Bayesian Inference for Transductive Learning of Kernel Matrix Using the Tanner-Wong Data Augmentation Algorithm

Zhihua Zhang ZHZHANG@CS.UST.HK
Dit-Yan Yeung DYYEUNG@CS.UST.HK
James T. Kwok JAMESK@CS.UST.HK
Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Abstract

In kernel methods, an interesting recent development seeks to learn a good kernel from empirical data automatically. In this paper, by regarding the transductive learning of the kernel matrix as a missing data problem, we propose a Bayesian hierarchical model for the problem and devise the Tanner-Wong data augmentation algorithm for making inference on the model. The Tanner-Wong algorithm is closely related to Gibbs sampling, and it also bears a strong resemblance to the expectation-maximization (EM) algorithm. For an efficient implementation, we propose a simplified Bayesian hierarchical model and the corresponding Tanner-Wong algorithm. We express the relationship between the kernel on the input space and the kernel on the output space as a symmetric-definite generalized eigenproblem. Based on this eigenproblem, an efficient approach to choosing the base kernel matrices is presented. The effectiveness of our Bayesian model with the Tanner-Wong algorithm is demonstrated through some classification experiments showing promising results.

1. Introduction

In recent years, kernel methods have rapidly gained much popularity due to their flexibility and theoretical elegance. For most kernel-based learning methods in existence, the practitioner has to prespecify a kernel function in advance before learning proceeds. Choosing a good kernel for the problem at hand is more of an art than a science. Since a kernel induces a feature space, it is easy to understand that an appropriate kernel choice should take into account the empirical data available for a learning problem. Since in practice we often deal with finite-sized data sets, almost all information in the kernel function can be encoded by the kernel matrix. As a result, one could bypass the learning of the kernel function by just learning the kernel matrix instead. For simplicity, from now on, we do not make any distinction between learning the kernel function and learning the kernel matrix.

Some recent studies pursue to learn the kernel matrix from empirical data automatically. One major issue to consider is what constitutes a good kernel matrix. More specifically, we need a criterion for optimizing the kernel matrix. The *alignment* was proposed as such a criterion defined in the form of a similarity measure between kernel matrices (Cristianini et al., 2002). Based on this criterion, several methods have been proposed for optimizing the kernel matrix, including a spectral method (Cristianini et al., 2002), semi-definite programming (SDP) (Lanckriet et al., 2002), the Gram-Schmidt method (Kandola et al., 2002), and a gradient-based method (Bousquet & Herrmann, 2003). Crammer et al. (2003) cast the kernel matrix learning problem under the boosting paradigm for constructing an accurate kernel from simple base kernels. In the case that there are missing data in the kernel matrix, Tsuda et al. (2003) developed a parametric approach to kernel matrix completion using the *em* algorithm based on information geometry (Amari, 1995). In their approach, the Kullback-Leibler (KL) divergence is used for measuring the similarity between kernel matrices.

Assuming that the kernel matrix is a random positive definite matrix following the Wishart distribution (Gupta & Nagar, 2000), Zhang et al. (2003b) first pro-

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

posed a generative model of the kernel matrix. In the transductive setting, Zhang et al. (2003a) presented a Bayesian hierarchical model and showed that the expectation-maximization (EM) algorithm (Dempster et al., 1977) can be used to learn the kernel matrix through *maximum a posteriori* (MAP) estimation or, more generally, maximum penalized likelihood estimation. In particular, given the kernel matrix on the training data, the EM algorithm is used to alternately infer the kernel matrix on the test data and the kernel matrix relating the training data to the test data, as well as the parameter matrix of the Wishart distribution for the kernel matrix.

In this paper, based on the Bayesian hierarchical model proposed by Zhang et al. (2003a), we use the Tanner-Wong data augmentation algorithm (Tanner & Wong, 1987) for Bayesian inference to solve the kernel matrix learning problem under the transductive setting. The Tanner-Wong algorithm is closely related to Gibbs sampling which is a type of Markov chain Monte Carlo (MCMC) method. Like the EM algorithm, the Tanner-Wong algorithm has been widely used in statistical inference for incomplete data problems (Schafer, 1997). Moreover, it also strongly resembles the EM algorithm. The Tanner-Wong algorithm consists of the *Imputation step* (I-step) and the *Posterior step* (P-step). The I-step simulates a random draw of some complete-data sufficient statistics, whereas the E-step of EM computes the expectation of the complete-data sufficient statistics. Typically, the implementation of the I-step is very similar to that of the E-step. On the other hand, the P-step of the Tanner-Wong algorithm is a random draw from some complete-data posterior, while the M-step of EM performs maximization of the complete-data likelihood. In other words, both the I-step and the E-step are used to estimate the values of the missing data, while the P-step and the M-step are both used to estimate the unknown values of the model parameters. Our results in Section 3 further demonstrate this strong resemblance between the Tanner-Wong algorithm and the EM algorithm.

For our kernel matrix learning problem, the computational requirements would be very high if the Tanner-Wong algorithm works on matrix variate distributions. To make our method feasible in practice, we present in this paper a simplified Bayesian hierarchical model so that the Tanner-Wong algorithm can work on distributions over random variables or random vectors (instead of random matrices). This is motivated by some existing kernel matrix learning methods (Crammer et al., 2003; Cristianini et al., 2002; Lanckriet et al., 2002; Tsuda et al., 2003), which constrain the target kernel matrix to a weighted combination of some fixed base kernel matrices so that the kernel matrix learning problem can be simplified to the estimation of the weighting coefficients. Apparently, the performance of these methods depends critically on the choice of the base kernel matrices. However, very little has been addressed on how to prespecify the base kernel matrices. Usually, they are obtained from the eigenvectors of an empirical kernel matrix on the input space. In this paper, by a symmetric-definite generalized eigenproblem we associate the kernel matrix on the input space and the kernel matrix on the output space. Based on this eigenproblem, we present an efficient approach to choosing the base kernel matrices by exploiting information not just from the input kernel matrix but also from the partial output kernel on the training set.

