Optimally Regularised Kernel Fisher Discriminant Classification

Kamel Saadi [a] Nicola L. C. Talbot [a] Gavin C. Cawley [a,*]

[a] *School of Computing Sciences, University of East Anglia, Norwich, Norfolk, U.K. NR4 7TJ*

[*] Corresponding author, tel: +44 (0)1603 593258, fax: +44 (0)1603 592245, email: gcc@cmp.uea.ac.uk

Abstract

Mika *et al.* [1] introduce a non-linear formulation of Fisher's linear discriminant, based the now familiar "kernel trick", demonstrating state-of-the-art performance on a wide range of real-world benchmark datasets. In this paper, we extend an existing analytical expression for the leave-one-out cross-validation error [2] such that the leave-one-out error can be re-estimated following a change in the value of the regularisation parameter with a computational complexity of only $\mathcal{O}(\ell^2)$ operations, which is substantially less than the $\mathcal{O}(\ell^3)$ operations required for the basic training algorithm. This allows the regularisation parameter to be tuned at an essentially negligible computational cost. This is achieved by performing the discriminant analysis in *canonical form*. The proposed method is therefore a useful component of a model selection strategy for this class of kernel machines that alternates between updates of the kernel and regularisation parameters. Results obtained on real-world and synthetic benchmark datasets indicate that the proposed method is competitive with model selection based on k-fold cross-validation in terms of generalisation, whilst being considerably faster.

Key words: model selection, cross-validation, least-squares support vector machine

1 Introduction

In recent years the "kernel trick" has been applied to construct non-linear equivalents of a wide range of classical linear statistical models, for instance ridge regression [3, 4], principal component analysis [5, 6] and Fisher's linear discriminant [1, 7], in addition to more modern techniques, such as the maximal margin classifier [8, 9] (for an introduction to kernel learning methods, see Schölkopf and Smola [10] or Shawe-Taylor and Cristianini [11]). An important advantage of kernel models is that the parameters of the model are typically given by the solution of a convex optimisation problem, with a single, global optimum [12]. The generalisation properties of kernel models are however typically governed by a small number of regularisation and kernel parameters. Good values for these parameters must be determined during the *model selection* process. There is generally no guarantee that the model selection criterion is unimodal, and so simple grid-based search procedures are often employed in practical applications. In this paper, we propose a simple and computationally efficient method for choosing the regularisation parameter in kernel Fisher discriminant analysis so as to minimise an approximation to the leave-one-out cross-validation error. The resulting optimally regularised kernel Fisher discriminant (ORKFD) analysis algorithm then becomes attractive for small to medium-scale applications (currently anything less than a few thousand training patterns) as the algorithm is easily implemented (only 15 lines of code in the MATLAB programming environment) and inherently resistant to over-fitting.

The remainder of this paper is structured as follows: Section 2 reviews the kernel Fisher discriminant classifier and introduces the notation used throughout.

Section 3 then proposes an efficient algorithm for selecting the regularisation for a KFD classifier, so as to minimise the leave-one-out cross-validation error, with a computational complexity of only $\mathcal{O}(\ell^2)$ operations instead of the $\mathcal{O}(\ell^3)$ operations of direct methods [1] [2]. Section 4 presents results obtained on a range of real-world benchmark datasets. The extension of this approach to closely related forms of least-squares kernel learning is discussed in Section 5. Finally, the work is summarised in section 6.

2 The Kernel Fisher Discriminant Classifier

Assume we are given training data $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell\} = \{\mathcal{X}_1, \mathcal{X}_2\} \subset \mathbb{R}^d$, where $\mathcal{X}_1 = \{\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \ldots, \boldsymbol{x}_{\ell_1}^1\}$ is a set of patterns belonging to class $\mathcal{C}_1$ and similarly $\mathcal{X}_2 = \{\boldsymbol{x}_1^2, \boldsymbol{x}_2^2, \ldots, \boldsymbol{x}_{\ell_2}^2\}$ is a set of patterns belonging to class $\mathcal{C}_2$; Fisher's linear discriminant (FLD) attempts to find a linear combination of input variables, $\boldsymbol{w} \cdot \boldsymbol{x}$, that maximises the average separation of the projections of points belonging to $\mathcal{C}_1$ and $\mathcal{C}_2$, whilst minimising the within class variance of the projections of those points. The Fisher discriminant is given by the vector $\boldsymbol{w}$ maximising

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}, \tag{1}$$

[1] The KFD classifier can also be trained using iterative methods, such as conjugate gradient descent, with a lower computational complexity [13]. However, using such methods the leave-one-out error can no longer be computed efficiently in closed form, and so in the remainder of this paper it is assumed that a direct approach, such as Cholesky factorisation, is taken.

where $\boldsymbol{S}_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$, is the between class scatter matrix, $\boldsymbol{m}_j$ is the mean of patterns belonging to $\mathcal{C}_j$,

$$\boldsymbol{m}_j = \frac{1}{\ell_j}\sum_{i=1}^{\ell_j} \boldsymbol{x}_i^j,$$

and $\boldsymbol{S}_W$ is the within class scatter matrix

$$\boldsymbol{S}_W = \sum_{i\in\{1,2\}}\sum_{j=1}^{\ell_i}(\boldsymbol{x}_j^i - \boldsymbol{m}_i)(\boldsymbol{x}_j^i - \boldsymbol{m}_i)^T.$$

The innovation introduced by Mika *et al.* [1] is to construct Fisher's linear discriminant in a fixed feature space $\mathcal{F}$ ($\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$) induced by a positive definite *Mercer* kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ defining the inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$ (see e.g. Cristianini and Shawe-Taylor [14]). Let the kernel matrices for the entire dataset, $\boldsymbol{K}$, and for each class, $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ be defined as follows:

$$\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{\ell}$$

and

$$\boldsymbol{K}_i = [k_{jk}^i = \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_k^i)]_{j,k=1}^{j=\ell,k=\ell_i}.$$

The theory of reproducing kernels indicates that $\boldsymbol{w}$ can then be written as an expansion of the form

$$\boldsymbol{w} = \sum_{i=1}^{\ell}\alpha_i\boldsymbol{\phi}(\boldsymbol{x}_i). \tag{2}$$

The objective function (1) can also be written such that the data $\boldsymbol{x} \in \mathcal{X}$ appear only within inner products, giving

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T\boldsymbol{M}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\boldsymbol{N}\boldsymbol{\alpha}}, \tag{3}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_\ell]^T$, $\boldsymbol{M} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$, $\boldsymbol{m}_i = \boldsymbol{K}_i\boldsymbol{u}_i$, $\boldsymbol{u}_i$ is a column vector containing ℓ_i elements with a common value of ℓ_i^{-1} and

$$\boldsymbol{N} = \sum_{i\in\{1,2\}}\boldsymbol{K}_i(\boldsymbol{I} - \boldsymbol{U_i})\boldsymbol{K}_i^T,$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{U}_i$ is a matrix with all elements equal to ℓ_i^{-1}. The coefficients, $\boldsymbol{\alpha}$, of the expansion (2) are then given by the leading eigenvector of $\boldsymbol{N}^{-1}\boldsymbol{M}$. Note that $\boldsymbol{N}$ is likely to be singular, or at best ill-conditioned, and so a regularised solution is obtained by substituting $\boldsymbol{N}_\mu = \boldsymbol{N} + \mu\boldsymbol{I}$, where μ is a regularisation constant. To complete the kernel Fisher discriminant classifier, $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) + b$, the bias, b, is given by

$$b = -\boldsymbol{\alpha}\frac{\ell_1\boldsymbol{M}_1 + \ell_2\boldsymbol{M}_2}{\ell}.$$

