Logistic Regression with an Auxiliary Data Source

Xuejun Liao XJLIAO@EE.DUKE.EDU
Ya Xue YX10@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708

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 mismatched distributions, 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 μ's 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-Logit" or M-Logit, is demonstrated successfully on simulated as well as real data sets.

1. Introduction

In supervised learning problems, the goal is to design a classifier using the training examples (labeled data) $\mathcal{D}^{tr} = \{(\mathbf{x}_i^{tr}, y_i^{tr})\}_{i=1}^{N^{tr}}$ such that the classifier predicts the label y_i^p correctly for unlabeled primary test data $\mathcal{D}^p = \{(\mathbf{x}_i^p, y_i^p) : y_i^p \text{ missing}\}_{i=1}^{N^p}$. The accuracy of the predictions is significantly affected by the quality of $\mathcal{D}^{tr}$, which is assumed to contain essential information about $\mathcal{D}^p$. A common assumption utilized by learning algorithms is that $\mathcal{D}^{tr}$ are a *sufficient* sample of the *same* source distribution from which $\mathcal{D}^p$ are drawn. Under this assumption, a classifier designed based on $\mathcal{D}^{tr}$ will generalize well when it is tested on $\mathcal{D}^p$. This assumption, however, is often violated in practice. First, in many applications labeling an observation is an expensive process, resulting in *insufficient* labeled data in $\mathcal{D}^{tr}$ that are not able to characterize the statistics of the primary data. Second, $\mathcal{D}^{tr}$ and $\mathcal{D}^p$ are typically collected under different experimental conditions and therefore often exhibit *differences* in their statistics.

Methods to overcome the insufficiency of labeled data have been investigated in the past few years under the names "active learning" [Cohn et al., 1995, Krogh & Vedelsby, 1995] and "semi-supervised learning" [Nigam & et al., 2000], which we do not discuss here, though we will revisit active learning in Section 5.

The problem of data mismatch has been studied in econometrics, where the available $\mathcal{D}^{tr}$ are often a non-randomly selected sample of the true distribution of interest. Heckman (1979) 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 [Zadrozny, 2004], where it is assumed that the primary test data $\mathcal{D}^p \sim \Pr(\mathbf{x}, y)$ while the training examples $\mathcal{D}^{tr} = \mathcal{D}^a \sim \Pr(\mathbf{x}, y|s = 1)$, where the variable s controls the selection of $\mathcal{D}^a$: if $s = 1$, $(\mathbf{x}, y)$ is selected into $\mathcal{D}^a$; if $s = 0$, $(\mathbf{x}, y)$ is not selected into $\mathcal{D}^a$. Evidently, unless s is independent of $(\mathbf{x}, y)$, $\Pr(\mathbf{x}, y|s = 1) \neq \Pr(\mathbf{x}, y)$ and hence $\mathcal{D}^a$ are mismatched with $\mathcal{D}^p$. By Bayes rule,

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

which implies that if one has access to $\frac{\Pr(s=1)}{\Pr(s=1|\mathbf{x},y)}$ one can correct the mismatch by weighting and resampling [Zadrozny et al., 2003, Zadrozny, 2004]. 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 [Zadrozny, 2004].

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

In the general case, however, it is difficult to estimate $\frac{\Pr(s=1)}{\Pr(s=1|\mathbf{x},y)}$, as we do not have a sufficient sample of $\Pr(\mathbf{x}, y, s)$ (if we do, we already have a sufficent sample of $\Pr(\mathbf{x}, y)$, which contradicts the assumption of the problem).

In this paper we consider the case in which we have a fully labeled auxiliary data set $\mathcal{D}^a$ and a partially labeled primary data set $\mathcal{D}^p = \mathcal{D}^p_l \cup \mathcal{D}^p_u$, where $\mathcal{D}^p_l$ are labeled and $\mathcal{D}^p_u$ unlabeled. We assume $\mathcal{D}^p$ and $\mathcal{D}^a$ are drawn from two distributions that are mismatched. Our objective is to use a mixed training set $\mathcal{D}^{tr} = \mathcal{D}^p_l \cup \mathcal{D}^a$ to train a classifier that predicts the labels of $\mathcal{D}^p_u$ accurately. Assume $\mathcal{D}^p \sim \Pr(\mathbf{x}, y)$. In light of equation (1), 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 domain of nonzero probability[1]. As explained in the previous paragraph, it is difficult to correct the mismatch by directly estimating $\frac{\Pr(s=1)}{\Pr(s=1|\mathbf{x},y)}$. Therefore we take an alternative approach. We introduce an auxiliary variable μ_i for each $(\mathbf{x}^a_i, y^a_i) \in \mathcal{D}^a$ to reflect its mismatch with $\mathcal{D}^p$ and to control its participation in the learning process. The μ's play a similar role as the weighting factors $\frac{\Pr(s=1)}{\Pr(s=1|\mathbf{x},y)}$ in (1). 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.

