Analytical Kernel Matrix Completion with Incomplete Multi-View Data

David Williams DPW@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA

Abstract

In multi-view remote sensing applications, incomplete data can result when only a subset of sensors are deployed at certain regions. We derive a closed-form expression for computing a Gaussian kernel when faced with incomplete data. This expression is obtained by analytically integrating out the missing data. This result can subsequently be used in conjunction with any kernel-based classifier. The superiority of the proposed method over two common imputation schemes is demonstrated on one benchmark data set and three real (measured) multi-view land mine data sets.

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 multi-view 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, information sources, or "views" will make such incomplete data problems more common (see (Tsuda, Akaho & Asai, 2003; Lanckriet et al., 2004)).

Incomplete data problems are often circumvented in the initial stage of analysis—before specific algorithms become involved—via imputation (*i.e.,* by "completing" the 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, $p\left(\boldsymbol{x}_i^{m_i}|\boldsymbol{x}_i^{o_i}\right)$).

When kernel methods such as the SVM (Schölkopf & Smola, 2002) are employed, one can either first complete the data and then compute the kernel matrix, or else complete and compute the kernel matrix in a single stage. Semidefinite programming has been used to complete kernel matrices that have only a *limited* number of missing elements (Graepel, 2002). The *em* algorithm (Tsuda, Akaho & Asai, 2003) is applicable when both an incomplete auxiliary kernel matrix and a complete primary kernel matrix exist, but not when the patterns of missing data are completely arbitrary. This assumption may be tolerable in certain applications, but it is not appropriate for the general missing data problem.

By making only two assumptions—that $p\left(\boldsymbol{x}_i^{m_i}|\boldsymbol{x}_i^{o_i}\right)$ is a Gaussian mixture model (GMM), and that a Gaussian kernel is employed—we can analytically calculate the kernel matrix from incomplete data by integrating out the missing data. The first assumption is mild, since it is well-known that a mixture of Gaussians can approximate any distribution. The second assumption is not overly limiting as the Gaussian kernel is one of the most commonly used kernel forms. In fact, if one assumes a linear or polynomial kernel instead of a Gaussian kernel, the missing data of the kernel matrix can still be analytically integrated out. However, the calculations for these kernels are trivial, so we focus here on the more interesting case of the Gaussian kernel. After obtaining the kernel matrix, any kernel-based method may be employed, as would be done for an ordinary complete-data problem.

This paper is organized as follows. In Section 2, we derive the expression to analytically compute a kernel

Appearing in *Proceedings of the Workshop on Learning with Multiple Views*, 22^{nd} ICML, Bonn, Germany, 2005.

matrix in the presence of incomplete data for Gaussian kernels. Experimental classification results on one benchmark machine learning data set and on three real multi-view land mine data sets are shown in Section 3, before concluding remarks are made in Section 4.

2. Kernel Matrix with Incomplete Data

A data point $\boldsymbol{x}_i$ may be mapped into feature space via the positive semidefinite kernel function K. Computing the kernel for every pair of data points results in the symmetric, positive semidefinite kernel matrix $\mathbf{K}$. The ij-th entry of this kernel matrix, K_{ij}, is a measure of similarity between two data points, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Our goal is to obtain the kernel matrix when incomplete data exists. We solve this task in a multi-view framework, treating incomplete data as the result of only a subset of views being observed for any given data point. However, this framework is not limiting because one can simply treat each individual feature as coming from a unique view.

2.1. Derivation of the Kernel Matrix

Let $\boldsymbol{x}_i^s$ denote the data (features) of the i-th data point from the set of views s. Let o_i and m_i denote the sets of observed views and missing views for data point $\boldsymbol{x}_i$, respectively. The notation $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ indicates that $\boldsymbol{x}$ is distributed as a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Using all available data, we model the joint probability $p\left(\boldsymbol{x}_i^{m_i}, \boldsymbol{x}_i^{o_i}\right)$ using a (Z-component) Gaussian mixture model