Xu *et al.* [15] show that the parameters of the kernel Fisher discriminant classifier are also given by the solution of the following system of linear equations:

$$\begin{bmatrix} \boldsymbol{K}^T\boldsymbol{K} + \mu\boldsymbol{I} & \boldsymbol{K}^T\boldsymbol{1} \\ (\boldsymbol{K}^T\boldsymbol{1})^T & \ell \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^T \\ \boldsymbol{1}^T \end{bmatrix} \boldsymbol{y}, \qquad (4)$$

where $\boldsymbol{1}$ is a column vector of ℓ ones and $\boldsymbol{y}$ is a column vector with elements $y_i = \ell/\ell_j \; \forall i : \boldsymbol{x}_i \in \mathcal{X}_j$. This illustrates the similarities between the kernel Fisher discriminant and the least-squares support vector machine (LS-SVM) [16]. The kernel Fisher discriminant (KFD) classifier has been shown experimentally to demonstrate near state-of-the-art performance on a range of artificial and real world benchmark datasets [1] and so is worthy of consideration for small to medium scale applications. In this paper we present an efficient algorithm for approximate cross-validation of kernel Fisher discriminant models, providing a practical criterion for model selection.

3 Method

In this section, we describe a training algorithm for the kernel Fisher discriminant classifier in which the system of linear equations (4) is solved in *canonical form*. This allows the model parameters to be updated following a change in the value of the regularisation parameter with a computational complexity of only $\mathcal{O}(\ell)$ operations. This also permits the extension of an existing analytic method [2] for re-evaluation of the leave-one-out cross-validation error in only $\mathcal{O}(\ell^2)$ operations, rather than the $\mathcal{O}(\ell^3)$ operations of the existing analytic method [2], or the $\mathcal{O}(\ell^4)$ of a naïve direct implementation. This is used to form the basis of a criterion for gradient descent optimisation of the regularisation parameter μ, at an essentially negligible computational expense.

3.1 The Kernel Fisher Discriminant in Canonical Form

Neglecting the bias parameter, the system of linear equations (4) can be written more concisely in the form

$$\boldsymbol{\alpha} = \left[\boldsymbol{K}^T\boldsymbol{K} + \mu\boldsymbol{I}\right]^{-1}\boldsymbol{K}^T\boldsymbol{y}, \tag{5}$$

Let $\boldsymbol{V}$ be an orthogonal matrix, the columns of which are the eigenvectors of $\boldsymbol{K}^T\boldsymbol{K}$, and $\boldsymbol{\Lambda}$ be a diagonal matrix containing the corresponding eigenvalues $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_\ell \geq 0$, such that

$$\boldsymbol{K}^T\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T, \quad \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}.$$

The principal components of $\boldsymbol{K}$ are then given by the columns of $\boldsymbol{U} = \boldsymbol{K}\boldsymbol{V}$; note that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{\Lambda}$. The system of linear equations (5) can then be expressed in *canonical form* (following the methods and terminology used in linear regression [17]) as

$$\boldsymbol{\beta} = \boldsymbol{C}^{-1}\boldsymbol{U}^T\boldsymbol{y} = [\boldsymbol{\Lambda} + \mu\boldsymbol{I}]^{-1}\boldsymbol{U}^T\boldsymbol{y}, \tag{6}$$

where $\boldsymbol{\beta} = \boldsymbol{V}^T\boldsymbol{\alpha}$. Note that a similar result could be obtained via singular value decomposition [18] of the kernel matrix, $\boldsymbol{K}$. The principal advantage of expressing the system of linear equations (5) in this form is that the matrix $\boldsymbol{C}$ is diagonal, and so can be inverted in linear time, i.e. $\mathcal{O}(\ell)$ operations. Let $\boldsymbol{\beta}^0$ represent the canonical form parameters for a KFD classifier with regularisation parameter μ^0. Furthermore, let υ_i represent the i^{th} element of $\boldsymbol{U}^T\boldsymbol{y}$, then as $[\boldsymbol{\Lambda} + \mu\boldsymbol{I}]$ is a diagonal matrix,

$$\beta_i^0 = \left(\lambda_i + \mu^0\right)^{-1}\upsilon_i,$$

and similarly, we can write

$$\frac{\beta_i}{\beta_i^0} = \frac{(\lambda_i + \mu)^{-1}\upsilon_i}{(\lambda_i + \mu^0)^{-1}\upsilon_i} = \frac{\lambda_i + \mu^0}{\lambda_i + \mu}$$

The canonical parameters of the kernel Fisher discriminant classifier with an arbitrary regularisation parameter can be found in linear time by scaling the parameters of a "reference" kernel Fisher discriminant (in canonical form) with regularisation parameter μ^0,

$$\beta_i = \frac{\lambda_i + \mu^0}{\lambda_i + \mu}\beta_i^0, \qquad i = 0, 1, 2, \ldots, \ell.$$

It should be noted that adopting the canonical form (6), the parameters of the kernel Fisher discriminant model in original form, $\boldsymbol{\alpha} = \boldsymbol{V}\boldsymbol{\beta}$, can be updated following a change in the regularisation parameter, μ, with a computational complexity of only $\mathcal{O}(\ell^2)$ operations. If instead we work with the equivalent canonical form parameters $\boldsymbol{\beta}$, the vector of regularisation parameters can be updated in *linear time*, i.e. $\mathcal{O}(\ell)$ operations.

3.2 Efficient Leave-One-Out Cross-Validation

In this section, we review an analytic expression for the leave-one-out cross-validation error of a kernel Fisher discriminant classifier introduced by Cawley and Talbot [2] (c.f. [17, 19]), and demonstrate that in canonical form it can be evaluated with a computational complexity of only $\mathcal{O}(\ell^2)$ operations, rather than the $\mathcal{O}(\ell^3)$ operations required using the original parameterisation. At each step of the leave-one-out cross-validation procedure, a kernel Fisher discriminant classifier is constructed excluding a single training pattern from the data. The vector of canonical model parameters, $\boldsymbol{\omega}_{(i)}$ at the i^{th} step, in which pattern i is excluded, is then given by the solution of a modified system of linear equations,

$$\boldsymbol{\beta}_{(i)} = \left[\mu \boldsymbol{I} + \boldsymbol{U}_{(i)}^T \boldsymbol{U}_{(i)}\right]^{-1} \boldsymbol{U}_{(i)}^T \boldsymbol{y}$$

where $\boldsymbol{U}_{(i)}$ is the sub-matrix formed by omitting the i^{th} row of $\boldsymbol{U}$. Note that $\boldsymbol{U}_{(i)}^T \boldsymbol{U}_{(i)}$ is in general no longer diagonal, and so the most computationally expensive step is normally the inversion of the matrix $\boldsymbol{C}_{(i)} = \left[\mu \boldsymbol{I} + \boldsymbol{U}_{(i)}^T \boldsymbol{U}_{(i)}\right]$, with a complexity of $\mathcal{O}(\ell^3)$ operations. Fortunately $\boldsymbol{C}_{(i)}$ can be written as a rank one modification of a matrix $\boldsymbol{C}$,

$$\boldsymbol{C}_{(i)} = \left[\mu \boldsymbol{I} + \boldsymbol{U}^T \boldsymbol{U} - \boldsymbol{u}_i \boldsymbol{u}_i^T\right] = \left[\boldsymbol{C} - \boldsymbol{u}_i \boldsymbol{u}_i^T\right], \tag{7}$$

where $\boldsymbol{u}_i$ is the i^{th} row of $\boldsymbol{U}$. The Bartlett matrix inversion lemma,