A related problem has been studied in [Wu & Dietterich, 2004], where the classifier is trained on two fixed and labeled data sets $\mathcal{D}^p$ and $\mathcal{D}^a$, where $\mathcal{D}^a$ is of lower quality and provides weaker evidence for the classifier design. The problem is approached by minimizing a weighted sum of two separate loss functions, with one defined for the primary data and the other for the auxiliary data. Our method is distinct from that in [Wu & Dietterich, 2004] in two respects. First, we introduce an auxiliary variable μ_i for each $(\mathbf{x}^a_i, y^a_i) \in \mathcal{D}^a$ and the auxiliary variables are estimated along with the classifier. A large μ_i implies large mismatch of $(\mathbf{x}^a_i, y^a_i)$ with $\mathcal{D}^p$ and accordingly less participation of $\mathbf{x}^a_i$ in learning the classifier. Second, we present an active learning strategy to define $\mathcal{D}^p_l \subset \mathcal{D}^p$ when $\mathcal{D}^p$ is initially fully unlabeled.

The remainder of the paper is organized as follows. A detailed description of the proposed method is provided in Section 2, followed by description of a fast learning algorithm in Section 3 and a theoretical discussion in 4. In Section 5 we present a method to actively define $\mathcal{D}^p_l$ when $\mathcal{D}^p_l$ is initially empty. We demonstrate example results in Section 6. Finally, Section 7 contains the conclusions.

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

2. Migratory-Logit: Learning Jointly on the Primary and Auxiliary Data

We assume $\mathcal{D}^p_l$ are fixed and nonempty, and without loss of generality, we assume $\mathcal{D}^p_l$ are always indexed prior to $\mathcal{D}^p_u$, i.e., $\mathcal{D}^p_l = \{(\mathbf{x}^p_i, y^p_i)\}_{i=1}^{N^p_l}$ and $\mathcal{D}^p_u = \{(\mathbf{x}^p_i, y^p_i) : y^p_i \text{ missing}\}_{i=N^p_l+1}^{N^p}$. We use N^a, N^p, and N^p_l to denote the size (number of data points) of $\mathcal{D}^a$, $\mathcal{D}^p$, and $\mathcal{D}^p_l$, respectively. In Section 5 we discuss how to actively determine $\mathcal{D}^p_l$ when $\mathcal{D}^p_l$ 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 $(\mathbf{x}^p_i, y^p_i) \in \mathcal{D}^p_l$, we follow standard logistic regression to write

$$\Pr(y^p_i|\mathbf{x}^p_i; \mathbf{w}) = \sigma(y^p_i \mathbf{w}^T \mathbf{x}^p_i) \tag{2}$$

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

$$\Pr(y^a_i|\mathbf{x}^a_i; \mathbf{w}, \mu_i) = \sigma(y^a_i \mathbf{w}^T \mathbf{x}^a_i + y^a_i \mu_i) \tag{3}$$

where μ_i is an auxiliary variable. Assuming the examples in $\mathcal{D}^p_l$ and $\mathcal{D}^a$ are drawn i.i.d., we have the log-likelihood function

$$\begin{aligned} &\ell(\mathbf{w}, \boldsymbol{\mu}; \mathcal{D}^p_l \cup \mathcal{D}^a) \\ &= \textstyle\sum_{i=1}^{N^p_l} \ln \sigma(y^p_i \mathbf{w}^T \mathbf{x}^p_i) + \sum_{i=1}^{N^a} \ln \sigma(y^a_i \mathbf{w}^T \mathbf{x}^a_i + y^a_i \mu_i) \end{aligned} \tag{4}$$

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

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

to change freely, 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:

$$\begin{aligned} \max_{\mathbf{w},\boldsymbol{\mu}} \quad & \ell(\mathbf{w}, \boldsymbol{\mu}; \mathcal{D}_l^p \cup \mathcal{D}^a) && (5)\\ \text{subject to} \quad & \tfrac{1}{N^a}\textstyle\sum_{i=1}^{N^a} y_i^a \mu_i \le C, \quad C \ge 0 && (6)\\ & y_i^a \mu_i \ge 0, \quad i = 1, 2, \cdots, N^a && (7) \end{aligned}$$

where the inequalities in (7) 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 (7), 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 (5)-(7) is referred to as "Migratory-Logit" or "M-Logit".

The C in (6) 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 4.

3. Fast Learning Algorithm

The optimization problem in (5), (6), and (7) 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 μ's can consume most of the computational time.

In this section, we give a fast algorithm for training the M-Logit, by taking a block-coordinate ascent approach [Bertsekas, 1999], in which we alternately solve for $\mathbf{w}$ and $\boldsymbol{\mu}$, keeping one fixed when solving 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 4 contains a discussion that helps to understand the theorem from an intuitive perspective.

