On Classification with Incomplete Data

David Williams, *Member*, *IEEE*, Xuejun Liao, *Senior Member*, *IEEE*, Ya Xue, Lawrence Carin, *Fellow*, *IEEE*, and Balaji Krishnapuram

Abstract—We address the incomplete-data problem in which feature vectors to be classified are missing data (features). A (supervised) logistic regression algorithm for the classification of incomplete data is developed. Single or multiple imputation for the missing data is avoided by performing analytic integration with an estimated conditional density function (conditioned on the observed data). Conditional density functions are estimated using a Gaussian mixture model (GMM), with parameter estimation performed using both Expectation-Maximization (EM) and Variational Bayesian EM (VB-EM). The proposed supervised algorithm is then extended to the semisupervised case by incorporating graph-based regularization. The semisupervised algorithm utilizes all available data—both incomplete and complete, as well as labeled and unlabeled. Experimental results of the proposed classification algorithms are shown.

Index Terms—Classification, incomplete data, missing data, supervised learning, semisupervised learning, imperfect labeling.

1 Introduction

The incomplete-data problem in which certain features are missing from particular feature vectors,exists in a wide range of fields, including social sciences, computer vision, biological systems, and remote sensing. For example, partial responses in surveys are common in the social sciences, leading to incomplete data sets with arbitrary patterns of missing data. In remote sensing applications, incomplete data can result when only a subset of sensors (e.g., radar, infrared, acoustic) are deployed at certain regions. Increasing focus in the future on using (and fusing data from) multiple sensors or information sources (e.g., [21], [11]) will make such incomplete-data problems increasingly common.

Incomplete-data problems are often circumvented via imputation—the "completion" of missing data by filling in specific values. Common imputation schemes include "completing" missing data with zeros, the unconditional mean, or the conditional mean (if one has an estimate for the distribution of missing features given the observed features, $p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})$). More sophisticated methods that have been used to complete missing data—and which can also be viewed as single imputation schemes—include semidefinite programming [7] and the *em* algorithm [21]. Because imputation treats the missing data as fixed known data, though, the uncertainty of the missing data is ignored [18].

The method of multiple imputation [19] instead generates $M > 1$ samples for every missing feature. This imputation (sampling) is performed only because the desired posterior distribution of a parameter involves an intractable integral (details on multiple imputation as applied to classification problems are provided in Section 4). The intractable integral can be avoided by requiring the data (i.e., features) to be discrete [9]. This discreteness assumption permits a "weighted EM" algorithm [9] from which maximum likelihood parameter estimates (e.g., classifier weights) can be obtained. Although this method—developed for generalized linear models with incomplete data—avoids imputation, it does not extend to the case of continuous features. An accessible introduction to, and summary of, the subject of dealing with missing data can be found in [20].

In this work, we develop supervised and semisupervised classification algorithms that explicitly account for incomplete data. We first tackle the incomplete (continuous) data problem for (supervised) logistic regression classification in a principled manner, avoiding explicit imputation. When calculating the posterior distribution of a parameter, it is proper to integrate out missing data [4]:

$$p(y_i|\mathbf{x}_i^{o_i}) = \int p(y_i|\mathbf{x}_i^{m_i}, \mathbf{x}_i^{o_i}) p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}) d\mathbf{x}_i^{m_i}, \qquad (1)$$

where $\mathbf{x}_i^{o_i}$ are the observed data (i.e., features) and $\mathbf{x}_i^{m_i}$ are the missing data. This integral is intractable in general. However, in the case of logistic regression (with y_i the class label), this integral can be solved analytically using two minor assumptions. The first assumption is that $p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})$ is a Gaussian mixture model (GMM). This assumption is mild since it is well-known that a mixture of Gaussians can approximate any distribution. The second (highly accurate) assumption is that the sigmoid function can be approximated as a probit function (i.e., the cumulative distribution function of a Gaussian). Since the integral in (1) can be solved analytically, the likelihood (in a supervised framework) can be maximized—in a manner analogous to the complete-data case—to obtain classifier weights. Once the weights are obtained, the classification algorithm can be applied to classify incomplete testing data.

We also extend this proposed supervised algorithm to the semisupervised case by using graph-based regularization. In this form, our algorithm utilizes all available data: both incomplete and complete data, as well as both labeled and unlabeled data. To our knowledge, no semisupervised algorithms exist for incomplete-data classification.

- *D. Williams, X. Liao, Y. Xue, and L. Carin are with the Department of Electrical and Computer Engineering, Duke University, Box 90291, Durham, NC 27708-0291.*
  E-mail: {dpw, yx10, lcarin}@ee.duke.edu, xjliao@ece.duke.edu.
- *B. Krishnapuram is with Siemens Medical Solutions, Malvern, PA 19355.*
  E-mail: balaji.krishnapuram@siemens.com.

Manuscript received 4 Aug. 2005; revised 13 May 2006; accepted 26 July 2006; published online 15 Jan. 2007.
Recommended for acceptance by J. Buhmann.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0419-0805.

The remainder of the paper is organized as follows: In Section 2, we derive the supervised logistic regression algorithm for classification of incomplete data and, in Section 3, we extend this supervised algorithm to the semisupervised case. Experimental results for the classification algorithms are shown in Section 4, followed by a discussion in Section 5. Concluding remarks and suggestions for future work are made in Section 6.

2 Supervised Classification of Incomplete Data

The work in this paper assumes that the missing data is either missing completely at random (MCAR) or missing at random (MAR), meaning that the values of the data have no affect on whether the data is missing or not (see [18], [5] for more details). When the missing data is not missing at random (NMAR), a model for the missing data must be created for the specific data set under study. Because of this fact, addressing the incomplete data problem when data is not missing at random is inherently a problem-specific issue. That is, a general algorithm cannot be constructed to address arbitrary data sets.

Assume we have a set of labeled incomplete data,

$$\mathcal{D}_L = \{(\mathbf{x}_i, y_i, \epsilon_i, m_i) : \mathbf{x}_i \in \mathbb{R}^d, x_{ia} \quad \text{missing} \quad \forall a \in m_i\}_{i=1}^{N_L}, \tag{2}$$

where $\mathbf{x}_i$ is the ith vector, labeled as $y_i \in \{-1, 1\}$ with known labeling error rate $\epsilon_i \in [0, 0.5)$; the features in $\mathbf{x}_i$ indexed by m_i (i.e., $x_{ia}, a \in m_i$) are missing. Each $\mathbf{x}_i$ has its own (possibly unique) set of missing features, m_i. One special case occurs when a subset of data share common missing features, as with multisensor data where the common missing features result from a sensor that has not collected data.

In logistic regression (with a hyperplane classifier) [14], the probability of label y_i given $\mathbf{x}_i$ is $p(y_i|\mathbf{x}_i, \mathbf{w}) = \sigma(y_i \mathbf{w}^T \mathbf{x}_i)$, where $\sigma(\nu) = (1 + \exp(-\nu))^{-1}$ is the sigmoid function and $\mathbf{w}$ constitutes a classifier. Accounting for imperfections in the labeling process arising from a known labeling error rate ϵ_i, the probability of label y_i given $\mathbf{x}_i$ and ϵ_i is [17]

$$p(y_i|\mathbf{x}_i, \epsilon_i, \mathbf{w}) = \epsilon_i + (1 - 2\epsilon_i)\sigma(y_i \mathbf{w}^T \mathbf{x}_i). \tag{3}$$

The labeling error rate is simply the probability that a true label was flipped (corrupted) to the wrong label (e.g., $\{y_i^{\text{true}} = 1\} \rightarrow \{y_i = -1\}$). For instance, to establish the (perfect) label of data in a land mine detection task, the buried object must be excavated, a dangerous and time-consuming endeavor. An imperfect label may instead be obtained by using a handheld (labeling) sensor, with the level of confidence (or labeling error rate) tied to the historical accuracy of the sensor. Note that the standard case of perfect labels is recovered when $\epsilon_i = 0$.