$$\begin{aligned} p\left(\boldsymbol{x}_i^{m_i}, \boldsymbol{x}_i^{o_i}\right) &= \sum_{\zeta=1}^{Z} \varpi_\zeta \mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\theta}_\zeta, \boldsymbol{\Theta}_\zeta\right) \\ &= \sum_{\zeta=1}^{Z} \varpi_\zeta \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{x}_i^{m_i} \\ \boldsymbol{x}_i^{o_i} \end{array}\right] \middle| \left[\begin{array}{c} \boldsymbol{\theta}_\zeta^{m_i} \\ \boldsymbol{\theta}_\zeta^{o_i} \end{array}\right], \right. \\ &\qquad \left. \left[\begin{array}{cc} \boldsymbol{\Theta}_\zeta^{[m_i m_i]} & \boldsymbol{\Theta}_\zeta^{[m_i o_i]} \\ \left(\boldsymbol{\Theta}_\zeta^{[m_i o_i]}\right)^T & \boldsymbol{\Theta}_\zeta^{[o_i o_i]} \end{array}\right]\right) \end{aligned} \quad (1)$$

where ϖ_ζ are the non-negative mixing coefficients that sum to unity.

Any conditional distribution derived from this joint probability will also be a mixture of Gaussians. Specifically,

$$p\left(\boldsymbol{x}_i^{m_i} \middle| \boldsymbol{x}_i^{o_i}\right) = \sum_{\zeta=1}^{Z} \pi_\zeta^i \mathcal{N}\left(\boldsymbol{x}_i^{m_i} \middle| \boldsymbol{\mu}_\zeta^{m_i}, \boldsymbol{\Sigma}_\zeta^{m_i}\right) \quad (2)$$

where

$$\begin{aligned} \pi_\zeta^i &= \frac{\varpi_\zeta \mathcal{N}\left(\boldsymbol{x}_i^{o_i} \middle| \boldsymbol{\theta}_\zeta^{o_i}, \boldsymbol{\Theta}_\zeta^{[o_i o_i]}\right)}{\sum_{\xi=1}^{Z} \varpi_\xi \mathcal{N}\left(\boldsymbol{x}_i^{o_i} \middle| \boldsymbol{\theta}_\xi^{o_i}, \boldsymbol{\Theta}_\xi^{[o_i o_i]}\right)} \\ \boldsymbol{\mu}_\zeta^{m_i} &= \boldsymbol{\theta}_\zeta^{m_i} + \boldsymbol{\Omega}\left(\boldsymbol{x}_i^{o_i} - \boldsymbol{\theta}_\zeta^{o_i}\right) \\ \boldsymbol{\Sigma}_\zeta^{m_i} &= \boldsymbol{\Theta}_\zeta^{[m_i m_i]} - \boldsymbol{\Omega}\left(\boldsymbol{\Theta}_\zeta^{[m_i o_i]}\right)^T \\ \boldsymbol{\Omega} &\equiv \boldsymbol{\Theta}_\zeta^{[m_i o_i]}\left(\boldsymbol{\Theta}_\zeta^{[o_i o_i]}\right)^{-1}. \end{aligned}$$

We also employ a Gaussian kernel function

$$\begin{aligned} K_{ij} = K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) &= z_\kappa \cdot \exp\left\{\frac{||\boldsymbol{x}_j - \boldsymbol{x}_i||_2^2}{-2\sigma_\kappa^2}\right\} \\ &= z_\kappa \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)^T \boldsymbol{\Sigma}_\kappa^{-1}\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)\right\} \end{aligned} \quad (3)$$

where $\boldsymbol{\Sigma}_\kappa = \mathrm{diag}\left(\sigma_\kappa^2\right)$ and $z_\kappa = (2\pi)^{-d/2}\left|\boldsymbol{\Sigma}_\kappa\right|^{-1/2}$. For $S > 1$ views, this kernel can be written as a product of the individual-view kernels in various forms:

$$\begin{aligned} K_{ij} &= \prod_{s=1}^{S} K_{ij}^s = \prod_{s=1}^{S} \mathcal{N}\left(\boldsymbol{x}_j^s \middle| \boldsymbol{x}_i^s, \boldsymbol{\Sigma}_\kappa^s\right) \\ &= \prod_{s=1}^{S} \mathcal{N}\left(\boldsymbol{x}_j^s - \boldsymbol{x}_i^s \middle| \boldsymbol{0}, \boldsymbol{\Sigma}_\kappa^s\right). \end{aligned}$$

