Hamdi and Frigui *EURASIP Journal on Advances in Signal Processing* (2015) 2015:75
DOI 10.1186/s13634-015-0260-8

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Ensemble hidden Markov models with application to landmine detection

Anis Hamdi and Hichem Frigui*

Abstract

We introduce an ensemble learning method for temporal data that uses a mixture of hidden Markov models (HMM). We hypothesize that the data are generated by K models, each of which reflects a particular trend in the data. The proposed approach, called ensemble HMM (eHMM), is based on clustering within the log-likelihood space and has two main steps. First, one HMM is fit to each of the N individual training sequences. For each fitted model, we evaluate the log-likelihood of each sequence. This results in an N-by-N log-likelihood distance matrix that will be partitioned into K groups using a relational clustering algorithm. In the second step, we learn the parameters of one HMM per cluster. We propose using and optimizing various training approaches for the different K groups depending on their size and homogeneity. In particular, we investigate the maximum likelihood (ML), the minimum classification error (MCE), and the variational Bayesian (VB) training approaches. Finally, to test a new sequence, its likelihood is computed in all the models and a final confidence value is assigned by combining the models' outputs using an artificial neural network. We propose both discrete and continuous versions of the eHMM.

Our approach was evaluated on a real-world application for landmine detection using ground-penetrating radar (GPR). Results show that both the continuous and discrete eHMM can identify meaningful and coherent HMM mixture components that describe different properties of the data. Each HMM mixture component models a group of data that share common attributes. These attributes are reflected in the mixture model's parameters. The results indicate that the proposed method outperforms the baseline HMM that uses one model for each class in the data.

Keywords: Hidden Markov models; Mixture models; Landmine detection; Ground-penetrating radar; Clustering

1 Introduction

Detection and removal of buried landmines is a worldwide humanitarian and military problem. The latest statistics [1] show that in 2012, a total of 3618 casualties from mines were recorded in 62 countries, the vast majority (78 %) of casualties were civilians. Detection and removal of landmines is therefore a significant problem and in recent years has attracted several researchers. One challenge in landmine detection lies in plastic or low metal mines that are difficult to detect by traditional metal detectors. Varieties of sensors have been proposed or are under investigation for landmine detection. Ground-penetrating radar (GPR) offers the promise of detecting landmines with little or no metal content. Unfortunately, landmine detection via GPR has proven to be a difficult problem [2, 3]. Although systems can achieve high detection rates, they have done so at the expense of high false alarm rates. The key challenge to mine detection technology lies in achieving a high rate of mine detection while maintaining a low false alarm rate. The performance of a mine detection system is therefore commonly measured by a receiver operating characteristics (ROC) curve that specifies the rate of true detection versus the rate of false alarm.

To improve the overall ROC of the landmine detection system, several algorithms have been introduced in the last decade. These algorithms use methods such as fuzzy logic [4], hidden Markov models [5–7], nearest neighbor classifiers [8, 9], support vector machines [10], or random forest [11] to assign a confidence that a mine is present at a point.

In [5, 6], hidden Markov modeling was proposed for detecting both metal and nonmetal mine types using data collected by a moving vehicle-mounted GPR system. These initial applications have proved that HMM

*Correspondence: h.frigui@louisville.edu
Department of Computer Engineering and Computer Science, University of Louisville, 40292 Louisville, KY, USA

techniques are feasible and effective for landmine detection. The initial work relied on simple gradient edge features. Subsequent work used an edge histogram descriptor (EHD) approach to extract features from the original GPR signatures. The baseline HMM classifier consists of two HMM models, one for mine and one for background. The mine (background) model captures the characteristics of the mine (background) signatures. The model initialization and subsequent training are based on global averaging over the training data corresponding to each class.

Most subsequent published works in the area of landmine detection using HMMs focused on feature-level fusion [12] and/or model-level fusion [13–15]. All of these methods still use a single model for each class. In this paper, we argue that a single model is not sufficient to capture the intra-class variability. In the context of landmine detection, variations in the class of mines may be caused by the different mine types, burial depth, soil type, and moisture. Similarly, background signatures may exhibit large variations due to different soil conditions and data preprocessing techniques. To generalize the HMM approach, we identify the variations within each class in an unsupervised manner and use multiple models to account for the intra-class variations.

The proposed approach consists of the construction of a mixture of HMMs to cover the diversity of the training data. This approach, called ensemble of hidden Markov models (eHMM), has four main components: similarity matrix computation, relational clustering, adaptive training scheme, and decision level fusion. These components are summarized by the block diagram in Fig. 1 and will be described in section 4.

The remainder of this paper is organized as follows. Section 2 provides background material on hidden Markov models. Section 3 highlights the motivations for adopting multiple models in our approach. Section 4 outlines the eHMM architecture and describes its different components. Section 5 reports the experimental results of our eHMM approach on large GPR collections and compare them to those of the baseline HMM detector. Finally, conclusions are provided in Section 6.

2 Background

2.1 Hidden Markov models

An HMM is a model of a doubly stochastic process that produces a sequence of random observation vectors at discrete times according to an underlying Markov chain. At each observation time, the Markov chain may be in one of N states $\{s_1, \ldots, s_N\}$ and, given that the chain is in a certain state, there are probabilities of moving to other states. These probabilities are called transition probabilities. Let T be the length of the observation sequence (i.e., number of time steps), let $O = \{O_1, \ldots, O_T\}$ be the observation sequence, and let $Q = \{q_1, \ldots, q_T\}$ be the state sequence. The compact notation

$$\lambda = (\mathbf{A}, \mathbf{B}, \pi) \tag{1}$$

is generally used to indicate the complete parameter set of the HMM model. In (1), $\mathbf{A} = [a_{ij}]$ is the state transition probability matrix, where $a_{ij} = Pr(q_t = j | q_{t-1} = i)$ for $i, j = 1, \ldots, N$; $\pi = [\pi_i]$, where $\pi_i = Pr(q_1 = s_i)$ are the initial state probabilities; and $\mathbf{B} = b_i(O_t), i = 1, \ldots, N$, where $b_i(O_t) = Pr(O_t | q_t = i)$ is the observation probability distribution in state i.

An HMM is called continuous if the observation probability density functions are continuous and discrete otherwise. In the case of the discrete HMM, the observation vectors are commonly quantized into a finite set of symbols, $\{v_1, v_2, \ldots, v_M\}$, called the codebook. Each state is represented by a discrete probability density function and each symbol has an associated probability of occurring given that the system is in a given state. In other words, B becomes a simple set of fixed probabilities for each state. That is, $b_i(O_t) = b_i(k) = Pr(v_k | q_t = i)$, where v_k is the nearest codebook symbol to O_t .

Given the form of the hidden Markov model defined in (1), Rabiner [16] defines three key problems of interest that must be solved for the model to be useful in real-world applications: (i) the classification problem; (ii) the problem of finding an optimal state sequence; and (iii) the problem of estimating the model parameters.

The classification problem involves computing the probability of an observation sequence $O = \{O_1, O_2, \ldots, O_T\}$ given a model λ, i.e, $Pr(O|\lambda)$. This probability is computed efficiently using the forward–backward procedure [16].

In most applications, it often turns out that computing an optimal state sequence is more useful than $Pr(O|\lambda)$. There are several possible optimality criteria. One that is particularly useful is to maximize $Pr(O, Q|\lambda)$ over all possible state sequences Q. The Viterbi algorithm [17] is an efficient and formal technique for finding this optimal state sequence and its probability.

The third problem in building an HMM is the training problem: how does one estimate the parameters of the model? The problem is difficult because there are several levels of estimation required in an HMM. First, the states themselves must be estimated. Then, the model parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ need to be estimated. For the discrete HMM, first the codebook is determined, usually using the K-means algorithm [18], or other vector quantization algorithms. Then, the parameters $(\mathbf{A}, \mathbf{B}, \pi)$ are estimated iteratively using the Baum-Welch learning algorithm [19].

2.2 Baseline HMM classifier for landmine detection

The baseline HMM classifier for GPR-based landmine detection was first introduced in [5]. It consists of two

