Neural Networks 22 (2009) 977–987

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A novel kernel-based maximum a posteriori classification method

Zenglin Xu[a], Kaizhu Huang[b], Jianke Zhu[a], Irwin King[a,*], Michael R. Lyu[a]

[a] *Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong*

[b] *Department of Engineering Mathematics, University Of Bristol, Bristol, BS8 1TR, United Kingdom*

ARTICLE INFO

Article history:
Received 19 March 2008
Revised and accepted 20 November 2008

Keywords:
Kernel methods
Maximum a posteriori
Discriminant analysis

ABSTRACT

Kernel methods have been widely used in pattern recognition. Many kernel classifiers such as Support Vector Machines (SVM) assume that data can be separated by a hyperplane in the kernel-induced feature space. These methods do not consider the data distribution and are difficult to output the probabilities or confidences for classification. This paper proposes a novel Kernel-based Maximum A Posteriori (KMAP) classification method, which makes a Gaussian distribution assumption instead of a linear separable assumption in the feature space. Robust methods are further proposed to estimate the probability densities, and the kernel trick is utilized to calculate our model. The model is theoretically and empirically important in the sense that: (1) it presents a more generalized classification model than other kernel-based algorithms, e.g., Kernel Fisher Discriminant Analysis (KFDA); (2) it can output probability or confidence for classification, therefore providing potential for reasoning under uncertainty; and (3) multi-way classification is as straightforward as binary classification in this model, because only probability calculation is involved and no one-against-one or one-against-others voting is needed. Moreover, we conduct an extensive experimental comparison with state-of-the-art classification methods, such as SVM and KFDA, on both eight UCI benchmark data sets and three face data sets. The results demonstrate that KMAP achieves very promising performance against other models.

1. Introduction

Kernel methods play an important role in machine learning and pattern recognition (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). They have achieved success in almost all traditional tasks of machine learning, i.e., supervised learning (Mika, Ratsch, Weston, Scholkopf, & Muller, 1999; Vapnik, 1998), unsupervised learning (Schölkopf, Smola, & Müller, 1998), and semi-supervised learning (Chapelle, Schölkopf, & Zien, 2006; Xu, Jin, Zhu, King, & Lyu, 2008; Xu, Zhu, Lyu, & King, 2007; Zhu, Kandola, Ghahramani, & Lafferty, 2005). We focus here on kernel methods for supervised learning, where the basic idea is to use the so-called kernel trick to implicitly map the data from the ordinal input space to a high dimensional feature space, in order to make the data more separable. Usually, the aim of kernel-based classifiers is to find an optimal linear decision function in the feature space, based on certain criteria. The optimal linear decision hyperplane could be, for example, the one that can maximize the margin between two different classes of data (as used in the Support Vector Machine (SVM) (Vapnik, 1998)), or the one that minimizes the within-class covariance and at the same time maximizes the between-class covariance (as used in the Kernel Fisher Discriminant Analysis (KFDA) (Mika et al., 1999, 2003)), or the one that minimizes the worst-case accuracy bound (as used in the Minimax Probability Machine (Huang, Yang, King, & Lyu, 2004; Huang, Yang, King, Lyu, & Chan, 2004; Huang, Yang, Lyu, & King, 2008; Lanckriet, Ghaoui, Bhattacharyya, & Jordan, 2002)).

These kernel methods usually achieve higher prediction accuracy than their linear forms (Schölkopf & Smola, 2002). The reason is that the linear discriminant functions in the feature space can represent complex separating surfaces when mapped back to the original input space. However, one drawback of standard SVM is that it does not consider the data distribution and cannot properly output the probabilities or confidences for the resultant classification (Platt, 1999; Wu, Lin, & Weng, 2004). One needs special transformation in order to output probabilities. Therefore, it takes a lot of extra effort in order to be applied in systems that contain inherent uncertainty. In addition, the linear discriminant function can only separate two classes. For multi-category problems, we may resort to approaches such as one-against-one or one-against-others to vote on which class should be assigned (Hsu & Lin, 2002).

One approach to obtaining classification probabilities is to use a statistical pattern recognition technique, in which the probability

[*] Corresponding author. Tel.: +852 2609 8398; fax: +852 2603 5024.
E-mail addresses: zlxu@cse.cuhk.edu.hk (Z. Xu), k.huang@bristol.ac.uk (K. Huang), jkzhu@cse.cuhk.edu.hk (J. Zhu), king@cse.cuhk.edu.hk (I. King), lyu@cse.cuhk.edu.hk (M.R. Lyu).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.11.005

density function can be derived from the data. Future items of data can then be classified using a Maximum A Posteriori (MAP) method (Duda, Hart, & Stork, 2000). One typical probability estimation method is to assume multivariate normal density functions over the data. The multivariate normal density functions are easy to handle; moreover some problems can also be regarded as Gaussian problems if there are enough examples, although in practice the Gaussian distribution cannot be easily satisfied in the input space.

