Feature Scaling for Kernel Fisher Discriminant Analysis Using Leave-One-Out Cross Validation

Liefeng Bo
blf0218@163.com
Ling Wang
wliiip@163.com
Licheng Jiao
lchjiao@mail.xidian.edu.cn
Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

Kernel fisher discriminant analysis (KFD) is a successful approach to classification. It is well known that the key challenge in KFD lies in the selection of free parameters such as kernel parameters and regularization parameters. Here we focus on the feature-scaling kernel where each feature individually associates with a scaling factor. A novel algorithm, named FS-KFD, is developed to tune the scaling factors and regularization parameters for the feature-scaling kernel. The proposed algorithm is based on optimizing the smooth leave-one-out error via a gradient-descent method and has been demonstrated to be computationally feasible. FS-KFD is motivated by the following two fundamental facts: the leave-one-out error of KFD can be expressed in closed form and the step function can be approximated by a sigmoid function. Empirical comparisons on artificial and benchmark data sets suggest that FS-KFD improves KFD in terms of classification accuracy.

1 Introduction

Fisher linear discriminant analysis (Fisher, 1936; Fukunaga,1990) is a classical classifier whose fundamental idea is to maximize the between-class scatter while minimizing the within-class scatter simultaneously. In many applications, Fisher linear discriminant analysis has proved to be very powerful. However, for real-world problems, only linear discriminant analysis is not good enough. Mika, Ratsch, and Weston (1999) and Mika (2002) introduced a class of nonlinear Fisher discriminant analysis using kernel tricks, named KFD. Extensive empirical comparisons have shown that KFD is comparable to other kernel-based classifiers, such as support vector machines (SVMs) (Vapnik, 1995, 1998) and least-squares support vector machines (LS-SVMs) (Gestel et al., 2002; Suykens & Vandewalle, 1999).

For kernel-based learning algorithms, the key challenge lies in the selection of kernel parameters and regularization parameters. Many researchers have identified this problem and tried to solve it. Weston et al. (2001) performed feature selection for SVMs by combining the feature scaling technique with the leave-one-out error bound. Chapelle, Vapnik, Bousquet, and Mukherjee (2002) tuned multiple parameters for two-norm SVMs by minimizing the radius margin bound or the span bound. Ong and Smola (2003) applied semidefinite programming to learn kernel function by hyperkernel. Lanckriet, Cristianni, Bartlett, Ghaoui, and Jordan (2004) designed kernel matrix directly by semidefinite programming. All of these algorithms have proved to be effective and boosted the development of this field.

We focus here on tuning the scaling factors of the feature scaling kernel (Williams & Barber, 1998; Krishnapuram, Hartemink, Carin, & Figueiredo, 2004). Two of the most popular feature-scaling kernels are polynomial kernel and gaussian kernel, as given below:

$$\mathrm{K}_{\theta}(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \sum_{k=1}^{d} \theta_k \mathbf{x}_i^{(k)} \mathbf{x}_j^{(k)}\right)^r, \tag{1.1}$$

$$\mathrm{K}_{\theta}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{k=1}^{d} \theta_k \left\| \mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)} \right\|^2\right). \tag{1.2}$$

In a feature-scaling kernel, each feature has its own scaling factor. If some feature is insignificant or irrelevant for classification, the associated scaling factor will be set smaller; otherwise, it will be set larger.

Cawley and Talbot (2003) gave a closed form of the leave-one-out error of KFD and demonstrated that it was superior to n-fold cross-validation error in terms of computational complexity. Motivated by this fact, we develop a novel algorithm, named FS-KFD, to tune multiple parameters for the feature-scaling kernel. FS-KFD is constructed in two steps: replacing the step function in the leave-one-out error with a sigmoid function and then optimizing the resulting smooth leave-one-out error via a gradient-descent algorithm. In FS-KFD, all the free parameters are analytically chosen, so the learning process is fully automatic. Extensive experimental comparisons show that FS-KFD improves the performance of KFD in the presence of many irrelevant features and obtains good classification accuracy.

The remainder of the letter is organized as follows. In section 2, kernel Fisher discriminant analysis is briefly reviewed. The expressions for the smooth leave-one-out error and for its derivative are given in section 3. FS-KFD is extended to multiclass classification in section 4. In section 5, the experimental results are reported. The direction of future research is indicated in section 6.

2 Kernel Fisher Discriminant Analysis

For real-world problems, linear discriminant analysis is not enough. Mika et al. constructed the linear discriminant analysis in the feature space induced by a Mercer kernel, thus implicitly yielding a nonlinear discriminant analysis in the input space. The leading model is named KFD, in which two scatter matrices—between-class scatter matrix and within-class scatter matrix—are defined by $S_b^F = (\mathbf{m}_1^F - \mathbf{m}_2^F)(\mathbf{m}_1^F - \mathbf{m}_2^F)^T$ and $S_w^F = \sum_{i=1}^{2}\sum_{j=1}^{l_i}(\Phi(\mathbf{x}_j^i) - \mathbf{m}_i^F)(\Phi(\mathbf{x}_j^i) - \mathbf{m}_i^F)^T$, where the mean $\mathbf{m}_i^F$ of the ith class is $\mathbf{m}_i^F = \frac{1}{l_i}\sum_{j=1}^{l_i}\Phi(\mathbf{x}_j^i)$. An optimal transformation $\mathbf{w}$ is given by maximizing the between-class scatter while simultaneously minimizing the within-class scatter:

$$\max_{\mathbf{w}}\left(\frac{\mathbf{w}^T S_b^F \mathbf{w}}{\mathbf{w}^T S_w^F \mathbf{w}}\right). \tag{2.1}$$