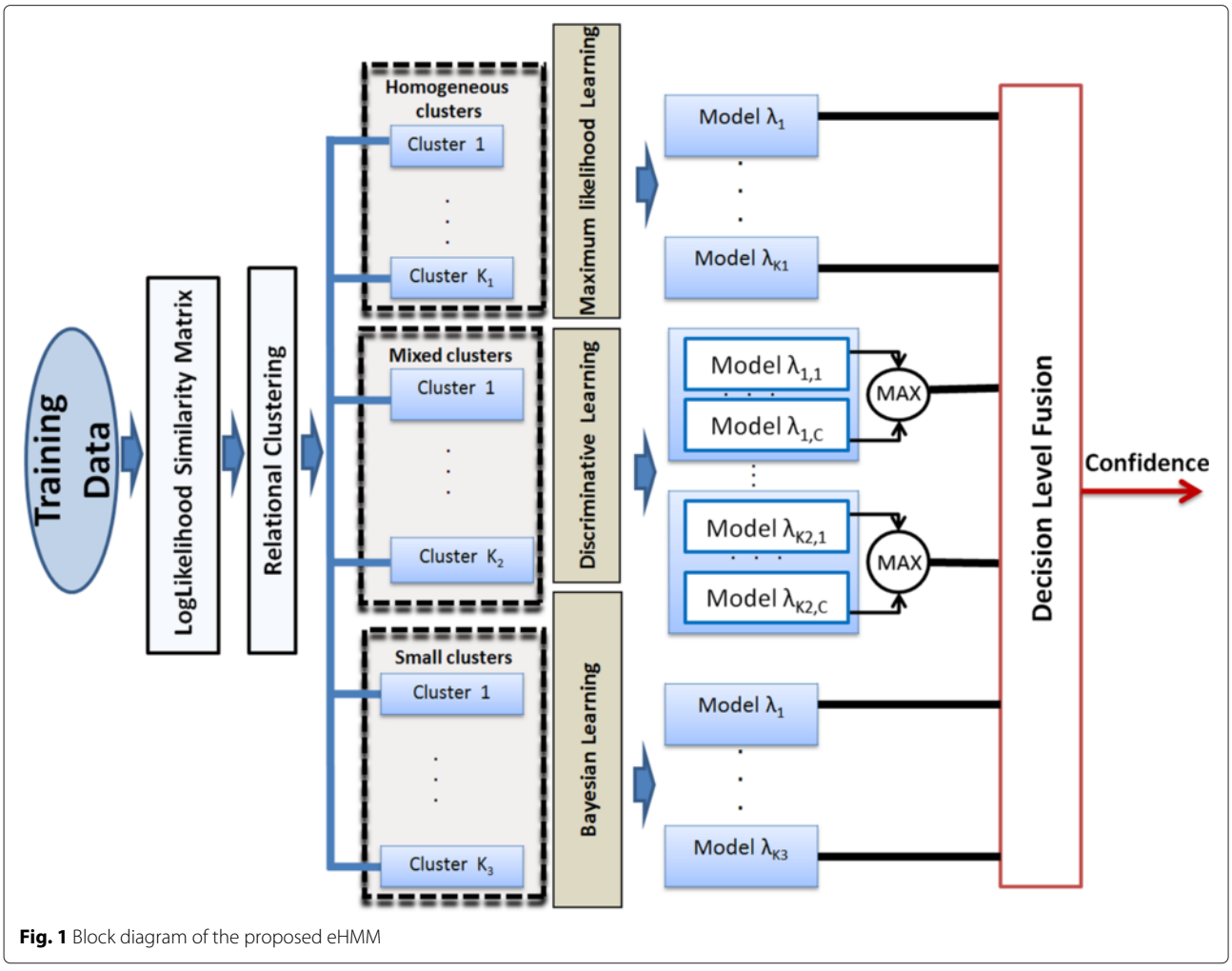


Fig. 1 Block diagram of the proposed eHMM

HMM models, one for mines and one for the background. Each model has four states and produces a probability value by backtracking through model states using the Viterbi algorithm [17]. The mine model, λ^m, is designed to capture the spatial distribution of the features. This model assumes that mine signatures have a hyperbolic shape comprised of a succession of rising, horizontal, and falling edges with variable duration in each state. The beginning and the end of the observation vectors correspond typically to a non-edge (or background) state. The background model, λ^b, is needed to capture the background and clutter characteristics. No prior information or assumptions are used for this model.

The architecture of the baseline HMM classifier is shown in Fig. 2. Full details of the model's initialization and training can be found in [5]. The probability value produced by the mine (background) model can be thought of as an estimate of the probability of the observation sequence given that there is a mine (background) present.

The confidence value assigned to each observation sequence, Conf(O), depends on: (1) the probability assigned by the mine model, $Pr(O|\lambda^m)$; (2) the probability assigned by the background model, $Pr(O|\lambda^c)$; and (3) the optimal state sequence. Thus,

$$\text{Conf}(O) = \begin{cases} \max\left(\log \frac{Pr(O|\lambda^m)}{Pr(O|\lambda^c)}, 0\right) & \text{if } \#\{s_t = 1, t = 1, \cdots, T\} \leq T_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\#\{s_t = 1, t = 1, \cdots, T\}$ corresponds to the number of observations assigned to the background state (state 1). $T_{\max}$ is defined experimentally based on the shortest mine signature. Equation (2) ensures that sequences with a large number of observations assigned to state 1 are considered non-mines.

2.3 Extensions to the baseline HMM for landmine detection

In an effort to improve performance and generalization, several extensions to the baseline HMM have been

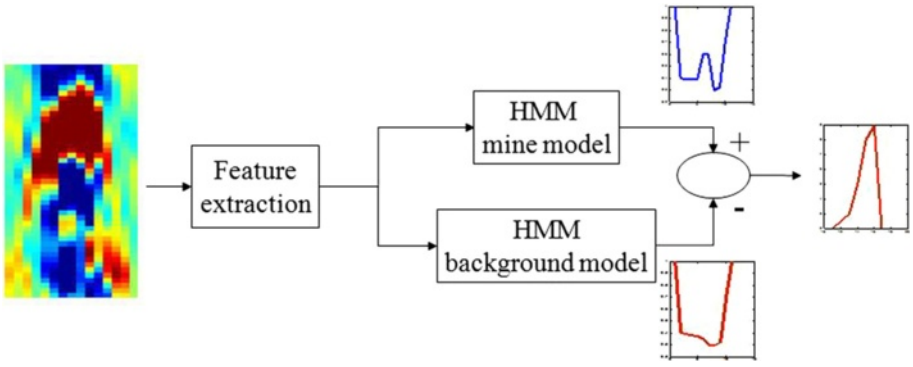


Fig. 2 Architecture of the baseline HMM classifier for landmine detection

proposed. For instance, in [12, 13], the authors proposed the multi-stream HMM (MSHMM) that combines multiple sets of features. An optimal weight for each feature was learned in the training phase. In [14], maximum likelihood (ML) and minimization of classification error (MCE) learning methods were derived for the MSHMM. In [15], HMMs with stick-breaking priors (SBHMM) [20] were employed to learn the number of HMM states in the baseline HMM landmine detector. This approach relies on a variational Bayesian learning technique in lieu of standard BW training.

3 Motivations

The baseline HMM represents each class by a single model learned from all the observations within that class. The goal is to generalize from all the training data in order to classify unseen observations. However, for complex classification problems with large intra-class variations, combining observations with different characteristics to learn one model might lead to too much averaging thus, lose the discriminating characteristics of the observations.

To illustrate this problem, we use the example of detecting buried landmines using GPR sensors[1]. In this case, the training data consists of a set of N GPR alarms labeled as mines (class 1) or clutter (class 0). The goal is to generalize from the training data in order to classify unlabeled GPR signatures. In Fig. 3, we show three groups of mines with different signature strengths. It is obvious that grouping all of these signatures, to learn a single model, would lead to poor generalization. Similarly, the false alarms could have significant variations as they are caused by different clutter objects and varied environment conditions. These issues are more acute when data are collected by multiple sensors and/or using various features.

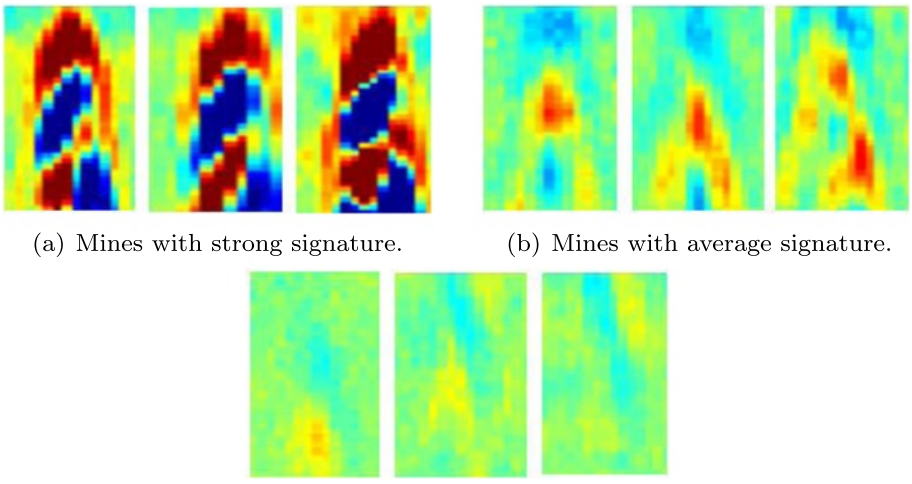
Consequently, learning a set of models that reflect different characteristics of the observations might be more beneficial than using one global model for each class. This is typical in many classifiers such as the K-NN [9], which uses different prototypes for each class, and the SVM [10], which uses multiple support vectors.

In this paper, we develop a new approach that replaces the two-model classifier with one that includes multiple models for each class. For instance, each group of signatures in Fig. 3 would be used to learn a different model. Our approach aims to capture the characteristics of the observations that would be lost under averaging in the two-model case.

We hypothesize that under realistic conditions, the data are generated by multiple models. The proposed approach, called ensemble HMM (eHMM), attempts to partition the training data within the log-likelihood space and identify multiple clusters in an unsupervised manner. Depending on each cluster's homogeneity and size, an appropriate training scheme is applied to learn the corresponding HMM parameters. The resulting K HMMs are then aggregated through a decision level fusion component to form a descriptive model for the data.

