On Classification with Incomplete Data

David Williams, Xuejun Liao, Ya Xue, and Lawrence Carin

Department of Electrical and Computer Engineering

Duke University

Durham, NC 27708

USA

{dpw,xjliao,yx10,lcarin}@ee.duke.edu

Balaji Krishnapuram

Siemens Medical Solutions USA

Malvern, PA 19355

USA

balaji.krishnapuram@siemens.com

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 semi-supervised case by incorporating graph-based regularization. The semi-supervised algorithm utilizes all available data — both incomplete and complete, as well as labeled and unlabeled. Experimental results of the proposed classification algorithms are shown.

I. 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, $p\left(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}\right)$). 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 IV). 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 semi-supervised 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]:

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

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 semi-supervised 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 semi-supervised algorithms exist for incomplete-data classification.

The remainder of the paper is organized as follows. In Section II we derive the supervised logistic regression algorithm for classification of incomplete data, and in Section III we extend this supervised algorithm to the semi-supervised case. Experimental results for the classification algorithms are shown in Section IV, followed by a discussion in Section V. Concluding remarks and suggestions for future work are made in Section VI.

II. 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 i-th 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 multi-sensor 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 $p\left(y_i|\mathbf{x}_i, \mathbf{w}\right) = \sigma(y_i \mathbf{w}^T \mathbf{x}_i)$, 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]

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

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

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

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

$$\begin{aligned} p\left(y_i|\mathbf{x}_i^{o_i}, \epsilon_i, \mathbf{w}\right) &= \int p\left(y_i|\mathbf{x}_i^{m_i}, \mathbf{x}_i^{o_i}, \epsilon_i, \mathbf{w}\right) p\left(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}\right) d\mathbf{x}_i^{m_i} & (5) \\ &= \epsilon_i + (1 - 2\epsilon_i) \int \sigma(y_i(\mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i} + \nu_i)) p\left(\nu_i|\mathbf{x}_i^{o_i}\right) d\nu_i. & (6) \end{aligned}$$

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

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

where the π_k are the non-negative mixture weights that sum to unity; necessarily $p\left(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}\right)$ 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.

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

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

with the parameters

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

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

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

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

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

where $\mathcal{G}(\nu_i) = (2\pi)^{-1/2} \exp\left\{-\nu_i^2/2\right\}$ 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 Figure 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.)

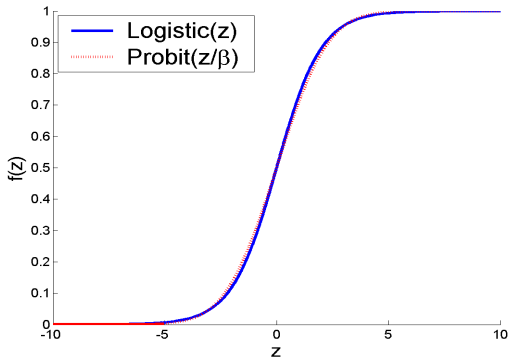


Fig. 1. Illustration of the accuracy of the approximation made between the logistic function and the (scaled) probit function.

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

$$
\begin{aligned}
& p\left(y_i|\mathbf{x}_i^{o_i}, \epsilon_i, \mathbf{w}\right) \\
&= \epsilon_i + (1-2\epsilon_i) \int\int_{-\infty}^{y_i(\mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i} + \nu_i)} \mathcal{G}\left(\frac{z}{\beta}\right) dz \sum_{k=1}^{K} \delta_k^i \mathcal{G}\left(\frac{\nu_i - \zeta_k^i}{\alpha_k^i}\right) d\nu_i && (15) \\
&= \epsilon_i + (1-2\epsilon_i) \int\int_{-\infty}^{y_i \mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z' + y_i\nu_i}{\beta}\right) dz' \sum_{k=1}^{K} \delta_k^i \mathcal{G}\left(\frac{\nu_i - \zeta_k^i}{\alpha_k^i}\right) d\nu_i && (16) \\
&= \epsilon_i + (1-2\epsilon_i) \sum_{k=1}^{K} \delta_k^i \int_{-\infty}^{y_i \mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i}} \int \mathcal{G}\left(\frac{z' + y_i\nu_i}{\beta}\right) \mathcal{G}\left(\frac{y_i\nu_i - y_i\zeta_k^i}{y_i\alpha_k^i}\right) d\nu_i \, dz' && (17) \\
&= \epsilon_i + (1-2\epsilon_i) \sum_{k=1}^{K} \delta_k^i \int_{-\infty}^{y_i \mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z' + y_i\zeta_k^i}{\sqrt{(y_i\alpha_k^i)^2 + \beta^2}}\right) dz' && (18) \\
&= \epsilon_i + (1-2\epsilon_i) \sum_{k=1}^{K} \delta_k^i \int_{-\infty}^{y_i \mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i}} \mathcal{G}\left(\frac{z' + y_i\zeta_k^i}{\beta} \frac{\beta}{\sqrt{(\alpha_k^i)^2 + \beta^2}}\right) dz' && (19) \\
&= \epsilon_i + (1-2\epsilon_i) \sum_{k=1}^{K} \delta_k^i \int_{-\infty}^{\frac{y_i\beta(\mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i} + \zeta_k^i)}{\sqrt{(\alpha_k^i)^2 + \beta^2}}} \mathcal{G}\left(\frac{z}{\beta}\right) dz && (20) \\
&= \epsilon_i + (1-2\epsilon_i) \sum_{k=1}^{K} \delta_k^i \, \sigma\left(\frac{y_i\beta(\zeta_k^i + \mathbf{w}_{o_i}^T \mathbf{x}_i^{o_i})}{\sqrt{(\alpha_k^i)^2 + \beta^2}}\right). && (21)
\end{aligned}
$$

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 $y_i^2 = 1$; (20) results from the change of variable $z = \frac{\beta(z' + y_i \zeta_k^i)}{\sqrt{(\alpha_k^i)^2 + \beta^2}}$; and (21) is obtained by reverting to sigmoid representation. Thus we have expressed $p\left(y_i | \mathbf{x}_i^{o_i}, \epsilon_i, \mathbf{w}\right)$ as a mixture of *sigmoids*.

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

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

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

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

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).

III. Semi-Supervised Classification of Incomplete Data

A. Preliminaries

Semi-supervised algorithms utilize both labeled and unlabeled data to build a classifier. Although many semi-supervised algorithms exist (see [23] for a thorough literature review), no semi-supervised algorithms have been proposed to handle the case of incomplete data. Here we extend a graph-based approach [10] to obtain a semi-supervised 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 ij-th 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 ii-th element is given by $D_{ii} = \sum_{j=1}^{N} K_{ij}$, the (unnormalized) graph Laplacian is defined to be

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

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

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

A fully connected, undirected graph with vertices $V = \{1, 2, ..., N\}$ 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 $\mathbf{f} = [f_1, ..., f_N]^T$ defined on V varies across the graph is by the quantity

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

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

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

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 $\mathbf{f} = [f_1, ..., f_N]^T = \mathbf{X}^T\mathbf{w}$, where the ai-th element of $\mathbf{X}$ corresponds to the a-th feature of the i-th data point. With this choice, $p(\mathbf{f})$ induces a Gaussian prior on $\mathbf{w}$,

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

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

B. Derivation

Our proposed semi-supervised algorithm will utilize the Gaussian prior formulation outlined in Section III-A. 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 semi-supervised 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 $\mathbf{w}$:

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

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

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

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):

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

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, $\mathbf{G}$; *cf.* (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

[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.

in the second stage, treating $\mathbf{\Delta}$ as a constant, the result of the requisite integration in (32) is

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

The derivation of (33) is shown in Appendix B. In (33), the ai-th element of $\widetilde{\mathbf{X}}$ is

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

and the ab-th element of $\mathbf{\Phi}$ is

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

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

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

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.