The rest of this paper is organized as follows. Sections 2 presents a basic Bayesian hierarchical model for the kernel matrix learning problem. In Section 3, we devise a general Tanner-Wong data augmentation algorithm for making inference on the basic model. Section 4 gives a simplified Bayesian hierarchical model and the corresponding Tanner-Wong algorithm. We also present an efficient method for choosing the base kernel matrices. Section 5 presents the experimental results and the last section gives some concluding remarks.

2. Basic Bayesian Hierarchical Model

We consider the kernel matrix learning problem for classification in a transductive setting. Let the training set and test set be $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_1}, y_{n_1})\}$ and $\tilde{\mathcal{T}} = \{(\mathbf{x}_{n_1+1}, y_{n_1+1}), \ldots, (\mathbf{x}_{n_1+n_2}, y_{n_1+n_2})\}$, respectively, where $\mathbf{x}_i \in \mathbb{R}^q$ for $i = 1, \ldots, n_1 + n_2$, $y_i \in \{1, 2, \ldots, c\}$ for $i = 1, \ldots, n_1$ and y_i's are unknown for $i = n_1+1, \ldots, n_1+n_2$. Letting $n = n_1+n_2$, we refer to $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}, \mathbf{x}_{n_1+1}, \ldots, \mathbf{x}_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_{n_1}, y_{n_1+1}, \ldots, y_n\}$ as the input set and output set, respectively. We define a kernel matrix $\mathbf{K}$ on $(\mathcal{T} \cup \tilde{\mathcal{T}}) \times (\mathcal{T} \cup \tilde{\mathcal{T}})$ and partition it as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}, \tag{1}$$

where $n_1 \times n_1$ and $n_2 \times n_2$ matrices $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ are defined on the training and test sets, respectively, and $n_2 \times n_1$ matrix $\mathbf{K}_{21}$ $(= \mathbf{K}_{12}')$ characterizes the similarity between the training and test data.

In general, definition of the kernel matrix $\mathbf{K}$ depends on the problem considered and the prior knowledge available. Let $k_I : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_O : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denote the input kernel and output kernel, respectively. We define $\mathbf{A}$ as the *input kernel matrix* for $\mathcal{X}$ using input kernel k_I and $\mathbf{B}$ as the *output kernel matrix*

for $\mathcal{Y}$ using output kernel k_O. Augmenting any input vector $\mathbf{x}$ with the corresponding output y to form a vector $\mathbf{z} = (\mathbf{x}, y)$, we define a kernel matrix $\mathbf{K}$ on $(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$ in this paper. Specifically, in our experiments to be presented later, we define the kernel matrix $\mathbf{K} = (\mathbf{A} + \mathbf{B})/2$ as the discriminant kernel (Zhang, 2003), where $\mathbf{A} = \left[\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\beta)\right]_{n\times n}$ is the standard Gaussian kernel and $\mathbf{B}$ is the *ideal kernel* (Cristianini et al., 2002) based on the training set, i.e.,

$$[\mathbf{B}]_{ij} = \begin{cases} 1 & y_i = y_j \\ 0 & y_i \neq y_j. \end{cases}$$

Following the generative model formulation of the kernel matrix in (Zhang et al., 2003a), we now assume that the kernel matrix $\mathbf{K}$ is distributed according to a Wishart distribution $\mathcal{W}_n(r, \mathbf{\Sigma})$ (Gupta & Nagar, 2000), as

$$p(\mathbf{K} \mid \mathbf{\Sigma}, r) = \frac{1}{C(n,r)} |\mathbf{\Sigma}|^{-r/2} |\mathbf{K}|^{(r-n-1)/2} \exp\left[-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{K})\right].$$

Here $\mathbf{\Sigma} \succ 0$ is an $n\times n$ positive definite parameter matrix,[1] $r \geq n$ is the degree of freedom, and $C(n,r) = 2^{rn/2}\pi^{n(n-1)/4}\prod_{j=1}^{n}\Gamma(\frac{r+1-j}{2})$ is a normalization term where $\Gamma(\cdot)$ is the Gamma function.

The parameter matrix $\mathbf{\Sigma}$ is left completely unspecified in the model. However, its uncertainty is influenced by a higher-level prior distribution. In particular, we use a conjugate prior on $\mathbf{\Sigma}$, i.e., $\mathbf{\Sigma}$ is distributed according to the inverted Wishart distribution $\mathcal{I}W_n(\eta, \mathbf{\Theta})$ (Gupta & Nagar, 2000), as

$$p(\mathbf{\Sigma} \mid \mathbf{\Theta}, \eta) = \frac{1}{C(n,\eta)} |\mathbf{\Theta}|^{\eta/2} |\mathbf{\Sigma}|^{-(\eta+n+1)/2} \exp\left[-\frac{1}{2}\mathrm{tr}(\mathbf{\Theta}\mathbf{\Sigma}^{-1})\right],$$

where $\mathbf{\Theta} \succ 0$ is an $n\times n$ hyperparameter matrix. It also follows that $\mathbf{D} = \mathbf{\Sigma}^{-1}$ is distributed according to $\mathcal{W}_n(\eta, \mathbf{\Theta}^{-1})$. Moreover, the conditional distribution of $\mathbf{D}$ on $\mathbf{K}$ is $\mathcal{W}_n(r+\eta, (\mathbf{K}+\mathbf{\Theta})^{-1})$ (Gupta & Nagar, 2000).

As for $\mathbf{\Sigma}$, we could again define $\mathbf{\Theta}$ as a random matrix and then incorporate another higher-level prior. However, for simplicity, $\mathbf{\Theta}$ is held fixed in this paper. In particular, we use the Gaussian kernel on the input set $\mathcal{X}$ to define $\mathbf{\Theta}$. In general, other kernels, such as the commonly used polynomial kernel, Laplacian kernel and linear kernel (defined on $\mathcal{X}$), may also be used. Another possibility is to define $\mathbf{\Theta}$ as a weighted combination of different kernel matrices. Moreover, for simplicity, r and η are also held fixed in this paper. Our model is thus a hierarchical model with three levels. The first level corresponds to a random Wishart matrix, the second level corresponds to the parameter matrix of the Wishart matrix, and the third level corresponds to the hyperparameter matrix of the parameter matrix.

