf{i}(\mathbf{x}, \mathbf{w}_i)$ is the output of the ith expert, which, in the original ME, was a linear function given by $\hat{\mathbf{y}}_i(\mathbf{x}, \mathbf{w}) = \mathbf{w}_i^T[\mathbf{x}, 1]$.

To make a single prediction, the expectation of (1) is used as the output of the architecture, given by

$$\hat{\mathbf{y}} = \sum_i g_i(\mathbf{x}, \mathbf{v}) \hat{\mathbf{y}}_\mathbf{i}(\mathbf{x}, \mathbf{w}_i). \tag{5}$$

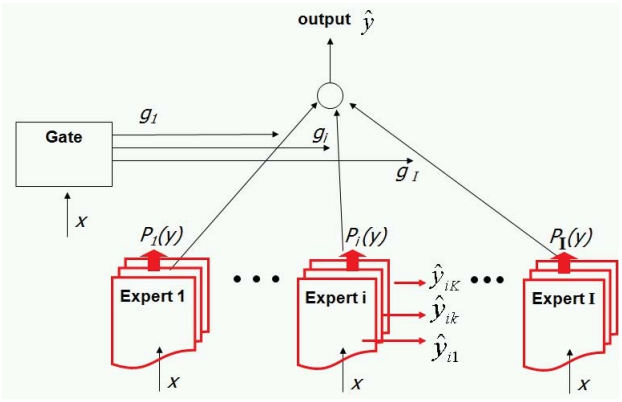


Fig. 5. ME architecture for classification. In the classification model, each expert produces as many outputs as there are classes. These outputs are denoted by $\hat{y}_{ik}$. The multinomial probabilistic outputs of the experts are denoted by P_i.

B. ME Classification Model

The ME architecture for a K-class classification problem is shown in Fig. 5, where K is the number of classes. Different from regression, the desired output $\mathbf{y}$ is of length K and $y_k = 1$ if $\mathbf{x}$ belongs to class k and 0 otherwise. Also, expert i has K parameters $\{\mathbf{w}_{ik}\}_{k=1}^{K}$, corresponding to the parameters of each class. For the ith expert and the kth class, the expert output per class is given by the softmax function

$$\hat{y}_{ik} = \frac{\exp(\mathbf{w}_{ik}^T[\mathbf{x}, 1])}{\sum_{r=1}^{K} \exp(\mathbf{w}_{ir}^T[\mathbf{x}, 1])} \tag{6}$$

which are the means of the experts' multinomial probability models

$$P_i(\mathbf{y}) = \prod_k \hat{y}_{ik}^{y_k}. \tag{7}$$

To make a single prediction, the outputs are computed per class, as

$$\hat{y}_k = \sum_i g_i(\mathbf{x}, \mathbf{v})\hat{y}_{ik}$$

and for practical purposes, the input $\mathbf{x}$ is classified as belonging to the class k that gives the maximum $\hat{y}_k, \quad k : 1, \ldots, K$.

C. Training of ME

The EM [46] algorithm is an iterative method for finding the maximum likelihood (ML) of a probability model in which some random variables are observed and others are hidden. In training the ME, the indicator variables $Z = \{\{z_i^{(n)}\}_{n=1}^{N}\}_{i=1}^{I}$ are introduced to solve the model with the EM algorithm. With the indicator variables, the complete log-likelihood can be written as

$$l(\Theta; D; Z) = \sum_{n=1}^{N}\sum_{i=1}^{I} z_i^{(n)}\{\log g_i^{(n)} + \log P_i(\mathbf{y}^{(n)})\} \tag{8}$$

where $g_i^{(n)} = g_i(\mathbf{x}^{(n)}, \mathbf{v})$ is a shortcut to denote the gate. Equation (8) is a random function of the missing random variables z_i; therefore, the EM algorithm is employed to average out z_i and maximize the expected complete data log-likelihood $E_Z(\log P(D, Z|\Theta))$. The expectation of the log-likelihood in (8) results in

$$\begin{aligned} Q(\Theta, \Theta^{(p)}) &= \sum_{n=1}^{N}\sum_{i=1}^{I} h_i^{(n)}\{\log g_i^{(n)} + \log P_i(\mathbf{y}^{(n)})\} \\ &= \sum_{i=1}^{I}(Q_i^g + Q_i^e) \end{aligned} \tag{9}$$

where p is the iteration index, $h_i^{(n)} = E[z_i^{(n)}|D]$, and

$$Q_i^g = \sum_n h_i^{(n)} \log g_i^{(n)} \tag{10}$$

$$Q_i^e = \sum_n h_i^{(n)} \log P_i(\mathbf{y}^{(n)}). \tag{11}$$

The parameter Θ is estimated by the iterations through the E and M steps given by:

1) E step: Compute $h_i^{(n)}$, the expectation of the indicator variables;
2) M step: Find a new estimate for the parameters, such that $\mathbf{v}_i^{(p+1)} = \underset{\mathbf{v}_i}{\text{argmax}} Q_i^g$, and $\theta_i^{(p+1)} = \underset{\theta_i}{\text{argmax}}\ Q_i^e$.

There are three important points regarding the training. First, (10) describes the cross-entropy between g_i and h_i. In the M step, h_i is held constant, so g_i learns to approximate h_i. Remembering that $0 \le h_i \le 1$ and $0 \le g_i \le 1$, the maximum Q_i^g is reached if both g_i and h_i are 1 and the others $(g_j, h_j, i \ne j)$ are zero. This is in line with the initial assumption from (1) that each pattern belongs to one and only one expert. If the experts actually share a pattern, they pay an entropy price for it. Because of this property, the ME algorithm is also referred to as *competitive learning among the experts*, as the experts are rewarded or penalized for sharing the data. The readers are encouraged to read more about the effect of entropy on the training [2], [64].

The second important point is that, by observing (10) and (11), the gate and the expert parameters are estimated separately owing to the use of the hidden variables. This decoupling gives a modular structure to the ME training, and has led to the development of the HME and to the use of other modular networks at the experts and the gate.

The third important point is that in regression $\max_{\mathbf{w}_i} Q_i^e$ can be attained by solving $\partial Q_i^e/\partial \mathbf{w}_i = 0$ if $\hat{y}_i = \mathbf{w}_i^T[\mathbf{x}, 1]$. However, in general, it cannot be solved analytically when $\hat{y}_i$ is nonlinear. Similarly, it is difficult to find an analytical solution to $\max_{\mathbf{v}_i} Q_i^g$ because of the softmax function. Therefore, one can either use the iterative recursive least squares (IRLS) technique for linear gate and expert models [2], the extended IRLS algorithm for nonlinear gate and experts [39], or the generalized EM algorithm that increases the Q function but does not necessarily fully maximize the likelihood [50]. An alternative solution to overcoming this problem is detailed in Section III-A.

D. Model Selection

Model selection for ME models refers to finding the depth and connections of the tree, which in effect determines the number of experts. Model selection for ME is not much different from model selection for other tree-based algorithms,

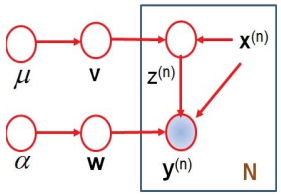


Fig. 6. Graphical representation of the variational ME model for classification. The box denotes a set of N i.i.d. observations. The output node $\mathbf{y}^{(n)}$ is shaded, indicating that these variables are observed. In the variational model, two hyperparameters have been introduced, where μ is the hyperparameter on the gating parameter $\mathbf{v}$, and α is the hyperparameter on the expert parameter $\mathbf{w}$. When compared to the regression model in Fig. 4, the classification model does not have the parameter Γ.

and the main difference is the cost function that evaluates the value of a branch. This cost function varies on the basis of how the optimal structure is searched for. To this end, the studies that aim to arrive at the optimal structure of the tree can be divided into four main categories as follows: growing models [2], [65], [66]; pruning models [49], [67], [68]; exhaustive search [45]; and Bayesian models [10], [18], [54], [69].

Growing models are based on adding more layers to a tree and determining the depth of the tree as well as the number of experts. Pruning models are aimed at reducing the computational requirements. They either keep the model parameters constant but consider the most likely paths as in [49], or prune the least used branches as in [67] and [68]. Exhaustive search refers to doing multiple runs and testing multiple models to find the best model. On the other hand, variational models [54] can simultaneously estimate the parameters and the model structure of an ME model, whereas DP models [18] do not place a bound on the number of experts (which is also referred to as the infinite number of experts). Finally, sparsity promotion studies [10], [69] use sparsity-promoting priors on the parameters to encourage a smaller number of nonzero weights.

Model selection for ME, as for other classes of model, is very difficult, and these studies are important attempts at solving this difficult problem. Unfortunately, to the best of our knowledge, a study to compare all these model selection methods does not exist.

III. Advances in ME for Regression

In the original ME model, maximization of the likelihood with respect to the parameters of the gate is analytically unsolvable due to the nonlinearity of the softmax function. Therefore, within the EM iterations, there is an inner loop of iterations to update the parameters using the IRLS algorithm. To avoid these inner loop iterations, Xu *et al.* [50] proposed making the gate analytically solvable and introduced an alternative gating function, which will be explained in Section III-A. Recently, this alternative ME model has been used in the training of mixture of GP experts, as explained in Section III-B.