IV. Experimental Results

A. 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 2-*d* data set, defined by a mixture of four Gaussians.

The true parameters of this GMM are as follows:

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

$$\boldsymbol{\pi} = \begin{bmatrix} 1/3 & 1/6 & 1/4 & 1/4 \end{bmatrix}$$

[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.

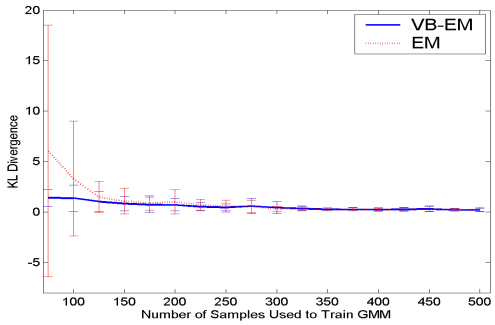


Fig. 2. Approximate KL divergence between the true GMM and the estimated GMMs using VB-EM and EM for the synthetic data set. Error bars represent one standard deviation about the mean value.

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \ \boldsymbol{\mu}_2 = \begin{bmatrix} 4 & 3 \end{bmatrix}^T, \ \boldsymbol{\mu}_3 = \begin{bmatrix} 1/2 & 13/2 \end{bmatrix}^T, \ \boldsymbol{\mu}_4 = \begin{bmatrix} 6 & 4 \end{bmatrix}^T$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 3/4 \\ 3/4 & 1 \end{bmatrix}, \ \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & -2/3 \\ -2/3 & 2/3 \end{bmatrix}, \ \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 3/5 \\ 3/5 & 1 \end{bmatrix}, \ \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1/8 & 1/4 \\ 1/4 & 1 \end{bmatrix}.$$

We randomly removed 40% of the features, and then built GMMs using the VB-EM and EM algorithms. For each number of samples used to train the GMM, fifty 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 Figure 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.

B. 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 Figure 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.

C. Multiple Imputation

Using the same synthetic data set used in Section IV-A (see (37)), we compared the proposed supervised method — from Section II that analytically integrates out missing data — with the method of multiple imputation [19]. Specifically, the 2-*d* data set was composed of 200 data points, with 40% 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 ninety 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)

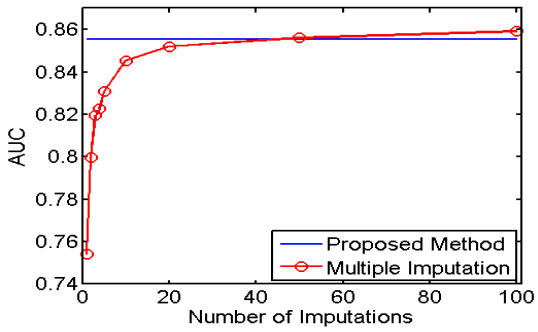


Fig. 3. Experimental results of the proposed supervised algorithm and the method of multiple imputation for the synthetic data set.

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.

The results of this set of experiments are shown in Figure 3. The paired t-test results are shown in Table I in Appendix D. As can be seen from Figure 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.

1) Supervised Classification with Perfect Labels: Experimental results for the supervised algorithm are shown in Figures 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 unconditional mean imputation. Specifically, in conditional mean imputation, the missing features of each data point are replaced with their conditional mean:

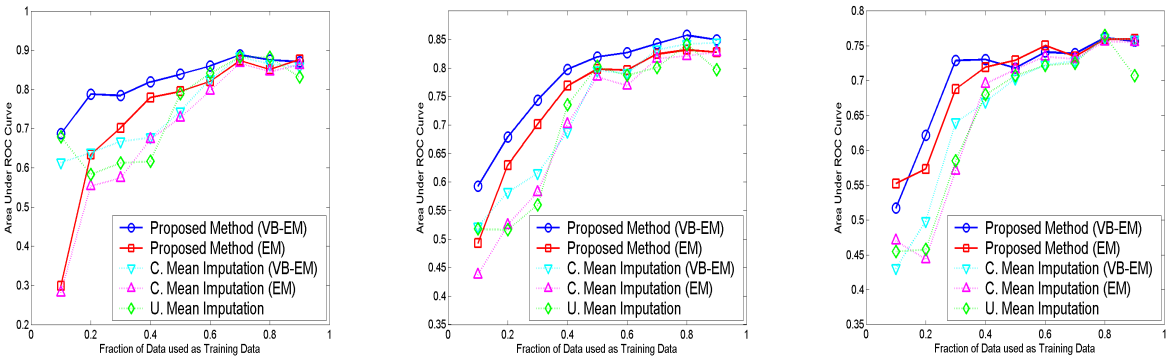
$$\mathbf{x}_i^{m_i} \leftarrow \mathbb{E}[\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}] = \sum_{k=1}^{K} \delta_k^i \boldsymbol{\xi}_k^{m_i}, \tag{39}$$

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 Figure 4 is an average over ten trials. The paired *t*-test results are shown in Table II in Appendix D. From Figure 4, it can be observed that the proposed method using VB-EM for the GMM estimation consistently performed better than the same

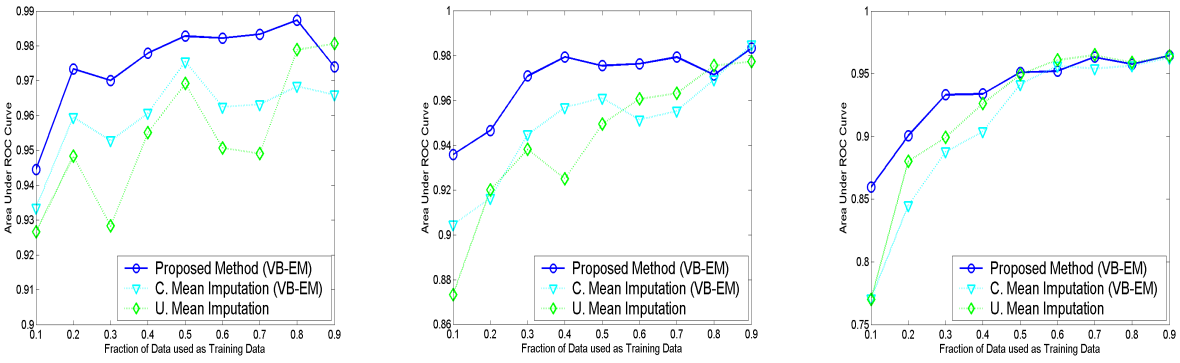


(a) (b) (c)

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%, (b) 50%, and (c) 75% of the features are missing.

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 (*cf.* Figure 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 Figures 4(a), 4(b), and 4(c), 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 (*cf.* Figures 2 and 4), all subsequent experiments use the VB-EM algorithm to estimate GMMs. Additional results — for the WDBC data set — shown in Figure 5 were obtained by following the same experimental setup as that used to obtain the results for the IONOSPHERE data set in Figure 4. The paired t-test results are shown in Table III in Appendix D. The proposed



(a) (b) (c)

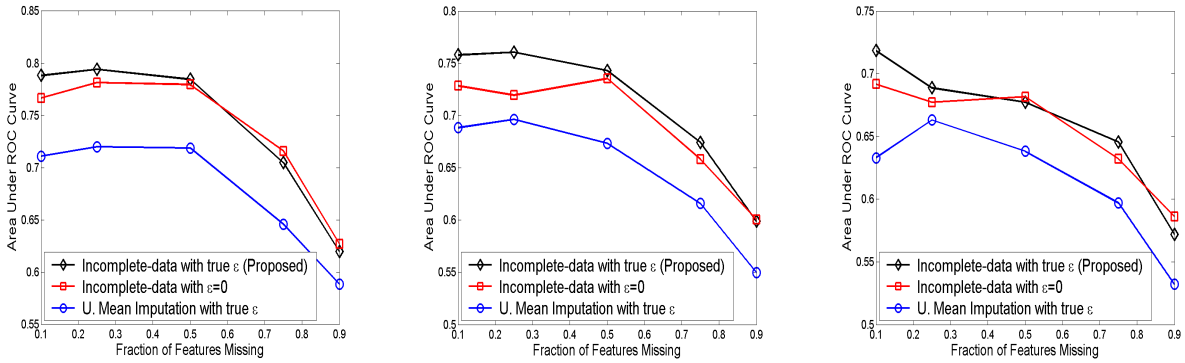
Fig. 5. Experimental results for the supervised algorithm on the WDBC data set. Refer to the caption of Figure 4 for additional details. The results are for the cases when (a) 25%, (b) 50%, and (c) 75% of the features are missing.

method again outperformed the mean imputation methods.