The observed data set provides a particular realization of $\mathbf{K}$. With an abuse of notation, we will denote this realization again by $\mathbf{K}$. Note that $\mathbf{K}$ represents the partially observed kernel matrix, since only the $\mathbf{K}_{11}$ part of $\mathbf{K}$ in (1) is available from the observed data, while both $\mathbf{K}_{21}$ (and hence $\mathbf{K}_{12}$) and $\mathbf{K}_{22}$ are missing. In other words, if we consider this as a missing data problem, then the incomplete (observable) data is $\mathbf{K}_{11}$, the complete data is $(\mathbf{K}_{11}, \mathbf{K}_{21}, \mathbf{K}_{22})$, and the goal is to infer the missing data ($\mathbf{K}_{21}$ and $\mathbf{K}_{22}$) and the unknown model parameters ($\mathbf{\Sigma}$, or equivalently $\mathbf{D} = \mathbf{\Sigma}^{-1}$). Consider that $\mathbf{K} \succ 0$ if and only if $\mathbf{K}_{11} \succ 0$ and $\mathbf{K}_{22\cdot1} \succ 0$ (Horn & Johnson, 1985), where $\mathbf{K}_{22\cdot1} = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}$ is the Schur complement of $\mathbf{K}_{11}$. We take $\{\mathbf{K}_{11}, \mathbf{K}_{21}, \mathbf{K}_{22\cdot1}\}$ instead as the complete data to ensure that $\mathbf{K}$ is always positive definite. Thus, the likelihood function of the complete data is

$$p(\mathbf{K}_{11}, \mathbf{K}_{21}, \mathbf{K}_{22\cdot1} \mid \mathbf{D}) = p(\mathbf{K}_{11} \mid \mathbf{D})\, p(\mathbf{K}_{22\cdot1} \mid \mathbf{D})\, p(\mathbf{K}_{21} \mid \mathbf{K}_{11}, \mathbf{D}).$$

While Zhang et al. (2003a) proposed an EM algorithm to solve this missing data problem, we propose using the Tanner-Wong data augmentation algorithm (Tanner & Wong, 1987) as an alternative method for solving the same problem.

3. Tanner-Wong Algorithm for Kernel Matrix Learning

In general, for the complete data $Y = (Y_{obs}, Y_{mis})$ with unknown parameter θ, the Tanner-Wong data augmentation starts from an initial parameter estimate $\theta^{(0)}$ and then repeats the following two steps in an alternating manner:

I-step (Imputation): Given the current $\theta^{(t)}$, sample a value of the missing data Y_{mis} from its conditional distribution,

$$Y_{mis}^{(t+1)} \sim p(Y_{mis} \mid Y_{obs}, \theta^{(t)}). \tag{2}$$

P-step (Posterior): Conditioning on $Y_{mis}^{(t+1)}$, sample a new value of the unknown parameter

[1] In this paper, $\mathbf{A} \succ 0$ means that $\mathbf{A}$ is positive definite and $\mathbf{A} \succeq 0$ means that it is positive semi-definite.

θ from its complete-data posterior distribution,

$$\theta^{(t+1)} \sim p(\theta \mid Y_{obs}, Y_{mis}^{(t+1)}). \tag{3}$$

It can be shown that this iterative procedure yields a stochastic sequence $\{(\theta^{(t)}, Y_{mis}^{(t)}) : t = 1, 2, \ldots\}$ with stationary distribution $p(\theta, Y_{mis} \mid Y_{obs})$. Furthermore, the stationary distributions of the subsequences $\{\theta^{(t)} : t = 1, 2, \ldots\}$ and $\{Y_{mis}^{(t)} : t = 1, 2, \ldots\}$ are $p(\theta \mid Y_{obs})$ and $p(Y_{mis} \mid Y_{obs})$, respectively. Obviously, data augmentation is a special case of Gibbs sampling with (Y_{mis}, θ) partitioned into Y_{mis} and θ. It is also closely related to the EM algorithm, where the I-step corresponds to the E-step and the P-step corresponds to the M-step.

For our hierarchical model defined in the previous section, we have $Y = \mathbf{K}$ with $Y_{obs} = \mathbf{K}_{11}$ and $Y_{mis} = (\mathbf{K}_{21}, \mathbf{K}_{22\cdot1})$, and $\theta = \mathbf{D}$. Thus, Bayesian inference is based on the joint density of all variables, i.e.,

$$\begin{aligned} &p(\mathbf{K}_{11}, \mathbf{K}_{21}, \mathbf{K}_{22\cdot1}, \mathbf{D}) = \\ &\quad p(\mathbf{K}_{11} \mid \mathbf{D})\, p(\mathbf{K}_{22\cdot1} \mid \mathbf{D})\, p(\mathbf{K}_{21} \mid \mathbf{K}_{11}, \mathbf{D})\, p(\mathbf{D}). \end{aligned} \tag{4}$$

As for $\mathbf{K}$ in (1), $\mathbf{\Sigma}$, $\mathbf{D}$ and $\mathbf{\Theta}$ are similarly partitioned as

$$\begin{aligned} \mathbf{\Sigma} &= \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix}, \\ \mathbf{D} &= \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix}, \\ \mathbf{\Theta} &= \begin{bmatrix} \mathbf{\Theta}_{11} & \mathbf{\Theta}_{12} \\ \mathbf{\Theta}_{21} & \mathbf{\Theta}_{22} \end{bmatrix}. \end{aligned}$$

From (Zhang et al., 2003a) or (Gupta & Nagar, 2000), we have

$$\begin{aligned} \mathbf{K}_{11} &\sim \mathcal{W}_{n_1}(r, \mathbf{\Sigma}_{11}), \\ \mathbf{K}_{22} &\sim \mathcal{W}_{n_2}(r, \mathbf{\Sigma}_{22}), \\ \mathbf{K}_{22\cdot1} &\sim \mathcal{W}_{n_2}(r - n_1, \mathbf{\Sigma}_{22\cdot1}), \\ \mathbf{K}_{21} \mid \mathbf{K}_{11} &\sim \mathcal{N}(\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{K}_{11}, \mathbf{\Sigma}_{22\cdot1} \otimes \mathbf{K}_{11}). \end{aligned}$$