A. Alternative Model for ME

The alternative ME model uses Gaussian parametric forms in the gate given by

$$g_i(\mathbf{x}, \mathbf{v}) = \frac{a_i P(\mathbf{x}|\mathbf{v}_i)}{\sum_j a_j P(\mathbf{x}|\mathbf{v}_j)}, \quad \sum_i a_i = 1, \quad a_i \geq 0 \tag{12}$$

where $P(\mathbf{x}|\mathbf{v}_i)$ are density functions from the exponential family such as the Gaussian. In addition, to make the maximization with respect to the gate analytically solvable with this new form, Xu *et al.* [50] proposed working on the joint density $P(\mathbf{y}, \mathbf{x}|\Theta)$ instead of the likelihood $P(\mathbf{y}|\mathbf{x}, \Theta)$. Assuming

$$P(\mathbf{x}) = \sum_j a_j P(\mathbf{x}|\mathbf{v}_j) \tag{13}$$

the joint density is given as

$$P(\mathbf{y}, \mathbf{x}|\Theta) = P(\mathbf{y}|\mathbf{x}, \Theta) P(\mathbf{x}) \tag{14}$$

$$= \sum_i a_i P(\mathbf{x}|\mathbf{v}_i) P(\mathbf{y}|i, \mathbf{x}, \theta_e). \tag{15}$$

Comparing this new parametric form to the original ME model, the M step requires finding three sets of parameters $\mathbf{a}$, $\mathbf{v}$, and θ, as opposed to the two sets of parameters $\mathbf{v}$ and θ in the original model. This alternative model, also referred to as the localized ME in the literature, does not require selecting a learning step-size parameter and leads to faster convergence, as the maximization with respect to the gate is solvable analytically. Following up on this paper, Fritsch *et al.* [70] used it for speech recognition, and Ramamurti and Ghosh [60] added RBF kernels to the input of the HME as preprocessors. Also, the alternative (localized) ME model has been used in the mixture of GP experts.

B. Mixture of GP Experts

GPs are powerful nonparametric models that can provide error terms for each data point. Recently, ME models have been combined with GPs to overcome the limitation of the latter, i.e., to make them more flexible by accommodating nonstationary covariance and noise levels, and to decrease their computational complexity. A GP is a generalization of the Gaussian distribution, specified by a *mean* function and a *covariance* function. It is defined by

$$\mathbf{y}^{(n)} = f(\mathbf{x}^{(n)}) + \epsilon_n \tag{16}$$

where $f(\mathbf{x}^{(n)})$ is a nonlinear function of $\mathbf{x}^{(n)}$, and $\epsilon_n \sim N(0, \sigma^2)$ is an error term on a data point [71]. The prior for the function f is assumed to be a GP, i.e., for each n, $f(\mathbf{x}^{(n)})$ has a multivariate normal distribution with zero mean and a covariance function $C(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$. The covariance function is a positive-definite kernel function such as the Gaussian kernel

$$C(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = A \exp\left(-\frac{||\mathbf{x}^{(n)} - \mathbf{x}^{(m)}||^2}{2s^2}\right) \tag{17}$$

with scale parameter s and amplitude A. Therefore, Y has a normal distribution with zero mean and covariance

$$\Psi_{A,s} = C_{A,s} + \sigma^2 I \tag{18}$$

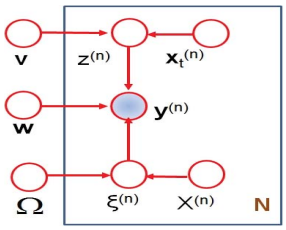


Fig. 7. Graphical representation of the modified HME model. Here, Ω is the parameter of the top gate called the S-gate that operates on an observation sequence $\mathbf{X^{(n)}}$. $\xi^{(n)}$ are the indicator variables that result from the S-gate, and they indicate the labels that specify $\mathbf{x}_t^{(n)}$ in $\mathbf{X^{(n)}}$. The rest of the model is similar to ME.

where I is an identity matrix, and $C_{A,s}$ is the $N \times N$ matrix with elements as defined in (17).

With this distribution, the log-likelihood of the training data is

$$L_{A,s} = -\frac{1}{2}\log|\Psi_{A,s}| - \frac{1}{2}Y^T {\Psi_{A,s}}^{-1} Y - \frac{N}{2}\log 2\pi. \quad (19)$$

From (19), the ML estimate of the parameters A and s can be computed with iterative optimization methods, which require the evaluation of ${\Psi_{A,s}}^{-1}$ and take time $O(N^3)$ [71].

There are two important limitations in this GP formulation. The first limitation is that the inference requires the inversion of the Gram matrix, which is very difficult for large training datasets. The second limitation is the assumption that the scale parameter s in (17) is stationary. A stationary covariance function limits the model flexibility if the noise is input-dependent and if the noise variance is different in different parts of the input space. For such data, one wishes to use smaller scale parameters in regions with high data density and larger scale parameters for regions that have little data. Recently, mixtures of GP experts have been introduced as a solution to these two limitations.

In 2001, Tresp [55] proposed a model in which a set of GP regression models with different scale parameters is used. This mixture of GP experts model can autonomously decide which GP regression model is best for a particular region of the input space. In this model, three sets of GPs need to be estimated: one set to model the mean of the experts; one set to model the input-dependent noise variance of the GP regression models; and a third set to learn the parameters for the gate. With this assumption, the following mixture of GP experts model is defined:

$$P(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{I} P(z=i|\mathbf{x}) G(y; f_i^{\mu}(\mathbf{x}), \exp(2f_i^{\sigma}(\mathbf{x}))) \quad (20)$$

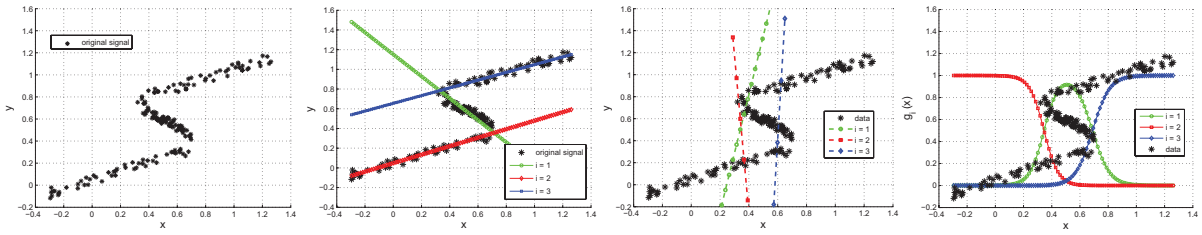
where G represents the GP experts and is also the Gaussian density with mean $f_i^{\mu}(\mathbf{x})$ and variance $\exp(2f_i^{\sigma}(\mathbf{x}))$. Here, f_i^{μ} is the GP that models the mean μ for expert i, and f_i^{σ} is the GP that models the variance. In addition, $P(z=i|\mathbf{x})$ is the gate given by

$$P(z=i|\mathbf{x}) = \frac{\exp(f_i^z(\mathbf{x}))}{\sum_{j=1}^{I}\exp(f_j^z(\mathbf{x}))} \quad (21)$$

where z is the discrete I-state indicator variable that determines which of the GP models (i.e., experts) is active for a given input $\mathbf{x}$. Just like the ME model, one can maximize the log posterior of (20) using EM updates. In Tresp's study, where the experts are GPs and the gate is a softmax of GPs, the mixture of GPs was shown to divide up complex tasks into subtasks and perform better than any individual model. For I experts, the model requires one to compute $3I$ GPs (for the mean and variance of the experts, and the gate) each of which requires computations over the entire dataset. In addition, the model requires one to specify the hyperparameters and the number of experts.

In 2002, Rasmussen and Ghahramani [18] argued that the independent identically distributed (i.i.d.) assumption in the traditional ME model is contrary to GP models that model the dependencies in the joint distribution. Therefore, as opposed to computing the expectations of the indicator variables, they suggested obtaining the indicator variables from Gibbs sampling. In addition, the gate was modified to be an input-dependent DP, and the hyperparameters of the DP controlled the prior probability of assigning a data point to a new expert. Unlike Tresp's work where the hyperparameters were fixed, the hyperparameters of the DP prior were inferred from the data. In doing so, instead of trying to find the number of experts or specifying a number, Rasmussen and Ghahramani assumed an infinite number of experts, most of which contributed only a small mass to the distribution. In addition, instead of using all the training data for all the experts, the experts were trained with only the data that was assigned to them. Thus, the problem was decomposed into smaller matrix inversions at the GP experts, achieving a significant reduction in the number of computations. In 2006, Meeds and Osindero [6] proposed learning the infinite mixture of Gaussian ME using the alternative ME model, which was described in Section III-A. This generative model has the advantage of dealing with partially specified data and providing inverse functional mappings. Meeds *et al*. also used clustering in the input space and trained their experts on this data; however, as they themselves pointed out, a potentially undesirable aspect of the strong clustering in input space is that it could lead to inferring several experts even if a single expert could do a good job of modeling the data.

The previous inference algorithms in [6] and [18] were based on Gibbs sampling, which can be very slow. To make the learning faster, Yuan and Neubauer [7] proposed using variational learning based on the alternative ME model. In this variational mixture of GP experts (VMGPE) study, the experts were still GPs, but were reformulated by a linear model. The linear representation of the GPs helped break the dependency of the outputs and the input variables, and made variational learning feasible. Similar to [6] and [18], the gate followed a Dirichlet distribution; but unlike those studies, in which the input could only have one Gaussian distribution, VMGPE models the inputs as GMM. With this structure in place, variational inference is employed to find the model parameters. Finally, in a recent study by Yang and Ma [61], an efficient EM algorithm was proposed that is based on the leave-one-out cross-validation probability decomposition. With this solution, the expectations of assignment variables can be solved directly in the E step.



(a) (b) (c) (d)

Fig. 8. Simple regression example. (a) Data to be approximated. (b) Three linear experts. (c) Linear parameters of the gate. (d) Corresponding softmax outputs of the gate. The division of the input space can be understood by looking at the regions and then the corresponding expert. For example, the third gating parameter in (d) is responsible of the right side of the space. In this part of the region, the blue line in (b) is effective.

To summarize, there are two aspects to be considered in the mixture of GPs: 1) the type of the gate and 2) the inference method. The gate can be a softmax of GPs [55], or it can follow a Dirichlet distribution [56], a DP [18], or be a Gaussian mixture model [7]. The second aspect is the inference method such as the EM [61], sampling [6], [18], and variational [7]. With all these model details considered, the main advantages of using a mixture of GP experts can be summarized as follows: 1) it helps accommodate nonstationary covariance and noise levels; 2) the computational complexity of inverting an $N \times N$ matrix is replaced by several inversions of smaller matrices leading to speedup and the possibility of solving larger scale datasets; and 3) the number of experts can be determined in the case of DP priors.

