Nonlinear Multiple Kernel Learning via Mixture of Probabilistic Kernel Discriminant Analysis

Zheng Zhao
zhaozheng@asu.edu
Computer Science & Engineering
Arizona State University

Jieping Ye
jieping.ye@asu.edu
Computer Science & Engineering
Arizona State University

Shipeng Yu
shipeng.yu@siemens.com
CAD and Knowledge Solutions
Siemens Medical Solutions USA, Inc.

Huan Liu
huan.liu@asu.edu
Computer Science & Engineering
Arizona State University

Abstract

Multiple kernel learning (MKL) provides a powerful tool for heterogenous data integration. Most existing MKL formulations are based on a linear kernel combination, which, however, restricts the flexibility of the learning model. In this paper, we propose a novel nonlinear multiple kernel learning formulation based on the model combination. The proposed formulation (called MPKDA) is derived from a novel probabilistic model for kernel discriminant analysis (KDA) and its mixture. Experimental results on various real applications demonstrate that the proposed MPKDA model provides competitive performance comparing with the representative approaches. We also analyze the relationship between the proposed model and the existing KDA-based MKL formulations, and show how to use the proposed MPKDA model to handle missing data and perform localized multiple kernel learning (LMKL).

1 Introduction

Kernel methods, such as the support vector machine (SVM) [Vap95], gained popularity due to their successful applications in solving a wide range of real-world problems [TFM07, BHOS+08]. Kernel methods [SS02, STC04] work by embedding the input data into a high-dimensional feature space, where the embedding can be determined uniquely by specifying a kernel function that computes the dot product between data points in the feature space implicitly. Thus one of the central problems in kernel methods is the learning of good kernels.

Recently, multiple kernel learning (MKL), which learns a linear combination of multiple input kernels, has been shown to improve the classification performance [LCB+04]. In [LCB+04], an optimal kernel matrix is learnt by linearly combining a set of pre-specified kernel matrices and the combination coefficients can be determined by solving a convex optimization problem. In [BLMIJkltSa04], this problem is reformulated as support kernel machines (SKM), and this SKM formulation is further reformulated as a semi-infinite linear program (SILP) to handle large-scale problems [SRSS06, RBCG07]. Similar problems are also studied in [TK06, LJN06, MP07]. While most existing approaches are based on SVM, the MKL formulation in [FDBR04] is based on kernel discriminant analysis (KDA). The problem is reformulated as a semidefinite program (SDP) in [KMB06], which is further extended to handle multiclass problems in [YCJ07]. MKL provides a powerful tool for integrating multiple data with heterogenous representations and has been successfully applied in many real-world problems [BTOM07].

One limitation of most existing MKL formulations is that they are based on the linear kernel combination. The additive nature of the combination, on one hand, makes the problem convex, so globally optimal solution exists, but on the other hand, may restrict the model flexibility. In this paper, we propose a novel nonlinear MKL formulation based on the mixture of probabilistic kernel discriminant analysis (MPKDA), where each input kernel corresponds to a component in the mixture model. We show that the outputs of the mixture model forms a projection of the input instances, and in the transformed space, instances can be better separated. A kernel can be constructed in the transformed space, which forms a nonlinear combination of the pre-specified kernels. We show that the proposed mixture model includes existing KDA-based MKL approaches, such as the one in [YCJ07], as its special cases. Additional benefits of the proposed model include: (1) its model parameters can be automatically tuned in the Bayesian framework via the maximization of the data likelihood [GCSR95]; (2) it can utilize the local information contained in the input kernels to achieve localized multiple kernel learning [GA08]; and (3) it can handle missing data in the input kernels. We performed experiments using real-world data sets. Experimental results showed that the kernel generated by the proposed approach provides superior performance in comparison with the representative MKL approaches.

2 Background

We first define the notations used in this paper. We denote $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ as the input data of N instances, and $\phi(\mathbf{x}_i)$ as the mapping of the ith instance in the kernel-induced feature space. Assume the data has C classes, $\mathbf{y} = \{y_1, \ldots, y_N\}$ denotes the class label, with $y_i \in \{1, \ldots, C\}$ being the label of the ith instance. Let $\kappa(\mathbf{x}, X) = \phi(X)^T \phi(\mathbf{x}_j)$ and $k_{i,j} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. We denote K as the kernel matrix with $k_{i,j}$ as its (i,j)th element. Let $P = I - n^{-1}\mathbf{1}\mathbf{1}^T$ be the centering matrix. A kernel matrix K can be centered by $K_c = PKP$, such that $\phi_c(\mathbf{x}) = \phi(\mathbf{x}_i) - \bar{\phi}(\mathbf{x}), \bar{\phi}(\mathbf{x}) = \sum \phi(\mathbf{x}_i)/N$. We use boldface lowercase characters, such as $\mathbf{w}$ and $\mathbf{b}$ to denote vectors. We use I to denote the identity matrix and $\mathbf{1}$, the vector of all ones.