In terms of reproducing kernel theory (Aronszajn, 1950), $\mathbf{w}$ can be formulated as $\mathbf{w} = \sum_{j=1}^{l}\alpha_j\Phi(\mathbf{x}_j)$. With equation 2.1, we can calculate $\boldsymbol{\alpha}$ by

$$\max_{\boldsymbol{\alpha}}\left(\frac{\boldsymbol{\alpha}^T \bar{S}_b^F \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \bar{S}_w^F \boldsymbol{\alpha}}\right), \tag{2.2}$$

where $\bar{S}_b^F = (\bar{\mathbf{m}}_1^F - \bar{\mathbf{m}}_2^F)(\bar{\mathbf{m}}_1^F - \bar{\mathbf{m}}_2^F)^T$ and $\bar{S}_w^F = \sum_{i=1}^{2}\sum_{j=1}^{l_j}(\boldsymbol{\beta}_j^i - \bar{\mathbf{m}}_i^F)(\boldsymbol{\beta}_j^i - \bar{\mathbf{m}}_i^F)^T$ with $\boldsymbol{\beta}_j^i = [\mathrm{K}(\mathbf{x}_1, \mathbf{x}_j^i), \ldots, \mathrm{K}(\mathbf{x}_l, \mathbf{x}_j^i)]^T$ and $\bar{\mathbf{m}}_i^F = \frac{1}{l_i}[\sum_{j=1}^{l_i}\mathrm{K}(\mathbf{x}_1, \mathbf{x}_j^i), \ldots, \sum_{j=1}^{l_i}\mathrm{K}(\mathbf{x}_l, \mathbf{x}_j^i)]^T$.

It can be seen that KFD is equivalent to finding the leading eigenvector of $(\bar{S}_w^F)^{-1}\bar{S}_b^F$. To improve numerical stability and generalization ability, we replace $\bar{S}_w^F$ with $\bar{S}_w^F + \lambda\mathbf{I}$, where λ is a regularization constant and $\mathbf{I}$ is an identity matrix. For a new sample $\mathbf{x}$, we can predict its label by

$$g(\mathbf{x}) = \mathrm{sgn}((\mathbf{w}\boldsymbol{\cdot}\Phi(\mathbf{x})) + b) = \mathrm{sgn}\left(\sum_{j=1}^{l}\alpha_j\mathrm{K}(\mathbf{x}_j, \mathbf{x}) + b\right), \tag{2.3}$$

where $b = -\alpha^T\frac{l_1\bar{\mathbf{m}}_1^F + l_2\bar{\mathbf{m}}_2^F}{l}$.

3 Optimization of the Smooth Leave-One-Out Cross-Validation Error

Let us denote the leave-one-out error by $\Gamma(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_l, y_l)$. It is well known that the leave-one-out error is almost an unbiased estimate of the expected generalization error.

Lemma 1 (Luntz & Brailovsky, 1969; Schölkopf & Smola, 2002).

$$E\left(p_{error}^{l-1}\right) = E\left(\frac{1}{l}\Gamma(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_l, y_l)\right),$$

where p_{error}^{l-1} is the probability of test error for the model trained on samples of size $l-1$ and the expectations are taken over the random choice of the samples.

This lemma suggests that the leave-one-out error is a good estimate for the generalization error. However, the leave-one-out cross validation is rarely adopted in a medium- or large-scale application due to its high computational cost; it requires running the training algorithm l times. The training algorithms for kernel machines, such as KFD, typically require a computational cost of $O(l^3)$. In this case, the computational cost of the leave-one-out cross-validation procedure is $O(l^4)$, which quickly becomes intractable as the number of training samples increases. Fortunately, there exists a computationally efficient implementation for the leave-one-out cross-validation procedure in KFD, which only a computational cost of incurs $O(l^3)$.

Xu, Zhang, and Li (2001) showed that KFD is equivalent to minimizing the following loss function,

$$f(\bar{\boldsymbol{\alpha}}) = \bar{\boldsymbol{\alpha}}^T(\mathbf{C}^T\mathbf{C} + \lambda\mathbf{U})\bar{\boldsymbol{\alpha}} - 2\bar{\boldsymbol{\alpha}}^T\mathbf{C}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}, \tag{3.1}$$

where $\bar{\boldsymbol{\alpha}} = [\begin{smallmatrix}\alpha \\ b\end{smallmatrix}]$, $\mathbf{C} = [\mathbf{K} \quad 1]$, $\mathbf{U} = [\begin{smallmatrix}\mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 0\end{smallmatrix}]$, and $\mathbf{I}$ denotes the unit matrix. Let $g_i(\mathbf{x})$ be the ith kernel Fisher classifier constructed from the data set excluding the ith training sample. Defining the residual error by $r_i = y_i - g_i(\mathbf{x}_i)$ for the ith training sample, Cawley and Talbot (2003) demonstrated the following:

Lemma 2. $\boldsymbol{r} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y} \odot (1 - D(\boldsymbol{H}))$, *where* $\boldsymbol{H} = \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C} + \lambda\boldsymbol{U})^{-1}\boldsymbol{C}^T$, *$D(\boldsymbol{H})$ denotes the diagonal elements of $\boldsymbol{H}$, and $\odot$ denotes element-wise division.*

