Migratory Logistic Regression for Learning Concept Drift Between Two Data Sets With Application to UXO Sensing

Xuejun Liao, *Senior Member, IEEE*, and Lawrence Carin, *Fellow, IEEE*

***Abstract*—To achieve good generalization in supervised learning, the training and testing examples are usually required to be drawn from the same source distribution. In this paper, we propose a method to relax this requirement in the context of logistic regression. Assuming $\mathcal{D}^p$ and $\mathcal{D}^a$ are two sets of examples drawn from two different distributions T and A (called concepts, borrowing a term from psychology), where $\mathcal{D}^a$ are fully labeled and $\mathcal{D}^p$ partially labeled, our objective is to complete the labels of $\mathcal{D}^p$. We introduce an auxiliary variable μ for each example in $\mathcal{D}^a$ to reflect its mismatch with $\mathcal{D}^p$. Under an appropriate constraint the μs are estimated as a byproduct, along with the classifier. We also present an active learning approach for selecting the labeled examples in $\mathcal{D}^p$. The proposed algorithm, called *migratory logistic regression*, is demonstrated successfully on simulated data as well as on real measured data of interest for unexploded ordnance cleanup.**

***Index Terms*—Concept drift, inverse problems, logistic regression, signal processing.**

I. Introduction

IN SUPERVISED classification problems, the goal is to design a classifier using the training examples (labeled data) such that the classifier correctly predicts the labels for unlabeled test data. The accuracy of the predictions is significantly affected by the quality of the training examples, which are assumed to contain essential information about the test instances for which predictions are desired. A common assumption utilized by learning algorithms is that the training examples and the test instances are drawn from the *same* source distribution.

As a practical example, consider the detection of a concealed entity based on sensor data collected in a noninvasive manner. This problem is of relevance in several practical problems, for example, in the medical imaging of potential tumors or other hidden anomalies. In the context of remote sensing, one is often challenged with the problem of detecting and characterizing a concealed (e.g., underground) target based on remotely collected sensor data. In an example that will be considered explicitly in this paper, consider the detection of buried unexploded ordnance (UXO) [1]. An UXO is a bomb that did not explode upon impact with the ground, and such items pose great danger if disturbed (excavated) without care. Sensors used for detecting and characterizing UXO include magnetometers and electromagnetic induction (EMI) [1]. In designing an algorithm for characterization of anomalies detected by such sensors, to determine if a given buried item is UXO or clutter, one typically requires training data. Such training data typically comes from other former bombing sites that have been cleaned, and there is a significant issue as to whether such extant labeled sensor data are relevant for a new site under test. The challenge addressed in this paper involves learning the relevance and relationship of existing labeled (training) data for analysis of a new unlabeled or partially labeled data set of interest. This type of problem has significant practical relevance for UXO sensing, for which results are presented on measured data, as well as for the aforementioned classes of problems, for which there is uncertainty concerning the appropriateness of existing labeled data for a new set of unlabeled data of interest.

To place this problem in a mathematical setting, let $\mathcal{T}(\mathbf{x}, y)$ be the probability distribution (or concept, borrowing a term from psychology)[1] from which test instances (each including a feature vector $\mathbf{x}$ and the associated class label y) are drawn. The goal in classifier design is to minimize the expected loss [3], [4], i.e.,

$$\min_{\zeta} \sum_{y} \int_{\mathbf{x}} L\left(y_i, \zeta(\mathbf{x})\right) \mathcal{T}(\mathbf{x}, y) \tag{1}$$

where $L(y, \zeta(\mathbf{x}))$ is a loss function, which provides a quantitative measure for the loss incurred by the classifier when it predicts $\zeta(\mathbf{x})$ for $\mathbf{x}$ whose true label is y. In practice, one only has access to N independent training examples $(\mathbf{x}, y)$ drawn from $\mathcal{T}(\mathbf{x}, y)$, therefore the minimization is performed to minimize the empirical loss [3], [4], i.e.,

$$\min_{\zeta} \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \zeta(\mathbf{x}_i)\right), \qquad \text{with } (\mathbf{x}_i, y_i) \sim \mathcal{T}(\mathbf{x}_i, y_i). \tag{2}$$

[1]Traditionally, the (probabilistic) mapping $\Pr(y|\mathbf{x})$ is called a concept, and $\Pr(\mathbf{x})$ is called a virtual concept (language describing the concept) [2]. For simplicity, usually, they are collectively called a concept.

Manuscript received June 15, 2007; revised August 4, 2008. First published March 16, 2009; current version published April 24, 2009. This work was supported by the Strategic Environmental Research and Development Program (SERDP).

X. Liao is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708-0291 USA (e-mail: xjliao@ee.duke.edu).

L. Carin is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708-0291 USA and also with Signal Innovations Group, Inc., Durham, NC 27703 USA (e-mail: lcarin@ee.duke.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.2008.2005268

The empirical loss is an Monte Carlo approximation of the expected loss and, therefore, will approach the expected loss when $N \to \infty$.

A learning algorithm based on the empirical loss minimization in (2) implicitly assumes that the training examples are also drawn from $\mathcal{T}(\mathbf{x}, y)$. It is this assumption that assures that the classifier generalizes to test instances when it is trained to minimize empirical loss on training examples. This assumption, however, is often violated in practice, since training examples and test instances may correspond to different collections of measurements (likely performed at different times under different experimental conditions) and the class memberships of the measurements may also change. These issues can introduce statistical *differences* between the training examples and the test instances; the UXO-sensing problem discussed above constitutes an important example for which the aforementioned statistical issues hold, concerning the utility of existing labeled (training) data.

Assume that one has training examples from a distribution $\mathcal{A}(\mathbf{x}, y)$ which is different from $\mathcal{T}(\mathbf{x}, y)$. For convenience of exposition, we call $\mathcal{T}(\mathbf{x}, y)$ the primary or target distribution and call $\mathcal{A}(\mathbf{x}, y)$ the auxiliary distribution. Accordingly, the examples drawn from $\mathcal{T}(\mathbf{x}, y)$ are called primary data, and the examples drawn from $\mathcal{A}(\mathbf{x}, y)$ are called auxiliary data.

In order to write the empirical loss for $\mathcal{T}$ in terms of examples drawn from $\mathcal{A}$, one may employ the technique of importance sampling [5]. By doing so, one makes the following modifications to the expression of empirical loss (2):

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{T}(\mathbf{x}_i, y_i)}{\mathcal{A}(\mathbf{x}_i, y_i)}L(y_i, \zeta(\mathbf{x}_i)), \qquad \text{with } (\mathbf{x}_i, y_i) \sim \mathcal{A}(\mathbf{x}_i, y_i) \quad (3)$$

where $\mathcal{T}(\mathbf{x}, y)/\mathcal{A}(\mathbf{x}, y)$ is the importance weight. It is known that the modification does not change the asymptotic behavior provided that $\mathcal{A}(\mathbf{x}, y)$ has the same nonzero support as $\mathcal{T}(\mathbf{x}, y)$.

Unfortunately, both $\mathcal{A}(\mathbf{x}, y)$ and $\mathcal{T}(\mathbf{x}, y)$ are unknown to the algorithm; all that is available are samples from $\mathcal{A}(\mathbf{x}, y)$. The challenge, therefore, is to learn a classifier on training examples drawn from $\mathcal{A}(\mathbf{x}, y)$ such that the resulting classifier still generalizes to test instances drawn from $\mathcal{T}(\mathbf{x}, y)$. Clearly, without assuming any knowledge about the relationship between $\mathcal{A}(\mathbf{x}, y)$ and $\mathcal{T}(\mathbf{x}, y)$, there is little one can do but to treat the training examples as if they are from the target distribution. This will, of course, introduce errors, which may be intolerable when the difference between $\mathcal{A}(\mathbf{x}, y)$ and $\mathcal{T}(\mathbf{x}, y)$ is large.

The problem of learning on examples from one distribution with the goal of generalizing to instances from a different distribution has been addressed in different contexts, using different names. In the following, we provide a brief review of this previous work.

A. Tackling Concept Drifts in Time-Varying Data

Many data come naturally in streams, collected over a period of time. Such applications include weather recordings, sales and customer data, surveillance video streams, to name a few. For streamed data, it is natural to consider online learning, in which the leaner is dynamically presented with the true label after it makes a prediction and updates its hypothesis based on newly received true labels. When the streamed data are recorded over an extended period of time, the statistics in the data are likely to change. The time-dependent variation of statistics in streamed data are termed concept drift in the literature [2], [6]–[9].

Concept drift falls under the general formulation in (3). Here, the target distribution characterizes the statistics of the most recent recordings, and there is an auxiliary distribution characterizing the recordings in each time interval in the past. The goal in concept-drift learning is to employ the available recordings to build up the target concept (i.e., the mapping from a feature vector to the associated class label) for the current moment. An important notion is the age of each recording, which determines the utility of the recording to the current prediction.

Three widely used methods for handling concept drift are the following: time windows [6], instance weighting [10], and ensemble learning [9]. In the first method, a time window is applied to the data stream, and the data within the window are employed to build the classifier for the current concept. The window keeps moving toward the future so that the most recent recording is always included in the window. The window acts like a limited memory, with the data outside the window forgotten. A key issue is to determine the window size, which should accurately capture the rate of concept drift. The method of instance weighting is based on the observation that the importance of a past example to the current concept does not abruptly change but rather gradually decreases. A weight is assigned to each past example to reflect its importance. Assuming that concept drift is monotonic (i.e., newer examples are always more important than older ones), one can construct the weights according to the age of each example; for example, one can choose weights that decrease exponentially with the age of examples [10]. The method of ensemble learning maintains an ensemble of classifiers, instead of a single one, for the current concept. This is done by dividing the data stream into chunks and learning a classifier based on each data chunk. The relevance of each classifier to the current concept is evaluated by the generalization error when applying the classifier to the most recent data chunk. The relevance is employed as a weight applied to each classifier and the weighted classifiers are employed to make predictions for the current data chunk.

B. Sample Selection Bias in Econometrics

In econometrics, the observed data are often a nonrandomly selected sample of the true distribution of interest. If the distribution of interest is $\mathcal{T}$, the selection bias results in samples drawn from $\mathcal{A}$ which is different from $\mathcal{T}$. Heckman [11] developed a method to correct the sample-selection bias for linear regression models. The basic idea of Heckman's method is that if one can estimate the probability of an observation being selected into the sample, one can use this probability estimate to correct the selection bias.