To solve these problems, in this paper we propose a Kernel-based Maximum A Posteriori (KMAP) classification method under a Gaussianity assumption in the feature space. With this assumption, we derive a non-linear discriminant function in the feature space, in contrast to current kernel-based discriminant methods that rely only on using an assumption of linear separability for the data. Moreover, the derived decision function can output the probabilities or confidences. In addition, the distribution can be very complex in the original input space when it is mapped back from the feature space. This is analogous to the case in which a hyperplane derived with KFDA or SVM in the feature space could lead to a complex surface in the input space. Therefore, this approach sets a more valid foundation than the traditional multivariate probability estimation methods that are usually conducted in the input space.

Generally speaking, distributions other than the Gaussian function can also be assumed in the feature space. However, under a distribution with a complex form, it is hard to get a closed-form solution and easy to over-fit. More importantly, with the Gaussian assumption, a kernelized version can be derived without knowing the explicit form of the mapping functions for our model, while it is still difficult to formulate the kernel version for other complex distributions.

It is important to relate our proposed model to other probabilistic kernel methods. Kernel-based exponential methods (Canua & Smola, 2006) use parametric exponential families to explicitly build mapping functions from the input space to the feature space. It is also interesting to discuss the Kernel Logistic Regression (KLR) (Zhu & Hastie, 2005), which employs the logistic regression to estimate the density function and still leads to a linear function in the kernel-induced feature space. The kernel-embedded Gaussian mixture model in Wang, Lee and Zhang (2003) is related to our model in that a similar distribution is assumed, but their model is restricted to clustering and cannot be directly used in classification.

The appealing features of KMAP are summarized as follows. First, one important feature of KMAP is that it can be regarded as a more generalized classification model than KFDA and other kernel-based algorithms. KMAP provides a rich class of family of kernel-based algorithms, based on different regularization implementations. Another important feature of KMAP is that it can output the probabilities of assigning labels to future data, which can be seen as the confidences of decisions. Therefore, the proposed method can also be seen as a Bayesian decision method, which can further be used in systems that make an inference under uncertainty (Smith, 1988). Moreover, multi-way classification is as easy as binary classification in this model because only probability calculation is involved and no one-against-one or one-against-others voting is needed. As shown in Section 2.4, KMAP has the time complexity $\mathcal{O}(n^3)$ (where n is the cardinality of data), which is in the same order as that of KFDA. In addition, the decision function enjoys the property of sparsity: only a small number of eigenvectors are needed for future prediction. This leads to low storage complexity.

The proposed algorithm can be applied in many pattern recognition tasks, e.g., face recognition, character recognition, and others. In order to evaluate the performance of our proposed method, extensive experiments are performed on eight benchmark data sets from the UCI repository and on three standard face data sets. Experimental results show that our proposed method achieves very competitive performance on UCI data. Moreover, its advantage is especially prominent in face data sets, where only a small amount of training data are available.

The remainder of this paper is organized as follows. In Section 2, we derive the kernel-based MAP classification model in the feature space and discuss the parameter estimation techniques. Then the kernel calculation procedure and the theoretical connections between the KMAP model and other kernel methods are discussed. Section 3 first reports the experiments on UCI data sets against other competitive kernel methods, then evaluates our model's performance on face data sets. Section 4 draws conclusions and lists possible future research directions.

