Context-based Classification via Mixture of Hidden Markov Model Experts with Applications in Landmine Detection

Seniha Esen Yuksel[1,*] and Paul D. Gader[2]

[1]Department of Electrical and Electronics Engineering, Hacettepe University, Ankara, Turkey
[2]Department of Computer & Information Science & Engineering, University of Florida, USA.
[*]E-mail: eyuksel@ee.hacettepe.edu.tr

Abstract: In many applications data classification may be hindered by the existence of multiple contexts that produce an input sample. To alleviate the problems associated with multiple contexts, context-based classification is a process that uses different classifiers depending on a measure of the context. Context-based classifiers offer the promise of increasing performance by allowing classifiers to become experts at classifying input samples of certain types, rather than trying to force single classifiers to perform well on all possible inputs. This paper introduces a novel mixture of experts model, the Mixture of Hidden Markov Model Experts (MHMME), for context-based classification of samples that are variable length sequences; and derives the update equations for a single probabilistic model that to learn the experts and a gate that connects the experts. The model has a similar high-level structure to the mixture of experts model but has the novelty that the gates and the experts are HMMs and the input data are sequences. Experimental results are presented on three datasets including one for landmine detection. Detailed analysis of the model is provided; which, over multiple runs and cross-validation experiments, show superior results over the compared algorithms.

1. Introduction

Time-series or sequential data often show multiple patterns owing to the different contexts that they appear in. For example, electricity usage has both seasonal and socio-economic patterns. Therefore, a software that detects fraud should consider electricity usage within these contexts. Similarly, in electrocardiogram (ECG) classification, certain ethnic groups and athletes show slower resting heartbeats. Therefore, one has to consider healthy versus unhealthy heartbeats in these contexts. Unfortunately, unlike these examples, contexts are generally hard to define, they are often interlaced, and do not have sharp boundaries. Moreover, context information might be inherent in the data, but not be known to the data modeler. In such cases, we define a context as a group of similar signatures.

A more involved example to demonstrate this problem is landmine detection. The estimated 60, 000, 000-100, 000, 000 active buried landmines around world [5] have various sizes and types. They are roughly categorized into four groups according to their metallic content and intended targets as high metal anti-tank (HMAT), high metal anti-personnel (HMAP), low metal anti-tank (LMAT), and low metal anti-personnel (LMAP). However, these groups mostly overlap, and the signals collected from these mines can be significantly affected by changes in temperature, humidity, and soil conditions.

One way to deal with multiple contexts is the mixture of experts model. In the ME architecture, a set of expert networks and a gating network cooperate with each other to solve a non-linear supervised learning problem by dividing the input space into a set of regions as shown in Fig. 1. In the traditional ME model, the gate and experts are simple surfaces; however, ME models have been found useful because of their modular and flexible structure, as described by the survey paper by Yuksel et al. [39] which summarizes the ME and the numerous advances with it taking place in the late 2000's.

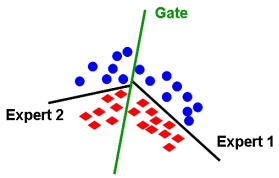


Fig. 1. *Classification using the ME model. Red diamonds and blue dots represent data coming from two classes. The gate divides the region in two with a soft decision, then the experts learn the simple surfaces to separate the two classes. Taken from [39].*

In this study, we are introducing a novel model, mixture of hidden Markov model experts (MHMME), that can both decompose time-series data into multiple contexts and learn expert classifiers for each context. In this model, a gate of hidden Markov models defines the contexts and cooperates with a set of hidden Markov model experts that provide multi-class classification. The MHMME model is inspired from the mixture of experts (ME) model [15, 16], and extends it to time-series (and sequential) data for classification. Therefore, MHMME carries the advantages of the ME model and also brings advantages that set it apart from the other models. The main advantages can be summarized as follows:

- MHMME model provides a divide and conquer approach, is probabilistic, and has soft boundaries – all of which support context learning. In addition, unlike the traditional mixture models where the mixture coefficient is a scalar, in MHMME the mixture coefficient (i.e. the gate) depends on the input and helps define the contexts that are unknown to the data modeler.
- The learning of the contexts and the classifiers is accomplished simultaneously, in one model. This papers gives the derivations to arrive at this single probabilistic model.
- During training, there is no hard clustering of data, which means that the sequences can freely move between contexts and classifiers during training.
- MHMME considers the temporal connections in time-series data, and is suitable for high-dimensional sequential data of varying lengths due to the use of the hidden Markov models (HMMs). In addition, HMMs at the gates and the experts can be of different topologies (number of states, observation symbols etc.).
- MHMME is suitable for multi-class classification.
- Experiments on synthetic and real data show that MHMME can perform better than ME and HMMs, and can do well in comparison to state-of-the-art models.

To this end, in the sections that follow, we first compare MHMME to the existing models in the literature and explain the need to develop MHMME. Then, in Sec. 3 we give a brief introduction to the ME model. In Sec. 4, we derive the update equations of the MHMME model and explain its implementation. We demonstrate the intuition behind the MHMME model on a synthetic example in Sec. 5. We show our results for landmine detection from metal detector data in Sec. 6, and also on an object recognition dataset consisting of varying length sequences in Sec. 7. We compare MHMME to the ME-only and HMM-only models, to its individual components, ie. the gate HMMs and the expert HMMs, and also compare our results to those in the literature.

2. Comparison of MHMME to the ME literature: Why the need arises

In the ME literature, a number of models [6, 7, 20, 30, 32, 33, 40] were described that extend the ME architecture to time-series data. These models, however, are only applicable to regression, and they use a one-step-ahead or multi-step-ahead prediction in which the last couple of values of the time-series data are used as features in a neural network. Such models cannot handle data of varying length and the use of multilayer network-type approaches prevent them from completely describing the temporal properties of a time-series dataset. In contrast to these models, our study is on classification, and is focused on varying length sequences. Note that with varying or uneven lengths, we mean that the observation sequences do not all have the same length.

On the other hand, there are a number of studies [12, 29, 34, 42] that combine HMMs and MEs. However, despite the similar names, these models are quite different than our model. To be specific, in [12], each state of the HMM is a mixture of experts (whereas in our paper, each HMM is a part of an expert). Similarly, in [29] an HMM model was modified to have two separate branches, one for slow speech and one for fast speech. In the study by Zhao et al. [42], hierarchically organized relatively small neural networks were trained to perform probability density estimation. Therefore, Zhao's model does not use HMMs, instead, it uses mixture models to mimick an HMM. Finally in [34], SVM classifiers were trained for each region in the brain, and they were connected with a Hidden Conditional Random Field (HCRF). One can think of the HCRFs as the gate, and the SVMs as the experts, but temporal sequences were not of interest.

The model proposed herein differs from these previously published models and has distinct properties that are worthwhile to investigate. It provides a stand-alone model to find the contexts and classifiers for high dimensional data sequences with uneven lengths, and considers their temporal properties in training. Unlike the traditional mixture models where the mixture coefficient is a scalar, in the MHMME model, the mixture coefficient (i.e. the gate) depends on the input. Therefore, the experts and the gate need to be trained simultaneously, and it is derived in this study.

3. Mixture of Experts

In this section, we provide a brief overview of the traditional ME algorithm. For a K-class classification problem, let $k \quad 1...K$ be the class index. Let I be the number of experts as shown in the figure with $i \quad 1...I$ denoting the expert index. Each expert is a K-class classifier. Then, these classifier experts are connected by a gate, which essentially gives a weight to each expert. Finally, the desired output $\mathbf{y}^{(n)}$ is of length K and $y_k^{(n)} = 1$ if the input $\mathbf{x}^{(n)}$ belongs to class k and 0 otherwise.

Considering all the experts, there are K parameter sets $\{\{\mathbf{w}_{ik}\}_{i=1}^{I}\}_{k=1}^{K}$ to be learned. Using these

weights, the expert outputs per class are found by softmax functions:

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

which are the means of the experts' multinomial probability models, $P(\mathbf{y}^{(n)}|i, \mathbf{x}^{(n)})$, which is in short referred to as $P_i(\mathbf{y})$:

$$P_i(\mathbf{y}) = \prod_k \hat{y}_{ik}^{y_k}. \tag{2}$$

The gate is the scalar defined by the softmax function:

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

where $\beta_i(\mathbf{x}, \mathbf{v})$ are functions of the gate parameter $\mathbf{v}$, and are linear given by $\beta_i(\mathbf{x}, \mathbf{v}) = \mathbf{v}_i^T \mathbf{x}$, 1 in the original ME. The softmax function is a smooth version of the winner-takes-all model.

In ME, it is assumed that the experts are mutually exclusive [33]. Hence, using Bayes' rule, given an input vector $\mathbf{x}$ and a desired (target) output vector $\mathbf{y}$, the total probability of observing the target vector $\mathbf{y}$ can be written in terms of the probabilities of belonging to the region specified by one of the I experts as:

$$P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{I} g_i P(\mathbf{y}^{(n)}|i, \mathbf{x}^{(n)}) \tag{4}$$

where $g_i = P(i|\mathbf{x}^{(n)}), \quad i = 1, ..., I$, is the probability of selecting the i^{th} expert depending on the input, and $P(\mathbf{y}^{(n)}|i, \mathbf{x}^{(n)})$ is the probability of the input belonging to the region specified by the i^{th} expert.

The ME training algorithm introduces hidden variables and maximizes the log of complete data likelihood obtained from the probability in Eq. 4 to learn the parameters of the experts and the gate. Interested readers are referred to [39] for details of obtaining the update equations.