In the following, we wish to derive the kernel K_{ij} between two arbitrary data points, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, with incomplete data. To do so, we will utilize the Gaussian mixture model of $p\left(\boldsymbol{x}_i^{m_i}, \boldsymbol{x}_i^{o_i}\right)$, which we assume has already been obtained. Because of the absence of data from some of the views, the incomplete data must be integrated out. For ease of reading, we give the complete derivation uninterrupted by text, opting to justify each step afterward. Note that $a \cap b$ indicates the intersection of sets a and b. The desired kernel is

$$\begin{aligned} & K\left(\boldsymbol{x}_i, \boldsymbol{x}_j \middle| \boldsymbol{x}_i^{o_i}, \boldsymbol{x}_j^{o_j}\right) \\ = & \int d\boldsymbol{x}_j^{m_j} \int d\boldsymbol{x}_i^{m_i} p\left(\boldsymbol{x}_i^{m_i}, \boldsymbol{x}_j^{m_j} \middle| \boldsymbol{x}_i^{o_i}, \boldsymbol{x}_j^{o_j}\right) \\ & K\left(\boldsymbol{x}_i, \boldsymbol{x}_j \middle| \boldsymbol{x}_i^{o_i}, \boldsymbol{x}_i^{m_i}, \boldsymbol{x}_j^{o_j}, \boldsymbol{x}_j^{m_j}\right) \\ \overset{(a)}{=} & \int d\boldsymbol{x}_j^{m_j} \int d\boldsymbol{x}_i^{m_i} p\left(\boldsymbol{x}_i^{m_i} \middle| \boldsymbol{x}_i^{o_i}\right) p\left(\boldsymbol{x}_j^{m_j} \middle| \boldsymbol{x}_j^{o_j}\right) \\ & K\left(\boldsymbol{x}_i, \boldsymbol{x}_j \middle| \boldsymbol{x}_i^{o_i}, \boldsymbol{x}_i^{m_i}, \boldsymbol{x}_j^{o_j}, \boldsymbol{x}_j^{m_j}\right) \\ \overset{(b)}{=} & K_{ij}^{o_i \cap o_j} \int d\boldsymbol{x}_j^{m_j} K_{ij}^{o_i \cap m_j} p\left(\boldsymbol{x}_j^{m_j} \middle| \boldsymbol{x}_j^{o_j}\right) \\ & \int d\boldsymbol{x}_i^{m_i} K_{ij}^{m_i} p\left(\boldsymbol{x}_i^{m_i} \middle| \boldsymbol{x}_i^{o_i}\right) \end{aligned}$$

$$
\begin{aligned}
&\overset{(c)}{=} K_{ij}^{o_i\cap o_j}\int d\boldsymbol{x}_j^{m_j} K_{ij}^{o_i\cap m_j} p\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{x}_j^{o_j}\right)\\
&\quad \int d\boldsymbol{x}_i^{m_i}\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_i^{m_i}\;\middle|\;\boldsymbol{\mu}_\zeta^{m_i},\boldsymbol{\Sigma}_\zeta^{m_i}\right)\\
&\quad \mathcal{N}\left(\boldsymbol{x}_j^{m_i}-\boldsymbol{x}_i^{m_i}\;\middle|\;\mathbf{0},\boldsymbol{\Sigma}_\kappa^{m_i}\right)\\
&\overset{(d)}{=} K_{ij}^{o_i\cap o_j}\int d\boldsymbol{x}_j^{m_j} K_{ij}^{o_i\cap m_j} p\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{x}_j^{o_j}\right)\\
&\quad \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i}\;\middle|\;\boldsymbol{\mu}_\zeta^{m_i},\boldsymbol{\Sigma}_\kappa^{m_i}+\boldsymbol{\Sigma}_\zeta^{m_i}\right)\\
&\overset{(e)}{=} K_{ij}^{o_i\cap o_j}\int d\boldsymbol{x}_j^{m_j} K_{ij}^{o_i\cap m_j} p\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{x}_j^{o_j}\right)\\
&\quad \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap m_j}\;\middle|\;\boldsymbol{f},\mathbf{F}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\\
&\overset{(f)}{=} K_{ij}^{o_i\cap o_j}\int d\boldsymbol{x}_j^{m_j} p\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{x}_j^{o_j}\right)\\
&\quad \mathcal{N}\left(\boldsymbol{x}_j^{o_i\cap m_j}\;\middle|\;\boldsymbol{x}_i^{o_i\cap m_j},\boldsymbol{\Sigma}_\kappa^{o_i\cap m_j}\right)\\
&\quad \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap m_j}\;\middle|\;\boldsymbol{f},\mathbf{F}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\\
&\overset{(g)}{=} K_{ij}^{o_i\cap o_j}\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\\
&\quad \int d\boldsymbol{x}_j^{m_j} p\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{x}_j^{o_j}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{a},\mathbf{A}\right)\\
&\overset{(h)}{=} K_{ij}^{o_i\cap o_j}\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\\
&\quad \int d\boldsymbol{x}_j^{m_j}\sum_{\xi=1}^{Z}\pi_\xi^j\mathcal{N}\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{b},\mathbf{B}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{a},\mathbf{A}\right)\\
&\overset{(i)}{=} K_{ij}^{o_i\cap o_j}\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\\
&\quad \sum_{\xi=1}^{Z}\pi_\xi^j\int d\boldsymbol{x}_j^{m_j} z_c\mathcal{N}\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{c},\mathbf{C}\right)\\
&\overset{(j)}{=} K_{ij}^{o_i\cap o_j}\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)\sum_{\xi=1}^{Z}\pi_\xi^j z_c.
\end{aligned}
\tag{4}
$$