A straightforward corollary of lemma 2 is that the leave-one-out error of KFD can be computed at a cost of $O(l^3)$. This indicates that it is feasible to apply leave-one-out model selection to a medium-size problem. In the following, we will discuss the smooth leave-one-out error derived by replacing the step function with a sigmoid function. According to lemma 2, the leave-one-out error of KFD is given by

$$loo(\boldsymbol{\theta}, \lambda) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1 - y_i \operatorname{sign}(y_i - r_i)}{2}\right), \tag{3.2}$$

where sign (a) is 1 if $a \geq 0$; otherwise, sign(a) is -1. From equation 3.2, we observe that there exists a step function sign $(\cdot)$ in $loo(\boldsymbol{\theta}, \lambda)$, implying that it is not differentiable. In order to use a gradient-descent method to minimize this estimate, we approximate the step function by a sigmoid function,

$$\tanh(\gamma t) = \frac{\exp(\gamma t) - \exp(-\gamma t)}{\exp(\gamma t) + \exp(-\gamma t)}, \tag{3.3}$$

where we set γ to be 10. Then the smooth leave-one-out error can be expressed as

$$loo(\boldsymbol{\theta}, \lambda) = \frac{1}{l} \sum_{i=1}^{l} \left(\frac{1 - y_i \tanh(\gamma(y_i - r_i))}{2} \right). \tag{3.4}$$

Figure 1 shows the leave-one-out error and the smooth leave-one-out error on the Breast Cancer data set. It can be seen from Figure 1 that the smooth leave-one-out error successfully follows the trend of the leave-one-out error. Thus, we can expect that a small, smooth leave-one-out error guarantees good generalization ability.

According to the chain rule, the derivative of $loo(\boldsymbol{\theta}, \lambda)$ is formulated as

$$\frac{\partial(loo(\boldsymbol{\theta}, \lambda))}{\partial \theta_k} = \frac{\partial(loo(\boldsymbol{\theta}, \lambda))}{\partial \mathbf{r}^T} \frac{\partial \mathbf{r}}{\partial \theta_k}. \tag{3.5}$$

It follows that we need only to calculate $\frac{\partial(loo(\boldsymbol{\theta},\lambda))}{\partial \mathbf{r}^T}$ and $\frac{\partial \mathbf{r}}{\partial \theta_k}$, respectively. With $\frac{\partial(\tanh(t))}{\partial t} = \sec\text{h}^2(t)$, we have

$$\frac{\partial\,(loo\,(\boldsymbol{\theta}, \lambda))}{\partial \mathbf{r}^T} = \left(\frac{\gamma \mathbf{y} \otimes \sec\text{h}^2\,(\gamma\,(\mathbf{y} - \mathbf{r}))}{2l} \right)^T, \tag{3.6}$$

where $\otimes$ denotes an element-wise proxduct. The derivative of $\mathbf{r}$ with respect to θ_k is given by

$$\begin{aligned} \frac{\partial \mathbf{r}}{\partial \theta_k} = &-\left(\frac{\partial \mathbf{H}}{\partial \theta_k} \mathbf{y} \right) \odot (\mathbf{1} - \mathrm{D}\,(\mathbf{H})) \\ &+ ((\mathbf{I} - \mathbf{H})\,\mathbf{y}) \odot (\mathbf{1} - \mathrm{D}\,(\mathbf{H})) \odot (\mathbf{1} - \mathrm{D}\,(\mathbf{H})) \otimes \mathrm{D}\left(\frac{\partial \mathbf{H}}{\partial \theta_k} \right). \end{aligned} \tag{3.7}$$

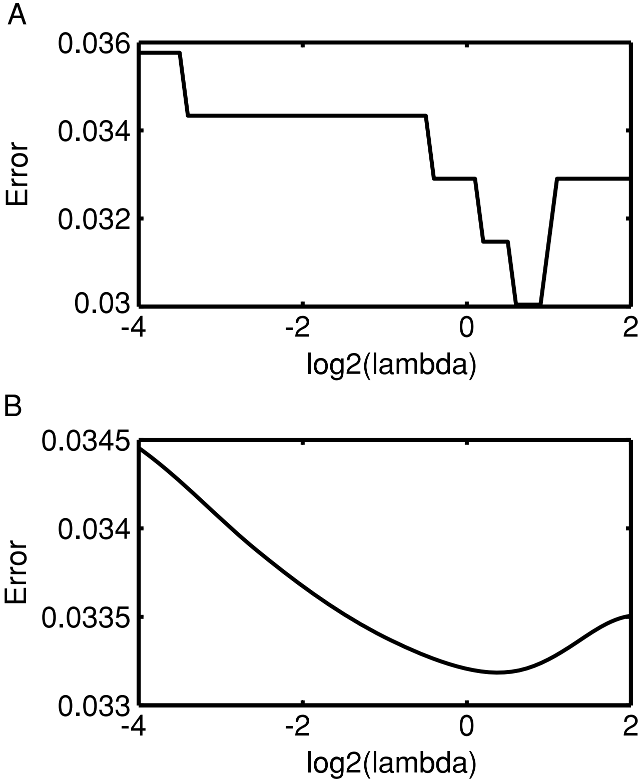

Figure 1: (A) Variation of the leave-one-out error with $\log 2(\lambda)$ on the Breast data set. (B) Variation of the smooth leave-one-out error with $\log 2(\lambda)$ on the Breast data set.