Once all the weights are learned, to make a single prediction, the outputs are computed per class:

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

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

4. Mixture of Hidden Markov Model Experts

The MHMME architecture introduced in this study is illustrated in Fig. 2 where the gate has I HMM models. Each branch of the gate is connected to an expert, and an expert is a set of K HMMs, one for each class. The gate partitions the set of all time-series data that can serve as inputs to the HMMs, and defines the contexts where the individual expert opinions are trustworthy. Experts discriminate data in these partitions based on class labels. Comparing Fig. **??** and Fig. 2,

the architectures are similar, which gives the divide-and-conquer property to MHMME. The important part then, is to derive the probabilistic update equations to train the gate and the experts simultaneously.

For the rest of the paper, the notation used for HMMs is as follows:

- W = number of states.
- M = number of symbols in the codebook.
- T = length of observation sequence.
- $V = \{v_1, ..., v_M\}$ the discrete set of observation symbols.
- $O = O_1 O_2 ... O_T$ denotes an observation sequence, where $O_t \in V$ is the observation at time t.
- $Q = q_1 q_2 ... q_T$ is a fixed state sequence, where q_t is the state at time t .
- $S = \{S_1, S_2, ..., S_W\}$ are the individual states.
- I = number of experts, and $i \quad 1...I$ is the expert index.
- K = number of classes, and $k \quad 1...K$ is the class index.
- λ_{ik} = HMM model for the k^{th} class at the i^{th} expert.
- $\psi_i = i^{th}$ HMM model at the gate.
- The initial state distribution $\pi = \{\pi_r\}_{r=1}^{W}$, where $\pi_r = P(q_1 = S_r)$ is the probability of being in state r at time $t = 1$.
- The state transition probability $A = \{\{a_{rj}\}_{r=1}^{W}\}_{j=1}^{W}$, where $a_{rj} = P(q_{t+1} = S_j | q_t = S_r)$ is the probability of being in state j at time $t + 1$ given that we are in state r at time t.
- The observation symbol probability distribution $B = \{\{b_j(m)\}_{j=1}^{W}\}_{m=1}^{M}$, where $b_j(m) = P(v_m \texttt{ at } t | q_t = j)$ is the probability of observing the symbol v_m given that we are in state j.

We denote the HMM models at the gate with $\Psi = \{\psi\}_{i=1}^{I}$, the HMM models at the experts with $\Lambda_i = \{\lambda_{ik}\}_{k=1}^{K}$, and finally, we denote the set of all the gate and expert parameters as $\Theta = \{\Psi, \Lambda\}$.

Let the data be denoted by $D = \{\mathbf{O}, Y\}$ where $\mathbf{O} = \{O^{(n)}\}_{n=1}^{N}$ represents the input sequences, and $Y = \{\mathbf{y}^{(n)}\}_{n=1}^{N}$ represents the class coded true outputs of training data such that $\mathbf{y}^{(n)} = y_1^{(n)}, ..y_k^{(n)}, ..y_K^{(n)}$, and

$$y_k^{(n)} = \begin{cases} 1 & \text{if } O^{(n)} \text{ belongs to class } k \text{ ;} \\ 0 & \text{otherwise.} \end{cases}$$

The probability function for the gate and experts is the following:

$$P(Y|\mathbf{O}, \Theta) = \prod_{n=1}^{N} \sum_{i=1}^{I} g_i(O^{(n)}, \Psi_i) \;\; P_i(\mathbf{y}^{(n)}|\Lambda_i) \tag{5}$$

where $g_i(O^{(n)}, \Psi_i)$ is the gate's probabilistic estimate that the sequence $O^{(n)}$ belongs to the space defined by expert i. In other words, $g_i(O^{(n)}, \Psi_i) = P(i|O^{(n)}, \Psi_i)$, the probability of selecting the

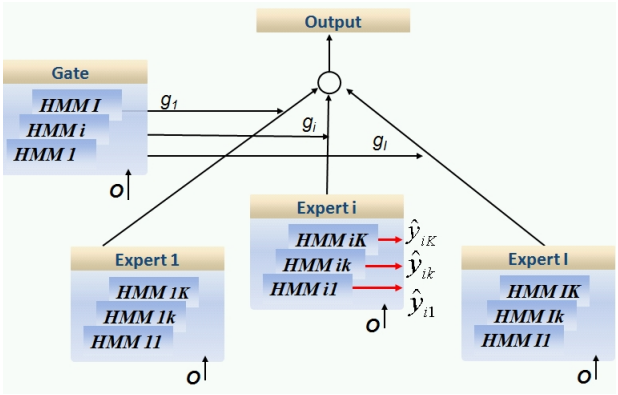


Fig. 2. *MHMME architecture with I experts for K classes. A gate partitions the set of all time-series data that can serve as inputs to the HMMs. Experts learn to discriminate the classes in these partitions.*

i^{th} expert given the sequence $O^{(n)}$. The second term in Eq. 5, $P_i(\mathbf{y}^{(n)}, \Lambda_i)$, is the probability that the i^{th} expert has generated $\mathbf{y}^{(n)}$ given the sequence $O^{(n)}$. In the rest of this paper, we will denote $g_i(O^{(n)}, \Psi_i)$ with $g_i^{(n)}$, and $P_i(\mathbf{y}^{(n)}|\Lambda_i)$ with $P_i(\mathbf{y}^{(n)})$.

The gate's probabilistic estimate is obtained by a softmax function that considers the confidences of all the HMM models at the gate, given as:

$$g_i^{(n)} = \frac{\exp f(O^{(n)}|\psi_i)}{\sum_{m=1}^{I} \exp f(O^{(n)}|\psi_m)} \tag{6}$$

where $f(O^{(n)}|\psi_i)$ is the Viterbi log-likelihood of observation $O^{(n)}$ for an HMM model ψ_i.

Similar to the gate, the HMMs at the experts compute the Viterbi log-likelihood

$$f(O^{(n)}|\lambda_{ik}) = \log P_{HMM}(O^{(n)}|Q, \lambda_{ik}), \tag{7}$$

where the Viterbi likelihood $P_{HMM}(O^{(n)}|Q, \lambda_{ik})$ is

$$P_{HMM}(O^{(n)}|Q, \lambda_{ik}) = \pi_{q_0}^{(ik)} \prod_{t=1}^{T-1} a_{q_t q_{t+1}}^{(ik)} \prod_{t=1}^{T} b_{q_t}^{(ik)}(o_t). \tag{8}$$

These log-likelihoods are converted to probabilities by a softmax function, and the output of expert i for class k is $\hat{y}_{ik}^{(n)}$, computed as:

$$\hat{y}_{ik}^{(n)} = \frac{\exp f(O^{(n)}|\lambda_{ik})}{\sum_{r=1}^{K} \exp f(O^{(n)}|\lambda_{ir})}. \tag{9}$$

which is also the mean of its multinomial probability model. More specifically, for a given sequence $O^{(n)}$, expert i produces a prediction with probability $P_i(\mathbf{y}^{(n)})$ following a multinomial distribution with mean $\hat{y}_{ik}$ such that:

$$P_i(\mathbf{y}^{(n)}) = \prod_k \hat{y}_{ik}^{y_k} \quad . \tag{10}$$

Finally, the output of the MHMME architecture, $\{\hat{y}_k\}_{k=1}^K$, is a weighted sum of the expert outputs:

$$\hat{y}_k^{(n)} = \sum_i g_i^{(n)} \ \hat{y}_{ik}^{(n)} \ . \tag{11}$$

Typically, the observation sequence $O^{(n)}$ is assigned to the k^{th} class that gives the maximum $\{\hat{y}_k\}_{k=1}^K$ as:

$$k^* = \underset{k}{\text{argmax}} \ \{\hat{y}_k^{(n)}\}_{k=1}^K \ . \tag{12}$$

It is worthy to notice the relationship between the mixture model Eq. 11 and its probabilistic counterpart in Eq. 5. The parameters of the MHMME model are learned using the probabilistic model in Eq. 5, which will be explained in the next section.

4.1. Training of the MHMME model

The parameters optimizing the distribution $P(Y|\mathbf{O}, \Theta)$ in Eq. 5 can be found by introducing latent variables Z and by maximizing the complete distribution $P(Y, Z|\mathbf{O}, \Theta)$ with the expectation-maximization (EM) algorithm. These latent variables are $Z = \{\{z_i^{(n)}\}_{n=1}^N\}_{i=1}^I$ such that

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

where R_i is the region specified by expert i. Hence, the complete data distribution becomes:

$$P(Y, Z|\mathbf{O}, \Theta) = \prod_n \prod_i \left(g_i^{(n)} P_i(y^{(n)}) \right)^{z_i^{(n)}} \tag{13}$$

So now, in the E step, we find the expectations of the hidden variables [16, 17] as:

$$h_i^{(n)} = \frac{g_i^{(n)} P_i(\mathbf{y}^{(n)})}{\sum_j g_j^{(n)} P_j(\mathbf{y}^{(n)})}. \tag{14}$$

In the M step, we maximize the expected complete data likelihood $E_Z(\log P(Y, Z|\mathbf{O}, \Theta))$ from the objective function:

$$\begin{aligned} Q(\Theta, \Theta^{(p)}) &= E_Z(l(Y, Z|\mathbf{O}, \Theta)) \\ &= \sum_{n=1}^N \sum_{i=1}^I h_i^{(n)} \log g_i^{(n)} + \sum_{n=1}^N \sum_{i=1}^I h_i^{(n)} \log P_i(\mathbf{y}^{(n)}) \ . \end{aligned} \tag{15}$$

In the M step, h_is are kept fixed, so the two terms on the right side of the equation are decoupled and can be computed independently for the experts and the gate. We refer to these objective functions as Q_g for the gate and as Q_e for the experts, given as:

$$Q_g = \sum_{n=1}^N \sum_{i=1}^I h_i^{(n)} \log g_i^{(n)} \tag{16}$$

$$Q_e = \sum_{n=1}^{N} \sum_{i=1}^{I} h_i^{(n)} \log P_i(\mathbf{y}^{(n)}) \ . \tag{17}$$

Note that these equations follow from the ME model and are given here for completeness. Unfortunately, $\max_{\psi_i} Q^g$ and $\max_{\lambda_{ik}} Q^e$ cannot be solved analytically because of the softmax function. Therefore, iterative and gradient based algorithms have been used in the past for the learning of the ME model [16,17,31]. Similarly, we use the gradient methods to find the parameters of the HMMs in the MHMME model.