In the derivation leading to (4), (a) follows because $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are independent; (b) follows by defining

$$
\begin{aligned}
K_{ij} &= K\left(\boldsymbol{x}_i,\boldsymbol{x}_j\;\middle|\;\boldsymbol{x}_i^{o_i},\boldsymbol{x}_i^{m_i},\boldsymbol{x}_j^{o_j},\boldsymbol{x}_j^{m_j}\right)\\
&= K_{ij}^{o_i}K_{ij}^{m_i}\\
&= K_{ij}^{o_i\cap m_j}K_{ij}^{o_i\cap o_j}K_{ij}^{m_i};
\end{aligned}
$$

(c) follows by writing

$$
\begin{aligned}
K_{ij}^{m_i} &= \mathcal{N}\left(\boldsymbol{x}_j^{m_i}-\boldsymbol{x}_i^{m_i}\;\middle|\;\mathbf{0},\boldsymbol{\Sigma}_\kappa^{m_i}\right)\\
&= \mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{x}_j^{m_i\cap m_j}-\boldsymbol{x}_i^{m_i\cap m_j}\\ \boldsymbol{x}_j^{o_i\cap m_j}-\boldsymbol{x}_i^{o_i\cap m_j}\end{array}\right]\;\middle|\;\left[\begin{array}{c}\mathbf{0}\\ \mathbf{0}\end{array}\right],\right.\\
&\qquad \left.\left[\begin{array}{cc}\boldsymbol{\Sigma}_\kappa^{m_i\cap m_j} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_\kappa^{m_i\cap o_j}\end{array}\right]\right)
\end{aligned}
$$

and

$$
\begin{aligned}
p\left(\boldsymbol{x}_i^{m_i}\;\middle|\;\boldsymbol{x}_i^{o_i}\right) &= \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_i^{m_i}\;\middle|\;\boldsymbol{\mu}_\zeta^{m_i},\boldsymbol{\Sigma}_\zeta^{m_i}\right)\\
&= \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{x}_i^{m_i\cap m_j}\\ \boldsymbol{x}_i^{o_i\cap m_j}\end{array}\right]\;\middle|\;\left[\begin{array}{c}\boldsymbol{\mu}_\zeta^{m_i\cap m_j}\\ \boldsymbol{\mu}_\zeta^{m_i\cap o_j}\end{array}\right],\right.\\
&\qquad \left.\left[\begin{array}{cc}\boldsymbol{\Sigma}_\zeta^{m_i[m_jm_j]} & \boldsymbol{\Sigma}_\zeta^{m_i[m_jo_j]}\\ \left(\boldsymbol{\Sigma}_\zeta^{m_i[m_jo_j]}\right)^T & \boldsymbol{\Sigma}_\zeta^{m_i[o_jo_j]}\end{array}\right]\right);
\end{aligned}
$$