Kernel Discriminant Analysis (KDA): We are given a centered kernel matrix K and the class label $\mathbf{y}$. We define three scatter matrices in KDA as follows: $S_t^K = KK$ is the total scatter matrix; $S_b^K = n^{-1}KYY^TK$ is the between-class scatter matrix; and $S_w^K = S_t^K - S_b^K$ is the within-class scatter matrix. Here $Y \in \mathbb{R}^{N \times C}$ is defined as:

$$y_{i,j} = \begin{cases} \sqrt{\frac{N}{N_j}} - \sqrt{\frac{N_j}{N}} & y_i = j \\ -\sqrt{\frac{N_j}{N}} & \text{otherwise,} \end{cases} \tag{1}$$

where, N_j is the number of instances in the jth class. KDA computes a transformation matrix B by maximizing the separability of instances in the transformed space. More specifically, KDA simultaneously minimizes the variance of each class, $\text{trace}(B^T S_w^K B)$, and maximizes the separation of the projected class means, $\text{trace}(B^T S_b^K B)$ by solving the following problem:

$$B = \arg\max_B \left\{ trace\left(\left(B^T S_w^K B\right)^{-1} B^T S_b^K B \right) \right\}. \tag{2}$$

However, the above formulation is prone to overfitting. One way to overcome overfitting is to apply regularization, resulting in the following optimization problem:

$$\max_B \left\{ trace\left(\left(B^T \left(S_t^K + \lambda K_c\right) B\right)^{-1} B^T S_b^K B \right) \right\}. \tag{3}$$

The optimal solution is given by the eigenvectors corresponding to the largest $C-1$ eigenvalues of the following eigenvalue problem:

$$\left(S_t^K + \lambda K_c\right)^+ S_b^K \mathbf{b}_i = \beta_i \mathbf{b}_i. \tag{4}$$

Probabilistic models have also been proposed for KDA to improve its robustness. For example in [CL06], by relating Rayleighs coefficient to a noise model, the authors showed that their model is equivalent to KDA. However, most existing probabilistic models rely on regressing the instances to [-1, +1] or $[n/n_+,\text{-}n/n_-]$, where n_+ and n_- are the numbers of instances in positive and negative class, respectively. This coding scheme restricts existing works to binary-class problems and limits their applicability.

3 Probabilistic KDA

In this section, we propose a probabilistic model for KDA, called PKDA, which is able to handle multiclass problems. PKDA forms the base component in the mixture model for nonlinear MKL, which will be presented in the next section. Below, we first extend the probabilistic model for the linear ridge regression to the kernel case. We then show that by regressing to Y defined in Equation (1), we can obtain a probabilistic model, which is equivalent to KDA, under some mild assumptions. This is a novel finding of this work.

Let $\phi(\mathbf{x})$ be the mapping of an instance $\mathbf{x}$ in the kernel induced feature space. The ridge regression in the kernel induced feature space has the following form: $t = \phi(\mathbf{x})^T \mathbf{w} + \epsilon$, where ϵ is the random noise; its corresponding probabilistic model can be expressed as:

$$P(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, I) \tag{5}$$

$$P(t \mid \phi(\mathbf{x}), \mathbf{w}, \beta) = \mathcal{N}\left(t \mid \phi(\mathbf{x})^T \mathbf{w}, \beta^{-1}\right). \tag{6}$$

As shown in [SS02], by applying the representer theorem, the weight vector $\mathbf{w}$ can be expressed as $\mathbf{w} = \phi(X)^T \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^{N\times 1}$, and

$$t = \phi(\mathbf{x})^T \phi(X) \mathbf{b} + \epsilon = \kappa(\mathbf{x}, X) \mathbf{b} + \epsilon. \tag{7}$$

Assuming that K is of full rank[1], we have $\mathbf{b} = K^{-1}\phi(X)^T \mathbf{w}$. This shows that $\mathbf{b}$ also follows a gaussian distribution with its mean and variance are given by the following equations:

$$E(\mathbf{b}) = \mathbf{0} \tag{8}$$

$$\begin{aligned} E\left(\mathbf{b}\mathbf{b}^T\right) &= E\left(K^{-1}\phi(X)^T \mathbf{w}\mathbf{w}^T \phi(X) K^{-1}\right) \\ &= K^{-1}\phi(X)^T E\left(\mathbf{w}\mathbf{w}^T\right) \phi(X) K^{-1} \\ &= K^{-1}. \end{aligned} \tag{9}$$

From Equations (7-9), we have $t = \kappa(\mathbf{x}, X)\mathbf{b} + \epsilon$, and its corresponding probability model is:

$$P(\mathbf{b} \mid K) = \mathcal{N}\left(\mathbf{b} \mid \mathbf{0}, K^{-1}\right). \tag{10}$$

$$P(t \mid \kappa(\mathbf{x}, X), \mathbf{b}, \beta) = \mathcal{N}\left(t \mid \kappa(\mathbf{x}, X)\mathbf{b}, \beta^{-1}\right) \tag{11}$$

