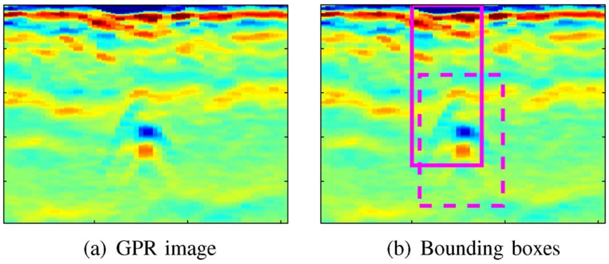
Multiple-Instance Hidden Markov Models With Applications to Landmine Detection

Seniha Esen Yuksel, *Member, IEEE*, Jeremy Bolton, *Member, IEEE*, and Paul Gader, *Fellow, IEEE*

***Abstract*—A novel multiple-instance hidden Markov model (MI-HMM) is introduced for classification of time-series data, and its training is developed using stochastic expectation maximization. The MI-HMM provides a single statistical form to learn the parameters of an HMM in a multiple-instance learning framework without introducing any additional parameters. The efficacy of the model is shown both on synthetic data and on a real landmine data set. Experiments on both the synthetic data and the landmine data set show that an MI-HMM can 1) achieve statistically significant performance gains when compared with the best existing HMM for the landmine detection problem, 2) eliminate the ad hoc approaches in training set selection, and 3) introduce a principled way to work with ambiguous time-series data.**

***Index Terms*—Expectation maximization (EM), ground penetrating radar (GPR), hidden Markov models (HMMs), landmine detection, multiple-instance HMM (MI-HMM), multiple-instance learning (MIL), stochastic EM, time-series data.**



(a) GPR image (b) Bounding boxes

Fig. 1. GPR data with a landmine signature in (a). Other signals in this image are the ground bounce at the top layer and the GPR echo from the differences in soil properties and from the clutter objects around the landmine. Traditionally, a bounding box is placed on the landmine signature, and training sets are formed from these signatures. However, placing a bounding box on the landmine signatures can be erroneous. Two alternative boxes are shown in (b) with solid and dashed rectangles.

I. Introduction

In standard learning techniques, an algorithm is typically presented with training samples from some number of classes, and its goal is to construct a characterization for each class. However, in some learning situations, class labels are not readily available for each sample in the training data. For example, in content-based image classification, an image may contain multiple objects, but it might not be easy to identify which of these objects are the relevant ones [1]–[3]. This type of data is also known as ambiguous data, and learning from ambiguous data remains a hard problem [4]–[10]. One of the areas where ambiguous data are encountered is landmine detection using ground penetrating radar (GPR). In radar images produced by GPR sensors, there are areas (subimages or feature sets) in an image that contain a target and areas that do not. One such example is shown in Fig. 1(a), where the signature in the middle indicates a landmine. However, there are other signals in this image such as the ground bounce and the GPR echo. Ground bounce is the reflections from the air–ground interface. These reflections can be very strong and can dominate the returns from the buried objects. GPR echo is the reflections from the media being sensed, and it can result from the clutter objects or from the changes in the soil itself. Both the ground bounce and the GPR echo from clutter can be thought of as subimages that require a label to indicate that they are not the landmine signatures. However, ground truth is provided only per image (which includes both the target and the artifacts) and not for each of these subimages. Therefore, this learning scenario provides one class label for multiple instances (a set of features), but it is ambiguous which of these instances is actually responsible for the landmine.

To combat this issue, researchers may segment the images manually or semiautomatically to extract target and nontarget exemplars for training [11]–[17]. An example is shown in Fig. 1(b) on a GPR image where two alternatives for a bounding box are plotted on the landmine signatures. However, this is not only an arduous task but also prone to errors resulting from GPR echoes and ground-truth errors, and furthermore, ambiguity still remains. Therefore, rather than struggling against the ambiguous nature of this learning problem, it may be best to use a model that explicitly accounts for ambiguous data.

One solution to learning from ambiguous data is multiple-instance learning (MIL). In the MIL scenario, class labels of all of the training data are not available; thus, it is not possible to present an algorithm with exemplar samples from each class [18]–[21]. Instead, an algorithm is presented with a collection of bags, or sets of samples, that are labeled positive or negative. Bags are labeled positive if there exists at least one sample that induces a target concept and are labeled negative if every sample is from the nontarget class. This view is illustrated in

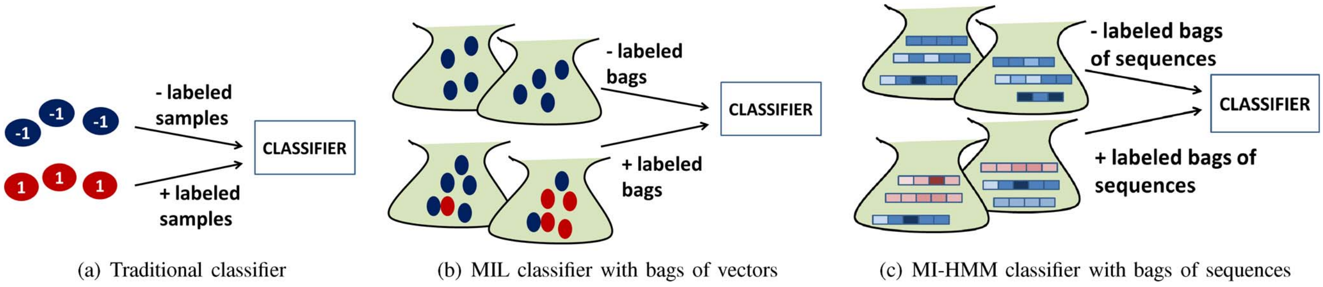
Manuscript received July 6, 2013; revised July 12, 2014 and March 12, 2015; accepted June 8, 2015. This work was supported in part by the Army Research Office under Grant W911NF0510067 and in part by the National Science Foundation under Grant 0730484.

S. E. Yuksel is with the Department of Electrical and Electronics Engineering, Hacettepe University, Ankara 06800, Turkey (e-mail: eyuksel@ee.hacettepe.edu.tr).

J. Bolton and P. D. Gader are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: jbolton@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/TGRS.2015.2447576



(a) Traditional classifier (b) MIL classifier with bags of vectors (c) MI-HMM classifier with bags of sequences

Fig. 2. Traditional MIL and MI-HMM classifiers. (a) Traditional classifier with labeled samples. In traditional supervised learning algorithms, a label is attached to each training sample, and the classifier is trained with these labeled samples. (b) In MIL, training class labels are attached to bags. A bag is a set of samples, and the samples within each bag are called instances. A bag is labeled positive if and only if at least one of its instances is positive; otherwise, it is labeled negative [18]. The bags on the right contain at least one red (positive) sample, which makes them positively labeled. (c) In MI-HMM, a bag is a set of sequences. These sequences can be of different lengths. A bag is labeled positive if and only if at least one of its sequences is positive. The bags on the right contain at least one red (positive) sequence, which makes them positively labeled.

Fig. 2, where the traditional classifiers are compared with MIL classifiers.

The MIL model has also been recently used in landmine detection to eliminate the problems associated with the bounding-box approach and other ad hoc methods and has shown considerable success [22], [23]. However, neither these studies nor the other MIL models in the literature could utilize time-series data. On the other hand, hidden Markov model (HMM)-based algorithms that utilize time-series data are known to be very useful in landmine detection [24]–[28]. Therefore, in this study, a novel multiple-instance HMM (MI-HMM) that uses MIL for time-series data is developed.