4 Ensemble HMM architecture

Let $\mathbb{O} = \{O_r, y_r\}_{r=1}^{R}$ be a set of R labeled sequences of length T where $O_r = \left\{O_r^{(1)}, \cdots, O_r^{(T)}\right\}$ and $y_r \in \{1, \cdots, C\}$ is the label (class) of sequence O_r. First, we need to identify subgroups of observations that have common patterns. Ground truth information could not be used for this task as it is insufficient and unreliable. For instance, a large deep buried mine can have a signature similar to a small shallow buried mine. Furthermore, the same mine buried at the same depth in soil with different properties may have different signatures. Thus, the partitioning needs to be done in an unsupervised way, i.e., regardless of the observation's labels and the limited ground truth information. In our approach, we use unsupervised learning to cluster the set of all observations, $\mathbb{O}$, into subgroups of "similar" observations. The first step in this approach is to define a measure of similarity between two observations.



(a) Mines with strong signature. (b) Mines with average signature.

(c) Mines with weak signature.

Fig. 3 Sample mine signatures manually categorized into three groups. **a** Mines with strong signature. **b** Mines with average signature. **c** Mines with weak signature

4.1 Similarity between observations in the log-likelihood space

4.1.1 Fitting individual models to sequences

Initially, each sequence in the training data, O_r, $1 \leq r \leq R$ is used to learn an HMM model λ_r. Even though using only one sequence of observations to learn an HMM might lead to over-fitting, this technique is only an intermediate step that aims to capture the characteristics of each sequence. The produced HMM model is meant to give a maximal description of each sequence, and therefore, over-fitting is not an issue in this context. In fact, it is desired that the model perfectly fits the observation sequence. In this case, the likelihood of each sequence with respect to its corresponding model is expected to be higher than those with respect to the remaining models.

Let $\left\{\lambda_r^{(0)}\right\}_{r=1}^{R}$ be the set of initial models and let $s_n^{(r)}$, $1 \leq n \leq N$, be the representative of each state in $\lambda_r^{(0)}$. Each model has N states. First, the model states can be assigned to the sequence observations either heuristically, using domain knowledge, or automatically by clustering the sequence observations into N clusters. In our approach, we use the latter and we define the states' means and observations as the center and elements of each resulting cluster, respectively. Consequently, the transition matrix and the initial probabilities of $\lambda_r^{(0)}$ are set according to the aforementioned associations. For the emission probabilities, the initialization differs whether we use the discrete or continuous HMM.

For the discrete case, the codewords $\{v_1, \cdots v_M\}$ of the initial individual DHMM model are the actual observations of the sequence $\{O_1, \cdots O_T\}$. The emission probability of each codeword in each state is inversely proportional to their distance to the mean of that state. We use

$$b_n(m) = \frac{\frac{1}{\|v_m - s_n\|}}{\sum_{l=1}^{N} \frac{1}{\|v_m - s_l\|}}, 1 \leq m \leq M, 1 \leq n \leq N. \quad (3)$$

To satisfy the requirement that $\sum_{m=1}^{M} b_n(m) = 1$, we normalize the values using

$$b_n(m) \longleftarrow \frac{b_n(m)}{\sum_{l=1}^{M} b_n(l)}, 1 \leq m \leq M, 1 \leq n \leq N. \quad (4)$$

In the continuous case, the emission probability density functions are modeled by mixtures of Gaussians. In the case of individual sequence models, as the number of observations is small, we use a single component mixture for each state. Thus, the observations belonging to each state are used to estimate the mean and covariance of that state's component. We use

$$\mu_n = \text{mean}\left\{O_t | O_t \in s_n\right\}, 1 \leq n \leq N, \quad (5)$$

and

$$\Sigma_n = \text{covariance}\left\{O_t | O_t \in s_n\right\}, 1 \leq n \leq N. \quad (6)$$

Then, the Baum-Welch algorithm [21] is used to adapt the model parameters to each given observation. Let $\{\lambda_r\}_{r=1}^{R}$ be the set of trained individual models.

Next, we need to define a measure that evaluates the similarity between pairs of observation sequences. While similarity between static data observations is straightforward and well defined, defining a similarity between observation sequences is more of a challenge. Within the context of HMM modeling, we consider two observation sequences similar if: (i) they fit each other's models; and (ii) they have similar Viterbi optimal paths [17].

4.1.2 Log-likelihood-based similarity

The log-likelihood, $\mathbf{L}(i,j)$, of sequence O_i being generated from model λ_j reflects the degree to which O_i fits λ_j and is defined as:

$$\mathbf{L}(i,j) = \log Pr\left(O_i|\lambda_j\right). \tag{7}$$

In (7), $\mathbf{L}$ can be computed using the forward–backward procedure mentioned in Section 2.1. When the log-likelihood value is high, it is likely that model λ_j generated sequence O_i. In this case, sequences O_i and O_j are expected to have common salient features and are considered to be similar. On the other hand, when the likelihood term is low, it is unlikely that model λ_j generated the sequence O_i. In this case, O_i and O_j are considered to be dissimilar. For each observation sequence O_r, $1 \leq r \leq R$, we compute its likelihood in each model λ_p, $Pr(O_r|\lambda_p)$, for $1 \leq p \leq R$. This will result in an $R \times R$ log-likelihood matrix.

4.1.3 Path-mismatch-based penalty

The likelihood-based similarity may not be always accurate. In fact, some observations can have high likelihood in a visually different model. This occurs when most of the elements of a sequence partially match only one or two of the states of the model. In this case, the observation sequence can have a high likelihood in the model but its optimal Viterbi path will deviate from the typical path. To alleviate this problem, we introduce a penalty term, $\mathbf{P}(i,j)$, to the log-likelihood measure that is related to the mismatch between the most likely sequence of hidden states of the test sequence (O_i) and that of the generating sequence (O_j), i.e.,

$$\mathbf{P}(i,j) = \mathbf{D_{Edit}}\left(Q^{(ji)}, Q^{(jj)}\right), \quad 1 \leq i,j \leq R. \tag{8}$$

In (8), $\mathbf{P}(i,j)$ is the distance between the Viterbi optimal path, $Q^{(ji)}$, of testing sequence O_i with model λ_j, and the Viterbi optimal path of testing sequence O_j with model λ_j, $Q^{(jj)}$. In (8), $\mathbf{D_{Edit}}$ is the "edit distance" [17] , commonly used in string comparisons. The "edit distance" between two strings, say p and q, is defined as the minimum number of single-character edit operations (deletions, insertions, and/or replacements) that would convert p into q. The Viterbi path mismatch term is intended to ensure that similar sequences have few mismatches in their corresponding Viterbi optimal paths. Since the Viterbi path is already available when using the forward–backward procedure for the likelihood computation, the penalty term does not require significant additional computation.

Finally, we define the similarity, $\mathbf{S}$, between two sequences O_i and O_j by combining (7) and (8):

$$\mathbf{S}(i,j) = \alpha\mathbf{L}(i,j) - (1-\alpha)\mathbf{P}(i,j). \tag{9}$$

In (9), the mixing factor, $\alpha \in [0,1]$, is a trade-off parameter between the log-likelihood-based similarity and the Viterbi-path-mismatch-based dissimilarity. It is estimated experimentally by maximizing the intra-class similarity and minimizing the inter-class similarity across the training data. A larger value of α corresponds to a dominant log-likelihood-based similarity where the need for the penalty mismatch is not significant. A smaller α corresponds to a more significant path mismatch penalty.

Using (9) to compute the similarity between all pairs of observations results in a similarity matrix that is not symmetric. Thus, we use the following three-step symmetrization scheme to transform it into a pairwise distance matrix:

$$\begin{cases} 1. & \mathbf{D}(i,j) = -\mathbf{S}(i,j) \quad 1 \leq i,j \leq R \\ 2. & \mathbf{D}(i,i) = 0, \quad 1 \leq i \leq R \\ 3. & \mathbf{D}(i,j) = \max(\mathbf{D}(i,j), \mathbf{D}(j,i)), \quad 1 \leq i,j \leq R. \end{cases} \tag{10}$$

4.2 Pairwise distance-based clustering

The distance matrix, computed using (10), reflects the degree to which pairs of sequences are considered similar. The largest variation is expected to be between sequences from different classes. Other significant variations may exist within the same class, e.g., the groups of signatures shown in Fig. 3. Our goal is to identify the similar groups so that one model can be learned for each group. This task can be achieved using any relational clustering algorithm. In our work, we use the standard agglomerative hierarchical algorithm [18].

Agglomerative hierarchical clustering is a bottom–up approach that starts with each data point as a cluster. It then proceeds by merging the most similar clusters to produce a sequence of clusters. Several measures have been used to assess the similarity between clusters [18]. Examples include single link, complete link, average link, and ward distance. The complete link method tends to produce a large number of small and compact clusters, while the single link method is known to result in few "elongated" clusters with large number of points. A compromise between the two is the minimum-variance distance, or ward distance [22]. This distance is defined as

$$d(i,j) = \frac{n_i n_j}{n_i + n_j} \left\| \mathbf{c_i} - \mathbf{c_j} \right\|^2 \tag{11}$$

where n_k and $\mathbf{c}_k$ are the cardinality and the centroid of cluster C_k, respectively. It has been shown in [17] that this approach merges the two clusters that lead to the smallest increase in the overall variance.

