Multiclass Probabilistic Kernel Discriminant Analysis

Zheng Zhao†, Liang Sun†, Shipeng Yu‡, Huan Liu†, Jieping Ye†
†Department of Computer Science and Engineering, Arizona State University, USA
‡CAD and Knowledge Solutions, Siemens Medical Solutions USA, Inc.
†{zhaozheng, sun.liang, huan.liu, jieping.ye}@asu.edu; ‡shipeng.yu@siemens.com

Abstract

Kernel discriminant analysis (KDA) is an effective approach for supervised nonlinear dimensionality reduction. Probabilistic models can be used with KDA to improve its robustness. However, the state of the art of such models could only handle binary class problems, which confines their application in many real world problems. To overcome this limitation, we propose a novel nonparametric probabilistic model based on Gaussian Process for KDA to handle multiclass problems. The model provides a novel Bayesian interpretation for KDA, which allows its parameters to be automatically tuned through the optimization of the marginal log-likelihood of the data. Empirical study demonstrates the efficacy of the proposed model.

1 Introduction

Learning in high-dimensional spaces is challenging due to the curse of dimensionality [Hastie *et al.*, 2001]. Linear and kernel discriminant analysis provide effective means to reduce dimensionality [McLachlan, 1992]. As an extension of the linear discriminant analysis (LDA), kernel discriminant analysis (KDA) extends LDA to the kernel-induced feature space [Scholköpf and Smola, 2002], so that the non-linear structures in the data can be handled effectively [Mika *et al.*, 1999]. KDA has been shown to be effective in many applications, such as image processing [Belhumeur *et al.*, 1997] and text information retrieval [Howland *et al.*, 2003], where KDA is used to generate low dimensional representations of the original data for subsequent analysis. KDA is known to be prone to overfitting, which causes the model to be sensitive to noise. One effective way to address the problem is to apply regularization for variance reduction [Friedman, 1989]. Various model selection approaches have been proposed to determine a good regularization parameter value from a finite set of candidate values based on cross-validation, which is, however, time consuming. In addition, the performance of cross-validation could also be influenced by the quality of the candidate set. Therefore it is important to study how to automatically determine the model parameters for KDA to improve its robustness.

Probabilistic models have been proposed for LDA and KDA. For example in [Centeno and Lawrence, 2006], by relating Rayleighs coefficient to a noise model, the authors showed that their model is equivalent to KDA. The authors also demonstrated that through a Bayesian model selection approach [Gelman *et al.*, 1995], parameters in their model can be efficiently tuned to avoid the cost of cross-validation. However, due to the coding scheme used in designing, existing models are limited to binary class problems, which significantly confines their application in real world problems. To address the limitation, in this paper, we develop a nonparametric probabilistic model for KDA based on Gaussian Process (GP) to handle multiclass problems. We show that under a mild assumption, which holds in most real applications, the proposed model is equivalent to KDA, when the noise terms approach zero. The proposed model provides a novel Bayesian interpretation for KDA, which allows its parameters to be automatically tuned through the optimization of the marginal log-likelihood of the data. We show that the optimization problem can be formulated as a DC-programming problem [Horst and Thoai, 1999], which can be solved efficiently. As a probabilistic model based on GP, many well studied techniques for GP can be directly applied to the proposed model and enable it, for example, to handle problems with large scale data [Snelson, 2007], and learn from multiple heterogenous data sources [Lanckriet *et al.*, 2004; Rasmussen and Ghahramani, 2002]. The proposed model substantially extends the capability of KDA. To evaluate its performance, we conduct experiments on benchmark data sets. Experimental results demonstrate its efficacy.

2 Background

We first define the notations used in the rest of this paper. We use $\phi(X) = \{\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)\}$ to denote a data set of n samples, where $\phi(\mathbf{x}_i)$ is the mapping of the ith sample $\mathbf{x}_i$ in the kernel-induced feature space. Assume the data has c classes, and $\mathbf{y} = \{y_1, \ldots, y_n\}$ denotes the class label, with $y_i \in \{1, \ldots, c\}$ being the label of the ith sample. Let $k_{i,j} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, and let K denote 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. $K_c = PKP$ denotes the centered kernel matrix, such that samples in the feature space induced from K_c are centered: $\phi_c(\mathbf{x}) = \phi(\mathbf{x}_i) - \bar{\phi}(\mathbf{x})$, $\bar{\phi}(\mathbf{x}) = \frac{1}{n}\sum \phi(\mathbf{x}_i)$.