IV. Advances in ME for Classification

The ME model was designed mainly for function approximation rather than classification; however, it has a significant appeal for multiclass classification due to the idea of training the gate together with the individual classifiers through one protocol. In fact, more recently, the ME model has been getting attention as a means of finding subclusters within the data, and learning experts for each of these subclusters. In doing so, the ME model can benefit from the existence of common characteristics among data of different classes. This is an advantage compared to other classifiers that do not consider the other classes when finding the class conditional density estimates from the data of each class.

The ME model for classification was discussed in Section II-B. Waterhouse and Robinson provided a nice overview on parameter initialization, learning rates, and stability issues in [72] for multiclass classification. Since then, there have been reports in the literature [51], [60], [73] that networks trained by the IRLS algorithm perform poorly in multiclass classification. Although these arguments have merit, it has also been shown that, if the step-size parameters are small enough, then the log-likelihood is monotone increasing and IRLS is stable [74]. In this section, we go over the arguments against multiclass classification, and review the solutions provided for it.

The IRLS algorithm makes a batch update, modifying all the parameter vectors of an expert $\{\mathbf{w}_{ik}\}_{k=1}^{K}$ at once, and implicitly assumes that these parameters are independent. Chen *et al*. [51] pointed out that IRLS updates result in an incomplete Hessian matrix, in which the off-diagonal elements are nonzero, implying the dependence of the parameters. In fact, in multiclass classification, each parameter vector in an expert relates to all the others through the softmax function in (6), and therefore, these parameter vectors cannot be updated independently. Chen *et al*. noted that such updates result in unstable log-likelihoods; so they suggested using the Newton–Raphson algorithm and computing the exact Hessian matrix in the inner loop of the EM algorithm. However, the use of the exact Hessian matrix results in expensive computations, and therefore they proposed using the generalized Bernoulli density in the experts for multiclass classification as an approximation to the multinomial density. With this approximation, all of the off-diagonal block matrices in the Hessian matrix are zero matrices, and the parameter vectors are separable. This approximation results in simplified Newton–Raphson updates and requires less time; however, the error rates increase because of the fact that it is an approximation.

Following this paper, Ng and McLachlan [74] ran several experiments to show that the convergence of the IRLS algorithm is stable if the learning rate is kept small enough, and the log-likelihood is monotone increasing even though the assumption of independence is incorrect. However, they also suggested using the expectation-conditional maximization (ECM) algorithm with which the parameter vectors can be estimated separately. The ECM algorithm basically learns the parameter vectors one by one, and uses the updated parameters while learning the next parameter vector. In doing so, the maximizations are over smaller dimensional parameter spaces and are simpler than a full maximization, and the convergence property of the EM algorithm is maintained. In 2007, Ng and McLachlan [52] presented an ME model for binary classification in which the interdependency between the hierarchical data was taken into account by incorporating a random effects term into the experts and the gate. The random effects term in an expert provided information as to whether there was a significant difference in local outputs from each expert, which was shown to increase the classification rates.

More recently, Yang and Ma [53] introduced an elegant least mean squares solution to directly update the parameters of a gate with linear weights. This solution eliminates the need for the inner loop of iterations, and has been shown to be faster and more accurate on a number of synthetic and real datasets.

In the aforementioned studies [51]–[53], [72], [74], the focus was on the training of the gate. In another batch of studies, the focus was clustering of the data, and ME

was found useful in classification studies that could benefit from the existence of subclusters within the data. In a study by Titsias and Likas [75], a three-level hierarchical mixture model for classification was presented. This model assumes that: 1) data are generated by I sources (clusters or experts) and 2) there are subclusters (class-labeled sources) within each cluster. These assumptions lead to the following log-likelihood:

$$l(\Theta; D; Z) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in X_k} \sum_{i=1}^{I} \log\{P(i) P(k|i) p(\mathbf{x}|k, i, \theta_{ki})\} \tag{22}$$

where X_k denotes all the data with class label k, $p(\mathbf{x}|k, i, \theta_{ki})$ is the Gaussian model of a subcluster of class k, and θ_{ki} is its corresponding parameter. With this formulation, the classical ME was written more explicitly, separating the probability of selecting an expert and the probability of selecting the subcluster of a class within an expert.

In a 2008 study by Xing and Hu [15], unsupervised clustering was used to initialize the ME model. In the first stage, a fuzzy C-means [76] based algorithm was used to separate all the unlabeled data into several clusters and a small fraction of these samples from the cluster centers were chosen as training data. In the second stage, several parallel two-class MEs were trained with the corresponding two-class training datasets. Finally, the class label of a test datum was determined by plurality vote of the MEs.

In using clustering approaches for the initialization of ME, a good cluster can be a good initialization to the gate and speed up the training significantly. On the other hand, strong clustering may lead to an unnecessary number of experts, or lead to overtraining. It might also force the ME to a local optimum that would be hard to escape. Therefore, it would be interesting to see the effect of initialization with clustering on the ME model. Another work that would be interesting would be to see the performance of ME for a K-class problem where $K > 2$ and compare it to the $\binom{K}{2}$ comparisons of two-class MEs and the decision from their popular vote.

V. Bayesian ME

HME model parameters are traditionally learned using ML estimation, for which there is an EM algorithm. However, the ML approach typically leads to overfitting, especially if the number of data points in the training set is low compared to the number of parameters in the model. Also, because the HME model is based on the divide-and-conquer approach, the experts' effective training sets are relatively small, increasing the likelihood of introducing bias into solutions as a result of low variance. Moreover, ML does not provide a way to determine the number of experts/gates in the HME tree, as it always prefers more complex models. To solve these problems, two Bayesian approaches have been introduced based on: 1) variational learning and 2) maximum *a posteriori* solution. These solutions are also not trivial because the softmax function at the gate does not admit conjugate priors. The sophisticated approximations needed to arrive at these solutions will be explained in this section.

A. Variational Learning of ME

Variational methods, also called ensemble methods, variational Bayes, or variational free energy minimization, are techniques for approximating a complicated posterior probability distribution P by a simpler ensemble Q. The key to this approach is that, as opposed to the traditional approaches where the parameter Θ is optimized to find the mode of the distribution, variational methods define an approximating probability distribution over the parameters $Q(\Theta, \Phi)$, and optimize this distribution by varying Φ so that it approximates the posterior distribution well [47]. Hence, instead of point estimates for Θ representing the mode of a distribution in the ML learning, variational methods produce estimates for the distribution of the parameters.

Variational learning can be summarized with three main steps. In the first step, we take advantage of the Bayesian methods, i.e., we assume prior distributions on the parameters and write the joint distribution. In the second step, we assume a factorizable distribution Q; and in the third step, we solve the variational learning equations to find the Q distribution that would best estimate the posterior distribution.

The earliest studies on the variational treatment of HME were by Waterhouse *et al*. [43], where they assumed Gaussian priors on the parameters of the experts and the gate given by

$$P(\mathbf{w}_{ik}|\alpha_{ik}) = N\left(\mathbf{w}_{ik}|0, \alpha_{ik}^{-1}\mathbf{I}\right)$$

and

$$P\left(\mathbf{v}_i|\mu_i\right) = N\left(\mathbf{v}_i|0, \mu_i^{-1}\mathbf{I}\right)$$

and Gamma priors on these Gaussian parameters given by

$$P(\alpha_{ik}) = \text{Gam}(\alpha_{ik}|a_0, b_0)$$

and

$$P(\mu_i) = \text{Gam}(\mu_i|c_0, d_0)$$

where a_0, b_0, c_0, and d_0 are the hyper-hyperparameters, where a_0, c_0 control the shape and b_0, d_0 control the scale in a Gamma distribution. The zero mean Gaussian priors on the parameters $\mathbf{w}_{ik}$ and $\mathbf{v}_i$ correspond to the weight decay in NNs. The hyperparameters α_{ik} and μ_i are the precisions (inverse variances) of the Gaussian distributions; so large hyperparameter values correspond to small variances, which constrain the parameters to be close to 0 (for the zero-mean Gaussian priors) [77]. The graphical representation of the parameters is given in Fig. 6.

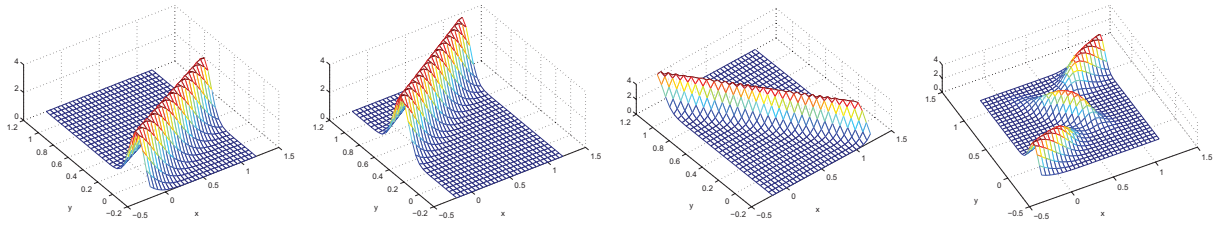
Denoting all the hyperparameters with Φ, and using these distributions of the hyperparameters, the joint distribution can be written as

$$P(\Theta, \Phi, Z, D) = P(Y, Z|\mathbf{w}, \mathbf{v}) P(\mathbf{w}|\alpha) P(\alpha) P(\mathbf{v}|\mu) P(\mu) \tag{23}$$

which is the first step in variational learning, as mentioned previously.

In the second step, the approximating distribution Q is assumed to factorize over the partition of the variables as

$$Q(\Theta, \Phi, Z) = Q(Z) \prod_i Q(\mathbf{v}_i) Q(\mu_i) \prod_k Q(\mathbf{w}_{ik}) Q(\alpha_{ik}). \tag{24}$$



(a) (b) (c) (d)

Fig. 9. Mesh of the input space covered by the experts showing the soft partitioning of the input space and the total probabilistic output for the data shown in Fig. 8. (a)–(c) Gaussian surfaces corresponding to the three experts. These are the probabilistic outputs P_i of expert i, where $i = \{1, 2, 3\}$. (d) Final result $\sum_i g_i P_i$.