4.3 Ensemble HMM initialization and training

The previous clustering step results in K clusters, each comprised of potentially similar sequences. Each cluster is then used to learn an HMM, resulting in an ensemble of K HMMs. Let N_k denote the number of sequences assigned to the same cluster k. Since our clustering step did not use

class labels, clusters may include sequences from different classes. Let $N_k^{(c)}$ be the number of sequences in cluster k that belong to class c, such that $\sum_{c=1}^{C} N_k^{(c)} = N_k$. For instance, for the landmine example, if we let $c = 1$ denote the class of mines and $c = 0$ denote the class of clutter, $N_k^{(1)}$ would be the number of mines assigned to cluster k.

The next step of our approach consists of learning a set of HMMs that reflect the diversity of the training data. Since a cluster contains a set of similar sequences, and each cluster may include observations from different classes, we learn one HMM model $\left\{\lambda_k^{(c)}\right\}$ for each set of sequences assigned to class c within cluster k. Let $\mathbb{O}_k^{(c)} = \left\{O_r^{(c)}, y_r^{(c)}\right\}_{r=1}^{N_k^{(c)}}$ be the set of sequences partitioned into cluster k that belong to class c and let $\left\{\lambda_r^{(c)}\right\}_{r=1}^{N_k^{(c)}}$ be their corresponding individual HMM models, $c \in \{1, \cdots, C\}$.

For each cluster, we devise one of the following optimized training methods based on the cluster's size and homogeneity.

- **Clusters dominated by sequences from only one class:** In this case, we learn only one model for this cluster. The sequences within this cluster are presumably similar and belong to the same ground truth class, denoted C_i. We assume that this cluster is a representative of that particular dominating class. It is expected that the class conditional posterior probability is uni-modal and peaks around the MLE of the parameters. Thus, a maximum likelihood estimation would result in an HMM that best fits this particular class. For these reasons, we use the standard Baum-Welch re-estimation procedure [21]. Let K_1 be the number of homogenous clusters that fit into this category and let $\left\{\lambda_i^{(C_i)}, i = 1, \cdots, K_1\right\}$ denote the set of BW-trained models.
- **Clusters with a mixture of observations belonging to different classes:** In this case, it is expected that the posterior distribution of the classes is multi-modal. Thus, we need to learn one model for each class represented in this cluster. The MLE approach is not adequate, and more discriminative learning techniques such as genetic algorithms [23] or simulated annealing optimization [24] are needed to address the multimodality. In our work, we build a model for each class within the cluster. We focus on finding the class boundaries within the posteriors rather than trying to approximate a joint posterior probability. Thus, the models' parameters are jointly optimized to minimize the overall misclassification error using a discriminative learning approach [25]. Let K_2 be the number of mixed clusters that fit into this category and let $\left\{\lambda_j^{(c)}, j = 1, \cdots, K_2, c = 1, \cdots, C\right\}$ be the set of MCE-trained models.
- **Clusters containing a small number of sequences:** The MLE and MCE learning approaches need a large number of data points to give robust estimates of the model parameters. Thus, when a cluster has few samples, the above approaches may not be reliable. Ignoring these clusters is not a good option as they may contain information about sequences with distinctive characteristics. The Bayesian training framework [26], on the other hand, is suitable to learn model parameters using a small number of training sequences. Specifically, we select only the dominating class for this cluster and learn a single model using a variational Bayesian approach [26] to approximate the class conditional posterior distribution. Let K_3 be the number of small clusters that fit into this category and let $\left\{\lambda_k^{(C_k)}, k = 1, \cdots, K_3\right\}$ denote the set of Bayesian-trained models.

To summarize, for each homogenous cluster i, we define one model $\lambda_i^{(C_i)}$, $i = 1, \cdots, K_1$, for the dominating class C_i. For mixed clusters, we define C models per cluster: $\lambda_j^{(c)}$, $c = 1 \ldots C$, $j = 1, \cdots, K_2$. For each small cluster, we define one model $\lambda_k^{(C_k)}$ for the dominating class C_k. The ensemble HMM mixture is defined as $\left\{\lambda_k^{(c)}\right\}$, where $k \in \{1, \cdots, K\}$, and $c = C_k$ if cluster k is dominated by sequences labeled with class C_k, and $c \in \{1 \cdots, C\}$ if cluster k is a mixed cluster.

For simplicity, we assume that all models $\lambda_k^{(c)}$ have a fixed number of states N. For each model $\lambda_k^{(c)}$, the initialization step consists of assigning the priors, the initial states transition probabilities, and the states parameters (initial means and initial emission probabilities) using observations $O_r^{(c)}$ and their respective individual models $\lambda_r^{(c)}$, $r \in \left\{1, \cdots, N_k^{(c)}\right\}$. In particular, the initial values for the priors and the state transition probabilities are obtained by averaging, respectively, the priors and the state transition probabilities of the individual models $\lambda_r^{(c)}, r \in \left\{1, \cdots, N_k^{(c)}\right\}$. The initialization of the emission probabilities in each state, $b_n^{(k,c)}$, depends on whether the HMM is discrete or continuous.

- *Discrete HMM* (*DHMM*): the state representatives and the codebook of model $\lambda_k^{(c)}$ are obtained by partitioning and quantizing the observations $\mathbb{O}_k^{(c)}$. First, sequences from cluster k that belong to class c, $O_r^{(c)}$, are "unrolled" to form a vector of observations $\mathbf{U}^{(k,c)}$ of length $N_k^{(c)} T$. The state representatives, $s_n^{(k,c)}$, are obtained by clustering $\mathbf{U}^{(k,c)}$ into N clusters and taking the centroid of each cluster as the state

representative. Similarly, the codebook $\mathbf{V}^{(k,c)} = \left[v_1^{(k,c)}, \cdots, v_M^{(k,c)}\right]$ is obtained by clustering $\mathbf{U}^{(k,c)}$ into M clusters. For each symbol $v_m^{(k,c)}$, the membership in each state $s_n^{(k,c)}$ is computed using

$$b_n^{(k,c)}(m) = \frac{\frac{1}{\|v_m^{(k,c)} - s_n^{(k,c)}\|}}{\sum_{l=1}^{N} \frac{1}{\left\|v_m^{(k,c)} - s_l^{(k,c)}\right\|}}, 1 \le m \le M. \quad (12)$$

To satisfy the requirement $\sum_{m=1}^{M} b_n^{(k,c)}(m) = 1$, we scale the values by:

$$b_n^{(k,c)}(m) \longleftarrow \frac{b_n^{(k,c)}(m)}{\sum_{l=1}^{M} b_n^{(k,c)}(l)} \quad (13)$$

- *Continuous HMM* (*CHMM*): we assume that each state has N_g Gaussian components. For each model $\lambda_k^{(c)}$, as in the discrete case, we define a vector of observations, $\mathbf{U}^{(k,c)}$. First, $\mathbf{U}^{(k,c)}$ is partitioned into N clusters and the center of cluster n is taken as state $s_n^{(k,c)}$. Let $\mathbf{U}_n^{(k,c)}$ be the observations assigned to cluster n. Next, we partition $\mathbf{U}_n^{(k,c)}$ into N_g clusters using the k-means algorithm [27]. The mean of each component, $\mu_n^{(k,c,g)}$, is the center of one of the resulting clusters, and the covariance, $\Sigma_n^{(k,c,g)}$, is estimated using the observations that belong to that same cluster. If we denote by $\mathbf{U}_n^{(k,c,g)}$ the observations that belong to component g of state $s_n^{(k,c)}$, the parameters of $\lambda_k^{(c)}$ are computed using

$$\mu_n^{(k,c,g)} = \text{mean}\left\{\mathbf{U}_n^{(k,c,g)}\right\}, 1 \le n \le N, 1 \le g \le N_g. \quad (14)$$

$$\Sigma_n^{(k,c,g)} = \text{covariance}\left\{\mathbf{U}_n^{(k,c,g)}\right\}, 1 \le n \le N, 1 \le g \le N_g. \quad (15)$$

For both the discrete and continuous cases, any clustering algorithm, such as the K-means [27] or the fuzzy c-means [28], could be used to identify the states, codebook, or the multiple components. After initialization, we use one of the training schemes described earlier, to update $\lambda_k^{(c)}$ parameters using the respective observations $\mathbb{O}_k^{(c)}$, $k \in \{1, \cdots, K\}$, $c \in \{1, \cdots, C\}$. As mentioned earlier, for homogenous clusters, BW training results in one model λ^{BW} per cluster; for mixed clusters, MCE training results in C models per cluster, $\lambda_c^{MCE}, c = 1 \ldots C$; and for small clusters, variational Bayesian learning results in one model per cluster, λ^{VB}. The output of Baum-Welch- and VB-trained cluster models is $Pr(O|\lambda_k)$ while the output of the MCE-trained cluster models is $\max_c Pr\left(O|\lambda_{k,c}^{MCE}\right)$.

4.4 Decision level fusion

The partial confidence values of the different models need to be combined into a single confidence value. Let $\Lambda = \left\{\lambda_i^{BW}, \lambda_j^{MCE}, \lambda_k^{VB}\right\}$ be the resulting mixture model composed of a total of K models, $K = K_1 + K_2 + K_3$.