In this paper, we use uppercase characters to denote matrices, boldface lowercase characters to denote vectors, and standard lowercase characters to denote scalars. Also, we use I to denote the identity matrix and $\mathbf{1}$ the vector of all ones. Below we give a brief introduction to KDA and GP.

Kernel Discriminant Analysis (KDA): Given the kernel matrix K and the class label $\mathbf{y}$, KDA determines a transformation matrix B to project samples, such that they can be best separated. Analogous to LDA, given K_c, the centered K, we denote $S_t^K = K_c K_c$ as the total scatter matrix, $S_b^K = n^{-1} K_c Y Y^T K_c$ as the between-class scatter matrix, and $S_w^K = S_t^K - S_b^K$ as the within-class scatter matrix. In the definition of S_b^K, $Y \in \mathbb{R}^{n \times c}$ is the coding matrix derived from $\mathbf{y}$ as [Ye, 2007]:

$$y_{i,j} = \begin{cases} \sqrt{\frac{n}{n_j}} - \sqrt{\frac{n_j}{n}} & y_i = j \\ -\sqrt{\frac{n_j}{n}} & \text{otherwise.} \end{cases} \quad (1)$$

KDA maximizes the separability of the samples in the dimensionality-reduced space by simultaneously minimizing trace($B^T S_w^K B$) and maximizing trace($B^T S_b^K B$), which correspond to the within-class distance and the between-class distance, respectively. KDA solves the following optimization problem:

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

However, the above formulation is prone to overfitting. Regularization is commonly applied to alleviate the problem:

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

And its optimal solution can be obtained by computing the principal eigenvectors of the following eigenvalue problem:

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

Here $(\cdot)^+$ denotes the matrix pseudo inverse [Golub and Van Loan, 1996]. Choosing a proper λ value is crucial to KDA.

Gaussian Process (GP) Regression: A Gaussian Process (GP) is a stochastic process defining a nonparametric prior over functions [Rasmussen and Williams, 2006]. A real function $f : \mathbb{R}^d \to \mathbb{R}$ follows a GP, denoted as $\mathcal{GP}(h; \kappa)$, if for any finite number of data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^n$ follows a multivariate Gaussian distribution $\mathcal{N}(\mathbf{h}, K)$, with a mean function $\mathbf{h} = \{h(\mathbf{x}_i)\}_{i=1}^n$ and a covariance matrix $K = \{\kappa(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n$. It is common to set $\mathbf{h} \equiv 0$. Given the training data and a test point $\mathbf{x}_*$, GP regression assumes $p(y_*|\mathbf{x}_*) = \int p(y_*|f_*) p(f_*) df_*$, where $f_* = f(\mathbf{x}_*)$ and $p(y_*|f_*)$ is an isotropic Gaussian with the variance λ specifying the system noise. By marginalizing f_*, we can obtain the joint distribution of $\mathbf{y}$ and y_*, which takes the form:

$$\mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \middle| 0, \begin{bmatrix} K_{X,X} + \sigma^2 I & \mathbf{k}_{X,\mathbf{x}_*} \\ \mathbf{k}_{\mathbf{x}_*,X} & k_{\mathbf{x}_*,\mathbf{x}_*} \end{bmatrix} \right), \quad (5)$$

where σ corresponds to the system noise. Using Equation (5), we can obtain the conditional distribution of $p(f_* \mid \mathbf{y})$, which is also a Gaussian with its mean and variance given by:

$$f_* \sim \mathcal{N}\left(f_* \mid m_*, \sigma_*^2 \right), \quad (6)$$

$$m_* = \mathbf{k}_{\mathbf{x}_*,X} \left(K_{X,X} + \sigma^2 I \right)^{-1} \mathbf{y}, \quad (7)$$

$$\sigma_*^2 = k_{x_*,x_*} - \mathbf{k}_{\mathbf{x}_*,X} \left(K_{X,X} + \sigma^2 I \right)^{-1} \mathbf{k}_{X,\mathbf{x}_*}. \quad (8)$$