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

$$n = \max\{m : m b_m - \textstyle\sum_{i=1}^m b_i \le R, 1 \le m \le N\} \quad (8)$$

Then the problem

$$\begin{aligned} \max_{\{z_i\}} \quad & \textstyle\sum_{i=1}^N f(b_i + z_i) && (9)\\ \textit{subject to} \quad & \textstyle\sum_{i=1}^N z_i \le R, \quad R \ge 0 && (10)\\ & z_i \ge 0, \quad i = 1, 2, \cdots, N && (11) \end{aligned}$$

has a unique global solution

$$z_i = \begin{cases} \frac{1}{n}\sum_{j=1}^n b_j + \frac{1}{n}R - b_i, & 1 \le i \le n \\ 0, & n < i \le N \end{cases} \quad (12)$$

For a fixed $\mathbf{w}$, the problem in (5)-(7) is simplified to maximizing $\sum_{i=1}^{N^a} \ln \sigma(y_i^a \mathbf{w}^T \mathbf{x}_i^a + y_i^a \mu_i)$ with respect to $\boldsymbol{\mu}$, subject to $\frac{1}{N^a}\sum_{i=1}^{N^a} y_i^a \mu_i \le C$, $C \ge 0$, and $y_i^a \mu_i \ge 0$ for $i = 1, 2, \cdots, N^a$. Clearly $\ln \sigma(z)$ is a twice continuously differentiable function of z and its second derivative $\frac{\partial^2}{\partial z^2} \ln \sigma(z) = -\sigma(z)\sigma(-z) < 0$ 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 $y_{k_1}^a \mathbf{w}^T \mathbf{x}_{k_1}^a \le y_{k_2}^a \mathbf{w}^T \mathbf{x}_{k_2}^a \le \cdots \le y_{k_{N^a}}^a \mathbf{w}^T \mathbf{x}_{k_{N^a}}^a$, where $k_1, k_2, \cdots, k_{N^a}$ is a permutation of $1, 2, \cdots, N^a$. Then we can write the solution of $\{\mu_i\}$ analytically,

$$\mu_{k_i} = \begin{cases} \frac{1}{n} y_{k_i}^a \sum_{j=1}^n y_{k_j}^a \mathbf{w}^T \mathbf{x}_{k_j}^a \\ \quad + \frac{N^a}{n} y_{k_i}^a C - \mathbf{w}^T \mathbf{x}_{k_i}^a \\ 0, \end{cases} \begin{array}{l} , \quad 1 \le i \le n \\ \\ \quad n < i \le N^a \end{array} \quad (13)$$

where

$$n = \max\Big\{m : m y_{k_m}^a \mathbf{w}^T \mathbf{x}_{k_m}^a - \textstyle\sum_{i=1}^m y_{k_i}^a \mathbf{w}^T \mathbf{x}_{k_i}^a \le N^a C, \; 1 \le m \le N^a\Big\} \quad (14)$$

For a fixed $\boldsymbol{\mu}$, we use the standard gradient-based method [Bertsekas, 1999] to find $\mathbf{w}$. The main procedures of the fast training algorithm for M-Logit are summarized in Table 1, where the gradient $\nabla_{\mathbf{w}} \ell$ and the Hessian matrix $\nabla_{\mathbf{w}}^2 \ell$ are computed from (4).

4. 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 a major "wealth" $y_i^a \mathbf{w}^T \mathbf{x}_i^a$ from $\mathbf{w}$ and an 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 $y_{k_1}^a \mathbf{w}^T \mathbf{x}_{k_1}^a$), gets a portion $y_{k_1}^a \mu_{k_1}$ from $N^a C$ first,

Table 1. Fast Learning Algorithm of M-Logit

Input: $\mathcal{D}^a \cup \mathcal{D}^p_l$ 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 $\{y^a_i \mathbf{w}^T \mathbf{x}^a_i\}_{i=1}^{N^a}$ in ascending order. Assume the result is $y^a_{k_1}\mathbf{w}^T\mathbf{x}^a_{k_1} \le y^a_{k_2}\mathbf{w}^T\mathbf{x}^a_{k_2} \le \cdots \le y^a_{k_{N^a}}\mathbf{w}^T\mathbf{x}^a_{k_{N^a}}$, where $k_1, k_2, \cdots, k_{N^a}$ is a permutation of $1, 2, \cdots, N^a$.
7. Find the n using (14).
8. Update the auxiliary variables $\{\mu_i\}_{i=1}^{N^a}$ using (13).
9. Check the convergence of ℓ: exit and output $\mathbf{w}$ and $\{\mu_i\}_{i=1}^{N^a}$ if converged; go back to 2 otherwise.