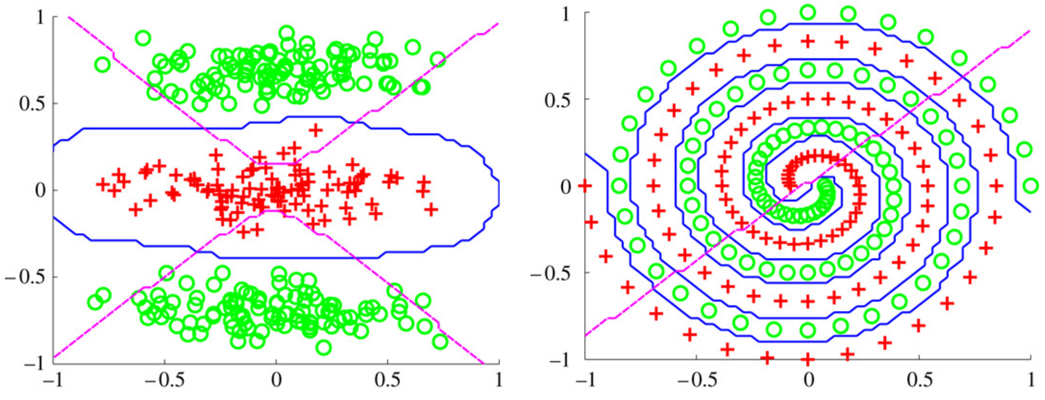
We use the following notation. Let $\mathcal{X} \in \mathbb{R}^d$ denote the original d-dimensional input space, where an instance $\mathbf{x}$ is generated from an unknown distribution. Let $\mathbb{C} = \{1, 2, \ldots, m\}$ be the set of labels where m is the number of classes. Let $P(C_i)$ denote the prior probability of class C_i. Let n_i be the number of observed data points in class C_i and n be the amount of training data. A Mercer kernel is defined as a symmetric function κ, such that $\kappa(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, where Φ is a mapping from $\mathcal{X}$ to a feature space $\mathcal{H}$. The form of kernel function κ could be a linear kernel function, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$, a Gaussian RBF kernel function, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2)$, or a polynomial kernel function, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$, for some σ and p respectively. A kernel matrix or Gram matrix $G \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix such that $G_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$. G can be further written as $[G^{(1)}, G^{(2)}, \ldots, G^{(m)}]$, where $G^{(i)}$ is an $n \times n_i$ matrix and denotes the subset of G relevant to class C_i. The covariance matrix of $G^{(i)}$ is denoted by $\Sigma_{G^{(i)}}$. We denote μ_i and Σ_i as the mean vector and covariance matrix of class C_i in the feature space, respectively. The set of eigenvalues and the set of eigenvectors belonging to Σ_i are represented as Λ_i and Ω_i. We write $p(\Phi(\mathbf{x})|C_i)$ as the probability density function of class C_i.

2. Kernel-based maximum a posteriori classification

In contrast with the assumption of traditional MAP algorithms, that the data points satisfy multivariate normal distribution in the input space, we assume that the mapped data in the high dimensional feature space follow such a distribution. This is meaningful in that the distribution can be very complex in the original input space when the Gaussian distribution is mapped back from the kernel-induced feature space. In the same sense, the decision boundary can be more complex when the quadratic decision boundary is projected into the input space.

In order to make a clear illustration of the reasonability of the Gaussian distribution in the kernel-induced feature space, two synthetic data sets, **Relevance** and **Spiral**, are used in this paper. We draw the decision boundary of discriminant functions conducted in the input space and the feature space, respectively. **Relevance** is a data set where only one dimension of the data is relevant to separate the data. **Spiral** can only be separated by highly non-linear decision boundaries. Fig. 1 plots the boundaries of the discriminant functions for the traditional MAP algorithm and the kernel-based MAP algorithm on these two data sets.

It can be observed that the MAP classifier with the Gaussian distribution assumption in the kernel-induced feature space always produces more reasonable decision boundaries. For **Relevance** data, a simple quadratic decision boundary in the input space cannot produce good prediction accuracy. However, the kernel-based MAP classifier separates these two classes of data smoothly. The difference between the boundaries of these two algorithms is especially significant for **Spiral**. This indicates that the kernel-based MAP classification algorithm can better fit the distribution of data points through the kernel trick.



(a) Relevance.

(b) Spiral.

Fig. 1. The decision boundaries on **Relevance** and **Spiral**. The separating lines were obtained by projecting test data over a grid. The lines in blue (dark) and magenta (dashed) represent decision boundaries for MAP algorithms with Gaussian distribution in the feature space and those in the input space, respectively.

2.1. Model formulation

Under the Gaussian distribution assumption, the conditional density function for each class $C_i (1 \le i \le m)$ is written as:

$$p(\Phi(\mathbf{x})|C_i) = \frac{1}{(2\pi)^{N/2}|\Sigma_i|^{1/2}} \times \exp\left\{-\frac{1}{2}(\Phi(\mathbf{x})-\mu_i)^{\mathrm{T}}\Sigma_i^{-1}(\Phi(\mathbf{x})-\mu_i)\right\}, \quad (1)$$

where N is the dimension of the feature space and $|\Sigma_i|$ is the determinant of the covariance matrix Σ_i. It is important to note that N could be infinite for the RBF kernel function. In such case, we seek to apply the kernel trick to avoid directly computing the density function, which will be verified in Section 2.3.