In the M step, we search for the HMM parameters that maximize these objective functions:

$$\lambda_{ik}^{(p+1)} = \underset{\lambda_{ik}}{\text{argmax}} \ Q_e \tag{18}$$

$$\psi_i^{(p+1)} = \underset{\psi_i}{\text{argmax}} \ Q_g \tag{19}$$

where p denotes the iteration number. Explicitly, the HMM parameters to be estimated in the experts are $\lambda_{ik} = \{A^{(ik)}, B^{(ik)}\}$. We will denote each element of the A matrix as $a_{rj}^{(ik)}$ with $r = 1...W$, $j = 1...W$, and each element of the B matrix as $b_{mj}^{(ik)}$ with $m = 1...M$, $j = 1...W$. To ensure that the estimated parameters satisfy the constraints $a_{rj} \geq 0$, $\sum_{j=1}^{W} a_{rj} = 1$, $b_{mj} \geq 0$, and $\sum_{m=1}^{M} b_{mj} = 1$, we map these parameters using log, and map them back with softmax functions:

$$a_{rj} \rightarrow \quad \tilde{a}_{rj} = \log a_{rj} \ , \tag{20}$$

$$a_{rj} = \quad \frac{\exp \tilde{a}_{rj}}{\sum_{j'=1}^{W} \exp \tilde{a}_{rj'}} \ , \tag{21}$$

$$b_{mj} \rightarrow \quad \tilde{b}_{mj} = \log b_{mj} \ , \tag{22}$$

$$b_{mj} = \quad \frac{\exp \tilde{b}_{mj}}{\sum_{m'=1}^{M} \exp \tilde{b}_{m'j}} \ . \tag{23}$$

Such mappings are common in gradient based training models such as [18,21]. The HMM parameters that maximize the objective functions are found by gradient ascent updates as:

$$\tilde{a}_{rj}^{(ik)}(p+1) = \tilde{a}_{rj}^{(ik)}(p) + \epsilon \frac{\partial Q_e(\Lambda(p))}{\partial \tilde{a}_{rj}^{(ik)}(p)}, \tag{24}$$

$$\tilde{b}_{mj}^{(ik)}(p+1) = \tilde{b}_{mj}^{(ik)}(p) + \epsilon \frac{\partial Q_e(\Lambda(p))}{\partial \tilde{b}_{mj}^{(ik)}(p)}, \tag{25}$$

where

$$\frac{\partial Q_e(\Lambda)}{\partial \tilde{a}_{rj}^{(ik)}} = \sum_{n=1}^{N}\sum_{m=1}^{K} h_i^{(n)} \left(y_k^{(n)} - \hat{y}_{ik}^{(n)}\right) \frac{\partial f(O^{(n)}, \Lambda_{ik})}{\partial a_{rj}^{(ik)}} \frac{\partial a_{rj}^{(ik)}}{\partial \tilde{a}_{rj}^{(ik)}}, \tag{26}$$

$$\frac{\partial Q_e(\Lambda)}{\partial \tilde{b}_{mj}^{(ik)}} = \sum_{n=1}^{N}\sum_{m=1}^{K} h_i^{(n)} \left(y_k^{(n)} - \hat{y}_{ik}^{(n)}\right) \frac{\partial f(O^{(n)}, \Lambda_{ik})}{\partial b_{mj}^{(ik)}} \frac{\partial b_{mj}^{(ik)}}{\partial \tilde{b}_{mj}^{(ik)}}, \tag{27}$$

and the gradients are

$$\frac{\partial f(O^{(n)}, \lambda_{ik})}{\partial a_{rj}^{(ik)}} = \frac{1}{a_{rj}^{(ik)}} \sum_{t=1}^{T} \delta(q_t^{(n)} = m, q_{t+1}^{(n)} = j), \tag{28}$$

$$\frac{\partial b_{ij}^{(ik)}}{\partial \tilde{b}_{mj}^{(ik)}} = b_{mj}^{(ik)}(1 - b_{mj}^{(ik)}), \tag{29}$$

$$\frac{\partial f(O^{(n)}, \lambda_{ik})}{\partial b_{mj}^{(ik)}} = \frac{1}{b_{mj}^{(ik)}} \sum_{t=1}^{T} \delta(q_t^{(n)} = m, Q_V(O_t^{(n)}) = j). \tag{30}$$

Similarly, the updates for the gate are:

$$\frac{\partial Q_g(\Psi)}{\partial \tilde{a}_{rj}^{(i)}} = \sum_{n=1}^{N}\sum_{m=1}^{K} (h_i^{(n)} - g_i^{(n)}) \frac{\partial f(O^{(n)}, \psi_i)}{\partial a_{rj}^{(i)}} \frac{\partial a_{rj}^{(i)}}{\partial \tilde{a}_{rj}^{(i)}}, \tag{31}$$

$$\frac{\partial Q_g(\Psi)}{\partial \tilde{b}_{mj}^{(i)}} = \sum_{n=1}^{N}\sum_{m=1}^{K} (h_i^{(n)} - g_i^{(n)}) \frac{\partial f(O^{(n)}, \psi_i)}{\partial b_{mj}^{(i)}} \frac{\partial b_{mj}^{(i)}}{\partial \tilde{b}_{mj}^{(i)}}. \tag{32}$$

Upon observing the gradients at the gate in Eq. 31 and remembering that $0 \leq h_i \leq 1$ and $0 \leq g_i \leq 1$, we see that the gate g_i would try to get closer to h_i (which is held constant at the M step). Therefore, the maximum Q_g is reached if both g_i and h_i are 1, and the others (g_j , h_j , $i \neq j$) are zero. This happens when a data sequence can be completely described by a single expert. Otherwise, the experts share a pattern and pay a price for it as described by the cross-entropy term in Eq. 16. Upon observing Eq. 26, we see that the HMM parameters at the experts are adjusted such that the expert output $\hat{y}_{ik}$ approximates the true class label y_k.

Learning is accomplished by computing the expectations h_i in the E step and learning the HMMs in the M step. The complete algorithm is given in Algorithm 4.1.

Algorithm 4.1: MHMME TRAINING(K, X, Y)

- Initialize the number of experts I
- Initialize the gating HMM parameters $\{\psi_i\}_{i=1}^{I}$
- Initialize the expert HMM parameters $\{\{\lambda_{ik}\}_{i=1}^{I}\}_{k=1}^{K}$

while $|Q(\Theta, \Theta^{(p-1)}) - Q(\Theta, \Theta^{(p)})| / Q(\Theta, \Theta^{(p)}) > 1e-5$

do
- **comment: E STEP**
- **do** Compute
  - Viterbi log likelihoods $f(O|\lambda_{ik})$ from Eq. 7
  - Expert outputs $\hat{y}_{ik}^{(n)}$ from Eq. 9
  - Expert probabilities $P_i(\mathbf{y})$ from Eq. 10
  - Gating outputs $g_i^{(n)}$ from Eq. 6
  - Posterior probabilities $h_i^{(n)}$ from Eq. 14
- **end**
- **comment: M STEP**
- **comment: Expert Updates**
- **while** Q_e in Eq. 17 is increasing (i.e. $\Delta Q_e > 1e-5$) for each expert

  do
  - Map $a_{rj} \rightarrow \widetilde{a}_{rj}$ and $b_{mj} \rightarrow \widetilde{b}_{mj}$ (Eqs. 20 & 22)
  - **do**
    - Update A from Eq. 24
    - Update B from Eq. 25
  - Map $\widetilde{a}_{rj} \rightarrow a_{rj}$ and $\widetilde{b}_{mj} \rightarrow b_{mj}$ (Eqs. 21 & 23)
- **comment: Gate Updates**
- **while** Q_g in Eq. 16 is increasing (i.e. $\Delta Q_g > 1e-5$)

  do
  - Map $a_{rj} \rightarrow \widetilde{a}_{rj}$ and $b_{mj} \rightarrow \widetilde{b}_{mj}$ (Eqs. 20 & 22)
  - **do**
    - Update A using the gradients in Eq. 31
    - Update B using the gradients in Eq. 32
  - Map $\widetilde{a}_{rj} \rightarrow a_{rj}$ and $\widetilde{b}_{mj} \rightarrow b_{mj}$ (Eqs. 21 & 23)
- Compute $Q(\Theta, \Theta^{(p)})$ from Eq. 15

- Compute $\hat{y}_k^{(n)}$ from Eq. 11
- Make a decision $k^* = \underset{k}{\text{argmax}} \ \{\hat{y}_k^{(n)}\}_{k=1}^{K}$.

return (k^*)

5. Synthetic Example

Illustrative synthetic data were created to examine the MHMME behaviour. In the sections below, we describe the data generation, MHMME initialisation and the results of MHMME.