and as soon as the total wealth $y^a_{k_1}\mathbf{w}^T\mathbf{x}^a_{k_1} + y^a_{k_1}\mu_{k_1}$ reaches the wealth of the second poorest $\mathbf{x}^a_{k_2}$, N^aC becomes equally distributed to $\mathbf{x}^a_{k_1}$ and $\mathbf{x}^a_{k_2}$ such that their total wealths are always equal. Then, as soon as $y^a_{k_1}\mathbf{w}^T\mathbf{x}^a_{k_1} + y^a_{k_1}\mu_{k_1} = y^a_{k_2}\mathbf{w}^T\mathbf{x}^a_{k_2} + y^a_{k_2}\mu_{k_2}$ reach the wealth of the third poorest, N^aC becomes equally distributed to three of them to make them equally rich. The distribution continues in this way until the budget N^aC is used up. The "poorest-first" rule is essentially a result of the concavity of the logarithmic sigmoid function $\ln\sigma(\cdot)$. The goal is to maximize $\sum_{i=1}^{N^a}\ln\sigma(y^a_i\mathbf{w}^T\mathbf{x}^a_i + y^a_i\mu_i)$. The concavity of $\ln\sigma(\cdot)$ dictates that for any given portion of N^aC, 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 wealth from $\mathbf{w}$ and is poor, 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 $y^a_i\mu_i$ whittles down the participation of $\mathbf{x}^a_i$ in learning $\mathbf{w}$. This is easily seen from the contribution of $(\mathbf{x}^a_i, y^a_i)$ to $\nabla_{\mathbf{w}}\ell$ and $\nabla^2_{\mathbf{w}}\ell$, which are obtained from (4) as $\sigma(-y^a_i\mathbf{w}^T\mathbf{x}^a_i - y^a_i\mu_i)y^a_i\mathbf{x}^a_i$ and $-\sigma(-y^a_i\mathbf{w}^T\mathbf{x}^a_i - y^a_i\mu_i)\sigma(y^a_i\mathbf{w}^T\mathbf{x}^a_i + y^a_i\mu_i)\mathbf{x}^a_i\mathbf{x}^{a\,T}_i$, respectively. When $y^a_i\mu_i$ is large, $\sigma(-y^a_i\mathbf{w}^T\mathbf{x}^a_i - y^a_i\mu_i)$ is close to zero and hence the contributions of $(\mathbf{x}^a_i, y^a_i)$ to $\nabla_{\mathbf{w}}\ell$ and $\nabla^2_{\mathbf{w}}\ell$ are ignorable. We in fact do not need an infinitely large $y^a_i\mu_i$ to make the contributions of $\mathbf{x}^a_i$ ignorable, because $\sigma(\mu)$ is almost saturated at $\mu = \pm 6$. If $y^a_i\mathbf{w}^T\mathbf{x}^a_i = -6$, $\sigma(-y^a_i\mathbf{w}^T\mathbf{x}^a_i) = 0.9975$, implying a large contribution of $(\mathbf{x}^a_i, y^a_i)$ to $\nabla_{\mathbf{w}}\ell$, which happens when $\mathbf{w}$ assigns $\mathbf{x}^a_i$ to the correct class y^a_i with probability of $\sigma(y^a_i\mathbf{w}^T\mathbf{x}^a_i) = \sigma(-6) = 0.0025$ only. In this nearly worst case, a compensation of $y^a_i\mu_i = 12$ can effectively remove the contribution of $(\mathbf{x}^a_i, y^a_i)$ because $\sigma(-y^a_i\mathbf{w}^T\mathbf{x}^a_i - y^a_i\mu_i) = \sigma(6 - 12) = \sigma(-6) = 0.0025$. 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 so because we want N^aC 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 M-Logit is relatively robust to C, and this will be demonstrated in Section 6.2 using an example data set.

5. Active Selection of $\mathcal{D}^p_l$

In Section 2 we assumed that $\mathcal{D}^p_l$ had already been determined. In this section we describe how $\mathcal{D}^p_l$ can be actively selected from $\mathcal{D}^p$, based on the Fisher information matrix [Fedorov, 1972, MacKay, 1992]. The approach is known as active learning [Cohn et al., 1995, Krogh & Vedelsby, 1995].

Let $\mathbf{Q}$ denote the Fisher information matrix of $\mathcal{D}^p_l \cup \mathcal{D}^a$ about $\mathbf{w}$. By definition of the Fisher information matrix [Cover & Thomas, 1991], $\mathbf{Q} = \mathbb{E}_{\{y^p_i\},\{y^a_i\}}\frac{\partial\ell}{\partial\mathbf{w}}\frac{\partial\ell}{\partial\mathbf{w}}^T$, and substituting (4) into this equation gives (a brief derivation is given in the appendix)

$$\mathbf{Q} = \sum_{i=1}^{N^p_l}\sigma^p_i(1-\sigma^p_i)\mathbf{x}^p_i\mathbf{x}^{p\,T}_i + \sum_{i=1}^{N^a}\sigma^a_i(1-\sigma^a_i)\mathbf{x}^a_i\mathbf{x}^{a\,T}_i \quad (15)$$