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}, \tag{8}$$

then allows $\boldsymbol{C}_{(i)}^{-1}$ to be found in only $\mathcal{O}(\ell^2)$ operations, given that $\boldsymbol{C}^{-1}$ is already known. Applying the (8) to the matrix inversion problem given in (7), we obtain

$$\boldsymbol{C}_{(i)}^{-1} = [\boldsymbol{C} - \boldsymbol{u}_i\boldsymbol{u}_i^T]^{-1} = \boldsymbol{C} + \frac{\boldsymbol{C}^{-1}\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{C}^{-1}}{1 - \boldsymbol{u}_i^T\boldsymbol{C}^{-1}\boldsymbol{u}_i}.$$

The computational complexity of the leave-one-out cross-validation process is thus reduced to only $\mathcal{O}(\ell^3)$ operations, which is the same as that of the basic training algorithm for the kernel Fisher discriminant classifier. For model selection purposes, however, we are not principally concerned with the values of the model parameters themselves, but only statistics such as the leave-one-out error rate

$$E_{loo} = \frac{1}{\ell}\sum_{i=1}^{\ell}\{1 - \Psi(1 - \mathrm{sign}(y_i)\left\{\boldsymbol{r}_{(i)}\right\}_i)\},$$

where Ψ is the Heaviside or unit step function,

$$\Psi(x) = \begin{cases} 1 \; x \geq 0 \\ \\ 0 \; x < 0 \end{cases}.$$

and $\left\{\boldsymbol{r}_{(i)}\right\}_i = \mathrm{sign}(y_i) - \boldsymbol{w}_{(i)} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) - b_{(i)}$ is the residual error for the i^{th} training pattern during the i^{th} iteration of the leave-one-out cross-validation procedure. It can be shown that

$$\left\{\boldsymbol{r}_{(i)}\right\}_i = \frac{1}{1 - h_{ii}} r_i.$$

where $r_i = \mathrm{sign}(y_i) - \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) - b$ is the residual error for the i^{th} training pattern for a kernel Fisher discriminant classifier trained on the entire dataset, and $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{C}^{-1}\boldsymbol{U}^T$ is the *hat* matrix of which h_{ii} is the i^{th} element of the leading diagonal [17]. In this case, $\boldsymbol{C}$ is diagonal and can be inverted in linear time, and therefore an individual element of the diagonal of the "hat" matrix can also be computed in linear time,

$$h_{ii} = \sum_{j=1}^{\ell} u_{ij}^2 c_{jj}^{-1} = \sum_{j=1}^{\ell} \frac{u_{ij}^2}{(\lambda_j + \mu)}.$$

The diagonal elements of the "hat" matrix can then be computed with a complexity of $\mathcal{O}(\ell^2)$ operations. Note that $\boldsymbol{r}$ can also be written as $\boldsymbol{r} = \mathrm{sign}(\boldsymbol{y}) - \boldsymbol{H}\boldsymbol{y}$ and hence

$$r_i = \mathrm{sign}(y_i) - \sum_{j=1}^{\ell}\sum_{k=1}^{\ell} \frac{u_{ik}u_{jk}}{\lambda_k + \mu} y_j = \mathrm{sign}(y_i) - \sum_{j=1}^{\ell} \frac{1}{\lambda_j + \mu} \sum_{k=1}^{\ell} u_{ij}u_{kj}y_k.$$

Provided that one pre-computes $\left[\sum_{k=1}^{\ell} u_{ij}u_{kj}y_k\right]_{i,j=1}^{\ell}$, which does not depend on μ, the residuals for a classifier trained on the entire dataset, $\boldsymbol{r}$, can also be computed with a complexity of $\mathcal{O}(\ell^2)$ operations. The leave-one-out error rate can thus be evaluated in closed form without explicit inversion of $\boldsymbol{C}_{(i)}$ $\forall i \in \{1, 2, \ldots, \ell\}$, with a computational complexity of only $\mathcal{O}(\ell^2)$ operations. This would not be the case for kernel Fisher discriminant analysis performed in the original parameterisation, where the computational complexity would be $\mathcal{O}(\ell^3)$ because $\boldsymbol{C}$ would no longer be a diagonal matrix.

3.3 Gradient Descent Optimisation

While the leave-one-out error (or an upper bound thereof) has been found to be an effective model selection criterion for a range of kernel machines, including kernel Fisher discriminant [20] and support vector machines [21], it is discrete and therefore it is difficult to form an efficient automated model selection procedure. In order to allow an efficient gradient descent model selection scheme, we approximate the Heaviside unit step function using a compressed sigmoidal logistic function,

$$f(x) = \frac{1}{1 + \exp\{-\gamma x\}}$$

as shown in figure 1, where γ represents the compression factor; clearly $f(x)$ approaches the Heaviside step function, $\Psi(x)$, increasingly closely as γ becomes large.

$$E = \frac{1}{\ell}\sum_{i=1}^{\ell}\{1 - f(1 - \text{sign}(y_i)\left\{\boldsymbol{r}_{(i)}\right\}_i)\} \approx E_{loo}. \tag{9}$$

For convenience we will assume, as is normally the case, a single regularisation parameter μ; it is then straight-forward to obtain the derivative of the modified model selection criterion with respect to the regularisation parameter:

$$\frac{\partial E}{\partial \mu} = \frac{1}{\ell}\sum_{i=1}^{\ell} f'(1 - \text{sign}(y_i)\left\{\boldsymbol{r}_{(i)}\right\}_i)\,\text{sign}(y_i)\frac{\partial}{\partial \mu}\left\{\boldsymbol{r}_{(i)}\right\}_i,$$

where

$$\begin{aligned}
\frac{\partial}{\partial \mu}\left\{\boldsymbol{r}_{(i)}\right\}_i &= \frac{r_i}{(1-h_{ii})^2}\frac{\partial h_{ii}}{\partial \mu} + \frac{1}{1-h_{ii}}\frac{\partial r_i}{\partial \mu}, \\
\frac{\partial h_{ii}}{\partial \mu} &= -\sum_{j=1}^{\ell}\frac{u_{ij}^2}{(\lambda_j + \mu)^2}, \\
\frac{\partial r_i}{\partial \mu} &= \sum_{j=1}^{\ell}\sum_{k=1}^{\ell}\frac{u_{ik}u_{jk}}{(\lambda_k + \mu)^2}y_j, \\
f'(x) &= \gamma[f(x)(1 - f(x))].
\end{aligned}$$