We can obtain its predictive distribution through the marginalization of $\mathbf{b}_c$, which has the following form:

$$P(t \mid \mathbf{y}_c, K, \beta_c) = \mathcal{N}\left(t \mid m_c(\mathbf{x}), \sigma_i^2(\mathbf{x})\right) \tag{12}$$

[1] Otherwise we can use K^+, the pseudo-inverse of K.

$$m_c(\mathbf{x}) = \kappa(\mathbf{x}, X)\left(K + \frac{1}{\beta_c} I\right)^{-1} \mathbf{y}_c \tag{13}$$

$$\sigma_c^2(\mathbf{x}) = \frac{1}{\beta_c} + \kappa(\mathbf{x}, X)\left(K + \beta_c K^2\right)^{-1} \kappa(X, \mathbf{x}). \tag{14}$$

We denote this predictive distribution as $\mathcal{PKR}(K, \beta)$. It can be shown that $\mathcal{PKR}(K, \beta)$ is closely related to the Gaussian Process [RW06], and the predictive distributions generated from the two models share the same mean. A similar model is also studied by [YTS05] in the multitask learning scenario. Based on $\mathcal{PKR}$, we present a probabilistic model for KDA with the following definition:

Definition 1 *Given a centered kernel matrix K, the* Probabilistic Kernel Discriminant Analysis*, or the PKDA, consists of a set of* C $\mathcal{PKR}_c(K, \beta_c)$*, where* C *is the number of classes, and $\mathcal{PKR}_c(K, \beta_c)$ uses $\mathbf{y}_c \in \mathbb{R}^n$ as its observations, and $\mathbf{y}_c$ is the cth column of Y defined in Equation (1).*

The model contains C $\mathcal{PKR}$'s, which share the same kernel matrix K. For $\mathcal{PKR}_c(K, \beta_c)$, β_c^{-1} is its noise term, which is analogous to the regularization parameter in KDA, $\mathbf{y}_c$ is the observation of $\mathcal{PKR}_c$, and can be used for model fitting. Given a data point $\mathbf{x}$, the PKDA model projects $\mathbf{x}$ to a C dimensional vector: $\mathbf{y} = (y_1, \ldots, y_C)$ with the cth element following the predictive distribution specified in Equations (12), (13) and (14). We show in Theorem 1 below that when the noise terms $1/\beta_i \to 0,\ i = 1, \ldots, C$, and under a mild assumption, KDA and PKDA are equivalent.

Theorem 1 *Assume instances are linearly independent in the kernel-induced feature space. When the noise terms $1/\beta_i \to 0$, the projection determined by the expectation of the predictive distributions in PKDA, $[m_1(\mathbf{x}), \ldots, m_C(\mathbf{x})]^T$, is equivalent to the projection generated by KDA, which is $B^T \kappa(X, \mathbf{x})$.*

In theorem 1, the equivalence means that the distances among instances under different projections are equal[2]. Various equivalent projections can be obtained by applying orthogonal transformations on an existing projection or by increasing the dimensionality of a input by adding dummy variables which contain only 0.

We first present two lemmas. The first one shows the assumption used in the theorem is indeed very mild.

Lemma 1 *When the RBF kernel function is used, as long as $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are distinct, K is of full rank, which means the instances are linearly independent in the feature space induced by K.*

The proof of the lemma can be found in [Mic84]. In real applications, it is usually the case that the given instances are distinct.

Let the truncated SVD [GV96] of K be $K = U_1 \Sigma_t U_1^T$, and the full SVD of $U_1^T Y$ be $U_1^T Y = P \Sigma_b Q$. We have the following result:

Lemma 2 *When instances are linearly independent in the kernel induced feature space, the following holds: $\Sigma_b^2 = \mathrm{diag}(1, \ldots, 1, 0)$.*

The Lemma can be proved by simultaneously diagonalizing S_t^K, S_w^K and S_b^K. Below we prove Theorem 1 to establish the equivalence between KDA and PKDA.

Proof of Theorem 1: We can show that for any input $\phi(\mathbf{x})$, $m_l(\mathbf{x}) = \kappa(\mathbf{x}, X)^T \left(K + \beta_i^{-1} I\right)^{-1} \mathbf{y}_l$. Therefore, when $\beta_i^{-1} \to 0$, the expectation of the predictive distributions in PKDA actually projects

[2]This definition for equivalence makes sense, since in kernel based learning algorithms, only the distance among instances are considered in model fitting.

data with a transformation matrix defined as $\hat{B} = (K)^+ Y$. The equivalence between KDA and PKDA can be established by studying the relationship between $\hat{B}$ and B. We first study on the structure of matrix $\hat{B}$. $\lim_{\beta_i^{-1}\to 0,\ i=1,\ldots,c} \hat{B} = K^+Y = U_1\Sigma_t^{-1}U_1^T H$. Recall that $U_1^T H = P\Sigma_b Q$, and let $P = [\mathbf{p}_1, \ldots, \mathbf{p}_c]$,

$$\lim_{\lambda\to 0} \hat{B} = U_1\Sigma_t^{-1}[\mathbf{p}_1, \ldots, \mathbf{p}_c]\Sigma_b Q. \tag{15}$$