Finally, in the third step, the goal is to find this factorizable distribution Q that best approximates the posterior distribution. Hence, the evidence $P(D)$ is decomposed using

$$\log P(D) = L(Q) + KL(Q||P) \tag{25}$$

where L is the lower bound

$$L(Q) = \int Q(\Theta, \Phi, Z) \log \frac{P(\Theta, \Phi, Z, D)}{Q(\Theta, \Phi, Z)} d\Theta d\Phi dZ \tag{26}$$

and KL is the Kullback–Leibler divergence defined as

$$KL(Q||P) = -\int Q(\Theta, \Phi, Z) \log \frac{P(\Theta, \Phi, Z|D)}{Q(\Theta, \Phi, Z)} d\Theta d\Phi dZ. \tag{27}$$

The Q distribution that best approximates the posterior distribution minimizes the KL divergence; however, working on the KL divergence would be intractable, so we look for the Q that maximizes the lower bound L instead [63]. Plugging the joint distribution (23) and the Q distribution (24) into the lower bound equation (26), one then computes each factorized distribution (24) of the Q distribution by taking the expectations with respect to all the other variables [78].

In maximizing the lower bound for the distribution of the gate parameter, Waterhouse *et al.* used a Laplacian approximation to compute the expectation that involved the softmax function. However, this trick introduces some challenges. Because of the Laplace approximation, the lower bound cannot get tight enough, so the full advantage of the ensemble learning methods are hard to obtain. Also, when predicting the distribution of the missing data, one must integrate the product of a Gaussian and the log of a sigmoid, requiring yet another approximation. The practical solution is to evaluate the sigmoid at the mean of the Gaussian [77].

One of the major advantages of variational learning is that it finds the *distributions* of the latent variables in the E step and the *distributions* of the parameters and the hyperparameters of the gate and the experts in the M step. In addition, the benefit of using a variational solution can be seen by the effect of the hyperparameters that appear as a regularizer in the Hessian. In fact, if one constrains the hyperparameters to be 0 and uses the delta function for Q, the EM algorithm is obtained. It has been shown that the variational approach avoids overfitting and outperforms the ML solution for regression [43] and for classification [23].

In comparison to ML learning, which prefers more complex models, Bayesian learning makes a compromise between model structure and data-fitting and hence makes it possible to learn the structure of the HME tree. Therefore, in another variational study, Ueda and Ghahramani [44], [54] provided an algorithm to simultaneously estimate the parameters and the model structure of an MEs based on the variational framework. To accomplish this goal, the number of experts was treated as a random variable, and a prior distribution $P(I)$ on the number of experts was included in the joint distribution. Hence, with M representing the maximum number of experts, and I representing the number of experts where $I = 1, \ldots, M$, the joint distribution was modified as

$$\begin{aligned} P(\Theta, \Phi, Z, D) = {} & P(Y, Z|\mathbf{w}, \mathbf{v}, I) P(\mathbf{w}|\alpha, I) \\ & P(\alpha|I) P(\mathbf{v}|\mu, I) P(\mu|I) P(I). \end{aligned} \tag{28}$$

To maximize $L(Q)$, first the optimal posteriors over the parameters for each I were found, and then they were used to find the optimal posterior over the model. This paper provided the first method for learning both the parameters and the model structure of an ME in a Bayesian way; however, it required optimization for every possible number of experts, requiring significant computation.

Therefore, in 2003, Bishop and Svensen [45] presented another Bayesian HME where they considered only binary trees. With this binary structure, the softmax function of the gate was modified to be

$$P(z_i|\mathbf{x}, \mathbf{v}_i) = \sigma(\mathbf{v}_i^T \mathbf{x})^{z_i} [1 - \sigma(\mathbf{v}_i^T \mathbf{x})]^{1-z_i} \tag{29}$$

$$= \exp(z_i \mathbf{v}_i^T \mathbf{x})^{z_i} \sigma(-\mathbf{v}_i^T \mathbf{x}) \tag{30}$$

where $\sigma(a) = (1/1 + \exp(-a))$ is the logistic sigmoid function and $z_i \in \{0, 1\}$ is a binary variable indicating the left and right branches. With this new representation, Bishop and Svensen wrote a variational lower bound for the logistic sigmoid in terms of an exponential function multiplied by a logistic sigmoid. The lower bound that is obtained from this approximation gives a tighter bound because of the fact that the logistic sigmoid function and its lower bound attain exactly the same values with an appropriate choice of the variational parameters. However, this model only admits binary trees, and assumes that a deep-enough tree would be able to divide the input space. Hence, to find the best model, they exhaustively search all possible trees with multiple runs and multiple initializations. Bishop and Svensen's algorithm was later used by Mossavat *et al.* [28] to estimate speech quality, and was found to work better than the P.563 standard and the Bayesian MARS (multivariate adaptive regression splines) algorithm.

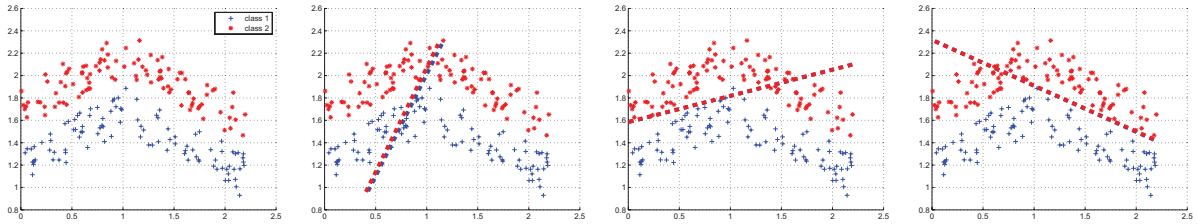


Fig. 10. Classification results on a simple example. (a) Blue and the red points belong to classes 1 and 2, respectively. The gate divides the region in two in (b), such that, to the left of the gating line the first expert is responsible and to the right of the gating line the second expert is responsible. In (c), expert 1 divides the red and blue points that are to the left of the gate. In (d), the second expert divides the red and the blue points that are to the right of the gate.

One problem associated with these variational techniques is that the variational bound will have many local maxima. Therefore, a good variational solution requires multiple runs and random initializations, which can be computationally costly. Another solution is to use sampling tools to approach a global solution; however, these are also known to increase computational complexity.

B. *Maximum* a Posteriori *Learning of ME*

In 2006, Kanaujia and Metaxas [69] used the maximum *a posteriori* approach to compute the Bayesian MEs. The quadratic weight decay term of prior distributions is very similar to the diagonalization term that appears in the variational approach, and it is reflected in the log-posterior as a regularization parameter. The estimated hyperparameters were used to prune out weights and to generate sparse models at every EM iteration. In 2007, Sminchisescu *et al*. [9], [10] used this MAP learning with sparse priors to estimate 3-D human motion in video sequences. In 2008, Bo *et al*. [16] extended this paper for training sparse conditional Bayesian mixtures of experts with high-dimensional inputs.

Recently, other ways of training the ME model have been proposed. Versace *et al*. [25] proposed using genetic training instead of gradient descent. Lu [17] introduced a regularization term to the cross-entropy term in ML training of an ME.

VI. Statistical Properties of Mixture of Experts

Formal statistical justification of ME has been a more recent development. EM training was shown to converge linearly to a local solution by Jordan *et al*. [39]. Jacobs [79] analyzed the bias and variance of ME architectures and showed that ME produces biased experts whose estimates are negatively correlated. Zeevi *et al*. [80] established upper bounds on the approximation error, and demonstrated that by increasing the number of experts, one-layer mixtures of linear model experts can approximate a class of smooth functions. They also showed that, by increasing the sample size, the least-squares method can be used to estimate the mean response consistently. Later, Jiang and Tanner [40] generalized these results using HME and the ML method, and showed that the HME mean functions can approximate the true mean function at a rate of $O(I^{-2/d})$ in the L_p norm, where I is the number of experts and d is the dimension of the input. In 2000, Jiang [81] proved that the Vapnik–Chervonenkis (VC) dimension of ME is bounded below by I and above by $O(I^4d^2)$. The VC dimension provides a bound on the rate of uniform convergence of empirical risk to actual risk [82], [83], and is used for planning the number of training samples and for estimating computational efficiency [84]. Jiang and Tanner [85] also provided regularity conditions on the gate and on the experts under which the ML method in the large sample limit produces a consistent and asymptotically normal estimator of the mean response. Under these regularity conditions, they showed that the ML estimators are consistent and asymptotically normal. In addition, they showed that ME is identifiable [86] if the experts are ordered and the gate is initialized. For a statistical model to support inference, it must be identifiable, that is, it must be theoretically possible to learn the true value of this model's underlying parameter after obtaining an infinite number of observations from it [87]. Jiang and Tanner [88] also showed that HME is capable of approximating any function in a Sobolev space, and that HME probability density functions can approximate the data generating density at a rate of $O(I^{-4/d})$ in KL divergence. Following these results, Carvalho and Tanner [41] presented a formal treatment of conditions to guarantee the asymptotic normality of the ML estimator under stationarity and nonstationarity. More recently, Yang and Ma [42] investigated the asymptotic convergence properties of the EM algorithm for ME. Ge and Jiang [89], [90] showed the consistency properties of Bayesian inference using mixtures of logistic regression models, and they gave conditions on choosing the number of experts so that Bayesian inference is consistent in approximating the underlying true relationship between $\mathbf{y}$ and $\mathbf{x}$. These statistical justifications have gone hand in hand with the development of the Bayesian ME models, which were described in the previous section.

VII. Modifications to ME to Handle Sequential Data

The original formulation of ME was for static data, and was based on the statistical independence assumption of the training pairs. Hence, it did not have a formulation to handle causal dependencies. To overcome this problem, several studies extended ME for time-series data. In the past decade, ME has been applied to regression of time-series data in a variety of applications that require time-series modeling. ME was found to be a good fit in such applications where the time-series data is nonstationary, meaning the time series

switch their dynamics in different regions of the input space, and it is difficult for a single model to capture the entire dynamics of the data. For example, Weigend *et al.* [64] used ME to predict the daily electricity demand of France, which switches among regimes depending on the weather, season, holidays, and workdays, establishing daily, seasonal, and yearly patterns. Similarly, Lu [17] used ME for climate prediction because the time-series dynamics switch seasonally. For such data, ME showed success in finding both the decomposition and the piecewise solution in parallel. In most of the following papers, the main idea is that the gate divides the data into regimes, and the experts learn local models for each regime. In the rest of this section, we will refer to the original ME as the static model, and the extensions as the time-series ME models.