We partition $\mathbf{x}_i$ into its observed and missing parts, $\mathbf{x}_i = [\mathbf{x}_i^{o_i}; \mathbf{x}_i^{m_i}]$, where $\mathbf{x}_i^{o_i} = [x_{ia}, a \in o_i]^T$, $\mathbf{x}_i^{m_i} = [x_{ia}, a \in m_i]^T$, and $o_i = \{1, \cdots, d\} \setminus m_i$ is the (complementary) set of observed features in $\mathbf{x}_i$. We apply the same partition to $\mathbf{w}$ to obtain $\mathbf{w} = [\mathbf{w}_{o_i}; \mathbf{w}_{m_i}]$, yielding

$$p(y_i|\mathbf{x}_i, \epsilon_i, \mathbf{w}) = \epsilon_i + (1 - 2\epsilon_i)\sigma(y_i(\mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i} + \nu_i)), \tag{4}$$

where $\nu_i = \mathbf{w}_{m_i}^T \mathbf{x}_i^{m_i}$. Because $\mathbf{x}_i^{m_i}$ (and, hence, ν_i) is missing, (4) cannot be evaluated. By integrating out the missing data $\mathbf{x}_i^{m_i}$, the needed probability of y_i given the observed features $\mathbf{x}_i^{o_i}$ can be written as

$$p(y_i|\mathbf{x}_i^{oi}, \epsilon_i, \mathbf{w}) = \int p(y_i|\mathbf{x}_i^{m_i}, \mathbf{x}_i^{o_i}, \epsilon_i, \mathbf{w}) p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}) d\mathbf{x}_i^{m_i} \tag{5}$$

$$= \epsilon_i + (1 - 2\epsilon_i) \int \sigma(y_i(\mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i} + \nu_i)) p(\nu_i|\mathbf{x}_i^{o_i}) d\nu_i. \tag{6}$$

It is important to note that the integral in (5) is, in general, multidimensional, while the integral in (6) is one-dimensional. The integration in (6) can be performed by making two minor assumptions. First, we assume that $p(\mathbf{x}_i)$ is a GMM:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_i^{o_i} \\ \mathbf{x}_i^{m_i} \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_k^{o_i} \\ \boldsymbol{\mu}_k^{m_i} \end{bmatrix}, \begin{bmatrix} \Sigma_k^{o_i o_i} & (\Sigma_k^{m_i o_i})^T \\ \Sigma_k^{m_i o_i} & \Sigma_k^{m_i m_i} \end{bmatrix} \right), \tag{7}$$

where the π_k are the nonnegative mixture weights that sum to unity; necessarily, $p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})$ is a GMM as well. The Expectation-Maximization (EM) [3] and Variational Bayesian EM (VB-EM) [2], [1] formulations for building the required GMM is described in Appendix A, which can be found at http://computer.org/tpami/archives.htm.

Because of the linear relation, $\nu_i = \mathbf{w}_{m_i}^T \mathbf{x}_i^{m_i}$, $p(\nu_i|\mathbf{x}_i^{o_i})$ is also a GMM,

$$p(\nu_i|\mathbf{x}_i^{o_i}) = \sum_{k=1}^{K} \delta_k^i \mathcal{G}\left(\frac{\nu_i - \zeta_k^i}{\alpha_k^i}\right), \tag{8}$$

with the parameters

$$\delta_k^i = \frac{\pi_k \mathcal{N}(\mathbf{x}_i^{o_i}; \boldsymbol{\mu}_k^{o_i}, \Sigma_k^{o_i o_i})}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\mathbf{x}_i^{o_i}; \boldsymbol{\mu}_\ell^{o_i}, \Sigma_k^{o_i o_i})}, \tag{9}$$

$$\zeta_k^i = \mathbf{w}_{m_i}^T \boldsymbol{\xi}_k^{m_i}, \tag{10}$$

$$\alpha_k^i = \sqrt{\mathbf{w}_{m_i}^T \boldsymbol{\Omega}_k^{m_i} \mathbf{w}_{m_i}}, \tag{11}$$

$$\boldsymbol{\xi}_k^{m_i} = \boldsymbol{\mu}_k^{m_i} + \Sigma_k^{m_i o_i} (\Sigma_k^{o_i o_i})^{-1} (\mathbf{x}_i^{o_i} - \boldsymbol{\mu}_k^{o_i}), \tag{12}$$

$$\boldsymbol{\Omega}_k^{m_i} = \Sigma_k^{m_i m_i} - \Sigma_k^{m_i o_i} (\Sigma_k^{o_i o_i})^{-1} (\Sigma_k^{m_i o_i})^T, \tag{13}$$

where $\mathcal{G}(\nu_i) = (2\pi)^{-1/2} \exp\{-\nu_i^2/2\}$ is the standard univariate Gaussian density function with zero mean and unit variance (i.e., $\mathcal{G}(u) \equiv \mathcal{N}(u; 0, 1)$).

The second assumption is that the sigmoid function can be approximated as a probit function (i.e., a Gaussian cumulative distribution function)

$$\sigma(\alpha) = \int_{-\infty}^{\alpha} \mathcal{G}\left(\frac{z}{\beta}\right) dz, \tag{14}$$

where $\beta = \frac{\pi}{\sqrt{3}}$. The accuracy of this approximation is shown in Fig. 1. (It should be noted that probit regression can be used instead of logistic regression, in which case, one would not need to invoke this second assumption.)

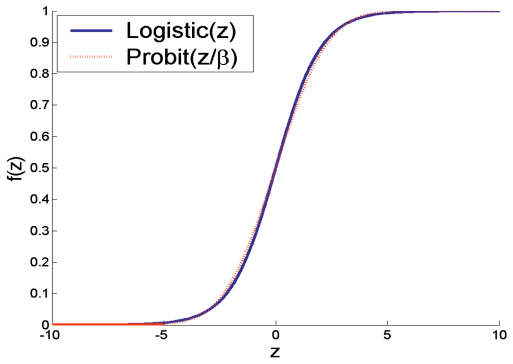


Fig. 1. Illustration of the accuracy of the approximation made between the logistic function and the (scaled) probit function.