Let $\mathbf{F}(k,r) = \log Pr(O_r|\lambda_k), 1 \le r \le R, 1 \le k \le K$, be the log-likelihood matrix obtained by testing the R training sequences with the K models. Each column $\mathbf{f}_r$ of matrix $\mathbf{F}$ represents the feature vector of each sequence in the decision space (recall that $\mathbf{f}_r$ is a K-dimensional vector while O_r is a sequence of vector observations of length T). In other words, each column represents the confidences assigned by the K models to each sequence r. Therefore, the set of sequences $\mathbb{O} = \{O_r, y_r\}_{r=1}^{R}$ is mapped to a confidence space $\{\mathbf{f}_r, y_r\}_{r=1}^{R}$. Finally, a combination function, $\mathbb{H}$, takes all the $\mathbf{f}_r$'s as input and outputs the final decision. The general framework for fusing the K outputs is highlighted in Algorithm 1.

Algorithm 1 Testing a new sequence using the eHMM

Require: Test observation O

Ensure:

1: Compute $Pr\left(O|\lambda_i^{BW}\right)$ for the K_1 clusters learned with BW
2: Compute $Pr\left(O|\lambda_j^{MCE}\right) = \max_c Pr\left(O|\lambda_{j,c}^{MCE}\right)$, for the K_2 clusters learned with MCE
3: Compute $Pr\left(O|\lambda_k^{VB}\right)$ for the K_3 clusters learned with VB
4: Combine the outputs of the multiple models: $Pr(O|\Lambda) = \mathbb{H}\left(Pr\left(O|\lambda_i^{BW}\right), Pr\left(O|\lambda_j^{MCE}\right), Pr\left(O|\lambda_k^{VB}\right)\right)$

Several decision level fusion techniques such as simple algebraic [29], artificial neural networks (ANN) [30], and hierarchical mixture of experts (HME) [31] can be used. In our work, we use an ANN with a single-layer perceptron and no hidden layers. The ANN weights are learned from the labeled training data using the backpropagation algorithm [32].

The architecture of the proposed eHMM is summarized in Fig. 1. It is composed of four main components: similarity matrix computation, relational clustering, adaptive training scheme, and decision level fusion. To test a new sequence, the outputs of the different models are aggregated into a single confidence value using Algorithm 1.

5 Application to landmine detection using ground-penetrating radar data

5.1 Data collections

The proposed eHMM was implemented and tested on GPR data collected with a NIITEK vehicle mounted GPR

system [33] (see Fig. 4). This system collects 51 channels of data. Adjacent channels are spaced approximately five centimeters apart in the cross-track direction, and sequences (or scans) are taken at approximately 1-cm down-track intervals. Each A-scan, that is, the measured waveform that is collected in one channel at one down-track position, contains 416 time samples at which the GPR signal return is recorded. We often refer to the time index as depth although, since the radar wave is traveling through different media, this index does not represent a uniform sampling of depth. Thus, we model an entire collection of input data as a three-dimensional matrix of sample values, $S(z,x,y), z = 1,\cdots,416; x = 1,\cdots,51; y = 1,\cdots,N_S$, where N_S is the total number of collected scans, and the indices z, x, and y represent depth, cross-track position, and down-track positions respectively. A collection of scans, forming a volume of data, is illustrated in Fig. 5.

Figure 6 displays several B-scans (sequences of A-scans) both down-track (formed from a time sequence of A-scans from a single sensor channel) and cross-track (formed from each channel's response in a single sample). The surveyed object position is highlighted in each figure. The objects scanned are (a) a high-metal content antitank mine, (b) a low-metal antipersonnel mine, and (c) a wood block.

Raw GPR data needs to be preprocessed and prescreened. Preprocessing includes ground-level alignment and signal and noise background removal. Prescreening is needed to focus attention and identify regions with subsurface anomalies. For this step, we use the adaptive least mean squares (LMS) prescreener [34]. The LMS flags locations of interest utilizing a computationally inexpensive algorithm so that more advanced algorithms can be applied only on the small subsets of data flagged by the prescreener.

In our experiments, data sets are comprised of a variety of mine and background signatures. In particular, we use

Fig. 4 NIITEK vehicle mounted GPR system

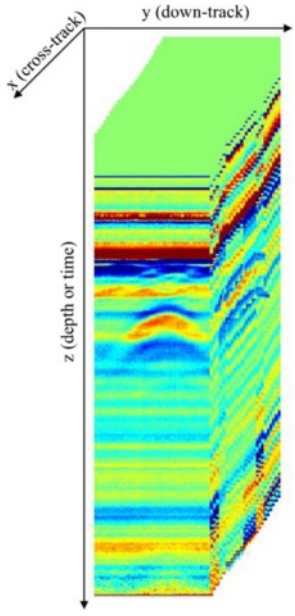


Fig. 5 A collection of few GPR scans

data collected from outdoor test lanes at three different locations. The first two locations, site 1 and site 2, were temperate regions with significant rainfall, whereas the third collection, site 3, was a desert region. The lanes are simulated roads with known mine locations. Multiple data collections were performed at each site at different dates. The statistics of these data sets are reported in Table 1. Data cubes of size (15 scans, 7 channels, 416 depths) were extracted from each scan position flagged by the prescreener and are presented to the classifier to discriminate between mines and false alarms.

5.2 Feature extraction

The goal of the feature extraction step is to transform original GPR data into a sequence of observation vectors. We use two types of features that have been proposed and used independently. Each feature represents a different interpretation of the raw data and aims at providing a good discrimination between mine and clutter signatures. These features are outlined in the following subsections.

5.2.1 EHD features

This feature is based on the edge histogram descriptor [9] (EHD) and characterizes edges in the spatial domain.

Table 1 Data collections

	Total number of prescreened alarms	Mine encounters	False alarms
$\mathfrak{D}_1$	2477	732	1745
$\mathfrak{D}_2$	1343	724	619
$\mathfrak{D}_3$	1843	613	1230

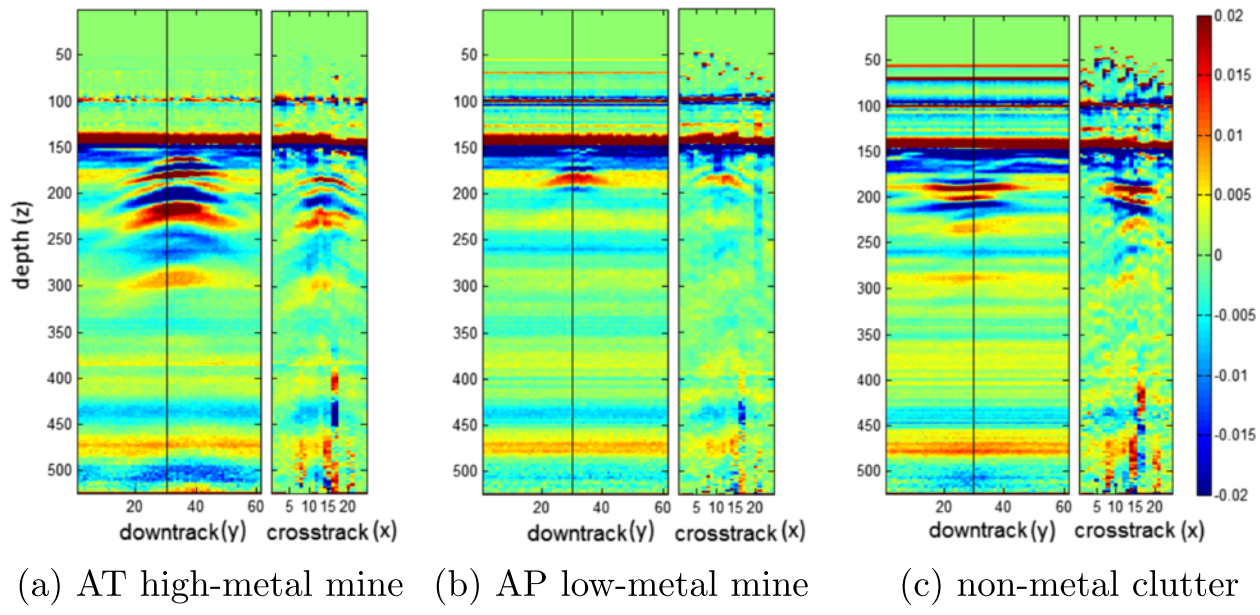


Fig. 6 NIITEK radar down-track and cross-track (at position indicated by a line in the down-track) B-scans pairs for **a** an Anti-Tank (AT) mine, **b** an Anti-Personnel (AP) mine, and **c** a non-metal clutter alarm

The EHD captures the salient properties of the 3D alarms in a compact and translation invariant representation. It extracts edge histograms capturing the frequency of occurrence of edge orientations in the data associated with a ground position. Simple edge detector operators are used to identify edges and group them into five categories: vertical, horizontal, diagonal, anti-diagonal, and isotropic (non-edge). Each B-scan position is then represented by a five-dimensional observation vector. Each dimension of this vector represents the percentage of pixels (in a small interval along the depth) that belong to each of the five edge categories.

5.2.2 Gabor features

Gabor features characterize edges in the frequency domain at multiple scales and orientations and are based on Gabor wavelets [7]. This feature is extracted by expanding the signature's B-scan (depth vs. down-track) using a bank of scale and orientation selective Gabor filters. Expanding a signal using Gabor filters provides a localized frequency description. In our experiments, we use a bank of filters tuned to the combination of three scales and four orientations. Each observation is then represented by a 12-dimension feature vector.