It is easy to see that $\mathbf{D}_{11} = \mathbf{\Sigma}_{11\cdot2}^{-1}$, $\mathbf{D}_{22} = \mathbf{\Sigma}_{22\cdot1}^{-1}$ and $\mathbf{D}_{22}^{-1}\mathbf{D}_{21} = -\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}$. Then, given the current parameter estimate $\mathbf{D}^{(t)}$, the I-step simulates $\mathbf{K}_{21}$ and $\mathbf{K}_{22\cdot1}$ by drawing from the following distributions:

$$\begin{aligned} \mathbf{K}_{21}^{(t)} \mid \cdots &\sim \mathcal{N}\big(-(\mathbf{D}_{22}^{(t)})^{-1}\mathbf{D}_{21}^{(t)}\mathbf{K}_{11}, (\mathbf{D}_{22}^{(t)})^{-1} \otimes \mathbf{K}_{11}\big), \\ \mathbf{K}_{22\cdot1}^{(t)} \mid \cdots &\sim \mathcal{W}_{n_2}\big(r - n_1, (\mathbf{D}_{22}^{(t)})^{-1}\big), \end{aligned} \tag{5}$$

and the P-step draws the parameter $\mathbf{D}$ from the following distribution:

$$\mathbf{D}^{(t+1)} \mid \cdots \sim \mathcal{W}_n\big(r + \eta, \mathbf{Q}^{(t)}\big). \tag{6}$$

Here $\mathbf{Q}^{(t)} = (\mathbf{K}^{(t)} + \mathbf{\Theta})^{-1}$ and it is partitioned into $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}$ as for $\mathbf{K}$, and "$\mid \cdots$" means conditioning on all other variables. The P-step in (6) can also be expressed as the following equivalent form:

$$\begin{aligned} \mathbf{D}_{11\cdot2}^{(t+1)} \mid \cdots &\sim \mathcal{W}_{n_1}\big(r + \eta - n_2, \mathbf{Q}_{11\cdot2}^{(t)}\big), \\ \mathbf{D}_{22}^{(t+1)} \mid \cdots &\sim \mathcal{W}_{n_2}\big(r + \eta, \mathbf{Q}_{22}^{(t)}\big), \\ \mathbf{D}_{21}^{(t+1)} \mid \cdots &\sim \mathcal{N}\big(\mathbf{D}_{22}^{(t+1)}(\mathbf{Q}_{22}^{(t)})^{-1}\mathbf{Q}_{21}^{(t)}, \mathbf{D}_{22}^{(t+1)} \otimes \mathbf{Q}_{11\cdot2}^{(t)}\big). \end{aligned}$$

From the I-step in (5), we can see that only the values of $\mathbf{D}_{22}^{-1}\mathbf{D}_{21}$ and $\mathbf{D}_{22}^{-1}$ are required. Since our ultimate goal is to estimate the missing data $\mathbf{K}_{21}$ and $\mathbf{K}_{22\cdot1}$ but not $\mathbf{D}$, this motivates us to directly draw $\mathbf{D}_{22}^{-1}\mathbf{D}_{21}$ and $\mathbf{D}_{22}^{-1}$, instead of $\mathbf{D}_{11\cdot2}$, $\mathbf{D}_{21}$ and $\mathbf{D}_{22}$, in the P-step. Note that $\mathbf{D}_{22}^{-1}\mathbf{D}_{21}$ is a regression matrix. From results on the distribution of the regression matrix (see (Gupta & Nagar, 2000)), we can obtain an alternative P-step as

$$\begin{aligned} (\mathbf{D}_{22}^{-1}\mathbf{D}_{21})^{(t+1)} \mid \cdots &\sim \mathcal{T}\big(s, (\mathbf{Q}_{22}^{(t)})^{-1}\mathbf{Q}_{21}^{(t)}, (\mathbf{Q}_{22}^{(t)})^{-1}, \mathbf{Q}_{11\cdot2}^{(t)}\big), \\ (\mathbf{D}_{22}^{-1})^{(t+1)} \mid \cdots &\sim \mathcal{I}W_{n_2}\big(r + \eta, (\mathbf{Q}_{22}^{(t)})^{-1}\big), \end{aligned}$$

where $s = r + \eta - n_2 + 1$ and $\mathbf{T} \sim \mathcal{T}\big(m, \mathbf{M}, \mathbf{X}, \mathbf{Y}\big)$ is a $p \times q$ random matrix with matrix variate t-distribution (Gupta & Nagar, 2000) as

$$\begin{aligned} &p(\mathbf{T} \mid m, \mathbf{M}, \mathbf{X}, \mathbf{Y}) = \\ &\quad \frac{\Gamma_p[\frac{1}{2}(m + p + q - 1)]}{\pi^{\frac{1}{2}pq}\Gamma_p[\frac{1}{2}(m + p - 1)]} |\mathbf{X}|^{-\frac{1}{2}q} |\mathbf{Y}|^{-\frac{1}{2}p}\, \cdot \\ &\quad \big|\mathbf{I} + \mathbf{X}^{-1}(\mathbf{T} - \mathbf{M})\mathbf{Y}^{-1}(\mathbf{T} - \mathbf{M})'\big|^{-\frac{1}{2}(m+p+q-1)}\, . \end{aligned}$$

As $\mathbf{Q}_{11\cdot2}^{(t)} = (\mathbf{K}_{11} + \mathbf{\Theta}_{11})^{-1}$, $(\mathbf{Q}_{22}^{(t)})^{-1}\mathbf{Q}_{21}^{(t)} = -(\mathbf{K}_{21}^{(t)} + \mathbf{\Theta}_{21})(\mathbf{K}_{11} + \mathbf{\Theta}_{11})^{-1}$ and $(\mathbf{Q}_{22}^{(t)})^{-1} = \mathbf{K}_{22\cdot1}^{(t)} + \mathbf{K}_{21}^{(t)}\mathbf{K}_{11}^{-1}(\mathbf{K}_{21}^{(t)})' + \mathbf{\Theta}_{22} - (\mathbf{K}_{21}^{(t)} + \mathbf{\Theta}_{21})(\mathbf{K}_{11} + \mathbf{\Theta}_{11})^{-1}(\mathbf{K}_{21}^{(t)} + \mathbf{\Theta}_{21})'$, our data augmentation algorithm has a strong resemblance to the EM algorithm presented in (Zhang et al., 2003a).