In MI-HMM, labels are attached to bags as in an MIL, but a bag is a set of sequences, and these sequences can be of different lengths, as shown in Fig. 2(c). The MI-HMM provides an elegant and simple way to learn the parameters of an HMM with a stochastic expectation maximization (EM)-type algorithm based on sampling that rejects or accepts the parameters using the MIL algorithm. Experiments on both the landmine data set and on synthetic data have shown that MI-HMM learning is very effective and outperforms MIL-only and HMM-only learning models as well as the state-of-the-art models that are implemented in real-time operational systems for landmine detection.

The novel contributions in this paper can be summarized as follows.

- The MI-HMM model introduced herein provides a simple solution to extend the MIL framework to capture the temporal properties of ambiguous time-series data. To the best of our knowledge, there is no other study that extends MIL learning for HMMs and does it in a single statistical form.
- A novel optimization scheme is introduced for the MI-HMM, which is accomplished with stochastic EM based on sampling. The use of sampling has overcome the well-known challenges of the noisy-OR formulation of the MIL model. Furthermore, the new optimization scheme provides a new way to optimize HMMs, which is intuitive and easy to integrate into an existing HMM code.
- In landmine detection, the results have shown significant improvement over the benchmark models that have been extensively tested and implemented in real time in several operational systems [11], [12], [14], [15], [24]–[27], [29]–[33]. The MI-HMM model has eliminated the manual and ad hoc preprocessing procedures and provided a systematic and automatic approach for the training of the classifiers. Furthermore, the MI-HMM model has half the parameters of that in the benchmark systems.

In the remainder of this paper, first, standard MIL learning is described in Section II. Then, notation for HMMs is introduced in Section III. Next, the MI-HMM model is proposed in Section IV, and its working principles are analyzed in Section V on synthetic data. Finally, the MI-HMM results on landmine GPR data are presented and discussed in Section VI.

II. Standard MIL

Let $\mathbf{x}$ denote a feature vector and $Y_{\mathbf{x}}$ denote the label of this vector. In the MI scenario, a learner is presented with N sets (bags) of K vectors or samples. For purposes of learning, a set, i.e., $X \subset \mathscr{R}^d$, is labeled target $(Y_X = 1)$ if there exists at least one target sample within the set. A set X is labeled negative $(Y_X = 0)$ if all constituent samples are nontarget. That is, $\exists \mathbf{x} \in X : Y_{\mathbf{x}} = 1 \Rightarrow Y_X = 1$ and $\forall \mathbf{x} \in X : Y_{\mathbf{x}} = 0 \Rightarrow Y_X = 0$. With this learning paradigm, the idea of uncertainty is incorporated using the set (or bag) structure, and learning the target concept from these bags of samples is called the MIL problem.

Maron *et al.* developed the diverse density (DD) [18] approach, which provides a statistical solution to the MIL problem based on Pearl's noisy OR-gate model [34]. Most MIL solutions adopt this noisy OR-gate model, which assumes that only one target sample within a bag is necessary and sufficient for a bag to induce a target concept. In standard DD approaches, a target concept f is learned given a collection of positively labeled bags, i.e., $\mathbf{B}^+$, and a collection of negatively labeled bags, i.e., $\mathbf{B}^-$. Assuming that observed sets X are independent, the target concept, i.e., f, is chosen to maximize the expression in

$$\hat{f} = \arg\max_f \prod_{X \in \mathbf{B}^+} P(f|X) \prod_{X \in \mathbf{B}^-} P(\neg f|X) \qquad (1)$$

where $\hat{f}$ is the desired target concept, and $\neg f$ are the samples that are not the targets [18]. Assuming a noisy OR-gate model [34], the posterior probability factors in (1) can be calculated in terms of the constituent samples in each bag $(x \in X)$ as follows:

$$P(f|X) = 1 - \prod_{\mathbf{x} \in X} (1 - P(f|\mathbf{x})) \tag{2}$$

$$P(\neg f|X) = \prod_{\mathbf{x} \in X} (1 - P(f|\mathbf{x})) . \tag{3}$$

In (1), the idea is to increase the probability of the target concept in the positive bag and to increase the probability of the nontarget concepts in the negative bags. With the noisy-OR assumption, in (2) and (3), the right-hand side of the equations has been described solely in terms of the target concept f.

III. HMMs

A hidden Markov model (HMM) is a very popular tool to represent time-series data. HMMs have been widely used in temporal pattern recognition for various applications, including speech, handwriting, and landmine recognition. Here, we will only provide the very basics and notations for HMM as they are used in Section IV. The notation used for HMMs is as follows.

- $W =$ number of states.
- $M =$ number of symbols in the codebook.
- $T_{\check{x}}$ is the length of an observation sequence $\check{x}$ and will be denoted as T from now on for simplicity.
- $V = \{v_1, \ldots, v_M\}$ the discrete set of observation symbols.
- $\check{x} = x_1 x_2, \ldots, x_t, \ldots, x_T$ denotes an observation sequence, where $x_t \in V$ is the observation at time t.
- $Q = q_1 q_2, \ldots, q_T$ is a fixed-state sequence, where q_t is the state at time t.
- $S = \{S_1, S_2, \ldots, S_W\}$ are the individual states.
- $\Theta = \{\boldsymbol{\pi}, A, B\}$ is the compact notation for an HMM model.
- The initial state distribution vector $\boldsymbol{\pi} = \{\pi_r\}_{r=1}^{W}$, where $\pi_r = P(q_1 = S_r)$ is the probability of being in state r at time $t = 1$.
- The state transition probability matrix $A = \{\{a_{rj}\}_{r=1}^{W}\}_{j=1}^{W}$, 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 matrix $B = \{\{b_j(m)\}_{j=1}^{W}\}_{m=1}^{M}$, where $b_j(m) = P(v_m \mathtt{at} t | q_t = j)$ is the probability of observing the symbol v_m, given that we are in state j.

Given an HMM model, the probability of a sequence is computed as

$$P_{HMM}(\check{x}|\Theta) = \sum_{Q} \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}} b_{q_t}(x_t). \tag{4}$$

The parameters of the standard HMM model are typically optimized generatively by the maximum-likelihood (ML) criterion [35]–[37] or discriminatively by the minimum classification error (MCE) criterion [25], [38], [39]. The ML criterion, when implemented by the EM algorithm, leads to a local optimum in the parameter space. Similarly, MCE models are generally learned using gradient-based approaches that also converge to a local minimum. One possibility to get around this problem is to use Markov chain Monte Carlo sampling methods to estimate the parameters of an HMM [40]–[44].

IV. MI-HMM: Learning From Bags of Sequences

MI-HMM is a tool that permits the learning of sequence models under the MI learning scenario. In Section IV-A, we formulate the MI-HMM, which is a discriminative model. Learning the parameters in a discriminative HMM model is generally difficult, and using gradient-based approaches is commonly subject to learning locally optimal parameter sets [25], [38], [39]. Furthermore, due to the noisy-OR formulation of MIL models, standard optimization methods will not yield a closed-form solution. Therefore, in Section IV-B, we describe a sampling scheme to update the parameters of MI-HMM.