5.3 Ensemble HMM implementation and results

In all experiments reported in this paper, we use a sixfold cross validation for each data collection $\mathfrak{D}_l$, $l \in \{1,2,3\}$. For each fold, a subset of the data ($\mathfrak{D}_{l_{Trn}}$) is used for training and the remaining data ($\mathfrak{D}_{l_{Tst}}$) is used for testing. $\mathfrak{O}_{l_{Trn}^{Feat}}$ denotes the set of observation sequences extracted from dataset $\mathfrak{D}_l$, using one of the feature extraction methods, "Feat" (EHD or Gabor).

The first step of the eHMM is the similarity matrix computation. This step requires fitting an individual HMM model for each sequence in the training data $\mathfrak{O}_{l_{Trn}^{Feat}}$. Figure 7 shows the log-likelihood and path mismatch penalty matrices for a training collection that has 521 mines and 1471 clutter signatures (first training fold of $\mathfrak{D}_1$ using EHD features, $\mathfrak{O}_{1_{Trn1}^{EHD}}$). In these figures, the indices are rearranged so that the first entries correspond to the mine signatures and the latter ones correspond to non-mine signatures. As it can be seen, the matrices are composed mainly of four blocks. The diagonal blocks correspond to testing mine signatures in mine models and non-mine signatures in non-mine models, and the off-diagonal blocks correspond to testing mine signatures in non-mine models and non-mine signatures in mine models. In these figures, dark pixels correspond to small values of the log-likelihood or path mismatch penalty and bright pixels correspond to larger entries of the corresponding matrices. Note that in the case of the log-likelihood matrix in Fig. 7a, the diagonal blocks are brighter than the off-diagonal blocks. This means that the signatures from the same class are more similar to each other than to signatures from different classes. Similarly, in the path mismatch penalty matrix of Fig. 7b, the diagonal blocks are darker than the off-diagonal blocks. This means that when different mines are tested with each other models, the paths are similar. The above observations are trivial

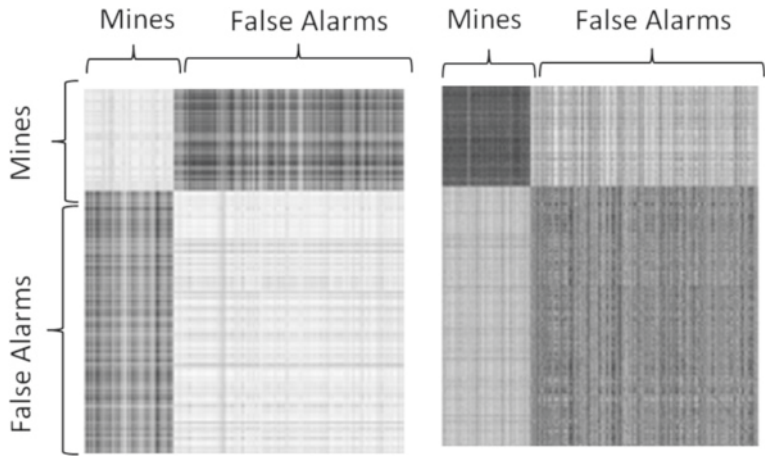


Fig. 7 Log-likelihood and path mismatch penalty matrices for a large collection of mine and clutter signatures. **a** Log-likelihood true and **b** path mismatch penalty

as alarms from the same class are expected to be more similar to each other than alarms from different classes. However, they could be used to validate our similarity (and penalty) measures in the log-likelihood space. A more important observation is that within each diagonal block, sub-blocks can be extracted. This is an indication of the existence of different clusters within the mines (and the clutter) themselves.

In the second step, the similarity matrix is transformed into a distance matrix, **D**, using (10). The hierarchical clustering algorithm [18] is then applied, using **D** with a fixed number of clusters $K = 10$, to identify sub-categories within the training data. For both the discrete and continuous versions, using any of the features and datasets, the eHMM clustering step successfully assigns groups of similar alarms into clusters. For instance, in Fig. 9, we show the hierarchical clustering results of the first crossvalidation fold of the eCHMM using the EHD features on dataset $\mathfrak{D}_1$. As it can be seen in Fig. 9a, we have a group of clutter dominated clusters (in brown) and a second group of clusters dominated by mines (in blue). In Fig. 8, we show sample signatures that belong to clusters 1, 6, and 10. As it can be seen from Figs. 8 and 9a, cluster 1 has only clutter and clusters 6 and 10 are composed exclusively of mine alarms. The mines that belong to cluster 6 have typically strong mine signatures. These mines, as shown in Fig. 9b, c, are typically mines with high metal content that are buried at shallow depths. The mines that belong to cluster 10 have weak GPR signatures. These mines, as shown in Fig. 9b, c, are typically mines with weak signatures that are either low metal mines or mines buried at deep depths.

Additional details of the clusters' contents per mine type and per burial depth are shown in Fig. 9b, c. To summarize, the training data includes four homogeneous clusters (Clusters 6, 7, and 10 contain only mines and cluster 1 has only clutter). The remaining clusters (2, 3, 4, 5, 8, and 9)

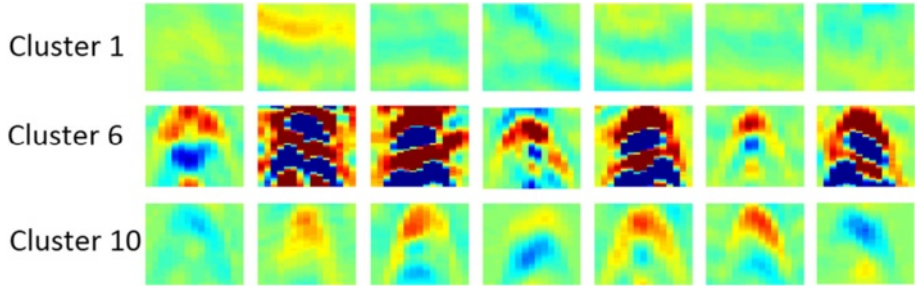


Fig. 8 Sample signatures from clusters 1, 6, and 10

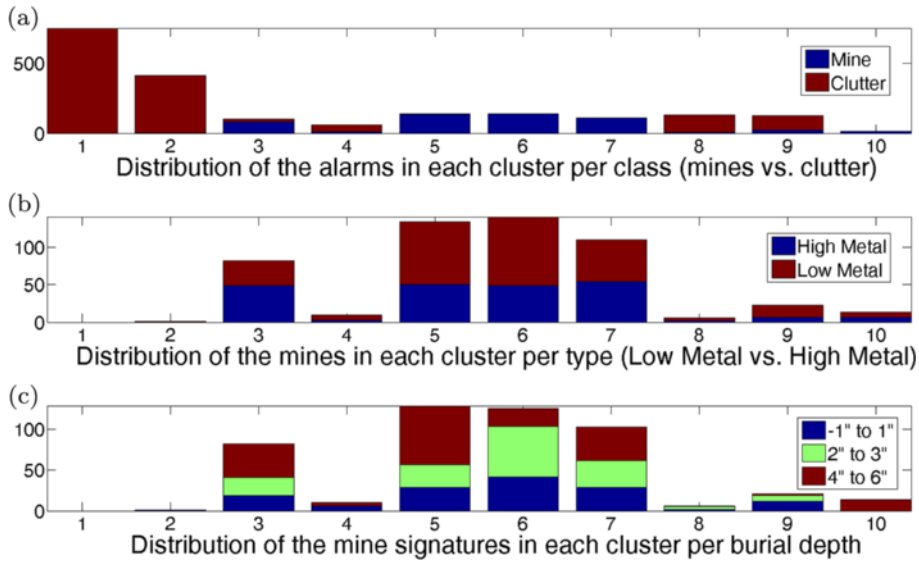


Fig. 9 eCHMM hierarchical clustering results of $\mathfrak{D}_{1_{Trn1}^{EHD}}$: distribution of the alarms in each cluster: **a** per class, **b** per type, **c** per depth

are mixed. Therefore, using the notation of Algorithm 1, we define our eHMM as:

$$\begin{cases} \lambda_i^{BW} = \left\{\lambda_1^{(F)}, \lambda_6^{(M)}, \lambda_7^{(M)}, \lambda_{10}^{(M)}\right\}, \\ \lambda_j^{MCE} = \left\{\lambda_j^{(M)}, \lambda_j^{(C)}\right\}, j \in \{2,3,4,5,8,9\}, \\ \lambda_k^{VB} = \emptyset. \end{cases} \tag{16}$$

In Table 2, we report the means and the weights of the components of each state of the BW-trained eCHMM model for cluster 6, $\lambda_6^{(M)}$, as well as its transition probability matrix. Cluster 6 contains "typical" mines that have strong-edge and near-perfect hyperbolic shape signatures with succession of states s_1, s_2, and s_3. Recall that s_1, s_2, and s_3 correspond respectively to the rising (Dg), flat (Hz), and falling (Ad) edges within the mine signature. Therefore, all the components of s_1 (resp. s_3) have their diagonal edge higher (resp. lower) than the anti-diagonal one. Similarly, components of s_2 have higher horizontal edge and comparable diagonal and anti-diagonal edges. As it can be seen in the transition matrix of Table 2, the probability of staying in s_1 (resp. s_2) is approximately three times (resp. two times) the probability of moving to s_2 (resp. s_3).