(d) follows because the right-most integral is a convolution of two Gaussians; (e) follows from conditioning on $\boldsymbol{x}_j^{o_i\cap m_j}$ so that

$$
\begin{aligned}
&\sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i}\;\middle|\;\boldsymbol{\mu}_\zeta^{m_i},\boldsymbol{\Sigma}_\kappa^{m_i}+\boldsymbol{\Sigma}_\zeta^{m_i}\right)\\
&\quad = \sum_{\zeta=1}^{Z}\pi_\zeta^i\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap m_j}\;\middle|\;\boldsymbol{f},\mathbf{F}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap o_j}\;\middle|\;\boldsymbol{g},\mathbf{G}\right)
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{f} &= \boldsymbol{\mu}_\zeta^{m_i\cap m_j}+\boldsymbol{\Upsilon}\left(\boldsymbol{x}_j^{m_i\cap o_j}-\boldsymbol{\mu}_\zeta^{m_i\cap o_j}\right)\\
\mathbf{F} &= \left(\boldsymbol{\Sigma}_\kappa^{m_i\cap m_j}+\boldsymbol{\Sigma}_\zeta^{m_i[m_jm_j]}\right)-\boldsymbol{\Upsilon}\left(\boldsymbol{\Sigma}_\zeta^{m_i[m_jo_j]}\right)^T\\
\boldsymbol{\Upsilon} &\equiv \boldsymbol{\Sigma}_\zeta^{m_i[m_jo_j]}\left(\boldsymbol{\Sigma}_\kappa^{m_i\cap o_j}+\boldsymbol{\Sigma}_\zeta^{m_i[o_jo_j]}\right)^{-1}\\
\boldsymbol{g} &= \boldsymbol{\mu}_\zeta^{m_i\cap o_j}\\
\mathbf{G} &= \boldsymbol{\Sigma}_\kappa^{m_i\cap o_j}+\boldsymbol{\Sigma}_\zeta^{m_j[o_jo_j]};
\end{aligned}
$$

(f) follows because

$$
K_{ij}^{o_i\cap m_j} = \mathcal{N}\left(\boldsymbol{x}_j^{o_i\cap m_j}\;\middle|\;\boldsymbol{x}_i^{o_i\cap m_j},\boldsymbol{\Sigma}_\kappa^{o_i\cap m_j}\right);
$$

(g) follows since

$$
\begin{aligned}
&\mathcal{N}\left(\boldsymbol{x}_j^{o_i\cap m_j}\;\middle|\;\boldsymbol{x}_i^{o_i\cap m_j},\boldsymbol{\Sigma}_\kappa^{o_i\cap m_j}\right)\mathcal{N}\left(\boldsymbol{x}_j^{m_i\cap m_j}\;\middle|\;\boldsymbol{f},\mathbf{F}\right)\\
&\quad = \mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{x}_j^{m_i\cap m_j}\\ \boldsymbol{x}_j^{o_i\cap m_j}\end{array}\right]\;\middle|\;\left[\begin{array}{c}\boldsymbol{f}\\ \boldsymbol{x}_i^{o_i\cap m_j}\end{array}\right],\right.\\
&\qquad\qquad \left.\left[\begin{array}{cc}\mathbf{F} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_\kappa^{o_i\cap m_j}\end{array}\right]\right)\\
&\quad \equiv \mathcal{N}\left(\boldsymbol{x}_j^{m_j}\;\middle|\;\boldsymbol{a},\mathbf{A}\right);
\end{aligned}
$$

(h) follows because

$$
\begin{aligned}
p\left(\boldsymbol{x}_j^{m_j} \;\middle|\; \boldsymbol{x}_j^{o_j}\right) &= \sum_{\xi=1}^{Z} \pi_\xi^j \mathcal{N}\left(\boldsymbol{x}_j^{m_j} \;\middle|\; \boldsymbol{\mu}_\xi^{m_j}, \boldsymbol{\Sigma}_\xi^{m_j}\right) \\
&= \sum_{\xi=1}^{Z} \pi_\xi^j \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{x}_j^{m_i \cap m_j} \\ \boldsymbol{x}_j^{o_i \cap m_j} \end{array}\right] \middle| \left[\begin{array}{c} \boldsymbol{\mu}_\xi^{m_i \cap m_j} \\ \boldsymbol{\mu}_\xi^{o_i \cap m_j} \end{array}\right], \right. \\
&\qquad \left. \left[\begin{array}{cc} \boldsymbol{\Sigma}_\xi^{m_j[m_i m_i]} & \boldsymbol{\Sigma}_\xi^{m_j[m_i o_i]} \\ \left(\boldsymbol{\Sigma}_\xi^{m_j[m_i o_i]}\right)^T & \boldsymbol{\Sigma}_\xi^{m_j[o_i o_i]} \end{array}\right]\right) \\
&\equiv \sum_{\xi=1}^{Z} \pi_\xi^j \mathcal{N}\left(\boldsymbol{x}_j^{m_j} \;\middle|\; \boldsymbol{b}, \mathbf{B}\right);
\end{aligned}
$$