2) Supervised Classification with Imperfect Labels: The IONOSPHERE data set was also used to evaluate the proposed supervised algorithm with imperfect labels.

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 Figure 6. The paired t-test results are shown in Table IV in Appendix D.

For this set of experiments, 50% of the data was labeled training data. Each point on every curve in Figure 6 is an average over fifteen 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.



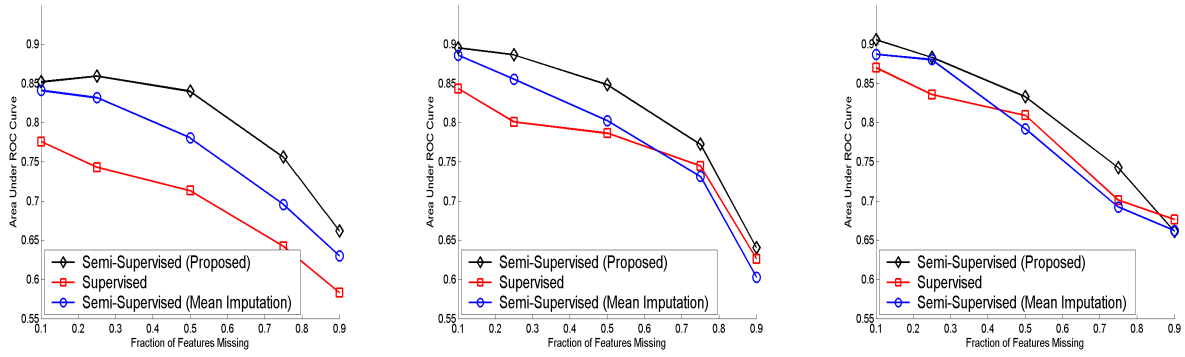
(a) (b) (c)

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) $\epsilon = 0.3$.

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.

3) Semi-Supervised Classification: The IONOSPHERE data set was again used to evaluate the proposed semi-supervised algorithm. We compared the proposed semi-supervised algorithm to two other algorithms: (1) the purely supervised version of the algorithm; and (2) the semi-supervised 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 Figure 7. The paired *t*-test results are shown in Table V in Appendix D.

Each point on every curve in Figure 7 is an average over fifteen trials. Every trial consists of



(a) (b) (c)

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

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

V. 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 Figure 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 semi-supervised 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 semi-supervised extension still achieves better performance than the purely supervised classifier. Moreover, it should be

emphasized that our algorithm is the first semi-supervised 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 non-linear 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 non-linear 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 non-linear classifier [22].

VI. 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 semi-supervised 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 semi-supervised 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 non-linear 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, University 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*, 7:453–464, 2003.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2000.

[5] Z. Ghahramani and M. Jordan. Supervised learning from incomplete data via the EM approach. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1994.

[6] J. Goldberger, H. Greenspan, and S. Gordon. An efficient similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In *Proceedings of the International Conference on Computer Vision*, 2003.

[7] T. Graepel. Kernel matrix completion by semidefinite programming. In *Proceedings of the International Conference on 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*, 143:29–36, 1982.

[9] J. Ibrahim. Incomplete data in generalized linear models. *Journal of the American Statistical Association*, 85:765–769, 1990.

[10] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo. On semi-supervised classification. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 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. In *Proceedings of the Pacific Symposium on Biocomputing 9*, pages 300–311, 2004.

[12] U. Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 857–864. MIT Press, 2004.

[13] D. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 5:698–714, 1992.

[14] P. McCullagh and J. Nelder. *Generalized Linear Models, 2nd Edition*. Chapman & Hall, 1989.

[15] N. Nasios and A. Bors. Variational expectation-maximization training for Gaussian networks. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 339–348, 2003.

[16] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A Bayesian missing value estimation method. *Bioinformatics*, 19:2088–2096, 2003.

[17] M. Opper and O. Winther. Gaussian processes and SVM: Mean field and leave-one-out. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 311–326. MIT Press, 2000.

[18] S. Rässler. The impact of multiple imputation for DACSEIS. Technical Report DACSEIS Research Paper Series 5, University 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*, 7(2), 2002.

[21] K. Tsuda, S. Akaho, and K. Asai. The *em* algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.

[22] D. Williams and L. Carin. Analytical kernel matrix completion with incomplete multi-view data. In *Proceedings of the 22nd International Conference Machine Learning (ICML) Workshop on Learning with Multiple Views*, pages 80–86, 2005.

[23] X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005.

APPENDIX A

We derive two methods for accurately estimating a GMM from incomplete data: the Expectation-Maximization (EM) algorithm [3] and the Variational Bayesian EM (VB-EM) algorithm [2], [1]. Whereas the result of the EM algorithm will be point estimates for the parameters, the result of the VB-EM algorithm will be *distributions* of the parameters. This fact allows the GMM learned via the VB-EM algorithm to be accurate even when faced with limited data.

By employing the VB-EM algorithm, one can also determine the appropriate number of GMM components, K, in a principled manner. One can consider several different values for K, and for each learn a GMM and compute the evidence [2] (which is an intermediate product of the VB-EM algorithm). The value for K that produces the maximum evidence is then chosen to be the true value of K. The variational formulation automatically penalizes overly complex models, whereas the standard EM algorithm will always favor increasingly complex models (*i.e.,* larger values of K).

Assume we have a set of N vectors (data), where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th vector. 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, and let $o_i = \{1, \cdots, d\} \setminus m_i$ be the (complementary) set of observed features in $\mathbf{x}_i$. Let $\gamma_i = k$ denote that $\mathbf{x}_i$ is generated by the k-th Gaussian of the GMM. In our GMM problem with incomplete data, $\{\mathbf{x}_i^{o_i}\}_{i=1}^N$ is the set of observed variables, $\Phi = \{\mathbf{x}_i^{m_i}, \gamma_i\}_{i=1}^N$ is the set of hidden variables, and $\Theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ is the set of parameters.

Since the $\mathbf{x}_i$ are independent, the complete-data log-likelihood function for $\{\mathbf{x}_i\}_{i=1}^N$ is

$$\begin{aligned}\ell(\Theta) &= \log \prod_{i=1}^{N} p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}, \gamma_i = k|\Theta\right) \\ &= \sum_{i=1}^{N} \log p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}, \gamma_i = k|\Theta\right) \\ &= \sum_{i=1}^{N} \log \left[p\left(\gamma_i = k|\Theta\right) p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}|\gamma_i = k, \Theta\right)\right] \end{aligned} \tag{41}$$

where $p\left(\gamma_i = k|\Theta\right) = \pi_k$ and

$$\begin{aligned} & p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}|\gamma_i = k, \Theta\right) = \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ & = (2\pi)^{-d/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left\{ -\frac{1}{2} \begin{bmatrix} \mathbf{x}_i^{o_i} - \boldsymbol{\mu}_k^{o_i} \\ \mathbf{x}_i^{m_i} - \boldsymbol{\mu}_k^{m_i} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Sigma}_k^{o_i o_i} & \boldsymbol{\Sigma}_k^{o_i m_i} \\ \boldsymbol{\Sigma}_k^{m_i o_i} & \boldsymbol{\Sigma}_k^{m_i m_i} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_i^{o_i} - \boldsymbol{\mu}_k^{o_i} \\ \mathbf{x}_i^{m_i} - \boldsymbol{\mu}_k^{m_i} \end{bmatrix} \right\}. \end{aligned}$$

The subsequent GMM estimation will utilize (41). It should be noted that both labeled and unlabeled data can be used in the GMM estimation because labels (y_i) are not used.