Table 2 λ_6^M CHMM model parameters of cluster 6

		Means					Weights	
		H	V	D	A	N		*w*
s_1	c_{11}	0.21	0.17	0.41	0.07	0.13	g_{11}	0.30
	c_{12}	0.36	0.12	0.25	0.11	0.17	g_{12}	0.22
	c_{13}	0.15	0.18	0.23	0.06	0.37	g_{13}	0.49
s_2	c_{21}	0.42	0.09	0.25	0.12	0.13	g_{21}	0.30
	c_{22}	0.37	0.10	0.10	0.30	0.13	g_{22}	0.20
	c_{23}	0.59	0.05	0.10	0.12	0.14	g_{23}	0.50
s_3	c_{31}	0.38	0.11	0.10	0.26	0.16	g_{31}	0.21
	c_{32}	0.20	0.17	0.07	0.43	0.13	g_{32}	0.30
	c_{33}	0.14	0.20	0.06	0.25	0.36	g_{33}	0.49

	A		
	s_1	s_2	s_3
s_1	0.73	0.27	0.00
s_2	0.00	0.67	0.33
s_3	0.00	0.00	1.00

Table 3 λ_{10}^M CHMM model parameters of cluster 10

		Means					Weights	
		H	V	D	A	N		*w*
s_1	c_{11}	0.14	0.13	0.17	0.08	0.48	g_{11}	0.27
	c_{12}	0.26	0.11	0.20	0.06	0.37	g_{12}	0.40
	c_{13}	0.16	0.04	0.10	0.05	0.66	g_{13}	0.32
s_2	c_{21}	0.30	0.07	0.10	0.14	0.39	g_{21}	0.50
	c_{22}	0.48	0.05	0.11	0.14	0.22	g_{22}	0.28
	c_{23}	0.27	0.12	0.07	0.37	0.16	g_{23}	0.21
s_3	c_{31}	0.09	0.11	0.03	0.18	0.59	g_{31}	0.60
	c_{32}	0.22	0.17	0.05	0.36	0.20	g_{32}	0.04
	c_{33}	0.10	0.20	0.02	0.31	0.36	g_{33}	0.36

	A		
	s_1	s_2	s_3
s_1	0.74	0.26	0.00
s_2	0.00	0.89	0.11
s_3	0.00	0.00	1.00

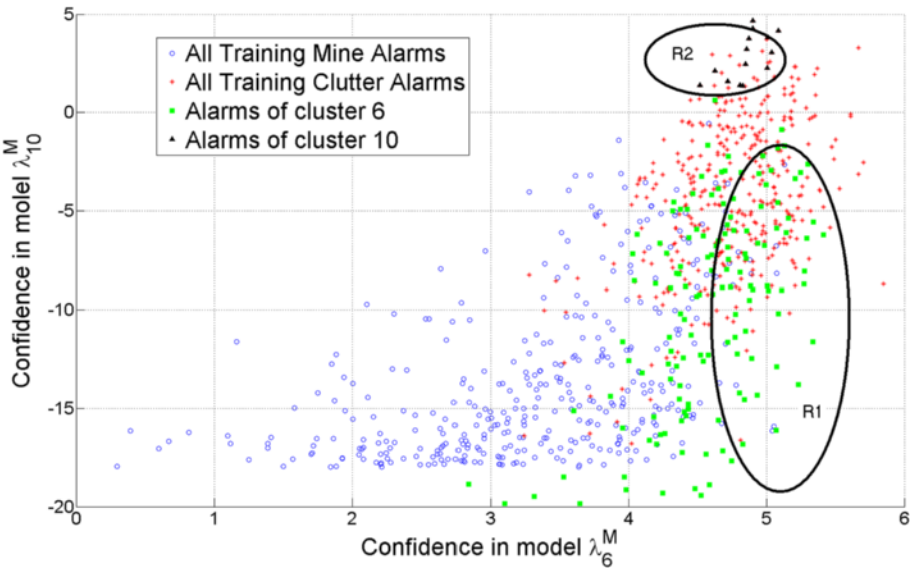


Fig. 10 Scatter plot of the confidences of the training data in cluster model $\lambda_6^{(M)}$ (strong mines) vs. cluster model $\lambda_{10}^{(M)}$ (weak mines)

Table 3 shows the BW-trained eCHMM model for cluster 10, $\lambda_{10}^{(M)}$. Recall that cluster 10 contains only mine signatures that have a low metal content and/or are buried at 4" or deeper, as it can be seen in Fig. 9c. Therefore, the alarms in cluster 10 are expected to have weak signatures and weak edge features. This could explain the large non-edge component of most of the states' means components of $\lambda_{10}^{(M)}$ reported in Table 3, compared to the non-edge components of $\lambda_6^{(M)}$'s states representatives. Nevertheless, the states representatives still characterize the hyperbolic shape of a typical mine signature, i.e., the succession of $Dg - Hz - Ad$ states. For instance, all s_1 components means have their diagonal D dimension larger than the anti-diagonal A dimension. For the transition matrix, we notice that $\lambda_{10}^{(M)}$ is more stationary in s_2, with a probability of 0.89, compared to $\lambda_6^{(M)}$. This means that, on average, sequences belonging to cluster 10 have a large number of observations with flat edge and fewer observations with strong diagonal or anti-diagonal edges.

Figure 10 shows the scatter plot of the confidences assigned by $\lambda_6^{(M)}$ and $\lambda_{10}^{(M)}$ to all the training data. In this figure, we display clutter and mine signatures that belong to each cluster using different symbols and colors. Even though the two models are dominated by mine signatures, we see that not all confidence values are highly correlated. On one hand, some strong mine signatures, particularly those belonging to cluster 6, have high log-likelihoods in model $\lambda_6^{(M)}$ and lower log-likelihoods in model $\lambda_{10}^{(M)}$ (lower right side of the scatter plot, region R1). This can be attributed to the fact that cluster 6 contains mainly strong mines and is more likely to yield high log-likelihood when testing a strong mine signature. On the other hand, in region R2, the performance of $\lambda_{10}^{(M)}$ is better as it gives higher likelihood values to the "weak" mines in that region, particularly those belonging to cluster 10.

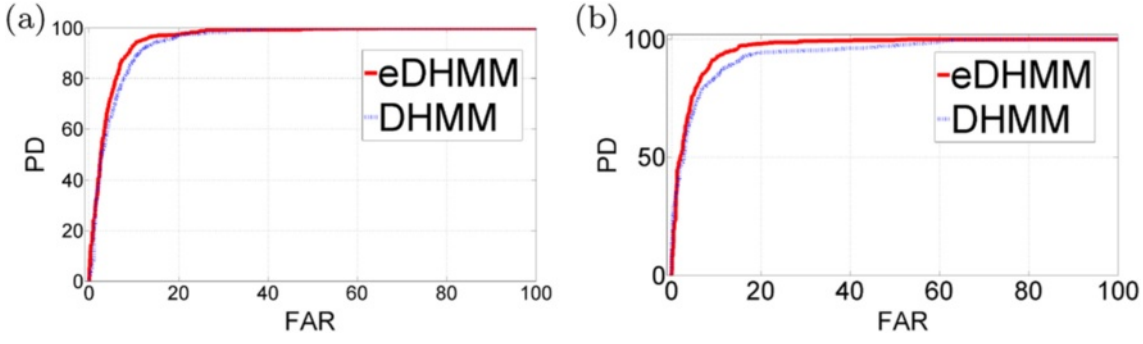


Fig. 11 ROCs generated by the eDHMM (*solid lines*) and baseline DHMM (*dashed lines*) classifiers using $\mathfrak{D}_1$ and **a** EHD, **b** Gabor features

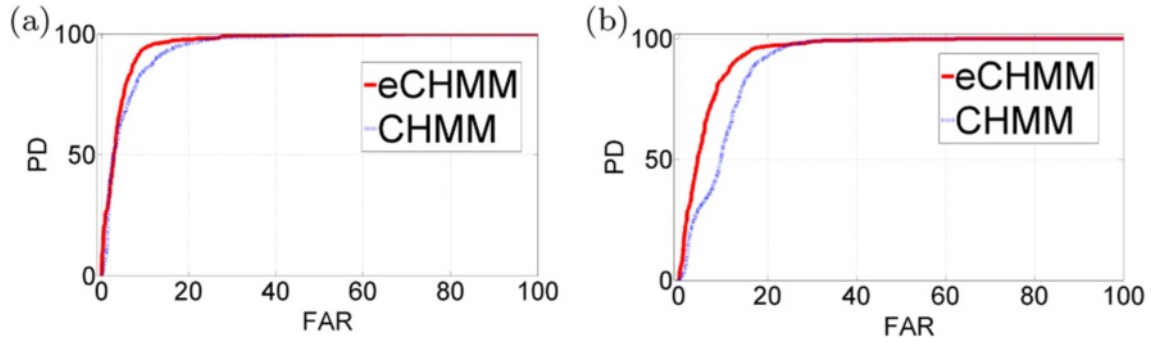


Fig. 12 ROCs generated by the eCHMM (*solid lines*) and baseline CHMM (*dashed lines*) classifiers using $\mathfrak{D}_1$ and **a** EHD, **b** Gabor features

In fact, this result is expected because cluster 10 contains weak mine signatures.

The main conclusion that we can draw from the above example is that $\lambda_6^{(M)}$ and $\lambda_{10}^{(M)}$ are very different and characterize two distinct subsets of the training data. The standard HMM approach would combine all alarms to learn a single model for mines (weak and strong) and a single model for clutter.

In the final step, the eHMM mixture is combined using a single-layer ANN. The ANN parameters are trained to fit the responses of the eHMM mixture models to the training data labels.