Heckman's model has recently been extended to classification problems [12], where it is assumed that the test instances are drawn from $\mathcal{T}(\mathbf{x}, y) = \Pr(\mathbf{x}, y)$ while the training examples are drawn from $\mathcal{A}(\mathbf{x}, y) = \Pr(\mathbf{x}, y|s = 1)$, where the variable s controls the selection of training examples: if $s = 1$,

$(\mathbf{x}, y)$ is selected into the training set; if $s = 0$, $(\mathbf{x}, y)$ is not selected into the training set. Evidently, unless s is independent of $(\mathbf{x}, y)$, $\Pr(\mathbf{x}, y|s = 1) \neq \Pr(\mathbf{x}, y)$, and hence, $\mathcal{T}(\mathbf{x}, y)$ is different from $\mathcal{T}(\mathbf{x}, y)$. By Bayes rule

$$\frac{\Pr(\mathbf{x}, y)}{\Pr(\mathbf{x}, y|s = 1)} = \frac{\Pr(s = 1)}{\Pr(s = 1|\mathbf{x}, y)} \tag{4}$$

or

$$\frac{\mathcal{T}(\mathbf{x}, y)}{\mathcal{A}(\mathbf{x}, y)} = \frac{\Pr(s = 1)}{\Pr(s = 1|\mathbf{x}, y)} \tag{5}$$

plugging this into (3), one has

$$\frac{1}{N}\sum_{i=1}^{N} \frac{\Pr(s = 1)}{\Pr(s = 1|\mathbf{x}, y)} L\left(y_i, \zeta(\mathbf{x}_i)\right) \qquad \text{with } (\mathbf{x}, y) \sim \Pr(\mathbf{x}, y|s = 1) \tag{6}$$

which implies that if one has access to $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y)$ one can correct the selection bias by using the empirical loss as expressed in (6). In the special case when $\Pr(s = 1|\mathbf{x}, y) = \Pr(s = 1|\mathbf{x})$, one may estimate $\Pr(s = 1|\mathbf{x})$ from a sufficient sample of $\Pr(\mathbf{x}, s)$ if such a sample is available [12]. In the general case, however, it is difficult to estimate $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y)$, as we do not have a sufficient sample of $\Pr(\mathbf{x}, y, s)$ (if we do, we already have a sufficient sample of $\Pr(\mathbf{x}, y)$, which contradicts the assumption of the problem).

C. Overview of This Work

In this paper, we propose an efficient algorithm for solving the general problem of learning on examples from $\mathcal{A}$ with the goal of generalizing to instances from $\mathcal{T}$, when $\mathcal{A}$ is different from $\mathcal{T}$. We consider the case in which we have a fully labeled auxiliary data set $\mathcal{D}^a$ and a partially labeled primary data set $\mathcal{D}^p = \mathcal{D}_l^p \cup \mathcal{D}_u^p$, where $\mathcal{D}_l^p$ are labeled and $\mathcal{D}_u^p$ unlabeled. We assume that $\mathcal{D}^p$ are examples of the primary concept $\mathcal{T}$ (the concept we are interested in) and $\mathcal{D}^a$ are examples of the auxiliary concept $\mathcal{A}$ (the one providing indirect and low-quality information about $\mathcal{T}$). Our objective is to use a mixed training set $\mathcal{D}^{\text{tr}} = \mathcal{D}_l^p \cup \mathcal{D}^a$ to train a classifier that predicts the labels of $\mathcal{D}_u^p$ accurately, with the hope that $\mathcal{D}_l^p$ is required to have a small number of examples.

Assume $\mathcal{D}^p \sim \Pr(\mathbf{x}, y)$. In light of (4), we can write $\mathcal{D}^a \sim \Pr(\mathbf{x}, y|s = 1)$ as long as the source distributions of $\mathcal{D}^p$ and $\mathcal{D}^a$ have the same support of nonzero probability.[2] As explained previously, it is difficult to correct the mismatch by directly estimating $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y)$. Therefore, we take an alternative approach. We introduce an auxiliary variable μ_i for each $(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a$ to reflect its mismatch with $\mathcal{D}^p$ and to control its participation in the learning process. The μs play a similar role as the weighting factors $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y)$ in (6). However, unlike the weighting factors, the auxiliary variables are estimated along with the classifier in the learning. We employ logistic regression as a specific classifier and develop our method in this context.

The remainder of this paper is organized as follows. A detailed description of the proposed method is provided in Section II, followed by description of a fast learning algorithm in Section III and a theoretical discussion in Section IV. In Section V, we present a method to actively define $\mathcal{D}_l^p$ when $\mathcal{D}_l^p$ is initially empty. In Section VI, we demonstrate the ideas presented here using simulated data, as well as real data of interest for detecting UXO. Finally, Section IX provides conclusions.

II. MigLogit: Learning Jointly on the Primary and Auxiliary Data

We assume $\mathcal{D}_l^p$ are fixed and nonempty, and without loss of generality, we assume $\mathcal{D}_l^p$ are always indexed prior to $\mathcal{D}_u^p$, i.e., $\mathcal{D}_l^p = \{(\mathbf{x}_i^p, y_i^p)\}_{i=1}^{N_l^p}$ and $\mathcal{D}_u^p = \{(\mathbf{x}_i^p, y_i^p) \colon y_i^p \text{ missing}\}_{i=N_l^p+1}^{N^p}$. We use N^a, N^p, and N_l^p to denote the size (number of data points) in $\mathcal{D}^a$, $\mathcal{D}^p$, and $\mathcal{D}_l^p$, respectively. In Section V, we discuss how to actively determine $\mathcal{D}_l^p$ when $\mathcal{D}_l^p$ is initially empty. We consider the binary classification problem and the labels $y^a, y^p \in \{-1, 1\}$. For notational simplicity, we let $\mathbf{x}$ always include a 1 as its first element to accommodate a bias (intercept) term, thus $\mathbf{x}^p, \mathbf{x}^a \in \mathbb{R}^{d+1}$ where d is the number of features. For a primary data point $(\mathbf{x}_i^p, y_i^p) \in \mathcal{D}_l^p$, we follow standard logistic regression to write

$$\Pr\left(y_i^p \middle| \mathbf{x}_i^p; \mathbf{w}\right) = \sigma\left(y_i^p \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^p\right) \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^{d+1}$ is a column vector of classifier parameters and $\sigma(\eta) = 1/(1 + \exp(-\eta))$ is the sigmoid function. For a auxiliary data point $(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a$, we define

$$\Pr\left(y_i^a \middle| \mathbf{x}_i^a; \mathbf{w}, \mu_i\right) = \sigma\left(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + y_i^a \mu_i\right) \tag{8}$$

where μ_i is an auxiliary variable. Assuming the examples in $\mathcal{D}_l^p$ and $\mathcal{D}^a$ are drawn independent identically distributed, we have the log-likelihood function

$$\ell\left(\mathbf{w}, \boldsymbol{\mu}; \mathcal{D}_l^p \cup \mathcal{D}^a\right) = \sum_{i=1}^{N_l^p} \ln \sigma\left(y_i^p \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^p\right) + \sum_{i=1}^{N^a} \ln \sigma\left(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + y_i^a \mu_i\right) \tag{9}$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{N^a}]^{\mathrm{T}}$ is a column vector of all auxiliary variables.

The auxiliary variable μ_i is introduced to reflect the mismatch of $(\mathbf{x}_i^a, y_i^a)$ with $\mathcal{D}^p$ and to control its participation in the learning of $\mathbf{w}$. A larger $y_i^a \mu_i$ makes $\Pr(y_i^a | \mathbf{x}_i^a; \mathbf{w}, \mu_i)$ less sensitive to $\mathbf{w}$. When $y_i^a \mu_i = \infty$, $\Pr(y_i^a | \mathbf{x}_i^a; \mathbf{w}, \mu_i) = 1$ becomes completely independent of $\mathbf{w}$. Geometrically, the μ_i is an extra intercept term that is uniquely associated with $\mathbf{x}_i^a$ and causes it to migrate toward class y_i^a. If $(\mathbf{x}_i^a, y_i^a)$ is mismatched with the primary data $\mathcal{D}^p$, $\mathbf{w}$ cannot make $\sum_{i=1}^{N_l^p} \ln \sigma(y_i^p \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^p)$ and $\ln \sigma(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a)$ large at the same time. In this case, $\mathbf{x}_i^a$ will be given an appropriate μ_i to allow it to migrate toward class y_i^a, so that $\mathbf{w}$ is less sensitive to $(\mathbf{x}_i^a, y_i^a)$ and can focus more on fitting $\mathcal{D}_l^p$. Evidently, if the μ's are allowed to change freely,

[2]For any $\Pr(\mathbf{x}, y|s = 1) \neq 0$ and $\Pr(\mathbf{x}, y) \neq 0$, there exists $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y) = \Pr(\mathbf{x}, y)/\Pr(\mathbf{x}, y|s = 1) \in (0, \infty)$ such that (4) is satisfied. For $\Pr(\mathbf{x}, y|s = 1) = \Pr(\mathbf{x}, y) = 0$, any $\Pr(s = 1)/\Pr(s = 1|\mathbf{x}, y) \neq 0$ makes (4) satisfied.

their influence will override that of $\mathbf{w}$ in fitting the auxiliary data $\mathcal{D}^a$ and then $\mathcal{D}^a$ will not participate in learning $\mathbf{w}$. To prevent this from happening, we introduce constraints on μ_i and maximize the log-likelihood subject to the constraints

$$\max_{\mathbf{w},\boldsymbol{\mu}} \quad \ell\left(\mathbf{w}, \boldsymbol{\mu}; \mathcal{D}_l^p \cup \mathcal{D}^a\right) \tag{10}$$

$$\text{subject to} \quad \frac{1}{N^a}\sum_{i=1}^{N^a} y_i^a \mu_i \leq C, \qquad C \geq 0 \tag{11}$$

$$y_i^a \mu_i \geq 0, \qquad i = 1, 2, \ldots, N^a \tag{12}$$

where the inequalities in (12) reflect the fact that in order for $\mathbf{x}_i^a$ to fit $y_i^a = 1$ (or $y_i^a = -1$), we need to have $\mu_i > 0$ (or $\mu_i < 0$), if we want μ_i to exert a *positive* influence in the fitting process. Under the constraints in (12), a larger value of $y_i^a \mu_i$ represents a larger mismatch between $(\mathbf{x}_i^a, y_i^a)$ and $\mathcal{D}^p$ and, accordingly, makes $(\mathbf{x}_i^a, y_i^a)$ play a less important role in determining $\mathbf{w}$. The classifier resulting from solving the problem in (10)–(12) is referred to as *migratory logistic regression* (MigLogit).

The C in (11) reflects the average mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$ and controls the average participation of $\mathcal{D}^a$ in determining $\mathbf{w}$. It can be learned from data if we have a reasonable amount of $\mathcal{D}_l^p$. However, in practice, we usually have no or very scarce $\mathcal{D}_l^p$ to begin with. In this case, we must rely on other information to set C. We will come back to a more detailed discussion on C in Section IV.

III. Fast Learning Algorithm

The optimization problem in (10)–(12) is concave and any standard technique can be utilized to find the global maxima. However, there is a unique μ_i associated with every $(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a$, and when $\mathcal{D}^a$ is large, using a standard method to estimate μs can consume most of the computational time.