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

It is well known the inverse Fisher information $\mathbf{Q}^{-1}$ lower bounds the covariance matrix of the estimated $\mathbf{w}$ [Cover & Thomas, 1991]. 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 $\mathbf{Q} = \sum_{i=1}^{N^a} \sigma_i^a(1-\sigma_i^a)\mathbf{x}_i^a \mathbf{x}_i^{a\,T}$. 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: $\mathbf{Q} \leftarrow \mathbf{Q} + \sigma_i^p(1-\sigma_i^p)\mathbf{x}_i^p(\mathbf{x}_i^p)^T$. At each iteration, the selection is based on

$$\begin{aligned} &\max_{\mathbf{x}_i^p \in \mathcal{D}_u^p} \det\left\{\mathbf{Q} + \sigma_i^p(1-\sigma_i^p)\mathbf{x}_i^p(\mathbf{x}_i^p)^T\right\} \\ &= \max_{\mathbf{x}_i^p \in \mathcal{D}_u^p} \left\{1 + \sigma_i^p(1-\sigma_i^p)(\mathbf{x}_i^p)^T\mathbf{Q}^{-1}\mathbf{x}_i^p\right\} \quad (16) \end{aligned}$$

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

Evaluation of (16) requires the true values of $\mathbf{w}$ and $\{\mu_i\}$, which are not known *a priori*. We follow Fedorov (1972) 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.

6. Results

In this section the performance of M-Logit is demonstrated and compared to the standard logistic regression, using test error rate as the performance index. The M-Logit 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 5. 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 $\mathcal{D}^p$ used for M-Logit. The four classifiers are tested on $\mathcal{D}_u^p = \mathcal{D}^p \setminus \mathcal{D}_l^p$, using the following decision rule: declare $y^p = -1$ if $\sigma(\mathbf{w}^T\mathbf{x}^p) \leq 0.5$ and $y^p = 1$ otherwise, for any $\mathbf{x}^p \in \mathcal{D}_u^p$.

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

6.1. A toy Example

In the first example, the primary data are simulated as two bivariate Gaussian distributions representing class "−1" and class "+1", respectively. In particularly, 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 $\boldsymbol{\mu}_0 = [0,0]^T$, $\boldsymbol{\mu}_1 = [2.3, 2.3]^T$, 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 [Zadrozny, 2004]. 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, $w_0 = -1$, $w_1 = \exp(1)$, $K(\mathbf{x}^p, \boldsymbol{\mu}_0^s; \boldsymbol{\Sigma}) = \exp\{-0.5(\mathbf{x}^p - \boldsymbol{\mu}_0^s)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}^p - \boldsymbol{\mu}_0^s)\}$ with $\boldsymbol{\mu}_0^s = [2,1]^T$, and $K(\mathbf{x}^p, \boldsymbol{\mu}_1^s; \boldsymbol{\Sigma}) = \exp\{-0.5(\mathbf{x}^p - \boldsymbol{\mu}_1^s)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}^p - \boldsymbol{\mu}_1^s)\}$ with $\boldsymbol{\mu}_1^s = [0,3]^T$. We obtain 150 samples of $\mathcal{D}^p$ and 150 samples of $\mathcal{D}^a$, which are shown in Figure 3.

The M-Logit and logistic regression classifiers are trained and tested as explained at the beginning of this section. The test error rates are shown in Figure 1 and Figure 2, as a function of number of primary labeled data used in training. The $\mathcal{D}_l^p$ in Figure 1 are randomly selected and the $\mathcal{D}_l^p$ in Figure 2 are actively selected as described in Section 5.

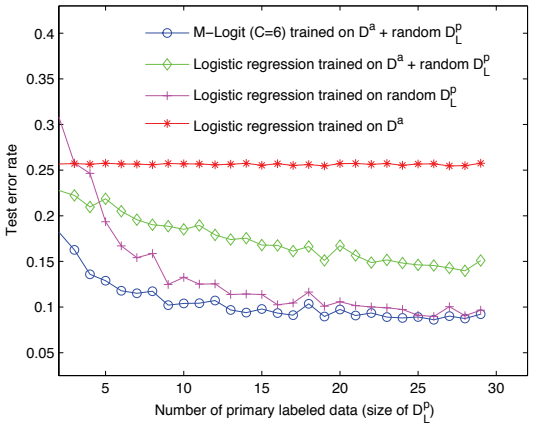

Figure 1. Test error rates of M-Logit and logistic regression on the toy data, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are randomly selected from $\mathcal{D}^p$. The error rates are an average over 50 independent trials of random selection of $\mathcal{D}_l^p$.

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

- The M-Logit consistently outperforms the three standard logistic regression classifiers, by a considerable margin. This improvement is a result of properly fusing $\mathcal{D}^a$ and $\mathcal{D}_l^p$, with $\mathcal{D}^a$ determining the classifier under the guidance of few $\mathcal{D}_l^p$.