4. A Simplified Bayesian Model for Kernel Matrix Learning

A major practical problem with the method presented in the previous section is its high computational requirements, since the Tanner-Wong algorithm has to work on matrix variate distributions. Moreover, to the best of our knowledge, it is intractable to draw a matrix from a matrix-variate normal distribution or a matrix-variate t-distribution, although it is tractable to draw a matrix from a Wishart distribution. Therefore, it is necessary to seek an effective and tractable

implementation of the data augmentation algorithm. In this section, we propose a simplified Bayesian model that makes it possible to develop an efficient implementation.

In the kernel matrix learning literature (Crammer et al., 2003; Cristianini et al., 2002; Lanckriet et al., 2002; Tsuda et al., 2003), it is common to constrain the target kernel to a weighted combination of some available base kernels so that the learning problem is simplified to the estimation of the weighting coefficients. In particular, the target kernel matrix $\mathbf{G}$ is expressed as $\mathbf{G} = \sum_{i=1}^{n} \lambda_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i'$ with $\lambda_i > 0$. Let $\mathbf{U}' = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n]$ and $\mathbf{L} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Then $\mathbf{G} = \mathbf{U}'\mathbf{L}\mathbf{U}$. We refer to $\mathbf{U}$ and $\boldsymbol{\mu}_i\boldsymbol{\mu}_i'$'s as the base matrix and the base kernel matrices, respectively. Now, given $\mathbf{U}$, we want to estimate λ_i's and hence $\mathbf{G}$ to approximate some desired kernel $\mathbf{H}$, such as the ideal kernel (Cristianini et al., 2002), based on some criterion like the kernel alignment or the KL divergence between $\mathbf{G}$ and $\mathbf{H}$. This motivates us to devise a simplified Bayesian hierarchical model and then an efficient implementation of the Tanner-Wong algorithm.

4.1. A Simplified Bayesian Hierarchical Model

We know that if $\mathbf{X} \sim \mathcal{W}_n(r, \mathbf{Y})$ and there exists a non-singular matrix $\mathbf{C}$ such that $\mathbf{C}'\mathbf{Y}\mathbf{C} = \boldsymbol{\Delta}$ where $\boldsymbol{\Delta}$ is diagonal, then $\mathbf{C}'\mathbf{X}\mathbf{C} \sim \mathcal{W}_n(r, \boldsymbol{\Delta})$ (Gupta & Nagar, 2000). We apply this property to our basic model presented in Section 2. Assume that we are given a non-singular matrix $\mathbf{C}$ such that $\mathbf{C}'\boldsymbol{\Sigma}\mathbf{C}$ is diagonal and we define $\mathbf{W} = \mathbf{C}'\mathbf{K}\mathbf{C}$, then $\mathbf{W}$ is distributed according to the Wishart distribution with a diagonal parameter matrix $\boldsymbol{\Delta} = \mathrm{diag}(\delta_1, \ldots, \delta_n) \succ 0$. In particular, we give a model as

$$\mathbf{W} \sim \mathcal{W}_n(r, \boldsymbol{\Delta}). \tag{7}$$

Note that a Wishart matrix with a diagonal parameter matrix is not diagonal. So our simplified model is feasible and the kernel matrix learning problem becomes estimating the matrices $\mathbf{W}_{21}$ and $\mathbf{W}_{22}$.

In a Bayesian hierarchical framework, $\boldsymbol{\Delta}$ can have its own prior distribution. Here, we assume that the δ_j's are *a priori* conditionally independent given some hyperparameters. In particular, we use the inverted Gamma distribution

$$\delta_j \sim \mathcal{I}G\left(\eta, \lambda^{-1}\right) \tag{8}$$

as a conjugate prior for the δ_j's. Thus, $\delta_j^{-1} \sim \mathcal{G}(\eta, \lambda)$, i.e., it follows a Gamma distribution. In this paper, we fix the hyperparameter η and use

$$\lambda \sim \mathcal{G}(\zeta, \tau) \tag{9}$$

as the prior for λ.

Our goal is to estimate both the missing data $\mathbf{W}_{21}$ and $\mathbf{W}_{22}$ and the unknown parameter $\boldsymbol{\Delta}$. In order to ensure that $\mathbf{W} \succ 0$, we treat $(\mathbf{W}_{11}, \mathbf{W}_{21}, \mathbf{W}_{22\cdot1})$ instead of $(\mathbf{W}_{11}, \mathbf{W}_{21}, \mathbf{W}_{22})$ as the complete data. Our problem then becomes estimating the missing data $\mathbf{W}_{21}$ and $\mathbf{W}_{22\cdot1}$ and the parameter $\boldsymbol{\Delta}$ from the observed data $\mathbf{W}_{11}$. Bayesian inference is based on the joint density of all variables, i.e.,

$$\begin{aligned} &p(\mathbf{W}_{11}, \mathbf{W}_{21}, \mathbf{W}_{22\cdot1}, \delta_1^{-1}, \ldots, \delta_n^{-1}, \lambda) = \\ &\quad p(\mathbf{W}_{11} \mid \boldsymbol{\Delta})\, p(\mathbf{W}_{21}, \mathbf{W}_{22\cdot1} \mid \boldsymbol{\Delta}, \mathbf{W}_{11}) \cdot \\ &\quad p(\lambda) \prod_{i=1}^{n} p(\delta_i^{-1} \mid \lambda). \end{aligned} \tag{10}$$

The following theorem (Gupta & Nagar, 2000) will be very useful in the sequel.