For time-series data, Zeevi *et al.* [91] and Wong *et al.* [92] generalized the autoregressive models for time-series data by combining them with an ME structure. In the speech-processing community, Chen *et al.* [73], [93], [94] used HME for text-dependent speaker identification. In [73], the features were calculated from a window of utterances to introduce temporal information into the solution. In [94], a modified HME structure was introduced. In the modified HME, a new gate was added to make use of the transitional information while the original HME architecture dealt with instantaneous information. The new gate, called the S-gate, is placed at the top of the tree and calculates a weighted average of the output of the HME model. Thus, for a given observation sequence $\mathbf{X}^{(n)} = \{\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_t^{(n)}, \ldots, \mathbf{x}_T^{(n)}\}$, where $\mathbf{x}_t^{(n)}$ is the input at time t, and a given static ME model $P(\mathbf{y}^{(n)}|\mathbf{x}_t^{(n)})$, the probabilistic model in (2) was modified for time-series ME as

$$P\left(\mathbf{y}^{(n)}|\mathbf{X}^{(\mathbf{n})}, \Theta\right) = \sum_{t=1}^{T} \lambda_X\left(\mathbf{x}_t^{(n)}; \Omega\right) P\left(\mathbf{y}^{(n)}|\mathbf{x}_t^{(n)}\right) \tag{31}$$

with

$$\lambda_X(\mathbf{x}_t^{(n)}; \Omega) = \frac{P\left(\mathbf{x}_t^{(n)}|\Omega\right)}{\sum_{s=1}^{T} P\left(\mathbf{x}_s^{(n)}|\Omega\right)} \tag{32}$$

where $P(\mathbf{x}_t^{(n)}|\Omega)$ is a Gaussian distribution and Ω is the parameters of a Gaussian distribution as shown in Fig. 7. Then, for a speaker identification system of population K, they selected the unknown speaker k^* that gives the highest regression probability out of the K models. Here, the traditional ME works on $\mathbf{x}_t^{(n)}$, and the extra gate includes the computations for all t. Using this model, Chen *et al.* [94] modified the EM update equations and solved for the parameters of the topmost Gaussian gate analytically, whereas the rest of the parameters for the experts and the gate were found iteratively. With this paper, HME gained the capability to handle sequences of observations, but the experts and the gate (except the extra gate) were still linear models.

For nonstationary data, Cacciatore and Nowlan [95] suggested using recurrence in the gate, setting one input of the gate to the ratio of the outputs from two preceding time steps. Weigend *et al.* [64] developed a gated ME to handle time-series data that switches regimes. A gate in the form of a multilayer perceptron combines the outputs of the neural network experts. Hence, while the gate discovers the hidden regimes, the experts learn to predict the next observed value. This gated ME was extended by Coelho *et al.* [3], where training was accomplished using genetic algorithms instead of gradient descent. A similar idea to detect switching regimes was also visited by Liehr *et al.* [96], where the gate was a transition matrix, and the experts were Gaussian functions.

Most of these ME models for time-series regression use a one-step-ahead or multistep-ahead prediction, in which the last d values of the time-series data are used as a feature of d dimensions in a neural network. The benefit of using such sliding-window techniques is that a sequential supervised learning problem can be converted into a classical supervised learning problem. However, these algorithms cannot handle data with varying length, and the use of multilayer network approaches prevents them from completely describing the temporal properties of time-series data. Such problems were discussed by Dietterich in [97].

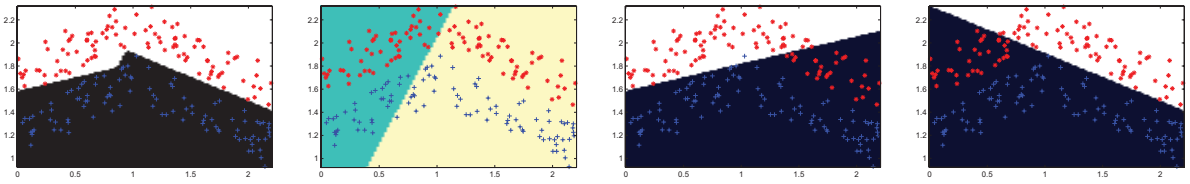
To remove the i.i.d. assumption of the data that was necessary in the original HME model, and to find a model appropriate for time-series data, Jordan *et al.* [98] described a decision tree with Markov temporal structure referred to as a hidden Markov decision tree, in which each decision in the tree is dependent on the decision taken at the node at the previous step. The result was an HMM in which the state at each moment in time was factorized, and the factors were coupled to form a decision tree. This model was an effort to combine adaptive graphical probabilistic models such as the HMM, HME, input-output HMM [99], and factorial HMM [100] in a variational study.

Other extensions of ME to time-series data include studies where the experts are the states of an HMM [4], [5], studies that mimic the probability density estimation of an HMM using ME systems [70], [99], [101]–[103], and studies on ME with recurrent neural nets associated with the states of an HMM [104], [105].

VIII. Comparison to Popular Algorithms

HME can be regarded as a statistical approach to decision tree modeling where the decisions are treated as hidden multinomial random variables. Therefore, in comparison to decision tree methods such as CART [106], HME uses soft boundaries, and allows us to develop inference procedures, measures of uncertainty, and Bayesian approaches. Also, Haykin [36] pointed out that an HME can recover from a poor decision somewhere further up the tree, whereas a standard decision tree suffers from a greediness problem, and gets stuck once a decision is made.

HME bears a resemblance to the boosting algorithm [107], [108] in that weak classifiers are combined to create a single strong classifier. ME finds the subsets of patterns that naturally exist within the data, and learns these easier subsets to solve the bigger problem. In boosting, on the other hand, each classifier becomes an expert on difficult patterns on which other classifiers make an error. Hence, the mixture coefficient in boosting depends on the classification error and provides a



(a) (b) (c) (d)

Fig. 11. Classification results are shown as areas for the data shown in Fig. 10. The final classification is shown in (a), where the dark region corresponds to class 1 (blue points) and the white region corresponds to class 2 (red points). The gate in (b) divides the area into two. On the dark region that has been marked by the gate, expert 1 is active and makes a correct decision in separating the red and the blue point at this region in (c). Similarly, expert 2 makes a successful classification between the red and the blue points at the yellow region that has been selected by the gate in (d).

linear combination, whereas the mixture coefficient (the gate) in the HME depends on the input and makes probabilistic combination of experts. This difference in mixing makes the training of these two algorithms significantly different. In boosting, the classifiers are trained sequentially on the basis of data that was filtered by the previously trained networks. Once the training data is specified, the boosting networks are learned independently, and combined with a mixture coefficient that is learned from the classification error. In HME, the experts compete with each other for the right to learn particular patterns; hence, all the experts are updated at each iteration depending on the gate's selection. In an effort to incorporate boosting into ME, Waterhouse and Cook [109] initialized a split of the training set learned from boosting to different experts. The benefits of ensembles (such as boosting and ME) in improving the bias/variance problem were discussed by Shimshoni and Intrator [110]. In another boosted ME study, Avnimelech and Intrator [38] added the experts one by one, and trained the new experts with the data on which the other experts were not confident. The gating function in this case was a function of the confidence. From the boosting perspective, boosted ME provides a dynamic combination model for the outputs of the networks. From the ME perspective, one can think of it as a smart preprocessing of the data. Another study on preprocessing was published by Tang *et al*. [33] where self-organizing maps (SOMs) were used, in which, as opposed to feeding all the data into the experts, local regions of the input space found by the SOM were assigned to individual experts.

MARS partitions the input space into overlapping regions and fits a univariate spline to the training data in each region. The same mixture coefficient argument also applies to the MARS model [111], which has the equational form of a sum of weighted splines. In comparison to the latent variables of HME, MARS defines the states by the proximity of the observed variables. On the other hand, the Bayesian MARS is nonparametric, and requires sampling methods. It was found that HME requires less memory because it is a parametric regression approach, and the variational Bayesian inference for HME converges faster [28] .

In comparison to NNs, Nowlan and Hinton [112] found ME to be better at fitting the training data. When forced to deal with relatively small training sets, ME was better at generalizing than a comparable single backpropagation network on a vowel recognition task. HME was shown to learn much faster than backpropagation for the same number of parameters by Jordan and Jacobs [113]; however, it is questionable whether this was a good comparison since the NNs were trained using a gradient-descent algorithm, whereas the HME was trained using second-order methods. Additionally, HMEs provide insightful and interpretable results, which NNs do not. In terms of the degree of approximation bounds, NNs and MEs were found to be equivalent by Zeevi *et al*. [80].

Other models of ME include the max-min propagation neural network by Estevez *et al*. [57], where the softmax function was replaced with max(min) units; the probit function by Geweke [58] which computes the inverse cumulative distribution function, and the model by Lima *et al*. [12], where NNs were used at the gate and SVMs at the experts.

IX. Applications of Mixture of Experts

Applications of ME have been seen in various areas, such as electricity demand prediction [64], climate prediction [17], handwriting recognition [19], [114], robot navigation [104], sensor fusion [22], face recognition [20], [115], electroencephalogram signal classification [26], electrocardiogram heart-beat classifier for personalized health care [116], [117], stellar data classification [27], text classification [118], bioinformatics [8], protein interaction prediction [21], gender and ethnic classification of human faces [119], speech recognition and quality estimation [28], [67], [120], audio classification [29], learning appearance models [121], 3-D object recognition [122], image transport regression [30], deformable model fitting [11], filter selection [123], nonlinear system identification of a robotic arm [113], connectivity analysis in the brain from fMRI (functional magnetic resonance imaging) data [13], 3-D human motion reconstruction [9], [10], and for landmine detection [23]. In social studies, ME has been used to analyze social networks and to model voting behavior in elections [124]–[126]. In financial analysis, ME has been used for financial forecasting [127], [128], for risk estimation of asset returns [24], for predicting the exchange rate between the U.S. dollar and the British pound (USD/GBP) [3], and for predicting the direction of variation of the closing price of the Dow Jones industrial average [25].