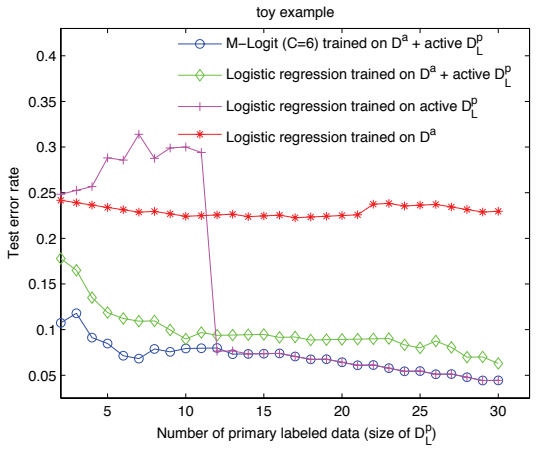

Figure 2. Error rates of M-Logit and logistic regression on the toy data, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$, using the method in Section 5.

- The performance of the logistic regression trained on $\mathcal{D}_l^p$ alone changes significantly 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 11 to iteration 12 in Figure 2 may be because the label found at iteration 12 is critical to determining $\mathbf{w}$.

- The logistic regression trained on $\mathcal{D}^a$ alone performs significantly worse than M-Logit, reflecting a marked mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$.

- 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 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 be correctly classified along with $\mathcal{D}^p$, if the mismatch is not compensated.

- As $\mathcal{D}_l^p$ grows, the logistic regression trained on $\mathcal{D}_l^p$ alone finally approaches to M-Logit, showing that without the interference of $\mathcal{D}^a$, a sufficient $\mathcal{D}_l^p$ can define a correct classifier.

- All four classifiers benefit from the actively selected $\mathcal{D}_l^p$, this is consistent with the general observation with active learning [Cohn et al., 1995, Krogh & Vedelsby, 1995].

To better understand the active selection process, we show in Figure 3 the first few iterations of active learning. Iteration 0 corresponds to the initially empty $\mathcal{D}_l^p$, and iterations 1, 5, 10, 13 respectively correspond to 1, 5, 10, 13 data points selected accumulatively 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 can be seen in Figure 3, the decision boundary does not change much after 10 data are selected, demonstrating convergence.

In Figure 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 small participation in determining $\mathbf{w}$. The auxiliary data that cannot be correctly classified along with the primary data are de-emphasized by the M-Logit. Usually the auxiliary data near the decision boundary are de-emphasized.

6.2. Results on the Wisconsin Breast Cancer Databases

In the second example we consider the Wisconsin Breast Cancer Databases from the UCI Machine Learning Repository. The data set consist of 569 instances with feature dimensionality 30. We randomly partition the data set into two subsets, one with 228 data points and the other with 341 data points. The first is used as $\mathcal{D}^p$, and the second as $\mathcal{D}^a$. We artificially make $\mathcal{D}^a$ mismatched with $\mathcal{D}^p$ by introducing errors into the labels and adding noise to the features. Specifically, we make changes to 50% randomly chosen $(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a$: change the signs of y_i^a and add 0 dB white Gaussian noise to $\mathbf{x}_i^a$. We then proceed, as in Section 6.1, to training and testing the four classifiers. We again consider both random $\mathcal{D}_l^p$ and actively selected $\mathcal{D}_l^p$. The test errors are summarized in Figures 4 and 5. The results are essentially consistent with those in Figures 1 and 2, extending the observations we made there to the real data here. It is particularly noted that the mismatch between $\mathcal{D}^a$ and $\mathcal{D}^p$ here is more prominent than in the toy data, as manifested by the error rates of logistic regression trained alone on $\mathcal{D}^a$. This makes M-Logit more advantageous in the comparison: not only does it give the best results but it also converges faster than others with the size of $\mathcal{D}_l^p$.

To examine the effect of C on the performance of M-Logit, we present in Figure 6 the test error rates of M-Logit using five different C: $C = 2, 4, 6, 8, 10$. Here the $\mathcal{D}_l^p$ are determined by active learning as described in Section 5. Clearly, the results for the 5 different C's are almost indistinguishable. This relative insensitivity of M-Logit to C may partly be attributed to the adaptivity brought about by active learning. With different C, the $\mathcal{D}_l^p$ are also selected differently, thus counteracting the effect of C and keeping M-Logit ro-

