# Parsimonious Kernel Fisher Discrimination

Kitsuchart Pasupa[1], Robert F. Harrison[1], and Peter Willett[2]

[1] Department of Automatic Control & Systems Engineering,
[2] Department of Information Studies,
The University of Sheffield, UK

**Abstract.** *By applying recent results in optimization transfer, a new algorithm for kernel Fisher Discriminant Analysis is provided that makes use of a non-smooth penalty on the coefficients to provide a parsimonious solution. The algorithm is simple, easily programmed and is shown to perform as well as or better than a number of leading machine learning algorithms on a substantial benchmark. It is then applied to a set of extreme small-sample-size problems in virtual screening where it is found to be less accurate than a currently leading approach but is still comparable in a number of cases.*

## 1 Introduction

Fisher discriminant analysis has a central role in pattern recognition. It seeks a linear projection that maximizes the separation between data belonging to two classes while minimizing the separation between those of the same class. Its properties are well-documented and under certain circumstances prove optimal [1]. However, the linearity of the approach is frequently insufficient to allow the required level of performance in practical applications. While explicit expansion of data in basis functions can resolve this for problems of low dimension, the combinatorial increase in the number of coefficients to be estimated may make this impractical. Recent focus on kernel machines in the machine learning community seeks to address this problem via the so-called "kernel trick" [2] and a number solutions have been provided e.g. [3] that can be thought of generically as kernel Fisher discriminant analysis (kFDA). While kernels lend the required degree of flexibility to the discrimination task, they bring their own challenges, the foremost being a potential to overspecialize to the sample data (over-fitting) and a computational complexity dominated by sample size which, in some problems, may be large. Complexity control is therefore essential for a good outcome yet it has not been widely explored in the context of kFDA. In [3] complexity is controlled through explicit regularization – placing an appropriate penalty on the coefficients of the estimator, while [4] exploits the connection between FDA and an associated least-squares (LS) problem where an orthogonalization technique is used for forward selection. In benchmarks, the latter technique is seen to be competitive with a number of leading machine-learning classifiers including kFDA while providing more parsimonious estimators.

The motivation for our work lies in the field of chemoinformatics, in particular in virtual screening. Virtual screening (VS) describes a set of computational methods that provide a fast and cheap alternative to biological screening which involves the selection, synthesis and testing of molecules to ascertain their biological activity in a particular domain, e.g. pain relief, reduction of inflammation. This is important because reducing the cost and crucially time in the early stages of compound development can have a disproportionate benefit in profitability in a cycle that has a short patent lifetime. The aim of VS is to score, rank and/or filter a set of chemical structures to ensure that those molecules with the highest likelihood of activity are assayed first in a "lead discovery programme"[5].

The use of machine learning methods for VS has been widely studied. Techniques such as artificial neural networks and support vector machines in addition to more conventional approaches such as similarity matching and nearest neighbour analysis have all been explored while little attention has been paid to the use of FDA or its variants. An important recent development is the technique of binary kernel discrimination (BKD) which produces scores based on the estimated likelihood ratio of active to inactive compounds that are then ranked. The likelihoods are estimated through a Parzen Windows approach using the binomial distribution function (to accommodate binary descriptor or "fingerprint" vectors representing the presence, or not, of certain substructural arrangements of atoms) in place of the usual Gaussian choice [6]. This choice of kernel function uses Hamming distance but by substituting the Jaccard/Tanimoto distance instead, additional active compounds can be retrieved [7]. We will use results from BKD and its variant for comparison.

Virtual screening suffers strongly from the so-called small-sample-size problem where the number of covariates is comparable to or exceeds the number of samples. Typically a task in VS comprises a sample of size $\mathcal{O}\left(10^2\right)$ of known descriptors but with fingerprints of dimension $\mathcal{O}\left(10^3\right)$. Clearly some form of complexity control is therefore necessary.

In this paper we again exploit the association of FDA with LS but control complexity by penalizing the likelihood function. It is well-known that penalty functions that induce sparsity lead to non-smooth formulations and these are traditionally solved via mathematical programming techniques [8]. In a departure, we apply a minorize-maximize (MM) technique to overcome this technical problem leading to a very simple iterative algorithm that is guaranteed to converge to the (penalized) maximum likelihood solution. In [9] a general MM framework is presented for variable selection via penalized maximum likelihood but there only a small LS problem in conjunction with the SCAD penalty is examined.

The paper is organized as follows. Section 2 briefly states the well-known link between FDA and LS [1], presents the kernel-based formulation and motivates the use of penalized maximum likelihood. The following section introduces the MM principle and sketches a derivation of the iterative algorithm. Section 4 presents a performance comparison with other leading machine learning methods on a well-studied set of benchmarks and the results of applying the proposed method to VS are given in section 5.

## 2 Fisher Discriminant Analysis and its Variants

The relationship between FDA and LS is well known [1]. Consider the matrix of $m$-dimensional sample vectors $\mathrm{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]$ comprising two groups, $\mathcal{G}_i$, of size, $n_i$, $i = 1, 2$ represented by the partition, $\begin{bmatrix}\mathrm{U}_1 & \mathrm{U}_2\end{bmatrix}$. Membership of $\mathcal{G}_1$ is denoted by $\hat{y} = +n/n_1$ and of $\mathcal{G}_2$ by $\hat{y} = -n/n_2$ then it is straightforward to verify that the solution, $\boldsymbol{\omega} = \begin{bmatrix}b & \boldsymbol{w}\end{bmatrix}^{\mathsf{T}}$, to the following LS problem lies in the same direction as the solution for the Fisher discriminant [1].

$$\arg\min_{(b,\boldsymbol{w})} \left\| \begin{bmatrix} \frac{n}{n_1}\mathbf{1}_{n_1} \\ -\frac{n}{n_2}\mathbf{1}_{n_2} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{n_1} & \mathrm{U}_1 \\ \mathbf{1}_{n_2} & \mathrm{U}_2 \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix} \right\|_2^2 \tag{1}$$

where $\mathbf{1}_p$ denotes a $p$-vector of ones.