X. Conclusion

This paper presented a comprehensive survey of developments in the ME model which has been used in a variety

of applications in the areas of regression, classification, and fusion. Over the years, researchers have studied the statistical properties of ME, suggested various training approaches for it (such as EM, variational learning, and Gibbs sampling), and attempted to combine various classification and regression models such as the SVMs and GPs using ME. Therefore, one could perhaps argue that the major advantage of ME is its flexibility in admitting different types of models, and learning model parameters with well-understood optimization methods. In doing so, ME has a niche in modeling nonstationary and piecewise continuous problems.

Given the progression of the ME model, one area that has been underdeveloped is the classification of time-series data. HMM models and ME have been combined for regression and density estimation, but no extension thus far has provided a natural method of finding the subsets of time-series data using an ME model. Such an extension would require a major change in the learning of the experts and the gate; however, it should be possible considering that both the HMM and ME models can be trained with the EM algorithm.

Another area that has not been fully addressed with ME models is context-based classification. In context-based classification, models are learned for the context in which they appear. For example, in landmine detection, radar signals vary significantly for the same underlying object depending on the weather conditions. Therefore, it makes sense to use features that reflect temperature and humidity, and to learn models that distinguish mines from non-mines based on these weather-based features. Although ME has been cited as a context-dependent method in the literature owing to its success in dividing the data in the input or the kernel space, examples such as the one given above have not been specifically addressed. One could, for instance, modify the gate to make distinctions based solely on weather conditions, and modify the experts to work just on the mine/non-mine models. Therefore, the gate and the experts structure of the ME model could provide a solid base from which to learn such context-dependent models.

On the other hand, even though the ME model has matured much over the years, a standard dataset for research has not been established, and the efficiency of the model on large volumes of data or on high-dimensional data has not been extensively tested. We hope that the publicly available datasets and the software listed in the Appendix can be useful for researchers who would like to go further in this area. For such a model as well known and well used as ME, detailed studies on robustness to outliers and noise would be very useful. In addition, although some of the closed-form solutions were listed throughout this paper, most of the inference algorithms are based on iterative techniques and require good initializations. The computational complexity of these inference techniques on small mid-size and large datasets, as well as their sensitivity to initialization, would be worth investigating. Also, in most of the studies, finding the number of experts has been left to the expertise of the software developer instead of an automatic approach. Hence, a combination of the latest Bayesian and clustering studies can potentially provide methods to automate the selection of the number of experts, and simplify the search for the pruning and tree-growing algorithms that have been discussed in this survey.

APPENDIX A
PUBLICLY AVAILABLE DATASETS

Some of the benchmark data that have been used to test HME include the following.

1) Motorcycle data from [129], available at: http://www.stat.cmu.edu/∼larry/all-of-statistics/.
2) DELVE data, specifically the Boston, Kin-8nm, and Pumadyn-32nm datasets, available at: http://www.cs.toronto.edu/∼delve/.
3) Vowel formant frequency database by Peterson and Barney [130], available at: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/speech/database/pb/0.html.
4) Thyroid dataset, available at: http://www.uni-koblenz.de/∼evol/bp-race/thyroid.html.
5) Robot arm data, explained by Bishop [45].

APPENDIX B
AVAILABLE SOFTWARE

The following software is freely available on the Internet.

1) The MIXLAB software in MATLAB, by P. Moeller, available at: http://ftp.idiap.ch/pub/OLD/perry/toolbox.
2) DELVE project in C++, available at: http://www.cs.toronto.edu/∼delve/methods/hme-el-1/codePage.html.
3) Fast training of ME model by C. Sminchisescu [16], at: http://sminchisescu.ins.uni-bonn.de/code/fbme.html.
4) Bayes Net Toolbox for MATLAB, available at: http://code.google.com/p/bnt/.
5) Stand-alone HME MATLAB code by D. Martin, at: http://www.ics.uci.edu/∼fowlkes/software/hme.
6) HME by L. Evers [131] in R environment [132], at: http://www.stats.gla.ac.uk/∼levers/software.html.
7) Mixtools [133] software in R environment, available at: http://CRAN.R-project.org/package=mixtools.
8) Integrative ME software in R environment, available at: http://cran.r-project.org/web/packages/integrativeME/.

APPENDIX C
ME REGRESSION EXAMPLE

In this appendix, we demonstrate the divide-and-conquer property of ME with a simple regression example shown in Fig. 8(a). Such data can be encountered in inverse problems as explained by Bishop [63]. For three experts, in 21 iterations, the randomly initialized expert and gating parameters become aligned as shown in Fig. 8(b) and (c). In Fig. 8(b), the colors green, red, and blue indicate the three experts, and the lines are the plots of the parameters $\mathbf{w}_i$ as a function of the input x; i.e., each parameter $\mathbf{w}_i = [a_i, b_i]$ is plotted as a line $a_i x^{(n)} + b_i$ for $i = 1, 2, 3$. Similarly, in Fig. 8(c), we plot the parameters of the gate $\mathbf{v}_i$. In Fig. 8(d), the gating parameters are evaluated in the softmax function, and the mixing coefficients $g_i(x^{(n)})$ are plotted. Note that for each

input $x^{(n)}$, the mixing coefficients sum to 1. It can be observed that around $x < 0.3$, the gate overwhelmingly picks the second expert with a soft decision on the boundary. In Fig. 8(b), this corresponds to choosing the red line for regression. However, it should be noted that the red line only shows the mean $w^T x^{(n)}$, where the Gaussian probabilities are centered. These Gaussian surfaces for the expert probabilistic outputs $P_i(x^{(n)})$ are displayed in Fig. 9(a)–(c), and the resulting probabilistic output $\sum_i g_i(x^{(n)}) P_i(x^{(n)})$ is displayed in Fig. 9(d).

Appendix D
ME Classification Example

A simple classification problem is shown in Fig. 10(a), in which the blue plus signs from the first class and the red asterisks from the second class are not linearly separable. However, once the data is partitioned in two groups by the gate shown in Fig. 10(b), the data on either side of the gate are linearly separable and can be classified by linear experts as shown in Fig. 10(c) and (d). The parameters of the gate and the experts have been plotted as lines. The gate divides the region into two in Fig. 10(b), such that, to the left of the gating line the first expert is responsible and to the right of the gating line the second expert is responsible. In Fig. 10(c), expert 1 separates the two classes of points that are to the left of the gate. In Fig. 10(d), expert 2 separates the two classes of points that are to the right of the gate. These regions have been explicitly shown in Fig. 11, with the final classification given in (a), the region divided by the gate in (b), and the regions divided by the experts in (c) and (d). The decisions of the gate and the experts are probabilistic, so these regions are actually soft partitions. To make these sharp plots, we picked the maximum of the probabilistic outputs.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[2] M. I. Jordan, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.

[3] A. Coelho, C. Lima, and F. Von Zuben, "Hybrid genetic training of gated mixtures of experts for nonlinear time series forecasting," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 5. Oct. 2003, pp. 4625–4630.

[4] X. Wang, P. Whigham, D. Deng, and M. Purvis, "Time-line hidden Markov experts for time series prediction," *Neural Inf. Process., Lett. Rev.*, vol. 3, no. 2, pp. 39–48, 2004.

[5] Y. Li, "Hidden Markov models with states depending on observations," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 977–984, 2005.

[6] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2006, pp. 883–890.

[7] C. Yuan and C. Neubauer, "Variational mixture of Gaussian process experts," in *Advances in Neural Information Processing Systems 21*. Cambridge, MA: MIT Press, 2009, pp. 1897–1904.

[8] K.-A. L. Cao, E. Meugnier, and G. J. McLachlan, "Integrative mixture of experts to combine clinical factors and gene markers," *Bioinformatics*, vol. 26, no. 9, pp. 1192–1198, 2010.

[9] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Learning to reconstruct 3D human motion from Bayesian mixture of experts, a probabilistic discriminative approach," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. CSRG-502, Oct. 2004.

[10] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "BM^3E: Discriminative density propagation for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 2030–2044, Nov. 2007.

[11] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting with a mixture of local experts," in *Proc. 12th Int. Conf. Comput. Vis.*, 2009, pp. 2248–2255.

[12] C. A. M. Lima, A. L. V. Coelho, and F. J. Von Zuben, "Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification," *Inf. Sci.*, vol. 177, no. 10, pp. 2049–2074, 2007.

[13] B. Yao, D. Walther, D. Beck, and L. Fei-Fei, "Hierarchical mixture of classification experts uncovers interactions between brain regions," in *Advances in Neural Information Processing Systems 22*. Cambridge, MA: MIT Press, 2009, pp. 2178–2186.

[14] L. Cao, "Support vector machines experts for time series forecasting," *Neurocomputing*, vol. 51, pp. 321–339, Apr. 2003.

[15] H.-J. Xing and B.-G. Hu, "An adaptive fuzzy c-means clustering-based mixtures of experts model for unlabeled data classification," *Neurocomputing*, vol. 71, nos. 4–6, pp. 1008–1021, Jan. 2008.

[16] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast algorithms for large scale conditional 3D prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[17] Z. Lu, "A regularized minimum cross-entropy algorithm on mixtures of experts for time series prediction and curve detection," *Pattern Recognit. Lett.*, vol. 27, no. 9, pp. 947–955, 2006.

[18] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 881–888.

[19] R. Ebrahimpour, M. R. Moradian, A. Esmkhani, and F. M. Jafarlou, "Recognition of Persian handwritten digits using characterization loci and mixture of experts," *J. Digital Content Technol. Appl.*, vol. 3, no. 3, pp. 42–46, 2009.

[20] R. Ebrahimpour, E. Kabir, H. Esteky, and M. R. Yousefi, "View-independent face recognition with mixture of experts," *Neurocomputing*, vol. 71, nos. 4–6, pp. 1103–1107, 2008.

[21] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph, "A mixture of feature experts approach for protein-protein interaction prediction," *BMC Bioinf.*, vol. 8, no. S10, p. S6, 2007.