Taking logs on both sides of Eq. (1) and removing the constants, we can get the Mahalanobis distance function of a data point ($\mathbf{x}_i$) to the class center (μ_i) in the feature space when each class has the same prior probability:

$$g_i(\Phi(\mathbf{x})) = (\Phi(\mathbf{x})-\mu_i)^{\mathrm{T}}\Sigma_i^{-1}(\Phi(\mathbf{x})-\mu_i) + \log|\Sigma_i|. \quad (2)$$

In the case that different class prior probabilities are assumed, we only need to subtract $2\log P(C_i)$ in the above equation. The intuitive meaning of the function is that a data point is more likely to be assigned to a certain class with a lower Mahalanobis distance between the data point and the class center.

We revert the Mahalanobis distance function to its original class conditional density function: $p(\Phi(\mathbf{x})|C_i) = \frac{1}{(2\pi)^{N/2}}\exp(-\frac{1}{2}g_i(\Phi(\mathbf{x})))$. According to the Bayesian Theorem, the posterior probability of class C_i is calculated by

$$P(C_i|\Phi(\mathbf{x})) = \frac{p(\Phi(\mathbf{x})|C_i)P(C_i)}{\sum_{j=1}^{m} p(\Phi(\mathbf{x})|C_j)P(C_j)}. \quad (3)$$

Based on Eq. (3), the decision rule can be formulated as below:

$$\mathbf{x} \in C_w \quad \text{if } P(C_w|\Phi(\mathbf{x})) = \max_{1\le j\le m} P(C_j|\Phi(\mathbf{x})). \quad (4)$$

This means that a test data point will be assigned to the class with the maximum of $P(C_w|\Phi(\mathbf{x}))$, i.e., the MAP. Since the MAP is calculated in the kernel-induced feature space, the output model is named as the KMAP classification.

Eq. (3) is of importance because it shows that KMAP output not only a class label, but also the probability of a data point belonging to a class. This probability can thus be seen as the confidence of classification of new data points. It can be used in statistical systems that make an inference under uncertainty (Smith, 1988). If the confidence is lower than some specified threshold, the system can refuse to make an inference. This is a distinct advantage over many kernel learning methods, including SVM, which cannot easily output these probabilities.

2.2. Parameter estimation

In order to compute the Mahalanobis distance function, the mean vector and the covariance matrix for each class must be estimated. Typically, the mean vector (μ_i) and the within-covariance matrix (Σ_i) are calculated by a maximum likelihood estimation. In the feature space, they are formulated as follows:

$$\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\Phi(\mathbf{x}_j), \quad (5)$$

$$\Sigma_i = S_i = \frac{1}{n_i}\sum_{j=1}^{n_i}(\Phi(\mathbf{x}_j)-\mu_i)(\Phi(\mathbf{x}_j)-\mu_i)^{\mathrm{T}}. \quad (6)$$

Directly employing the maximum likelihood estimation S_i as the covariance matrix will generate quadratic discriminant functions in the feature space. However, the covariance estimation problem is clearly ill-posed, because the number of data points in each class is usually much smaller than the number of dimensions in the kernel-induced feature space. This problem is especially obvious in face recognition tasks. The treatment of this ill-posed problem is to introduce regularization. There are several kinds of regularization methods. One of them is to replace the individual within-covariance matrices with their average, i.e.,

$$\Sigma_i = \frac{\sum_{i=1}^{m} S_i}{m} + rI, \quad (7)$$

where I is the identity matrix and r is a regularization coefficient. This method is able to substantially reduce the number of free parameters to be estimated. Moreover, it also reduces the discriminant function between two classes to a linear one. Therefore, a linear discriminant analysis method can be obtained. We will discuss its connection to Kernel Fisher Discriminant Analysis (KFDA) in Section 2.4.

Alternatively, we can estimate the covariance matrix by combining the above linear discriminant function with the quadratic one. Instead of estimating the covariance matrix in the input space (Friedman, 1989), we can apply this method in the kernel-induced feature space. After the data are centered (see Schölkopf et al. (1998) for centering data), the formulation in the feature space is as follows:

$$\Sigma_i = (1-\eta)\tilde{\Sigma}_i + \eta\frac{trace(\tilde{\Sigma}_i)}{n}I, \tag{8}$$

where

$$\tilde{\Sigma}_i = (1-\theta)S_i + \theta\tilde{S}, \tag{9}$$

$$\tilde{S} = \frac{1}{n}\sum_{l=1}^{n}\Phi(\mathbf{x}_l)\Phi(\mathbf{x}_l)^{\mathrm{T}}. \tag{10}$$

In the equations, θ ($0 \le \theta \le 1$) is a coefficient linked with the linear and quadratic discriminant term. Also, η ($0 \le \eta \le 1$) determines the shrinkage to a multiple of the identity matrix. Note that the formulation of Eq. (10) differs from the one in Friedman (1989), where $S = \sum_{i=1}^{m} S_i$. This is because it is more accurate to estimate the covariance from all samples rather than only from those belonging to a single class. The effect is particularly significant in face recognition, where the sample size is relatively small and the dimensionality of the feature space is quite high. Because of this, our approach is more capable of adjusting the effect of the regularization.