Expectation-Maximization (EM) Algorithm for GMM Estimation

We employ the EM algorithm [3] to maximize the objective function in (41). The EM algorithm is guaranteed to converge to a (local) maximum by alternating the two steps:

$$\text{Expectation (E) step:} \quad Q\left(\Theta|\widehat{\Theta}\right) = \mathbb{E}_{\Phi}\left[\ell(\Theta)|\{\mathbf{x}_i^{o_i}\}_{i=1}^N, \widehat{\Theta}\right] \tag{42}$$

$$\text{Maximization (M) step:} \quad \Theta = \arg\max_{\Theta} Q\left(\Theta|\widehat{\Theta}\right) \tag{43}$$

where $\widehat{\Theta}$ denotes the set of parameters from the previous iteration. The maximum log-likelihood parameters Θ at convergence will then constitute the GMM.

The algorithm for estimating a GMM from incomplete data via the EM algorithm has been given in [5]. However, that derivation admittedly assumed equal priors for the Gaussians. Here we address the general case and explicitly show the update equations. Most details of the derivation are omitted here because the spirit of the derivation is similar to that in [5].

Taking the expectation of the objective function with respect to the hidden variables results in the E-step:

$$\begin{aligned} Q\left(\Theta|\widehat{\Theta}\right) &= \mathbb{E}_{\Phi}\left[\ell(\Theta)|\{\mathbf{x}_i^{o_i}\}_{i=1}^N, \widehat{\Theta}\right] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\int d\mathbf{x}_i^{m_i}\, p\left(\gamma_i = k|\mathbf{x}_i^{o_i}, \widehat{\Theta}\right) p\left(\mathbf{x}_i^{m_i}|\gamma_i = k, \mathbf{x}_i^{o_i}, \widehat{\Theta}\right) \\ &\quad \times \log p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}, \gamma_i = k|\Theta\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_k^i\left\{\log\pi_k - \log\mathcal{N}(\widetilde{\mathbf{x}}_i^k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\widetilde{\boldsymbol{\Omega}}_i^k\right)\right\} \end{aligned} \tag{44}$$

where

$$\widehat{\boldsymbol{\xi}}_k^{m_i} = \widehat{\boldsymbol{\mu}}_k^{m_i} + \widehat{\boldsymbol{\Sigma}}_k^{m_i o_i}\left(\widehat{\boldsymbol{\Sigma}}_k^{o_i o_i}\right)^{-1}(\mathbf{x}_i^{o_i} - \widehat{\boldsymbol{\mu}}_k^{o_i}) \tag{45}$$

$$\widehat{\boldsymbol{\Omega}}_k^{m_i} = \widehat{\boldsymbol{\Sigma}}_k^{m_i m_i} - \widehat{\boldsymbol{\Sigma}}_k^{m_i o_i}\left(\widehat{\boldsymbol{\Sigma}}_k^{o_i o_i}\right)^{-1}\left(\widehat{\boldsymbol{\Sigma}}_k^{m_i o_i}\right)^T \tag{46}$$

$$\delta_k^i = p\left(\gamma_i = k|\mathbf{x}_i^{o_i}, \widehat{\Theta}\right) = \frac{\widehat{\pi}_k\mathcal{N}\left(\mathbf{x}_i^{o_i}; \widehat{\boldsymbol{\mu}}_k^{o_i}, \widehat{\boldsymbol{\Sigma}}_k^{o_i,o_i}\right)}{\sum_{\ell=1}^{K}\widehat{\pi}_\ell\mathcal{N}\left(\mathbf{x}_i^{o_i}; \widehat{\boldsymbol{\mu}}_\ell^{o_i}, \widehat{\boldsymbol{\Sigma}}_\ell^{o_i,o_i}\right)} \tag{47}$$

$$\widetilde{\mathbf{x}}_i^k = \begin{bmatrix} \mathbf{x}_i^{o_i} \\ \widehat{\boldsymbol{\xi}}_k^{m_i} \end{bmatrix} \qquad \text{and} \qquad \widetilde{\boldsymbol{\Omega}}_i^k = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widehat{\boldsymbol{\Omega}}_k^{m_i} \end{bmatrix}. \tag{48}$$

The results of the M-step are the three update equations:

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} \delta_k^i \widetilde{\mathbf{x}}_i^k}{\sum_{j=1}^{N} \delta_k^j} \tag{49}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{N} \delta_k^i \left\{ \left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right) \left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right)^T + \widetilde{\boldsymbol{\Omega}}_i^k \right\}}{\sum_{j=1}^{N} \delta_k^j} \tag{50}$$

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \delta_k^i. \tag{51}$$

Variational Bayesian EM (VB-EM) Algorithm for GMM Estimation

The accuracy of the GMM estimation using the EM algorithm will decrease as the amount of data available to make the estimation decreases. To combat this performance degradation, we derive the Variational Bayesian EM (VB-EM) algorithm [2], [1] for estimating a GMM from incomplete data.

Variational methods provide a lower bound on the log marginal likelihood. The log marginal likelihood of $\mathbf{x}_i^{o_i}$ can be lower bounded by writing [2], [1]

$$\begin{aligned}
\log p(\mathbf{x}_i^{o_i}) &= \log \int p(\mathbf{x}_i^{o_i}, \Phi, \Theta) d\Phi d\Theta \\
&= \log \int q(\Phi, \Theta) \frac{p(\mathbf{x}_i^{o_i}, \Phi, \Theta)}{q(\Phi, \Theta)} d\Phi d\Theta \\
&\geq \int q(\Phi, \Theta) \log \frac{p(\mathbf{x}_i^{o_i}, \Phi, \Theta)}{q(\Phi, \Theta)} d\Phi d\Theta \qquad (52) \\
&\approx \int q(\Phi) q(\Theta) \log \frac{p(\mathbf{x}_i^{o_i}, \Phi, \Theta)}{q(\Phi) q(\Theta)} d\Phi d\Theta \qquad (53)
\end{aligned}$$

where (52) follows from Jensen's inequality, and (53) is the result of making the factorized approximation $q(\Phi, \Theta) \approx q(\Phi) q(\Theta)$. The Variational Bayesian EM (VB-EM) algorithm maxi-

mizes (53) with respect to the distributions $q(\Phi)$ and $q(\Theta)$. Since these two distributions are coupled, functional derivatives with respect to each distribution are iteratively taken while the opposite distribution is held fixed. The resulting Variational Bayesian Expectation (VB-E) and Maximization (VB-M) steps are [2]

$$\text{VB-E step:} \quad q(\Phi) \propto \exp\left\{\int \log p(\mathbf{x}_i^{o_i}, \Phi|\Theta) q(\Theta) d\Theta\right\} \tag{54}$$

$$\text{VB-M step:} \quad q(\Theta) \propto p(\Theta) \exp\left\{\int \log p(\mathbf{x}_i^{o_i}, \Phi|\Theta) q(\Phi) d\Phi\right\}, \tag{55}$$

respectively. The algorithm for estimating a GMM from complete data using the VB-EM algorithm has been done previously (*e.g.,* [15]), but the *incomplete*-data version has not. Here we derive the algorithm for the incomplete-data case. In the following derivation, we write expectations as $\mathbb{E}_{\boldsymbol{\alpha}}[f(\boldsymbol{\alpha})] = \langle f(\boldsymbol{\alpha}) \rangle_{\boldsymbol{\alpha}}$.