5.1. Data Generation

A training set of 80 sequences from two classes was generated as follows. The interval 0, 1 was divided into 10 sub-intervals. For each sub-interval $i = 1, \ldots, 10$, a sample x_i was drawn from a uniform distribution on that sub-interval. Then,

- 20 sequences were generated from class 1 using $y_{in}^{(1)} = x_{in} + N(0, \sigma^2)$ where $n = 1, \ldots, 20$,

- 20 sequences were generated from class 1 using $y_{in}^{(1)} = -x_{in} + 1 + N(0, \sigma^2)$ where $n = 21, \ldots, 40$.
- 20 sequences were generated from class 2 using $y_{in}^{(2)} = x_{in}^2 + N(0, \sigma^2)$ where $n = 1, \ldots, 20$
- 20 sequences were generated from class 2 using $y_{in}^{(2)} = -x_{in}^2 + 1 + N(0, \sigma^2)$ where $n = 21, \ldots, 40$.

In all cases σ was set to 0.08. These sequences are displayed in Fig. 3. We refer to them as $x, -x, x^2$ and $-x^2$.

Using the same parameters and protocol, a test set was generated containing 40 samples per class. The y values were used as the data; the x values were ignored. Therefore, the features were 1D sequences as if the data were projected onto the y-axis. The values were discretized to five symbols, linearly spaced between 0, 1.

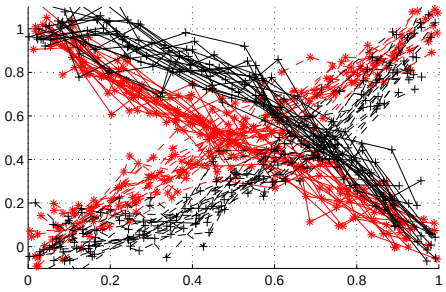


Fig. 3. *Synthetic data for two classes. The sequences that belong to the first class are displayed in red (∗), and the sequences that belong to the second class are displayed in black (+). The first class has* 20 *sequences generated from the function* $y = x + N(0, \sigma^2)$ *and* 20 *sequences generated from* $y = -x + 1 + N(0, \sigma^2)$ *overlayed on each other. Similarly, the second class has* 20 *sequences generated from the function* $y = x^2 + N(0, \sigma^2)$ *and* 20 *sequences generated from* $y = -x^2 + 1 + N(0, \sigma^2)$. *These y-values were quantized to* 5 *symbols and the discretized values were used as the data. Each data sequence has length* 10.

5.2. Initialization of the MHMME model

To discriminate the sequences, an MHMME model was trained with two experts. All HMMs in the MHMME model had three states and five symbols. To initialize the gate, the data were clustered by k-means [19] into two, and a Baum-Welch (BW) HMM [23] was fit to each of these clusters. To initialize the experts, the sequences from each class that were highly weighted by the gate were modeled with BW-HMM.

With this initialization, the first gate HMM immediately gave more weight to the $y = x$ and $y = x^2$ sequences, and the second gave more weight to the $y = -x + 1$ and $y = -x^2 + 1$ sequences.

5.3. Results of MHMME

Upon the MHMME training, the first expert learned HMM models that discriminate among $y = x$ and $y = x^2$, and the second expert learned HMM models that discriminate among $y = -x + 1$

and $y = -x^2 + 1$. The patterns learned by each HMM are shown in Fig. 4 where the bigger circles denote the contexts defined by the gate, and the smaller circles denote the results of classification by the experts. The A and B matrices of the HMM models at the gate and at the experts are displayed in Figs. 5 and 6 as Hinton diagrams in which the area occupied by a white square is proportional to the magnitude of the matrix entry. In Fig. 5, observe that the first HMM at the gate learns sequences with positive slope, and the second learns sequences with negative slope. Recall that the five symbols were linearly spaced between 0, 1. So the top row in the B matrices denotes the highest symbol, and the bottom row the lowest . These sequences are discriminated with expert HMM models as shown in Fig. 6. The B matrices suggest that each HMM is learning the sequence shapes with stress on their discriminative properties.

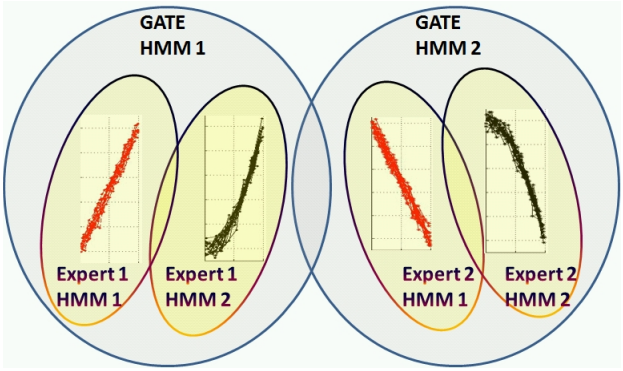


Fig. 4. *MHMME results on synthetic data. The first HMM in the gate learns a model for $y = x$ and $y = x^2$ and defines the context to be the positive slope. The second HMM in the gate learns a model for $y = -x$ and $y = -x^2$ and defines the context to be the negative slope. Then, the first expert learns to discriminate between $y = x$ and $y = x^2$, and the second expert learns HMM models to distinguish between $y = -x$ and $y = -x^2$. The HMM parameters that lead to this result are plotted in Fig. 5 for the gate and in Fig. 6 for the experts.*

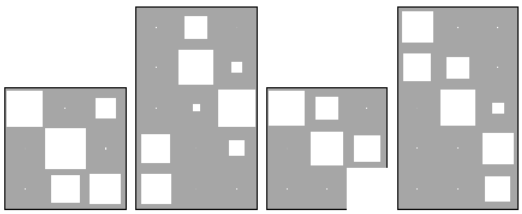


Fig. 5. *HMM models learned at the gate for the synthetic data. The transition matrices (A) and the observation matrices (B) are displayed as Hinton diagrams, in which the area occupied by a white square is proportional to the magnitude of the matrix entry. The first HMM model at the gate learns the sequences with a positive slope (x and x^2) as described by the A_1 matrix in (a) and the B_1 matrix in (b). The second HMM model at the gate learns the sequences with a negative slope ($-x$ and $-x^2$) as described by the A_2 matrix in (c) and the B_2 matrix in (d).*

The initial classification rates were 65% on the training data and 60% on the test data. After

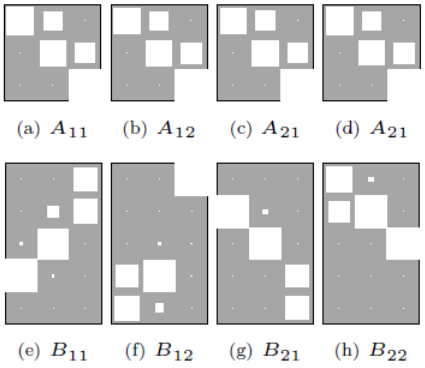


Fig. 6. *HMM models learned at the experts for synthetic data. A subscript ij means i^{th} expert j^{th} class. The transition matrices (A) and the observation matrices (B) are displayed as Hinton diagrams. There are only slight differences in the A matrices (a-d), but the figures (e)-(h) show what the B matrices have learned to model and discriminate the sequences of Fig. 3.*

MHMME learning, the classification rates reached 98% on the training data, and 94% on the test data. The improvement in the objective function with respect to the outer iterations is displayed in Fig. 7. With one outer iteration, we mean a complete E-M step where all the parameters in the experts and the gate are updated. Note that at each update of an HMM, there are several inner iterations that are run until convergence, as given in Algorithm 4.1.

The red dashed curve is the objective function of the gate, Q_g in Eq. 16, and the green dotted curve is the objective function of the experts, Q_e in Eq. 17. Q_g and Q_e are summed to get the total objective function Q in Eq. 15, which is displayed as the solid blue curve. The patterns in the iterations point to an interesting observation. In the first iteration, Q_g, the objective function of the gate stays the same, and the experts update relatively quickly. Then Q_e attains a smaller incline while Q_g shows a significant increase for the next two iterations. Finally, when the gate updates become almost constant, the experts keep adjusting themselves until both the experts and the gate reach a steady solution. With these adjustments at each iteration, the experts strive to best represent the sequences that are highly weighted by their corresponding gate.

It is noted that with different initializations or numbers of experts, the gate could partition the space differently as there are many solutions to this problem. In that case, one expert/gate combination finds meaningful patterns for data that received low confidences from the other experts. With this interpretation, one can compare it to AdaBoost [10]. However, there is at least one major difference: the experts and gate are learned and updated simultaneously, whereas the experts are learned successively in the original AdaBoost algorithm and there is no going back when an expert is learned.

6. Experimental Results with Landmine Data

A dataset of measurements collected with a wide-band electromagnetic induction (WEMI) sensor over regions of earth containing buried landmines and non-mine (clutter) objects is used for this application. The data were collected in two outdoor environments at which landmines and clutter objects were buried in the centers of cells in rectangular grids. The cells were 1.5 meters x 1.5 meters. Some cells, called blanks, had nothing buried in them. Table 1 tabulates the number of

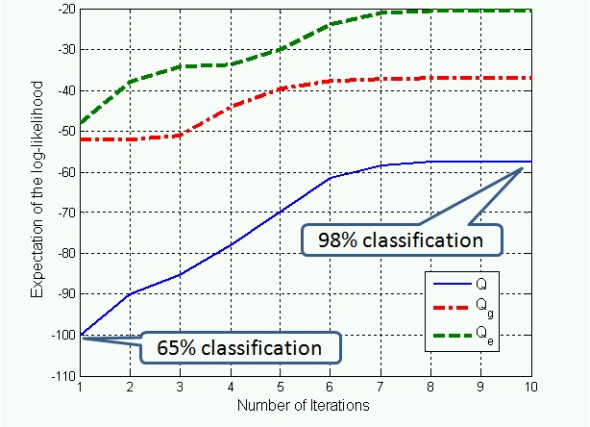


Fig. 7. *MHMME objective function. The objective functions for the gate (Q_g) and the experts (Q_e) versus the number of iterations. The total objective function Q displayed as the solid line is the sum of Q_g and Q_e.*

objects in both sites.