Next, we have the structure of B as:

$$\left(S_t^K\right)^+ S_b^K = U_1\Sigma_t^{-1}P\Sigma_b^2 P^T\Sigma_t U_1^T.$$

Since $P^T\Sigma_t U_1^T U_1\Sigma_t^{-1}P = I$, it can be verified that the top $c-1$ eigenvectors of $\left(S_t^K\right)^+ S_b^K$ is given by the first $c-1$ columns of $U_1\Sigma_t^{-1}P$. Therefore we have:

$$B = U_1\Sigma_t^{-1}[\mathbf{p}_1, \ldots, \mathbf{p}_{c-1}]. \tag{16}$$

As Q is orthogonal, and Σ_b^2 =diag$(1, \ldots, 1, 0)$, we conclude that these two projections are equivalent. ■

4 Mixture of Probabilistic KDA

Given a collection of L kernels, $\{K_1, \ldots, K_L\}$, L different PKDA models can be constructed. In the following, we show how to derive nonlinear MKL via the mixture of PKDA models. MPKDA projects instances into a dimensionality-reduced space, where they can be better separated. It can be shown that the kernel constructed in the reduced space corresponds to a nonlinear composite of the L given kernels. We will also connect MPKDA to several existing work, and study MPKDA's capability on handling missing data. Some existing mixture models for regression, such as the one proposed in [RG02], split instances into disjoint sets by a gating network, and do not allow the base models to share instances with each other. We found this formulation render inferior performance in our applications due to the shortage of instances in each base model. In contrast, the proposed mixture model shares some similarity with the Gaussian mixture model (GMM) [Bis06], which allows all models to share training instances.

Given L kernels, we construct L PKDA models. In a PKDA model, we use $\mathcal{PKR}_c\left(K, \beta_c\right)$ to project an instance $\mathbf{x}$ to the cth coordinate of the transformed space. Let PKDA(K_l, c) denote $\mathcal{PKR}_c\left(K, \beta_c\right)$ in the PKDA constructed from the kernel K_l. The probability of using PKDA(K_l, c) to project $\mathbf{x}$ to t on the cth coordinate of the transformed space is: $p\left(y \mid \mathbf{y}_c, K_l, \beta_{l,c}\right)$, which is given in Equation (12). In the generating process, let the prior probability of picking PKDA(K_l, c) to generate t be π_l with $\pi_1 + \ldots + \pi_L = 1$. For each input $\mathbf{x}_i$, we introduce an L dimensional latent variable $\mathbf{z}_i$ that indicates from which model t_i, the response of $\mathbf{x}_i$, is generated, that is if t_i is generated from PKDA(K_l, c), then $z_{i,l} = 1$ and all other elements in $\mathbf{z}_i$ is set to 0. We can express the joint likelihood of using MPKDA to project $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ to $\mathbf{t} = (t_1, \ldots, t_N)$ in the cth coordinate of the transformed space as:

$$\begin{aligned} &p\left(\mathbf{t}, Z | K_1, K_2, \cdots, K_L, \mathbf{y}_c, \Theta\right) \\ &\quad = \prod_{i=1}^{N}\prod_{l=1}^{L} \pi_l^{z_{k,l}} \mathcal{N}\left(t_i | m_{i,c}^{(l)}, \left(\sigma_{i,c}^{(l)}\right)^2\right)^{z_{i,l}}. \end{aligned} \tag{17}$$

In Equation (17), Z is the set of latent variables $Z = (z_{i,l})_{N\times L}$. Θ corresponds to the model parameters, which include prior probabilities π_l and the noise term $\beta_{c,l}^{-1}$ for each kernel K_l. Θ and Z are unknown and can be obtained by maximizing the joint likelihood specified in Equation (17). Next, we present an EM algorithm for estimating these parameters.

4.1 An EM Algorithm for Fitting MPKDA

E Step. Assume Π is known, we can show that the posterior distribution of Z takes the following form:

$$P\left(Z|K_1,\cdots,K_L,\mathbf{t}\right) \propto P\left(Z\right) P\left(\mathbf{t}|K_1,\cdots,K_L,Z\right).$$

$$= \prod_{i=1}^{N}\prod_{l=1}^{L}\left\{\pi_l \mathcal{N}\left(t_i|m_{i,c}^{(l)},\left(\sigma_{i,c}^{(l)}\right)^2\right)\right\}^{z_{i,l}}. \tag{18}$$

Using standard techniques, we can show that the responsibility of PKDA(K_l, c) for $\mathbf{x}_i$ is given by

$$\gamma_{i,l} = E\left(z_{i,l}\right) = \frac{\pi_l \mathcal{N}\left(t_i|m_{i,c}^{(l)},\left(\sigma_{i,c}^{(l)}\right)^2\right)}{\sum_{j=1}^{L}\pi_j \mathcal{N}\left(t_i|m_{i,c}^{(j)},\left(\sigma_{i,c}^{(j)}\right)^2\right)}. \tag{19}$$