Substituting (8) and (14) into (6), we obtain

$$p(y_i|\mathbf{x}_i^{o_i},\epsilon_i,\mathbf{w}) = \epsilon_i + (1-2\epsilon_i)\iint_{-\infty}^{y_i(\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}+\nu_i)} \mathcal{G}\left(\frac{z}{\beta}\right)dz \sum_{k=1}^{K}\delta_k^i \mathcal{G}\left(\frac{\nu_i-\zeta_k^i}{\alpha_k^i}\right)d\nu_i \tag{15}$$

$$= \epsilon_i + (1-2\epsilon_i)\iint_{-\infty}^{y_i\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z'+y_i\nu_i}{\beta}\right)dz' \sum_{k=1}^{K}\delta_k^i \mathcal{G}\left(\frac{\nu_i-\zeta_k^i}{\alpha_k^i}\right)d\nu_i \tag{16}$$

$$= \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i \int_{-\infty}^{y_i\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}}\int \mathcal{G}\left(\frac{z'+y_i\nu_i}{\beta}\right)\mathcal{G}\left(\frac{y_i\nu_i-y_i\zeta_k^i}{y_i\alpha_k^i}\right)d\nu_i\,dz' \tag{17}$$

$$= \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i \int_{-\infty}^{y_i\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z'+y_i\zeta_k^i}{\sqrt{(y_i\alpha_k^i)^2+\beta^2}}\right)dz' \tag{18}$$

$$= \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i \int_{-\infty}^{y_i\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z'+y_i\zeta_k^i}{\beta}\frac{\beta}{\sqrt{(\alpha_k^i)^2+\beta^2}}\right)dz' \tag{19}$$

$$= \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i \int_{-\infty}^{\frac{y_i\beta(\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i}+\delta_k^i)}{\sqrt{(\alpha_k^i)^2+\beta^2}}} \mathcal{G}\left(\frac{z}{\beta}\right)dz \tag{20}$$

$$= \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i\,\sigma\left(\frac{y_i\beta(\zeta_k^i+\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i})}{\sqrt{(\alpha_k^i)^2+\beta^2}}\right). \tag{21}$$

In the derivation leading to (21), (16) results from the change of variable $z' = z - y_i\nu_i$, (17) is due to exchanging the order of integrals and summation, (18) results because the convolution of two Gaussians is a Gaussian, (19) holds because $y_i^2 = 1$, and (20) results from the change of variable

$$z = \sigma\frac{\beta(z'+y_i\zeta_k^i)}{\sqrt{(\alpha_k^i)^2+\beta^2}},$$

and (21) is obtained by reverting to sigmoid representation. Thus, we have expressed $p(y_i|\mathbf{x}_i^{o_i},\epsilon_i,\mathbf{w})$ as a mixture of *sigmoids*.

Substituting (10) and (11) into (21), we obtain the probability of y_i given only the observed portion of $\mathbf{x}_i$:

$$p(y_i|\mathbf{x}_i^{o_i},\epsilon_i,\mathbf{w}) = \epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i\,\sigma\left(\frac{y_i\beta(\mathbf{w}_{m_i}^T\boldsymbol{\xi}_k^{m_i}+\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i})}{\sqrt{\mathbf{w}_{m_i}^T\boldsymbol{\Omega}_k^i\mathbf{w}_{m_i}+\beta^2}}\right). \tag{22}$$

For the incomplete and possibly imperfectly labeled data in (2), assuming the data points are independent of each other, we obtain the log-likelihood function

$$\ell(\mathbf{w}) = \log p\left(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i^{o_i}\}_{i=1}^{N_L},\{\epsilon_i\}_{i=1}^{N_L},\mathbf{w}\right) = \sum_{i=1}^{N_L}\log\left[\epsilon_i + (1-2\epsilon_i)\sum_{k=1}^{K}\delta_k^i\,\sigma\left(\frac{y_i\beta(\mathbf{w}_{m_i}^T\boldsymbol{\xi}_k^{m_i}+\mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i})}{\sqrt{\mathbf{w}_{m_i}^T\boldsymbol{\Omega}_k^i\mathbf{w}_{m_i}+\beta^2}}\right)\right]. \tag{23}$$

The objective function (23) to be maximized is not concave for two reasons: First, the concavity is destroyed by the imperfect labels resulting from ϵ_i. Even in the case of perfect labels, though, (23) is not concave because of the particular form of the argument of the sigmoid function, arising from the incomplete data. Since (23) is not concave, the solution may get trapped in local maxima. A good initialization is important, so we initialize $\mathbf{w}$ as follows: We "complete" the data set by replacing the missing features $\mathbf{x}_i^{m_i}$ with the conditional mean $\mathbb{E}[\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}] = \sum_{k=1}^{K}\delta_k^i\boldsymbol{\xi}_k^{m_i}$, where δ_k^i and $\boldsymbol{\xi}_k^{m_i}$ are defined in (9) and (12), respectively. For the initialization, we also treat all labels as perfect, artificially setting all $\epsilon_i = 0$. This "completed," "perfectly" labeled data set is submitted to the standard logistic regression to obtain $\mathbf{w}_0$, which is then used as the initialization of $\mathbf{w}$ in maximizing (23) by gradient ascent.

Thus, the maximum-likelihood (ML) logistic regression classifier $\mathbf{w}$ is obtained in the presence of missing data (and imperfect labels). Thereafter, the class predictions of an unlabeled testing data point with incomplete (missing) features is computed trivially using (22) (with $\epsilon_i = 0$ since no actual labeling will have transpired).

3 Semisupervised Classification of Incomplete Data

3.1 Preliminaries

Semisupervised algorithms utilize both labeled and unlabeled data to build a classifier. Although many semisupervised algorithms exist (see [23] for a thorough literature review), no semisupervised algorithms have been proposed to handle the case of incomplete data. Here, we extend a graph-based approach [10] to obtain a semisupervised algorithm that handles incomplete data.

In addition to the labeled data set in (2), assume we have a set of unlabeled incomplete data,

$$\mathcal{D}_U = \{(\mathbf{x}_i, m_i) : \mathbf{x}_i \in \mathbb{R}^d, x_{ia} \quad \text{missing} \quad \forall a \in m_i\}_{i=N_L+1}^{N}. \tag{24}$$

A kernel function measures the similarity between two data points. Computing the kernel function for every pair of N data points (both labeled and unlabeled) results in the symmetric, positive semidefinite kernel matrix $\mathbf{K}$. The ijth element of the kernel matrix—K_{ij}—is a measure of similarity between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. With $\mathbf{D}$ the diagonal matrix whose iith element is given by $D_{ii} = \sum_{j=1}^{N} K_{ij}$, the (unnormalized) graph Laplacian is defined to be

$$\Delta' = \mathbf{D} - \mathbf{K}. \tag{25}$$

Theoretical work [12] has shown the necessity of normalizing the graph Laplacian, with one such acceptable normalization being

$$\Delta = \mathbf{D}^{-1/2}\Delta'\mathbf{D}^{-1/2}. \tag{26}$$

A fully connected, undirected graph with vertices $V = \{1, 2, \ldots, N\}$ can be summarized by the above kernel matrix $\mathbf{K}$ in the following manner [10]: By assigning one vertex of the graph to each data point, the edge of the graph joining vertices i and j can be represented by the weight K_{ij}. A natural way to measure how much a function $\mathbf{f} = [f_1, \ldots, f_N]^T$ defined on V varies across the graph is by the quantity

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_{ij}(f_i - f_j)^2 = \mathbf{f}^T\Delta'\mathbf{f}. \tag{27}$$

By defining a Gaussian random field (GRF) on the vertices V (using the *normalized* graph Laplacian $\boldsymbol{\Delta}$ instead of the unnormalized version $\boldsymbol{\Delta}'$),

$$p(\mathbf{f}) \propto \exp\{(-\lambda/2)\ \mathbf{f}^T\Delta\mathbf{f}\}, \tag{28}$$

smooth functions $\mathbf{f}$ are deemed more probable. In (28), λ is a positive regularization parameter. If we define $f_i = \mathbf{w}^T\mathbf{x}_i$, then $\mathbf{f} = [f_1, \ldots, f_N]^T = \mathbf{X}^T\mathbf{w}$, where the aith element of $\mathbf{X}$ corresponds to the ath feature of the ith data point. With this choice, $p(\mathbf{f})$ induces a Gaussian prior on $\mathbf{w}$,

$$\begin{aligned} p(\mathbf{f}) &= p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}\Big) \\ &\propto \exp\{(-\lambda/2)\mathbf{w}^T\mathbf{X}\Delta\mathbf{X}^T\mathbf{w}\} = \exp\{(-\lambda/2)\mathbf{w}^T\mathbf{G}\mathbf{w}\}, \end{aligned} \tag{29}$$

with the precision matrix $\mathbf{G} = \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T$. This formulation encourages "similar" data points to have similar class labels.