(i) follows from being a product of Gaussians where

$$
\begin{aligned}
\mathbf{C} &= \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} \\
\boldsymbol{c} &= \mathbf{C}\mathbf{A}^{-1}\boldsymbol{a} + \mathbf{C}\mathbf{B}^{-1}\boldsymbol{b} \\
z_c &= (2\pi)^{-d/2} |\mathbf{C}|^{+1/2} |\mathbf{A}|^{-1/2} |\mathbf{B}|^{-1/2} \\
&\times \exp\left\{-\frac{1}{2}\left[\boldsymbol{a}^T\mathbf{A}^{-1}\boldsymbol{a} + \boldsymbol{b}^T\mathbf{B}^{-1}\boldsymbol{b} - \boldsymbol{c}^T\mathbf{C}^{-1}\boldsymbol{c}\right]\right\};
\end{aligned}
$$

and (j) follows since the integral of a Gaussian is unity.

Thus, the Gaussian kernel between any two data points with incomplete data can be obtained analytically using (4). If a linear or polynomial kernel is chosen instead, the missing data can still be integrated out analytically. These cases are less interesting and quite trivial though. For instance, in the linear kernel case, analytically integrating out the missing data is equivalent to conditional mean imputation.

3. Experimental Results

We use a logistic regression classifier in this work. In logistic regression, the probability of label $y_i \in \{1, -1\}$ given the data point $\boldsymbol{x}_i$ is $p\left(y_i|\boldsymbol{x}_i\right) = \sigma(y_i \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i))$, where $\sigma(z) = (1+\exp(-z))^{-1}$ is the sigmoid function. A data point $\boldsymbol{x}_i$ is embedded into feature space via the transformation

$$
\boldsymbol{\phi}(\boldsymbol{x}_i) = \left[\begin{array}{cccc} 1 & K(\boldsymbol{x}_i, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_i, \boldsymbol{x}_N) \end{array}\right]
$$

where we use the positive semidefinite Gaussian kernel function K. As a result of this mapping, a nonlinear kernel classifier in the original input space can be constructed via a linear classifier in the transformed feature space. For a data set of N labeled data points, the (supervised) linear classifier $\boldsymbol{w}$ can be learned by maximizing the log-likelihood function $\ell(\boldsymbol{w}) = \sum_{i=1}^{N} \ln \sigma(y_i \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i))$.

Our proposed kernel matrix completion method is driven by the GMM in (1). In (Ghahramani & Jordan, 1994), the algorithm is given for estimating a GMM from incomplete data via the expectation-maximization (EM) algorithm.

We compared our proposed method to two common imputation schemes on four data sets. The difference among the three methods is how the kernel matrix is computed. Our proposed method uses (4) to analytically integrate out the missing data for the kernel matrix. In conditional mean imputation, all missing data is "completed" with the conditional mean, which is obtained via the GMM in (2). Specifically, the missing features of each data point are replaced with their conditional mean:

$$
\boldsymbol{x}_i^{m_i} \leftarrow \mathbb{E}[\boldsymbol{x}_i^{m_i} | \boldsymbol{x}_i^{o_i}] = \sum_{\zeta=1}^{Z} \pi_\zeta^i \boldsymbol{\mu}_\zeta^{m_i}.
$$

In unconditional mean imputation, all missing data is "completed" with the unconditional mean, which does not require a model of the data. For example, if $\boldsymbol{x}_i$ is missing feature a (*i.e.,* $a \in m_i$), unconditional mean imputation will make the substitution

$$
x_i^a \leftarrow \mathbb{E}[x_i^a] = \frac{1}{M} \sum_{j=1}^{M} x_{s(j)}^a
$$

where there are M data points for which feature a was observed, and $s(j)$ is the index of the j-th such data point. The Gaussian kernel matrices for these two imputation methods were then computed as for a regular complete-data problem (using (3)).