In this section, we give a fast algorithm for training MigLogit, by taking a block-coordinate ascent approach [13], in which we alternately solve for $\mathbf{w}$ and $\boldsymbol{\mu}$, keeping one fixed when solving for the other. The algorithm draws its efficiency from the analytic solution of $\boldsymbol{\mu}$, which we establish in the following theorem. Proof of the theorem is given in the Appendix, and Section IV contains a discussion that helps to understand the theorem from an intuitive perspective.

Theorem 1: Let $f(z)$ be a twice continuously differentiable function with its first derivative $f'(z) >= 0$ and its second derivative $f''(z) < 0$ for any $z \in \mathbb{R}$. Let $b_1 \leq b_2 \leq \cdots \leq b_N$, $R \geq 0$, and

$$n = \max\left\{m : m b_m - \sum_{i=1}^{m} b_i \leq R, 1 \leq m \leq N\right\}. \tag{13}$$

Then, the problem

$$\max_{\{z_i\}} \quad \sum_{i=1}^{N} f(b_i + z_i) \tag{14}$$

$$\text{subject to} \quad \sum_{i=1}^{N} z_i \leq R, \qquad R \geq 0 \tag{15}$$

$$z_i \geq 0, \qquad i = 1, 2, \ldots, N \tag{16}$$

TABLE I
Fast Learning Algorithm of MigLogit

```
Input: $\mathcal{D}^a \cup \mathcal{D}_l^p$ and $C$;  Output: $\mathbf{w}$ and $\{\mu_i\}_{i=1}^{N^a}$
1. Initialize $\mathbf{w}$ and $\mu_i = 0$ for $i = 1, 2, \cdots, N^a$.
2. Compute the gradient $\nabla_{\mathbf{w}}\ell$ and Hessian matrix $\nabla^2_{\mathbf{w}}\ell$.
3. Compute the ascent direction $\mathbf{d} = -(\nabla^2_{\mathbf{w}}\ell)^{-1}\nabla_{\mathbf{w}}\ell$.
4. Do a linear search for the step-size $\alpha^* = \arg\max_\alpha \ell(\mathbf{w} + \alpha\mathbf{d})$.
5. Update $\mathbf{w}$: $\mathbf{w} \leftarrow \mathbf{w} + \alpha^*\mathbf{d}$.
6. Sort $\{y_i^a \mathbf{w}^T \mathbf{x}_i^a\}_{i=1}^{N^a}$ in ascending order. Assume the result is
   $y_{k_1}^a \mathbf{w}^T \mathbf{x}_{k_1}^a \leq y_{k_2}^a \mathbf{w}^T \mathbf{x}_{k_2}^a \leq \cdots \leq y_{k_{N^a}}^a \mathbf{w}^T \mathbf{x}_{k_{N^a}}^a$, where
   $k_1, k_2, \cdots, k_{N^a}$ is a permutation of $1, 2, \cdots, N^a$.
7. Find the $n$ using (19).
8. Update the auxiliary variables $\{\mu_i\}_{i=1}^{N^a}$ using (18).
9. Check the convergence of $\ell$: exit and output $\mathbf{w}$ and $\{\mu_i\}_{i=1}^{N^a}$ if
   converged; go back to 2 otherwise.
```

has a unique global solution

$$z_i = \begin{cases} \frac{1}{n}\sum_{j=1}^{n} b_j + \frac{1}{n}R - b_i, & 1 \leq i \leq n \\ 0, & n < i \leq N. \end{cases} \tag{17}$$

For a fixed $\mathbf{w}$, the problem in (10)–(12) is simplified to maximizing $\sum_{i=1}^{N^a} \ln\sigma(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + y_i^a \mu_i)$ with respect to $\boldsymbol{\mu}$, subject to $(1/N^a)\sum_{i=1}^{N^a} y_i^a \mu_i \leq C$, $C \geq 0$, and $y_i^a \mu_i \geq 0$ for $i = 1, 2, \ldots, N^a$. Clearly, $\ln\sigma(z)$ is a twice continuously differentiable function of z with its first derivative $(\partial/\partial z)\ln\sigma(z) = \sigma(-z) > 0$ and its second derivative $(\partial^2/\partial z^2)\ln\sigma(z) = -\sigma(z)\sigma(-z) < 0$ for $-\infty < z < \infty$. Thus, Theorem 1 applies. We first solve $\{y_i^a \mu_i\}$ using Theorem 1, then $\{\mu_i\}$ are trivially solved using the fact $y_i^a \in \{-1, 1\}$. Assume $y_{k_1}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_1}^a \leq y_{k_2}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_2}^a \leq \cdots \leq y_{k_{N^a}^a} \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_{N^a}^a}$, where $k_1, k_2, \ldots, k_{N^a}$ is a permutation of $1, 2, \ldots, N^a$. Then, we can analytically write the solution of $\{\mu_i\}$

$$\mu_{k_i} = \begin{cases} \frac{1}{n} y_{k_i}^a \sum_{j=1}^{n} y_{k_j}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_j}^a \\ \quad + \frac{N^a}{n} y_{k_i}^a C - \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_i}^a, & 1 \leq i \leq n \\ 0, & n < i \leq N^a \end{cases} \tag{18}$$

where

$$n = \max\left\{m : m y_{k_m}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_m}^a - \sum_{i=1}^{m} y_{k_i}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_i}^a \leq N^a C, \right.$$
$$\left. 1 \leq m \leq N^a \right\}. \tag{19}$$

For a fixed $\boldsymbol{\mu}$, we use the standard gradient-based method [13] to find $\mathbf{w}$. The main procedures of the fast training algorithm for MigLogit are summarized in Table I, where the gradient $\nabla_{\mathbf{w}}\ell$ and the Hessian matrix $\nabla^2_{\mathbf{w}}\ell$ are computed from (9).

It is seen from Table I that for a single iteration, MigLogit exactly requires the same amount of computation as standard logistic regression except for the computation of $\boldsymbol{\mu}$. Since the computation of $\boldsymbol{\mu}$ involves only a sorting and some arithmetic operations, it is not difficult to verify that the computational complexity of MigLogit is on the same order as that of standard logistic regression. Although we have not theoretically established that the convergence of MigLogit is no slower than that of standard logistic regression, we have not found noticeable difference in our experiments regarding the convergence of the two methods.

IV. Auxiliary Variables and Choice of C

Theorem 1 and its constructive proof in the appendix offers some insight into the mechanism of how the mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$ is compensated through the auxiliary variables $\{\mu_i\}$. To make the description easier, we think of each data point $\mathbf{x}_i^a \in \mathcal{D}^a$ as getting principal wealth $y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a$ from $\mathbf{w}$ and additional wealth $y_i^a \mu_i$ from a given budget totaling $N^a C$ (C represents the average budget for a single $\mathbf{x}^a$). From the Appendix, $N^a C$ is distributed among the auxiliary data $\{\mathbf{x}_i^a\}$ by a "poorest-first" rule: the poorest $\mathbf{x}_{k_1}^a$ (that which has the smallest $y_{k_1}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_1}^a$), gets a portion $y_{k_1}^a \mu_{k_1}$ from $N^a C$ first, and when the total wealth $y_{k_1}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_1}^a + y_{k_1}^a \mu_{k_1}$ reaches the principal wealth of the second poorest $\mathbf{x}_{k_2}^a$, $N^a C$ becomes equally distributed to $\mathbf{x}_{k_1}^a$ and $\mathbf{x}_{k_2}^a$ such that their total wealth are always equal. Then, when $y_{k_1}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_1}^a + y_{k_1}^a \mu_{k_1} = y_{k_2}^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_{k_2}^a + y_{k_2}^a \mu_{k_2}$ reach the principal wealth of the third poorest, $N^a C$ becomes equally distributed to three of them to make them equally wealthy. The distribution continues in this way until the budget $N^a C$ is used up. The "poorest-first" rule is essentially a result of the monotonically increasing and concave function $\ln \sigma(\cdot)$. The goal is to maximize $\sum_{i=1}^{N^a} \ln \sigma(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + y_i^a \mu_i)$. The monotonically increasing and concave $\ln \sigma(\cdot)$ dictates that for any given portion of $N^a C$, distributing it to the poorest makes the maximum gain in $\ln \sigma$.

The C is used as a means to compensate for the loss that $\mathcal{D}^a$ may suffer from $\mathbf{w}$. The classifier $\mathbf{w}$ is responsible for correctly classifying both $\mathcal{D}^a$ and $\mathcal{D}^p$. Because $\mathcal{D}^a$ and $\mathcal{D}^p$ are mismatched, $\mathbf{w}$ cannot satisfy both of them: One must suffer if the other is to gain. As $\mathcal{D}^p$ is the primary data set, we want $\mathbf{w}$ to classify $\mathcal{D}^p$ as accurately as possible. The auxiliary variables are therefore introduced to represent compensations that $\mathcal{D}^a$ get from C. When $\mathbf{x}^a$ gets small contribution from $\mathbf{w}$ and is small, it is because $\mathbf{x}^a$ is mismatched and in conflict with $\mathcal{D}^p$ (assuming perfect separation of $\mathcal{D}^a$, no conflict exists among themselves). By the "poorest-first" rule, the most mismatched $\mathbf{x}^a$ gets compensation first.

A high compensation $y_i^a \mu_i$ whittles down the participation of $\mathbf{x}_i^a$ in learning $\mathbf{w}$. This is readily seen from the contribution of $(\mathbf{x}_i^a, y_i^a)$ to $\nabla_{\mathbf{w}} \ell$ and $\nabla_{\mathbf{w}}^2 \ell$, which are obtained from (9) as $\sigma(-y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a - y_i^a \mu_i) y_i^a \mathbf{x}_i^a$ and $-\sigma(-y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a - y_i^a \mu_i) \sigma(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + y_i^a \mu_i) \mathbf{x}_i^a \mathbf{x}_i^{aT}$, respectively. When $y_i^a \mu_i$ is large, $\sigma(-y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a - y_i^a \mu_i)$ is close to zero, and hence, the contribution of $(\mathbf{x}_i^a, y_i^a)$ to $\nabla_{\mathbf{w}} \ell$ and $\nabla_{\mathbf{w}}^2 \ell$ are ignorable. We, in fact, do not need an infinitely large $y_i^a \mu_i$ to make the contributions of $\mathbf{x}_i^a$ ignorable, because $\sigma(\mu)$ is almost saturated at $\mu = \pm 6$. If $y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a = -6$, $\sigma(-y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a) = 0.9975$, implying a large contribution of $(\mathbf{x}_i^a, y_i^a)$ to $\nabla_{\mathbf{w}} \ell$, which happens when $\mathbf{w}$ assigns $\mathbf{x}_i^a$ to the correct class y_i^a with probability of $\sigma(y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a) = \sigma(-6) = 0.0025$ only. In this nearly worst case, a compensation of $y_i^a \mu_i = 12$ can effectively remove the contribution of $(\mathbf{x}_i^a, y_i^a)$ because $\sigma(-y_i^a \mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a - y_i^a \mu_i) = \sigma(6 - 12) = \sigma(-6) = 0.0025$. To effectively remove the contributions of N^m auxiliary data, one needs a total budge $12N^m$, resulting in an average budget $C = 12N^m/N^a$.