Table 1 Number of landmine and clutter objects

Notation	Meaning	Number
HMAP	High metal anti-personnel mine	30
LMAP	Low metal anti-personnel mine	66
HMAT	High metal anti-tank mine	11
LMAT	Low metal anti-tank mine	49
HMC	High metal clutter	89
MMC	Medium metal clutter	28
LMC	Low metal clutter	28
NMC or B	Non-metallic clutter or blank cell	142

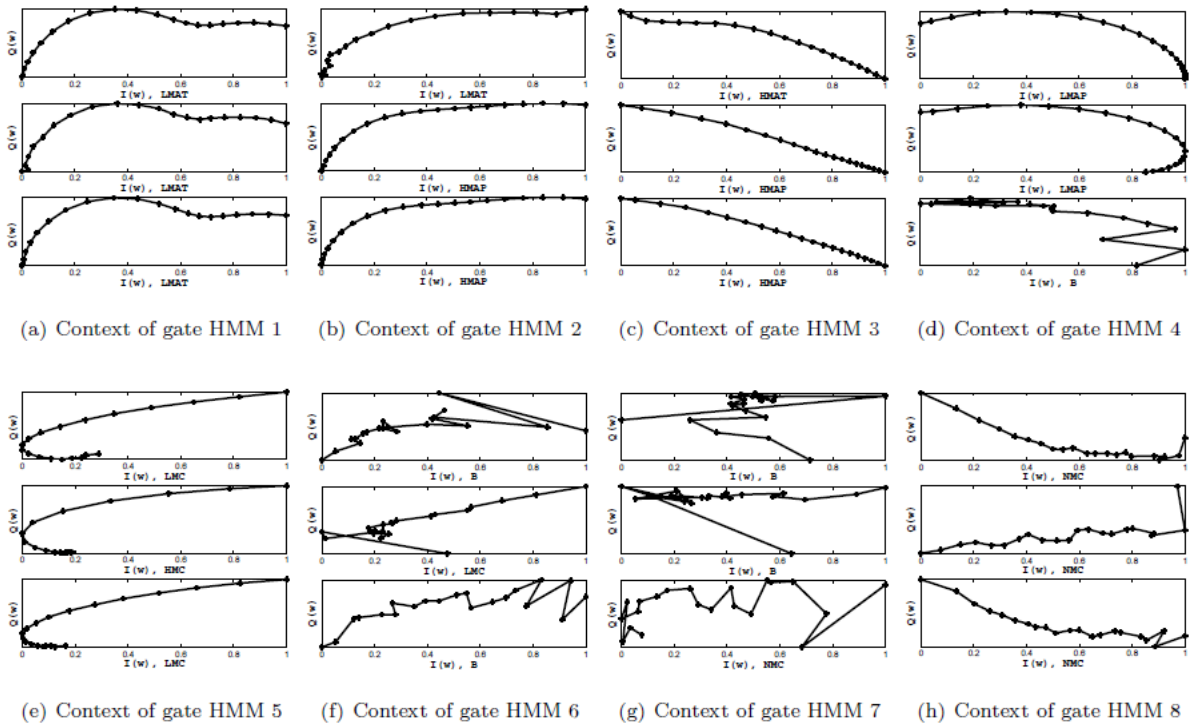
The sensors are described in detail in [9,27]. The WEMI sensors detect metal and produce characteristic signatures of many metallic objects. They collect complex responses in 21 frequencies between 330Hz and 90, 030Hz equally spaced in log space; In the configuration used to collect the data described here, data were collected at 1 cm intervals in the direction of travel (called the down-track direction) using three sensors aligned perpendicular to the direction of travel. A WEMI sensor response can be modeled as

$$S(w) = AI(w) + iQ(w). \tag{33}$$

where w is the frequency, A is the magnitude, $I(w)$ is the real (in-phase) response and $Q(w)$ is the imaginary (quadrature) response. This shape can be represented by the Argand diagram, that is, the plot of $I(w)$ with respect to $Q(w)$ [27]. The Argand diagram shape can characterize the type and distribution of metal in a target [9]. Mines of the same type can show similar Argand curves that are scaled versions of each other depending on depth. Thus, it is possible to discriminate between some landmines and clutter [11,24,35,38]. However, the extent of the ability to discriminate is not known at this time.

Example Argand diagrams are shown in Fig. 8. Small mines with small amounts of metal are generally buried close to the ground surface and generate a faint WEMI response that can be

confused with surface clutter. Large mines with small amounts of metal are often found at deeper depths and their WEMI response can also be faint. Furthermore, a small mine that is mostly metal can be buried deeply but appear similar to a small mine with a small amount of metal buried at a shallow depth. As a result, the features of these subclasses are interlaced and it is difficult to appoint a model as an expert to identify a particular subclass of mines [11, 25, 36]. The MHMME model offers promise for better discrimination between mines and clutter by identifying contexts.



(a) Context of gate HMM 1 (b) Context of gate HMM 2 (c) Context of gate HMM 3 (d) Context of gate HMM 4

(e) Context of gate HMM 5 (f) Context of gate HMM 6 (g) Context of gate HMM 7 (h) Context of gate HMM 8

Fig. 8. *Contexts defined by the gate HMMs. Each sequence is a normalized Argand diagram which is the plot of the real EMI response $I(\omega)$ vs. the imaginary EMI response $Q(\omega)$. On the x-axis, the type of the mine or nonmine object is given; such as LMAP, LMAT, HMAT, HMAP and so on. Each column shows the top three Argand sequences among all the training sequences that are assigned the highest weight by the gate HMM. For example, in (a) the first gate HMM learns the LMAT objects of a specific shape to be the first context. In (d), LMAP objects and a blank cell fall under the same context because of their similar shapes. It will now be the job of the experts to identify the intricate details to distinguish between the mine and non-mine sequences.*

The data measured by the middle sensor were used for analysis [24]. During training, pairs of in-phase and quadrature data $\{I(\omega), Q(\omega)\}$ were discretized to 50 cluster centers using fuzzy c-means (FCM) clustering [1]. An example is displayed in Fig. 9. The complex response collected at 21 frequencies are discretized using this clustering resulting in an observation sequence of length 21. These observation sequences are the sequences that the HMM processes.

6.1. Initialization

The MHMME architecture was set to have 8 experts based on our prior knowledge of Table 1. This also corresponds to 8 HMMs at the gate to be able to produce a weight for each expert; and 2

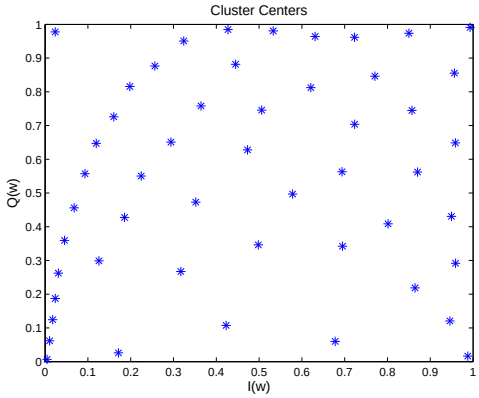


Fig. 9. *Cluster centers of the landmine data. In-phase and Quadrature components were used for quantization.*

HMMs at each expert. All the HMMs were set to have 3 states, determined experimentally.

The gate HMMs were initialized using clustering HMMs (Cl-HMM), which is the abbreviation for Smyth's sequence clustering approach [28]. A brief description of Cl-HMM is as follows: an HMM model is fit to each training sequence, resulting in N HMM models for N sequences. All N training sequences are tested on all N HMM models, and sequences with similar likelihoods are clustered together into G groups using hierarchical clustering. Then a new HMM is learned for each cluster using the Baum-Welch algorithm [23]. To initialize the gate, 4 HMMs were learned from class 1 (mines) and 4 HMMs were learned from class 2 (non-mines) using Cl-HMM. To initialize the experts, the sequences from each class that were highly weighted by the gate were modeled with BW-HMM.

6.2. Landmine Detection Results

A sample output of the MHMME classifier is given in Fig. 8. Each subfigure shows the sequences that received the highest probabilities by the gate HMMs. For example, the first HMM at the gate, as shown in Fig. 8(a), gave the highest probabilities to mainly LMAT objects. Similarly, LMAT and HMAP objects of a particular shape share the second context. This is an interesting and a very intuitive result. Low metal anti tank (LMAT) mines are bigger in size, but they have lower metallic content and they are buried in deeper levels of the ground. High metal anti personnel (HMAP) mines on the other hand are much smaller than the AT mines but are buried closer to the ground levels. Therefore, even if these are completely different objects, their metallic response may look the same. As a result, they fall into the same context, and instead of learning one model for HMAP mines and another for LMAT mines; it is more meaningful to automatically identify the similar ones and consider them under a single context.