A. Formulation

Assume that $\check{x} = x_1 x_2, \ldots, x_t, \ldots, x_T$ is a sequence and that each observation x_t in the sequence is a symbol from a codebook. A bag X is a set of sequences, and it is labeled as target $(Y_X = 1)$ if there exists at least one target sequence within the set. A bag X is labeled negative $(Y_X = 0)$ if all constituent sequences are nontarget. That is, $\exists \check{x} \in X : Y_{\check{x}} = 1 \Rightarrow Y_X = 1$ and $\forall \check{x} \in X, Y_{\check{x}} = 0 \Rightarrow Y_X = 0$. Given a collection of positively labeled bags, i.e., $\mathbf{B}^+$, and a collection of negatively labeled bags, i.e., $\mathbf{B}^-$, a standard approach would be to learn the HMM parameters, given the following objective:

$$\Theta = \arg\max_{\Theta} P(Y_{X_1}, \ldots, Y_{X_N}, X_1, \ldots, X_N | \Theta). \tag{5}$$

The objective in (5) is to maximize the joint probability of the bags of sequences and the corresponding class labels for the bags. Assuming independence between the bags, assuming the noisy-OR relationship between the sequences within each bag, and using the Bayes rule with the uniform prior assumption as in the standard MIL [2], (5) can be turned into the following problem:

$$\Theta = \arg\max_{\Theta} \prod_{X \in \mathbf{B}^+} P(Y = 1|X, \Theta) \prod_{X \in \mathbf{B}^-} P(Y = 0|X, \Theta) \tag{6}$$

where

$$P(Y = 1|X, \Theta) = 1 - \prod_{\check{x} \in X} (1 - P_{HMM}(\check{x}|\Theta)) \tag{7}$$

$$P(Y = 0|X, \Theta) = \prod_{\check{x} \in X} (1 - P_{HMM}(\check{x}|\Theta)) . \tag{8}$$

In (7) and (8), the products are computed using all the sequences in a given positive or negative bag, respectively, in accordance with (6). Moreover, it is worthwhile to emphasize that in (6)–(8), noisy-OR does not introduce any further parameters beyond those found in the HMM model. The only parameters to learn are the HMM parameters from (4). Therefore, no additional parameters have been introduced by providing the MI learning of HMMs, but rather, a new model has been introduced that does not require individual labels for target sequences. We also alter (6) to be a ratio as

$$\hat{\Theta} = \arg\max_{\Theta} R^*(\Theta) = \arg\max_{\Theta} \frac{\prod_{X \in \mathbf{B}^+} P(Y=1|X,\Theta)}{\prod_{X \in \mathbf{B}^-} P(Y=1|X,\Theta)} \quad (9)$$

where R^* is the name of our new objective function. The main reason for using (9) is that HMMs yield probabilities $P(Y = 1|X, \Theta)$ that are much less than 1 for long sequences. Hence, (6) may assign a very low probability to the event $Y = 1$, which is not the desired behavior. Adopting the criterion in (9) may correct for this issue to a degree.

B. Parameter Learning

The noisy-OR objective function is not easy to solve with gradient-based algorithms, and the optimization task presented in (9) is quite difficult. Therefore, a sampling-based learning scheme is proposed as a global optimization method. The general idea behind the proposed sampling scheme is to draw parameters $\Theta = \{\boldsymbol{\pi}, A, B\}$ from a proposal density and then perform a rejection step based on our objective R^* in (9). Here, $\boldsymbol{\pi}$ can be sampled from a Dirichlet distribution, but for the sake of time, we opted to simply estimate from the current transition matrix. For the estimation of $\boldsymbol{\pi}$, the first state is always considered to be the start state, as described in [45] and [46]. For the other parameter estimates, a Metropolis–Hastings-type sampling scheme is proposed. In this scheme, samples are generated from a simpler distribution, which is the so-called *proposal density* [47], and are used to search the parameter space of the objective in (9). We should explicitly note here that, since the Markov chain is broken due to the estimation of $\boldsymbol{\pi}$ from the transition matrix as opposed to sampling it, the learning is a stochastic EM and not a Metropolis–Hastings sampling. However, since we use the steps of Metropolis–Hastings sampling for the estimation of the parameters other than $\boldsymbol{\pi}$, we may refer to them as the Metropolis step in the rest of this paper.

Note that the columns in the state transition matrix $A_{\bullet j}$ and rows in the emission matrix $B_{i\bullet}$ are all multinomial distributions. Therefore, an intuitive choice for the proposal density is the Dirichlet distribution $\mathscr{D}(\boldsymbol{\alpha})$ [48]. A more generalized choice would be to use a Dirichlet mixture, which is a linear combination of several simple Dirichlet distributions [49], [50]. Our proposed method assumes a mixture of Dirichlet distributions as given in

$$\begin{aligned} &\forall j, A_{\bullet j} \sim c_1\mathscr{D}(k_1\boldsymbol{\alpha}) + c_2\mathscr{D}(k_2\boldsymbol{\alpha}) \\ &\forall i, B_{i\bullet} \sim c_1\mathscr{D}(k_1\boldsymbol{\alpha}') + c_2\mathscr{D}(k_2\boldsymbol{\alpha}') \end{aligned} \quad (10)$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ are the Dirichlet parameters and have the same dimensionality as $A_{\bullet j}$ and $B_{i\bullet}$, respectively. The parameters c_1 and c_2 are the mixture components, and k_1 and k_2 determine the "focused" or "random" nature of the component in sampling. If k_1 is chosen to be a smaller value and k_2 is chosen to be a larger value, new parameters are a mix of samples from $\mathscr{D}(k_2\boldsymbol{\alpha})$, which are more focused around $\boldsymbol{\alpha}$, and samples from $\mathscr{D}(k_1\boldsymbol{\alpha})$, which are less focused around $\boldsymbol{\alpha}$. This allows for not only sampling from the close vicinity of the parameters but also moving away and getting out of the local minimum during sampling.

The Dirichlet mixture model in (10) is our proposal distribution. New samples of parameters ($A_{\bullet j}$ and $B_{i\bullet}$) are drawn from this mixture model. New samples obtained at each draw are accepted or rejected by a Metropolis step as in [47].

For the Metropolis step, variable θ' forthwith denotes tentative new states of either $A_{\bullet j}$ or $B_{i\bullet}$. Notice that these tentative new states were sampled from the mixture of Dirichlet distributions in (10). Moreover, θ^c, which is the current state, denotes the sample accepted at iteration c. The tentative new state, i.e., θ', is accepted or rejected based on ratio r at iteration $c+1$ [47]. This ratio is computed as

$$r_{c+1}(\theta') = \min\left\{1, \frac{R^*(\theta')}{R^*(\theta^c)} \frac{\mathscr{D}(\theta^c;\theta')}{\mathscr{D}(\theta';\theta^c)}\right\}. \quad (11)$$

Although the notations are similar, it is important to notice that in (10), random samples are generated from the Dirichlet distribution, which requires only one set of parameters, whereas in (11), the Dirichlet distribution is being evaluated, which requires two sets of parameters [51]. Note that sampling from the first Dirichlet distribution in (10) is achieved by, first, drawing independent random samples from the Gamma distribution with parameters Gamma$(k_1\boldsymbol{\alpha}, \mathbf{1})$ and, second, normalizing these values by their sum.

In (11), θ' is accepted if $r_{c+1}(\theta')$ is equal to or larger than 1. Otherwise, θ' is accepted with probability $r_{c+1}(\theta')$. Due to this accept/reject property, the sampling-based training of MI-HMM is able to evolve with new parameters and can avoid getting stuck in the local minimum.

Let C be the total number of iterations, and let $\sum\mathscr{D}(\boldsymbol{\alpha})$ denote the mixture of Dirichlet distributions. Given this optimization framework, the sampling schedule is shown in Algorithm IV.1.