[22] S. E. 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 *Proc. IEEE Int. Geosci. Remote Sensing Symp.*, vol. 2. Jul. 2008, pp. 177–180.

[23] S. E. Yuksel and P. Gader, "Variational mixture of experts for classification with applications to landmine detection," in *Proc. Int. Conf. Pattern Recognit.*, 2010, pp. 2981–2984.

[24] M. S. Yumlu, F. S. Gurgen, and N. Okay, "Financial time series prediction using mixture of experts," in *Proc. 18th Int. Symp. Comput. Inf. Sci.*, 2003, pp. 553–560.

[25] M. Versace, R. Bhatt, O. Hinds, and M. Shiffer, "Predicting the exchange traded fund dia with a combination of genetic algorithms and neural networks," *Expert Syst. Appl.*, vol. 27, no. 3, pp. 417–425, 2004.

[26] I. Guler, E. D. Ubeyli, and N. F. Guler, "A mixture of experts network structure for EEG signals classification," in *Proc. 27th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2005, pp. 2707–2710.

[27] Y. Jiang and P. Guo, "Mixture of experts for stellar data classification," in *Proc. Int. Symp. Neural Netw.*, vol. 2. 2005, pp. 310–315.

[28] S. Mossavat, O. Amft, B. de Vries, P. Petkov, and W. Kleijn, "A Bayesian hierarchical mixture of experts approach to estimate speech quality," in *Proc. 2nd Int. Workshop Qual. Multimedia Exper.*, 2010, pp. 200–205.

[29] H. Harb, L. Chen, and J.-Y. Auloge, "Mixture of experts for audio classification: An application to male female classification and musical genre recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 2. Jun. 2004, pp. 1351–1354.

[30] F. Michel and N. Paragios, "Image transport regression using mixture of experts and discrete Markov random fields," in *Proc. IEEE Int. Symp. Biomed. Imaging: Nano Macro*, Apr. 2010, pp. 1229–1232.

[31] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[33] B. Tang, M. Heywood, and M. Shepherd, "Input partitioning to mixture of experts," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 1. 2002, pp. 227–232.

[34] A. A. Montillo, "Random forests," in *Statistical Foundations of Data Analysis*. New York: Springer-Verlag, Apr. 2009.

[35] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[36] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[37] T. Kim, J. Shotton, and B. Stenger, "Boosting and random forest for visual recognition," in *Proc. IEEE Conf. Comput. Vision (ICCV 2009)*, Tutorial, 2009, Available: http://www.iis.ee.ic.ac.uk/tkkim/iccv09_tutorial.

[38] R. Avnimelech and N. Intrator, "Boosted mixture of experts: An ensemble learning scheme," *Neural Comput.*, vol. 11, no. 2, pp. 483–497, 1999.

[39] M. I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures, *Neural Netw.*, vol. 8, no. 9, pp. 1409–1431, 1995.

[40] W. Jiang and M. Tanner, "On the approximation rate of hierarchical mixtures-of-experts for generalized linear models," *Neural Comput.*, vol. 11, no. 5, pp. 1183–1198, 1999.

[41] A. Carvalho and M. Tanner, "Mixtures-of-experts of autoregressive time series: Asymptotic normality and model specification," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 39–56, Jan. 2005.

[42] Y. Yang and J. Ma, "Asymptotic convergence properties of the EM algorithm for mixture of experts," *Neural Comput.*, vol. 23, no. 8, pp. 2140–2168, 2011.

[43] S. Waterhouse, D. Mackay, and T. Robinson, "Bayesian methods for mixtures of experts," in *Advances in Neural Information Processing Systems 7*. Cambridge, MA: MIT Press, 1996, pp. 351–357.

[44] N. Ueda and Z. Ghahramani, "Bayesian model search for mixture models based on optimizing variational bounds," *Neural Netw.*, vol. 15, no. 10, pp. 1223–1241, 2002.

[45] C. M. Bishop and M. Svensen, "Bayesian hierarchical mixtures of experts," in *Proc. 19th Conf. Uncertain. Artif. Intell.*, 2003, pp. 57–64.

[46] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[47] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[48] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1998.

[49] S. Waterhouse and A. Robinson, "Pruning and growing hierarchical mixtures of experts," in *Proc. 4th Int. Conf. Artif. Neural Netw.*, 1995, pp. 341–346.

[50] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Advances in Neural Information Processing Systems 7*. Cambridge, MA: MIT Press, 1995, pp. 633–640.

[51] K. Chen, L. Xu, and H. Chi, "Improved learning algorithms for mixture of experts in multiclass classification," *Neural Netw.*, vol. 12, no. 9, pp. 1229–1252, 1999.

[52] S. Ng and G. McLachlan, "Extension of mixture-of-experts networks for binary classification of hierarchical data," *Artif. Intell. Med.*, vol. 41, no. 1, pp. 57–67, 2007.

[53] Y. Yang and J. Ma, "A single loop EM algorithm for the mixture of experts architecture," in *Proc. 6th Int. Symp. Neural Netw.*, 2009, pp. 959–968.

[54] N. Ueda and Z. Ghahramani, "Optimal model inference for Bayesian mixture of experts," in *Proc. IEEE Workshop Neural Netw. Signal Process.*, vol. 1. Dec. 2000, pp. 145–154.

[55] V. Tresp, "Mixtures of Gaussian processes," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 654–660.

[56] J. Shi, R. M. Smith, and D. M. Titterington, "Bayesian regression and classification using mixtures of multiple Gaussian processes," *Int. J. Adapt. Control Signal Process.*, vol. 17, no. 2, pp. 149–161, 2003.

[57] P. Estevez and R. Nakano, "Hierarchical mixture of experts and max-min propagation neural networks," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1. Nov.–Dec. 1995, pp. 651–656.

[58] J. Geweke and M. Keane, "Smoothly mixing regressions," *J. Econometrics*, vol. 138, no. 1, pp. 252–290, 2007.

[59] M. I. Jordan and R. A. Jacobs, "Hierarchies of adaptive experts," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1991, pp. 985–992.

[60] V. Ramamurti and J. Ghosh, "Advances in using hierarchical mixture of experts for signal classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 1996, pp. 3569–3572.

[61] Y. Yang and J. Ma, "An efficient EM approach to parameter learning of the mixture of Gaussian processes," in *Proc. 8th Int. Conf. Adv. Neural Netw.*, 2011, pp. 165–174.

[62] G. Hinton, *Introduction to Machine Learning, Decision Trees and Mixtures of Experts*. Cambridge, MA: MIT Press, 2007.

[63] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York: Springer-Verlag, 2006.

[64] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," *Int. J. Neural Syst.*, vol. 6, no. 4, pp. 373–399, 1995.

[65] K. Saito and R. Nakano, "A constructive learning algorithm for an HME," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 2. Jun. 1996, pp. 1268–1273.

[66] J. Fritsch, M. Finke, and A. Waibel, "Adaptively growing hierarchical mixtures of experts," in *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1996, pp. 459–465.

[67] F. Peng, R. A. Jacobs, and M. A. Tanner, "Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition," *J. Amer. Stat. Assoc.*, vol. 91, no. 435, pp. 953–960, 1996.

[68] R. A. Jacobs, F. Peng, and M. A. Tanner, "A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures," *Neural Netw.*, vol. 10, no. 2, pp. 231–241, 1997.

[69] A. Kanaujia and D. Metaxas, "Learning ambiguities using Bayesian mixture of experts," in *Proc. 18th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2006, pp. 436–440.

[70] J. Fritsch, M. Finke, and A. Waibel, "Context-dependent hybrid HME/HMM speech recognition using polyphone clustering decision trees," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1997, pp. 1759–1762.

[71] J. Q. Shi, R. Murray-Smith, and D. M. Titterington, "Hierarchical Gaussian process mixtures for regression," *Stat. Comput.*, vol. 15, no. 1, pp. 31–41, 2005.

[72] S. Waterhouse and A. Robinson, "Classification using hierarchical mixtures of experts," in *Proc. IEEE Workshop Neural Netw. Signal Process. IV*, Sep. 1994, pp. 177–186.

[73] K. Chen, D. Xie, and H. Chi, "Speaker identification based on the time-delay hierarchical mixture of experts," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. Nov.–Dec. 1995, pp. 2062–2066.

[74] S.-K. Ng and G. McLachlan, "Using the EM algorithm to train neural networks: Misconceptions and a new algorithm for multiclass classification," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 738–749, May 2004.

[75] M. K. Titsias and A. Likas, "Mixture of experts classification using a hierarchical mixture model," *Neural Comput.*, vol. 14, no. 9, pp. 2221–2244, 2002.

[76] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[77] S. R. Waterhouse, "Classification and regression using mixtures of experts," Ph.D. thesis, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 1997.

[78] D. M. Titterington, *The EM Algorithm, Variational Approximations and Expectation Propagation for Mixtures*. New York: Wiley, 2011, ch. 1, pp. 1–29.

[79] R. A. Jacobs, "Bias/variance analyses of mixtures-of-experts architectures," *Neural Comput.*, vol. 9, no. 2, pp. 369–383, 1997.

[80] A. Zeevi, R. Meir, and V. Maiorov, "Error bounds for functional approximation and estimation using mixtures of experts," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1010–1025, May 1998.

[81] W. Jiang, "The VC dimension for mixtures of binary classifiers," *Neural Comput.*, vol. 12, no. 6, pp. 1293–1301, 2000.

[82] V. N. Vapnik, "Principles of risk minimization for learning theory," in *Advances in Neural Information Processing Systems*, vol. 4. Cambridge, MA: MIT Press, 1992, pp. 831–838.

[83] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[84] M. Anthony and N. Biggs, *Computational Learning Theory: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[85] W. Jiang and M. Tanner, "On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 1005–1013, May 2000.

[86] W. Jiang and M. A. Tanner, "On the identifiability of mixtures-of-experts," *Neural Netw.*, vol. 12, no. 9, pp. 1253–1258, 1999.

[87] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Pacific Grove, CA: Brooks/Cole, 2001.