3 *PKDA*: A Probabilistic Model for Kernel Discriminant Analysis

We propose a probabilistic model for KDA based on Gaussian Process. We call the model, "Probabilistic Kernel Discriminant Analysis" or PKDA. Existing probabilistic models for discriminant analysis, such as the ones proposed in [Ioffe, 2006] and [Centeno and Lawrence, 2006] rely on regressing the samples to [-1, +1] or [n/n_1,-n/n_2], where n_1 and n_2 are the numbers of samples in positive and negative classes, respectively. The coding scheme only works for binary class problems. In PKDA, we propose to regress samples to Y defined in Equation (1). We can show that assuming the samples are linearly independent in the kernel-induced feature space, PKDA is equivalent to KDA, when the noise terms approach zero. We first present the PKDA model as follows:

Definition 1 *Given a centered kernel matrix K_c, the* Probabilistic Kernel Discriminant Analysis*, or the PKDA, consists of a set of c Gaussian Processes,*

$$\mathcal{GP}_l(\mathbf{h}_l; K_c), \quad l = 1, \ldots, c, \quad (9)$$

where c is the number of classes, and the lth GP, $\mathcal{GP}_l$, uses $\mathbf{y}_l \in \mathbb{R}^n$ as its observations with $\mathbf{y}_l$ being the lth column of Y defined in Equation (1).

Figure 1 shows the graphical model of PKDA. The model contains c GPs making predictions with noisy observations. In the model, the c GPs share the same covariance matrix K. For the lth GP, $\lambda_l = \sigma_l^2$ is its noise term, which is analogous to the regularization parameter in KDA, and $\mathbf{h}_l$ is its mean vector. Following the convention, we fix $\mathbf{h}_i \equiv 0$. $\mathbf{y}_l$ is its observation, which is the lth column of Y, where $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_c]$ is defined as in Equation (1). Given a test data point $\phi(\mathbf{x}_*)$, the PKDA model projects $\phi(\mathbf{x}_*)$ to a vector of c dimensions: $\mathbf{y}_* = (y_{*,1}, \ldots, y_{*,c})$ with the ith element following the predictive distribution specified as:

$$y_{*,i} \sim \mathcal{N}\left(y_{*,i} \mid m_i(\mathbf{x}_*), \sigma_i^2(\mathbf{x}_*) \right), \quad (10)$$

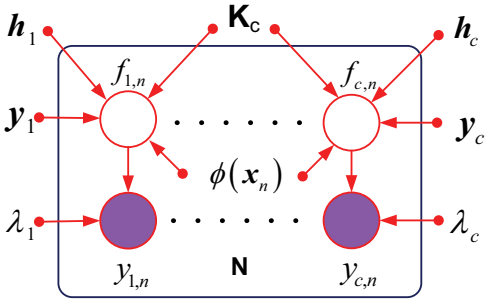

Figure 1: The graphical model for probabilistic kernel discriminant analysis (PKDA). Following the convention, we fix $\mathbf{h}_l \equiv 0$, and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_c]$ is defined as in Equation (1).

and $m_i(\mathbf{x}_*)$ and $\sigma_i^2(\mathbf{x}_*)$ are given by Equations (7) and (8). With the computed $m_i(\mathbf{x}_*)$ and $\sigma_i^2(\mathbf{x}_*)$, the projections of the test point can be either sampled from the normal distributions specified in Equation (10), or obtained directly using $\mathbf{m}(x_*) = (m_1(x_*), \ldots, m_c(x_*))$, which maximizes the likelihood of the observations. In theorem 1 below, we show the equivalent relationship between KDA and PKDA under certain mild conditions.