Theorem 1 *Suppose* $\mathbf{W} \sim \mathcal{W}_n(r, \boldsymbol{\Delta})$ *with* $\mathbf{W}, \boldsymbol{\Delta} \succ 0$ *partitioned as*

$$\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix}, \qquad \begin{bmatrix} \boldsymbol{\Delta}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_2 \end{bmatrix},$$

where $\mathbf{W}_{11}$ *and* $\boldsymbol{\Delta}_1$ *are both of size* $n_1 \times n_1$. *Then*

(i) $\mathbf{W}_{11} \sim \mathcal{W}_{n_1}(r, \boldsymbol{\Delta}_1)$ *and* $\mathbf{W}_{22} \sim \mathcal{W}_{n_2}(r, \boldsymbol{\Delta}_2)$;

(ii) $\mathbf{W}_{21} | \mathbf{W}_{11} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Delta}_2 \otimes \mathbf{W}_{11})$; *and*

(iii) $\mathbf{W}_{22\cdot1} \sim \mathcal{W}_{n_2}(r - n_1, \boldsymbol{\Delta}_2)$ *and is independent of* $\mathbf{W}_{21}$ *and* $\mathbf{W}_{11}$.

Using Theorem 1(iii), (10) can thus be rewritten as

$$\begin{aligned} &p(\mathbf{W}_{11}, \mathbf{W}_{21}, \mathbf{W}_{22\cdot1}, \delta_1^{-1}, \ldots, \delta_n^{-1}, \lambda) = \\ &\quad p(\mathbf{W}_{11} \mid \boldsymbol{\Delta}_1)\, p(\mathbf{W}_{21} \mid \boldsymbol{\Delta}_2, \mathbf{W}_{11})\, p(\mathbf{W}_{22\cdot1} \mid \boldsymbol{\Delta}_2) \cdot \\ &\quad p(\lambda) \prod_{i=1}^{n} p(\delta_i^{-1} \mid \lambda), \end{aligned} \tag{11}$$

leading to a simplified hierarchical model.

4.2. Tanner-Wong Algorithm for the Simplified Model

In this subsection, we apply the Tanner-Wong algorithm to carry out Bayesian inference on the simplified hierarchical model in (11). That is, we simulate the missing data $\mathbf{W}_{21}$ and $\mathbf{W}_{22\cdot1}$ and the parameter $\boldsymbol{\Delta}$ iteratively by (2) and (3). Given the current parameter estimate $\boldsymbol{\Delta}^{(t)} = \mathrm{diag}(\delta_1^{(t)}, \ldots, \delta_n^{(t)})$, the I-step simulates $\mathbf{W}_{21}$ and $\mathbf{W}_{22\cdot1}$ by drawing from the following distributions:

$$\mathbf{W}_{21}^{(t)} \mid \cdots \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Delta}_2^{(t)} \otimes \mathbf{W}_{11}), \tag{12}$$

$$\mathbf{W}_{22\cdot1}^{(t)} \mid \cdots \sim \mathcal{W}_{n_2}(r - n_1, \boldsymbol{\Delta}_2^{(t)}), \tag{13}$$

and the P-step draws the parameters δ_j's and hyperparameter λ from the following distributions:

$$\lambda^{(t+1)} \mid \cdots \sim \mathcal{G}\Big(n\eta + \zeta, \tau + \sum_{i=1}^{n} \big(\delta_i^{-1}\big)^{(t)}\Big),$$
$$\big(\delta_j^{-1}\big)^{(t+1)} \mid \cdots \sim \mathcal{G}\Big(\eta + r/2, \lambda^{(t+1)} + w_j^{(t)}/2\Big).$$

Here $w_j^{(t)}$ is the (j,j)th element of $\mathbf{W}^{(t)}$. The I-step can be easily obtained from Theorem 1(ii) and (iii), and the derivation of the P-step is based on the full condition. Let $\mathbf{W}_{21}' = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_2}]$ where $\mathbf{b}_j$ is an n_1-dimensional vector. Then the I-step (12) is equivalent to one that separately simulates $\mathbf{b}_j$ for $j = 1, \ldots, n_2$, instead of $\mathbf{W}_{21}$, by

$$\mathbf{b}_j^{(t)} \mid \cdots \sim \mathcal{N}(\mathbf{0}, \delta_{n_1+j}^{(t)} \mathbf{W}_{11}). \tag{14}$$

Our revised Tanner-Wong data augmentation algorithm is tractable because it involves only random draws of the Gamma variable, the Gaussian vector, and the Wishart matrix.

We can see that the computational cost of the current algorithm is dominated by the I-step (13) for $\mathbf{W}_{22\cdot1}$. Here we give a further predigestion. In particular, we approximate $\mathbf{W}_{22\cdot1}$ with a diagonal matrix $\mathrm{diag}(\beta_1, \beta_2, \ldots, \beta_{n_2})$. Then the I-step (13) reduces to

$$\beta_j^{(t)} \mid \cdots \sim \mathcal{G}\Big((r - n_1)/2, \big(\delta_{j+n_1}^{-1}\big)^{(t)}/2\Big), \tag{15}$$

for $j = 1, \ldots, n_2$.

4.3. Generalized Eigenproblem for Choosing the Base Kernel Matrices

Having formulated a simplified hierarchical model and devised a tractable Tanner-Wong algorithm, we now come to the question of how to choose the nonsingular matrix $\mathbf{C}$. Note that $\mathbf{C}$ plays the same role as the base matrix $\mathbf{U}$ mentioned earlier in this section. In most existing kernel matrix learning methods, a widely used approach is to set $\mathbf{U}$ to be the eigenvector matrix of an empirical input kernel matrix. For our model, since the hyperparameter matrix $\mathbf{\Theta}$ is *a priori* specified, we can use the matrix consisting of the eigenvectors of $\mathbf{\Theta}$ as a base matrix. In this subsection, by formulating a symmetric-definite generalized eigenproblem, we present an efficient approach to specifying the base kernel matrices.

Like $\mathbf{K}$, we partition the input kernel matrix $\mathbf{A}$ and output kernel matrix $\mathbf{B}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}.$$

Obviously, $\mathbf{A}$ is known. For $\mathbf{B}$, however, $\mathbf{B}_{11}$ is known but $\mathbf{B}_{12}$ and $\mathbf{B}_{22}$ are both unknown. Therefore, our problem is to estimate $\mathbf{B}_{12}$ and $\mathbf{B}_{22}$ from $\mathbf{A}$ and $\mathbf{B}_{11}$.