3.2 Derivation

Our proposed semisupervised algorithm will utilize the Gaussian prior formulation outlined in Section 3.1. To employ this formulation when faced with incomplete data, we will again analytically integrate out the missing data. In the derivation of the requisite integration, two approximations will be invoked. First, we will integrate out the missing data from the *log*-prior instead of the prior. Second, we will integrate out the missing data in a two-stage procedure, as will be shown in greater detail below. Developing this semisupervised method will allow unlabeled data to be exploited explicitly in learning the classifier.

The maximum a posteriori (MAP) classifier maximizes the posterior of $\mathbf{w}$, which is proportional to the product of the likelihood of the data and the prior of $\mathbf{w}$:

$$\begin{aligned} &p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}, \{y_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}\Big) \\ &\quad \propto p\Big(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}, \mathbf{w}\Big)p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}\Big). \end{aligned} \tag{30}$$

Ideally, the missing data would be integrated out from the posterior in (30):

$$\begin{aligned} &\int p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}, \{y_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}\Big)\left[\prod_{i=1}^{N} p\big(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}\big)\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &\propto \int p\Big(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}, \mathbf{w}\Big) \\ &\quad p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}\Big)\left[\prod_{i=1}^{N} p\big(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}\big)\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N}. \end{aligned} \tag{31}$$

Since this integral is, unfortunately, intractable, we appeal to Jensen's inequality, noting that the concavity of the logarithm function leads to a lower bound on the logarithm of (31):

$$\begin{aligned} &\log\int p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}, \{y_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}\Big) \\ &\quad \left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &\geq \int \log p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}, \{y_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}\Big) \\ &\quad \left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &\propto \int \log\Big\{p\Big(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}, \mathbf{w}\Big) \\ &\quad p(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N})\Big\}\left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &= \int \log p\Big(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i\}_{i=1}^{N_L}, \{\epsilon_i\}_{i=1}^{N_L}, \mathbf{w}\Big) \\ &\quad \left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &\quad + \int \log p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}\Big)\left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\ &= \ell(\mathbf{w}) + \int \log p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^{N}\Big)\left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N}. \end{aligned} \tag{32}$$

We therefore integrate out the missing data for the *log*-posterior. Since the expression for the log-likelihood $\ell(\mathbf{w})$ has already been obtained in (23), we direct our attention to integrating the log-prior (or, equivalently, $\mathbf{G}$; see, (29)) in (32).

If a normalized graph Laplacian is to be used in $\mathbf{G}$, as we desire, a closed-form expression cannot be obtained for this integral. Instead, we use a two-stage approach in computing

this integral.[1] It was shown in [22] that, when faced with missing data, the kernel matrix can be analytically completed by integrating out the missing data (for a Gaussian kernel). From this completed kernel matrix, the graph Laplacian can be readily computed using (25), and then normalized using (26), resulting in $\mathbf{\Delta}$. We follow this path, replacing the graph Laplacian within $\mathbf{G}$ with the analytically completed $\mathbf{\Delta}$, which is no longer a function of the missing data. Then, in the second stage, treating $\mathbf{\Delta}$ as a constant, the result of the requisite integration in (32) is

$$\begin{aligned}
&\log p\Big(\mathbf{w}|\{\mathbf{x}_i^{o_i}\}_{i=1}^N\Big) \\
&= \int \log p\Big(\mathbf{w}|\{\mathbf{x}_i\}_{i=1}^N\Big)\left[\prod_{i=1}^N p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\
&= (-\lambda/2)\int \mathbf{w}^T\mathbf{X}\Delta\mathbf{X}^T\mathbf{w}\left[\prod_{i=1}^N p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})\right] d\mathbf{x}_1^{m_1}\cdots d\mathbf{x}_N^{m_N} \\
&= (-\lambda/2)\mathbf{w}^T\Big(\widetilde{\mathbf{X}}\Delta\widetilde{\mathbf{X}}^T + \Phi\Big)\mathbf{w}.
\end{aligned} \tag{33}$$

The derivation of (33) is shown in Appendix B, which can be found at http://computer.org/tpami/archives.htm. In (33), the aith element of $\widetilde{\mathbf{X}}$ is

$$\widetilde{X}_{ai} = \begin{cases} x_{ia} & \text{if } a \in o_i \\ \sum_{k=1}^K \delta_k^i \xi_k^{m_i[a]} & \text{if } a \in m_i \end{cases} \tag{34}$$

and the abth element of $\mathbf{\Phi}$ is

$$\Phi_{ab} = \sum_{i=1}^N \Delta_{ii} \sum_{k=1}^K \delta_k^i \Omega_k^{m_i[ab]} \mathbf{1}_{a\in m_i}\mathbf{1}_{b\in m_i}, \tag{35}$$

with $\mathbf{1}_z$ an indicator function that is unity if z is true, but is zero otherwise. Note that $\xi_k^{m_i[a]}$ is the element in $\boldsymbol{\xi}_k^{m_i}$ that corresponds to feature a and $\Omega_k^{m_i[ab]}$ is the covariance element in $\mathbf{\Omega}_k^{m_i}$ that corresponds to features a and b.

The two-stage approach to the integration in (32) retains tractability while also limiting the propagation of errors due to missing data. By first analytically integrating out the missing data in the completion of the kernel matrix, we establish a very accurate relationship between every pair of data points. Because subsequent calculations depend on these pairwise relationships, errors in these quantities would compound and spread throughout $\mathbf{G}$.

Our proposed semisupervised classifier is then the (MAP-like[2]) classifier $\mathbf{w}$ that maximizes the sum of (23) and (33):

$$\begin{aligned}
\mathbf{w} = \arg\max_{\mathbf{w}} \Bigg\{ &\sum_{i=1}^{N_L} \log\left[\epsilon_i + (1-2\epsilon_i)\right. \\
&\left.\sum_{k=1}^K \delta_k^i\, \sigma\left(\frac{y_i\beta(\mathbf{w}_{m_i}^T\boldsymbol{\xi}_k^{m_i} + \mathbf{w}_{o_i}^T\mathbf{x}_i^{o_i})}{\sqrt{\mathbf{w}_{m_i}^T\mathbf{\Omega}_k^i\mathbf{w}_{m_i} + \beta^2}}\right)\right] \\
&+ (-\lambda/2)\mathbf{w}^T(\widetilde{\mathbf{X}}\Delta\widetilde{\mathbf{X}}^T + \Phi)\mathbf{w}\Bigg\}.
\end{aligned} \tag{36}$$

1. It has been our experience that the inelegance of the two-stage integration is worth the gains to be reaped from using a *normalized* graph Laplacian.

2. The $\mathbf{w}$ that maximizes the posterior in (30) may not be the same $\mathbf{w}$ that maximizes the log-posterior in (32) because of the integration.

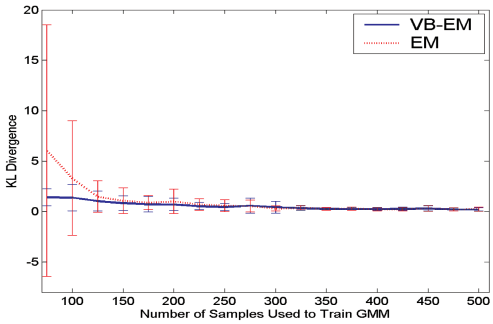


Fig. 2. Approximate KL divergence between the true GMM and the estimated GMMs using VB-EM and EM for the synthetic data set. Error bars represent one standard deviation about the mean value.