Remark. Other regularization methods can also be employed for estimating the covariance matrices. The criteria of selecting the regularization are based on specific applications of KMAP. For example, when the number of training samples is small, it is better to use the regularization method based on Eq. (8).

2.3. Kernel calculation

It is critical to represent the above formulations in a kernelized or inner product form. In the following, we demonstrate how the KMAP formulations can be kernelized without knowing the explicit form of the mapping functions.

Obviously, Eq. (2) is poorly-posed, since we are estimating the means and covariance matrices from n samples. To avoid this problem in calculating the Mahalanobis distance function, the spectral representation of the covariance matrix, i.e., $\Sigma_i = \sum_{j=1}^{N}\Lambda_{ij}\Omega_{ij}\Omega_{ij}^{\mathrm{T}}$ (where Λ_{ij} and Ω_{ij} are the jth eigenvalue and eigenvector of Σ_i, respectively), is utilized instead of a direct calculation (Ruiz & Lopez-de Teruel, 2001). The small eigenvalues will, in particular, drastically degrade the performance of the function overwhelmingly, because they are underestimated due to the small number of examples. In this paper, we only estimate the k largest eigenvalues and replace each remaining eigenvalue with a nonnegative number h_i. This technique is similar to that used in Principal Component Analysis (PCA) (Jolliffe, 1986), except that the non-principal eigenvalues are replaced by a constant h_i. Thus Eq. (2) can be reformulated as follows:

$$g_i(\Phi(\mathbf{x})) = \sum_{j=1}^{k}\frac{1}{\Lambda_{ij}}[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2 + \sum_{j=k+1}^{N}\frac{1}{h_i}[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2 + \log\left(h_i^{N-k}\prod_{j=1}^{k}\Lambda_{ij}\right).$$

In the above equation, $g_i(\Phi(\mathbf{x}))$ can further be represented as follows:

$$\frac{1}{h_i}\left(\sum_{j=1}^{N}[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2 - \sum_{j=1}^{k}\left(1-\frac{h_i}{\Lambda_{ij}}\right)[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2\right).$$

We define $g_{1i}(\Phi(\mathbf{x})) = \sum_{j=1}^{N}[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2$ and $g_{2i}(\Phi(\mathbf{x})) = \sum_{j=1}^{k}(1-\frac{h_i}{\Lambda_{ij}})[\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)]^2$, such that

$$g_i(\Phi(\mathbf{x})) = \frac{1}{h_i}[g_{1i}(\Phi(\mathbf{x})) - g_{2i}(\Phi(\mathbf{x}))] + \log\left(h_i^{N-k}\prod_{j=1}^{k}\Lambda_{ij}\right).$$

In the following, we show that $g_{1i}(\Phi(\mathbf{x}))$ and $g_{2i}(\Phi(\mathbf{x}))$ can be entirely written in a kernel form. To formulate the above equations, we need to calculate the eigenvalues Λ_i and eigenvectors Ω_i. However, due to the unknown dimensionality of the feature space, Σ_i cannot be computed directly. Moreover, because of the limited number of training samples, we can only express each eigenvector as the span of all the data points, as done in Schölkopf et al. (1998). The eigenvectors are in the space spanned by all the training samples, i.e., each eigenvector Ω_{ij} can be written as a linear combination of all the training samples:

$$\Omega_{ij} = \sum_{l=1}^{n}\gamma_{ij}^{(l)}\Phi(\mathbf{x}_l) = U\gamma_{ij}, \tag{11}$$

where $\gamma_{ij} = (\gamma_{ij}^{(1)}, \gamma_{ij}^{(2)}, \ldots, \gamma_{ij}^{(n)})^{\mathrm{T}}$ is an n dimensional column vector and $U = (\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n))$.

Theorem 1. *γ_{ij} and Λ_{ij} are the eigenvector and eigenvalue of the covariance matrix $\Sigma_{G^{(i)}}$, respectively.*

The proof of Theorem 1 can be found in the Appendix. Based on Theorem 1, we can express $g_{1i}(\Phi(\mathbf{x}))$ in the kernel form:

$$\begin{aligned} g_{1i}(\Phi(\mathbf{x})) &= \sum_{j=1}^{n}\gamma_{ij}^{\mathrm{T}}U^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)U\gamma_{ij} \\ &= \sum_{j=1}^{n}\left[\gamma_{ij}^{\mathrm{T}}\left(K_{\mathbf{x}} - \frac{1}{n_i}\sum_{l=1}^{n_i}K_{\mathbf{x}_l}\right)\right]^2 \\ &= \left\|K_{\mathbf{x}} - \frac{1}{n_i}\sum_{l=1}^{n_i}K_{\mathbf{x}_l}\right\|_2^2, \end{aligned}$$

where $K_{\mathbf{x}} = \{K(\mathbf{x}_1, \mathbf{x}), \ldots, K(\mathbf{x}_n, \mathbf{x})\}^{\mathrm{T}}$.