The derivative of $\mathbf{H}$ with respect to θ_k is given by

$$\frac{\partial \mathbf{H}}{\partial \theta_k} = \frac{\partial \mathbf{C}}{\partial \theta_k}\left(\mathbf{C}^T\mathbf{C} + \lambda\mathbf{U}\right)^{-1}\mathbf{C}^T + \mathbf{C}\frac{\partial\left(\mathbf{C}^T\mathbf{C} + \lambda\mathbf{U}\right)^{-1}}{\partial \theta_k}\mathbf{C}^T + \mathbf{C}\left(\mathbf{C}^T\mathbf{C} + \lambda\mathbf{U}\right)^{-1}\frac{\partial \mathbf{C}^T}{\partial \theta_k}. \tag{3.8}$$

Now let us focus on computing $\frac{\partial(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U})^{-1}}{\partial\theta_k}$. A good solution is based on the equality: $\mathbf{T}^{-1}\mathbf{T} = \mathbf{I}$ (Bengio, 2000). Differentiating both sides of the equation with respect to θ_k and then isolating $\frac{\partial \mathbf{T}^{-1}}{\partial \theta_k}$, we have

$$\frac{\partial \mathbf{T}^{-1}}{\partial \theta_k} = -\mathbf{T}^{-1}\frac{\partial \mathbf{T}}{\partial \theta_k}\mathbf{T}^{-1}. \tag{3.9}$$

Substituting $\mathbf{C}^T\mathbf{C} + \lambda\mathbf{U}$ for $\mathbf{T}$, we have

$$\begin{aligned}\frac{\partial\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}}{\partial\theta_k} &= -\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}\frac{\partial\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)}{\partial\theta_k}\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}\\ &= -\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}\left(\frac{\partial\mathbf{C}^T}{\partial\theta_k}\mathbf{C}+\mathbf{C}^T\frac{\partial\mathbf{C}}{\partial\theta_k}\right)\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}\end{aligned} \tag{3.10}$$

Combining equations 3.5, 3.6, 3.7, 3.8, and 3.10, we can compute the derivative of the smooth leave-one-out error with respect to θ_k.

The derivative of $\mathbf{H}$ with respect to λ is given by

$$\frac{\partial\mathbf{H}}{\partial\lambda} = -\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}\left(\mathbf{C}^T\mathbf{C}+\lambda\mathbf{U}\right)^{-1}. \tag{3.11}$$

So we can compute the derivative of $loo(\boldsymbol{\theta}, \lambda)$ with respect to λ in a similar manner. From the derivation, it can be easily verified that the computational complexity of FS-KFD is

$$\#\,(\text{Iteration}) \times \#\,(\text{free parameters}) \times l^3. \tag{3.12}$$

4 Extension to Multiclass Classification

In this section, we attempt to extend FS-KFD to multiclass classification using the one-against-all scheme that has been independently devised by several researchers. Rifkin and Klautau (2004) carefully compared the one-against-all scheme with some other popular multiclass schemes and concluded that it is as accurate as any other scheme if the underlying binary classifiers are well-tuned, regularized classifiers.

One-against-all reduces a c-class problem to c binary problems. For the sth binary problem, all samples labeled $y_i = s$ are considered positive samples and the others negative samples. For a new sample prediction, c classifiers are run, and the classifier that outputs the largest value is chosen.

Let $g^{(s)}(\mathbf{x}_i)$ denote the output of the sth binary classifier on a sample $\mathbf{x}_i$. According to the one-against-all scheme, the predicted label for $\mathbf{x}_i$ is

$$\hat{y}_i = \arg\max_{s\in\{1,\ldots,c\}}\left(g^{(s)}\left(\mathbf{x}_i\right)\right). \tag{4.1}$$

Thus, the leave-one-out error of multiclass classification can be written as

$$mloo(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{l}\sum_{i=1}^{l}\left(1 - \text{equal}\left(y_i, \arg\max_s\left(g^{(s)}(\mathbf{x}_i)\right)\right)\right), \tag{4.2}$$

where equal $(a, b) = 1$ if $a = b$; otherwise equal $(a, b) = 0$, and $y_i \in \{1, 2, \ldots, c\}$. It becomes intractable to approximate equation 4.2 by a sigmoid function due to the discontinuity of the inner function $\arg\max_s(g^{(s)}(\mathbf{x}_i))$. In the following, we consider an alternative strategy where the upper bound of the leave-one-out error of multiclass classification is optimized.

Theorem 1. *Let $loo^{(s)}$ denote the leave-one-out error of the sth binary classifier. If the one-against-all scheme is used, the following inequality holds:*

$$mloo \leq \sum_{s=1}^{c} loo^{(s)}. \tag{4.3}$$

Proof. If all c binary classifiers classify the sample $\mathbf{x}_i$ correctly, we have

$$y_i^{(s)} g^{(s)}(\mathbf{x}_i) > 0, \quad s = 1, \ldots, c, \tag{4.4}$$

where $y_i^{(s)} = 1$, if $y_i = s$; otherwise $y_i^{(s)} = -1$. Inequality 4.4 can be further simplified to

$$\begin{cases} g^{(y_i)}(\mathbf{x}_i) > 0 \\ g^{(s)}(\mathbf{x}_i) < 0, \quad s \neq y_i \end{cases}. \tag{4.5}$$

Since only the output of the y_ith classifier is greater than zero, we have

$$\arg\min_s \left(g^{(s)}(\mathbf{x}_i)\right) = y_i. \tag{4.6}$$

This means that if all c binary classifiers classify the sample $\mathbf{x}_i$ correctly, the final multiclass classifier also classifies $\mathbf{x}_i$ correctly. The equivalent proposition is that if the multiclass classifier classifies $\mathbf{x}_i$ incorrectly, there exists at least one binary classifier misclassifying $\mathbf{x}_i$. This completes the proof of theorem 1.

This theorem allows us to control the leave-one-out error of multiclass classification by controlling the sum of the leave-one-out error of all the binary classifiers. Three multiclass schemes can be derived by considering whether the kernel parameters and the regularization parameters are shared by all the binary classifiers.