As in the supervised version, this $\mathbf{w}$ is found using gradient ascent. Evidence maximization [13] is used to select the value of λ; the procedure is shown in Appendix C, which can be found at http://computer.org/tpami/archives.htm.

4 Experimental Results

4.1 GMM Estimation

One of the main goals of this work is to develop a principled means of extending logistic regression to allow for the classification of incomplete data. Since the GMM density estimation plays a major role in the classification algorithm, an auxiliary goal is to compare the performance of the VB-EM and EM algorithms in estimating a GMM. To accomplish this secondary goal, we created a synthetic $2d$ data set, defined by a mixture of four Gaussians.

The true parameters of this GMM are as follows:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k,\Sigma_k), \tag{37}$$

$$\begin{aligned}
\boldsymbol{\pi} &= \begin{bmatrix} 1/3 & 1/6 & 1/4 & 1/4 \end{bmatrix}, \\
\boldsymbol{\mu}_1 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 4 & 3 \end{bmatrix}^T, \\
\boldsymbol{\mu}_3 &= \begin{bmatrix} 1/2 & 13/2 \end{bmatrix}^T, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} 6 & 4 \end{bmatrix}^T, \\
\Sigma_1 &= \begin{bmatrix} 1 & 3/4 \\ 3/4 & 1 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1 & -2/3 \\ -2/3 & 2/3 \end{bmatrix}, \\
\Sigma_3 &= \begin{bmatrix} 1 & 3/5 \\ 3/5 & 1 \end{bmatrix}, \quad \Sigma_4 = \begin{bmatrix} 1/8 & 1/4 \\ 1/4 & 1 \end{bmatrix}.
\end{aligned}$$

We randomly removed 40 percent of the features and then built GMMs using the VB-EM and EM algorithms. For each number of samples used to train the GMM, 50 trials were run. Each trial consisted of different data generated from the true GMM and different patterns of missing features.

An approximation to the Kullback-Leibler (KL) divergence between two Gaussian mixture models can be computed analytically using the unscented transform [6]. The smaller the KL divergence, the closer the estimated distribution is to the true distribution. The results of this experiment appear in Fig. 2. The difference between the VB-EM and EM algorithms is most pronounced when a small amount of data is available to build the GMMs, in which case, the VB-EM GMM is superior.

4.2 Classification

The area under a receiver operating characteristic (ROC) curve (AUC) is given by the Wilcoxon statistic [8]

$$\text{AUC} = (MN)^{-1} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{1}_{a_m > b_n}, \tag{38}$$

where $a_1, \ldots, a_M$ are the classifier decisions (e.g., the probabilities from (22)) of data belonging to class 1, $b_1, \ldots, b_N$ are the classifier decisions of data belonging to class -1, and $\mathbf{1}$ is an indicator function. We present the results of our classification algorithms in terms of the AUC.

We applied our proposed classification algorithms to the IONOSPHERE and WISCONSIN DIAGNOSTIC BREAST CANCER (WDBC) benchmark data sets from the UCI Machine Learning Repository. We also provide a comparison to multiple imputation for the data considered in Fig. 2 (see (37)). The IONOSPHERE data set has 351 data points and 34 features, while the WDBC data set has 569 data points and 30 features. In all experiments, missing features were artificially created in both training and testing data. Artificially creating missing data affords us the opportunity to observe algorithm performance as a function of various parameters (e.g., amount of missing data).

In the following experiments, every trial consists of a random partition of training and testing data and a random pattern of missing features, the amounts of which are determined by the given parameters. Because both the training sets as well as the patterns of missing features in every trial are unique, performance can vary widely between trials. The relative differences between two methods over all trials vary less. That is, the methods have a consistent relative difference in performance, even though the absolute difference in performance may vary widely from trial to trial. Therefore, for all experiments, in lieu of error bars, we report the results of paired t-tests between the proposed method and the other competing methods. All of these t-test results are shown in Appendix D, which can be found at http://computer.org/tpami/archives.htm.

4.3 Multiple Imputation

Using the same synthetic data set used in Section 4.1 (see (37)), we compared the proposed supervised method—from Section 2 that analytically integrates out missing data—with the method of multiple imputation [19]. Specifically, the $2d$ data set was composed of 200 data points, with 40 percent of the features randomly removed. Data points generated by one of the first two mixture components belong to class $y = 1$ and data points generated by the third or fourth mixture component belong to class $y = -1$. Ten percent of the data was used as training data, while the remaining 90 percent was used as testing data. We conducted 200 independent trials, where each trial consisted of a unique partition of the data into training and testing sets and a unique pattern of missing features. The VB-EM algorithm was used to estimate the (GMM) density function required by both methods.

For each trial, several different numbers of imputations were considered for the multiple imputation method. The process of classification with multiple imputation with M imputations proceeded as follows: First, the data set with missing features is replicated M times. For each of the M data sets, one sample is drawn from the estimated density function for each missing feature. These samples are inserted for the previously missing features, which produces complete data sets missing no features. For each of these M (artificially) complete data sets, a logistic regression classifier is learned. Each testing data point is then evaluated by each of the M classifiers (with any missing features of the testing data points first replaced by samples from the density function). The resulting M predictions (i.e., the probability of belonging to a given class) for each data point are then averaged. This procedure results in a single prediction (i.e., class probability) for each data point. Finally, the AUC is computed using these averaged predictions.

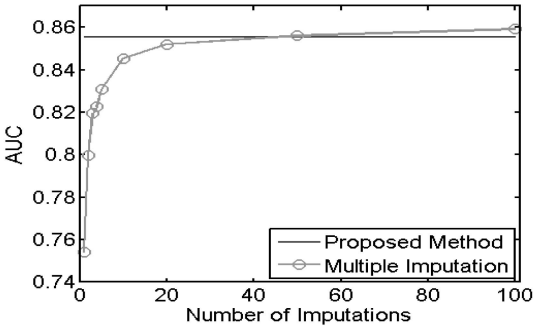


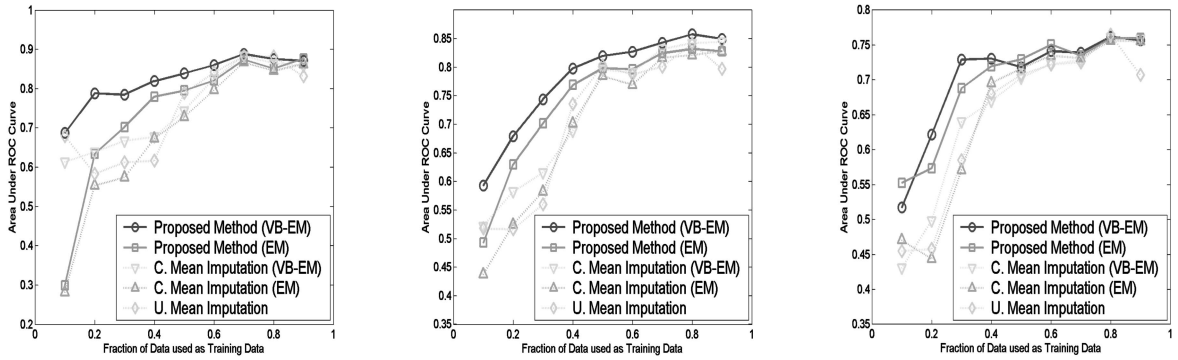
Fig. 3. Experimental results of the proposed supervised algorithm and the method of multiple imputation for the synthetic data set.