Likewise, the second order derivative is given by

$$\begin{aligned}
\frac{\partial^2 E}{\partial \mu^2} = \frac{1}{\ell}\sum_{i=1}^{\ell}\Bigg\{ & -f''(1 - \text{sign}(y_i)\left\{\boldsymbol{r}_{(i)}\right\}_i)\left[\frac{\partial}{\partial \mu}\left\{\boldsymbol{r}_{(i)}\right\}_i\right]^2 \\
& + f'(1 - \text{sign}(y_i)\left\{\boldsymbol{r}_{(i)}\right\}_i)\,\text{sign}(y_i)\frac{\partial^2}{\partial \mu^2}\left\{\boldsymbol{r}_{(i)}\right\}_i\Bigg\},
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial^2}{\partial\mu^2}\left\{\boldsymbol{r}_{(i)}\right\}_i &= \frac{2r_i}{(1-h_{ii})^3}\left[\frac{\partial h_{ii}}{\partial\mu}\right]^2 + \frac{r_i}{(1-h_{ii})^2}\frac{\partial^2 h_{ii}}{\partial\mu^2} \\
&\quad + \frac{2}{(1-h_{ii})^2}\frac{\partial h_{ii}}{\partial\mu}\frac{\partial r_i}{\partial\mu} + \frac{1}{1-h_{ii}}\frac{\partial^2 r_i}{\partial\mu^2}, \\
\frac{\partial^2 h_{ii}}{\partial\mu^2} &= 2\sum_{j=1}^{\ell}\frac{u_{ij}^2}{(\lambda_j+\mu)^3}, \\
\frac{\partial^2 r_i}{\partial\mu^2} &= -2\sum_{j=1}^{\ell}\sum_{k=1}^{\ell}\frac{u_{ik}u_{jk}}{(\lambda_k+\mu)^3}y_j, \\
f''(x) &= \gamma^2 f(x)(1-f(x))(1-2f(x)).
\end{aligned}$$

The locally optimal value of regularisation parameter can then be determined using a simple Newton-Raphson second-order gradient descent optimisation procedure,

$$\mu_{t+1} = \mu_t - \eta\left[\frac{\partial^2 E}{\partial\mu^2}\right]^{-1}\frac{\partial E}{\partial\mu}.$$

In practise, we adopt a simple step halving heuristic to ensure convergence to a local minima. At each iteration, we begin by making a full Newton step, i.e. $\eta = 1$, if this does not reduce the value of the model selection criterion, E, a step is made in the same direction, but of half the original magnitude, i.e. $\eta \leftarrow 0.5\eta$. This process is repeated until a reduction in the value of the model selection criterion is obtained, or the number of step halvings exceeds a predetermined threshold. Furthermore, for a smoother minima search, instead of differentiating with respect to μ we differentiate with respect to $\log(\mu)$. This also ensures that μ takes on only positive values.

$$\log(\mu_{t+1}) = \log(\mu_t) - \eta\left[\frac{\partial^2 E}{\partial\log(\mu)^2}\right]^{-1}\frac{\partial E}{\partial\log(\mu)}$$

where

$$\begin{aligned}
\frac{\partial}{\partial\log(\mu)} &= \mu\frac{\partial}{\partial\mu}, \\
\frac{\partial^2}{\partial\log(\mu)^2} &= \mu\frac{\partial}{\partial\mu} + \mu^2\frac{\partial^2}{\partial\mu^2}.
\end{aligned}$$

3.4 Accommodating an Unregularised Bias Parameter

Accommodating a *regularised* bias parameter to the canonical form kernel Fisher discriminant is relatively straignt-forward; we simply augment the kernel matrix with a vector of ones, such that $\tilde{\boldsymbol{K}} \leftarrow [\boldsymbol{K}\ \mathbf{1}]$ and procede as before. However, a regularised bias term is somewhat inelegant, as the regularisation term is primarily intended to express a preference for relatively smooth functions, a property of the model that is independent of the bias. The incorporation of an *unregularised* bias term is however rather more difficult. The model parameters of the kernel Fisher discriminant are given by the solution of the the system of linear equations, (4). In canonical form, the kernel Fisher discriminant including an unregularised bias parameter, is given by

$$\begin{bmatrix} \boldsymbol{\Lambda} + \mu \boldsymbol{I} & \boldsymbol{U}^T\mathbf{1} \\ \mathbf{1}^T\boldsymbol{U} & \ell \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}^T\boldsymbol{y} \\ \mathbf{1}^T\boldsymbol{y} \end{bmatrix},$$

where the definitions of $\boldsymbol{\Lambda}$, $\boldsymbol{U}$, and $\boldsymbol{\beta}$ are the same as those given in the previous section. The Hat matrix, which transforms the targets onto the outputs, $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$, is then given by

$$\boldsymbol{H} = [h_{ij}]_{i,j=1}^{\ell} = \begin{bmatrix} \boldsymbol{U} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda} + \mu \boldsymbol{I} & \boldsymbol{U}^T\mathbf{1} \\ \mathbf{1}^T\boldsymbol{U} & \ell \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{U}^T \\ \mathbf{1}^T \end{bmatrix}.$$

The leave-one-out cross-validation behaviour of the sparse kernel Fisher discriminant is governed by

$$y_i - \hat{y}_i^{(-i)} = \frac{y_i - \hat{y}_i}{1 - h_{ii}}$$

and so in order to evaluate a leave-one-out cross-validation based model selection criterion, the model output $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$ and the diagonal elements of the Hat matrix, $\boldsymbol{H}$, are required. These can be obtained via the block matrix inversion formula, giving

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{U}^T\mathbf{1} \\ \mathbf{1}^T\boldsymbol{U} & \ell \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{M}^{-1} + \boldsymbol{M}^{-1}\boldsymbol{U}^T\mathbf{1}S_M^{-1}\mathbf{1}^T\boldsymbol{U}\boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\boldsymbol{U}^T\mathbf{1}S_M^{-1} \\ -S_M^{-1}\mathbf{1}^T\boldsymbol{U}\boldsymbol{M}^{-1} & S_M^{-1} \end{bmatrix},$$

where $\boldsymbol{M} = \boldsymbol{\Lambda} + \mu\boldsymbol{I}$ and $S_M = \ell - \mathbf{1}\boldsymbol{M}^{-1}\mathbf{1}$ is the Schur complement of $\boldsymbol{M}$. The diagonal entries of the hat matrix can then be written as

$$h_{ii} = \sum_{j=1}^{n} \frac{u_{ij}^2}{\lambda_j + \mu} + \frac{1}{S_M}\left[\eta_i^2 - 2\eta_i + 1\right],$$

where

$$S_m = \ell - \sum_{i=1}^{n} \frac{\zeta_i^2}{\lambda_i + \mu}, \qquad \eta_i = \sum_{j=1}^{n} \frac{u_{ij}\zeta_j}{\lambda_j + \mu} \qquad \text{and} \qquad \zeta_i = \sum_{j=1}^{\ell} u_{ji}.$$

Likewise, the output of the kernel Fisher discriminant is given by

$$\hat{y}_i = \nu_i + \frac{\eta_i - 1}{S_M}\left[\sum_{j=1}^{\ell} \eta_j y_j - \sum_{j=1}^{\ell} y_j\right],$$

where

$$\xi_i = \sum_{j=1}^{n} u_{ji} y_j \qquad \text{and} \qquad \nu_i = \sum_{j=1}^{n} \frac{u_{ij}\xi_j}{\lambda_j + \mu}.$$

The individual elements of the principal diagonal of the Hat matrix, $\boldsymbol{H}$, and of $\hat{\boldsymbol{y}}$, can be computed with a computational complexity of $\mathcal{O}(\ell)$ operations. Once the model has been expressed in canonical form, therefore, the leave-one-out cross-validation estimate of the model selection criterion can be evaluated with a complexity of only $\mathcal{O}(\ell^2)$ operations, although the algorithm is a little more complex than for the kernel Fisher discriminant without an unregularised bias parameter.