To make a right choice of C, the N^m/N^a should represent the rate that $\mathcal{D}^a$ are mismatched with $\mathcal{D}^p$. This is because we want $N^a C$ to be distributed only to that part of $\mathcal{D}^a$ that is mismatched with $\mathcal{D}^p$, thus permitting us to use the remaining part in learning $\mathbf{w}$. The quantity N^m/N^a is usually unknown in practice. However, $C = 12N^m/N^a$ gives one a sense of at least what range C should be in. As $0 \le N^m \le N^a$, letting $0 \le C \le 12$ is usually a reasonable choice. In our experiences, the performance of MigLogit is relatively robust to C, as demonstrated in Section VI-B.

V. Active Selection of $\mathcal{D}_l^p$

In Section II, we assumed that $\mathcal{D}_l^p$ had already been determined. In this section, we describe how $\mathcal{D}_l^p$ can actively be selected from $\mathcal{D}^p$, based on the Fisher information matrix [14], [15]. The approach is known as active learning [16], [17].

Let $\mathbf{Q}$ denote the Fisher information matrix of $\mathcal{D}_l^p \cup \mathcal{D}^a$ about $\mathbf{w}$. By definition of the Fisher information matrix [18], $\mathbf{Q} = \mathbb{E}_{\{y_i^p\},\{y_i^a\}} (\partial \ell / \partial \mathbf{w})(\partial \ell / \partial \mathbf{w})^{\mathrm{T}}$, and substituting (9) into this equation gives (a brief derivation is given in the Appendix)

$$\mathbf{Q} = \sum_{i=1}^{N_l^p} \sigma_i^p (1 - \sigma_i^p) \mathbf{x}_i^p \mathbf{x}_i^{pT} + \sum_{i=1}^{N^a} \sigma_i^a (1 - \sigma_i^a) \mathbf{x}_i^a \mathbf{x}_i^{aT} \quad (20)$$

where $\sigma_i^p = \sigma(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i^p)$ for $i = 1, 2, \ldots, N_l^p$, and $\sigma_i^a = \sigma(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i^a + \mu_i)$ for $i = 1, 2, \ldots, N^a$, and $\mathbf{w}$ and $\{\mu_i\}$ represent the true classifier and auxiliary variables.

It is well known that the inverse Fisher information $\mathbf{Q}^{-1}$ lower bounds the covariance matrix of the estimated $\mathbf{w}$ [18]. In particular, $[\det(\mathbf{Q})]^{-1}$ lower bounds the product of variances of the elements in $\mathbf{w}$. The goal in selecting $\mathcal{D}_l^p$ is to reduce the variances, or uncertainty, of $\mathbf{w}$. Thus, we seek the $\mathcal{D}_l^p$ that maximize $\det(\mathbf{Q})$.

The selection proceeds in a sequential manner. Initially, $\mathcal{D}_u^p = \mathcal{D}^p$, $\mathcal{D}_l^p$ is empty, and $\mathbf{Q} = \sum_{i=1}^{N^a} \sigma_i^a (1 - \sigma_i^a) \mathbf{x}_i^a \mathbf{x}_i^{aT}$. Then, one at a time, a data point $\mathbf{x}_i^p \in \mathcal{D}_u^p$ is selected and moved from $\mathcal{D}_u^p$ to $\mathcal{D}_l^p$. This causes $\mathbf{Q}$ to be updated as: $\mathbf{Q} \leftarrow \mathbf{Q} + \sigma_i^p (1 - \sigma_i^p) \mathbf{x}_i^p (\mathbf{x}_i^p)^{\mathrm{T}}$. At each iteration, the selection is based on

$$\begin{aligned} &\max_{\mathbf{x}_i^p \in \mathcal{D}_u^p} \det \left\{ \mathbf{Q} + \sigma_i^p (1 - \sigma_i^p) \mathbf{x}_i^p (\mathbf{x}_i^p)^{\mathrm{T}} \right\} \\ &\qquad = \max_{\mathbf{x}_i^p \in \mathcal{D}_u^p} \left\{ 1 + \sigma_i^p (1 - \sigma_i^p) (\mathbf{x}_i^p)^{\mathrm{T}} \mathbf{Q}^{-1} \mathbf{x}_i^p \right\} \quad (21) \end{aligned}$$

where we assume the existence of $\mathbf{Q}^{-1}$, which can often be assured by using sufficient auxiliary data $\mathcal{D}^a$.

Evaluation of (21) requires the true values of $\mathbf{w}$ and $\{\mu_i\}$, which are not known *a priori*. We follow [14] and replace them with the $\mathbf{w}$ and $\{\mu_i\}$ that are estimated from $\mathcal{D}^a \cup \mathcal{D}_l^p$, where $\mathcal{D}_l^p$ are the primary labeled data selected up to the present.

VI. Results

In this section, the performance of MigLogit is demonstrated and compared to the standard logistic regression. The MigLogit is trained using $\mathcal{D}^a \cup \mathcal{D}_l^p$, where $\mathcal{D}_l^p$ are either randomly selected from $\mathcal{D}^p$, or actively selected from $\mathcal{D}^p$ using the method in Section V. When $\mathcal{D}_l^p$ are randomly selected, 50 independent trials are performed, and the results are obtained as an average over the trials. Three logistic regression classifiers are trained using different combinations of $\mathcal{D}^a$ and $\mathcal{D}_l^p : \mathcal{D}^a \cup \mathcal{D}_l^p$, $\mathcal{D}_l^p$ alone, and $\mathcal{D}^a$ alone, where $\mathcal{D}_l^p$ are identical to the $\mathcal{D}_l^p$ used by MigLogit. The four classifiers are tested on $\mathcal{D}_u^p = \mathcal{D}^p \setminus \mathcal{D}_l^p$ to produce the test-error rate or the area under the ROC curve (AUC). Calculation of test error rates is based on the following decision rule: declare $y^p = -1$ if $\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}^p) \leq 0.5$ and $y^p = 1$ otherwise, for any $\mathbf{x}^p \in \mathcal{D}_u^p$.

The performance of MigLogit is demonstrated on two problem domains. The first is a simulated example, and the second is detection of UXO, where the UXO signatures are site sensitive.

Throughout this section, the C in MigLogit is set to $C = 6$ when the comparison is made to logistic regression. In addition, we present a comparison of MigLogit with different Cs, to examine the sensitivity of MigLogit's performance to C.

A. Synthesized Data

In the first example, the primary data are simulated as two bivariate Gaussian distributions representing class "-1" and class "$+1$," respectively. In particular, we have $\Pr(\mathbf{x}^p|y^p = -1) = \mathcal{N}(\mathbf{x}^p; \boldsymbol{\mu}_0, \boldsymbol{\Sigma})$ and $\Pr(\mathbf{x}^p|y^p = 1) = \mathcal{N}(\mathbf{x}^p; \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$, where the Gaussian parameters $\boldsymbol{\mu}_0 = [0, 0]^{\mathrm{T}}$, $\boldsymbol{\mu}_1 = [2.3, 2.3]^{\mathrm{T}}$, and $\boldsymbol{\Sigma} = \begin{bmatrix} 1.75 & -0.433 \\ -0.433 & 1.25 \end{bmatrix}$. The auxiliary data $\mathcal{D}^a$ are then a selected draw from the two Gaussian distributions, as described in [12]. We take the selection probability $\Pr(s|\mathbf{x}^p, y^p = -1) = \sigma(w_0 + w_1 K(\mathbf{x}^p, \boldsymbol{\mu}_0^s; \boldsymbol{\Sigma}))$ and $\Pr(s|\mathbf{x}^p, y^p = +1) = \sigma(w_0 + w_1 K(\mathbf{x}^p, \boldsymbol{\mu}_1^s; \boldsymbol{\Sigma}))$, where σ is the sigmoid function, $w_0 = -1$, $w_1 = \exp(1)$, $K(\mathbf{x}^p, \boldsymbol{\mu}_0^s; \boldsymbol{\Sigma}) = \exp\{-0.5(\mathbf{x}^p - \boldsymbol{\mu}_0^s)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}^p - \boldsymbol{\mu}_0^s)\}$ with $\boldsymbol{\mu}_0^s = [2, 1]^{\mathrm{T}}$, and $K(\mathbf{x}^p, \boldsymbol{\mu}_1^s; \boldsymbol{\Sigma}) = \exp\{-0.5(\mathbf{x}^p - \boldsymbol{\mu}_1^s)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}^p - \boldsymbol{\mu}_1^s)\}$ with $\boldsymbol{\mu}_1^s = [0, 3]^{\mathrm{T}}$. We obtain 150 samples of $\mathcal{D}^p$ and 150 samples of $\mathcal{D}^a$, which are shown in Fig. 3.

The MigLogit and logistic regression classifiers are trained and tested as explained at the beginning of this section. The results are represented as test error rate as a function of number of primary labeled data used in training and are shown in Figs. 1 and 2. Each curve in Fig. 1 is an average over 50 independent trials, with each trial having an independent random selection of $\mathcal{D}_l^p$. Fig. 2 shows the active learning results, with $\mathcal{D}_l^p$ actively selected as described in Section V.

Several observations are made from inspection of Figs. 1 and 2.

1) The MigLogit consistently outperforms the three standard logistic regression classifiers, by a considerable margin. This improvement is attributed to a selective usage of the examples in $\mathcal{D}^a$. In particular, each example in $\mathcal{D}^a$ is employed according to its agreement with q: a good agreement warrants a higher contribution to determination of the classifier while a poor agreement makes the contribution discounted. The selectivity is implemented through the auxiliary variables which are estimated based on a few examples from $\mathcal{D}^p$.
2) The performance of the logistic regression trained on $\mathcal{D}_l^p$ alone significantly changes with the size of $\mathcal{D}_l^p$. This is understandable, considering that $\mathcal{D}_l^p$ are the only examples determining the classifier. The abrupt drop of errors from iteration 10 to iteration 11 in Fig. 2 may be because the label found at iteration 11 is critical to determining $\mathbf{w}$.
3) The logistic regression trained on $\mathcal{D}^a$ alone significantly performs worse than MigLogit, reflecting a marked mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$.
4) The logistic regression trained on $\mathcal{D}^a \cup \mathcal{D}_l^p$ improves, but mildly, as $\mathcal{D}_l^p$ grows, and it is ultimately outperformed by the logistic regression trained on $\mathcal{D}_l^p$ alone, demonstrating that some data in $\mathcal{D}^a$ are mismatched with $\mathcal{D}^p$ and, hence, cannot correctly be classified along with $\mathcal{D}^p$, if the mismatch is not compensated.
5) As $\mathcal{D}_l^p$ grows, the logistic regression trained on $\mathcal{D}_l^p$ alone finally approaches to MigLogit, showing that without the

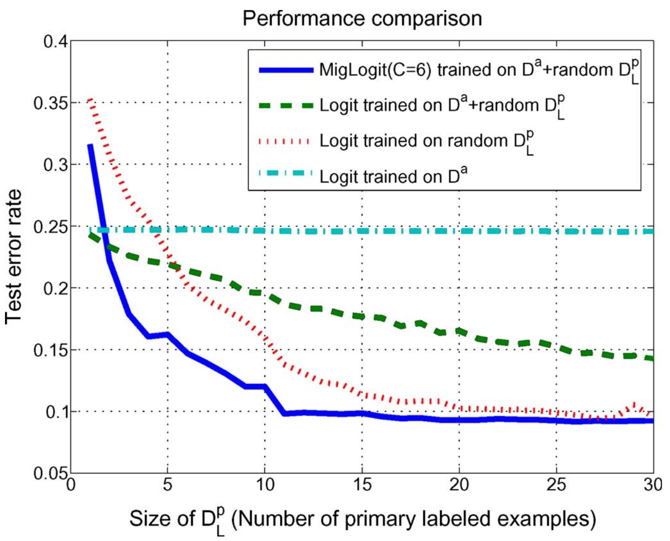


Fig. 1. Test error rates of MigLogit and logistic regression on the synthesized data, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are randomly selected from $\mathcal{D}^p$. The error rates are an average over 50 independent trials of random selection of $\mathcal{D}_l^p$.