The results of this set of experiments are shown in Fig. 3. The paired t-test results are shown in Table 1 in Appendix D, which can be found at http://computer.org/tpami/archives.htm. As can be seen from Fig. 3, as the number of imputations increases, the performance of the multiple imputation method approaches the performance of the proposed algorithm. However, it should be noted that the computational cost of the multiple imputation method scales linearly as a function of the number of imputations (M). Whereas multiple imputation requires substantial sampling—as well as learning multiple classifiers—the proposed algorithm requires no sampling and must learn only a single classifier. With a sufficient number of imputations—what constitutes "sufficient" is unknown a priori in practice—and enough computational resources, multiple imputation will result in comparable performance to the proposed method. In subsequent experiments, we compare the proposed method to more computationally feasible methods that share similar levels of computational complexity.

4.3.1 Supervised Classification with Perfect Labels

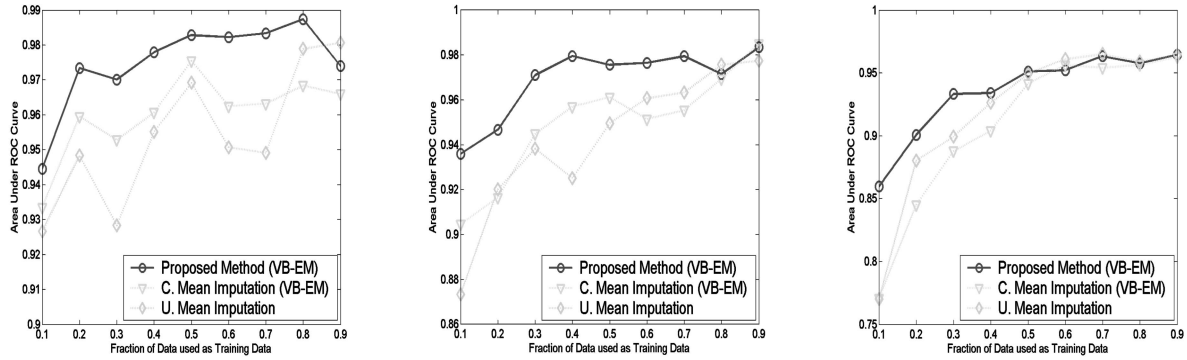
Experimental results for the supervised algorithm are shown in Figs. 4 and 5 for the IONOSPHERE and WDBC data sets, respectively. To allow one to observe the performance of the methods as a function of data-set size, the GMMs are trained using only training (labeled) data. In practice, all available data (labeled and unlabeled) can be used to build the GMMs because labels are not used in this density estimation.

Five different methods were compared for the experiments on the IONOSPHERE data set. Two methods use the proposed supervised algorithm; to estimate the GMM, one of these methods uses the VB-EM algorithm, while the other method uses the EM algorithm. Three mean imputation methods were also considered. These methods first "complete" all missing data using conditional mean imputation (utilizing the GMM estimated using VB-EM or EM) or



(a) (b) (c)

Fig. 4. Experimental results for the supervised algorithm on the IONOSPHERE data set. The proposed methods use the new logistic regression method (no imputation), with the requisite GMMs trained using the VB-EM or EM algorithm. The other three methods complete the missing data via imputation using the conditional mean (obtained via the VB-EM or EM GMMs) or the unconditional mean. The results are for the cases when (a) 25 percent, (b) 50 percent, and (c) 75 percent of the features are missing.



(a) (b) (c)

Fig. 5. Experimental results for the supervised algorithm on the WDBC data set. Refer to the caption of Fig. 4 for additional details. The results are for the cases when (a) 25 percent, (b) 50 percent, and (c) 75 percent of the features are missing.

unconditional mean imputation. Specifically, in conditional mean imputation, the missing features of each data point are replaced with their conditional mean:

$$\mathbf{x}_i^{m_i} \leftarrow \mathbb{E}[\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}] = \sum_{k=1}^{K} \delta_k^i \boldsymbol{\xi}_k^{m_i}, \tag{39}$$

where δ_k^i and $\boldsymbol{\xi}_k^{m_i}$ are defined in (9) and (12), respectively. In unconditional mean imputation, all missing data is "completed" with the unconditional mean, which does not require a model of the data. If $\mathbf{x}_i$ is missing feature a (i.e., $a \in m_i$), unconditional mean imputation will make the substitution

$$x_{ia} \leftarrow \mathbb{E}[x_{ia}] = \frac{\sum_{j=1}^{N} x_{ja} \mathbf{1}_{a \in o_j}}{\sum_{\ell=1}^{N} \mathbf{1}_{a \in o_\ell}}. \tag{40}$$

Standard (complete-data) logistic regression was then used for these three mean imputation methods.

Each point on every curve in Fig. 4 is an average over 10 trials. The paired t-test results are shown in Table 2 in Appendix D, which can be found at http://computer.org/tpami/archives.htm. From Fig. 4, it can be observed that the proposed method using VB-EM for the GMM estimation consistently performed better than the same method using EM for the GMM estimation. In particular, this difference was most significant when a small number of data points were available to train the GMM (see Fig. 2 also). We also observed that both of these versions of the proposed method were superior to the three single imputation schemes considered. For the proposed method using VB-EM, having fewer training data points with a higher fraction of features present appears to be more important (in terms of performance) than having more training data points with a lower fraction of features present (e.g., when the fraction of training data points is 0.2, 0.3, and 0.6 in Figs. 4a, 4b, and 4c, respectively, the training set has the same total number of present features).

Confident of the superiority of the VB-EM algorithm over the EM algorithm for the GMM estimation (see Figs. 2 and 4), all subsequent experiments use the VB-EM algorithm to estimate GMMs. Additional results—for the WDBC data set—shown in Fig. 5 were obtained by following the same experimental setup as that used to obtain the results for the IONOSPHERE data set in Fig. 4. The paired t-test results are shown in Table 3 in Appendix D, which can be found at http://computer.org/tpami/archives.htm. The proposed method again outperformed the mean imputation methods.

4.3.2 Supervised Classification with Imperfect Labels

The IONOSPHERE data set was also used to evaluate the proposed supervised algorithm with imperfect labels.

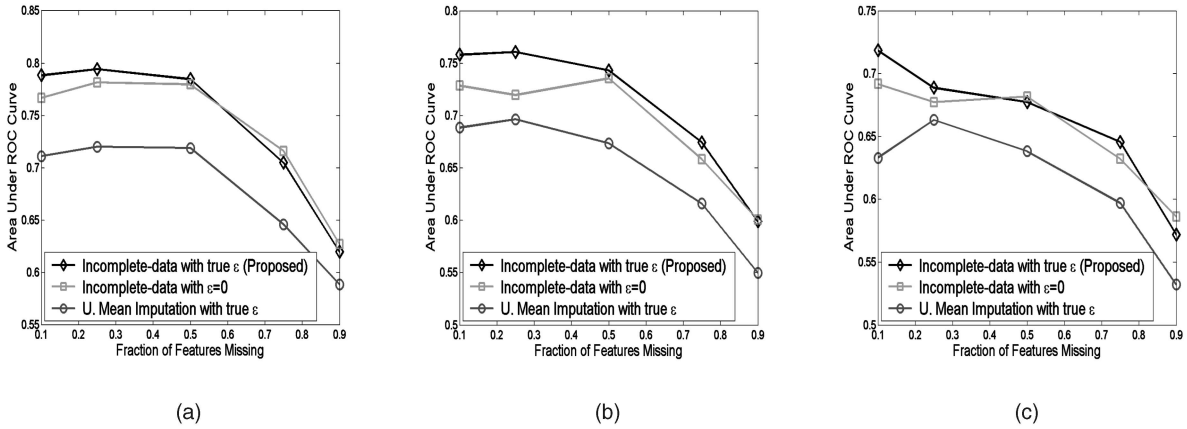


Fig. 6. Experimental results for the supervised algorithm with imperfect labels when the labeling error rate is (a) $\epsilon = 0.1$, (b) $\epsilon = 0.2$, and (c) $\epsilon = 0.3$.