3.5 Computational Complexity

In order to obtain a well-regularised kernel Fisher discriminant classifier using the original parameterisation, we must train the classifier and evaluate the leave-one-out error N times, if we are to search for the best of N candidate values for the regularisation term. The most expensive step in training the KFD lies in the solution of a set of linear equations, with a computational complexity of $\mathcal{O}(\ell^3)$ operations (we will assume that the inverse of $\boldsymbol{C}$ produced as a by-product). In this case, the computation of the diagonal elements of the "hat" matrix, required in evaluation of the leave-one-out error, is also $\mathcal{O}(\ell^3)$ as $\boldsymbol{C}^{-1}$ is a full matrix. In this case, the only computation shared between models with different regularisation parameters lies in the evaluation of the kernel matrix, $\boldsymbol{K}$, and $\boldsymbol{K}^T\boldsymbol{K}$, $\mathcal{O}(\ell^2)$ and $\mathcal{O}(\ell^3)$ processes respectively. On the other hand, if training is performed in *canonical form*, we first compute the eigendecomposition of $\boldsymbol{K}^T\boldsymbol{K}$, at a cost of $\mathcal{O}(\ell^3)$ operations. We may then evaluate the leave-one-out error for N values for the regularisation parameter, μ, with a complexity of only $\mathcal{O}(\ell^2)$ operations. The canonical parameters for the optimal value of the regularisation parameter are then given by the solution of a *diagonal* system of linear equations, with a complexity of $\mathcal{O}(\ell)$ operations, which are then transformed to give the coefficients of the original kernel expansion, with a complexity of $\mathcal{O}(\ell^2)$ operations. The overall computational complexity of obtaining a well regularised classifier is $\mathcal{O}(\ell^3)$, regardless of the parameterisation, however the canonical parameterisation is faster in practise as the eigen-decomposition of $\boldsymbol{K}^T\boldsymbol{K}$ is only performed once, and the cost amortised over the evaluation of the N candidate values of the regularisation parameter.

3.5.1 Sparse Kernel Models

The training algorithm of the kernel Fisher discriminant classifier exhibits a computational complexity of $\mathcal{O}(\ell^3)$ operations, which makes it suitable for only small or medium-scale applications (currently up to a few thousand training patterns). For larger-scale applications, a sparse kernel expansion can be used instead, where only a sub-set of the available training patterns are used (see e.g. [22]). The computational complexity of the training algorithm then falls to only $\mathcal{O}(\ell n^2)$, where n is the number of non-zero terms remaining in the kernel expansion. There are a variety of means by which a sub-set of training patterns can be selected, from random selection, incomplete Cholesky factorisation [23] or greedy selection (see e.g. [10, 24]). The proposed scheme for selecting an optimal value for the regularisation parameter can be applied essentially unaltered in the case of a sparse kernel machine; the overall computational complexity remains $\mathcal{O}(\ell n^2)$, however the cost of expressing the model in canonical form can again be amortised over the evaluation of a number of candidate values for the regularisation parameter, allowing a very efficient search for its optimal value. The computational advantage of the proposed method is thus exactly the same for sparse and fully dense kernel models. A similar scheme has recently been proposed independently by Wang *et al.* [25].

4 Results

The runtime for model selection based on the proposed optimally regularised kernel Fisher discriminant classifier is evaluated over a series of randomly generated synthetic datasets. In each case, approximately one quarter of the data belong to class $\mathcal{C}_1$ and three-quarters to class $\mathcal{C}_2$. The patterns comprising class $\mathcal{C}_1$ are drawn from a bivariate Normal distribution with zero mean and unit variance. The patterns forming class $\mathcal{C}_2$ form an annulus; the radii of the data are drawn from a normal distribution with a mean of 4 and unit variance, and the angles uniformly distributed. The datasets vary in size between 8 and 4096 patterns, an example of which is shown in Figure 2.

Figure 3 shows a graph of the time taken to select the best value for the regularisation parameter, from 21 candidate values ($\mu \in \{2^i \mid i = -10, -9, \ldots, 0, \ldots, +9, +10\}$), for the ORKFD and for a conventional KFD using the existing analytic leave-one-out cross-validation procedure (without re-parameterisation into canonical form) [2]. The for large ℓ, the gradients of the lines representing total elapsed time for both the ORKFD and KFD are approximately three (on log-log axes), indicating that both algorithms have a computational complexity of $\mathcal{O}(\ell^3)$. The ORKFD is however approximately an order of magnitude faster, representing a significant advance over existing approaches (e.g. [2]). The dashed lines in Figure 3 represent the time spent exclusively on the search for the optimal value of the regularisation parameter, μ. For the conventional KFD, this is almost equal to the total time as there is relatively little shared computational effort in estimating the leave-one-out error for models differing only in the value of μ. In this case, the only significant computational expense that can be amortised over the search lies in the evaluation of the kernel matrix and in computing $\boldsymbol{K}^T\boldsymbol{K}$. For the ORKFD, on the other hand, once the model has been expressed in canonical form, the leave-one-out cross-validation error can be evaluated very efficiently for different values of μ. For large ℓ, the gradient of the dashed line representing ORKFD search time is approximately two (on log-log axes), indicating that this procedure has a computational complexity of only $\mathcal{O}(\ell^2)$ operations. This means that, for large ℓ, the cost of optimising the value of the regularisation parameter becomes essentially negligible (e.g. 0.4% of the total run-time for $\ell = 4096$).

It is known that *down-dating* a system of linear equations, as occurs in the analytic leave-one-out cross-validation process, can be numerically unstable [26], and so we also investigate the stability of the algorithm, with respect to direct implementation of the leave-one-out cross-validation procedure. Let the relative approximation error be defined as

$$e = \frac{\|\tilde{\boldsymbol{r}} - \hat{\boldsymbol{r}}\|^2}{\|\tilde{\boldsymbol{r}}\|^2},$$

where $\tilde{\boldsymbol{r}}$ is a vector of leave-one-out residual errors computed via the direct approach and $\hat{\boldsymbol{r}}$ is the corresponding vector of residual errors resulting from the proposed method. Figure 4 shows a graph of the mean relative approximation error, as a function of the number of training patterns. The approximation error is small for datasets of more than ≈ 30 training patterns.

In order to verify that the optimally regularised kernel Fisher discriminant classifier is competitive in terms of generalisation, it is evaluated against a range of leading classification methods over a suite of 13 real world and synthetic benchmark problems from the UCI repository [27]. We adopt the experimental procedure used in the study by Rätsch *et al.* [28], where 100 different random training and test splits are defined (20 in the case of the large-scale image and splice datasets). Model selection is performed on the first five training splits, taking the median of the estimated values for the optimal regularisation (γ) and kernel (σ) parameters. Generalisation is then measured by the mean error rate over the 100 test splits (20 for image and splice datasets). The benchmarks, including test and training splits are available from `http://ida.first.fhg.de/projects/bench`. The results obtained are also compared with those from Mika *et al.* [29], including kernel Fisher discriminant models where the model selection procedure minimised a 10-fold cross-validation estimate of the test error rate, as well as a range of other state-of-the art classification algorithms, namely radial basis function networks (RBF), AdaBoost [30] using RBF networks (AB), LP-AdaBoost [31] (ABL), QP-AdaBoost [28] (ABQ), regularised AdaBoost [28] (AB$_R$) and the support vector machine with radial basis kernel [8, 9, 32] (SVM) [2] .