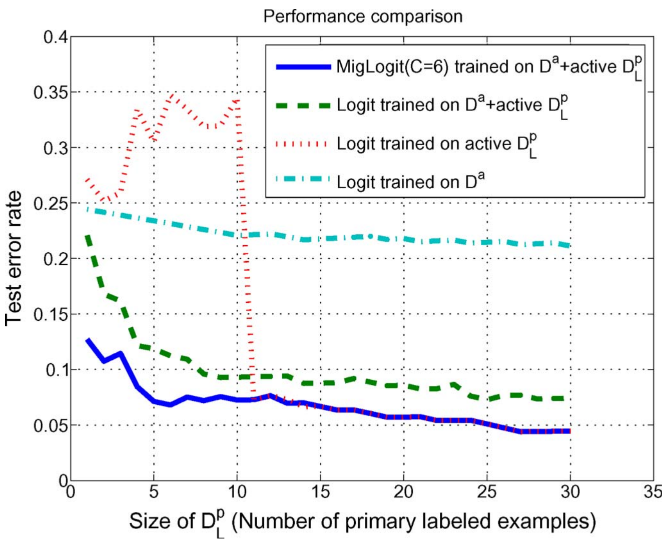


Fig. 2. Error rates of MigLogit and logistic regression on the synthesized data, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$, using the method in Section V.

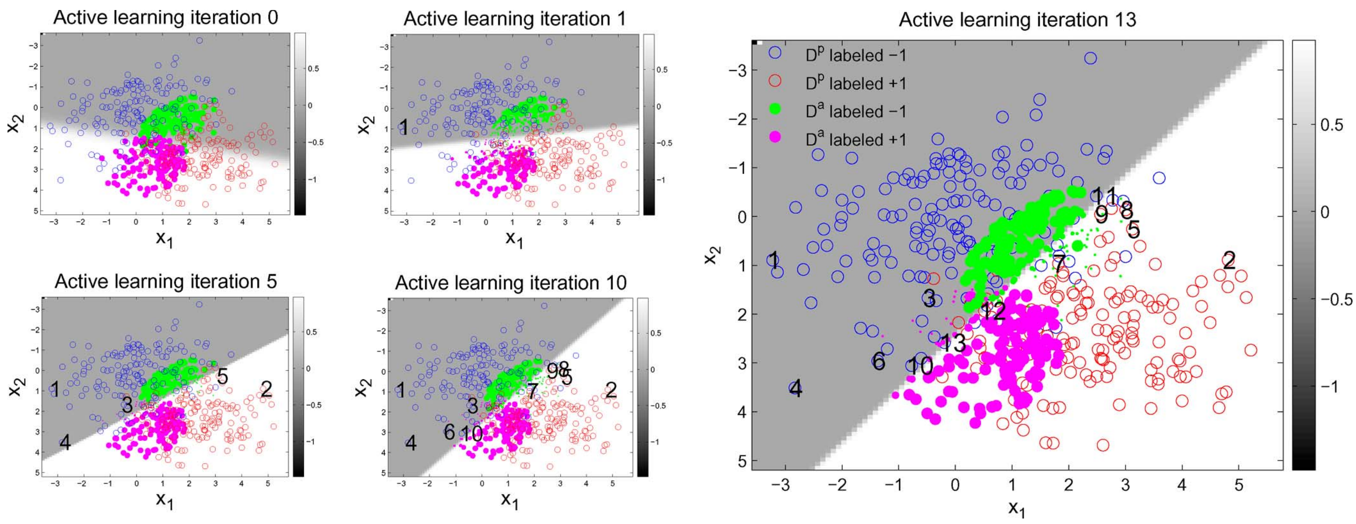


Fig. 3. Illustration of active data selection by MigLogit. Only iterations 0, 1, 5, 10, and 13 are shown. The different symbols are defined as: blue ● = $\mathcal{D}^p$ labeled "−1," red ● = $\mathcal{D}^p$ labeled "+1," green ● = $\mathcal{D}^a$ labeled "−1," and magenta ● = $\mathcal{D}^a$ labeled "+1." The numbers in black denote $\mathcal{D}_l^p$ and represent the order of selection. The smaller ● near the decision boundaries symbolize weakened participation of the associated $\mathcal{D}^a$ in determining $\mathbf{w}$. This may only be visible in the zoomed figure (iteration 13).

interference of $\mathcal{D}^a$, a sufficient $\mathcal{D}_l^p$ can define a correct classifier.

6) All four classifiers benefit from the actively selected $\mathcal{D}_l^p$, and this is consistent with the general observation with active learning [16], [17].

The labeled primary examples $\mathcal{D}_l^p$ play double roles in the learning process. On the one hand, they help to find the correct $\mathbf{w}$, and on the other hand, they serve as representative primary labeled data in finding the degree of agreement of each auxiliary example with primary data (i.e., estimating the auxiliary variables). Suppose that it requires n_1 primary labeled examples to find the auxiliary variables for compensating the mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$, and that it requires n_2 labeled primary examples alone (without using auxiliary examples) to find the correct classifier. One may conjecture that n_1 is smaller than n_2. Although we have not rigorously proven this, the results in Figs. 1 and 2 provide empirical evidence for this being true: Note that MigLogit uses much fewer primary labeled examples to find the correct classifier than Logit trained on $\mathcal{D}_l^p$ does.

The double roles assumed by the primary labeled examples make it a critical issue how to select these examples. We already see that actively selected examples give significant boost to the performance. This makes it clear that active learning is a more appropriate strategy than pure random selection, and contributes in an important manner to the proposed method.

To better understand the active selection process, we show in Fig. 3 the first few iterations of active learning. Iteration 0 corresponds to the initially empty $\mathcal{D}_l^p$, and iterations 1, 5, 10, and 13 respectively correspond to 1, 5, 10, and 13 data points accumulatively selected from $\mathcal{D}_u^p$ into $\mathcal{D}_l^p$. Each time a new data point is selected, the $\mathbf{w}$ is retrained, yielding the different decision boundaries. As shown in Fig. 3, the decision boundary does not change much after ten data are selected, demonstrating convergence.

In Fig. 3, each auxiliary data point $\mathbf{x}_i^a \in \mathcal{D}^a$ is symbolically displayed with a size in proportion to $\exp(-y_i^a \mu_i/12)$, hence a small symbol of auxiliary data corresponds to large $y_i^a \mu_i$ and, hence, indicates a discounted contribution of the ith auxiliary example to determination of $\mathbf{w}$. The auxiliary data that cannot be correctly classified along with the primary data are de-emphasized by the MigLogit. Usually, the auxiliary data near the decision boundary are de-emphasized.

B. Application to Detection of Site-Sensitive UXO

UXO consists of ordnance that did not explode upon impact with the ground. The UXO items are typically buried and consist of significant quantities of metal. Sensing of UXO is typically performed using EMI and magnetometer sensors. The principal challenge involves distinguishing actual UXO from buried nonordnance conducting materials. For a more detailed general description of UXO sensing, see [1].

The sensor signature of a given UXO item is dependent on the soil properties as well as the history of the site in which it is located, the latter having a particular strong influence on the signature. The site history is dictated by complex factors such as colocated ordnance, the way the ordnance impacted the soil, and the surrounding man-made conducting clutter and UXO fragments. Therefore, UXO detection is a typical site-sensitive problem.

The site-sensitivity makes standard supervised classification techniques an inappropriate choice for UXO detection, due to the difficulty in constituting a universal training set for classifier design. The training examples collected at previous sites are often not appropriate for use for analysis of the current site since the current site is often different from the previous ones (in the sense described above). Despite these disparities, the examples from previous sites are not totally useless; indeed, they can provide quite useful information about the examples for the current site (particularly for the UXO, since the ordnance types at different sites are often the same or similar; the clutter signatures are most often site specific). The usefulness of existing labeled data for a new site of interest is dictated by the characteristics of the new site, as well as on the characteristics of the sites from which the labeled data were acquired; these interrelationships are complex and often difficult to characterize *a priori* (often accurate records are not available about the history of a former bombing site).

Let the examples at the current UXO cite be distributed according to $\mathcal{T}(\mathbf{x}, y)$, and the examples at a previous UXO cite be distributed according to $\mathcal{A}(\mathbf{x}, y)$. It is seen that the empirical loss for detection of UXO at the current cite is well described by (3). Therefore, one can employ the technique of MigLogit to design the desired classifier.

To demonstrate the utility of MigLogit in UXO detection, we here consider two UXO sites and design the classifier for the primary site (the one we are interested in) by using examples from another site (the auxiliary site). The auxiliary site is called *Jefferson Proving Ground (JPG)*, for which one is provided with the EMI and magnetometer measurements as well the associated labels (which are binary: UXO or non-UXO). The examples from the auxiliary site constitutes the auxiliary data $\mathcal{D}^a$. The primary site we are interested in is called *Badlands*, for which we have unlabeled EMI and magnetometer measurement for constituting the primary data $\mathcal{D}^p$. The labeled JPG data consists of 104 total items, of which 16 are UXO, and 88 are non-UXO. The Badlands site consists of a total of 492 items, 57 of which are UXO, and the remaining 435 are non-UXO. These two former bombing ranges exist at two very different geographical locations within the United States.