Algorithm IV.1 Sampling Schedule (X, Y, Θ^0, C)

```
$c = 0$
initialize $c_1, c_2, k_1, k_2, A, B$
while $c < C$
do {
    for $j \leftarrow 1$ to $numberOfStates$
    do {
        $A_{\bullet j} \sim \sum\mathscr{D}(\boldsymbol{\alpha})$
        re − estimate $\boldsymbol{\pi}$
        Accept $A'_{\bullet j}$ with prob. $\min\left(1, r_{c+1}\left(A'_{\bullet j}\right)\right)$
        if $A'_{\bullet j}$accepted
            then $A^{c+1}_{\bullet j} \leftarrow A'_{\bullet j}$
            else $A^{c+1}_{\bullet j} \leftarrow A^{c}_{\bullet j}$
    for $i \leftarrow 1$ to $numberOfStates$
    do {
        $B_{i\bullet} \sim \sum\mathscr{D}(\boldsymbol{\alpha}')$
        Accept $B'_{i\bullet}$with prob. $\min\left(1, r_{c+1}\left(B'_{i\bullet}\right)\right)$
        if $B'_{i\bullet}$accepted
            then $B^{c+1}_{i\bullet} \leftarrow B'_{i\bullet}$
            else$B^{c+1}_{i\bullet} \leftarrow B^{c}_{i\bullet}$
return $\Theta^c$
```

Verbally, each row of the A matrix is sampled from (10) and is accepted or rejected based on the Metropolis ratio. Similarly, each row of B is sampled from (10) and is accepted or rejected based on (11). This is continued until convergence is satisfied or the number of iterations is reached. In our experiments, the number of iterations was empirically chosen as the termination condition.

V. Results on Synthetic Data

Here, the MI-HMM is compared with a standard HMM. The standard HMM uses two HMMs (a target and a nontarget model) and uses the EM [35] learning scheme to optimize each model. This standard HMM will be referred to as the EM-HMM hereafter. Specifically, this section compares the MI-HMM and the EM-HMM in their ability to discern a target sequence when it is observed with other nontarget sequences. The assumption here is that the target HMM will have difficulty optimizing an objective function if there is ambiguity in the target sequences, whereas the optimization of the MI-HMM will have the ability to account for ambiguity, and therefore, the MI-HMM will successfully characterize the target sequences.

To provide reproducible experiments, sequences were generated from two HMM models with known parameters. Negatively labeled sequences were generated from HMM-1, and positively labeled sequences were generated from HMM-2. All of the sequences were of length 10 and were generated from two-state four-symbol models using the following:

HMM-1 parameters (negatively labeled sequences)

$$A = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.66 & 0.26 & 0.04 & 0.04 \\ 0.26 & 0.66 & 0.04 & 0.04 \end{pmatrix}.$$

HMM-2 parameters (positively labeled sequences)

$$A = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.08 & 0.08 & 0.26 & 0.58 \\ 0.08 & 0.08 & 0.58 & 0.26 \end{pmatrix}.$$

Then, 100 bags were arranged such that each bag had 25 sequences. Of the 100 bags, 50 were labeled positive, and 50 were labeled negative. All of the sequences in the negative bags were generated from HMM-1. Only one sequence in each positive bag was generated from HMM-2, and the rest were generated from HMM-1. These sequences were used in MI-HMM training as well as EM-HMM [35] training. For a fair comparison, all HMMs were initialized with

$$A = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \quad B = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}.$$

For EM-HMM, two HMM models were learned, one for the sequences in the positive bags (target model) and one for the sequences in the negative bags (background model). For scoring, the difference of the log-likelihoods between the target model and the background model was computed in generating the receiver operating characteristic (ROC) curves shown in Fig. 6.

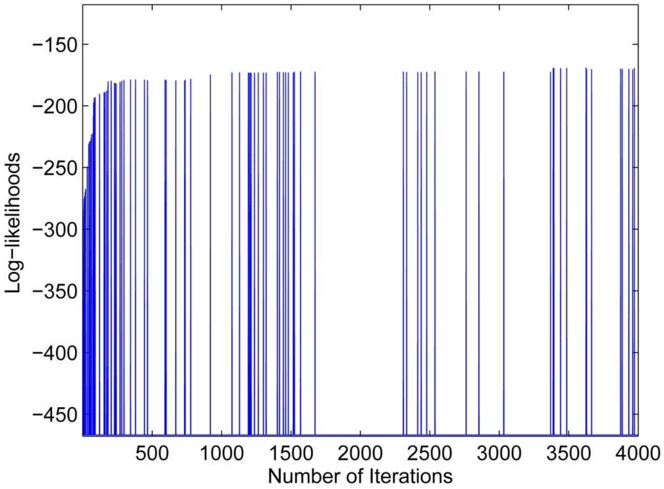


Fig. 3. Log-likelihoods with respect to the number of iterations. Accepted log-likelihoods are plotted as the blue lines, and rejected log-likelihoods are left blank. The denseness of the blue lines at the beginning indicates that more log-likelihoods are initially accepted. As the iterations progress, the log-likelihoods start to get rejected. This property is highly related to the ratios, which are plotted in Fig. 5.

For MI-HMM, the mixture proportions were $c_1 = 0.5$ and $c_2 = 0.5$, and the focus parameters were $k_1 = 1$ and $k_2 = 10$. Therefore, new parameters are a mix of samples from $\mathscr{D}(k_2\boldsymbol{\alpha})$, which are more focused around $\boldsymbol{\alpha}$, and samples from $\mathscr{D}(k_1\boldsymbol{\alpha})$, which are less focused and, therefore, may help in getting out of a local minimum.

The number of iterations was set to $C = 1000$. MI-HMM computes the log-likelihood for each row of the transition and emission matrices. Hence, there are $2 * W * C$ likelihood computations, where W is the number of states. However, not all of these likelihoods are accepted. This aspect is shown in Fig. 3, where the log-likelihoods are displayed as a function of the number of iterations. In this figure, at the beginning of the iterations, many of the proposed parameters result in an increase in the log-likelihood, and therefore, they are accepted. As the number of iterations progresses, it becomes sparse to accept a new parameter, and the new log-likelihoods are mostly rejected as indicated by the blank spaces. The accepted log-likelihoods are concatenated together and plotted in Fig. 4 for clarity. Notice that there is a general trend to increase the log-likelihood, but it does not have to increase at every iteration. The reason is that a new parameter set can still be accepted with probability r if ratio r is less than 1, as previously explained. These r values are displayed in Fig. 5.

Classification results comparing the MI-HMM and the standard HMM are presented via an ROC curve in Fig. 6. An ROC curve is a plot of the probability of detection (PD) versus probability of false alarm (PFA). Each ROC curve is displayed with error bars that show the 95% confidence interval assuming a binomial distribution on the PD. In this figure, at 90% detection, MI-HMM has only a 6% PFA, whereas the EM-HMM has a 50% PFA. Moreover, the difference between MI-HMM and EM-HMM is statistically significant as evidenced by the nonoverlapping error bars.

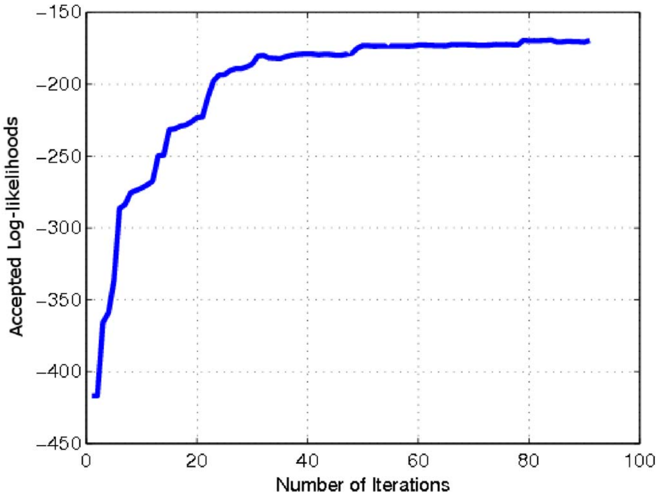


Fig. 4. Accepted log-likelihoods from Fig. 3 are concatenated and plotted for clarity. Among all the iterations, only about 90 of them were accepted. Although there is a general trend to increase the log-likelihood, it does not have to be increasing at every iteration, as shown by this plot.