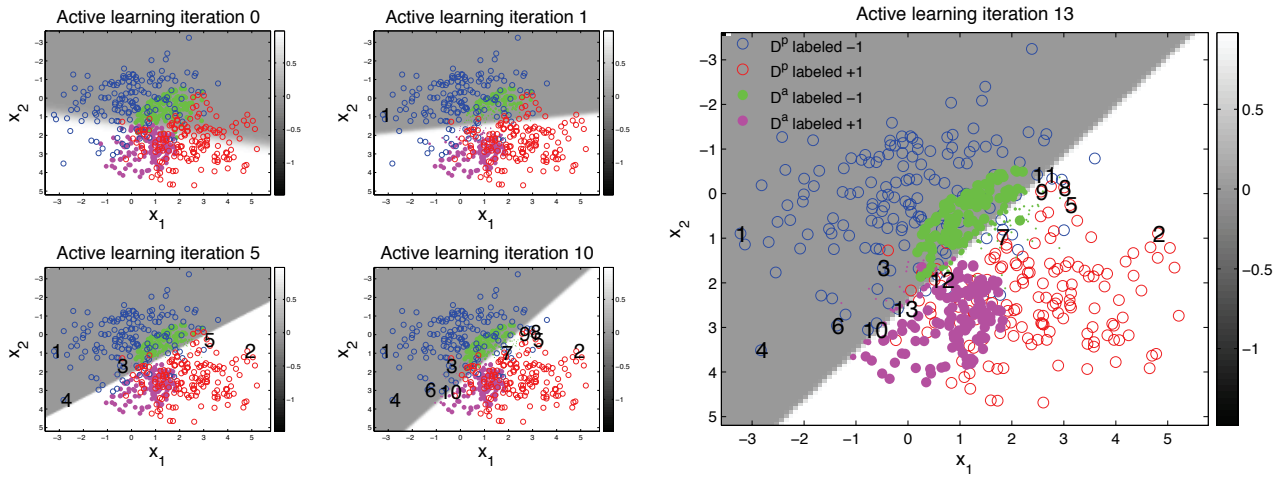

Figure 3. Illustration of active data selection by M-Logit. Only iterations 0,1,5,10,13 are shown. The different symbols are defined as: blue ○ = $\mathcal{D}^p$ labeled "−1", red ○ = $\mathcal{D}^p$ labeled "+1", green ● = $\mathcal{D}^a$ labeled "−1", and magenta ● = $\mathcal{D}^a$ labeled "+1". The 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).

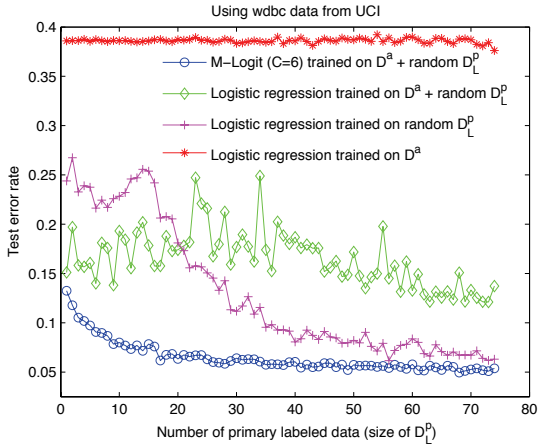

Figure 4. Test error rates of M-Logit and logistic regression on the Wisconsin Breast Cancer Databases of UCI, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are randomly selected from $\mathcal{D}^p$. The error rates are an average over 50 independent trials of random selection of $\mathcal{D}_l^p$.

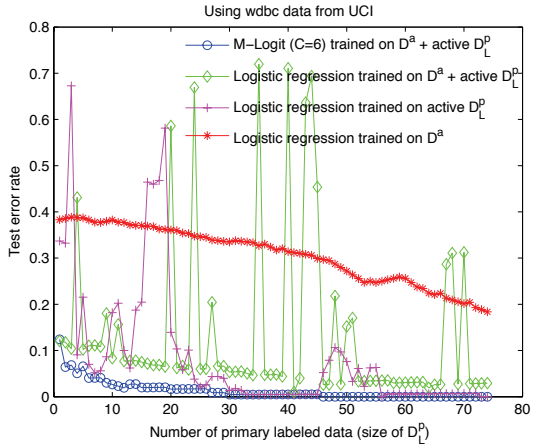

Figure 5. Test error rates of M-Logit and logistic regression on the Wisconsin Breast Cancer Databases of UCI, as a function of size of $\mathcal{D}_l^p$. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$, using the method in Section 5.

bust.

7. Conclusions

We have proposed an algorithm, the "Migratory-Logit" or M-Logit, which is capable of learning in the presence of mismatch 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 be correctly classified along with $\mathbf{x}^p$ by the classifier. The migrations of $\mathcal{D}^a$ are controlled by the inequality constraint $\frac{1}{N^a}\sum_{i=1}^{N^a} y_i^a \mu_i \leq C$, 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, we have presented an method to actively selecting $\mathcal{D}_l^p$, which enhances the adaptivity of the entire learning process. We have developed a fast

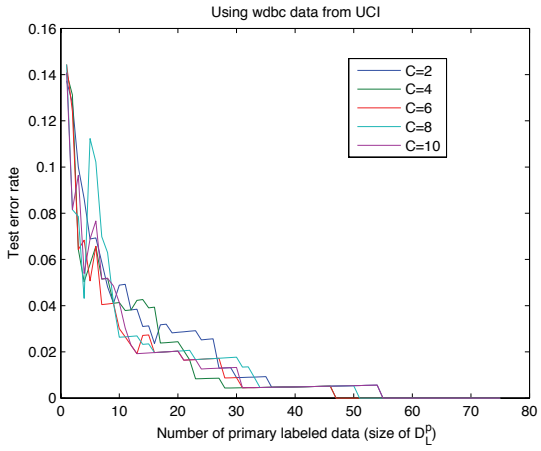

Figure 6. Comparison of M-Logit with different C's, using the Wisconsin Breast Cancer Databases of UCI. The primary labeled data $\mathcal{D}_l^p$ are actively selected from $\mathcal{D}^p$, using the method in Section 5.

learning algorithm to enhance the ability of M-Logit to handle large auxiliary data sets.