In the first scheme, all the binary classifiers share the kernel parameters and the regularization parameters (Hsu & Lin, 2002; Rifkin & Klautau, 2004). The sum of the smooth leave-one-out errors of c binary classifiers can be formulated as

$$sloo\,(\boldsymbol{\theta}, \lambda) = \sum_{s=1}^{c} \left(loo^{(s)}\,(\boldsymbol{\theta}, \lambda) \right). \tag{4.7}$$

$loo^{(s)}(\boldsymbol{\theta}, \lambda)$ can be expanded into

$$loo^{(s)}\,(\boldsymbol{\theta}, \lambda) = \frac{1}{l} \sum_{i=1}^{l} \left(\frac{1 - y_i^{(s)} \tanh\left(\gamma \left(y_i^{(s)} - r_i^{(s)}\right)\right)}{2} \right), \tag{4.8}$$

where $r_i^{(s)}$ is the residual error on the ith sample for the sth binary problem. The derivative of $sloo(\boldsymbol{\theta}, \lambda)$ with respect to θ_k is given by

$$\frac{\partial (sloo(\boldsymbol{\theta}, \lambda))}{\partial \theta_k} = \sum_{s=1}^{c} \frac{\partial (loo^{(s)}(\boldsymbol{\theta}, \lambda))}{\partial (\mathbf{r}^{(s)})^T} \frac{\partial \mathbf{r}^{(s)}}{\partial \theta_k}, \tag{4.9}$$

where

$$\frac{\partial (loo^{(s)}(\boldsymbol{\theta}, \lambda))}{\partial \left(\mathbf{r}^{(s)}\right)^T} = \left(\frac{\gamma \mathbf{y}^{(s)} \otimes \mathrm{sec\,h}^2(\gamma(\mathbf{y}^{(s)} - \mathbf{r}^{(s)}))}{2l} \right)^T, \tag{4.10}$$

$$\begin{aligned} \frac{\partial \mathbf{r}^{(s)}}{\partial \theta_k} = & - \left(\frac{\partial \mathbf{H}}{\partial \theta_k} \mathbf{y}^{(s)} \right) \odot (\mathbf{1} - \mathrm{D}(\mathbf{H})) \\ & + ((\mathbf{I} - \mathbf{H})\mathbf{y}^{(s)}) \odot (\mathbf{1} - \mathrm{D}(\mathbf{H})) \odot (\mathbf{1} - \mathrm{D}\,(\mathbf{H})) \otimes \mathrm{D} \left(\frac{\partial \mathbf{H}}{\partial \theta_k} \right) \end{aligned} \tag{4.11}$$

Thus, we can compute the derivative of $sloo\,(\boldsymbol{\theta}, \lambda)$ with respect to θ_k by combining equations 4.9, 4.10, and 4.11. The derivative of $sloo\,(\boldsymbol{\theta}, \lambda)$ with respect to λ can be computed in a similar manner. It is easily checked that the computational complexity of this multiclass scheme is the same as that of FS-KFD for binary classification since all the binary classifiers share $\mathbf{H}$.

In the second scheme, only the kernel parameters are shared. As a result, the binary classifiers no longer share $\mathbf{H}$ due to the difference among the regularization parameters. The computational complexity of this scheme becomes

$$c \times \#\,(\text{Iteration}) \times \#\,(\text{free parameters}) \times l^3. \tag{4.12}$$

In the third scheme, the kernel parameters and the regularization parameters are not shared. Therefore, we independently optimize the free parameters of each binary classifier. The computational complexity of this scheme is the same as that of the second one.

5 Performance Comparison

In order to demonstrate the effectiveness of FS-KFD, we compare its performance with those of KFD, SVMs, and k-nearest neighbors (KNN) (Lowe, 1995) on an artificial XOR problem, benchmark data sets from UCI Machine Learning Repository (Blake & Merz, 1998), and the radar target recognition problem. All the algorithms were implemented in MATLAB 7.0. And all the experiments were run on a personal computer with 2.4 GHz P4 processors, 2 GB memory, and Windows XP operation system. Unless otherwise specified, the FS-KFD mentioned in the following uses the gaussian kernel.

For FS-KFD, a gradient-descent method is used to search for the optimal values for free parameters, and thus one needs to choose good optimization software. We recommend using an available optimization package to avoid the numerical problems. Here we use the function *fminunc* in the optimization toolbox of MATLAB that implements BFGS quasi-Newton algorithm to solve medium-scale problems. The maximum number of iterations allowed is set to be 50, the termination tolerance on the function value and variable value is set to be 0.0001, and the cubic polynomial line search procedure is used to find the optimal step size. To avoid adding positive constraints in the optimization problem, we use parameterizations $\beta = (\log(\boldsymbol{\theta}), \log(\lambda))$. The initial values of the scaling factors and regularization parameters are $\log(\boldsymbol{\theta}) = \log(\frac{1}{d}) \times \mathbf{1}$ and $\log(\lambda) = 0$, respectively, where d is the feature dimensionality.

In general, choosing the optimal value for γ is difficult. Throughout the article, γ is set to be 10. We have found that using the same setting for various data sets works well. We can also try several different values for γ and choose the one leading to the smallest leave-one-out error.