The UXO sensor measurements are mapped to 4-D feature vectors $[\log(M_p), \log(M_z), z, \log(M_p/M_z)]$, where M_p and M_z are the dipole moments perpendicular and parallel to the target axis, respectively, and z is the approximate target depth [1]. These parameters are estimated by fitting the EMI and magnetometer measurements to a physical model [1]; the features from this paper are available upon request to the authors. Each feature is normalized to have zero mean and unitary variance. In UXO detection, one is interested in the receiver's operating characteristic (ROC) curve, particularly the AUC [19].

The results are presented in Figs. 4(a) and 5(a), where each curve is the AUC as a function of the size of $\mathcal{D}_l^p$. The results in Fig. 4(a) are obtained by randomly labeling primary data and by averaging the AUCs over 50 independent trials. The results in Fig. 5(a) are obtained by actively labeling primary data using the method in Section V. For a better view of the improvement achieved by MigLogit, we plot in Figs. 4(b) and 5(b) the AUC of MigLogit with the AUC of each logistic regression classifier subtracted. A positive difference indicate performance improvement while a negative difference indicates performance degradation. We have the following observations.

1) With $\mathcal{D}_l^p$ randomly determined, MigLogit outperforms all logistic regression classifiers except at the early part of the curves, where there are very few examples in $\mathcal{D}_l^p$. As discussed at the end of Section VI-A, the primary labeled examples are critical to the performance of MigLogit. With a few randomly selected examples one may not be able to find the appropriate auxiliary variables, leading to a poor compensation of the mismatch between $\mathcal{D}^p$ and $\mathcal{D}^a$ and, therefore, performance degradation.
2) With $\mathcal{D}_l^p$ actively determined, MigLogit outperforms all logistic regression classifiers, regardless of the number of primary labeled examples. This verifies that a good choice of $\mathcal{D}_l^p$ is important to the performance of MigLogit.
3) Active learning is not only beneficial to MigLogit, but to other classifiers as well, again demonstrating the advantage of active learning.
4) All conclusions observed in the results on synthesize data extend to the UXO results here.

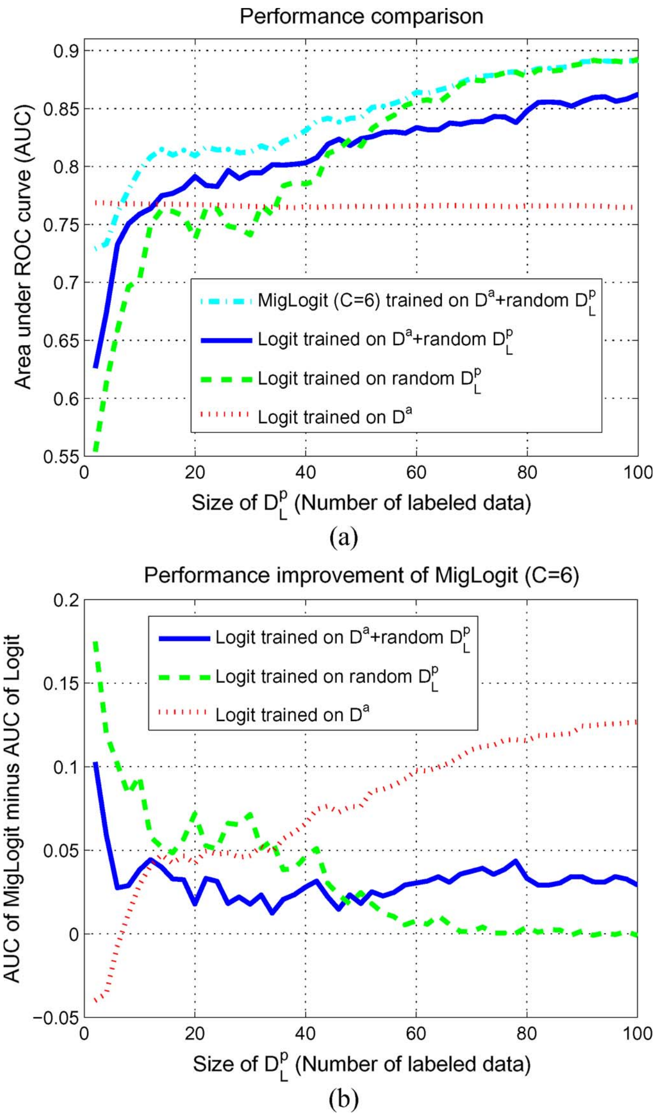
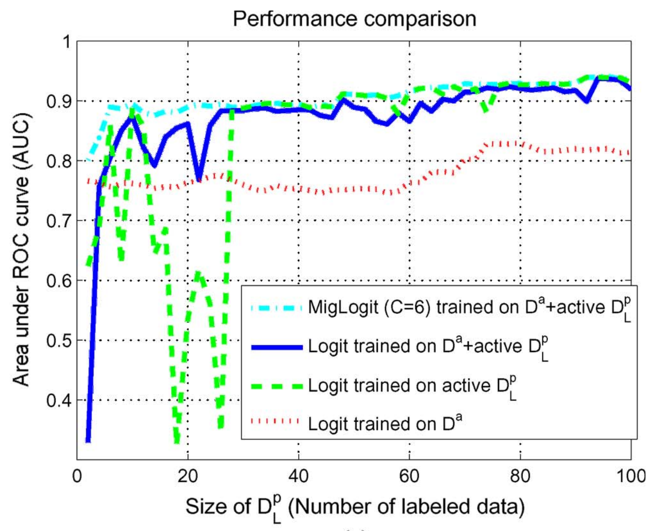
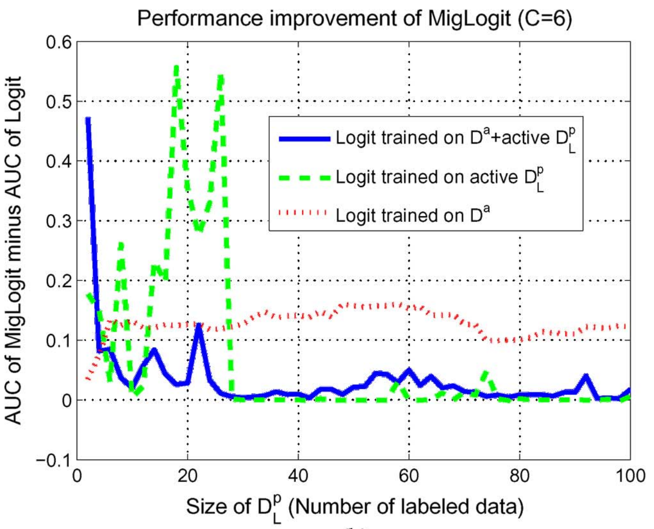


Fig. 4. (a) AUC of MigLogit and logistic regression on the UXO data, as a function of size of $\mathcal{D}_l^p$. (b) AUC of MigLogit minus the AUCs of logistic regression. The auxiliary data are collected at *JPG* and the primary data are collected at *Badlands*. The primary labeled data $\mathcal{D}_l^p$ are randomly selected from $\mathcal{D}^p$. Each curve is an average over 50 independent trials of random selection of $\mathcal{D}_l^p$.

These observations suggest that MigLogit successfully leverages the auxiliary data from previous UXO sites to quickly find the correct classifier for the new site, requiring much fewer labeled data from the new site than standard classifiers. The



(a)



(b)

Fig. 5. (a) AUC of MigLogit and logistic regression on the UXO data, as a function of size of $\mathcal{D}_l^p$. (b) AUC of MigLogit minus the AUCs of logistic regression. The auxiliary data are collected at *JPG* and the primary data are collected at *Badlands*. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$, based on the method in Section V.

results for the actual (measured) UXO data suggest that the algorithm captures the concept drift associated with realistic problems of practical importance.

C. Robustness of MigLogit

We have discussed in Section 4 how to choose C in MigLogit. In this section, we show that when the choice is not accurate, MigLogit still yields robust results.

We consider the same UXO data and use the same experimental settings as in the previous section, except that we vary C in MigLogit to examine its robustness. The $\mathcal{D}_l^p$ are determined by active learning as described in Section V. We consider eight different values of C, $C = 2, 4, 6, 8, 10, 12, 14, 16$, to examine the differences in the results obtained under these settings. The results are shown in Fig. 6. It is seen that over this wide range of choices for C, MigLogit consistently yields superior performances except in a few cases, which occur when the size of $\mathcal{D}_l^p$ is very small, and C is either too large or too small. These results demonstrate the robustness of MigLogit to the choice of C, particularly when active learning is invoked. With different C, the $\mathcal{D}_l^p$ are also differently selected, which counteracts the effect of C and increase the robustness of MigLogit.

VII. Exploiting MigLogit to Handle Multiple Data Sets

In the formulation of MigLogit, we have assumed that we have a single auxiliary set of labeled data $\mathcal{D}^a$, for which the goal is to learn relationships with a primary set of (unlabeled or partially labeled) data $\mathcal{D}^p$. In some problems, we may have $M-1$ existing labeled data sets, indexed by $m = 1, 2, \ldots, M-1$, and we are interested in learning the characteristics (concept) of a new (the Mth) unlabeled or partially labeled data set. In this scenario, the first $M-1$ data sets constitute a sequence of auxiliary sets, which we assume to be denoted as $\mathcal{D}_1^a$, $\mathcal{D}_2^a$, ..., $\mathcal{D}_{M-1}^a$, and the new (the Mth) data set constitutes our primary set $\mathcal{D}^p$, of which a labeled subset $\mathcal{D}_l^p$ is defined via random or active selection. The MigLogit algorithm developed in previous sections can be employed to handle the situation of multiple data sets, with no modification or small modifications to the algorithm, as discussed in more detail below.

A. Aggregating the Multiple Auxiliary Data Sets

A simple method of applying the MigLogit algorithm to multiple data sets is to construct one big auxiliary set $\mathcal{D}^a$ by aggregating the data from all original auxiliary sets, i.e., letting $\mathcal{D}^a = \mathcal{D}_1^a \cup \mathcal{D}_1^a \cup \cdots \cup \mathcal{D}_{M-1}^a$, and then apply MigLogit in the way as described in previous sections. Since the aggregation of multiple auxiliary sets does not change the fact that a unique auxiliary variable μ is associated with each auxiliary data point, the ability of MigLogit to control the participation of *individual* auxiliary data points is still retained when learning the primary classifier.

The only consequence arising from the aggregation is that all auxiliary sets are assumed to have the same average mismatch C with the primary set and, therefore, the same average participation in determining the primary classifier. This is equivalent to saying that, *a priori*, all auxiliary sets contribute equally to the primary learning task. Nonetheless, the prior assumption is not decisive, since the ultimate contribution of each auxiliary set will also depend on how well it matches the primary set, with this indicated by the μ variables that are automatically learned along with primary classifier.