On the other hand, the fourth gate HMM in Fig. 8(d) is also very interesting. Here, the gate HMM-4 has identified some LMAP mines and a blank (B) cell. There is nothing buried inside the blank cell; however, due to several environmental effects, the signals from the LMAP mine and the blank cell, which are completely different, may look alike in their argand diagrams. Therefore, they fall into the same context. Now that the gate has identified which sequences are similar; it is the experts' duty to discriminate between these sequences and come to a mine/non-mine decision.

The average classification results obtained from twenty experiments of 10-fold cross-validation

are reported in Table 2. In running these experiments, our goal is to fundamentally understand what is the experts' success rate, the gate's success rate, what is the contribution of the initialization on the overall success, and to understand if ME or HMM is the main identifying component and what would be the results if we used a discriminative classifier such as the MCE-HMM. In particular, we compare MHMME to the (i) Cl-HMM model used to initialize the MHMME, (ii) ME-only and HMM-only models including a discriminative HMM, and (iii) gate and experts when they are used individually as classifiers. Each of the classifiers that are compared are explained below.

- *Cl-HMM:* Sequences from each class are clustered into 4 using [28], and an HMM is learned for each cluster, resulting in 8 HMMs. A test sequence is assigned to the class whose HMM yields the highest log-likelihood.
- *Gate:* The gate HMMs of the full MHMME model are used as classifiers to test their individual performance. The first four HMMs are assumed to represent the first class, and the next four HMMs are assumed to represent the second class. A test sequence is assigned to the class corresponding to the HMM that yields the highest log-likelihood.
- *Experts:* Each expert HMM is used as a classifier.
- *PCA + ME:* The real and the imaginary parts of the data are combined to form a sequence of length 42. Then PCA is applied and the dimensionality is reduced to 10. These feature vectors are used to train a standard ME model.
- *MCE-HMM:* Minimum Classification Error HMM is a discriminative learning method that minimizes the total misclassification error. It was introduced by Juang et al. [18] and used in [14,41] for landmine detection. The parameters of MCE-HMM as they appear in [41] were set as follows: $\eta = 1$, $\gamma = 8$, $\theta = 0$, $\epsilon = 0.1$.

Table 2 Classification rates on the landmine data

Model	Mean	Std. Deviation
MHMME	0.80	0.05
MCE-HMM	0.75	0.05
PCA + ME	0.73	0.05
Gate	0.71	0.05
Cl-HMM	0.70	0.02
Experts	0.61	0.02

Classification rates are given in Table 2 in decreasing order. The mean and standard deviation of classification rates are computed from 20 independent training/testing runs, each of which employed a 10-fold cross-validation. When compared the other algorithms, MHMME performs far better than the HMM-only and the ME-only methods such as the MCE-HMM and the PCA+ME.

In addition, our goal was to understand what is gained by the full MHMME model beyond what can be achieved by the individual components. Upon observing the classification rates of the experts and the gate as if they were used as classifiers; MHMME significantly increases the classification rates beyond those obtained by the components. It is also interesting to compare the Gate and the Cl-HMM; although the gate was initialized with Cl-HMM, it did not necessarily increase the classification rates after training, rather it worked towards the goal of increasing the overall probability of the gate and experts combined in an MHMME. These results show that it is

not the experts or the gate alone, but rather it is their combination in the MHMME model that gives good classification rates.

To sum up, MHMME found contexts that are similar to what a human expert would find; such as one context for high metallic anti personnel (HMAP) mines, one context for low metal anti-personnel mines (LMAP) and so on. However, it also showed that some of the non-mines and landmines are very similar; and that an empty cell can also look very much like an LMAP. In such interesting cases, the MHMME model first groupd together these signals that look alike (ie. learn the context), and learn the experts to classify these data into mine/ non-mine decisions.

7. Experimental Results on the CP Dataset

In this section, we evaluate the behavior of MHMME on the chicken pieces (CP) dataset [13]. We describe the data in Sec. 7.1 and the MHMME initialization in Sec. 7.2. In Sec.7.3 we analyse the MHMME classifier and the characteristics of the trained experts and the gate, the likelihood distributions, and reliability and reject rates. We then analyze the internal structure of MHMME in Sec. 7.4 and evaluate the performance of each component of MHMME and how the components compare to using the ME or HMM models had they been combined in the MHMME way.

7.1. Data Set

The chicken pieces dataset contains 446 binary images of five classes of pieces of chicken. There are 117 images in the Wing, 96 images in the Drumstick, 76 images in the Back, 61 images in the Thigh and Back, and 96 images in the Breast classes. This dataset is publicly available at `http://algoval.essex.ac.uk/data/sequence/chicken/`. The features are fully described by Bicego et al. [2, 3]. Briefly, the binary image contours were approximated by line segments. The features were recorded as the angles between consecutive segments. Therefore, a data sample is a sequence of features from one image. The lengths of the sequences in the CP dataset are varying between the minimum length of 18 and the maximum length of 104. The mean length of the sequences is 54 and the median length is 51. The CP sequences exhibit significant variation within each class, making it a good test case for our model.

On another note, it should be stated the MHMME model was developed based on a need for landmine detection; and that there might be better models to discriminate the CP dataset such as the ones based on SVMs (also reported in our own paper in [35]). However, our goal in this section is to demonstrate the inner workings of the MHMME on a dataset everyone can freely access and quickly understand; and to show in detail what each and every component is learning, and how that is contributing to the overall analysis.

7.2. Initialization

The MHMME was initialized with 10 experts with 3 states and 20 symbols per HMM. The number of states was selected from published results [4, 8] and was confirmed experimentally. The number of symbols was selected by minimizing misclassification rates for a basic HMM classifier. The symbols were found by clustering the data with fuzzy c-means (FCM) [1]. The number of experts was selected based on achieving the highest classification rates with the Cl-HMM classifier, which was described in Sec.6.1. The Cl-HMMs achieved the best classification results for $G = 2$ clusters for each class of the CP data. Therefore, we took $G = 2$ clusters, and initialized the 10 gate HMMs with the CI-HMMs .

Sequences that produced the highest log-likelihoods from the 10 gate HMMs were used to train expert HMMs using the Baum-Welch algorithm. Thus, contexts are initially designed to be associated with classes, however, during MHMME training, contexts get updated to increase the overall probability that is defined by both gate and experts, and need not represent a specific class. Within each context, it is the expert's duty to learn models that discriminate the 5 classes.

7.3. Analysis of MHMME Classifier

Two-fold cross-validation was used for MHMME training for comparison to previous work [4, 8, 22, 26]. The contexts defined by the gates are represented in Fig. 10. Each column shows two sequences that represent a learned context. The sequences have fairly variable shape and are not class specific; they may be shared between classes. This observation is illustrated in Table 3 where each sequence has been assigned to the gate HMM that produced the highest log-likelihood. The first column shows that most sequences from the class 1 are represented by the first and the second gate HMMs, but other sequences from class 1 were better represented by the $4-7$th gate HMMs. From another perspective, the 5th gate HMM (5th row) represents at least one sequence from every class. The same applies to the 4th gate HMM (4th row). Therefore, the gates are not sufficient for high performance classification, and the experts are needed. Note that all assignments are actually probabilistic, and there is no such hard clustering of data during training. The hard assignments have been provided for evaluation.

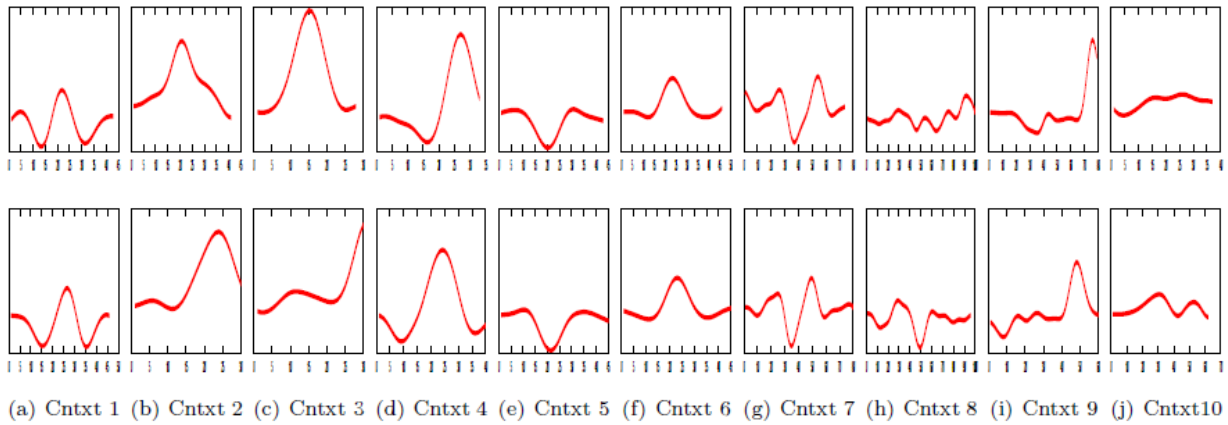


Fig. 10. *Graphical depiction of the gates for each learned context. Each context has a gate HMM associated with it. These HMMs were run on all the* 223 *training sequences. The HMMs produced a likelihood value for each sequence. For each context, the two sequences that were assigned the two highest likelihood values by the gate HMM for that context are shown in the figure. The y-axis is in the range* $-0.35,\ 0.93$ *and shows the amplitude of the features.*

Classification rates from combining expert and gating models are shown Table 4. Note that the misclassification rate is significantly reduced. These class assignments are probabilistic and the outputs can be thresholded. A more thorough look given by reliability and reject rates as shown in Fig. 11 can measure this characteristic. They are defined as follows: let th be a threshold on classifier confidence. Let N_{th} denote the number of samples, x, with confidence $C(x)$ such that $C(x) \geq th$. Let C_{th} denote the number of correctly classified samples with $C(x) \geq th$. The reliability R of the classifier at threshold th is defined as $R(th) = C_{th}/N_{th}$, and the reject rate

$J(th) = (N - N_{th})/N$ where N is the total number of samples. Fig. 11 shows that with no rejection, about 18% of the samples are misclassified. However, if we reject 40% of the samples, this corresponds to the threshold value of 0.6, and then only about 4% from the remaining 60% are misclassified. Thus, about 60% of the patterns are easily classified. Also, some patterns are ambiguous so completely accurate classification is unrealistic.