5.4 eHMM vs. baseline HMM results

In this section, we compare the performance of the proposed eHMM to the baseline HMM [5]. For the eHMM, we show the results using the ANN fusion and the hierarchical agglomerative clustering with $K = 20$. In Fig. 11a, b, we show the ROCs generated by the discrete versions of the eHMM and the baseline HMM on dataset $\mathfrak{D}_1$, using EHD and Gabor features. Similarly, in Fig. 12a, b, we report the ROCs generated by the continuous versions, i.e., the eCHMM and the baseline CHMM, on dataset $\mathfrak{D}_1$. As it can be seen, in all the ROCs of Figs. 11 and 12, at a given false alarm rate (FAR), the eHMM has a better probability of detecting targets. For instance, in Fig. 11a, at a FAR of 10 %, the eDHMM using EHD features successfully identifies 94 % of the mines while the baseline DHMM identifies only 87 % of the targets. At the same FAR of 10 %, the ROCs of Fig. 12a show that the eCHMM successfully identifies 95 % of the targets while the baseline CHMM probability of detection is 85 %.

The results for all three datasets are summarized in terms of the Area Under ROC Curve (AUC) and are reported in Table 4. As it can be seen, in all experiments, the eHMM outperforms the baseline HMM.

6 Conclusions

In this work, we have proposed a novel ensemble HMM classification method that is based on clustering sequences in the log-likelihood space. The eHMM uses multiple HMM models and fuses them for final decision making. We hypothesized that the data are generated by multiple models. These different models reflect the fact that samples from the same class can have different characteristics resulting in large intra-class variability.

The eHMM, in its discrete and continuous versions, was implemented and evaluated using large collections of landmine GPR data. We examined the intermediate steps of the eHMM and compared its performance to the baseline HMM. Results on three GPR data collections show that the proposed method can identify meaningful and coherent HMM mixture models that describe different properties of the data. Each individual HMM characterizes a group of data that share common attributes. The experiments show that the proposed eHMM intermediate results are inline with the expected behavior. The results

Table 4 AUC of the ensemble HMM and baseline HMM classifiers

Dataset	Classifier using:	EHD	Gabor
$\mathfrak{D}_1$	Ensemble DHMM	712	719
	Baseline DHMM	643	499
	Ensemble CHMM	718	617
	Baseline CHMM	614	472
$\mathfrak{D}_2$	Ensemble DHMM	402	127
	Baseline DHMM	107	30
	Ensemble CHMM	359	122
	Baseline CHMM	209	102
$\mathfrak{D}_3$	Ensemble DHMM	343	296
	Baseline DHMM	272	122
	Ensemble CHMM	326	226
	Baseline CHMM	284	140

also indicate that, for both the continuous and discrete versions, the proposed method outperforms the baseline HMM that uses one model for each class in the data.

Endnote

[1]The details of the landmine detection application using GPR signatures will be presented in section 5.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was supported in part by U.S. Army Research Office Grants Number W911NF-13-1-0066 and W911NF-14- 1-0589. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, or the U.S. Government.

Received: 8 April 2015 Accepted: 4 August 2015
Published online: 19 August 2015

References

1. *Landmine Monitor Report*, (2013). http://www.themonitor.org/
2. TR Witten, in *SPIE Conf Detection and Remediation Technologies for Mines and Minelike Targets III*. Present state of the art in ground-penetrating radars for mine detection (Orlando, FL, 1998), pp. 576–586
3. PD Gader, H Frigui, BN Nelson, G Vaillette, JM Keller, in *SPIE Conf Detection and Remediation Technologies for Mines and Minelike Targets IV*. New results in fuzzy set based detection of landmines with GPR (Orlando, FL, 1999), pp. 1075–1084
4. PD Gader, B Nelson, H Frigui, G Vaillette, JM Keller, Fuzzy logic detection of landmines with ground penetrating radar. Signal Process. Special Issue Fuzzy Logic Signal Process. **80**, 1069–1084 (2000)
5. PD Gader, M Mystkowski, Y Zhao, Landmine detection with ground penetrating radar using hidden Markov models. IEEE Trans. Geosci. Remote Sensing. **39**, 1231–1244 (2001)
6. H Frigui, DKC Ho, PD Gader, Real-time landmine detection with ground-penetrating radar using discriminative and adaptive hidden Markov models. EURASIP J. Appl. Signal Process. **12**, 1867–1885 (2005)
7. H Frigui, O Missaoui, PD Gader, in *SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets XII*. Landmine detection using discrete hidden Markov models with Gabor features (Louisville, KY, USA, 2007)
8. H Frigui, PD Gader, S Kotturu, in *SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets*. Detection and discrimination of landmines in ground penetrating radar using an eigenmine and fuzzy membership function approach, (2004). doi:10.1109/TFUZZ.2008.2005249
9. H Frigui, PD Gader, in *Proceedings of the IEEE International Conference on Fuzzy Systems*. Detection and discrimination of land mines based on edge histogram descriptors and fuzzy k-nearest neighbors (Vancouver, BC, Canada, 2006)
10. A Karem, A Fadeev, H Frigui, Gader, P, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Comparison of different classification algorithms for landmine detection using GPR, vol. 7664, (2010), p. 2. doi:10.1117/12.852257
11. PA Torrione, KD Morton, R Sakaguchi, LM Collins, Histograms of oriented gradients for landmine detection in ground-penetrating radar data. Geosci. Remote Sens. IEEE Trans. **52**(3), 1539–1550 (2014). doi:10.1109/TGRS.2013.2252016
12. O Missaoui, H Frigui, P Gader, in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*. Model level fusion of edge histogram descriptors and Gabor wavelets for landmine detection with ground penetrating radar, (2010), pp. 3378–3381. doi:10.1109/IGARSS.2010.5650350
13. O Missaoui, H Frigui, P Gader, in *Machine Learning and Applications, 2009. ICMLA '09. International Conference On*. Discriminative multi-stream discrete hidden Markov models, (2009), pp. 178–183. doi:10.1109/ICMLA.2009.121
14. O Missaoui, H Frigui, P Gader, Multi-stream continuous hidden Markov models with application to landmine detection. EURASIP J. Adv. Signal Process. **1** (2013). doi:10.1186/1687-6180-2013-40
15. CR Ratto, KD Morton, LM Collins, PA Torrione, in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*. A hidden Markov context model for GPR-based landmine detection incorporating stick-breaking priors, (2011), pp. 874–877. doi:10.1109/IGARSS.2011.6049270
16. LR Rabiner, in *Proceedings of the IEEE*. A tutorial on hidden Markov models and selected applications in speech recognition (San Francisco, CA, USA, 1989), pp. 257–286
17. S Theodoridis, K Koutroumbas, *Pattern Recognition, Fourth Edition*. (Academic Press, Inc, Orlando, FL, USA, 2009)
18. R Duda, P Hart, *Pattern Classification and Scene Analysis*. (John Wiley and Sons, New York, 1973)
19. LE Baum, T Petrie, Statistical Inference for Probability Functions of Finite State Markov Chains. Ann. Math. Stat. **37**, 1554–1563 (1966)
20. J Paisley, L Carin, Hidden Markov models with stick-breaking priors. Signal Process. IEEE Trans. **57**(10), 3905–3917 (2009). doi:10.1109/TSP.2009.2024987
21. LE Baum, T Petrie, Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. **37**(6), 1554–1563 (1966). doi:10.2307/2238772
22. JH Ward, Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. **58**(301), 236–244 (1963)
23. JH Holland, *Adaptation in Natural and Artificial Systems*. (MIT Press, Cambridge, MA, USA, 1992)
24. S Kirkpatrick, CD Gelatt, MP Vecchi, Optimization by Simulated Annealing. Science. **220**(4598), 671–680 (1983). doi:10.1126/science.220.4598.671. http://www.sciencemag.org/cgi/reprint/220/4598/671.pdf
25. B-H Juang, W Chou, C-H Lee, Minimum Classification Error Rate Methods for Speech Recognition. Trans. Speech Audio Process. **5**(3), 257–265 (1997)
26. DJC MacKay, *Information Theory, Inference and Learning Algorithms*. (Cambrdige University Press, New York, 2003)
27. AK Jain, RC Dubes, *Algorithms for Clustering Data*. (Prentice Hall, Upper Saddle River, NJ, USA, 1988)
28. JC Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. (Kluwer Academic Publishers, Norwell, MA, USA, 1981)
29. G Fumera, F Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 942–956 (2005). doi:10.1109/TPAMI.2005.109
30. DS Lee, SN Srihari, in *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*. A theory of classifier combination: the neural network approach (IEEE Computer Society Washington, DC, USA, 1995), p. 42
31. MI Jordan, RA Jacobs, in *NIPS*. Hierarchies of adaptive experts (Denver, 1991), pp. 985–992
32. DE Rumelhart, GE Hinton, RJ Williams, Learning internal representations by error propagation, 318–362 (1986). ISBN:0-262-68053-X
33. KJ Hintz, in *Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets IX*. SNR improvements in Niitek ground penetrating radar (Orlando, FL, USA, 2004)
34. PA Torrione, CS Throckmorton, LM Collins, Performance of an adaptive feature-based processor for a wideband ground penetrating radar system. Aerospace and Electronic Systems, IEEE Transactions on. **2**. pp. 644,658, April 2006