Theorem 1 *Assume that samples are linearly independent in the kernel-induced feature space. When the noise terms $\lambda_l \to 0$, $l = 1, \ldots, c$, the projection determined by the expectation of the predictive distributions of PKDA is equivalent to that generated by KDA. More specifically, for $\mathbf{m}(x_*) = (m_1(\mathbf{x}_*), \ldots, m_c(\mathbf{x}_*))^T$ and $\mathbf{k}_* = K_{X,\mathbf{x}_*}$, we have $\mathbf{m}(x_*) = B^T\mathbf{k}_*$ up to an expansion with a dummy variable and an orthogonal transformation.*

Here the equivalence means that the distance among samples under different projections are the same[1]. Various projections can be obtained by applying orthogonal transformations on an existing projection or by increasing the dimensionality of the inputs by adding dummy variables which has 0 as their only value. We first present two lemmas, which pave the way for the proof. The first lemma tells that 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 all distinct, the kernel matrix K is of full rank.*

Proof of the lemma can be found in [Micchelli, 1984]. The Lemma tells that when RBF kernel function is used, as long as the samples are different, K will be of full rank, which means that the samples will be linearly independent in the feature space induced by K. In real applications, it is usually sensible to assume that the given samples are all distinct, therefore when RBF kernel function is used, the linear independent assumption will always hold. Let the compact SVD [Golub and Van Loan, 1996] of K_c be $K_c = U_1\Sigma_t U_1^T$, and the full SVD of $U_1^T Y$ be $U_1^T Y = P\Sigma_b Q$, we have:

Lemma 2 *When samples are linearly independent in the kernel induced feature space, we have $\Sigma_b^2 = diag(1, \ldots, 1, 0)$.*

Proof: Let $S_w^K = S_t^K - S_b^K = K_c(I - \frac{1}{n}YY^T)K_c$. $I - n^{-1}YY^T$ is positive semidefinite (psd) since:

$$\mathbf{x}^T\left(I - \frac{1}{n}YY^T\right)\mathbf{x} = \sum_{j=1}^{c}\left(\sum_{i=1}^{n_j}\left(x_i^j\right)^2 - \frac{1}{n_j}\left(\sum_{i=1}^{n_j}x_i^j\right)^2\right).$$

This means S_w^K is also psd. Let G be a matrix defined as:

$$G = U\begin{pmatrix}\Sigma_t^{-1}P & 0\\ 0 & I\end{pmatrix}, U = [U_1, U_2], \qquad (11)$$

where U consists of the whole set of singular vectors of K_c, and U_2 contains the singular vectors corresponding to the zero singular values. Under the assumption, we know that rank(K)=n and rank(K_c)=$n-1$. Therefore U_2 contains only one column: $U_2 = [(1/\sqrt{n})\mathbf{1}]$. With the definition of G, it can be shown that, $G^T S_t^K G = \text{diag}(I, 0)$, $G^T S_b^K G = \text{diag}\left(\Sigma_b^2, 0\right)$ and $G^T S_w^K G = \text{diag}(\Sigma_w, 0)$, therefore we have: diag $(I, 0)$ = diag$(\Sigma_w, 0)$ + diag$(\Sigma_b, 0)$. Since S_w^K and S_b^K are all positive semidefinite, the diagonal elements of Σ_w and Σ_b must be nonnegative. It is easy to verify that rank$(S_b^K) \le c-1$ and rank$(S_w) \le n - c$. Therefore we have: rank(diag$(I, 0)$) $\le$ rank(diag$(\Sigma_w, 0)$) + rank(diag$(\Sigma_b, 0)$)=$n - 1$. Thus, we can conclude that all nonzero diagonal entries in Σ_b are 1. Note that rank$(U_1^T Y)$ = rank$(Y) = c - 1$ since U_1^T contains orthonormal columns, thus the number of nonzero diagonal entries in Σ_b is $c - 1$. This completes the proof. ■

We are now ready to prove the theorem, which establishes the equivalence between KDA and PKDA.

Proof of Theorem 1: We can show that for any input $\phi(x_*)$, $m_l(x_*) = \mathbf{k}_*^T(K_c + \lambda_l I)^{-1}\mathbf{y}_l$. Therefore, when $\lambda_l \to 0$, the expectation of the predictive distributions in PKDA actually projects data with a transformation matrix defined as $\hat{B} = (K_c)^+ Y$. The equivalence between KDA and PKDA can be established by studying the relationship between $\hat{B}$ and B. We first study the structure of matrix $\hat{B}$. $\hat{B} = K_c^+ Y = U_1\Sigma_t^{-1}U_1^T Y$. Recall that $U_1^T Y = P\Sigma_b Q$. Let $P = [\mathbf{p}_1, \ldots, \mathbf{p}_c]$. We have:

$$\hat{B} = U_1\Sigma_t^{-1}[\mathbf{p}_1, \ldots, \mathbf{p}_c]\Sigma_b Q. \qquad (12)$$