Note that after obtaining the kernel matrix for each of the methods, we possess a standard complete-data classification problem to which any kernel-based algorithm can be applied. For each of the three methods, we used the same logistic regression classifier form. As a result, any differences in performance among the three methods are strictly the result of the kernel matrix calculation.

A measure of classifier performance is the area under a receiver operating characteristic curve (AUC), which is given by the Wilcoxon statistic (Hanley & McNeil, 1982)

$$
\mathrm{AUC} = (MN)^{-1} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{1}_{x_m > y_n} \tag{5}
$$

where $x_1, \ldots, x_M$ are the classifier outputs of data belonging to class 1, $y_1, \ldots, y_N$ are the classifier outputs of data belonging to class -1, and $\mathbf{1}$ is an indicator function.

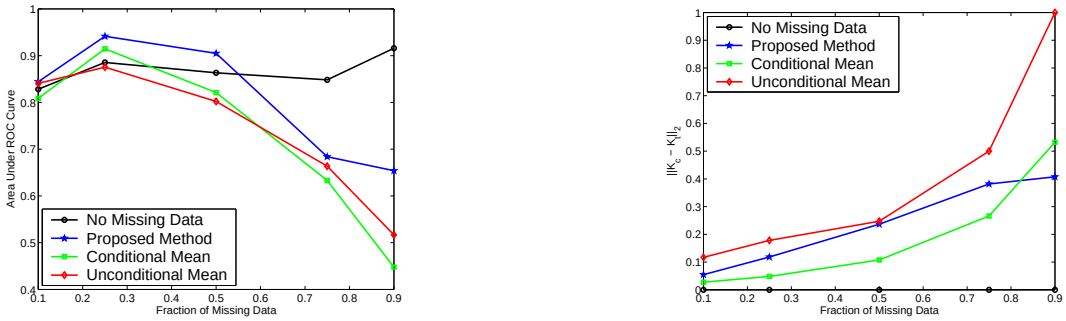

Figure 1. Experimental results on the IONOSPHERE data set. The figures show (a) AUC, and (b) the relative distances of the "estimated" kernel matrices from the "true" kernel matrix.

Table 1. Details of the 2-VIEW LAND MINE DETECTION data sets.

DATA SET	NUMBER OF MINES	NUMBER OF CLUTTER	NUMBER OF DATA POINTS WITH VIEW 1 ONLY	VIEW 2 ONLY	BOTH VIEWS	FRACTION OF MISSING DATA
AREA A	93	768	423	126	312	0.32
AREA B	139	693	134	146	552	0.17

3.1. Ionosphere

The proposed algorithm was first applied to the IONOSPHERE data set (from the UCI Machine Learning Repository), which has 351 data points and 34 features. In this example, the 34 features constitute 34 "views." Experimental results are shown in Figure 1(a) in terms of AUC, computed using (5). Each point on every curve is an average over 40 trials. Every trial consists of a random partition of training and testing data, and a random pattern of missing features (removed artificially). In every trial, 25% of the data was used as training data.

Since we artificially removed features, we can also build a classifier when there is no missing data. When no missing data exists, performance still varies as a function of the fraction of missing data because of the random partitions of training and testing data. From Figure 1(a), it can be seen that the proposed method consistently outperforms the imputation methods, with the most significant difference occurring when a large fraction of the data is missing. Remarkably, the proposed method sometimes achieves a larger AUC than that of the method with no missing data. We hypothesize that this phenomenon might occur if the missing feature values actually decrease or confuse class separation. In this case, their absence would be more beneficial than their presence.

We can also compute the Euclidean distance of each of the "estimated" kernel matrices (*i.e.,* the kernel matrices from the proposed or imputation methods) from the "true" kernel matrix (*i.e.,* the kernel matrix that would be obtained if all data was present). From Figure 1(b), it can be seen that as the fraction of missing data increases, the relative distance of the "estimated" kernel matrices to the "true" kernel matrix increases. Interestingly, the kernel matrix completed via conditional mean imputation is actually closer to the true kernel matrix than the proposed method's kernel matrix. We hypothesize that the proximity of the kernel matrix for the imputation method does not lead to better AUC because the single value imputation ignores the uncertainty of the missing data (Rässler, 2004).

3.2. 2-View Land Mine Detection

The proposed algorithm was also applied to two real data sets of 2-view land mine detection data. The goal for this data set is to classify mines (class 1) and clutter (class -1). The first view was an electro-optic infrared (EOIR) sensor, while the second view was a synthetic aperture radar (SAR) sensor. Data from each of the sensors were characterized by nine features. Details of these two data sets are summarized in Table 1.