Table 1 shows the outcome of a comparison of model selection procedures for optimally regularised kernel Fisher discriminant (ORKFD) models and a range of state-of-the-art statistical pattern recognition algorithms. The ORKFD outperforms the KFD with 10-fold cross-validation (sum of squares) model selection (KFD) on seven of the thirteen datasets (banana, german, heart, image, ringnorm, titanic and waveform), and performs worse on six (breast cancer, diabetis, solar flare, splice, thyroid and twonorm). This clearly demonstrates that the optimally regularised kernel Fisher discriminant classifier is competitive, in terms of generalisation, with conventional kernel Fisher discriminant classifiers with a 10-fold cross-validation based model selection strategy adopted by Mika *et al.* [29]. This performance is however achieved at a greatly reduced computational expense. The superior performance of the ORKFD method, against the range of state-of-the-art algorithms, should also be noted, providing the lowest error rate on five of the thirteen datasets and the second best on three of the remaining benchmarks.

[2] Details of simulation parameters are also available from `http://ida.first.fhg.de/projects/bench`.

5 Optimal Regularisation for Related Formulations

The use of the eigendecomposition, or equivalently the singular value decomosition, to isolate the effect of the regularisation term can also be used to develop optimally regularised variants of kernel ridge regression [4] (also known as the regularization network [33] and regularised least squares [34]) and the least-squares support vector machine [16]. Kernel ridge regression [4] constructs a kernel model, without a bias term, minimising a least-squares loss function with a regularisation term acting on the primal model parameters rather than the dual parameters. The dual model parameters are then given by the solution of the following system of linear equations

$$\left[\boldsymbol{K} + \mu\boldsymbol{I}\right]\boldsymbol{\alpha} = \boldsymbol{y}.$$

Let $\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ represent the eigendecomposition of $\boldsymbol{K}$, where $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ represent the eigenvectors and eigenvalues as before. Substituting, and noting that $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, we obtain

$$\left[\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T + \mu\boldsymbol{I}\right]\boldsymbol{\alpha} = \boldsymbol{V}\left[\boldsymbol{\Lambda} + \mu\boldsymbol{V}^T\boldsymbol{I}\boldsymbol{V}\right]\boldsymbol{V}^T\boldsymbol{\alpha} = \boldsymbol{V}\left[\boldsymbol{\Lambda} + \mu\boldsymbol{I}\right]\boldsymbol{V}^T\boldsymbol{\alpha} = \boldsymbol{y}$$

So in cannonical form, the parameters of the kernel ridge regression model are given by

$$\boldsymbol{\beta} = \left[\boldsymbol{\Lambda} + \mu\boldsymbol{I}\right]^{-1}\boldsymbol{\xi}$$

where $\boldsymbol{\beta} = \boldsymbol{V}\boldsymbol{\alpha}$ and $\boldsymbol{\xi} = \boldsymbol{V}^T\boldsymbol{y}$. Again, since $\boldsymbol{\Lambda}$ is a diagonal matrix, the canonical parameters can be updated following a change in the regularisation parameter with a computational complexity of only $\mathcal{O}(\ell)$ operations. This formed a component of an alternative convex approach to selection of the regularisation parameter using a validation set [35]. Alternatively, the optimal value for the regularisation parameter, μ, can be found by minimising the leave-one-out cross-validation estimate of the model selection criterion. The leave-one-out cross-validation behaviour of the kernel ridge regression model is governed by (e.g. [36])

$$y_i - \hat{y}_i^{(i)} = \frac{\alpha_i}{\boldsymbol{C}_{ii}^{-1}}$$

where $\boldsymbol{C} = [\boldsymbol{K} + \mu \boldsymbol{I}]$. However, as we are dealing with kernel ridge regression in canonical form, we can write

$$\alpha_i = \boldsymbol{v}_i \left[\boldsymbol{\Lambda} + \mu \boldsymbol{I}\right]^{-1} \boldsymbol{\xi} \qquad \text{and} \qquad C_{ii}^{-1} = \boldsymbol{v}_i \left[\boldsymbol{\Lambda} + \mu \boldsymbol{I}\right]^{-1} \boldsymbol{v}_i^T$$

where $\boldsymbol{v}_i$ is the i^{th} row of $\boldsymbol{V}$. Thus, leave-one-out cross-validation of least-squares support vector machines can be performed in canonical form, following a change in the value of the regularisation parameter, at a cost of only $\mathcal{O}(\ell^2)$ operations. This approach was also developed independently by Rifkin and Lippert [37].

5.1 Incorporating an Unregularised Bias Term

The least-squares support vector machine (LS-SVM) [16] includes an unregularised bias term, such that the model parameters are given by the solution of the following system of linear equations,

$$\begin{bmatrix} \boldsymbol{K} + \mu \boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 1 \end{bmatrix}.$$

This system of linear equations can be solved by solving two smaller, positive definite systems [13],

$$\boldsymbol{M}\boldsymbol{\eta} = \boldsymbol{1} \quad \text{and} \quad \boldsymbol{M}\boldsymbol{\nu} = \boldsymbol{y} \qquad \text{such that} \qquad b = \frac{\boldsymbol{1}^T\boldsymbol{\nu}}{\boldsymbol{1}^T\boldsymbol{\eta}} \quad \text{and} \quad \boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\eta} b,$$

where $\boldsymbol{M} = \boldsymbol{K} + \mu \boldsymbol{I}$. We may then obtain a canonical form training algorithm for the least-squares support vector machine including a bias term by performing an eigen-decomposition of $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}$, such that $\boldsymbol{\Lambda}$ as before, and make use of the block matrix inversion formula,

$$\boldsymbol{C}^{-1} = \begin{bmatrix} \boldsymbol{M} & \mathbf{1} \\ \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{M}^{-1} + \boldsymbol{M}^{-1}\mathbf{1}S_M^{-1}\mathbf{1}^T\boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\mathbf{1}S_M^{-1} \\ \\ -S_M^{-1}\mathbf{1}^T\boldsymbol{M}^{-1} & S_M^{-1} \end{bmatrix} \tag{10}$$

where $S_M = -\mathbf{1}\boldsymbol{M}^{-1}\mathbf{1}$ is the Schur complement of $\boldsymbol{M}$. We must therefore find a computationally efficient manner in which to evaluate α_i and $\boldsymbol{C}_{ii}^{-1}$ in order to evaluate the leave-one-out error, beginning with expressions for the individual elements of $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$,

$$\nu_i = \sum_{j=1}^{\ell} \frac{v_{ij}\xi_j}{\lambda_j + \mu} \qquad \text{and} \qquad \eta_i = \sum_{j=1}^{\ell} \frac{v_{ij}\zeta_j}{\lambda_j + \mu}$$

where

$$\xi_j = \sum_{k=1}^{\ell} v_{kj} y_k \qquad \text{and} \qquad \zeta_j = \sum_{k=1}^{\ell} v_{kj}.$$

Note that, provided we have pre-computed $\boldsymbol{\xi} = [\xi_i]_{i=1}^{\ell}$ and $\boldsymbol{\zeta} = [\zeta_i]_{i=1}^{\ell}$, which do not depend on μ, then $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$, and hence $\boldsymbol{\alpha}$ may be updated at a cost of only $\mathcal{O}(\ell^2)$ operations following a change in the value of the regularisation parameter, μ. Using the block matrix inversion formula (10), following a somewhat lengthy algebraic manipulation, the diagonal elements of $\boldsymbol{C}^{-1}$ can be computed as

$$C_{ii}^{-1} = \frac{\eta_i^2}{S_M} + \sum_{j=1}^{\ell} \frac{v_{ij}^2}{\lambda_j + \mu}, \qquad \text{noting that} \qquad S_M = -\sum_{j=1}^{\ell} \eta_j.$$

An individual element of the principal diagonal of $\boldsymbol{C}^{-1}$ can therefore be recomputed with a computational complexity of $\mathcal{O}(\ell)$ operations, in response to a change in the value of the regularisation parameter. This permits the leave-one-out cross-valudation of a least-squares support vector machine with a computational complexity of only $\mathcal{O}(\ell^2)$, provided that it has already been placed in canonical form. This novel formulation produces results comparable with those obtained using the optimally regularized kernel Fisher discriminant.