5.1 Artificial XOR Problem. This experiment aims at validating the robustness of FS-KFD against the inclusion of the irrelevant features. To this end, a variant of XOR is constructed, with each feature drawn from a uniform distribution on the interval $[-1, 1]$. Regardless of the feature dimensionality d, the output label for a given data point is related to only the first two features of the data and is defined as

$$y = \begin{cases} +1 & \textit{if } x_1 x_2 \geq 0 \\ -1 & \textit{otherwise} \end{cases} \qquad x_1, x_2 \in U(-1, +1). \tag{5.1}$$

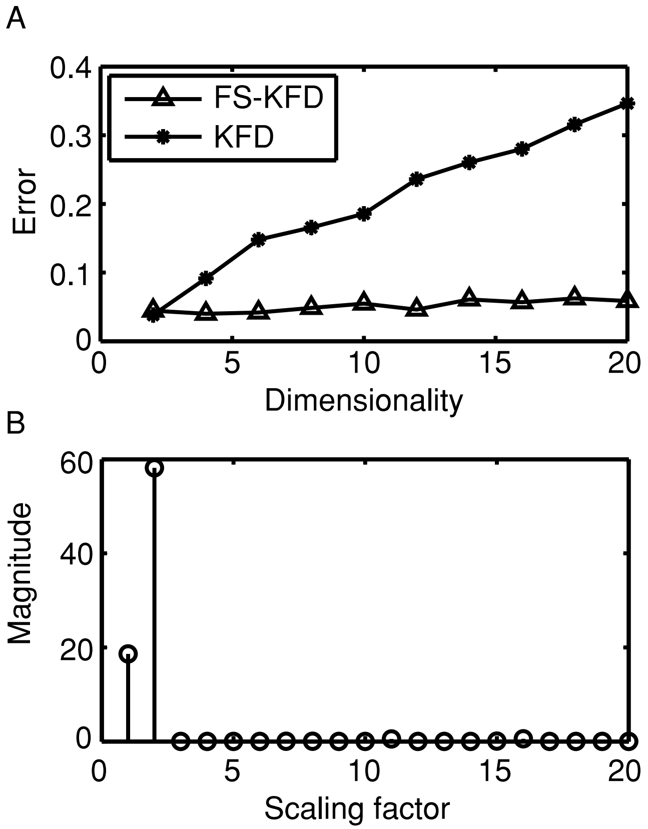

Figure 2: (A) Variation of the errors of FS-KFD and KFD with the dimensionality. (B) Scaling factors with the dimensionality $d = 20$.

This suggests that there exist $d - 2$ irrelevant features for the data with d features. The optimal decision function of this problem is nonlinear, and the highest recognition rate of linear classifiers is only 66.67%. FS-KFD and KFD are constructed on the training set with 200 samples and tested on the independent test set with 5000 samples. The results are averaged over 10 random realizations. To study the scaling property of the errors of FS-KFD and KFD as the feature dimensionality, we sequentially increase the feature dimensionality from 2 to 20 at an interval of 2. The plots of the errors of the two algorithms as the function of the feature dimensionality are shown in Figure 2A. The scaling factors with the dimensionality $d = 20$ are shown in Figure 2B.

From Figure 2A, we observe that FS-KFD is much more robust to the increase of the irrelevant features compared with KFD. Furthermore, the

Table 1: Information on Benchmark Data Sets.

Problem	Training/Test	Class	Attribute
Breast	400/299	2	9
German	600/400	2	20
Liver	200/145	2	6
Diabetes	400/368	2	8
Vote	250/185	2	16
Glass	150/64	6	9
Yeast	100/108	5	79
Splice	500/1675	3	240
Segment	500/1810	7	18
Vehicle	500/346	4	18

feature selection ability of FS-KFD is clearly exhibited in Figure 2B. The scaling factors corresponding to the relevant features are significantly larger than those corresponding to the irrelevant features. The rapid performance degradation of KFD suggests that the feature-scaling technique is indeed necessary in the presence of many irrelevant features.

5.2 Benchmark Comparison. The purpose of this experiment is to compare FS-KFD with KFD, SVM, and KNN on a collection of benchmark data sets from the UCI Machine Learning Repository. These data sets have been extensively used in testing the performance of diversified kinds of learning algorithms. Information on these benchmark data sets is summarized in Table 1.

The sizes of training set and test set are shown in the second column of Table 1. For each training-test pair, the training samples are scaled into zero mean and unit variance, and the test samples are adjusted using the same linear transformation. The final errors are averaged over 10 random splits of the full data sets, which are reported in Tables 2 and 3.

Note that all model selection procedures are independently performed for each training-test pair so that the standard error of the mean includes the variability due to the sensitivity of the model selection criterion to the partitioning of the data. The detailed experimental setups are summarized as follows:

1. For KFD, the leave-one-out error is used for model selection. We perform a grid search on intervals $\log 2(\theta) = [-12, -10, \ldots, 4]$ and $\log 2(\lambda) = [-10, -9, \ldots, 1]$. Three possible multiclass schemes are considered: KFD with shared kernel parameters and regularization parameters, KFD with only shared kernel parameters, and KFD without shared free parameters.

Table 2: Mean and Variance of Test errors Obtained by FS-KFD, KFD, Span Bound–Based SVM, and KNN.