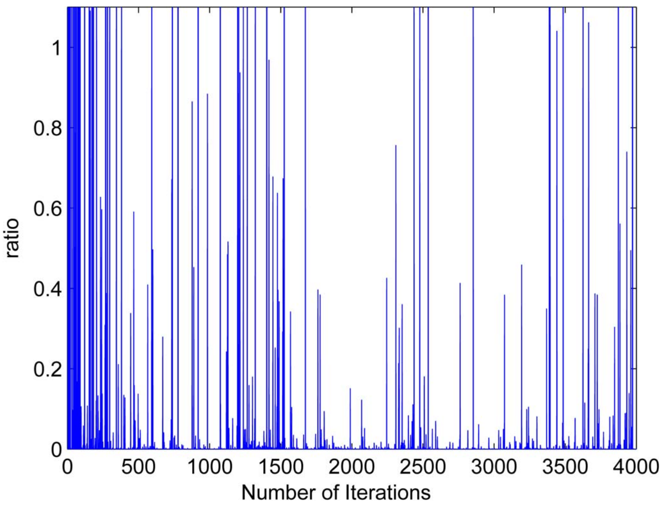


Fig. 5. Ratio r values as a function of the number of iterations. At the beginning, most r values are 1, resulting in the acceptance of the parameters. Later on, such definite accepted parameters are decreased, and the sampled parameters are accepted with probability r. These ratio values determine if a log-likelihood is accepted, which was observed in the patterns of accept/reject in Fig. 3.

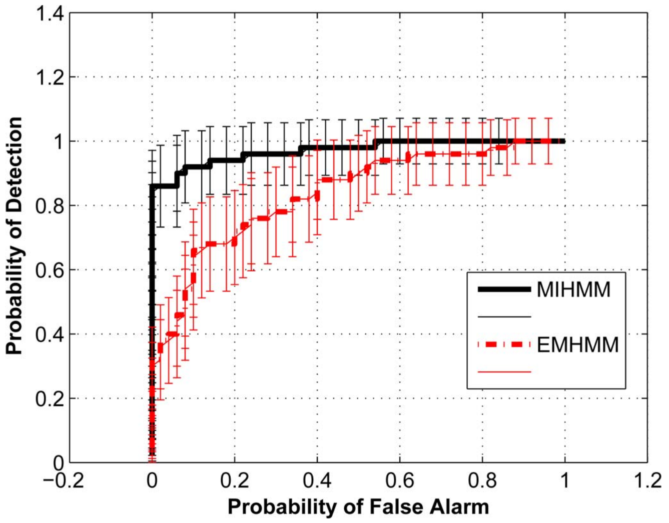


Fig. 6. Comparison of EM-HMM and MI-HMM with the ROCs. Error bars are displayed, which show the 95% confidence interval assuming a binomial distribution. At 90% detection, MI-HMM has only a 6% PFA, whereas the EM-HMM has a 50% PFA. In addition, the error bars do not overlap until an 85% PD, showing that the difference is statistically significant.

TABLE I
COMPARISON OF AUC

	AUC	Variance
MI-HMM	0.94	0.0014
EM-HMM	0.88	0.0014

TABLE II
COMPARISON OF AUC FOR INCREASING NUMBER OF ITERATIONS

Iterations	10	20	40	60	100
Max AUC	0.933	0.967	0.963	0.994	0.987
Min AUC	0.656	0.798	0.838	0.860	0.841
Avg AUC	0.814	0.888	0.908	0.934	0.939
Std AUC	0.091	0.067	0.048	0.036	0.049

Furthermore, the same setup of experiments was run 30 times with random training and testing data sets. For each ROC curve obtained, the area under the curve (AUC) was recorded as shown in Table I. The mean AUC for MI-HMM was a significant improvement of 0.94 as opposed to the mean AUC of EM-HMM of 0.88. In both cases, the AUC variance was 0.0014.

In another experiment, MI-HMM was tested for AUC for an increasing number of iterations. The MI-HMM was run ten times with $C = 10$, $C = 20$, $C = 40$, $C = 60$, and $C = 100$ iterations. As before, this corresponds to $2 \times 2 \times C$ parameter updates using the previous HMM structure. In addition, each experiment was independent from the others, in that the train and test data sets were randomly generated for each experiment using the bag structure as before. Table II shows the mean, maximum, minimum, and standard deviation of the AUC values obtained from the ten experiments for each iteration experiment. Although the number of iterations was kept at smaller values, in some instances, the MI-HMM was able to reach successful (close to 1) AUC values rather quickly without many parameter updates.

VI. Landmine Detection

In the following, a real-world landmine data set is tested. Typical landmine and clutter signatures are shown in Fig. 7. In GPR images, scanning from left to right, a landmine signature would appear as a rising edge followed by a falling edge. Therefore, edge features are computed from GPR images, and edge feature sequences are constructed for each horizontal image scan. The goal is to learn the horizontal patterns indicative of a landmine signature using an HMM model. In the following, first, the data set is described, then ***MI-HMM is compared with a state-of-the-art HMM*** [26], [52]; ***a benchmark approach that is currently operational***, ***which is referred to hereafter as the*** "***sampling HMM***" ***or HMMSamp***.

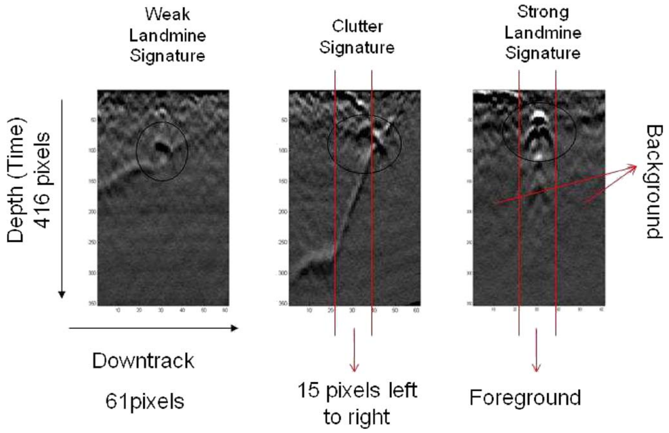


Fig. 7. Three GPR downtrack images showing landmine and clutter signatures (which are circled). The foreground and background areas used for preprocessing are also shown.

A. Data

A NIITEK Inc. landmine detection system with a GPR sensor was used to collect data from various test sites consisting of gravel and dirt roads containing buried landmines and clutter objects. The system uses an array of V-dipole antennas that generate a wideband pulse ranging from 200 MHz to 7 GHz. Subsurface objects appear within the GPR data, as shown in Fig. 7.

Prescreener: To lessen the computational burden of more complex algorithms, a standard prescreening algorithm [53], [54] is run to identify areas of interest, which are also called alarms. A GPR alarm is a 3-D data cube: 416 samples in depth, 61 samples downtrack (down the road), and 24 samples crosstrack (for each channel in the GPR antenna). The resulting data collection consists of approximately 1000 target alarms and 2500 nontarget alarms.