6 Summary

Model selection, the optimal choice of the values for a small number of regularisation and kernel parameters, is the key step in maximising generalisation performance using kernel learning methods. Conventional k-fold and leave-one-out cross-validation strategies provide computationally expensive, but highly effective solutions. In this paper, we extend an existing analytic method for efficient evaluation of the leave-one-out cross-validation error of a kernel Fisher discriminant classifier [2, 20], based on methods from classical linear regression [17]. By performing the kernel discriminant analysis in *canonical form*, the leave-one-out error can be re-evaluated in response to a change in the value of the regularisation parameter with a computational complexity of only $\mathcal{O}(\ell^2)$, instead of the $\mathcal{O}(\ell^3)$ operations required by existing methods based on the "hat" matrix, or the $\mathcal{O}(\ell^4)$ of a naïve direct implementation. The training algorithms for the kernel Fisher discriminant classifier in the original and canonical forms both exhibit computational complexities of $\mathcal{O}(\ell^3)$. The cost of obtaining a KFD classifier with a fully optimised regularisation parameter therefore remains $\mathcal{O}(\ell^3)$, however in canonical form, the cost of optimising the regularisation parameter becomes negligible. This makes the ORKFD attractive for small to medium scale applications (currently anything less than a few thousand training patterns) as an important element of the model selection process is solved essentially “for free”. An experimental evaluation over thirteen benchmark datasets shows that the generalisation performance of the ORKFD is competitive with that of conventional KFD classifiers with a 10-fold cross-validation model selection strategy.

Acknowledgements

We thank the anonymous reviewers for their helpful and constructive comments. This work was supported by Biological and Biotechnology Research Council (BBSRC) of the United Kingdom (grant number 83/EGM16128) and the Royal Society (grant number RSRG-22270).

References

[1] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing*, volume IX, pages 41–48. IEEE Press, New York, 1999.

[2] G. C. Cawley and N. L. C. Talbot. Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition*, 36(11):2585–2592, November 2003.

[3] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[4] C. Saunders, A. Gammermann, and V. Vovk. Ridge regression in dual variables. In J. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-1998)*. Morgan Kaufmann, 1998.

[5] I. T. Jolliffe. *Principal component analysis.* Springer Series in Statistics. Springer-Verlag, New York, second edition, 2002.

[6] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, USA, 1999.

[7] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Conference on Learning Theory (COLT)*, pages 144–152, Pittsburgh, PA, USA, 1992. ACM Press.

[9] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[10] B. Schölkopf and A. J. Smola. *Learning with kernels : Support vector machines, regularization, optimisation and beyond.* Adaptive computation and machine learning. MIT Press, 2002.

[11] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, 2004.

[12] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[13] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, and J. Vandewalle. *Least-squares support vector machines.* World Scientific, Singapore, 2002.

[14] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, Cambridge, U.K., 2000.

[15] J. Xu, X. Zhang, and Y. Li. Kernel MSE algorithm: A unified framework for KFD, LS-SVM and KRR. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1486–1491, Washington, DC, July 2001.

[16] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, June 1999.

[17] S. Weisberg. *Applied linear regression.* John Wiley and Sons, New York, second edition, 1985.

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

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning - data mining, inference and prediction.* Springer Series in Statistics. Springer,

2001.

[20] G. C. Cawley and N. L. C. Talbot. Efficient cross-validation of kernel Fisher discriminant classifiers. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN-2003)*, pages 241–246, Bruges, Belgium, April 23–25 2003.

[21] O. Chapelle, N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[22] G. C. Cawley and N. L. C. Talbot. Improved sparse least-squares support vector machines. *Neurocomputing*, 48:1025–1031, October 2002.

[23] S. Fine and K. Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, December 2001.

[24] G. C. Cawley and N. L. C. Talbot. A greedy training algorithm for sparse least-squares support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-2002)*, volume 2415 of *Lecture Notes in Computer Science (LNCS)*, pages 681–686, Madrid, Spain, August 27–30 2002. Springer.

[25] L. Wang, L. Bo, and L. Jiao. Sparse kernel ridge regression using backward deletion. In *Proceedings of Pacific Rim International Conference on Artificial Intelligence*, volume 4099 of *Lecture Notes in Computer Science*, pages 365–374. Springer, 2006.

[26] W. W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, June 1989.

[27] S. D. Bay. The UCI KDD archive [http://kdd.ics.uci.edu/]. University of California, Department of Information and Computer Science, Irvine, CA,

1999.

[28] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[29] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. J. Smola, and K.-R. Müller. Invariance feature extraction and classification in feature spaces. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 526–532. MIT Press, 2000.

[30] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[31] A. J. Grove and D. Schuurmans. Boosting in the limit : maximizing the margin of learned features. In *Proceedings of the fifteenth National Conference on Artificial Intelligence*, pages 692–699, Madison, Wisconsin, USA, 1998.

[32] V. N. Vapnik. *Statistical Learning Theory.* Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control. Wiley, New York, 1998.

[33] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), 1990.

[34] R. M. Rifkin. *Everything old is new again : A fresh look at historical approaches to machine learning.* PhD thesis, Massachusetts Institute of Technology, 2002.

[35] K. Pelckmans, J. A. K. Suykens, and B. De Moor. Additive regularization trade-off: Fusion of training and validation levels in kernel methods. *Machine Learning*, 62(3):217–252, March 2006.

[36] G. C. Cawley and N. L. C. Talbot. Preventing over-fitting in model selection via Bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research* (accepted), 2007.

[37] R. M. Rifkin and R. Lippert. Notes on regularized least-squares. unpublished research note, 2006.