We tackle this problem by formulating a *symmetric-definite generalized eigenproblem* (Golub & Loan, 1996). For the kernel matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$, we consider a generalized eigenvalue system of the form

$$\mid \mathbf{A} - \lambda \mathbf{B} \mid = 0. \tag{16}$$

Here and later, we assume that $\mathbf{A} \succ 0$ and $\mathbf{B} \succeq 0$. With the generalized eigenproblem (Golub & Loan, 1996), we know that $\mathbf{A}$ and $\mathbf{B}$ can be simultaneously diagonalized. More specifically, there exists a nonsingular matrix $\mathbf{Q}$ such that

$$\mathbf{A} = \mathbf{Q}'\mathbf{Q}, \quad \mathbf{B} = \mathbf{Q}'\mathbf{\Lambda}\mathbf{Q},$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix and $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ are the roots of (16). However, it is intractable to determine such $\mathbf{Q}$ since some entries of $\mathbf{B}$ are unknown. On the other hand, it is easy to simultaneously diagonalize $\mathbf{A}_{11}$ and $\mathbf{B}_{11}$. Our point of departure is to obtain an approximation of $\mathbf{Q}$ through co-diagonalizing $\mathbf{A}_{11}$ and $\mathbf{B}_{11}$. The following theorem provides a method for our purpose.

Theorem 2 *Given $\mathbf{A} \succ 0$ and $\mathbf{B} \succeq 0$ partitioned as in (1), there exists a nonsingular matrix*

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{17}$$

such that

$$\mathbf{A} = \mathbf{C}'\mathbf{C}, \quad \mathbf{A}_{11} = \mathbf{C}_1'\mathbf{C}_1, \quad \mathbf{B}_{11} = \mathbf{C}_1'\mathbf{\Lambda}_1\mathbf{C}_1, \tag{18}$$

where $\mathbf{\Lambda}_1$ is diagonal.

The proof of this theorem is given in Appendix A. This theorem gives us a base matrix $\mathbf{C}$ and its proof procedure provides a method to compute $\mathbf{C}$. Denote the ith column vector of $\mathbf{C}$ by $\mathbf{c}_i$, then $\mathbf{c}_i\mathbf{c}_i$'s constitute a set of base kernel matrices. Recall that, unlike the usual base matrix $\mathbf{U}$, $\mathbf{C}$ is not orthonormal. Although orthonormality is unnecessary for our problem, one may choose to orthonormalize $\mathbf{C}$ by using such methods as the Gram-Schmidt method since $\mathbf{C}$ is nonsingular. It is worthy to note that $\mathbf{C}$ is defined by using not only information from the input kernel matrix $\mathbf{A}$ but also information from the partial output kernel matrix $\mathbf{B}_{11}$. Information from both input and output is expected to be useful for classification and regression problems.

Let $\mathbf{B} = \mathbf{C}'\mathbf{S}\mathbf{C}$ and $\mathbf{K} = \mathbf{C}'\mathbf{W}\mathbf{C}$. We partition $\mathbf{S}$ and $\mathbf{W}$ into $\begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12}' & \mathbf{S}_{22} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{12}' & \mathbf{W}_{22} \end{bmatrix}$, respectively. By simple arithmetic calculations, it is obvious

that $\mathbf{S}_{11} = \mathbf{\Lambda}_1$ is diagonal. However, it is not always true that $\mathbf{S}_{12} = \mathbf{0}$ and $\mathbf{S}_{22}$ is diagonal. As a result, this implies that $\mathbf{S}$ is not necessarily diagonal. We now work with $\mathbf{W}$ instead of $\mathbf{K}$ so that the original transductive learning problem can be simplified. In other words, we treat $\mathbf{C}$ as a base matrix and $\mathbf{W}$ as a matrix to be estimated.

5. Experiments

To demonstrate the efficacy of our method, we apply it to the classification problem on the ionosphere and wine data sets from the UCI Machine Learning Repository and also the USPS handwritten digit data set with 16×16 digit images. For simplicity, we only use digits 1, 2, 3 and 4 corresponding to four classes with 1269, 929, 824 and 852 examples, respectively. As discussed in Section 2, we set the complete kernel matrix as $\mathbf{K} = (\mathbf{A} + \mathbf{B})/2$. We then follow the procedure described above to estimate the missing entries in $\mathbf{K}$. We first apply the method given in Section 4.3 to obtain the base matrix $\mathbf{C}$ as defined in (17). Next, we let $\mathbf{W}_{11} = (\mathbf{I}+\mathbf{\Lambda}_1)/2$ where $\mathbf{\Lambda}_1$ is defined in (18) and then use the Tanner-Wong algorithm given in Section 4.2 to obtain $\mathbf{W}_{21}$ and $\mathbf{W}_{22}$. We run the algorithm for 2,000 iterations. The first 1,000 iterations are treated as burn-in; inference is based only on the remaining 1,000 iterations. We use the following settings for the hyperparameters: $r = n + 1$ in (7), $\eta = 3.0$ in (8), $\zeta = 0.5$ and $\tau = \alpha\, n_1/\mathrm{tr}(\mathbf{W}_{11})$ in (9) where α is a constant in $[0.9, 3.0]$ (here we set $\alpha = 1.2$). In the last step, we set $\mathbf{K} = \mathbf{C}'\mathbf{W}\mathbf{C}$ to obtain the complete kernel matrix $\mathbf{K}$.

After obtaining $\mathbf{K}$, we implement the kernel nearest mean (KNM) classifier using $\mathbf{K}$. Details of its implementation can be found in (Zhang et al., 2003a). To compare it with some existing kernel classification methods based on the input kernel $\mathbf{A}$, we also implement kernel Fisher discriminant analysis (KFDA), support vector machine (SVM), and KNM. Moreover, we also implement the KNM classifier described in (Zhang et al., 2003a), which uses the EM algorithm to complete $\mathbf{K}$. For convenience, we denote it as KNM-EM. Experiments on these classifiers are performed with the following parameter settings. We set $\beta = 2.5$ for the ionosphere and wine data sets and set β to be the average Euclidean distance for the training examples in the USPS data set. For SVM, we set the regularization parameter $C = 300$. The results are averaged over 30 random splits of the data. For ionosphere and wine, we use 60% of the data for training and the remaining 40% for testing, while for USPS we use 99% for training and 1% for testing.