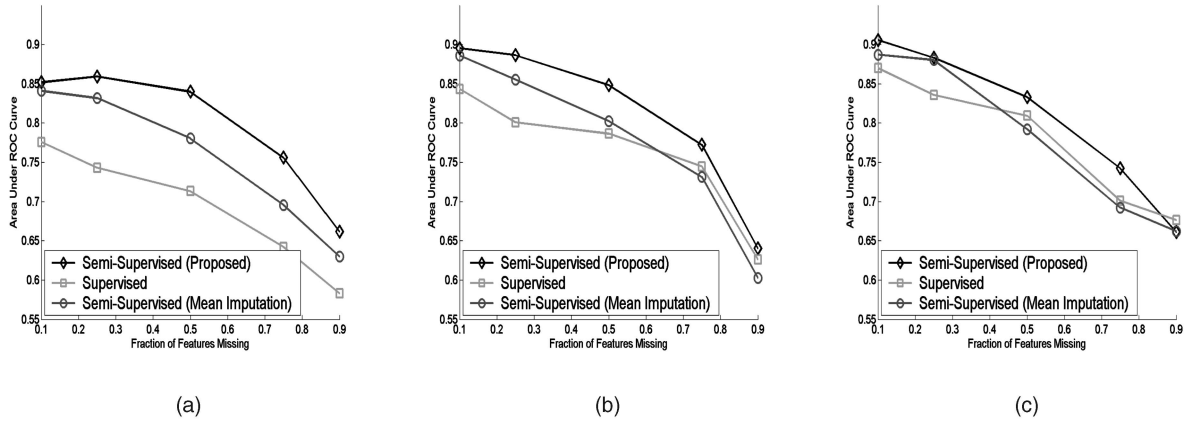


Fig. 7. Experimental results for the semisupervised algorithm. The results are for the cases when (a) 25 percent, (b) 50 percent, and (c) 75 percent of the data is labeled.

We compared the proposed supervised algorithm with imperfect labels to two other algorithms: 1) the same supervised algorithm except without the imperfect label capability (i.e., with $\epsilon = 0$ incorrectly) and 2) the supervised (logistic regression) algorithm with imperfect label capability, except all missing data is first "completed" with the unconditional mean values. The training data labels were randomly made incorrect at the given labeling error rate ϵ. The results of these experiments appear in Fig. 6. The paired t-test results are shown in Table 4 in Appendix D, which can be found at http://computer.org/tpami/archives.htm.

For this set of experiments, 50 percent of the data was labeled training data. Each point on every curve in Fig. 6 is an average over 15 trials. Every trial consists of a random partition of training and testing data and a random pattern of missing features. For each trial, all three methods considered use the same data partitions, missing data patterns, and corrupted training labels.

The proposed incomplete-data method using the true labeling error rate ϵ consistently achieves better performance than the method that incorrectly assumes perfect labeling (i.e., $\epsilon = 0$). This latter method using the wrong labeling error rate value still achieves better performance than unconditional mean imputation with the true ϵ. These results suggest that using the proposed algorithm with an inaccurate labeling error rate is still better than performing mean imputation. This result is particularly important because an accurate estimate of the labeling error rate may be difficult to obtain in practice.

4.3.3 Semisupervised Classification

The IONOSPHERE data set was again used to evaluate the proposed semisupervised algorithm. We compared the proposed semisupervised algorithm to two other algorithms: 1) the purely supervised version of the algorithm and 2) the semisupervised algorithm of the same form (i.e., logistic regression with a GRF prior), except all missing data is first "completed" with the unconditional mean values. The results of these experiments appear in Fig. 7. The paired t-test results are shown in Table 5 in Appendix D, which can be found at http://computer.org/tpami/archives.htm.

Each point on every curve in Fig. 7 is an average over 15 trials. Every trial consists of a random partition of training and testing data and a random pattern of missing features. For each trial, all three methods considered use the same data partitions and missing data patterns.

From Fig. 7, it is seen that the proposed semisupervised algorithm consistently outperforms both the supervised algorithm as well as the semisupervised mean imputation method. The advantage of the proposed semisupervised algorithm was most significant when there was limited labeled (training) data.

5 Discussion

The incomplete-data problem, and in particular our proposed approach using GMMs, raises several questions. For instance, the number of data points required to accurately estimate the

GMM will increase as the square of the feature dimension because the covariance matrix is modeled. In contrast, the number of parameters in the standard logistic regression is equal to the feature dimension. Despite this ostensibly increased data size requirement, our proposed algorithm using the VB-EM GMM still performs better than single imputation schemes when the number of training data points is small. For example, in Fig. 4, when the fraction of training data points is 0.1 (corresponding to only 35 training data points, each of which have 34 features), our proposed algorithm still outperforms the single imputation methods. This result suggests that the benefits of our algorithm outweigh the added parameter estimation burden. It must be noted, however, that the proposed approach is not feasible for data sets with many (e.g., thousands) of features, such as gene expression data sets [16]. Future work will focus on developing methods to handle such data sets.

Another question the incomplete-data problem raises is whether ignoring data with missing features is better than using an incomplete-data method (either our proposed method or even a simple imputation scheme). It is, of course, displeasing to discard data (information), but can doing so improve performance? There is a major problem with simply ignoring data with missing features. It is true that ignoring data with missing features in the training stage will eliminate incomplete-data training issues. However, in the testing stage, one cannot simply ignore a data point to be classified because it is missing some features. One would still be forced to resort to ad hoc procedures such as filling in zeros or the unconditional mean for the missing features of such incomplete testing data. In contrast, our principled proposed method does not rely on any ad hoc methods in either the training or testing stage.

Our proposed classification algorithm does, however, have some drawbacks. The semisupervised extension uses two approximations to retain tractability in integrating out the missing data: A two-stage approach was employed to perform the integration of the log-posterior (instead of the posterior). Despite the inelegance of this approach, the proposed semisupervised extension still achieves better performance than the purely supervised classifier. Moreover, it should be emphasized that our algorithm is the first semisupervised algorithm that handles incomplete data.

Perhaps the largest drawback of our general classification algorithm is the restriction to linear classifiers. The integration in (5) cannot be performed analytically as we have done if a nonlinear kernel function is used to first map data into a new feature space. If a typical kernel is used, all components for which data is missing would appear in each of the new features. In a sense, the kernel mapping would actually "create" more missing data. If it is imperative that a nonlinear classifier be used for a certain incomplete-data problem, we suggest instead using the analytical kernel matrix completion idea [22] that was used to build the graph Laplacian. Although this method "completes" all of the missing data, it does so in a principled manner. This approach has already been used successfully to classify incomplete data using a nonlinear classifier [22].

6 Conclusion

Our main contribution is the development of a logistic regression algorithm for the classification of incomplete data. By making two mild assumptions, the proposed supervised algorithm solves the incomplete-data problem in a principled manner, avoiding imputation heuristics.