The results from both toy data and the Wisconsin Breast Cancer Databases show that M-Logit 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.

References

Bertsekas, D. P. (1999). *Nonlinear programming (2nd edition)*. Athena Scientific.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1995). Active learning with statistical models. *Advances in Neural Information Processing Systems*, *7*, 705–712.

Cover, T. M., & Thomas, J. (1991). *Elements of information theory*. New York: Wiley.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Academic Press.

Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica*, *47*, 153–161.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, *7*, 231–238.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*, 590–604.

Nigam, K., & et al. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, *39(2/3)*, 103–134.

Wu, P., & Dietterich, T. G. (2004). Improving svm accuracy by training on auxiliary data sources. *Proceedings of the 21st ICML*.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. *Proceedings of the 21st ICML*.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost sensitive learning by cost-proportionate example weighting. *Proceedings of the Third IEEE International Conference on Data Mining*, 435–442.

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 (9) is equivalent to

$$\max_{\{z_i\}} \quad \sum_{i=1}^N \int_0^{z_i} f'(b_i + x)dx \tag{A-1}$$

Because $f''(z) < 0$, we have for any $\tau_1 \le \tau_2$ that $f'(\tau_1 + x) \ge f'(\tau_2 + x)$ and consequently

$$\int_0^{\Delta} f'(\tau_1 + x)dx \ge \int_0^{\Delta} f'(\tau_2 + x)dx \quad \forall \quad \tau_1 \le \tau_2 \quad \text{and} \quad \Delta \ge 0 \tag{A-2}$$

By (8), there exists $0 \le r < n(b_{n+1} - b_n)$ 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 $k = 1, \cdots, n-1$, and $\Delta_n = r/n$. We now use (A-2) to distribute $\Delta_1, 2\Delta_2, \cdots, n\Delta_n$ to $z_1, z_2, \cdots, z_N$ such that the resulting $\{z_i\}$ maximize (A-1). As $\Delta_k \ge 0$ for $k = 1, \cdots, n$, 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 (10) and (11) are automatically satisfied.

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

As $\Delta_1 = b_2 - b_1 \ge 0$, by (A-2), $\int_0^{\Delta_1} f'(b_1 + x)dx \ge \int_0^{\Delta_1} f'(b_2 + x)dx$, 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 $\Delta_2 = b_3 - b_2 \ge 0$, by (A-2), $\int_0^{\Delta_2} f'(b_2 + x)dx \ge \int_0^{\Delta_2} f'(b_3 + x)dx$, 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 $z_1, z_2, \cdots, z_k$. After the distribution of $k\Delta_k$, $k = 1, 2, \cdots, n$, we have $z_k = \sum_{i=k}^n \Delta_i$ for $k = 1, 2, \cdots, n$ and $z_k = 0$ for $k = n+1, n+2, \cdots, N$, which is equal to the solution in (12). Because the problem is strictly concave, the solution is unique and globally optimal. □

Derivation of Equation (15): By definition of logistic regression, $\mathbf{w}$ is the parameter of the conditional distribution $\Pr(y|\mathbf{x}) = \sigma(y\mathbf{w}^T\mathbf{x})$, with $\mathbf{x}$ given and fixed. Let $\mathbf{g} = \partial \ln \sigma(y\mathbf{w}^T\mathbf{x})/\partial \mathbf{w} = [1 - \sigma(y\mathbf{w}^T\mathbf{x})]y\mathbf{x}$. Then $\mathbb{E}_y(\mathbf{g}\mathbf{g}^T) = \sum_{y=-1,1} \sigma(y\mathbf{w}^T\mathbf{x})[1 - \sigma(y\mathbf{w}^T\mathbf{x})]^2\mathbf{x}\mathbf{x}^T$. Using $\sigma(-\mathbf{w}^T\mathbf{x}) = 1 - \sigma(\mathbf{w}^T\mathbf{x})$, we obtain $\mathbb{E}_y(\mathbf{g}\mathbf{g}^T) = \sigma(\mathbf{w}^T\mathbf{x})[1 - \sigma(\mathbf{w}^T\mathbf{x})]\mathbf{x}\mathbf{x}^T$. Summing $\mathbb{E}(\mathbf{g}\mathbf{g}^T)$ over all primary and auxiliary data points (assuming the data are independent), we obtain the formula of $\mathbf{Q}$. □