[88] W. Jiang and M. A. Tanner, "Hierarchical mixtures-of-experts for exponential family regression models: Approximation and maximum likelihood estimation," *Ann. Stat.*, vol. 27, no. 3, pp. 987–1011, 1999.

[89] Y. Ge and W. Jiang, "A note on mixtures of experts for multiclass responses: Approximation rate and consistent Bayesian inference," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 329–335.

[90] Y. Ge and W. Jiang, "On consistency of Bayesian inference with mixtures of logistic regression," *Neural Comput.*, vol. 18, no. 1, pp. 224–243, 2006.

[91] A. Zeevi, R. Meir, and R. Adler, "Time series prediction using mixtures of experts," in *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp. 309–315.

[92] C. Wong and W. Li, "On a logistic mixture autoregressive model," *Biometrika*, vol. 88, no. 3, pp. 833–846, 2001.

[93] K. Chen, D. Xie, and H. Chi, "Combine multiple time-delay HMEs for speaker identification," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. Jun. 1996, pp. 2015–2020.

[94] K. Chen, D. Xie, and H. Chi, "A modified HME architecture for text-dependent speaker identification," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1309–1313, Sep. 1996.

[95] T. W. Cacciatore and S. J. Nowlan, "Mixtures of controllers for jump linear and non-linear plants," in *Advances in Neural Information Processing Systems 6*. Cambridge, MA: MIT Press, 1993, pp. 719–726.

[96] S. Liehr, K. Pawelzik, J. Kohlmorgen, S. Lemm, and K.-R. Müller, "Hidden Markov mixtures of experts for prediction of non-stationary dynamics," in *Proc. Workshop Neural Netw. Signal Process.*, 1999, pp. 195–204.

[97] T. G. Dietterich, "Machine learning for sequential data: A review," in *Proc. Joint IAPR Int. Workshop Struct., Syntactic, Stat. Pattern Recognit.*, 2002, pp. 15–30.

[98] M. I. Jordan, Z. Ghahramani, and L. K. Saul, "Hidden Markov decision trees," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 1996, pp. 501–507.

[99] Y. Bengio and P. Frasconi, "Input output HMMs for sequence processing," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1231–1249, Sep. 1996.

[100] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, no. 2, pp. 245–273, 1997.

[101] Y. Zhao, R. Schwartz, J. Sroka, and J. Makhoul, "Hierarchical mixtures of experts methodology applied to continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5. May 1995, pp. 3443–3446.

[102] M. Meila and M. I. Jordan, "Learning fine motion by Markov mixtures of experts," in *Advances in Neural Information Processing Systems 8*. Cambridge, MA: MIT Press, 1996, pp. 1003–1009.

[103] J. Fritsch and M. Finke, "Improving performance on switchboard by combining hybrid HME/HMM and mixture of Gaussians acoustic models," in *Proc. Eurospeech*, 1997, pp. 1963–1966.

[104] E. Trentin and R. Cattoni, "Learning perception for indoor robot navigation with a hybrid HMM/recurrent neural networks approach," *Connect. Sci.*, vol. 11, nos. 3–4, pp. 243–265, 1999.

[105] E. Trentin and D. Giuliani, "A mixture of recurrent neural networks for speaker normalisation," *Neural Comput. Appl.*, vol. 10, no. 2, pp. 120–135, 2001.

[106] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[107] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[108] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 512–518.

[109] S. Waterhouse and G. Cook, "Ensemble methods for phoneme classification," in *Advances in Neural Information Processing Systems*, vol. 9. Cambridge, MA: MIT Press, 1997.

[110] Y. Shimshoni and N. Intrator, "Classification of seismic signals by integrating ensembles of neural networks," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1194–1201, May 1998.

[111] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.

[112] S. J. Nowlan and G. E. Hinton, "Evaluation of adaptive mixtures of competing experts," in *Advances in Neural Information Processing Systems 3*. Cambridge, MA: MIT Press, 1991.

[113] M. Jordan and R. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2. 1993, pp. 1339–1344.

[114] G. E. Hinton, M. Revow, and P. Dayan, "Recognizing handwritten digits using mixtures of linear models," in *Advances in Neural Information Processing Systems 7*. Cambridge, MA: MIT Press, 1995, pp. 1015–1022.

[115] A. Hajiany, N. T. Makhsoos, and R. Ebrahimpour, "View-independent face recognition with biological features based on mixture of experts," in *Proc. ISDA*, 2009, pp. 1425–1429.

[116] Y. H. Hu, S. Palreddy, and W. Tompkins, "Customized ECG beat classifier using mixture of experts," in *Proc. IEEE Workshop Neural Netw. Signal Process.*, Aug.–Sep. 1995, pp. 459–464.

[117] Y. H. Hu, S. Palreddy, and W. Tompkins, "A patient-adaptable ECG beat classifier using a mixture of experts approach," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 9, pp. 891–900, Sep. 1997.

[118] A. Estabrooks and N. Japkowicz, "A mixture-of-experts framework for text classification," in *Proc. Workshop Comput. Natural Lang. Learn., Assoc. Comput. Linguist.*, 2001, pp. 1–8.

[119] S. Gutta, J. Huang, P. Jonathon, and H. Wechsler, "Mixture of experts for classification of gender, ethnic origin, and pose of human faces," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 948–960, Jul. 2000.

[120] A. Tuerk, "The state based mixture of experts HMM with applications to the recognition of spontaneous speech," Ph.D. thesis, Dept. Eng., Univ. Cambridge, Cambridge, U.K., Sep. 2001.

[121] C. Bregler and J. Malik, "Learning appearance based models: Mixtures of second moment experts," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, pp. 845–851.

[122] P. Walter, I. Elsen, H. Muller, and K. Kraiss, "3D object recognition with a specialized mixtures of experts architecture," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 5. 1999, pp. 3563–3568.

[123] W. Chaer, R. Bishop, and J. Ghosh, "A mixture-of-experts framework for adaptive Kalman filtering," *IEEE Trans. Syst., Man, Cybern., Part B, Cybern.*, vol. 27, no. 3, pp. 452–464, Jun. 1997.

[124] I. C. Gormley and T. B. Murphy, "A mixture of experts model for rank data with applications in election studies," *Ann. Appl. Stat.*, vol. 2, no. 4, pp. 1452–1477, 2008.

[125] I. C. Gormley and T. B. Murphy, "A mixture of experts latent position cluster model for social network data," *Stat. Methodol.*, vol. 7, no. 3, pp. 385–405, 2010.

[126] I. C. Gormley and T. B. Murphy, *Mixture of Experts Modelling with Social Science Applications*. New York: Wiley, 2011, ch. 5, pp. 101–121.

[127] Y. M. Cheung, W. M. Leung, and L. Xu, "Application of mixture of experts model to financial time series forecasting," in *Proc. Int. Conf. Neural Netw. Signal Process.*, 1995, pp. 1–4.

[128] A. S. Weigend and S. Shi, "Predicting daily probability distributions of S&P500 returns," *J. Forecast.*, vol. 19, no. 4, pp. 375–392, 2000.

[129] B. Silverman, "Some aspects of the spline smoothing approach to non-parametric curve fitting," *J. Royal Stat. Soc. B*, vol. 47, no. 1, pp. 1–52, 1985.

[130] B. Peterson, "Control methods used in a study of the vowels," *J. Acoust. Soc. Amer.*, vol. 24, no. 2, pp. 175–184, 1952.

[131] L. Evers. (2007, Apr.). *HME: Methods for Fitting Hierarchical Mixtures of Experts R Package Version 0.1-0* [Online]. Available: http://www.stats.gla.ac.uk/levers/software.html

[132] *The R Project for Statistical Computing*. (2012) [Online]. Available: http://www.r-project.org

[133] T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young, "An R package for analyzing finite mixture models," *J. Stat. Softw.*, vol. 32, no. 6, pp. 1–29, 2009.

Seniha Esen Yuksel (M'11) received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2003, the M.Sc. degree in electrical and computer engineering from the University of Louisville, Louisville, KY, in 2006, and the Ph.D. degree from the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, in 2011.

She is currently a Post-Doctoral Associate with the Department of Materials Science and Engineering, University of Florida, where she is developing algorithms for explosives detection from hyperspectral data. Her current research interests include machine learning, statistical data analysis, applied mathematics, medical imaging, and computer vision.

Dr. Yuksel was a recipient of the University of Florida College of Engineering Outstanding International Student Award in 2010, and the Phyllis M. Meek Spirit of Susan B. Anthony Award from the University of Florida in 2008.

Joseph N. Wilson (M'05) received the B.S. degree in applied mathematics with an emphasis on computer science from Florida State University, Tallahassee, in 1977, and the M.S. degree in applied mathematics and computer science and the Ph.D. degree in computer science from the University of Virginia, Charlottesville, in 1980 and 1985, respectively.

He has been with the Faculty of the Computer and Information Science and Engineering Department, University of Florida, Gainesville, since 1984, where he served as an Associate Chair from 1994 to 2001. His current research interests include machine learning, computer vision, and image and signal processing.

Paul D. Gader (M'86–SM'9–F'11) received the Ph.D. degree in mathematics for image processing related research from the University of Florida, Gainesville, in 1986. His doctoral dissertation focused on algebraic methods for parallel image processing.

He was a Senior Research Scientist with Honeywell Systems and Research Center, Minneapolis, MN, a Research Engineer and Manager with the Environmental Research Institute of Michigan, Ann Arbor, MI, and has been a Faculty Member with the University of Wisconsin, Oshkosh, and the University of Missouri, Columbia. He is currently a Professor of computer and information science and engineering with the University of Florida. He performed his first research in image processing in 1984, working on algorithms for the detection of bridges in forward-looking infrared imagery as a Summer Student Fellow with Eglin Air Force Base, Valparaiso, FL. He has published many refereed papers in journals and conference proceedings. His current research interests include theoretical and applied problems, including fast computing with linear algebra, mathematical morphology, fuzzy sets, Bayesian methods, handwriting recognition, automatic target recognition, biomedical image analysis, landmine detection, human geography, and hyperspectral and LiDAR image analysis.