For these experiments, 25% of the data was used as training data, while the remainder was used as testing data. Results shown in Table 2 are an average over 100 trials, where each trial represents a random partition of training and testing data. Since data is truly missing, no features are artificially removed.

Table 2. The mean AUC of 100 trials of each method for the 2-View Land Mine Detection data sets.

Data Set	Proposed Method	Conditional Mean Imputation	Unconditional Mean Imputation
Area A	0.6865	0.5604	0.6305
Area B	0.6579	0.5355	0.6171

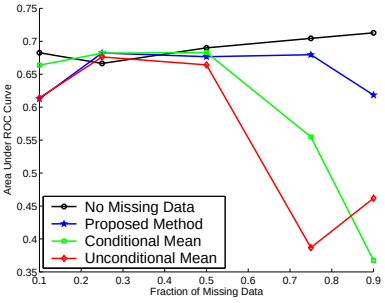

Figure 2. Experimental results in terms of AUC for the 4-View Land Mine Detection data set.

3.3. 4-View Land Mine Detection

The proposed algorithm was also applied to a real data set of 4-view land mine detection data. The goal for this data set is to again classify mines and clutter. The four views were a ground-penetrating radar (GPR) sensor, an EOIR sensor, a Ku-band SAR sensor, and an X-band SAR sensor. The sensors were characterized by 17, 6, 9, and 9 features, respectively. The data set had 713 total data points, only 91 of which were mines.

Unlike the 2-view data sets, every data point had data from each of the four views. Therefore, for the experiments, views (*i.e.,* sensors, or blocks of features) were randomly chosen to be artificially removed and thereafter treated as missing.

For the experiments, 25% of the data was used as training data, while the remainder was used as testing data. Each point in Figure 2 is an average over 10 trials, where each trial represents a random partition of training and testing data, and a random pattern of missing sensors (blocks of features).

From Figure 2, it can be seen that the proposed method again outperforms the imputation methods, with the most significant difference occurring when higher fractions of data are missing. The performance of the algorithm in which there is no missing data varies with the fraction of missing data because of the random partitions of training and testing data.

4. Conclusion

We have derived the expression for a Gaussian kernel function (or matrix) when faced with incomplete data. We analytically integrated out the missing data to obtain a closed-form expression for the kernel. As a result, incomplete data need no longer be a hindrance for general multi-view algorithms. We have demonstrated the superiority of this proposed method over two common imputation schemes, on both a benchmark data set as well as on three real multi-view land mine data sets. The advantage of the proposed method has been found to be most pronounced when a large amount of data is missing. The feature vectors for the multi-view land mine data sets will be made available to interested investigators upon request.

Analytical integration over the missing data can still be performed if one employs a linear or polynomial kernel instead of a Gaussian kernel, so our choice here of a Gaussian kernel is not overly restrictive. This kernel matrix completion work can also be utilized in semi-supervised algorithms. Many semi-supervised algorithms use the idea of a graph and the graph Laplacian (Zhu, Ghahramani & Lafferty, 2003), which can be directly computed from a kernel matrix. Future work will use our kernel matrix completion method to extend supervised algorithms to semi-supervised versions, when faced with incomplete data.

References

Ghahramani, Z. & Jordan, M. (1994). Supervised learning from incomplete data via the EM approach. In J. Cowan and G. Tesauro and J. Alspector (Eds.), *Advances in Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann.

Graepel, T. (2002). Kernel matrix completion by semidefinite programming. *Proceedings of the International Conference on Artificial Neural Networks* (pp. 694–699).

Hanley, J. & McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology 143*, 29–36.

Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., & Noble, W. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. *Proceedings of the Pacific Symposium on Biocomputing 9* (pp. 300–311).

Rässler, S. (2004). *The impact of multiple imputation for DACSEIS* (DACSEIS Research Paper Series 5). University of Erlangen-Nürnberg, Nürnberg, Germany.

Schölkopf, B. & Smola, A. (2002). *Learning with kernels*. Cambridge: MIT Press.

Tsuda, K., Akaho, S., & Asai, K. (2003). The *em* algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research 4*, 67–81.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proceedings of the Twentieth International Conference on Machine Learning*.