The responsibilities can be used to determine the expectation of the complete data log likelihood, which defines the Q function:

$$Q\left(\Theta,\Theta^{\text{old}}\right) = E_z\left(\ln P\left(\mathbf{t},Z|\Theta\right)\right)$$
$$= \sum_{i=1}^{N}\sum_{l=1}^{L}\gamma_{i,l}\left\{\ln\pi_l + \ln\mathcal{N}\left(t_i|m_{i,c}^{(j)},\left(\sigma_{i,c}^{(j)}\right)^2\right)\right\}.$$

M Step. Assume Z is known, we can find the Θ by maximizing the Q function under certain constraints which leads to the following updating:

$$\pi_l^{new} = \frac{1}{N}\sum_{i=1}^{N}\gamma_{i,l}.\ \beta_{c,l}^{\text{new}} = \frac{\sum_{i=1}^{N}\gamma_{i,l}}{\sum_{i=1}^{N}\gamma_{i,l}\left(t_i - m_{i,c}^{(l)}\right)^2}. \tag{20}$$

4.2 Projection with MPKDA

With the computed priors $\pi_1,\ldots,\pi_L$ for the cth coordinate in the transformed space, we can obtain the projection of a new point $\mathbf{x}^*$ on the the cth coordinate by calculating the expectation of y^*:

$$P\left(y^*|\mathbf{x}^*\right) = \sum_{l=1}^{L} P\left(y^*|\mathbf{y}_c,K_l,\mathbf{x}^*\right) P\left(K_l|\mathbf{y}_i,\mathbf{x}^*\right), \tag{21}$$

$$m_c\left(\mathbf{x}^*\right) = \int y\cdot P\left(y|\mathbf{x}^*\right)dy$$
$$= \sum_{l=1}^{L} m_c^{(l)}\left(\mathbf{x}^*\right) P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right). \tag{22}$$

In Equation (22), $P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right)$ is the posterior of using the PKDA(K_l,c) for projecting, when $\mathbf{x}^*$ is given. We can assume that the selection of the model is independent of $\mathbf{x}^*$, and we have:

$$\begin{aligned} p\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right) &= p\left(K_l|\mathbf{y}_c\right) = \pi_l \\ m_c\left(\mathbf{x}^*\right) &= \sum_{l=1}^{L} m_c^{(l)}\left(\mathbf{x}^*\right)\pi_l. \end{aligned} \tag{23}$$

We can also make $P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right)$ dependent on $\mathbf{x}^*$ by using the following formulation:

$$P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right) \propto P\left(\mathbf{x}^*|K_l,\mathbf{y}_c\right)P\left(K_l|\mathbf{y}_c\right) \tag{24}$$

$$P\left(\mathbf{x}^*|K_l,\mathbf{y}_c\right) = \frac{\sum_{i=1}^{N}\gamma_{i,l}\kappa_l\left(\mathbf{x}_i,\mathbf{x}^*\right)}{\sum_{l=1}^{L}\sum_{i=1}^{N}\gamma_{i,l}\kappa_l\left(\mathbf{x}_i,\mathbf{x}^*\right)}. \tag{25}$$

For a given test data point, we can not use Equation (12) to obtain its generating probability, since its label is not known. We propose Equation (25) to estimate this probability by utilizing the similarity between a test point and the training points. A test point can obtain a large $P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right)$ value, only if it is similar to the training points that the lth model is "responsible" to. Based on this definition, we have:

$$m_c\left(\mathbf{x}^*\right) = \sum_{l=1}^{L}\frac{\sum_{i=1}^{N}\gamma_{i,l}\kappa_l\left(\mathbf{x}_i,\mathbf{x}^*\right)}{\sum_{l=1}^{L}\sum_{i=1}^{N}\gamma_{i,l}\kappa_l\left(\mathbf{x}_i,\mathbf{x}^*\right)}m_c^{(l)}\left(\mathbf{x}^*\right)\pi_l \tag{26}$$

It is interesting to notice that by making $P\left(K_l|\mathbf{x}^*,\mathbf{y}_c\right)$ dependent on $\mathbf{x}^*$, we obtain the localized MLK [GA08].

4.3 MPKDA for Nonlinear MKL

The pseudocode of MPKDA for multiple kernel learning is given in Algorithm 1. It contains 5 steps: (1) In Line 3, we construct the target Y. (2) In Lines 4-6, we construct L PKDA as the base models. (3) In Lines 7-10, we build the mixture model by calculating the generating probabilities for training instances, and learning the mixture coefficient using the EM algorithm specified in Section 4.1. (4) In Lines 11-13, we project instances to the dimensionality-reduced space using the obtained mixture model. And (5) in Lines 14-15, we obtain the final learnt kernel using the transformed instances in the reduced space. It can be shown that the overall time complexity of MPKDA for kernel learning is $O\left(C\max\left(LN^3,M^2\right)\right)$, where, C is the number of classes, L is the number of input kernels, N and M are the size of the training and the whole data, respectively. The kernel learnt from MPKDA is based on the output of a mixture model. It is a nonlinear composite of the input kernels.