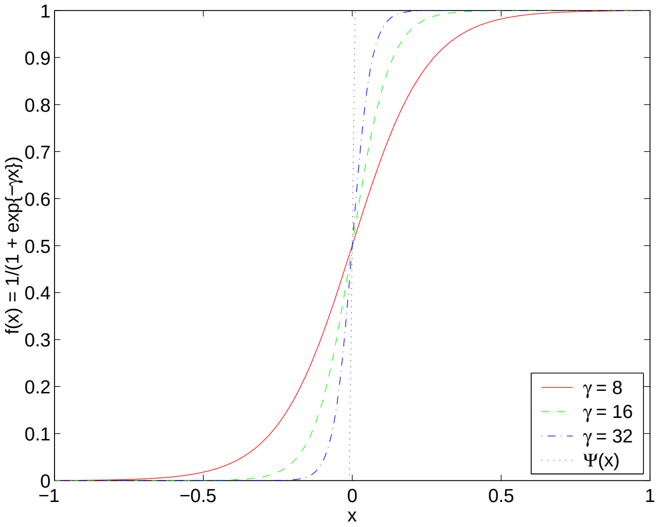
Figure Captions

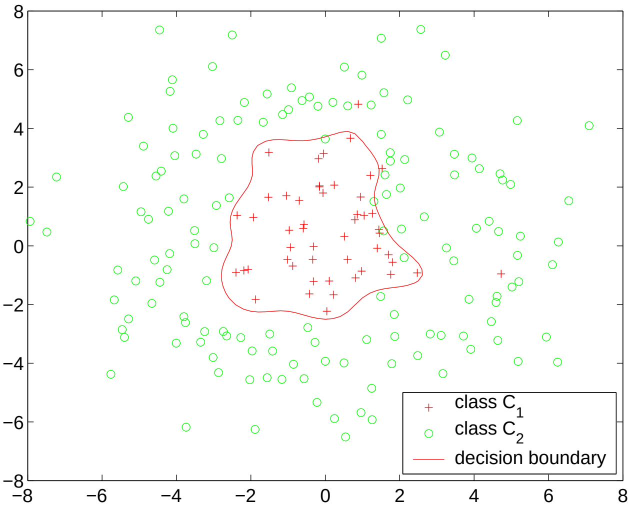
1 Approximation of the Heaviside unit step function, $\Psi(x)$, by the scaled logistic function, $f(x) = 1/(1 + \exp\{-\gamma x\})$.

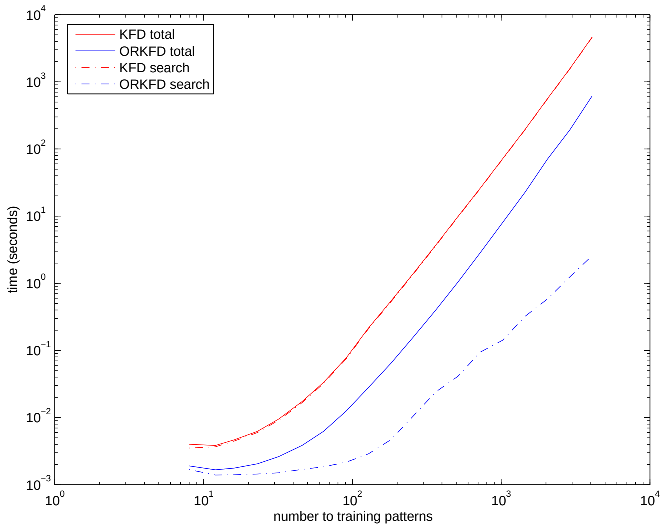
2 Example of the output of an optimally regularised kernel Fisher discriminant classifier on the synthetic torus dataset (see section 6 for details).

3 Graph of run-time as a function of the number of training patterns for leave-one-out cross-validation of kernel Fisher discriminant classifiers via direct and fast approximate methods (mean of 20 trials).

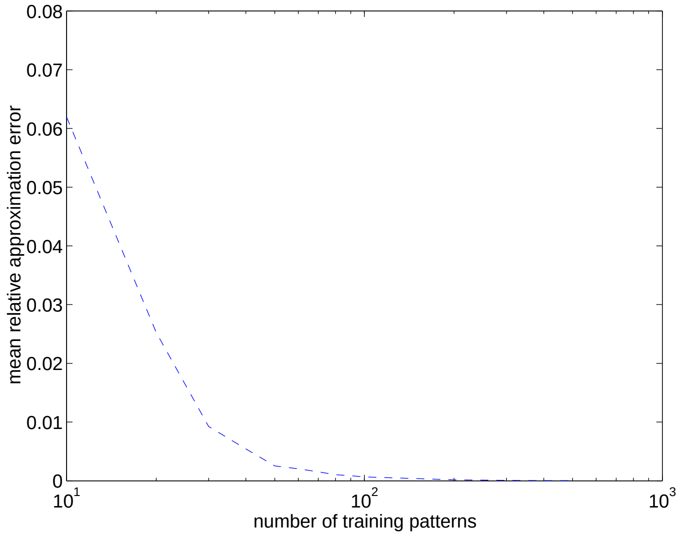
4 Graph of the mean relative approximation error as a function of the number of training patterns for the proposed fast approximate leave-one-out cross-validation method (mean of 20 trials).


f(x) = 1/(1 + exp{−γx})
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−1
−0.5
0
0.5
1
x
γ = 8
γ = 16
γ = 32
Ψ(x)


8
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
8
class C_1
class C_2
decision boundary



KFD total
ORKFD total
KFD search
ORKFD search
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
time (seconds)
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
number to training patterns


0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
mean relative approximation error
10^1
10^2
10^3
number of training patterns

List of Tables

Benchmark	**ORKFD**	**RBF**	**AB**	**ABL**	**ABQ**	**AB$_R$**	**SVM**	**KFD**
Banana	**10.51±0.42**	10.76±0.42	12.26±0.67	*10.73±0.43*	10.90±0.46	10.85±0.42	11.53±0.66	10.75±0.45
Breast cancer	27.00±4.68	27.64±4.71	30.36±4.73	26.79±6.08	*25.91±4.61*	26.51±4.47	26.04±4.74	**24.77±4.63**
Diabetis	*23.22±1.76*	24.29±1.88	26.47±2.29	24.11±1.90	25.39±2.20	23.79±1.80	23.53±1.73	**23.21±1.63**
German	**23.62±2.16**	24.71±2.38	27.45±2.50	24.79±2.22	25.25±2.14	24.34±2.08	**23.62±2.07**	23.71±2.20
Heart	**15.78±3.30**	17.55±3.25	20.29±3.44	17.49±3.53	17.17±3.44	16.47±3.51	*15.95±3.26*	16.14±3.39
Image	4.03±0.58	3.32±0.65	*2.73±0.66*	2.76±0.61	**2.67±0.63**	**2.67±0.61**	2.96±0.60	4.76±0.58
Ringnorm	**1.47±0.09**	1.70±0.21	1.93±0.24	2.24±0.46	1.86±0.22	1.58±0.12	1.66±0.12	*1.49±0.12*
Solar flare	34.09±1.62	34.37±1.95	35.70±1.79	34.74±2.00	36.22±1.80	34.20±2.18	**32.43±1.82**	*33.16±1.72*
Splice	10.80±0.70	*9.95±0.78*	10.14±0.51	10.22±1.59	10.11±0.52	**9.50±0.65**	10.88±0.66	10.51±0.64
Thyroid	4.88±1.97	4.52±2.12	*4.40±2.18*	4.59±2.22	4.35±2.18	4.55±2.19	4.80±2.19	**4.20±2.07**
Titanic	*22.53±1.05*	23.26±1.34	22.58±1.18	23.98±4.38	22.71±1.05	22.64±1.20	**22.42±1.02**	23.25±2.05
Twonorm	*2.64±0.19*	2.85±0.28	3.03±0.28	3.17±0.43	2.97±0.26	2.70±0.24	2.96±0.23	**2.61±0.23**
Waveform	**9.55±0.33**	10.66±1.08	10.84±0.58	10.53±1.02	10.07±0.51	*9.79±0.81*	9.88±0.43	9.86±0.44