Several preprocessing steps are performed to exaggerate edges, as explained in detail in [14], [24], and [32]. To state simply, first, a gradient filter is run over the image to accentuate edges [24]. This removes the stationary effects that remain relatively constant from scan to scan. Second, the image is separated into foreground and background, as shown in Fig. 7. Then, the foreground is "whitened." Whitening at each depth involves subtracting the mean of the background from each pixel in the foreground and then dividing by the standard deviation of the background.

Feature Extraction: After preprocessing, edge features are computed. First, erosion is applied for edge detection [55]. Two 5×5 windows are used to identify whether a "rising edge" or a "falling edge" is present. As a result, each pixel contains a 4-D feature vector: rising edge template for positive values, rising edge template for the magnitude of negative values, falling edge template for positive values, and falling edge template for the magnitude of negative values. These features are then condensed into a 2-D feature vector by averaging both rising features and both falling features, respectively. Finally, a 10×1 (vertical) max is taken within this feature vector image. The goal of this step is to "blur" the features, such that resulting sequences may not be severely affected by missing features or gaps. This feature extraction step has been explained in detail in [24] and demonstrated in [52].

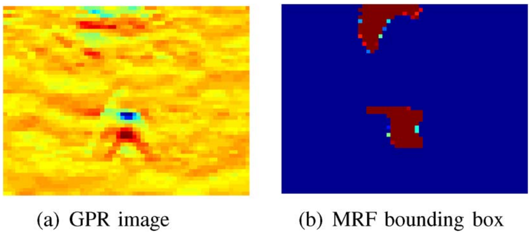


Fig. 8. Preprocessing steps for the GPR image shown in Fig. 1. (a) GPR image after ground removal. (b) Automatic placement of the MRF bounding box. Although the MRF bounding box eliminates many of the nontarget sequences, it is not perfect, and much ambiguity during training remains.

With these steps, horizontal image scans are converted into feature vector sequences that indicate the presence of various edge types. As a result, at each fixed depth, there is a horizontal sequence of edge feature vectors of length 15. These sequences can identify whether there are rising or falling edges within a particular horizontal scan, and they will be the sequences used in testing our classifier.

Each image has 416 potential training sequences but has only one class label associated with the image. Therefore, to reduce the number of nontarget sequences from a target image, the selection of training samples is aided using a Markov random field (MRF) "bounding box" [56]. The goal of the MRF is to bound the subimage with the highest energy—the target. This is a standard and automated procedure to reduce the arduousness of the task and the enormity of data typically used. Although this initial step eliminates many of the nontarget sequences within a target image, it is not perfect, and much ambiguity during training remains, as shown in Fig. 8.

B. Experimental Design

The proposed MI-HMM is compared with the sampling HMM [26], [52] using the aforementioned landmine data. As mentioned before, the sampling HMM is the best performing HMM classifier that is implemented in the operational landmine detection systems. The sampling HMM has been compared with many alternative methods in [12] and [29] and has been selected as the best algorithm in large-scale evaluations on a testing/training unified framework that is designed to provide an objective and consistent evaluation of different algorithms.

The sampling HMM uses a Gibbs-sampling-based training. Therefore, it provides a good comparison between traditional and MIL-based HMM classifiers. In the following, both MI-HMM and sampling HMM are described.

MI-HMM uses the MIL objective and learns a single HMM model with four states. It is a discrete model; hence, feature vectors were discretized (uniformly) to one of 25 different symbols. Training for the MI-HMM is as follows: For each target image, five evenly spaced sequences were selected from within the MRF bounding box and placed into positive bags, and five randomly selected sequences were chosen from nontarget images and placed into negative bags. Testing using the MI-HMM is performed by summing up the log of the probabilities of each of the 416 sequences in each of the 24 images for each alarm. This accumulated value is considered the target confidence

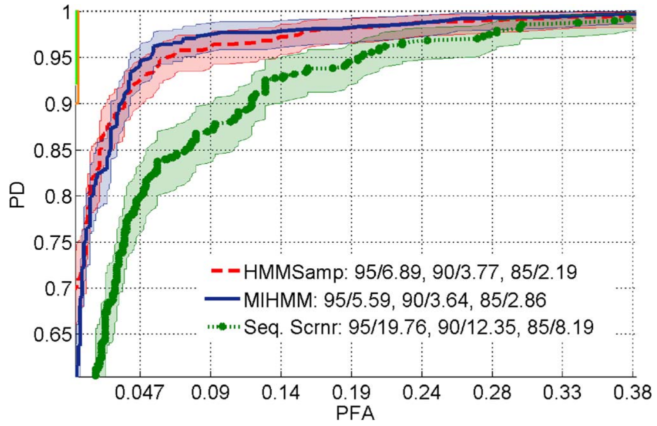


Fig. 9. ROC curves for the sequence screener algorithm, MI-HMM, and the sampling HMM **after** the application of the sequence screener. The sampling HMM has been labeled as HMMSamp. The error bars on the ROC curves show the 95% confidence interval assuming a binomial distribution on the PD.

for each alarm. The parameters of MI-HMM were set to be $c_1 = 0.5$, $c_2 = 0.5$, and $k_1 = 10$, $k_2 = 80$.

Sampling HMM uses a joint probability objective and optimizes it using a Gibbs sampling schedule. It uses continuous (nondiscretized) sequences as input. The sampling HMM algorithm learns two HMMs, i.e., one for the target model and one for the background model. It trains a target HMM using a training set of sequences from target images and trains a nontarget HMM using a training set of sequences from nontarget images. These HMMs also had four states and one Gaussian component per state. For a fair comparison, the same sequences used in MI-HMM were used to train the two HMMs of the sampling HMM. Testing for the sampling HMM is the log of the ratio of the probability of the target model over the probability of the nontarget model.

Sequence screener is an ad hoc algorithm designed to eliminate many of the nonmine sequences. Simply speaking, the main target concept that the HMMs should be learning is a sequence with a rising edge followed by a falling edge. Therefore, the sequence screener simply sifts through all of the test images and disregards all sequences that do not have a strong rising edge followed by a strong falling edge. It also disregards the sequences that are too short in between the rising and falling edges. The sequence screener detection statistic is a Boolean operator that is aggregated across each row of the image. If that row has a "mine-like" feature sequence that passes the screening, then the detection statistic is incremented by 1. These values are aggregated across each row of the image. Details can be found in [52]. Data are pruned with the sequence screener before showing it to the HMMs in the testing stage.

C. Experimental Results on Landmine Data

Classification results comparing the MI-HMM and the sampling HMM are presented via an ROC curve in Figs. 9 and 10 for tenfold cross validation. Each ROC curve is displayed with error bars that show the 95% confidence interval assuming a binomial distribution on the PD.

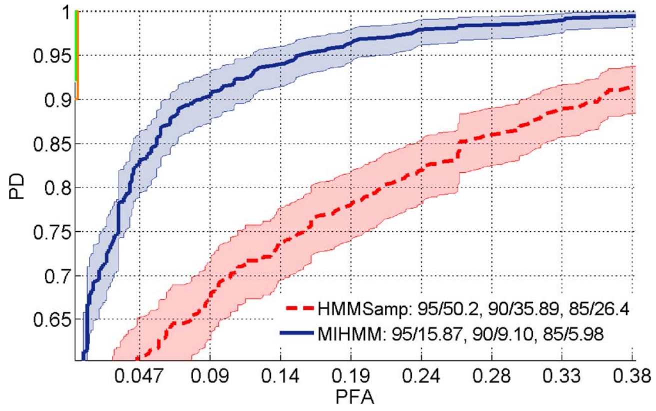


Fig. 10. ROC curves for MI-HMM and sampling HMM (HMMSamp) alone, without the sequence screener. The error bars on the ROC curves show the 95% confidence interval assuming a binomial distribution on the PD.