A conjugate prior is a prior distribution that when combined with a particular likelihood will cause the posterior distribution to be of the same form as the prior. For the GMM, we choose conjugate-exponential priors for tractability [15]; that is, we choose a Dirichlet distribution on the mixing coefficients ($\boldsymbol{\pi}$), normal distributions on the means ($\boldsymbol{\mu}_k$), and Wishart distributions on the precisions (inverse covariances, $\boldsymbol{\Sigma}_k^{-1}$). The prior distribution of the GMM parameters is therefore

$$p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) p(\boldsymbol{\Sigma}_k) \tag{56}$$

where

$$p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}) = Z_{\boldsymbol{\pi}}^{-1} \prod_{k=1}^{K} \pi_k^{\lambda_k^0 - 1} \tag{57}$$

$$p\left(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k\right) = \mathcal{N}\left(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k\right) = Z_{\boldsymbol{\mu}_k}^{-1} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_k - \mathbf{m}_k^0\right)^T \beta_k^{0,-1} \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{\mu}_k - \mathbf{m}_k^0\right)\right\} \tag{58}$$

$$p\left(\boldsymbol{\Sigma}_k\right) = \mathcal{W}\left(\boldsymbol{\Sigma}_k^{-1}\right) = Z_{\boldsymbol{\Sigma}_k}^{-1} \left|\boldsymbol{\Sigma}_k^{-1}\right|^{\left(\alpha_k^0 - d - 1\right)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}_k^{0,-1}\boldsymbol{\Sigma}_k^{-1}\right)\right\} \tag{59}$$

with normalization constants

$$Z_{\boldsymbol{\pi}} = \frac{\prod_{k=1}^K \Gamma\left(\lambda_k^0\right)}{\Gamma\left(\sum_{k=1}^K \lambda_k^0\right)} \tag{60}$$

$$Z_{\boldsymbol{\mu}_k} = \left(2\pi\right)^{d/2} \left|\beta_k^0 \boldsymbol{\Sigma}_k\right|^{1/2} \tag{61}$$

$$Z_{\boldsymbol{\Sigma}_k} = 2^{\alpha_k^0 d/2} \pi^{d(d-1)/4} |\mathbf{S}_k^0|^{\alpha_k^0/2} \prod_{j=1}^d \Gamma\left(\frac{\alpha_k^0 + 1 - j}{2}\right). \tag{62}$$

In (57)–(62), λ_k^0, $\mathbf{m}_k^0$, β_k^0, α_k^0, and $\mathbf{S}_k^0$ are parameters of the priors, and Γ is the gamma function, defined as $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$.

The VB-E step from (54) is

$$\begin{aligned}
q\left(\mathbf{x}_i^{m_i}, \gamma_i = k\right) &\propto \exp\{\left\langle \log p\left(\mathbf{x}_i^{o_i}, \mathbf{x}_i^{m_i}, \gamma_i = k|\Theta\right)\right\rangle_\Theta\} \\
&= \exp\left\{\left\langle \log \pi_k \right\rangle_{\boldsymbol{\pi}} - \frac{d}{2}\log 2\pi + \frac{1}{2}\left\langle \log\left|\boldsymbol{\Sigma}_k^{-1}\right|\right\rangle_{\boldsymbol{\Sigma}}\right. \\
&- \left.\frac{1}{2}\mathrm{tr}\left(\left\langle \boldsymbol{\Sigma}_k^{-1}\left\langle\left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)^T\left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)\right\rangle_{\boldsymbol{\mu}|\boldsymbol{\Sigma}}\right\rangle_{\boldsymbol{\Sigma}}\right)\right\} \\
&= \exp\left\{\psi\left(\lambda_k\right) - \psi\left(\sum_{k=1}^K \lambda_k\right) - \frac{d}{2}\log 2\pi + \frac{1}{2}\sum_{j=1}^d \psi\left(\frac{\alpha_k + 1 - j}{2}\right)\right. \\
&+ \left.\frac{d}{2}\log 2 + \frac{1}{2}\log|\mathbf{S}_k| - \frac{1}{2}\left(\mathbf{x}_i - \mathbf{m}_k\right)^T \alpha_k \mathbf{S}_k \left(\mathbf{x}_i - \mathbf{m}_k\right) - \mathrm{tr}\left(\beta_k \mathbf{I}_d\right)\right\} \\
&= A_k \mathcal{N}\left(\mathbf{x}_i; \mathbf{m}_k, \alpha_k^{-1}\mathbf{S}_k^{-1}\right)
\end{aligned} \tag{63}$$

where

$$A_k = \exp\left\{\psi\left(\lambda_k\right) - \psi\left(\sum_{k=1}^{K}\lambda_k\right) + \frac{1}{2}\sum_{j=1}^{d}\psi\left(\frac{\alpha_k + 1 - j}{2}\right) + \frac{d}{2}\log 2 - \frac{1}{2}\log\alpha_k - \mathbf{tr}\left(\beta_k \mathbf{I}_d\right)\right\}, \tag{64}$$

$\mathbf{I}_d$ is the d-dimensional identity matrix, and ψ is the digamma function, defined as $\psi(z) = \frac{d}{dz}\log\Gamma(z)$. Upon normalizing, the updated quantity is

$$\begin{aligned} q\left(\mathbf{x}_i^{m_i}, \gamma_i = k\right) &= \frac{A_k \mathcal{N}\left(\mathbf{x}_i; \mathbf{m}_k, \alpha_k^{-1}\mathbf{S}_k^{-1}\right)}{\sum_{\ell=1}^{K} A_\ell \mathcal{N}\left(\mathbf{x}_i^{o_i}; \mathbf{m}_\ell^{o_i}, \alpha_\ell^{-1}\mathbf{S}_\ell^{-1,o_i o_i}\right)} \\ &= \widetilde{\delta}_k^i \mathcal{N}\left(\mathbf{x}_i^{m_i}; \mathbf{m}_k^{m_i|o_i}, \mathbf{S}_k^{m_i|o_i}\right) \end{aligned} \tag{65}$$

where

$$\widetilde{\delta}_k^i = \frac{A_k \mathcal{N}\left(\mathbf{x}_i^{o_i}; \mathbf{m}_k^{o_i}, \alpha_k^{-1}\mathbf{S}_k^{-1,o_i o_i}\right)}{\sum_{\ell=1}^{K} A_\ell \mathcal{N}\left(\mathbf{x}_i^{o_i}; \mathbf{m}_\ell^{o_i}, \alpha_\ell^{-1}\mathbf{S}_\ell^{-1,o_i o_i}\right)} \tag{66}$$

$$\mathbf{m}_k^{m_i|o_i} = \mathbf{m}_k^{m_i} + \mathbf{S}_k^{-1,m_i o_i}(\mathbf{S}_k^{-1,o_i o_i})^{-1}\left(\mathbf{x}_i^{o_i} - \mathbf{m}_k^{o_i}\right) \tag{67}$$

$$\mathbf{S}_k^{m_i|o_i} = \alpha_k^{-1}\left(\mathbf{S}_k^{-1,m_i m_i} - \mathbf{S}_k^{-1,m_i o_i}(\mathbf{S}_k^{-1,o_i o_i})^{-1}(\mathbf{S}_k^{-1,m_i o_i})^T\right). \tag{68}$$

Notation of the form $\mathbf{S}_k^{-1,o_i o_i}$ denotes a particular sub-matrix *after* the inverse is taken.