4.4 Connection to Existing MKL Approaches

Existing MKL approaches for KDA, such as the ones in [KMB06] and [YCJ07], are closely related to MPKDA. It can be shown that for KDA, the projection vector $\mathbf{b}_c$ is uniquely determined by the input kernel matrix K through Equation (4). Therefore, existing KDA-based MKL approaches project an instance $\mathbf{x}^*$ as follows:

$$m_c\left(\mathbf{x}^*\right) = \mathbf{b}_c^T\sum_{l=1}^{L}\pi_l\kappa\left(X,\mathbf{x}^*\right), \tag{27}$$

Algorithm 1 MPKDA for Nonlinear MKL

Input: $K_1, \ldots, K_L$ and class label $\mathbf{y}$
Output: K^{NEW}, the learnt kernel
Construct Y from $\mathbf{y}$ using Equation (1).
for each class c, and each kernel K_l **do**
    Construct the PKDA(K_l, c).
end for
for each training instance **do**
    Calculate its generating probability on the L obtained PKDA models.
end for
Obtain $\Pi = \{\pi_{l,c}\}_{L \times C}$ using the EM algorithm.
for each instances **do**
    Project it to the dimensionality reduced space using the methods proposed in Section 4.2.
end for
Construct K^{NEW} using the transformed instances.
Return: K^{NEW}.

which can be compared to Equation (23) which has the form:

$$m_c(\mathbf{x}^*) = \sum_{l=1}^{L} m_l(\mathbf{x}^*)\pi_l = \sum_{l=1}^{L} \pi_l \mathbf{b}_{l,c}^T \kappa(X, \mathbf{x}^*). \tag{28}$$

We can see that, although both approaches try to project data to a space where instances can be best separated, the former one has requires different kernels to share the same $\mathbf{b}$, therefore it can be regarded as a special case of the later one.

4.5 Handling Missing Data Points

In some applications, some data points may be missing in the kernels. For tractional MKL methods, there are two possible ways to address this issue: (1) finding the instances that are present in all kernels and learn the kernel combination coefficient using these instances only. (2) adding zeros to the columns and rows corresponding to the missing data points in the kernel matrices. The first approach may not effectively use the data, when there are many kernels and their intersection is small. The second approach is also problematic, since the zero blocks in the kernel will generate many instances on the origin of the feature space, which may not be desirable for supervised algorithms. MPKDA is a more flexible model based on the model combination. In MPKDA, different PKDA models can be trained on the data that it observed, and their output can then be combined according to the model responsibilities. In the process we do not need to remove (or add dummy) instances from (or to) any kernels. Therefore the flexibility of MPKDA enables it to effectively deal with the missing data points in kernels.

5 Empirical Study

In this section, we empirically evaluate the performance of PKDA and MPKDA. Nine Microarray data sets from the Gene Expression Omnibus (GEO)[3] are used in the experiments. Detailed

[3]http://www.ncbi.nlm.nih.gov/geo/. Data set IDs are: GDS1412, GDS1454, GDS2771, GDS0968, GDS1627, GDS1962, GDS2545, GDS1975 and GDS2415.

Table 1: Summary of the benchmark data sets.

Data Sets	# Instance	# Feature	# Class	# Kernels
HRT	89	2759	2	15
CLL-SUB	100	11340	2	15
SMK-CAN	187	19993	2	15
TOX	171	5748	4	15
BRE-CAN-CEL	83	20163	4	15
BRA-CAN	180	49151	4	15
PRO-CAN	153	11302	3	15
GLI-III-IV	85	22283	2	15
BRE-CAN-REC	118	19200	2	15

information on these data sets are given in Table 1. To obtain the data, we filter the raw data from GEO by removing genes with low variance, and then normalize the data using the standard technique. On each data set, we generate 15 kernels as follows. According to the biological process related to each data set, we obtain 15 relevant biological pathways from our biological collaborators. For instance, CLL-SUB data contains gene expression information of samples from different B-cell chronic lymphocytic leukemia (CLL) subtypes [HSSD04]. For the data *P-53 signalling pathways* and *Chronic myeloid leukemia pathways* are included in the pathway list, since they are known to be related to CLL. After obtained the pathways, we used the genes in the pathways to filter the data and build kernels on the filtered data. For each data set, we generate 15 kernels. Different pathways associate to different biological functions. Kernels derived from these pathways provides us different angles to observe the relationships among instances under the influence from the different biological functions. Our task is to learn a kernel form these 15 kernels, which reflects the intrinsic relationships among the instances.

We compare the MPKDA model with two representative MKL algorithms: KDAQCQP [YCJ07] and SVMSILP [RBCG07]. We implemented MPKDA in two different ways: MPKDA-Pri and MPKDA-Pos, corresponding to different projection schemes in Equation (23) and Equation (26), respectively. The average performance from 10 different runs using different training and test sets is reported. All algorithms are implemented in Matlab.