TABLE III
TENFOLD CLASSIFICATION RATES ON LANDMINE GPR DATA

	Model	PD 85%	PD 90%	PD 95%
PFA %	MI-HMM w. Prscrnr.	2.86	3.64	5.59
	HMMSamp w. Prscrnr.	2.19	3.77	6.89
	Sequence screener	8.19	12.35	19.76
	MI-HMM	5.98	9.10	15.87
	HMMSamp	26.40	35.89	50.20

In Fig. 9, first, the sequence screener is applied as usual. The sequence screener is an ad hoc algorithm that eliminates sequences that are unmistakably nontarget like. It provides 12.35% PFA at 90%PD, which is the standard operating PD for this landmine detection system. When the sampling HMM and the MI-HMM are applied *after* the sequence screener, they decrease the PFA rates to around 3.77% and 3.64% at 90%PD. At this point, between MI-HMM and the sampling HMM, the difference in detection rates at 90%PD is not statistically significant. However, this is, in fact, a big accomplishment for MI-HMM as it does not require the manual labor needed to train the sampling HMM. In addition, MI-HMM is a single HMM as opposed to the two HMMs in the sampling HMM algorithm and has half the parameters of the sampling HMM. Furthermore, the MI-HMM ROC dominates well outside the 95% confidence interval, which indicates improved classification.

A second scenario is presented in Fig. 10, where the sequence screener is removed, and MI-HMM and sampling HMM are run alone. Without the sequence screener algorithm, there is more clutter in the data set. Therefore, a drop in classification rates is expected for both the MI-HMM and the sampling HMM. However, this drop is extremely pronounced in the sampling HMM, which shows a 35.89% PFA at 90% PD. On the other hand, the MI-HMM stays more robust and only drops to 9.10% PFA at 90% PD. In fact, the ROC of the MI-HMM dominates the ROC of the sampling HMM at all operating thresholds above a PD of 70%.

The PFA values for these plots are given in Table III for 85%, 90%, and 95% PD values. The results show a significant improvement in classification results using the proposed MI-HMM versus the sampling HMM.

The comparison of the HMMs with or without the sequence screener is quite notable. When the sequence screener algorithm is used, both models have statistically similar performances. This indicates that the sampling HMM algorithm had a similar potential performance upper bound as the MI-HMM but failed to achieve similar performance results without the sequence screener. On the other hand, the MI-HMM could perform near its MI-HMM+sequence screener version even without the sequence screener algorithm. Therefore, it showed to be more robust to changes in the data set and, more specifically, to ambiguous data. With all these comparisons combined, MI-HMM provides a good principled alternative to replace the ad hoc sequence screener methods while increasing classification rates.

VII. Conclusion

MIL is one of the flourishing areas in machine learning. Recently, it has gained recognition for learning models to represent ambiguous data, where the data associated with a particular object are a collection of feature vectors, but only a subset of those feature vectors is associated with the object's class. Such data are commonly observed in landmine detection in GPR images where both landmines and clutter signals can be observed in a given landmine alarm.

In this paper, an HMM with an MI learning scheme has been presented and tested on both synthetic data as well as landmine data. MI-HMM has a very clean mathematical model since there is no addition of parameters, but rather an assumption of the learning scenario. Within the landmine data experiments, the MI-HMM significantly outperformed the sampling HMM and EM-HMM algorithms that made use of two HMMs (twice the parameters). Moreover, the performance of MI-HMM did not significantly degrade without the sequence screener, whereas the performance did degrade significantly for the sampling HMM. Given the results of these experiments as well as the synthetic data experiments, it is clear that the use of an MI learning scheme when an MI scenario is present can increase classification results or at least provide principled automated methods for classification.

As one of our reviewers suggested, for future work, one could try sampling all the parameters without breaking the Markov chain and also experimenting with different values of k including values less than 1. That would greatly change the behavior of the method but would be interesting to look into. In this paper, we only tried k values that were bigger than 1.

Finally, the MI-HMM could be useful not only for landmine detection but also in many other applications that involve ambiguous time-series data, such as the analysis of video sequences and the classification of sounds in a scene.

References

[1] Q. Zhang, S. A. Goldman, W. Yu, and J. E. Fritts, "Content-based image retrieval using multiple-instance learning," in *Proc. 19th Int. Conf. Mach. Learn.*, Jul. 2002, pp. 682–689.

[2] J. Yang, "Review of multi-instance learning and its applications," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 2005.

[3] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artif. Intell.*, vol. 176, no. 1, pp. 2291–2320, Jan. 2012.

[4] Z. Zhou, "Multi-instance learning: A survey," AI Lab, Dept. Comput. Sci. Technol., Nanjing Univ., Nanjing, China, Tech. Rep., Mar. 2004.

[5] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez, and A. Pharmaceutical, "Solving the multiple-instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1/2, pp. 31–71, Jan. 1997.

[6] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 288–303, Feb. 2010.

[7] A. Fathi, X. Ren, and J. Rehg, "Learning to recognize objects in egocentric activities," in *Proc. IEEE CVPR*, Jun. 2011, pp. 3281–3288.

[8] Z. Qi, Y. Xu, L. Wang, and Y. Song, "Online multiple instance boosting for object detection," *Neurocomputing*, vol. 74, no. 10, pp. 1769–1775, May 2011.

[9] M. Li, J. T. Kwok, and B.-L. Lu, "Online multiple instance learning with no regret," in *Proc. IEEE CVPR*, 2010, pp. 1395–1401.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[11] H. Frigui, P. Gader, and D. Ho, "Real time landmine detection with ground penetrating radar using discriminative and adaptive hidden Markov models," *EURASIP J. Appl. Signal Process.*, vol. 12, no. 1, pp. 1867–1885, Jan. 2005.

[12] J. Wilson, P. Gader, W.-H. Lee, H. Frigui, and K. Ho, "A large-scale systematic evaluation of algorithms using ground-penetrating radar for landmine detection and discrimination," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2560–2572, Aug. 2007.

[13] S. E. Yuksel *et al.*, "Hierarchical methods for landmine detection with wideband electro-magnetic induction and ground penetrating radar multisensor systems," in *Proc. IEEE IGARSS*, Jul. 2008, vol. 2, pp. 177–180.

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

[15] H. Frigui, L. Zhang, and P. Gader, "Context-dependent multisensor fusion and its application to land mine detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2528–2543, Jun. 2010.

[16] S. E. Yuksel and P. D. Gader, "Mixture of HMM experts with applications to landmine detection," in *Proc. IEEE IGARSS*, 2012, pp. 6852–6855.

[17] S. E. Yuksel, G. B. Akar, and S. Ozturk, "Fusion of forward-looking infrared camera and down-looking ground penetrating radar for buried target detection," in *Proc. SPIE Detection Sens. Mines, Explosive Obj., Obscured Targets*, 2015, p. 945 418.

[18] O. Maron and T. Lozano-Prez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA, USA: MIT Press, 1998, pp. 570–576.

[19] V. C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, and R. B. Rao, "Bayesian multiple instance learning: Automatic feature selection and inductive transfer," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 808–815.

[20] C. Bergeron, G. Moore, J. Zaretzki, C. Breneman, and K. Bennett, "Fast bundle algorithm for multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1068–1079, Jun. 2012.

[21] S. E. Yuksel, J. Bolton, and P. D. Gader, "Landmine detection with multiple instance hidden Markov models," in *Proc. IEEE Int. Workshop MLSP*, 2012, pp. 1–6.

[22] J. Bolton and P. Gader, "Cross entropy optimization of the random set framework for multiple instance learning," in *Proc. 20th ICPR*, Aug. 2010, pp. 3907–3910.