For a single data point $\mathbf{x}_i$, the VB-M step from (55) is

$$
\begin{aligned}
q(\Theta) &\propto p(\Theta)\exp\left(\langle\log p(\mathbf{x}_i^{o_i},\mathbf{x}_i^{m_i},\gamma_i=k|\Theta)\rangle_{\mathbf{x}_i^{m_i},\gamma_i=k}\right)\\
&= p(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})\exp\left\{\left\langle \log\pi_k-\frac{d}{2}\log 2\pi-\frac{1}{2}\log|\boldsymbol{\Sigma}_k|\right.\right.\\
&- \left.\left.\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\langle(\mathbf{x}_i-\boldsymbol{\mu}_k)^T(\mathbf{x}_i-\boldsymbol{\mu}_k)\rangle_{\mathbf{x}_i^{m_i}|\gamma_i=k}\right)\right\rangle_{\gamma_i=k}\right\}\\
&= p(\boldsymbol{\pi})\,p(\boldsymbol{\mu},\boldsymbol{\Sigma})\exp\left\{\sum_{k=1}^{K}\widetilde{\delta}_k^i\left(\log\pi_k-\frac{1}{2}\log|\boldsymbol{\Sigma}_k|\right.\right.\\
&- \frac{1}{2}\left(\begin{bmatrix}\mathbf{x}_i^{o_i}\\ \mathbf{m}_k^{m_i|o_i}\end{bmatrix}-\boldsymbol{\mu}_k\right)^T\boldsymbol{\Sigma}_k^{-1}\left(\begin{bmatrix}\mathbf{x}_i^{o_i}\\ \mathbf{m}_k^{m_i|o_i}\end{bmatrix}-\boldsymbol{\mu}_k\right)\\
&- \left.\left.\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_k^{m_i|o_i}\end{bmatrix}\right)\right)-\frac{d}{2}\log 2\pi\right\}\\
&\propto p(\boldsymbol{\pi})\,p(\boldsymbol{\mu}|\boldsymbol{\Sigma})\,p(\boldsymbol{\Sigma})\prod_{k=1}^{K}\pi_k^{\widetilde{\delta}_k^i}\left|\boldsymbol{\Sigma}_k^{-1}\right|^{\widetilde{\delta}_k^i/2}\exp\left\{-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\widetilde{\delta}_k^i\widetilde{\mathbf{S}}_k^i\right)\right\}\\
&\times \exp\left\{-\frac{1}{2}\widetilde{\delta}_k^i\left(\widetilde{\mathbf{x}}_i^k-\boldsymbol{\mu}_k\right)^T\boldsymbol{\Sigma}_k^{-1}\left(\widetilde{\mathbf{x}}_i^k-\boldsymbol{\mu}_k\right)\right\} \qquad (69)
\end{aligned}
$$

where

$$
\widetilde{\mathbf{x}}_i^k=\begin{bmatrix}\mathbf{x}_i^{o_i}\\ \mathbf{m}_k^{m_i|o_i}\end{bmatrix}\qquad\text{and}\qquad\widetilde{\mathbf{S}}_k^i=\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_k^{m_i|o_i}\end{bmatrix}. \qquad (70)
$$

For the entire data set $\{\mathbf{x}_i\}_{i=1}^N$, we obtain

$$
\begin{aligned}
q(\Theta) &\propto p(\boldsymbol{\pi})\, p(\boldsymbol{\mu}|\boldsymbol{\Sigma})\, p(\boldsymbol{\Sigma}) \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k^{\widetilde{\delta}_k^i} \left|\boldsymbol{\Sigma}_k^{-1}\right|^{\widetilde{\delta}_k^i/2} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\widetilde{\delta}_k^i \widetilde{\mathbf{S}}_k^i\right)\right\} \\
&\times \exp\left\{-\frac{1}{2}\widetilde{\delta}_k^i \left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right)\right\} \\
&\propto \left[\prod_{k=1}^{K} \pi_k^{\lambda_k^0 - 1} e^{-\frac{1}{2}\mathrm{tr}\left\{\left(\boldsymbol{\mu}_k - \mathbf{m}_k^0\right)\left(\boldsymbol{\mu}_k - \mathbf{m}_k^0\right)^T \beta_k^{0,-1}\boldsymbol{\Sigma}_k^{-1}\right\}} \left|\boldsymbol{\Sigma}_k^{-1}\right|^{\left(\alpha_k^0 - d - 1\right)/2} e^{-\frac{1}{2}\mathrm{tr}\left\{\mathbf{S}_k^{0,-1}\boldsymbol{\Sigma}_k^{-1}\right\}}\right] \\
&\times \left[\prod_{k=1}^{K} \pi_k^{\sum_{i=1}^N \widetilde{\delta}_k^i}\right] \left[\prod_{k=1}^{K} \left|\boldsymbol{\Sigma}_k^{-1}\right|^{\frac{1}{2}\sum_{i=1}^N \widetilde{\delta}_k^i} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\sum_{i=1}^{N}\widetilde{\delta}_k^i \widetilde{\mathbf{S}}_k^i\right)\right)\right. \\
&\times \left.\exp\left\{-\frac{1}{2}\sum_{i=1}^{N}\widetilde{\delta}_k^i \left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\widetilde{\mathbf{x}}_i^k - \boldsymbol{\mu}_k\right)\right\}\right] \\
&= \prod_{k=1}^{K} \pi_k^{\lambda_k^0 + \sum_{i=1}^N \widetilde{\delta}_k^i - 1} \left|\boldsymbol{\Sigma}_k^{-1}\right|^{\frac{1}{2}\left(\alpha_k^0 - d - 1 + \sum_{i=1}^N \widetilde{\delta}_k^i\right)} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}\left[\mathbf{S}_k^{0,-1}\right.\right.\right. \\
&+ \left.\left.\left.\sum_{i=1}^{N}\widetilde{\delta}_k^i \widetilde{\mathbf{S}}_k^i + \beta_k^{0,-1}\mathbf{m}_k^0 \mathbf{m}_k^{0\,T} + \sum_{i=1}^{N}\widetilde{\delta}_k^i \widetilde{\mathbf{x}}_i^k \widetilde{\mathbf{x}}_i^{k\,T} - \left(\beta_k^{0,-1} + \sum_{i=1}^{N}\widetilde{\delta}_k^i\right)\overline{\mathbf{x}}_i^k \overline{\mathbf{x}}_i^{k\,T}\right]\right)\right\} \\
&\times \exp\left\{-\frac{\beta_k^{0,-1} + \sum_{i=1}^N \widetilde{\delta}_k^i}{2}\left(\boldsymbol{\mu}_k - \overline{\mathbf{x}}_i^k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{\mu}_k - \overline{\mathbf{x}}_i^k\right)\right\}
\end{aligned} \tag{71}
$$

where

$$
\overline{\mathbf{x}}_i^k = \frac{\beta_k^{0,-1}\mathbf{m}_k^0 + \sum_{i=1}^N \widetilde{\delta}_k^i \widetilde{\mathbf{x}}_i^k}{\beta_k^{0,-1} + \sum_{i=1}^N \widetilde{\delta}_k^i}. \tag{72}
$$

The update formulas of the GMM posterior parameters are then

$$
\lambda_k^{\text{new}} = \lambda_k^0 + \sum_{i=1}^{N}\widetilde{\delta}_k^i \tag{73}
$$

$$
\mathbf{m}_k^{\text{new}} = \frac{\beta_k^{0,-1}\mathbf{m}_k^0 + \sum_{i=1}^N \widetilde{\delta}_k^i \widetilde{\mathbf{x}}_i^k}{\beta_k^{0,-1} + \sum_{i=1}^N \widetilde{\delta}_k^i} \tag{74}
$$

$$
\beta_k^{\text{new},-1} = \beta_k^{0,-1} + \sum_{i=1}^{N}\widetilde{\delta}_k^i \tag{75}
$$

$$\mathbf{S}_k^{\text{new},-1} = \mathbf{S}_k^{0,-1} + \sum_{i=1}^{N} \widetilde{\delta}_k^i \widetilde{\mathbf{S}}_k^i + \beta_k^{0,-1} \mathbf{m}_k^0 \mathbf{m}_k^{0\,T} + \sum_{i=1}^{N} \widetilde{\delta}_k^i \widetilde{\mathbf{x}}_i^k \widetilde{\mathbf{x}}_i^{k\,T} - \left(\beta_k^{0,-1} + \sum_{i=1}^{N} \widetilde{\delta}_k^i \right) \overline{\mathbf{x}}_i^k \overline{\mathbf{x}}_i^{k\,T} \quad (76)$$

$$\alpha_k^{\text{new}} = \alpha_k^0 + \sum_{i=1}^{N} \widetilde{\delta}_k^i. \quad (77)$$

APPENDIX B

Here we derive (33), the expression for the log-prior with missing data integrated out:

$$\begin{aligned} \log p(\mathbf{w}|\{\mathbf{x}_i^{o_i}\}_{i=1}^N) &= (-\lambda/2) \int \mathbf{w}^T \mathbf{X} \boldsymbol{\Delta} \mathbf{X}^T \mathbf{w} \left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}) \right] d\mathbf{x}_1^{m_1} \cdots d\mathbf{x}_N^{m_N} \\ &= (-\lambda/2) \mathbf{w}^T (\widetilde{\mathbf{X}} \boldsymbol{\Delta} \widetilde{\mathbf{X}}^T + \boldsymbol{\Phi}) \mathbf{w} \end{aligned} \quad (78)$$

with $\widetilde{\mathbf{X}}$ and $\boldsymbol{\Phi}$ defined in (34) and (35), respectively.