In the same way, $g_{2i}(\Phi(\mathbf{x}))$ can be formulated as the follows:

$$g_{2i}(\Phi(\mathbf{x})) = \sum_{j=1}^{k}\left(1-\frac{h_i}{\Lambda_{ij}}\right)\Omega_{ij}^{\mathrm{T}}(\Phi(\mathbf{x})-\mu_i)(\Phi(\mathbf{x})-\mu_i)^{\mathrm{T}}\Omega_{ij}.$$

Substituting (11) into the above $g_{2i}(\Phi(\mathbf{x}))$, we have:

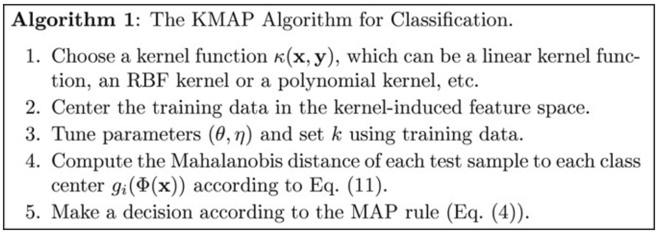
$$g_{2i}(\Phi(\mathbf{x})) = \sum_{j=1}^{k}\left(1-\frac{h_i}{\Lambda_{ij}}\right)\gamma_{ij}^{\mathrm{T}}\left(K_{\mathbf{x}} - \frac{1}{n_i}\sum_{j=1}^{n_i}K_{\mathbf{x}_j}\right) \times \left(K_{\mathbf{x}} - \frac{1}{n_i}\sum_{j=1}^{n_i}K_{\mathbf{x}_j}\right)^{\mathrm{T}}\gamma_{ij}.$$

Remark. In calculating $g_{2i}(\Phi(\mathbf{x}))$, only the k largest eigenvalues and relevant eigenvectors are selected for each class. In Williams and Seeger (2000) and Yang, Frangi, Yang, Zhang and Jin (2005), it is shown that the eigenvalue spectrum of the covariance matrix of the Gram matrix rapidly decays and thus is of low rank. This reinforces the theoretical basis of KMAP from another perspective.

Now, the discriminant function in the feature space $g_i(\Phi(\mathbf{x}))$ can be finally written in a kernel form, where N is substituted with the cardinality of data n.

We summarize the proposed KMAP algorithm in Fig. 2.

The overall time complexity of the algorithm is determined by Step 5 and Step 6. These steps involve computing the within-class



```
Algorithm 1: The KMAP Algorithm for Classification.
  1. Choose a kernel function $\kappa(\mathbf{x}, \mathbf{y})$, which can be a linear kernel function, an RBF kernel or a polynomial kernel, etc.
  2. Center the training data in the kernel-induced feature space.
  3. Tune parameters $(\theta, \eta)$ and set $k$ using training data.
  4. Compute the Mahalanobis distance of each test sample to each class center $g_i(\Phi(\mathbf{x}))$ according to Eq. (11).
  5. Make a decision according to the MAP rule (Eq. (4)).
```

Fig. 2. The KMAP algorithm for classification.

Table 1
The relationship among KMAP and other kernel methods.

Parameter setting		Kernel methods
θ	η	
0	0	A quadratic discriminant method
1	0	A linear discriminant method
1	1	The nearest mean classifier
0	1	The weighted nearest mean classifier

covariance matrix, and the complexity is $\mathcal{O}(n^2)$. In addition, it will cost $\mathcal{O}(n^3)$ operations to solve the eigenvalues and eigenvectors. Hence, KMAP has the same time complexity as KFDA. The storage complexity, which involves $\mathcal{O}(kn)$ for storing k columns of the covariance matrix, can be deduced because the value of k is much smaller than n. We will evaluate the scale of k in the experiments.

2.4. Connection with other kernel methods

The KMAP model is a generalized classification model and can be reduced to other kernel-based classification methods with different implementations of parameter estimation.