B. Retaining Separation of the Multiple Auxiliary Data Sets

A more advanced method of applying MigLogit to multiple data sets is to keep the $M-1$ auxiliary sets separate as they originally are and let each auxiliary set have its own average mismatch with the primary set, specifically, let $\mathcal{D}_m^a$ have an average mismatch C_m with $\mathcal{D}^p$. Accordingly, the linear

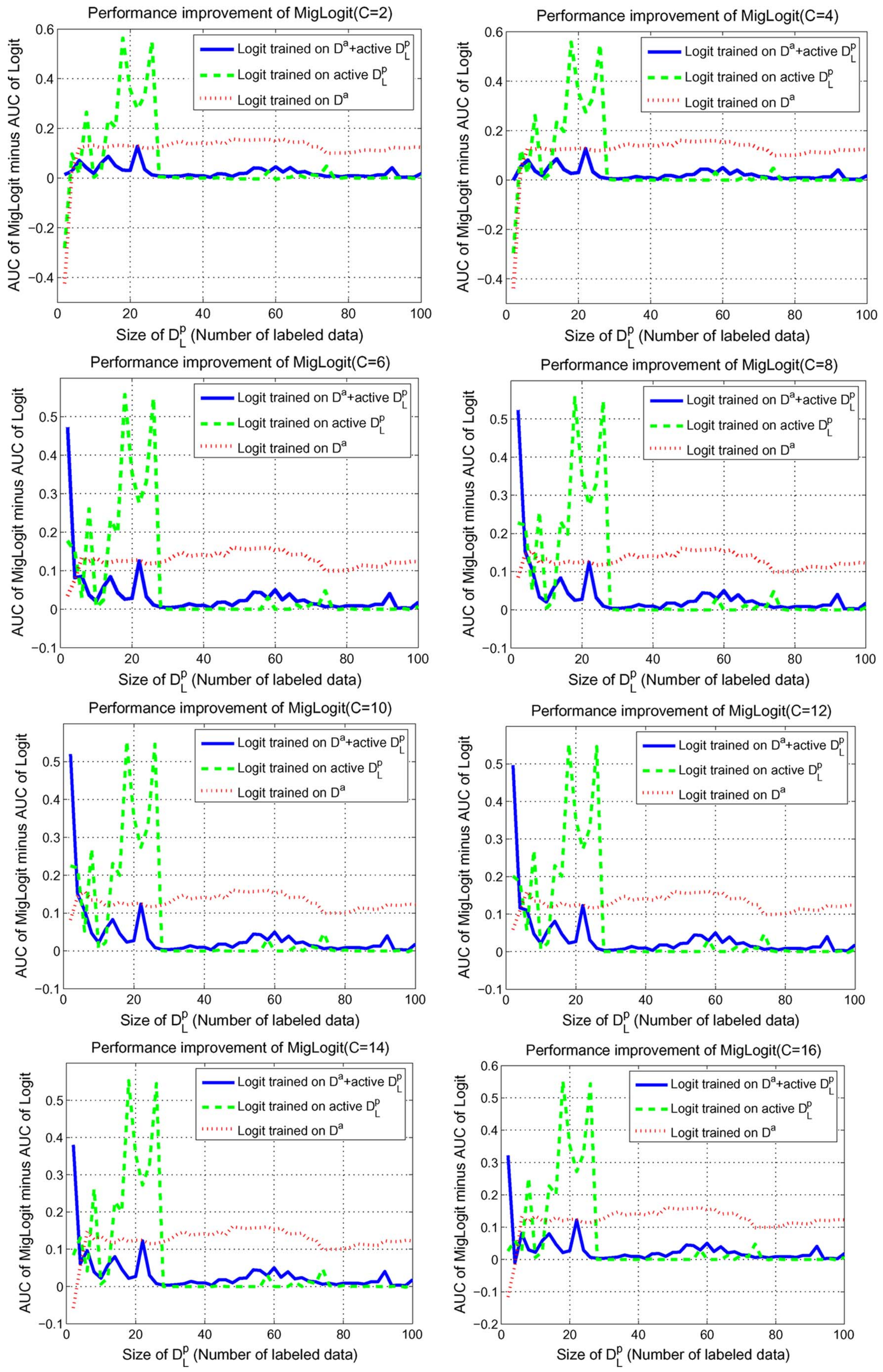


Fig. 6. Performance of MigLogit with different choices of C, in the UXO detection problem. The vertical axis is the AUC of MigLogit minus the AUCs of logistic regression. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$. From top-left to right-bottom, $C = 2, 4, 6, 8, 10, 12, 14, 16$.

constraint in (11) is replaced with $M-1$ linear constraints, and the associated optimization problem is modified as

$$\max_{\mathbf{w},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_{M-1}} \ell\left(\mathbf{w},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_{M-1};\mathcal{D}_l^p\cup\mathcal{D}_1^a\cup\cdots\cup\mathcal{D}_{M-1}^a\right) \tag{22}$$

s.t.

$$\frac{1}{N_m^a}\sum_{i=1}^{N_m^a} y_{m,i}^a\mu_{m,i}\le C_m,\qquad C_m\ge 0,\qquad m=1,\ldots,M-1 \tag{23}$$

$$y_{m,i}^a\mu_{m,i}\ge 0,\qquad i=1,\ldots,N_m^a,\quad m=1,\ldots,M-1 \tag{24}$$

with the logarithmic likelihood function

$$\begin{aligned}\ell&\left(\mathbf{w},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_{M-1};\mathcal{D}_l^p\cup\mathcal{D}_1^a\cup\cdots\cup\mathcal{D}_{M-1}^a\right)\\ &=\sum_{i=1}^{N_l^p}\ln\sigma\left(y_i^p\mathbf{w}^{\mathrm{T}}\mathbf{x}_i^p\right)\\ &\quad+\sum_{m=1}^{M-1}\sum_{i=1}^{N_m^a}\ln\sigma\left(y_{m,i}^a\mathbf{w}^{\mathrm{T}}\mathbf{x}_{m,i}^a+y_{m,i}^a\mu_{m,i}\right)\end{aligned} \tag{25}$$

where $\boldsymbol{\mu}_m=[\mu_{m,1},\mu_{m,2},\ldots,\mu_m,N_m^a]^{\mathrm{T}}$ is a column vector of the auxiliary variables for $\mathcal{D}_m^a$ and the subscript m is used to index the auxiliary sets. The primary labeled set $\mathcal{D}_l^p$ along with all $M-1$ auxiliary sets define a suite of M labeled data sets. Which of the M data sets is used as the primary set in fact does not affect the algorithm and is a matter totally indicated by the application being considered. In general, the algorithm can be applied to any M labeled data sets, with one of them assumed primary for which a classifier is desired, and the remaining assumed auxiliary sets providing potentially relevant information to the primary task. It is noted that the relevance is quantified at the level of individual data points by the μ variables, thus which particular data points are relevant and which are not will be automatically discovered by the algorithm.

The parameter C_m provides the prior information about the average mismatch between $\mathcal{D}_m^a$ and $\mathcal{D}_l^p$, with the ultimately learned mismatch jointly dictated by C_m and the data. An independent C for each auxiliary set implies that the multiple auxiliary sets may *a priori* be assumed to contribute unequally to the primary learning task, a property in contrast to that of the aggregated auxiliary sets. The unequal prior is particularly useful when the auxiliary sets are ordered temporally or spatially, in which case the difference in age or spatial distance may dictate that different auxiliary sets have different prior mismatches to the primary set.

Solving the optimization problem in (22)–(24) is as easy as solving that in (10)–(12). Given $\mathbf{w}$, the auxiliary variables for each auxiliary data set are solved independently, using the analytic solutions in Section III.

VIII. Relations to Existing Work

The MigLogit is a technique to enhance training examples used to learn a given primary concept through a controlled exploitation of the training examples from an auxiliary concept or a sequence of auxiliary concepts, assuming that the auxiliary concepts may potentially be related to the primary concept in some (unknown) manner. The controlled exploitation is accomplished by a set of auxiliary variables, with one unique such variable associated with each individual example of the auxiliary concepts. The variables are automatically learned from the data, requiring no user-provided information other than the average drift (mismatch) between the primary concept and the auxiliary concepts.

The basic idea behind MigLogit, i.e., borrowing information from related concepts to the benefit of learning the primary concept, has been used in other methodologies, particularly multitask learning (MTL) [20]–[22], and example selection/weighting [23]–[26]. However, the MigLogit accomplishes the idea of information borrowing in a unique manner, which distinguishes it from existing methods.

MTL methods have implemented information borrowing by a common internal representation in neural networks [20], [27], [28], a kernel matrix in regularization methods [29], a common prior distribution in Bayesian hierarchical models [30], and a common Dirichlet process (DP) [31] in nonparametric Bayesian models [32]. All these methods implement information borrowing at the level of tasks or concepts, in contrast to the MigLogit, which implements information borrowing at the level of individual data points.

The methods of example selection/weighting usually apply windows to a data stream to break it into batches, and then select the window size for each batch based on generalization performances on the most recent batch or weight the data batches according to their ages [23]–[25]. In these methods, the information borrowing is implemented at the level of data batches, instead of data points as in MigLogit. In addition, the weights are usually specified as a function of data age, instead of being automatically learned from data. The method in [26] learns a classifier based on a weighting of two data sets, with the auxiliary set given lower weight to reflect that it has a discounted contribution to the classifier learning. It is clear that here information borrowing is implemented at the level of data sets, not data points.

The way in which MigLogit accomplishes information borrowing makes it an advantageous method in may ways. First, by borrowing information at the level of individual data points, MigLogit can discover the mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$ in a more robust way, particularly when some data in $\mathcal{D}^a$ match well to $\mathcal{D}^p$ while others do not. Second, the auxiliary variables in MigLogit, which characterize the amount of information borrowed from individual data points, are automatically learned based on data, thus the information required from the user is reduced. In contrast to many other methods, which require the user to specify a form of internal representation, or a kernel matrix, or a prior distribution, etc., MigLogit requires no input from the user other than the data and the average mismatch C. We believe that, in terms of the amount of information required from the user, MigLogit is no more demanding than the methods based on DPs, since C plays a similar role as the precision parameter in a DP, and a DP has a base distribution that need be specified by the user.

IX. Conclusion

We have proposed an algorithm, called MigLogit, for learning in the presence of concept change between the (auxiliary) training data $\mathcal{D}^a$ and the (primary) testing data $\mathcal{D}^p$. The basic idea of our method is to introduce an auxiliary variable μ_i for each example $(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a$, which allows $\mathbf{x}_i^a$ to migrate to the class y_i^a when it cannot correctly be classified along with $\mathbf{x}^p$ by the classifier. The migrations of $\mathcal{D}^a$ are controlled by the inequality constraint $(1/N^a)\sum_{i=1}^{N^a} y_i^a \mu_i \leq C$, where $C \geq 0$ is an appropriate bound limiting the average migration. The primary labeled data $\mathcal{D}_l^p$ play a pivotal role in correctly learning the classifier, and we have presented a method to actively selecting $\mathcal{D}_l^p$, which enhances the adaptivity of the entire learning process. We have developed a fast learning algorithm to enhance the ability of MigLogit to handle large auxiliary data sets.