First define

$$\widetilde{\mathbf{G}} = \int \mathbf{X} \boldsymbol{\Delta} \mathbf{X}^T \left[\prod_{i=1}^{N} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}) \right] d\mathbf{x}_1^{m_1} \cdots d\mathbf{x}_N^{m_N} \quad (79)$$

so

$$\log p(\mathbf{w}|\{\mathbf{x}_i^{o_i}\}_{i=1}^N) = (-\lambda/2) \mathbf{w}^T \widetilde{\mathbf{G}} \mathbf{w}. \quad (80)$$

The ab-th element of $\widetilde{\mathbf{G}}$ can be written

$$\widetilde{G}_{ab} = \sum_{i=1}^{N} \sum_{j=1}^{N} \widetilde{G}_{ab}^{ij} \quad (81)$$

by defining the contribution from x_{ia} and x_{jb} as

$$\widetilde{G}_{ab}^{ij} = \int x_{ia} \Delta_{ij} x_{jb} p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i}) p(\mathbf{x}_j^{m_j}|\mathbf{x}_j^{o_j}) d\mathbf{x}_i^{m_i} d\mathbf{x}_j^{m_j}. \quad (82)$$

When $i \neq j$, we can re-write $\widetilde{G}_{ab}^{ij}$ as

$$\widetilde{G}_{ab}^{ij} = \int x_{ia}\Delta_{ij}x_{jb}p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})p(\mathbf{x}_j^{m_j}|\mathbf{x}_j^{o_j})d\mathbf{x}_i^{m_i}d\mathbf{x}_j^{m_j} \tag{83}$$

$$= \Delta_{ij}\int x_{ia}p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})d\mathbf{x}_i^{m_i}\int x_{jb}p(\mathbf{x}_j^{m_j}|\mathbf{x}_j^{o_j})d\mathbf{x}_j^{m_j}. \tag{84}$$

Since the distribution $p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})$ is a GMM, it is easy to see that

$$\widetilde{G}_{ab}^{ij} = \begin{cases} \Delta_{ij}\left[\sum_{k=1}^K \delta_k^i \xi_k^{m_i[a]}\right]\left[\sum_{\ell=1}^K \delta_\ell^i \xi_\ell^{m_j[b]}\right] & \text{if } a \in m_i \text{ and } b \in m_j \\ \Delta_{ij}x_{ia}\left[\sum_{k=1}^K \delta_k^i \xi_k^{m_j[b]}\right] & \text{if } a \in o_i \text{ and } b \in m_j \\ \Delta_{ij}\left[\sum_{k=1}^K \delta_k^i \xi_k^{m_i[a]}\right]x_{jb} & \text{if } a \in m_i \text{ and } b \in o_j \\ \Delta_{ij}x_{ia}x_{jb} & \text{if } a \in o_i \text{ and } b \in o_j, \end{cases} \tag{85}$$

where δ_k^i and $\xi_k^{m_i}$ are defined in (9) and (12), respectively. When $i = j$, we can re-write $\widetilde{G}_{ab}^{ii}$ as

$$\widetilde{G}_{ab}^{ii} = \Delta_{ii}\int x_{ia}x_{ib}p(\mathbf{x}_i^{m_i}|\mathbf{x}_i^{o_i})d\mathbf{x}_i^{m_i}. \tag{86}$$

It is again easy to see that

$$\widetilde{G}_{ab}^{ii} = \begin{cases} \Delta_{ii}\sum_{k=1}^K \delta_k^i(\xi_k^{m_i[a]}\xi_k^{m_i[b]} + \Omega_k^{m_i[ab]}) & \text{if } a \in m_i \text{ and } b \in m_i \\ \Delta_{ii}x_{ia}\left[\sum_{k=1}^K \delta_k^i \xi_k^{m_i[b]}\right] & \text{if } a \in o_i \text{ and } b \in m_i \\ \Delta_{ii}\left[\sum_{k=1}^K \delta_k^i \xi_k^{m_i[a]}\right]x_{ib} & \text{if } a \in m_i \text{ and } b \in o_i \\ \Delta_{ii}x_{ia}x_{ib} & \text{if } a \in o_i \text{ and } b \in o_i. \end{cases} \tag{87}$$

Combining (85) and (87) elegantly then gives (33).

APPENDIX C

For the semi-supervised classifier, evidence maximization [13] is used to select the appropriate value of λ. From Bayes' rule, the log-evidence for the labeled data (for *any* $\mathbf{w}$) is

$$\log p(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i^o\}_{i=1}^{N},\{\epsilon_i\}_{i=1}^{N_L},\lambda) = \log p(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i^o\}_{i=1}^{N_L},\{\epsilon_i\}_{i=1}^{N_L},\mathbf{w}) + \log p(\mathbf{w}|\{\mathbf{x}_i^o\}_{i=1}^{N},\lambda) - \log p(\mathbf{w}|\{y_i\}_{i=1}^{N_L},\{\mathbf{x}_i^o\}_{i=1}^{N},\lambda). \quad (88)$$

The result of the gradient ascent algorithm (for a given value of λ) is a classifier $\widehat{\mathbf{w}}$. A Laplace approximation [13] for the posterior of $\mathbf{w}$ can then be made about this $\widehat{\mathbf{w}}$. This Laplace approximation is a Taylor series expansion that results in the approximation

$$p(\mathbf{w}|\{y_i\}_{i=1}^{N_L},\{\mathbf{x}_i^o\}_{i=1}^{N},\lambda) \approx \mathcal{N}(\mathbf{w};\widehat{\mathbf{w}},\boldsymbol{\Sigma}_w^{-1}) \quad (89)$$

where $\widetilde{\mathbf{G}} = \widetilde{\mathbf{X}}\boldsymbol{\Delta}\widetilde{\mathbf{X}}^T + \boldsymbol{\Phi}$ (see (33)), $\mathbf{H} = -\frac{\partial^2 \ell(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^T}\Big|_{\mathbf{w}=\widehat{\mathbf{w}}}$ is the negative Hessian of the log-likelihood evaluated at $\widehat{\mathbf{w}}$, and $\boldsymbol{\Sigma}_w = \lambda\widetilde{\mathbf{G}} + \mathbf{H}$. The log-evidence evaluated at $\widehat{\mathbf{w}}$ is then

$$\begin{aligned}
\log p(\{y_i\}_{i=1}^{N_L}|\{\mathbf{x}_i^o\}_{i=1}^{N},\{\epsilon_i\}_{i=1}^{N_L},\lambda) &\approx \ell(\widehat{\mathbf{w}}) + \log\mathcal{N}(\widehat{\mathbf{w}};\mathbf{0},(\lambda\widetilde{\mathbf{G}})^{-1}) - \log\mathcal{N}(\widehat{\mathbf{w}};\widehat{\mathbf{w}},\boldsymbol{\Sigma}_w^{-1}) \\
&= \ell(\widehat{\mathbf{w}}) + \frac{1}{2}\log|\lambda\widetilde{\mathbf{G}}| - \frac{\lambda}{2}\widehat{\mathbf{w}}^T\widetilde{\mathbf{G}}\widehat{\mathbf{w}} + \frac{1}{2}\log|\boldsymbol{\Sigma}_w^{-1}| \\
&= \ell(\widehat{\mathbf{w}}) - \frac{\lambda}{2}\widehat{\mathbf{w}}^T\widetilde{\mathbf{G}}\widehat{\mathbf{w}} + \frac{1}{2}\log\left[\frac{|\lambda\widetilde{\mathbf{G}}|}{|\lambda\widetilde{\mathbf{G}}+\mathbf{H}|}\right], \quad (90)
\end{aligned}$$

where the log-likelihood function $\ell(\widehat{\mathbf{w}})$ is given by (23). For each value of λ considered, a unique classifier is trained and the log-evidence is computed. We then choose to use that value of λ for which the log-evidence was a maximum. The drawback of this method, of course, is that the classifier must be trained several times.