5.1 Accuracy Comparison

Table 2 presents the performance of different MKL algorithms. Based on the results, we have the following observations. First, among the four MKL approaches, MPKDA-Pri achieved the best performance. For instance, on the nine benchmark data sets, MPKDA-Pri achieved an averaged accuracy of 0.65, which is followed by MPKDA-Pos (0.64), SVMSILP (0.63) and KDAQCQP (0.62). In addition, among the nine data sets, MPKDA-Pri achieved the highest accuracy on 8 of them. We conjecture that the flexibility of the nonlinear MKL model based on the proposed MPKDA may help achieve better performance. Compared with the results obtained from single kernels, MPKDA-Pri achieved similar performance as the best single kernel. We also observed in the BRA-CAN and BRE-CAN-CEL data sets that the MKL algorithms can achieve a better accuracy when complementary information exists. This is consistent with the observations in [LBC+04].

Second, in comparison to MPKDA-Pos, MPKDA-Pri achieved a better performance. As we discussed in the theoretical part, MPKDA-Pos performs prediction using the posterior distribution, and it is able to handle local information in the learning process. We conjecture that the inferior

Table 2: The performance of different MKL algorithms on the benchmark data sets. For each algorithm, the first column contains the averaged accuracy and the second column contains the standard deviation. The numbers with bold facetype indicates the highest accuracy rates achieved by MKL algorithms on each data set. "Max Single" and "Ave Single" denotes the highest and the averaged accuracy achieved by the 15 input kernels on each data set, respectively.

Data Sets	MPKDA-Pri		MPKDA-Pos		KDAQCQP		SVMSILP		Max Single		Ave Single
HRT	**0.65**	0.09	0.63	0.08	0.64	0.11	0.63	0.10	0.66	0.09	0.53
CLL-SUB	**0.64**	0.07	**0.64**	0.09	0.62	0.09	0.62	0.09	0.66	0.11	0.60
SMK-CAN	**0.66**	0.04	**0.66**	0.03	0.65	0.04	0.65	0.04	0.66	0.03	0.63
TOX	**0.71**	0.08	0.70	0.08	**0.71**	0.06	0.69	0.06	0.70	0.05	0.48
BRE-CAN-CEL	**0.60**	0.09	**0.60**	0.08	0.53	0.09	0.56	0.09	0.54	0.08	0.48
BRA-CAN	**0.59**	0.06	0.57	0.09	0.56	0.04	0.57	0.06	0.56	0.04	0.41
PRO-CAN	**0.55**	0.04	0.53	0.05	0.52	0.06	0.53	0.05	0.57	0.03	0.50
GLI-III-IV	0.79	0.07	**0.80**	0.07	0.76	0.09	0.76	0.09	0.81	0.05	0.72
BRE-CAN-REC	**0.64**	0.13	0.62	0.16	0.57	0.13	0.58	0.13	0.65	0.15	0.56
AVE & WIN	**0.65**	8	0.64	4	0.62	1	0.63	0	0.65		0.55

performance of MPKDA-Pos may be due to the variance in the data. The benchmark data sets used in the experiment are all of small size, therefore the estimation of the posterior may not be reliable given the small sample size. MPKDA-Pri is thus more effective in this case.

Third, in our experiment, we observed that the set of the combination coefficients Π generated by MPKDA is sparse. Usually no more than five kernels are selected for each class. We observed similar trends in the combination coefficients generated by KDAQCQP and SVMSILP too. In general KDAQCQP generates the sparsest combination coefficient, which is followed by MPKDA, and then SVMSILP. Recall that most benchmark data sets are cancer related. We observed from our experiments that the *MAPK signaling pathway* is frequently selected by the four MKL approaches. *MAPK signaling pathway* is a highly conserved module that is involved in various cellular functions, including cell proliferation, differentiation and migration, and is known to be cancer related. This observation validates our experiment design in the use of pathways to induce kernels.