Next we see the structure of B can be expressed 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 $\left(S_t^K\right)^+ S_b^K$ are given by the first $c-1$ columns of $U_1\Sigma_t^{-1}P$. Therefore we have:

$$B = U_1\Sigma_t^{-1}[\mathbf{p}_1, \ldots, \mathbf{p}_{c-1}]. \qquad (13)$$

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

3.1 Efficient Model Selection for PKDA

Theorem 1 establishes the connection between PKDA and KDA. Given a kernel K and the class label vector $\mathbf{y}$, PKDA projects test points onto the reduced space where the data can be well separated. Below we show how to automatically determine the proper value for the regularization parameters (or the noise terms) by minimizing the negative log marginal likelihood. Denote $\tilde{K} = K_c + \lambda I$, where $\lambda \ge 0$. For each $\mathcal{GP}$ in PKDA model, its negative log marginal likelihood is:

$$-\log P\left(\mathbf{y}|\tilde{K}\right) = \frac{1}{2}\mathbf{y}^T\tilde{K}^{-1}\mathbf{y} + \frac{1}{2}\log\left|\tilde{K}\right| + \frac{n}{2}\log 2\pi. \qquad (14)$$

Minimizing the negative log marginal likelihood specifies the following nonlinear constrained optimization problem:

$$\min_{\lambda} -\log P\left(y|K_c + \lambda I\right) \quad st. \quad \lambda \ge 0. \qquad (15)$$

[1]This definition for equivalence is sensible , since in many classification and clustering approaches, such as the SVM and k-means, only the distance among samples are used to fit model.

Let $K_c = U\Sigma U^T$, $U = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$, $y_i = \mathbf{u}_i^T \mathbf{y}$ and Σ =diag$(\alpha_1, \ldots, \alpha_N)$. We have:

$$-\log P\left(y|\tilde{K}\right) \propto \frac{1}{2}\sum_{i=1}^{N}\left(\frac{\hat{y}_i^2}{\alpha_i+\lambda} - \log\frac{1}{\alpha_i+\lambda}\right). \quad (16)$$

In the equation, the constant term $\frac{n}{2}\log 2\pi$ is ignored. The equation shows that the optimization problem specified in Equation (15) is a DC (difference of convex functions) Programming [Horst and Thoai, 1999] problem, and efficient techniques such as the DC algorithm (DCA) [An and Tao, 2005] can be applied. To minimize Equation (16), we define:

$$f(\lambda) = -\log P\left(y|\tilde{K}\right) = g(\lambda) - h(\lambda), \quad (17)$$

where

$$g(\lambda) = \frac{1}{2}\sum_{i=1}^{N}\left(\frac{\hat{y}_i^2}{\alpha_i+\lambda}\right) + \frac{n}{2}\log 2\pi \quad (18)$$

$$h(\lambda) = \frac{1}{2}\sum_{i=1}^{N}\left(\log\frac{1}{\alpha_i+\lambda}\right). \quad (19)$$

Let $g^*(z) = \sup_\lambda \{z\lambda - g(\lambda)\}$ be the conjugate function of $g(\lambda)$ and ∂ be the subdifferential operator, it is easy to see that the following equations for $\partial h(\lambda)$ and $\partial g^*(z)$ hold:

$$\frac{\partial}{\partial\lambda}h(\lambda) = -\frac{1}{2}\sum_{i=1}^{N}\frac{1}{\alpha_i+\lambda}, \quad (20)$$

$$\partial g^*(z) = \arg\max_\lambda\{z\lambda - g(\lambda)\}. \quad (21)$$

Equation (21) can be solved by minimizing the convex function $g(\lambda) - z_1\lambda$, subject to the nonnegative constrains on λ. The above equations, can be used in DCA proposed in [An and Tao, 2005] to compute solutions for the optimization problem specified in Equation (16). In our experiments we found that DCA usually returns globally optimal solutions.

The model parameters of PKDA can also be tuned by optimizing the leave-one-out log predictive probability [Rasmussen and Williams, 2006] in a similar way.