APPENDIX D

The results of single-sided paired t-tests to test whether the proposed method (and its extensions, where applicable) is better than various competing methods are presented here. The results in Tables I, II, III, IV, and V, correspond to the results shown in Figures 3, 4, 5, 6, and 7, respectively. Bold values in all tables indicate that the proposed method is better at a significance level of 95%.

TABLE I

SINGLE-SIDED PAIRED t-TEST VALUES TO TEST WHETHER THE PROPOSED METHOD THAT ANALYTICALLY INTEGRATES OUT THE MISSING DATA IS BETTER THAN MULTIPLE IMPUTATION, FOR THE RESULTS SHOWN IN FIGURE 3.

NUMBER OF IMPUTATIONS	1	2	3	4	5	10	20	50	100
t-VALUE	**14.965**	**11.598**	**8.143**	**7.396**	**6.525**	**2.901**	1.017	-0.175	-1.043

TABLE II

SINGLE-SIDED PAIRED t-TEST VALUES FOR THE RESULTS SHOWN IN FIGURE 4. THE COMPETING METHODS ARE COMPARED TO THE PROPOSED METHOD THAT USES THE VB-EM ALGORITHM FOR THE DENSITY FUNCTION ESTIMATION.

COMPETING METHOD	FRACTION OF FEATURES MISSING	FRACTION OF DATA USED AS TRAINING DATA								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
PROPOSED (EM)	0.25	**7.145**	**3.412**	**4.910**	**2.441**	**3.821**	1.699	1.395	1.409	-0.302
C. MEAN IMP. (VB)	0.25	**3.744**	**6.090**	**6.949**	**5.178**	**4.642**	**6.485**	1.248	**2.006**	1.651
C. MEAN IMP. (EM)	0.25	**7.789**	**5.289**	**9.065**	**6.557**	**3.823**	**2.762**	**1.934**	1.748	0.357
U. MEAN IMP.	0.25	0.143	**10.085**	**5.539**	**5.223**	**2.519**	0.956	0.412	-0.966	**2.296**
PROPOSED (EM)	0.50	**2.102**	**2.037**	**2.966**	0.989	**2.077**	**2.533**	1.296	1.833	0.800
C. MEAN IMP. (VB)	0.50	**3.371**	**4.774**	**6.872**	**5.989**	**3.192**	**6.513**	**2.008**	**2.226**	1.008
C. MEAN IMP. (EM)	0.50	**2.882**	**6.062**	**9.811**	**3.304**	**3.133**	**5.512**	1.810	**2.543**	0.784
U. MEAN IMP.	0.50	1.588	**2.911**	**5.869**	**1.952**	1.223	**4.567**	**3.329**	1.239	**2.367**
PROPOSED (EM)	0.75	-1.443	**1.841**	**2.624**	1.232	-1.142	-1.125	0.320	0.142	-0.086
C. MEAN IMP. (VB)	0.75	**9.819**	**8.019**	**3.825**	**2.195**	**3.453**	**2.884**	**2.750**	0.455	-0.827
C. MEAN IMP. (EM)	0.75	1.431	**6.871**	**3.660**	**2.837**	0.198	0.552	0.542	0.346	0.035
U. MEAN IMP.	0.75	1.513	**9.997**	**4.927**	**4.374**	0.958	1.241	1.172	-0.138	1.656

TABLE III

SINGLE-SIDED PAIRED t-TEST VALUES FOR THE RESULTS SHOWN IN FIGURE 5. THE COMPETING METHODS ARE COMPARED TO THE PROPOSED METHOD THAT USES THE VB-EM ALGORITHM FOR THE DENSITY FUNCTION ESTIMATION.

COMPETING METHOD	FRACTION OF FEATURES MISSING	FRACTION OF DATA USED AS TRAINING DATA								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
C. MEAN IMP.	0.25	1.726	**2.075**	**2.645**	1.496	**2.367**	**3.742**	**3.339**	**2.531**	0.729
U. MEAN IMP.	0.25	1.042	**2.848**	**2.355**	**2.916**	**2.262**	**3.272**	**6.229**	**2.128**	-0.697
C. MEAN IMP.	0.50	**1.942**	**1.875**	**3.806**	**2.405**	**2.607**	**2.5560**	1.753	0.368	-1.109
U. MEAN IMP.	0.50	**2.175**	1.801	**3.563**	**3.608**	**4.249**	**1.944**	1.686	-0.959	1.427
C. MEAN IMP.	0.75	**4.300**	**4.278**	**5.328**	**3.854**	**2.353**	-1.605	1.433	0.365	0.327
U. MEAN IMP.	0.75	**3.050**	0.969	**2.317**	1.022	0.337	-2.378	-0.530	-0.505	0.079

TABLE IV

SINGLE-SIDED PAIRED t-TEST VALUES FOR THE RESULTS SHOWN IN FIGURE 6. THE COMPETING METHODS ARE COMPARED TO THE PROPOSED METHOD THAT ACCOUNTS FOR THE IMPERFECT LABELING.

COMPETING METHOD	TRUE ϵ	FRACTION OF FEATURES MISSING				
		0.10	0.25	0.50	0.75	0.90
INCOMPLETE-DATA WITH $\epsilon = 0$	0.1	1.2321	1.3967	0.7150	-1.7654	-1.4812
U. MEAN IMPUTATION	0.1	**2.7691**	**2.6505**	**3.7313**	**3.6148**	1.7102
INCOMPLETE-DATA WITH $\epsilon = 0$	0.2	**2.9620**	**2.8248**	0.5561	**2.7896**	-0.2977
U. MEAN IMPUTATION	0.2	**4.5836**	**2.9459**	**4.3023**	**3.9020**	**2.7614**
INCOMPLETE-DATA WITH $\epsilon = 0$	0.3	**2.3147**	1.3337	-0.3414	1.2117	-1.2967
U. MEAN IMPUTATION	0.3	**4.9198**	**1.8443**	**2.1837**	**1.9789**	**1.7808**

TABLE V

SINGLE-SIDED PAIRED t-TEST VALUES FOR THE RESULTS SHOWN IN FIGURE 7. THE COMPETING METHODS ARE COMPARED TO THE PROPOSED ALGORITHM WITH THE SEMI-SUPERVISED EXTENSION.

COMPETING METHOD	FRACTION OF DATA LABELED	FRACTION OF FEATURES MISSING				
		0.10	0.25	0.50	0.75	0.90
PROPOSED SUPERVISED	0.25	**4.1382**	**7.0724**	**5.1539**	**9.1933**	**5.6102**
SEMI-SUPERVISED (MEAN IMPUTATION)	0.25	**2.5336**	**3.5985**	**9.1012**	**4.0352**	**2.7173**
PROPOSED SUPERVISED	0.50	**5.7923**	**6.4795**	**3.7683**	**3.3061**	1.1096
SEMI-SUPERVISED (MEAN IMPUTATION)	0.50	**1.8168**	**4.3674**	**3.0337**	**4.3149**	**2.2872**
PROPOSED SUPERVISED	0.75	**2.9355**	**3.4984**	**2.4087**	**3.3309**	-2.0963
SEMI-SUPERVISED (MEAN IMPUTATION)	0.75	**4.4778**	0.3937	**3.2437**	**3.0519**	-0.0687