Table 3 Gate results

		Class				
		1	2	3	4	5
Gate	1	10	0	0	0	0
	2	29	1	1	3	0
	3	0	17	0	1	3
	4	9	9	1	4	6
	5	1	5	21	1	11
	6	1	3	24	0	7
	7	9	4	0	19	11
	8	0	0	0	4	3
	9	0	0	0	0	4
	10	0	0	2	0	3

Table 4 Confusion Matrix

	Class				
Decision	56	3	2	10	4
	0	26	1	0	2
	3	4	46	0	2
	0	0	0	15	0
	0	6	0	7	40

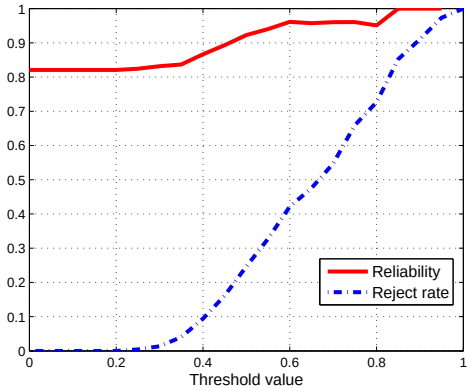


Fig. 11. *Reliability and rejection rates.*

The HMM log-likelihoods at the gate and the experts for the training data of a given class are shown in Fig. 12. The x-axis shows the difference of the log-likelihoods between two experts. The

y-axis shows the log-likelihoods obtained from the gate HMMs. In this plot, every two gate HMMs were assumed to specialize in one class (HMMs 1&2 describe class 1, HMMs 3&4 describe class 2 and so on), which is consistent with the initialization process. The gate HMMs define a context, and the expert HMMs specialize within these contexts. For each class, at least one gate associated with each class produces a "high" log-likelihood for almost all samples (where high here means above about -3). As a result, in most classes, the experts and the gate complement each other resulting in the butterfly effect: if a gate/expert pair performs poorly in a region of the space, the other gate/expert pair performs better and dominates the classification decision. The effect is more pronounced in classes 2 and 5. For example, in the case of class 2, if gate 3 is high, then expert 3 is high and if gate 4 is high, then expert 4 is high. To be clear, the gate HMMs are not designed to be classifiers; they are designed to model contexts. On the other hand, it is interesting to note that the gates do contain classification information.

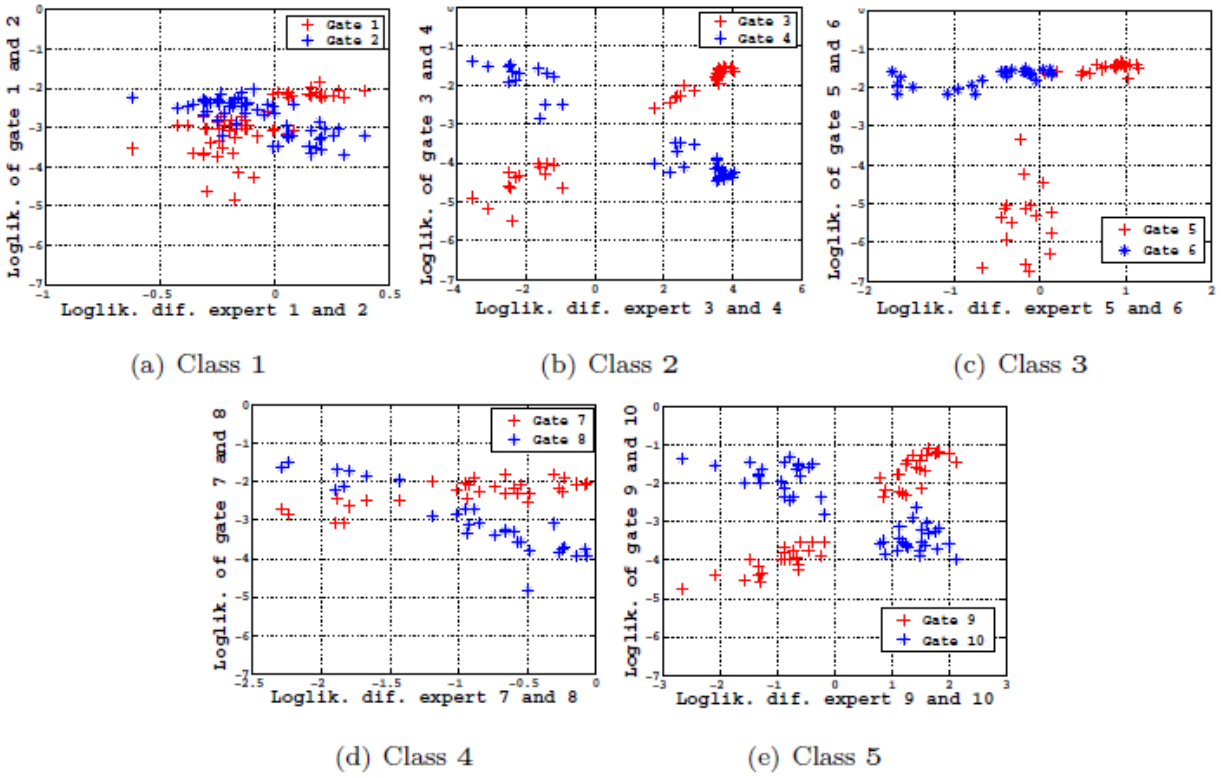


Fig. 12. *The log-likelihoods of sequences of a given class at the gate and the experts. The x-axis shows the difference of the log-likelihoods between two experts. The y-axis shows the log-likelihoods of the gate HMMs.*

7.4. Analysis of internal structure of MHMME

In this section, we compare MHMME to the components of MHMME as well as to the computing models. Similar to the previous section, our goal is to understand what is gained by the full MHMME model beyond what can be achieved by the individual components. In particular, we compare MHMME to the (i) Cl-HMM model used to initialize the MHMME, (ii) ME-only and

HMM-only models, and (iii) gate and experts when they are used individually as classifiers. The methods for using components as classifiers are briefly described below.

- *PCA+ME:* To use the standard ME, the vectors in each sequence are concatenated to form a high-dimensional vector. Since sequences have variable lengths, they are resampled to length 110. Principal component analysis (PCA) is used to reduce dimensionality to 10. These feature vectors are used to train a standard ME model [16, 17].
- *PCA+VMEC:* The same as above except variational ME is used for classification (VMEC) [37] with γ hyperparameters set to 1.

  In the next four classifiers, a test sequence is assigned to the class corresponding to the HMM that yields the highest log-likelihood.
- *HMM:* Baum-Welch is used to train one HMM per class.
- *Cl-HMM:* This method was described in Sec. 6.1. Two HMMs are constructed for each class.
- *Gate:* The gate HMMs of the full MHMME model are used as classifiers to test their individual performance. Every two gate HMMs are assumed to describe a class (HMMs 1&2 describe the first class, HMMs 3&4 describe the second class and so on).
- *Experts:* Each expert HMM is used as a classifier.
- *HMM + MAP* [4]: One HMM is learned per class. Then classification of an unlabeled sequence is performed by maximum-a-posteriori (MAP) approach.

Classification rates are given in Table 5. The mean and standard deviation of 20 classification rates calculated from 20 independent training runs, each of which employed a 2-fold cross-validation. Note that there are 5 classes so randomly assigning samples to classes should yield an average classification rate of 20%. Among these classifiers, PCA+ME and PCA+VMEC are the two ME-only methods, whereas HMM and Cl-HMM are the two HMM-only methods. When compared to these, MHMME performs far better than the HMM-only and the ME-only methods.

The results are consistent with those of the landmine data. First, MHMME significantly increases the classification rates beyond those obtained by the components. Second, when the Gate and Cl-HMM rates are compared, it can be observed that the gate worked towards the goal of increasing the overall probability of the gate and experts combined in an MHMME. These results again show that it is not the experts or the gate alone, but rather it is their combination in the MHMME model that gives good classification rates.

Table 5 Classification rates on the CP dataset for 2-fold cross-validation training on 20 runs

Model	Mean	Std. Dev.
PCA + ME [17]	0.43	0.01
PCA + VMEC [37]	0.44	0.01
HMM	0.41	0.05
Cl-HMM [28]	0.61	0.03
HMM + MAP [4]	0.57	0.008
Gate	0.59	0.08
Experts	0.53	0.04
MHMME	**0.73**	**0.02**

These results indicate that MHMME is useful for datasets that have multiple contexts that are interlaced between classes. It allows the simultaneous probabilistic learning of the sub-regions from multi-class data and the discriminative classification of the data in these sub-regions. The soft partitioning is provided by the gate whereas the discriminative classification is performed at the experts. One direct consequence of soft partitioning of the data should be emphasized: HMMs at each expert are affected by all the data points, but the effect of each data point is weighted by the gate. Therefore, even if a sequence does not have a high weight as determined by the gate, it still affects the experts' decision but with a lower weight. In this way, HMMs are less prone to over-fitting than other models that use hard clusters of the data while specializing in a context.