Time Complexity of PKDA: It turns out that the model selection step for PKDA does not increase the computation cost significantly, since the SVD of K_c generated in the model selection step is also used in the subsequent prediction step. We provide the complexity analysis for PKDA. Assume there are n training and m test samples. Calculating the SVD for K_c has a cost of $O(n^3)$. The results from SVD will be used for model selection and the prediction of the test points. DC-Programming can usually be solved in $O(n^3)$ operations. Given the SVD of K_c, computing its inverse costs $O(n^3)$ operations and the projection step for the m test samples costs $O(cmn^2)$ operations. Therefore, the total time complexity of PKDA is $O(\max(n, cm)n^2)$. The complexities of PKDA and KDA are of the same order.

4 Empirical Study

In this section, we empirically evaluate the performance of PKDA. Nine data sets are used in the experiments. They are: two image data sets, AR10P[2] and ORL10P[3]; five UCI data sets [Murphy and Aha, 1994]: DNA, SPLICE, SOLAR, SOYBEAN and ADVERTISE; and two text data sets from the 20-news-group data: TXT4C and TXT2C[4]. The details of the benchmark data sets are summarized in Table 1. The last column of Table 1 lists the rank of the kernels obtained on each data set using the RBF kernel. It shows that all kernels are of full rank, that is, samples are linearly independent in the kernel-induced space. The results suggest that the condition assumed in Theorem 1 is indeed very mild.

Dataset	Inst	Dim	Classes	rank(K)
AR10P	130	2400	10	130
ORL10P	100	10304	10	100
TXT4C	3933	8298	4	3933
TXT2C	1425	4322	2	1425
DNA	3001	180	3	3001
SPLICE	3005	60	3	3005
SOLAR	158	12	6	158
SOYBEAN	531	35	15	531
ADVERTISE	800	1558	2	800

Table 1: Statistics of the benchmark data sets.

In the experiment, the parameters in PKDA are obtained in two ways: using the proposed automatic tuning mechanism (PKDA-OP) and using cross-validation (PKDA-CV). We compare PKDA with KDA, whose parameters are also obtained in two ways: using a default value (λ=0, KDA) and using cross-validation (KDA-CV). To evaluate the qualities of the dimensionality-reduced spaces found by different approaches, we train a 1-nearest-neighbor (1-nn) classifier in each space, and use its accuracy to determine how well the samples from different classes are separated in that space. A higher accuracy suggests a higher quality of the space in terms of sample separability. We also train a support vector machine (SVM) [Vapnik, 1995] using the given kernel and record its accuracy. Note that SVM is a classifier rather than a dimension reduction approach. We include SVM in the experiment to verify whether the accuracy of the 1-nn obtained in the dimensionality-reduced space is reasonably good. The parameter of the SVM is obtained in two ways: using a default parameter (C=1, SVM) and using cross-validation (SVM-CV). We implemented PKDA and KDA in Matlab and LibSVM [Chang and Lin, 2001] is used for SVM. We use the RBF kernel function to construct the kernels. Each algorithm is tested for 25 times on each data set by randomly sampling (at most) 200 instances from each class and we split the data into training and test sets of a ratio 4:1. The obtained averaged accuracy rates are presented in the paper. For PKDA-CV, KDA-CV and SVM-CV we apply 5-fold cross-validation on the training data. For CV based model selection, choosing the candidate parameter values is a difficult problem. In the

[2]http://rvl1.ecn.purdue.edu/ aleix/aleix face DB.html. Data set is subsampled down to the size of 60×40 = 2400

[3]http://www.uk.research.att.com/facedatabase.html. Data set is subsampled down to the size of 100×100 = 10000

[4]TXT4C: Baseball, Hockey, PC and MAC; TXT2C: Religion and Atheism. http://people.csail.mit.edu/jrennie/20Newsgroups/

experiments, to improve the probability that there is at least one good regularization parameter value for each CV based algorithm on each data set, we picked 20 candidates from a wide range between 0 and 10^4.

4.1 Experimental Results

Below we present the experimental results comparing different approaches on the quality of the dimensionality-reduced spaces they generated, and their efficiency.

Accuracy Comparison

Table 2 presents the accuracy of different approaches on the nine benchmark data sets. Based on the accuracy results, we summarize the following observations.