In the regularization method based on Eq. (8), by varying the settings of θ and η, other kernel-based classification methods can be derived. When ($\theta = 0, \eta = 0$), the KMAP model represents a quadratic discriminant method in the kernel-induced feature space; when ($\theta = 1, \eta = 0$), it represents a kernel discriminant method; and when ($\theta = 0, \eta = 1$) or ($\theta = 1, \eta = 1$), it represents the nearest mean classifier. Therefore, by varying θ and η, different models can be generated from different combinations of quadratic discriminant, linear discriminant and the nearest mean methods. The relationship among these kernel methods is summarized in Table 1.

We show in the following, that a special case of the regularization method when $\theta = 1$ and $\eta = 0$ will reduce to the well-known Kernel Fisher Discriminant Analysis (KFDA). If both classes are assumed to have the same covariance structure for a binary class problem (i.e., $\Sigma_i = \frac{\Sigma_1+\Sigma_2}{2}$) it leads to a linear discriminant function. Assuming all classes have the same class prior probabilities, $g_i(\Phi(\mathbf{x}))$ can be derived as:

$$g_i(\Phi(\mathbf{x})) = (\Phi(\mathbf{x}) - \mu_i)^{\mathrm{T}} \Sigma_i^{-1} (\Phi(x) - \mu_i)$$
$$= (\Phi(\mathbf{x}) - \mu_i)^{\mathrm{T}} \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\Phi(x) - \mu_i),$$

where $i = 1, 2$. We can reformulate this equation in the following form: $g_i(\Phi(\mathbf{x})) = \mathbf{w}_i\Phi(\mathbf{x}) + b_i$, where

$$\mathbf{w}_i = -4(\Sigma_1 + \Sigma_2)^{-1}\mu_i,$$
$$b_i = 2\mu_i^{\mathrm{T}}(\Sigma_1 + \Sigma_2)^{-1}\mu_i.$$

The decision hyperplane is $f(\Phi(\mathbf{x})) = g_1(\Phi(\mathbf{x})) - g_2(\Phi(\mathbf{x}))$, i.e.,

$$f(\Phi(\mathbf{x})) = (\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)\Phi(\mathbf{x})$$
$$- \frac{1}{2}(\mu_1 - \mu_2)^{\mathrm{T}}(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 + \mu_2).$$

Table 2
Overview of the experimental data sets used.

Data set	# samples	# features	# classes
Twonorm	1000	21	2
Breast	683	9	2
Ionosphere	351	34	2
Pima	768	8	2
Sonar	208	60	2
Iris	150	4	3
Wine	178	13	3
Segment	210	19	7

This equation is just the formulation of KFDA (Kim, Magnani, & Boyd, 2006; Mika et al., 1999). Therefore, KFDA can be viewed as a special case of KMAP when all classes have the same covariance structure.

Remark. KMAP thus provides a rich class of kernel-based classification algorithms using different regularization methods. This makes KMAP a flexible framework for classification adaptive to data distribution.

3. Experiments

In this section, we evaluate the proposed KMAP method on eight UCI data sets and three facial image data sets. As the classical methods and the state-of-the-art method in the face recognition task differ from tradition classification problems, we employ different comparison algorithms.

3.1. Experimental data sets

We describe these two batches of data sets for further evaluation in the following. The first batch comprises eight UCI data sets and the second comprises three facial image data sets.

3.1.1. UCI data sets

Eight data sets from the UCI machine learning repository, with different numbers of samples, features and classes, are chosen to test the performance of a variety of methods. Table 2 summarizes the information of these data sets.

3.1.2. Facial image data sets

To make comprehensive evaluations, we have collected three different kinds of data sets for our experiments. One is the Facial Recognition Technology (FERET) Database (Phillips, Moon, Rizvi, & Rauss, 2000). The second is the Face Recognition Grand Challenge (FRGC) data set (Phillips et al., 2005). The above two data sets are the de-facto standard data sets for face recognition evaluation. The third data set is the Yahoo! News facial images data set, which was obtained by crawling from the Web (Berg et al., 2004). These facial data sets are widely used for the performance evaluation of face recognition (Zhu, Hoi, & Lyu, 2008). In the following, we first describe the details of these data sets. Then we discuss our preprocessing methods for face extraction and feature representation.

FERET Face Data Set. In our experiment, 239 persons in the FERET data set are selected, and there are four gray scale 256 × 384 images for each individual. Among the four images, two images are from the FA/FB set, respectively, and the remaining two images are from DupI set. Therefore, there are a total of 956 images for evaluation. Since the images are acquired from different photo sessions, both the illumination conditions and the facial expressions may vary. All images are cropped and normalized by aligning the centers of the eyes to predefined positions, according to the manually located eye positions supplied by the FERET data. Fig. 3 depicts six individuals from this data set. The top two rows show the example images, the

Fig. 3. Example images from the FERET data set, cropped and normalized to the size of 128 × 128.