Table 1 shows the classification results. The standard deviations over 30 random splits are also shown inside brackets. In the table, KNM-TW1 refers to our method with the I-step for $\mathbf{W}_{22\cdot1}$ based on (13), while KNM-TW2 refers to the method when (15) is used instead for the I-step. We can see that the two methods give very similar results. This shows that even though the more efficient KNM-TW2 method only makes use of the approximate I-step in (15), it is effective enough in delivering comparable performance. Thus, for the much larger USPS data set, we use only KNM-TW2 which is more efficient. Note that KNM-EM also gives comparable classification accuracies as KNM-TW's, although KNM-EM works under a fully Bayesian setting on the kernel matrix $\mathbf{K}$. The classification accuracies of KFDA and SVM are also close, though generally lower.

6. Conclusion

In this paper, we have proposed two Bayesian hierarchical models for kernel matrix learning using the Tanner-Wong data augmentation algorithm, which is a variant of MCMC methods for missing data problems. Moreover, by formulating a symmetric-definite generalized eigenproblem, we present a method for choosing the base kernel matrices. We have demonstrated the flexibility and efficacy of our method under the classification setting. Although the formulation in this paper is based on the classification problem, it can be extended to the regression problem when the output space is continuous rather than discrete. In our previous work (Zhang et al., 2003b), we showed that the r parameter in the Wishart distribution is equal to the dimensionality of the feature space induced by $\mathbf{K}$ and it can be estimated using EM (Zhang et al., 2003a; Zhang et al., 2003b). In principle, we may also define a prior distribution for r, such as a Gamma distribution, and then use the approach presented in this paper to estimate it.

Acknowledgments

This research has been partially supported by the Research Grants Council of the Hong Kong Special Administrative Region under grants HKUST6195/02E and DAG03/04.EG36.

A. Proof of Theorem 2

Since $\mathbf{A} \succ 0$ and $\mathbf{B} \succ 0$, it then follows from (Golub & Loan, 1996) that $\mathbf{A}_{11} \succ 0$, $\mathbf{A}_{22\cdot1} \succ 0$, and $\mathbf{B}_{11} \succ 0$. Furthermore, from (Golub & Loan, 1996), there exist

Table 1. Test set accuracies obtained from the classification experiments (the highest values are shown in boldface).

	KNM-TW1	KNM-TW2	KNM-EM	KFDA	SVM	KNM
ionosphere	94.33 (±1.48)	**94.60** (±1.14)	94.50 (±1.87)	92.14 (±2.05)	91.71 (±2.05)	68.05 (±4.74)
wine	97.93 (±1.47)	97.51 (±1.80)	**98.12** (±1.45)	95.59 (±2.76)	96.85 (±1.51)	94.04 (±3.50)
USPS	–	**93.05** (±1.15)	92.28 (±1.18)	91.09 (±4.22)	90.27 (±1.78)	92.04 (±1.29)

nonsingular matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ such that

$$\mathbf{A}_{11} = \mathbf{C}_1'\mathbf{C}_1, \quad \mathbf{B}_{11} = \mathbf{C}_1'\mathbf{\Lambda}_1\mathbf{C}_1,$$

where $\mathbf{\Lambda}_1$ is diagonal and $\mathbf{C}_2'\mathbf{C}_2 = \mathbf{A}_{22\cdot 1}$. Note that

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22\cdot 1} \end{bmatrix}.$$

So we have

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C}_1' & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2' \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Putting $\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, we have $\mathbf{A} = \mathbf{C}'\mathbf{C}$.

References

Amari, S. (1995). Information geometry of the EM and *em* algorithms for neural networks. *Neural Networks*, *8*, 1379–1408.

Bousquet, O., & Herrmann, D. J. L. (2003). On the complexity of learning the kernel matrix. *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press.

Crammer, K., Keshet, J., & Singer, Y. (2003). Kernel design using boosting. *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press.

Cristianini, N., Kandola, J., Elisseeff, A., & Shawe-Taylor, J. (2002). On kernel target alignment. *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *39*, 1–38.

Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*. Baltimore: The Johns Hopkins University Press. Third edition.

Gupta, A., & Nagar, D. (2000). *Matrix variate distributions*. Boca Raton, FL: Chapman & Hall/CRC.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge, UK: Cambridge University Press.

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002). *Optimizing kernel alignment over combinations of kernels* (Technical Report 2002-121). NeuroCOLT.

Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P., & Jordan, M. I. (2002). Learning the kernel matrix with semi-definite programming. *Proceedings of the 19th International Conference on Machine Learning* (pp. 323–330). Sydney, Australia.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Chapman & Hall.

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation (with discussion). *Journal of the American Statistical Association*, *82*, 528–550.

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.

Zhang, Z. (2003). Learning metrics via discriminant kernels and multidimensional scaling: Toward expected Euclidean representation. *Proceedings of the 20th International Conference on Machine Learning* (pp. 872–879). Washington, D.C., USA.

Zhang, Z., Kwok, J. T., Yeung, D. Y., & Xiong, Y. (2003a). *Bayesian transductive learning of the kernel matrix using Wishart processes* (Technical Report HKUST-CS03-09). Department of Computer Science, Hong Kong University of Science and Technology. Available from ftp://ftp.cs.ust.hk/pub/techreport/03/tr03-09.ps.gz.

Zhang, Z., Yeung, D. Y., & Kwok, J. T. (2003b). *Gaussian-Wishart processes: A statistical view of kernels and its application to kernel learning* (Technical Report HKUST-CS03-15). Department of Computer Science, Hong Kong University of Science and Technology. Available from ftp://ftp.cs.ust.hk/pub/techreport/03/tr03-15.ps.