First, compared to KDA using the default parameter value $\lambda = 0$, PKDA-OP performs much better. On 7 out of 9 data sets, PKDA-OP performs significantly better than KDA. We observed that on several data sets, such as the DNA and the SOYBEAN data, the performance of KDA is poor. This is likely due to overfitting. This suggests that it is necessary to apply regularization to make KDA more robust.

Second, when comparing PKDA-OP with PKDA-CV, we observed that the two approaches perform equally well. Compared to cross-validation, the automatic tuning mechanism does not need to run PKDA multiple times on different splits of the training data, therefore it is more efficient. In our experiments, we observed that the computational cost of PKDA-OP is usually comparable to that of KDA and is significantly lower than that of KDA-CV and PKDA-CV. It is known that picking the right candidate parameter values in cross-validation may not be easy. The proposed tuning process can also automatically find good values for the parameters.

Third, in comparison with KDA-CV, we found that PKDA is significantly better on 4 out of 9 data sets. We also noticed that comparing KDA with KDA-CV, KDA-CV performs significantly better, which verifies the effectiveness of cross-validation as well as the regularization.

Finally, comparing PKDA-OP with SVM-CV, we observed that by using the dimensionality-reduced spaces generated by PKDA-OP, the 1-nn classifier is able to achieve accuracy rates comparable to those of the SVM with cross-validation. This result clearly shows the high quality of the dimensionality-reduced spaces generated by PKDA-OP.

Overall, on the 9 benchmark data sets, PKDA-OP+1nn achieved the best average accuracy of 0.88 which is followed by SVM-CV (0.87), PKDA-CV+1nn (0.87), SVM (0.84), KDA-CV+1nn (0.83) and KDA+1nn (0.67).

Efficiency Comparison

On the nine benchmark data sets, the averaged running time of PKDA-OP, KDA PKDA-CV and KDA-CV are: 1.58s, 1.27s, 114.63s and 114.04s, respectively. PKDA-OP has almost the same running time as KDA, while having the advantage of being able to automatically tune its model parameters. Figure 2 plots the running time of PKDA-OP, KDA, PKDA-CV, KDA-CV on five benchmark data sets, on which algorithms have relatively longer running time than on the other four benchmark data sets. The plots in the figure show that compared with cross-validation, the automatic tuning mechanism is significantly faster. We observed similar trends on the other four data sets. The results on efficiency comparison is consistent with our complexity analysis for PKDA.

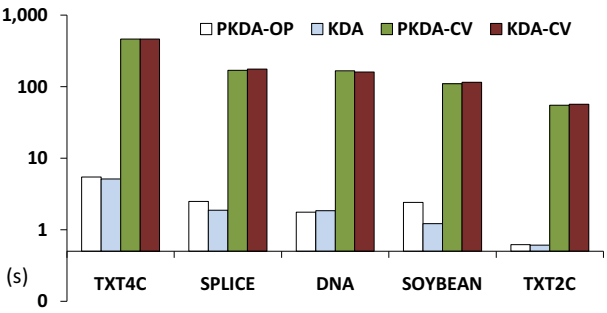

Figure 2: Efficiency comparison on five data sets. y-axis is for running time in logarithmic scale measured by seconds.

The results demonstrated that the proposed multiclass probabilistic model for KDA is robust and efficient.

5 Conclusion

In this paper, we proposed a probabilistic model for KDA, which is able to handle multiclass problems. The proposed model is based on Gaussian Process and its model parameters can be automatically tuned in an efficient way. Experimental results demonstrated that the proposed model offers good performance, and is very efficient. Based on Gaussian Process, the proposed model can handle large scale data via approximation methods, such as BCM [Tresp, 2000] and SPGP [Snelson and Ghahramani, 2006]. Also as a probabilistic model, PKDA allows the integration of multiple heterogeneous data sources via Bayesian mixture models [Svensen and Bishop, 2005], which leads to an interesting nonlinear kernel combination formulation: Mixture of Discriminant Gaussian Process (MPKDA). Our preliminary results show that MPKDA outperforms the existing linear kernel combination approaches [Ye *et al.*, 2008]. This forms one line of our ongoing research work.

Acknowledgment

This work was supported by NSF IIS-0612069, IIS-0812551, CCF-0811790, NIH R01-HG002516, and NGA HM1582-08-1-0016.