[23] J. Bolton, P. Gader, H. Frigui, and P. Torrione, "Random set framework for multiple instance learning," *Inf. Sci.*, vol. 181, no. 11, pp. 2061–2070, Jun. 2011.

[24] P. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 6, pp. 1231–1244, Jun. 2001.

[25] Y. Zhao, P. Gader, P. Chen, and Y. Zhang, "Training DHMMs of mine and clutter to minimize landmine detection errors," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 1016–1024, May 2003.

[26] X. Zhang, S. E. Yuksel, P. Gader, and J. Wilson, "Simultaneous feature and HMM model learning for landmine detection using ground penetrating radar," in *Proc. 6th IAPR Workshop PRRS*, Aug. 2010, pp. 1–4.

[27] O. Missaoui, H. Frigui, and P. Gader, "Land-mine detection with ground-penetrating radar using multistream discrete hidden Markov models," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2080–2099, Jun. 2011.

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

[29] H. Frigui *et al.*, "An evaluation of several fusion algorithms for anti-tank landmine detection and discrimination," *Inf. Fusion*, vol. 13, no. 2, pp. 161–174, Apr. 2012.

[30] A. Mendez-Vazquez, P. D. Gader, J. M. Keller, and K. Chamberlin, "Minimum classification error training for Choquet integrals with applications to landmine detection," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 225–238, Feb. 2008.

[31] M. Popescu, P. D. Gader, and J. M. Keller, "Fuzzy spatial pattern processing using linguistic hidden Markov models," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 81–92, Feb. 2006.

[32] P. Gader, W.-H. Lee, and J. Wilson, "Detecting landmines with ground-penetrating radar using feature-based rules, order statistics, and adaptive whitening," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 11, pp. 2522–2534, Nov. 2004.

[33] K. C. Ho and P. D. Gader, "A linear prediction land mine detection algorithm for hand held ground penetrating radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 6, pp. 1374–1384, Jun. 2002.

[34] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1988.

[35] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[36] M. Mohamed and P. Gader, "Generalized hidden Markov models. I. Theoretical frameworks," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 1, pp. 67–81, Feb. 2000.

[37] M. Mohamed and P. Gader, "Generalized hidden Markov models. II. Application to handwritten word recognition," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 1, pp. 82–94, Feb. 2000.

[38] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[39] A. Biem, "Minimum classification error training for online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1041–1051, Jul. 2006.

[40] P. Djuric and J.-H. Chun, "An MCMC sampling approach to estimation of nonstationary hidden Markov models," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1113–1123, May 2002.

[41] J. Paisley and L. Carin, "Hidden Markov models with stick-breaking priors," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3905–3917, Oct. 2009.

[42] L. S. Scott, "Bayesian methods for hidden Markov models: Recursive computing in the 21st century," *J. Amer. Stat. Assoc.*, vol. 97, no. 457, pp. 337–351, Mar. 2002.

[43] M. Johnson, "Why doesn't EM find good HMM POS-taggers," in *Proc. Joint Conf. EMNLP*, 2007, pp. 296–305.

[44] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," in *Proc. 45th Annu. Meet. Assoc. Comput. Linguist.*, Jun. 2007, pp. 744–751.

[45] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[46] Hidden Markov Models, Mathworks, Natick, MA, USA, 2014. [Online]. Available: http://www.mathworks.com/help/stats/hidden-markov-models-hmm.html

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

[48] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag, 2006.

[49] K. Sjlander *et al.*, "Dirichlet mixtures: A method for improved detection of weak but significant protein sequence homology," *Comput. Appl. Biosci.*, vol. 12, no. 4, pp. 327–345, Aug. 1996.

[50] V.-A. Nguyen, J. L. Boyd-Graber, and S. F. Altschul, "Dirichlet mixtures, the Dirichlet process, and the structure of protein space," *J. Comput. Biol.*, vol. 20, no. 1, pp. 1–18, Jan. 2013.

[51] L. Hortensius, "Notes on the Dirichlet Distribution," Feb. 2012. [Online]. Available: http://www.tc.umn.edu/~horte005/docs/Dirichletdistribution.pdf

[52] X. Zhang, J. Bolton, and P. Gader, "A new learning method for continuous hidden Markov models for subsurface landmine detection in ground penetrating radar," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 3, pp. 813–819, Mar. 2014.

[53] P. A. Torrione, L. M. Collins, F. Clodfelter, S. Frasier, and I. Starnes, "Application of the LMS algorithm to anomaly detection using the Wichmann/NIITEK ground-penetrating radar," in *Proc. SPIE Detect. Remediation Technol. Mines Minelike Targets VIII*, 2003, pp. 1127–1136.

[54] P. Torrione, C. Throckmorton, and L. Collins, "Performance of an adaptive feature-based processor for a wideband ground penetrating radar system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 2, pp. 644–658, Apr. 2006.

[55] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[56] L. Torrione and P. A. Collins, "Application of Markov random fields to landmine detection in ground penetrating radar data," in *Proc. SPIE Detect. Sens. Mines, Explosive Obj., Obscured Targets XIII*, 2008, vol. 6953, Art. ID. 69531B.

Seniha Esen Yuksel (S'01–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, USA, in 2006; and the Ph.D. degree in computer and information science and engineering from the University of Florida, Gainesville, FL, USA, in 2011.

She was a Postdoctoral Associate with the Department of Materials Science and Engineering, University of Florida, where she developed algorithms for explosive detection from hyperspectral data. She is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Hacettepe University, Ankara. She has worked on several target detection problems with data from ground penetrating radar, hyperspectral, electromagnetic induction, and LiDAR sensors, with a special focus on landmine detection. Her research interests include machine learning, pattern recognition, hyperspectral image analysis, statistical data analysis, computer vision, and medical imaging.

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 at the University of Florida in 2008.

Jeremy Bolton (S'07–M'09) received the B.S. degree in computer engineering and the M.Eng. and Ph.D. degrees from the University of Florida, Gainesville, FL, USA, in 2003 and 2008, respectively.

He is currently a Consultant in the area of academic course design for a variety of e-learning platforms. Previously, he was a Research Scientist with the Computational Science and Intelligence Laboratory, Department of Computer and Information Sciences and Engineering, University of Florida. His research included the development of algorithms, methodologies, and models with various applications, with a focus on landmine detection, applied to a variety of data including hyperspectral, multispectral, radar, and infrared. Notable research includes the development of the Choquet metric, the random-set framework for context-based classification, and random-set methods for multiple-instance learning techniques.

Dr. Bolton is a member of the IEEE Computational Intelligence Society, the IEEE Geoscience and Remote Sensing Society, and the Society of Photographic Instrumentation Engineers.

Paul Gader (M'86–SM'99–F'11) received the Ph.D. degree in mathematics for image-processing-related research from the University of Florida, Gainesville, FL, USA, in 1986.

He was a Senior Research Scientist with Honeywell, Minneapolis, MN, USA; a Research Engineer and Manager with the Environmental Research Institute of Michigan, Ann Arbor, MI, USA; and a Faculty Member with the University of Wisconsin—Oshkosh, Oshkosh, WI, USA; the University of Missouri, Columbia, MO, USA; and the University of Florida, where he was the Department Chair and is currently a Professor with the Department of Computer and Information Science and Engineering. He has published hundreds of refereed journal and conference papers. His research interests include a wide variety of theoretical and applied research 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 light detection and ranging image analysis projects.

Dr. Gader became a Fellow of the IEEE for his work on algorithms for landmine detection.