8. Conclusion

In this study, we addressed the problems encountered when designing classifiers for classes that contain multiple subclasses whose characteristics are dependent on the context. It is sometimes the case that when the appropriate context is chosen, classification is relatively easy, whereas in the absence of contextual information, classification may be difficult. Therefore, in this study, simultaneous learning of context and classification has been addressed for sequential data, and the mixture of hidden Markov model experts has been developed. The updates of HMM parameters in an ME framework have been derived, and the benefits of ME have been extended to time-series data. The MHMME model allows for the simultaneous probabilistic learning of the sub-regions from multi-class sequential data and the discrimination of the classes in these sub-regions. The output is a mixture of the HMM decisions, but the mixture coefficient is not fixed once it is learned, rather it depends on the input data. The MHMME model has been applied to a synthetic dataset, to the CP data as well as the landmines data. It has been shown that the combination of ME and HMM models in the MHMME model increases the performance of any single classifier. When compared to its individual components, i.e. the HMMs at the experts and at the gate, MHMME combination increases the classification rates. In addition, it has been shown that MHMME can do well in comparison to competing models.

In this work, the number of experts and the number of states were selected experimentally such that the initialization starts at a higher rate for the landmine data. For the CP dataset, it was selected based on the literature. In the future, it would be worthwhile to investigate the sampling methods for training as opposed to EM, the optimum number of experts and the optimum number of states. In addition, it would be interesting to see whether or not another level of hierarchy would increase the classification rates for data that have a deeper level of contexts. Also, the MHMME model herein uses discrete HMMs, but it would be worthwhile to investigate the update equations using continuous HMMs.

9. References

[1] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[2] BICEGO, M., CRISTANI, M., MURINO, V., PEKALSKA, E., AND DUIN, R. P. W. Clustering-based construction of hidden Markov models for generative kernels. In *EMMCVPR* (2009), pp. 466–479.

[3] BICEGO, M., MURINO, V., AND FIGUEIREDO, M. A. T. Similarity-based classification of sequences using hidden Markov models. *Pattern Recogn. 37*, 12 (2004), 2281–2291.

[4] BICEGO, M., PEKALSKA, E., TAX, D. M. J., AND DUIN, R. P. W. Component-based discriminative classification for hidden Markov models. *Pattern Recogn. 42*, 11 (2009), 2637–2648.

[5] BUREAU OF POLITICAL-MILITARY AFFAIRS. Hidden killers: The global landmine crisis. report 10575, Office of Humanitarian Demining Programs, United States Department of State, Sep 1998.

[6] CHEN, K., XIE, D., AND CHI, H. A modified HME architecture for text-dependent speaker identification. *IEEE Transactions on Neural Networks 7* (1996), 1309–1313.

[7] COELHO, A., LIMA, C., AND VON ZUBEN, F. Hybrid genetic training of gated mixtures of experts for nonlinear time series forecasting. In *IEEE Int. Conf. on Systems, Man and Cybernetics* (2003), vol. 5, pp. 4625–4630.

[8] DALIRI, M. R., AND TORRE, V. Robust symbolic representation for shape recognition and retrieval. *Pattern Recogn. 41* (May 2008), 1799–1815.

[9] FAILS, E. B., TORRIONE, P. A., WAYMOND R. SCOTT, J., AND COLLINS, L. M. Performance of a four parameter model for modeling landmine signatures in frequency domain wideband electromagnetic induction detection systems. In *SPIE Detection and Remediation Technologies for Mines and Minelike Targets XII* (2007), pp. 65531–7.

[10] FREUND, Y., AND SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 1 (August 1997), 119–139.

[11] FRIGUI, H., ZHANG, L., AND GADER, P. Context-dependent multisensor fusion and its application to land mine detection. *IEEE Trans. Geoscience and Remote Sensing 48*, 6 (June 2010), 2528–2543.

[12] FRITSCH, J., FINKE, M., AND WAIBEL, A. Context dependent hybrid HME HMM speech recognition using polyphone clustering decision trees. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1997), pp. 1759–1762.

[13] G. ANDREU, A. CRESPO, J. V. Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition,. In *International Conference on Neural Networks* (1997), vol. 2, pp. 1341–1346.

[14] HAMDI, A., MISSAOUI, O., FRIGUI, H., AND GADER, P. Landmine detection using ensemble discrete hidden Markov models with context dependent training methods. In *Proc. SPIE* (2010), p. 76642J.

[15] JACOBS, R. A., JORDAN, M. I., NOWLAN, S. J., AND HINTON, G. E. Adaptive mixtures of local experts. *Neural Comput. 3*, 1 (1991), 79–87.

[16] JORDAN, M. I. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6* (1994), 181–214.

[17] JORDAN, M. I., AND XU, L. Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks 8* (1995), 1409–1431.

[18] JUANG, B.-H., HOU, W., AND LEE, C.-H. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing 5*, 3 (May 1997), 257–265.

[19] KANUNGO, T., MOUNT, D. M., NETANYAHU, N. S., PIATKO, C. D., SILVERMAN, R., AND WU, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell. 24* (July 2002), 881–892.

[20] LU, Z. A regularized minimum cross-entropy algorithm on mixtures of experts for time series prediction and curve detection. *Pattern Recognit. Lett. 27*, 9 (2006), 947–955.

[21] MISSAOUI, O., FRIGUI, H., AND GADER, P. Land-mine detection with ground-penetrating radar using multistream discrete hidden Markov models. *IEEE Trans. Geosci. Remote Sens. 49*, 6 (2011), 2080–2099.

[22] NEUHAUS, M., AND BUNKE, H. Edit distance-based kernel functions for structural pattern classification. *Pattern Recogn. 39* (October 2006), 1852–1863.

[23] RABINER, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE* (1989), pp. 257–286.

[24] RAMACHANDRAN, G., GADER, P., AND WILSON, J. GRANMA: Gradient angle model algorithm on wideband EMI data for land-mine detection. *IEEE Geosci. Remote Sens. Letters 7*, 3 (July 2010), 535–539.

[25] RATTO, C., TORRIONE, P., MORTON, K., AND COLLINS, L. Context-dependent landmine detection with ground-penetrating radar using a hidden Markov context model. In *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)* (July 2010), pp. 4192–4195.

[26] SCHOLKOPF, B., AND SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[27] SCOTT, W. Broadband array of electromagnetic induction sensors for detecting buried landmines. In *IEEE Int. Geoscience and Remote Sensing Symp.(IGARSS)* (July 2008), vol. 2, pp. 375–378.

[28] SMYTH, P. Clustering sequences with hidden Markov models. In *Advances in Neural Inf. Proc. Systems (NIPS)* (1997), pp. 648–654.

[29] TUERK, A. *The state based mixture of experts HMM with applications to the recognition of spontaneous speech*. PhD thesis, University of Cambridge, September 2001.

[30] WANG, X., WHIGHAM, P., DENG, D., AND PURVIS, M. Time-line hidden Markov experts for time series prediction. *Neural Information Processing - Letters and Reviews 3*, 2 (May 2004), 39–48.

[31] WATERHOUSE, S., AND ROBINSON, A. Classification using hierarchical mixtures of experts. In *Proc. IEEE Workshop on Neural Networks for Signal Processing IV* (1994), pp. 177–186.

[32] WEIGEND, A., AND GERSHENFELD, N., Eds. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, 1994.

[33] WEIGEND, A. S., MANGEAS, M., AND SRIVASTAVA, A. N. Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6 (1995), 373–399.

[34] YAO, B., WALTHER, D., BECK, D., AND FEI-FEI, L. Hierarchical mixture of classification experts uncovers interactions between brain regions. In *Advances in Neural Inf. Proc. Systems (NIPS) 22*. 2009, pp. 2178–2186.

[35] YUKSEL, S., AND GADER, P. Mixture of hmm experts with applications to landmine detection. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International* (July 2012), pp. 6852–6855.

[36] YUKSEL, S. E., BOLTON, J., AND GADER, P. D. Multiple-Instance Hidden Markov Models With Applications to Landmine Detection. *Geoscience and Remote Sensing, IEEE Transactions on 53*, 12 (2015), 6766 – 6775.

[37] YUKSEL, S. E., AND GADER, P. Variational mixture of experts for classification with applications to landmine detection. In *International Conference on Pattern Recognition (ICPR)* (2010), pp. 2981–2984.

[38] YUKSEL, S. E., RAMACHANDRAN, G., GADER, P., WILSON, J., HO, D., AND HEO, G. Hierarchical methods for landmine detection with wideband electro-magnetic induction and ground penetrating radar multi-sensor systems. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (July 2008), vol. 2, pp. II–177–II–180.

[39] YUKSEL, S. E., WILSON, J. N., AND GADER, P. D. Twenty years of mixture of experts. *Neural Networks and Learning Systems, IEEE Transactions on 23*, 8 (2012), 1177–1193.

[40] YUMLU, M. S., GURGEN, F. S., , AND OKAY, N. Financial time series prediction using mixture of experts. In *18th Int. Symp. on Computer and information sciences (ISCIS)* (2003), vol. 2869, pp. 553–560.

[41] ZHAO, Y., GADER, P., CHEN, P., AND ZHANG, Y. Training dhmms of mine and clutter to minimize landmine detection errors. *IEEE Trans. Geosciences and Remote Sensing 41*, 5 (May 2003), 1016–1024.

[42] ZHAO, Y., SCHWARTZ, R., SROKA, J., AND MAKHOUL, J. Hierarchical mixtures of experts methodology applied to continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1995), vol. 5, pp. 3443–3446.