References

[An and Tao, 2005] Le Thi Hoai An and Pham Dinh Tao. The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. *Annals of Operations Research*, 133:23–46, 2005.

[Belhumeur *et al.*, 1997] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[Centeno and Lawrence, 2006] Tonatiuh Pena Centeno and Neil D. Lawrence. Optimising kernel parameters and regu-

Dataset	PKDA-OP	PKDA-CV		KDA		KDA-CV		SVM		SVM-CV	
	Acc	Acc	p-val	Acc	p-val	Acc	p-val	Acc	p-val	Acc	p-val
AR10P	0.95	0.95	1.00	0.96	0.41	0.96	0.41	0.77	0.00 +	0.91	0.01 +
ORL10P	0.96	0.96	1.00	0.95	0.24	0.95	0.24	0.91	0.04 +	0.95	0.24
TXT4C	0.91	0.91	1.00	0.89	0.05 +	0.89	0.04 +	0.89	0.05 +	0.90	0.06 +
TXT2C	0.84	0.84	0.24	0.78	0.02 +	0.79	0.14	0.83	0.34	0.85	0.19
DNA	0.92	0.92	0.37	0.39	0.00 +	0.74	0.00 +	0.91	0.01 +	0.91	0.01 +
SPLICE	0.86	0.87	0.36	0.41	0.00 +	0.79	0.00 +	0.88	0.06 -	0.88	0.06 -
SOLAR	0.62	0.62	0.92	0.56	0.00 +	0.63	0.72	0.63	0.68	0.64	0.06 -
SOYBEAN	0.90	0.92	0.01 -	0.29	0.00 +	0.89	0.11	0.93	0.00 -	0.93	0.00 -
ADVERTISE	0.91	0.87	0.01 +	0.76	0.00 +	0.85	0.00 +	0.83	0.00 +	0.91	1.00
Ave.	0.88	0.87		0.67		0.83		0.84		0.87	
Win/Loss		1/1		7/0		4/0		5/2		3/3	

Table 2: The accuracy achieved by algorithms on benchmark datasets. PKDA-OP denotes the PKDA with the automatic tune process and PKDA-CV, KDA-CV and SVM-CV denotes PKDA, KDA and SVM with cross-validation. The p-val of each algorithm is generated by comparing its accuracy with PKDA-OP using a two-tailed t-Test. The symbols "+" and "-" identify statistically significant (at 0.10 level) if PKDA-OP wins over or loses to the compared algorithm, respectively. Note that SVM is a classification algorithm, rather than a dimension reduction algorithm.

larisation coefficients for non-linear discriminant analysis. *J. Mach. Learn. Res.*, 7:455–491, 2006.

[Chang and Lin, 2001] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[Friedman, 1989] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

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

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

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[Horst and Thoai, 1999] R. Horst and N. V. Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 130:1–41, 1999.

[Howland *et al.*, 2003] P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications,*, 25:165–179, 2003.

[Ioffe, 2006] Sergey Ioffe. Probabilistic linear discriminant analysis. In *ECCV*, 2006.

[Lanckriet *et al.*, 2004] 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.

[McLachlan, 1992] Geoffrey J. McLachlan. *Discriminant analysis and statistical pattern recognition*. Wiley, 1992.

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

[Mika *et al.*, 1999] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Mullers. Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Signal Processing Society Workshop*, 1999.

[Murphy and Aha, 1994] P.M. Murphy and D.W. Aha. UCI repository of machine learning databases. 1994.

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

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

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

[Snelson and Ghahramani, 2006] E. Snelson and Z Ghahramani. Sparse gaussian processes using pseudo-inputs. In *NIPS 18*, The MIT Press, 2006.

[Snelson, 2007] E. Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2007.

[Svensen and Bishop, 2005] Markus Svensen and Christopher M. Bishop. Robust bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2005.

[Tresp, 2000] V. Tresp. A bayesian committee machine. *Neural Computation*, 12:2719–2741, 2000.

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

[Ye *et al.*, 2008] Jieping Ye, Shuiwang Ji, and Jianhui Chen. Multi-class discriminant kernel learning via convex programming. *J. Mach. Learn. Res.*, 9:719–758, 2008.

[Ye, 2007] Jieping Ye. Least squares linear discriminant analysis. In *ICML*, 2007.