We then extended this supervised algorithm to the semisupervised case in which all available data is utilized—both incomplete and complete, as well as labeled and unlabeled. Experimental results have shown the advantages of the various features of this algorithm. The proposed algorithm has also been successful even when a high percentage of features are missing. Moreover, despite the additional parameters to be estimated, the proposed algorithm has been successful when the training set size is small. In fact, the semisupervised extension improves performance most significantly in this very regime. Allowing for imperfect labels extends the theme of utilizing all available data to perform classification.

We have also derived the equations for building a GMM with incomplete data via the EM and VB-EM algorithms. Experimental evidence has shown that the VB-EM algorithm is markedly superior in terms of density estimation when data is scarce.

Several exciting directions exist for future research. One topic deserving of future study is the development of a principled algorithm that allows a nonlinear classifier to be used to classify incomplete data. Additional research will focus on extending the present algorithm both to handle the case of multinomial classification and to permit data sets with very large feature dimensions. Additional work will focus on establishing the relative (theoretical) value of incomplete data.

References

[1] M. Beal, "Variational Algorithms for Approximate Bayesian Inference," PhD thesis, Gatsby Computational Neuroscience Unit, Univ. College London, 2003.
[2] M. Beal and Z. Ghahramani, "The Variational Bayesian EM Algorithm for Incomplete Data: Application to Scoring Graphical Model Structures," *Bayesian Statistics,* vol. 7, pp. 453-464, 2003.
[3] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.
[4] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* Wiley, 2000.
[5] Z. Ghahramani and M. Jordan, "Supervised Learning from Incomplete Data via the EM Approach," *Proc. Advances in Neural Information Processing Systems (NIPS),* 1994.
[6] J. Goldberger, H. Greenspan, and S. Gordon, "An Efficient Similarity Measure Based on Approximations of KL-Divergence between Two Gaussian Mixtures," *Proc. Int'l Conf. Computer Vision,* 2003.
[7] T. Graepel, "Kernel Matrix Completion by Semidefinite Programming," *Proc. Int'l Conf. Artificial Neural Networks,* 2002.
[8] J. Hanley and B. McNeil, "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology,* vol. 143, pp. 29-36, 1982.
[9] J. Ibrahim, "Incomplete Data in Generalized Linear Models," *J. Am. Statistical Assoc.,* vol. 85, pp. 765-769, 1990.
[10] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo, "On Semi-Supervised Classification," *Proc. Advances in Neural Information Processing Systems (NIPS),* 2005.
[11] G. Lanckriet, M. Deng, N. Cristianini, M. Jordan, and W. Noble, "Kernel-Based Data Fusion and Its Application to Protein Function Prediction in Yeast," *Proc. Pacific Symp. Biocomputing 9,* pp. 300-311, 2004.

[12] U. Luxburg, O. Bousquet, and M. Belkin, "Limits of Spectral Clustering," *Proc. Advances in Neural Information Processing Systems (NIPS),* pp. 857-864, 2004.
[13] D. MacKay, "The Evidence Framework Applied to Classification Networks," *Neural Computation,* vol. 5, pp. 698-714, 1992.
[14] P. McCullagh and J. Nelder, *Generalized Linear Models,* second ed. Chapman & Hall, 1989.
[15] N. Nasios and A. Bors, "Variational Expectation-Maximization Training for Gaussian Networks," *Proc. IEEE Workshop Neural Networks for Signal Processing,* pp. 339-348, 2003.
[16] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii, "A Bayesian Missing Value Estimation Method," *Bioinformatics,* vol. 19, pp. 2088-2096, 2003.
[17] M. Opper and O. Winther, "Gaussian Processes and SVM: Mean Field and Leave-One-Out," *Advances in Large Margin Classifiers,* A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds., pp. 311-326, MIT Press, 2000.
[18] S. Rässler, "The Impact of Multiple Imputation for DACSEIS," Technical Report DACSEIS Research Paper Series 5, Univ. of Erlangen-Nürnberg, Nürnberg, Germany, 2004.
[19] D. Rubin, *Multiple Imputation for Nonresponse in Surveys.* Wiley, 1987.
[20] J. Schafer and J. Graham, "Missing Data: Our View of the State of the Art," *Psychological Methods,* vol. 7, no. 2, 2002.
[21] K. Tsuda, S. Akaho, and K. Asai, "The *em* Algorithm for Kernel Matrix Completion with Auxiliary Data," *J. Machine Learning Research,* vol. 4, pp. 67-81, 2003.
[22] D. Williams and L. Carin, "Analytical Kernel Matrix Completion with Incomplete Multi-View Data," *Proc. 22nd Int'l Conf. Machine Learning (ICML) Workshop Learning with Multiple Views,* pp. 80-86, 2005.
[23] X. Zhu, "Semi-Supervised Learning with Graphs," PhD thesis, Carnegie Mellon Univ., 2005.

David Williams received the BSE (magna cum laude), MS, and PhD degrees in 2002, 2003, and 2006, respectively, all from Duke University. While at Duke, he was the recipient of a James B. Duke Graduate Fellowship and a National Defense Science and Engineering Graduate Fellowship. His principal technical interests lie in the fields of machine learning and automatic target recognition. He is a member of the IEEE.

Xuejun Liao (SM '04) received the BS and MS degrees in electrical engineering from Hunan University, China, in 1990 and 1993, respectively, and the PhD degree in electrical engineering from Xidian University, China, in 1999. From 1993 to 1995, he was with the Department of Electrical Engineering, Hunan University, working on electronic instruments. From 1995 to 2000, he was with the National Key Lab for Radar Signal Processing, Xidian University, working on automatic target recognition (ATR) and radar imaging. Since May 2000, he has been working as a research associate with the Department of Electrical and Computer Engineering, Duke University. His current research interests are in planning under uncertainty, machine learning, bioinformatics, signal, and image processing. He is a senior member of the IEEE.

Ya Xue received the BS degree in electrical engineering in July 2000 from Tsinghua University, Beijing, China, and the MS degree in electrical engineering in December 2002 from Arizona State University, Tempe. She received the PhD degree from Duke University in electrical and computer engineering in December 2006. Her interests are machine learning and nonparametric statistical techniques.

Lawrence Carin (F '01-SM '96) received the BS, MS, and PhD degrees in electrical engineering from the University of Maryland, College Park, in 1985, 1986, and 1989, respectively. In 1989, he joined the Electrical Engineering Department at Polytechnic University (Brooklyn) as an assistant professor, and became an associate professor there in 1994. In September 1995, he joined the Electrical and Computer Engineering Department at Duke University, where he is now the William H. Younger Distinguished Professor. Dr. Carin has been the principal investigator on several large research programs, including two Multidisciplinary University Research Initiative (MURI) programs. He is the cofounder of the small business Signal Innovations Group (SIG), which was purchased in 2006 by Integrian, Inc. He was an associate editor of the *IEEE Transactions on Antennas and Propagation* from 1996-2001. His current research interests include signal processing and machine learning for sensing applications. He is a member of the Tau Beta Pi and Eta Kappa Nu honor societies. He is a fellow of the IEEE.

Balaji Krishnapuram received the BTech degree from the Indian Institute of Technology (ITT) Kharagpur in 1999 and the PhD degree from Duke University in 2004, both in electrical engineering. He works as a scientist with Siemens Medical Solutions in Malvern, Pennsylvania. His research interests include statistical pattern recognition, Bayesian inference, and computational learning theory. He is also interested in applications in computer-aided medical diagnosis, signal processing, computer vision, and bioinformatics.