Fig. 4. Example images from the FRGC data set, cropped and normalized to the size of 128 × 128.

Fig. 5. Example images from the Yahoo! News Face data set, cropped and normalized to the size of 128 × 128.

Table 3
Overview on the face image data sets used in the experiments.

Data set	# total	# person	# per person
FERET	956	239	4
FRGC	1920	80	24
Yahoo! News	1940	97	20

first row from FA, and the second one from FB; while the bottom two rows are the examples from DupI.

FRGC Data Set. The FRGC data set (Phillips et al., 2005)[1] is the current benchmark for performance evaluation of face recognition techniques. We adopt the FRGC version-1 data set (Spring 2003) for the evaluation of our face recognition method. The data set used in our experiment consists of 1920 images, corresponding to 80 individuals selected from the original collection. Each individual has 24 controlled or uncontrolled color images. The faces are automatically detected and normalized through a face detection method and an extraction method. Fig. 4 shows geometrically normalized face images cropped from the original FRGC images, with the cropped regions resized to a size of 128 × 128.

Yahoo! News Face Data Set. The Yahoo! News Face data set was constructed by Berg et al. (2004) from about half a million captioned news images collected from the Yahoo! News Web site. It consists of a large number of photographs taken in real life conditions, rather than in the controlled environments widely used in face recognition evaluation. As a result, there are a large variety of poses, illuminations, expressions, and environmental conditions. There are 1940 images, corresponding to 97 largest face clusters selected to form our experimental data set, in which each individual cluster has 20 images. As with the other data sets, faces are cropped from the selected images using the face detection and extraction methods. Only relevant face images are retained when there are multiple faces in one image. Fig. 5 presents examples selected from the Yahoo! News images and the extracted faces. All these face images are geometrically normalized.

Facial Feature Extraction. To enable an automatic face recognition scheme, we cascade a face detector (Viola & Jones, 2004) with the Active Appearance Models (AAMs) (Cootes, Edwards, & Taylor, 2001) to locate faces and facial features in the input images. The performance in terms of the correct registration is greatly dependent on the image conditions. In fact, only about 30 images failed for the FRGC data set (5660 images). Similarly, the correct registration rate for the Yahoo! News face data set was around 80%. Many effective feature extraction methods have been proposed to address the task, such as Local Binary Pattern (Ahonen, Hadid, & Pietikainen, 2004; Rodriguez & Marcel, 2006) and Gabor wavelets transform. Among these methods, the Gabor wavelets representation of facial image has been widely accepted as a promising approach (Liu & Wechsler, 2002). From earlier studies in the area of signal processing, Lades et al. (1993) empirically suggested that good performance can be achieved by extracting Gabor wavelet features of 5 different scales and 8 orientations. In our experiments, we employ a similar approach by applying Gabor wavelet transform on each image (scaled to 128 × 128) at 5 scales and 8 orientations. Finally, we normalize each sub-image to form a feature vector $\mathbf{x} \in \mathbf{R}^n$ with the sample scale reduced to 64, which results in a 10240-dimensional feature vector for each facial image.

In summary, the detailed statistics of the data sets used in our experiments is listed in Table 3.

3.2. Experiments on UCI data sets

In this section, we conduct experiments on eight benchmark data sets. We first implement many other competitive methods and compare them with our proposed algorithm. Then we discuss and analyze the experimental results.

3.2.1. Comparison algorithms

We provide a brief introduction to the comparison algorithms in this section. Specifically, we compare our proposed model with the Modified Quadratic Discriminant Function (Kimura, Takashina, T. S., & M. Y., 1987), KFDA, the Kernel Fisher Quadratic Discriminant Analysis (KFQDA) (Huang, Hwang, & Lin, 2005), and SVM. Due to the popularity of SVM, we only focus on introducing MQDF, KFDA, and KFQDA in the following.

In statistical pattern recognition, the probability density function can first be estimated from the data. Then future examples could be assigned to the class with the MAP. One typical probability estimation method is to assume a multivariate normal density function over the data. From the multivariate normal distribution, the Quadratic Discriminant Function (Duda et al., 2000; Fukunaga, 1990) can be derived, which achieves the minimum mean error rate under Gaussianity and is also monotonic with an increase of the feature size (Waller & Jain, 1978). In Kimura et al. (1987), a Modified Quadratic Discriminant Function (MQDF) less sensitive to the estimation error, is proposed. Friedman (1989) improves the performance of QDF by the covariance matrix interpolation.

[1] Accessible from http://www.frvt.org/FRGC.