Problem	FS-KFD(1)	KFD(1)	SVM(Span)	KNN
Breast	4.05 ± 0.71	4.11 ± 0.77	4.45 ± 0.76	3.85 ± 1.01
German	24.75 ± 1.88	23.35 ± 2.74	24.22 ± 2.19	27.35 ± 2.10
Diabetes	24.67 ± 1.75	23.45 ± 2.05	24.86 ± 1.59	26.68 ± 2.06
Liver	30.14 ± 5.34	29.72 ± 5.16	31.72 ± 5.26	39.66 ± 4.09
Vote	5.14 ± 1.40	5.62 ± 1.98	5.08 ± 1.86	7.08 ± 1.83
Glass	32.81 ± 9.63	33.28 ± 7.51	32.97 ± 6.93	31.87 ± 5.95
Splice	6.33 ± 1.27	6.90 ± 1.09	6.91 ± 0.61	10.32 ± 1.06
Yeast	5.83 ± 1.80	5.85 ± 2.02	6.67 ± 1.79	8.89 ± 2.23
Segment	4.59 ± 0.67	7.87 ± 0.80	6.57 ± 1.25	8.25 ± 1.03
Vehicle	17.72 ± 2.21	20.17 ± 2.00	17.05 ± 2.38	31.56 ± 1.83

Notes: FS-KFD(1) denotes FS-KFD with shared kernel parameters and regularization parameters. KFD(1) denotes KFD with shared kernel parameters and regularization parameters.

2. For SVM, the span bound (Vapnik & Chapelle, 2000) is used to optimize the kernel parameters and the regularization parameters. Initial setups are the same as in FS-KFD.
3. For KNN, the leave-one-out error is used to find the best number of neighbors k. We consider 50 different values from the interval $[1, \ldots, l-1]$ (uniformly in logarithm) (Rätsch, 2001), where l is the size of the training set.

Two-tailed t-tests with the significant level 0.05 are performed to determine whether there is a significant difference between FS-KFD and other algorithms. The conclusions are summarized as follows. FS-KFD is significantly better than KFD on the Segment and Vehicle data sets. As for the remaining data sets, FS-KFD and KFD achieve similar performance.

Table 3: Mean and Variance of Test errors Obtained by FS-KFD and KFD.

Problem	FS-KFD(2)	FS-KFD(3)	KFD(2)	KFD(3)
Glass	34.53 ± 11.13	31.87 ± 8.31	33.44 ± 9.44	31.71 ± 9.58
Splice	6.16 ± 1.19	5.87 ± 1.00	6.95 ± 0.93	6.71 ± 0.93
Yeast	6.29 ± 2.61	6.67 ± 2.38	6.48 ± 2.58	7.59 ± 3.02
Segment	4.36 ± 0.66	4.61 ± 0.75	8.04 ± 0.98	7.62 ± 1.00
Vehicle	17.89 ± 2.07	18.58 ± 2.09	20.64 ± 2.13	20.40 ± 1.93

Notes: FS-KFD(2) and FS-KFD(3) denote FS-KFD with only shared kernel parameters and without shared free parameters, respectively. KFD(2) and KFD(3) denote KFD with only shared kernel parameters and without shared free parameters, respectively.

FS-KFD and span bound–based SVM obtain similar performance on all data sets except Segment. FS-KFD is much better than KNN on all data sets except Breast and Glass.

Pairwise two-tailed t-tests with a significance level of 0.05 are performed to determine whether there is a significant difference among the three multiclass schemes of FS-KFD and KFD. The resulting p-values indicate that there is no significant difference among the three multiclass schemes.

In general, the feature-scaling technique improves the generalization performance of KFD and leads to a natural feature selection when irrelevant features occur. For example, on the Segment data set, the four largest scaling factors are 13.98, 3.50, 2.72, and 2.26, and yet other scaling factors are smaller than 0.5.

5.3 Radar Target Recognition.

Radar target recognition refers to the detection and recognition of target signatures using high-resolution range profiles—in our case, in inverse synthetic aperture radar. A radar image represents a spatial distribution of microwave reflectivity that is sufficient to characterize the illuminated target. Range resolution allows the sorting of reflected signals on the basis of range. When range-gating or time-delay sorting is used to interrogate the entire range extent of the target space, a one-dimensional image, called a range profile, will be generated. Figure 3 is an example of such signature of three different planes: J-6, J-7, and B-52.

Our task is to recognize the range profile of the three different plane models—J-6, J-7, and B-52—based on experimental data acquired in a microwave anechoic chamber. The dimensionality of the range profiles is 64. The full data set is split into 359 training samples and 719 test samples. The training samples consist of 103 one-dimensional images of J-6, 149 one-dimensional images of J-7, and 107 one-dimensional images of B-52. The test samples consist of 206 one-dimensional images of J-6, 299 one-dimensional images of J-7, and 214 one-dimensional images of B-52. Experimental results for several classifiers are summarized in Table 4. It can be observed that FS-KFD is superior to other classifiers in terms of classification accuracy on this data set.

6 Discussion

Our algorithm is not yet applicable to the problems where the number of feature dimensionality is on the order of several hundred and that of training samples on the order of several thousand due to the high computational cost. This limitation can be overcome by integrating a feature preselection step into FS-KFD. An alternative way to break this limitation is to allow some associated features to share the same scaling factors. For example, in image recognition problems, it is reasonable that the neighboring features share the same scaling factors. Exploiting effective feature preselection and reasonable feature-sharing schemes is an interesting research direction.