The results from both synthesized data and data collected at actual UXO sites show that MigLogit yields significant improvements over the standard logistic regression, demonstrating that if the classifier trained on $\mathcal{D}^a$ is to generalize well to $\mathcal{D}^p$, the mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$ must be compensated.

Although the core MigLogit algorithm has been developed in the setting of two data sets, we have also presented the methods for applying the MigLogit algorithm to multiple data sets, with no modification or some small modifications to the core algorithm. MigLogit utilizes the notion of information borrowing as do MTL and example selection/weighting. Yet, the way in which information borrowing is accomplished in MogLogit is unique and offers many advantages over other methods. These advantages, along with the convexity of the associated optimization problem and the computational efficiency, makes MigLogit particularly suitable for practical applications such as UXO detection.

There are several interesting directions for future research. The first one involves theoretical analysis of MigLogit, particularly examination of its performance bounds. The second direction is to extend the work to other models like support vector machines (SVMs). One possible extension to SVMs is to use the μ variables as a set of data-dependent slack variables. Since the SVM is a convex problem, one may still get analytic solutions for the μ variables. Finally, it is also interesting to investigate the practical meaning that the μ variables may possess in certain applications (this may be possible because the μ variables have strong geometric interpretations). This paper can be important to knowledge discovery, for example, filtering out outdated cases from the case base in medicine.

Appendix
Proof of Theorem 1

Let $f'(z)$ be the first derivative of $f(z)$. We have $\sum_{i=1}^N f(b_i + z_i) = \sum_{i=1}^N f(b_i) + \sum_{i=1}^N \int_0^{z_i} f'(x+b_i)\,dx$. The first term on the right side is a constant and hence, the problem in (14) is equivalent to

$$\max_{\{z_i\}} \sum_{i=1}^{N} \int_0^{z_i} f'(b_i + x)\,dx. \qquad \text{(A-1)}$$

Because $f''(z) < 0$ and $f'(z) >= 0$, we have for any $\tau_1 \leq \tau_2$ that $f'(\tau_1 + x) \geq f'(\tau_2 + x) >= 0$ and, consequently

$$\int_0^{\Delta} f'(\tau_1 + x)\,dx \geq \int_0^{\Delta} f'(\tau_2 + x)\,dx >= 0$$
$$\forall\, \tau_1 \leq \tau_2 \text{ and } \Delta \geq 0. \qquad \text{(A-2)}$$

By (13), there exists $0 \leq r < n(b_{n+1} - b_n)$ such that $R = nb_n - \sum_{k=1}^n b_k + r = \sum_{k=1}^n k\Delta_k$ where $\Delta_k = b_{k+1} - b_k$ for $k = 1, \ldots, n-1$, and $\Delta_n = r/n$. We now use (A-2) to distribute $\Delta_1, 2\Delta_2, \ldots, n\Delta_n$ to $z_1, z_2, \ldots, z_N$ such that the resulting $\{z_i\}$ maximize (A-1). As $\Delta_k \geq 0$ for $k = 1, \ldots, n$, and any distribution of $\{k\Delta_k\}_{k=1}^N$ to $\{z_k\}_{k=1}^N$ makes $\sum_{i=1}^N z_i = \sum_{k=1}^n k\Delta_k = R$, the constraints of (15) and (16) are automatically satisfied.

Initially, $z_i = 0$ for $i = 1, 2, \ldots, N$.

As $\Delta_1 = b_2 - b_1 \geq 0$, by (A-2), $\int_0^{\Delta_1} f'(b_1 + x)dx \geq \int_0^{\Delta_1} f'(b_2 + x)dx$, therefore Δ_1 is distributed to z_1, i.e., $z_1 \leftarrow z_1 + \Delta_1$, which makes $b_1 + z_1 = b_2$.

Similarly, $\Delta_2 = b_3 - b_2 \geq 0$, by (A-2), $\int_0^{\Delta_2} f'(b_2 + x)dx \geq \int_0^{\Delta_2} f'(b_3 + x)dx$, therefore $2\Delta_2$ is equally distributed to z_1 and z_2, i.e., $z_1 \leftarrow z_1 + \Delta_2$ and $z_2 \leftarrow z_2 + \Delta_2$, which makes $b_1 + z_1 = b_2 + z_2 = b_3$.

Generally, $k\Delta_k$ is equally distributed to $z_1, z_2, \ldots, z_k$. After the distribution of $k\Delta_k$, $k = 1, 2, \ldots, n$, we have $z_k = \sum_{i=k}^n \Delta_i$ for $k = 1, 2, \ldots, n$ and $z_k = 0$ for $k = n+1, n+2, \ldots, N$, which is equal to the solution in (17). Because the problem is strictly concave, the solution is unique and globally optimal. ■

Derivation of Equation (20): By definition of logistic regression, $\mathbf{w}$ is the parameter of the conditional distribution $\Pr(y|\mathbf{x}) = \sigma(y\mathbf{w}^{\mathrm{T}}\mathbf{x})$, with $\mathbf{x}$ given and fixed. Let $\mathbf{g} = \partial \ln \sigma(y\mathbf{w}^{\mathrm{T}}\mathbf{x})/\partial\mathbf{w} = [1 - \sigma(y\mathbf{w}^{\mathrm{T}}\mathbf{x})]y\mathbf{x}$. Then, $\mathbb{E}_y(\mathbf{g}\mathbf{g}^{\mathrm{T}}) = \sum_{y=-1,1} \sigma(y\mathbf{w}^{\mathrm{T}}\mathbf{x})[1 - \sigma(y\mathbf{w}^{\mathrm{T}}\mathbf{x})]^2\mathbf{x}\mathbf{x}^{\mathrm{T}}$. Using $\sigma(-\mathbf{w}^{\mathrm{T}}\mathbf{x}) = 1 - \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$, we obtain $\mathbb{E}_y(\mathbf{g}\mathbf{g}^{\mathrm{T}}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})[1 - \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})]\mathbf{x}\mathbf{x}^{\mathrm{T}}$. Summing $\mathbb{E}(\mathbf{g}\mathbf{g}^{\mathrm{T}})$ over all primary and auxiliary data points (assuming the data are independent), we obtain the formula of $\mathbf{Q}$. ■

Acknowledgment

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

References

[1] Y. Zhang, L. Collins, H. Yu, C. E. Baum, and L. Carin, "Sensing of unexploded ordnance with magnetometer and induction data: Theory and signal processing," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 1005–1015, May 2003.

[2] G. Widmer and M. Kubat, "Effective learning in dynamic environments by explicit concept tracking," in *Proc. Eur. Conf. Mach. Learning*, P. B. Brazdil, Ed., 1993, pp. 224–227.

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

[4] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[5] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[6] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, Apr. 1996.

[7] J. B. Tenenbaum, "Bayesian modeling of human concept learning," in *Advances in Neural Information Processing Systems*, vol. 11, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999.

[8] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in *Proc. 17th ICML*, P. Langley, Ed., 2000, pp. 487–494.

[9] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining KDD*, 2003, pp. 226–235.

[10] R. Klinkenberg and S. Ruping, "Concept drift and the importance of examples," in *Text Mining—Theoretical Aspects and Applications*, J. Franke, G. Nakhaeizadeh, and I. Renz, Eds. Heidelberg, Germany: Physica-Verlag, 2003, pp. 55–77.

[11] J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, no. 1, pp. 153–161, Jan. 1979.

[12] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. 21st ICML*, 2004, pp. 903–910.

[13] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[14] V. V. Fedorov, *Theory of Optimal Experiments*. New York: Academic, 1972.

[15] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, no. 4, pp. 590–604, Jul. 1992.

[16] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," in *Advances in Neural Information Processing Systems*, vol. 7. Cambridge, MA: MIT Press, 1995, pp. 705–712.

[17] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in Neural Information Processing Systems*, vol. 7. Cambridge, MA: MIT Press, 1995, pp. 231–238.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[19] J. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.

[20] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[21] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, pp. 149–198, 2000.

[22] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. 22nd ICML*, 2005, pp. 1012–1019.

[23] W. Fan, "Systematic data selection to mine concept-drifting data streams," in *Proc. KDD*, 2004, pp. 128–137.

[24] R. Klinkenberg, "Learning drifting concepts: Example selection vs. example weighting," *Intell. Data Anal. (IDA)*, vol. 8, no. 3, pp. 281–300, Aug. 2004.

[25] M. Scholz and R. Klinkenberg, "Boosting classifiers for drifting concepts," *Intell. Data Anal. (IDA)*, vol. 11, no. 1, pp. 3–28, Jan. 2007.

[26] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proc. 21st ICML*, 2004, p. 110.

[27] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, no. 1, pp. 83–99, Dec. 2003.

[28] X. Liao and L. Carin, "Radial basis function network for multi-task learning," in *Advances in Neural Information Processing Systems*, vol. 18. Cambridge, MA: MIT Press, 2006, pp. 795–802.

[29] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 615–637, Apr. 2005.

[30] F. Dominici, G. Parmigiani, K. H. Reckhow, and R. L. Wolpert, "Combining information from related regressions," *J. Agric., Biol., Environ. Stat.*, vol. 2, no. 3, pp. 313–332, Sep. 1997.

[31] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Stat.*, vol. 1, no. 2, pp. 209–230, Mar. 1973.

[32] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res. (JMLR)*, vol. 8, pp. 35–63, May 2007.

Xuejun Liao (SM'04) was born in Qinghai, China. He received the B.S. and M.S. degrees in electrical engineering from Hunan University, Hunan, China, in 1990 and 1993, respectively, and the Ph.D. degree in electrical engineering from Xidian University, Xi'an, China, in 1999.

From 1993 to 1995, he was with the Department of Electrical Engineering, Hunan University, where he worked on electronic instruments. From 1995 to 2000, he was with the National Key Laboratory for Radar Signal Processing, Xidian University, where he worked on automatic target recognition and radar imaging. Since May 2000, he has been with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, first as a Research Associate and, since January 2008, as an Assistant Research Professor. His current research interests are in machine learning, signal processing, and applications to remote sensing, data mining, and bioinformatics.

Lawrence Carin (SM'96–F'01) was born in Washington, DC, on March 25, 1963. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1985, 1986, and 1989, respectively.

In 1989, he was with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY, as an Assistant Professor, and became an Associate Professor in 1994. Since September 1995, he has been with the Department of Electrical Engineering, Duke University, Durham, NC, where he is currently the William H. Younger Professor of Engineering. He was the Principal Investigator (PI) on a Multidisciplinary University Research Initiative (MURI) on demining (1996–2001), and he is currently the PI of a MURI dedicated to multimodal inversion. He was the Cofounder of Signal Innovations Group, Inc. (SIG), which is currently a subsidiary of Integrian, Inc. He currently serves as the Director of Technology at SIG. His current research interests include signal processing, sensing, and machine learning.

Dr. Carin is a member of the Tau Beta Pi and Eta Kappa Nu honor societies. He was an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION from 1995 to 2004.