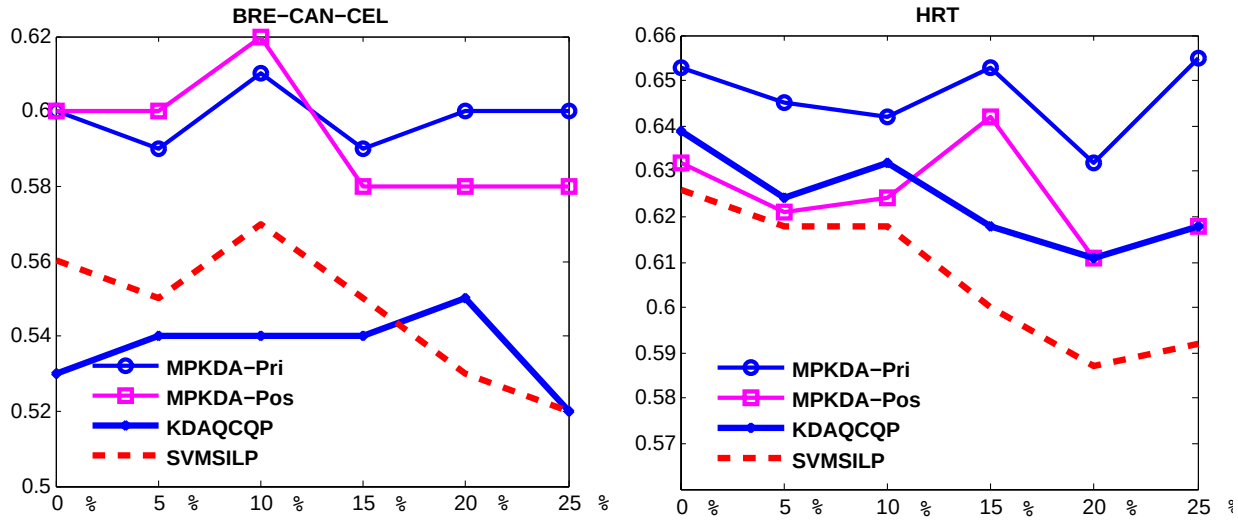

Figure 1: Comparison of different algorithms in the presence of missing data points. The horizontal axis denotes the percentage of the missing data and the vertical axis denotes the accuracy.

5.2 Handle Missing Data Points

Figure 1 shows the performance of the algorithms when missing data points are presented in the kernels. In this experiment, we randomly select 5%, 10%, 15%, 20% and 25% training instances as missing instances for each kernel. For KDAQCQP and SVMSILP, the missing data points are handled by adding zero columns and rows to the matrices to complete the kernel matrix, which corresponds to the second strategy mentioned in Section 4.5. The first strategy does not work well in our experiment, since the intersection of the kernels is usually very small. The experimental results in Figure 1 show that MPKDA-Pri is quite effective in dealing with missing data points. We also observed that MPKDA-Pos is less robust and we conjecture that this may also be caused by overfitting when the number of training instances is small.

6 Conclusion

In this paper, we proposed a nonlinear MKL formulation, called MPKDA, which is based on the mixture of multiple probabilistic KDA models. As compared to existing linear kernel combination approaches, MPKDA is more flexible and it can effectively deal with missing data in the kernels. Our experimental results demonstrated the effectiveness of the proposed model.

The estimation of the generating probability for training instance is important for MPKDA. We found in our experiment that when the training sample size is small, the estimation of the generating probability becomes less reliable. To address this problem, we plan to ultilize the t-process [YTY07] to improve model robustness. Another line of our ongoing work is to apply MPKDA to other real-world applications involving multiple heterogenous data sources.

References

[BHOS+08] A. Ben-Hur, C. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch. Support vector machines and kernels for computational biology. *PLoS Computational Biology*, 4, 2008.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BLMIJkltSa04] F. R. Bach, G. R. G. Lanckriet, conic duality M. I. JordanMultiple kernel learning, and the SMO algorithm. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[BTOM07] T. D. Bie, L. C. Tranchevent, L. Oeffelen, and Y. Moreau. Kernel-based data fusion for gene prioritization. *Bioinformatics*, 23:i125–i132, 2007.

[CL06] Tonatiuh Pena Centeno and Neil D. Lawrence. Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis. *J. Mach. Learn. Res.*, 7:455–491, 2006.

[FDBR04] G. Fung, M. Dundar, J. Bi, and B. Rao. A fast iterative algorithm for fisher discriminant using heterogeneous kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.

[GA08] Mehmet Gonen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25 th International Conference on Machine Learning*, 2008.

[GCSR95] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 1995.

[GV96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[HSSD04] C. Haslinger, N. Schweifer, S. Stilgenbauer, and H. Dhner. Microarray gene expression profiling of b-cell chronic lymphocytic leukemia subgroups defined by genomic aberrations and vh mutation status. *J Clin Oncol*, 22:3937–49, 2004.

[KMB06] S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006.

[LBC+04] G.R.G. Lanckriet, T. De Bie, N. Cristianini, M.I. Jordan, and W.S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004.

[LCB+04] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[LJN06] D. Lewis, T. Jebara, and W. S. Noble. Nonstationary kernel combination. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006.

[Mic84] Charles A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1984.

[MP07] C. A. Micchelli and M. Pontil. Feature space perspectives for learning the kernel. *Machine Learning*, 66:297–319, 2007.

[RBCG07] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, 2007.

[RG02] Carl Edward Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In *NIPS*, 2002.

[RW06] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[SRSS06] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[SS02] B. Scholköpf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[TFM07] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16:349–366, 2007.

[TK06] I. W. Tsang and J. T. Kwok. Efficient hyperkernel learning using second-order cone programming. *IEEE Trans. on Neural Networks*, 17:48–58, 2006.

[Vap95] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., 1995.

[YCJ07] J. Ye, J. Chen, and S. Ji. Discriminant kernel and regularization parameter learning via semidefinite programming. In *The Twenty-Fourth International Conference on Machine Learning*, 2007.

[YTS05] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML*, 2005.

[YTY07] Shipeng Yu, Volker Tresp, and Kai Yu. Robust multi-task learning with t-processes. In *Proceedings of the 24 th International Conference on Machine Learning*, 2007.