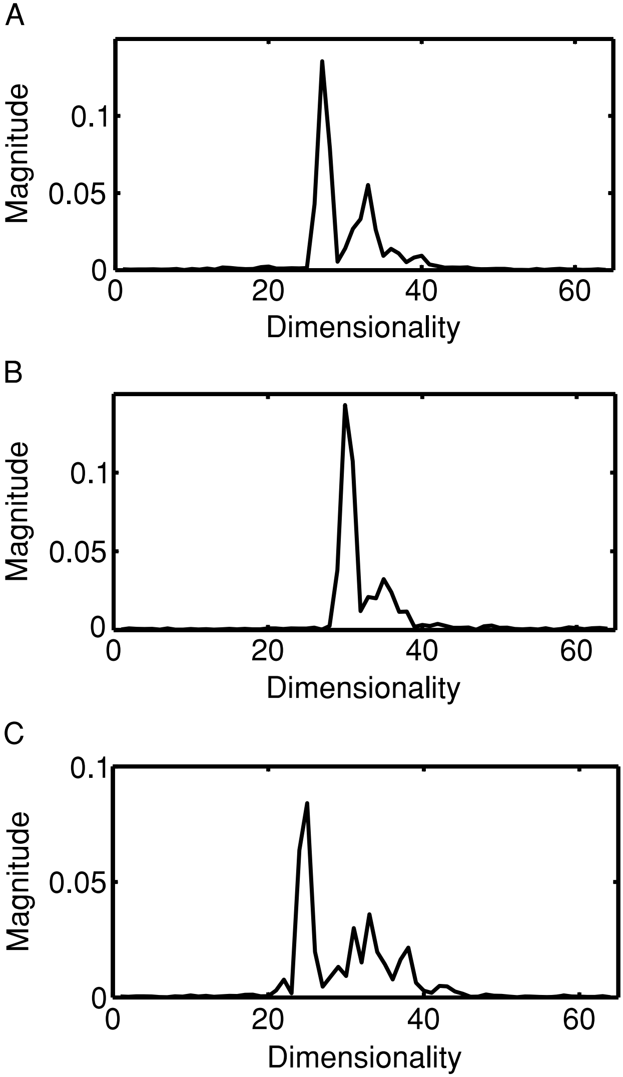

Figure 3: (A) One-dimensional image of J-6. (B) One-dimensional image of J-7. (C) One-dimensional image of B-52.

It is well known that the kernel function plays an important role in KFD. Choosing different kernel functions may result in different performance. The determination of an appropriate kernel for a specific application is far from fully understood. Consequently, combining FS-KFD and kernel

Table 4: Number of Misclassifications of Several Classifiers on the Radar Target Recognition Problem.

Classifier	J-6/J-7/B-52
SVM (gaussian kernel)	11
LS-SVM (gaussian kernel)	11
RVM (gaussian kernel) (Tipping, 2001)	12
SPR (gaussian kernel) (Figueiredo, 2003)	12
KFD (gaussian kernel)	13
FS-KFD (feature-scaling gaussian kernel)	7

construction trick to improve the performance of KFD in a specific application is of potential importance.

One phenomenon worth mentioning is that the leave-one-out error resulting from the gradient-descent algorithm is smaller than the test error. The reason is that the leave-one-out error suffers from a large variance in small sample cases. If some countermeasure, such as regularization on the leave-one-out error is taken, this problem can be overcome. This is a topic we will pursue in the future research.

Acknowledgments

We thank two reviewers for their helpful comments that greatly improved the article and Lin Shi for her help in proofreading the manuscript. This work was supported by the National Natural Science Foundation of China under grants 60372050 and 60133010 and National 863 Project grant 2002AA135080.

References

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society, 68*, 337–404.

Bengio, Y. (2000). Gradient-based optimization of hyper-parameters. *Neural Computation, 12*, 1889–1900.

Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science. Available online at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Cawley, G. C., & Talbot, N. L. C. (2003). Efficient leave-one-out cross validation of kernel Fisher discriminant classifiers. *Pattern Recognition, 36*, 2585–2592.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning, 46*, 131–159.

Figueiredo, M. A. T. (2003). Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*, 1150–1159.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual of Eugenics, 7*, 179–188.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Orlando, FL. Academic Press.

Gestel, T. V., Suykens, J., Lanckriet, G., Lambrechts, A., Moor, B. D., & Vandewalle, J. (2002). Bayesian framework for least squares support vector machine classifiers, gaussian processes and kernel Fisher discriminant analysis. *Neural Computation, 15*, 1115–1148.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks, 13*, 415–425.

Krishnapuram, B., Hartemink, A., Carin, L., & Figueiredo, M. (2004). A Bayesian approach to joint feature selection and classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*, 1105–1111.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research, 5*, 27–72.

Lowe, D. (1995). Similarity metric learning for a variable-kernel classifier, *Neural Computation, 7*, 72–85.

Luntz, A., & Brailovsky, V. (1969). On estimation of characters obtained in statistical procedure of recognition. (In Russian). *Techicheskaya Kibernetica*, 3.

Mika, S. (2002). *Kernel fisher discriminants*. Unpublished doctoral dissertation, University of Technology, Berlin.

Mika, S., Ratsch, G., & Weston, J. (1999). Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing* (pp. 41–48). Piscataway, NJ: IEEE Press.

Ong, C. S., & Smola, A. J. (2003). Machine learning with hyperkernels. In *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 568–575). Menlo Park, CA: AAAI Press.

Rätsch, G. (2001). *Robust boosting via convex optimization*. Unpublished doctoral dissertation, University of Potsdam, Potsdam, Germany.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research, 5*, 101–141.

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

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers, *Neural Processing Letters, 9*, 293–300.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal Machine Learning Research, 1*, 211–244.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Vapnik, V. (1998). *Statistical learning theory*, New York: Wiley.

Vapnik, V., & O. Chapelle. (2000). Bounds on error expectation for support vector machines. *Neural Computation, 12*, 2013–2036.

Weston, J., Mukherjee, M., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13* (pp. 668–674), Cambridge, MA: MIT Press.

Williams, C. K. I., & Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*, 1342–1351.

Xu, J., Zhang, X., & Li, Y. (2001). Kernel MSE algorithm: A unified framework for KFD, LS-SVM and KRR. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 1486–1491). Piscataway, NJ: IEEE Press.

Received October 26, 2004; accepted August 4, 2005.