To accommodate more complex discriminants, data can be mapped into a new feature space, $\mathcal{F}$, via some function, $\phi$, say. However, vectors in $\mathcal{F}$ will typically be of very high dimension precluding any practical manipulation. The "kernel trick" recognizes that the coefficients, $\boldsymbol{w}$, in the linear model implicit in (1) can themselves be written as a linear combination of the mapped data, leading to a formulation entirely based on inner products that can be computed through the agency of a suitable kernel. These ideas have been explored thoroughly elsewhere e.g. so we omit further exposition and simply present the kernelized version of the LS problem (see e.g. [4] for details).

Let $K$ denote the Gram matrix associated with the kernel, $k(.,.)$, i.e. $k_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j)$, $i, j = 1, 2, \ldots, N$ then the solution, $\boldsymbol{\omega} = \begin{bmatrix}b & \boldsymbol{\alpha}\end{bmatrix}^{\mathsf{T}}$, to the following LS problem provides the coefficients of a linear discriminant in the feature space associated with $k(.,.)$ hence non-linear discrimination in the data space.

$$\arg\min_{(b,\boldsymbol{\alpha})} \left\| \begin{bmatrix} \frac{n}{n_1}\mathbf{1}_{n_1} \\ -\frac{n}{n_2}\mathbf{1}_{n_2} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{n_1} & \mathrm{K}_1 \\ \mathbf{1}_{n_2} & \mathrm{K}_2 \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2^2 \tag{2}$$

It is common to introduce a quadratic penalty into LS regression and this can be interpreted in the Bayesian framework as placing a Gaussian prior on the values of the coefficients. The quadratic penalty improves numerical condition when data are strongly correlated, militates against over-fitting and also suggests a method for selecting variables – those with relatively small coefficient magnitudes can be discarded. Essentially, large deviations from zero are strongly penalized while small values are only very lightly affected. Instead of using a Gaussian penalty, a prior distribution with a sharp peak has the effect of penalizing non-zero coefficients much more strongly so that the pay-off for setting small coefficients exactly to zero is relatively much greater. A log-likelihood penalty of the form $\|\boldsymbol{\omega}\|_q^q$ $0 < q \leq 1$ has precisely this property. The "sparsity-inducing" property of this penalty is well-known [2]. Introducing this penalty leads to a difficulty in gradient-based optimization owing to its discontinuous first derivative. Mathematical programming is the usually adopted to address this but here we exploit the MM principle to provide a simple, iterative algorithm.

## 3 Algorithm Development via the MM Principle

The MM principle seeks to replace a difficult optimization problem, in our case, non-smooth, with a simpler (smooth) one having the same solution. In the case of maximization, the idea is to find a non-unique *surrogate* function that *minorizes* the objective function of interest and then to maximize this. Here we are able to replace the non-smooth element of the likelihood function with a quadratic function and then iterate toward the solution.

Let $\boldsymbol{\omega}(n)$ denote the $n^{\text{th}}$ step in an iterative procedure, then a function, $S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right)$, is said to minorize the function, $\ell\left(\boldsymbol{\omega}\right)$, if it is everywhere less than $\ell$ and is tangent to it at $\boldsymbol{\omega}(n)$ e.g. [10]. Such a function that minorizes a concave objective function can itself be maximized, often analytically, and this fact can be exploited. The minorizing function, $S(.,.)$, acts as a surrogate for the original objective function. The *ascent* property e.g. [10] guarantees that the value of $\ell\left(\boldsymbol{\omega}\right)$ never decreases. The iteration will therefore converge to the global maximum for a concave objective function. Minorization is closed under the operations of addition and multiplication.

We outline the derivation of a very simple, Newton-like algorithm for the penalized maximum likelihood estimation of the kFDA coefficients, c.f. [9]. The log-likelihood function, $\ell\left(\boldsymbol{\omega}\right)$, is written as the sum of two functions, $\ell_e\left(\boldsymbol{\omega}\right) = -\frac{1}{2}\|\hat{\boldsymbol{y}} - \tilde{\mathrm{K}}\boldsymbol{\omega}\|_2^2$, and $\ell_p\left(\boldsymbol{\omega}\right) = -\rho N\|\boldsymbol{\omega}\|_q^q$, where $\hat{\boldsymbol{y}} = \begin{bmatrix} \frac{n}{n_1}\mathbf{1}_{n_1} \\ -\frac{n}{n_2}\mathbf{1}_{n_2} \end{bmatrix} \in \mathbb{R}^N$ and $\tilde{\mathrm{K}} = \begin{bmatrix}\mathbf{1}_N \; K\end{bmatrix} \in \mathbb{R}^{N\times(N+1)}$, giving:

$$\ell\left(\boldsymbol{\omega}\right) = -\frac{1}{2}\|\hat{\boldsymbol{y}} - \tilde{\mathrm{K}}\boldsymbol{\omega}\|_2^2 - \rho N\|\boldsymbol{\omega}\|_q^q \tag{3}$$

It is clear that in the case of interest, $0 < q \leq 1$, no closed-form solution exists for the maximization of (3), however, by exploiting the fact that $-|\omega|^q$ is convex on $\mathbb{R}_+$ and $-|\omega|^q = -\left(\omega^2\right)^{\frac{q}{2}}$ it can be shown that $\ell_p\left(\boldsymbol{\omega}\right)$ is minorized at every point, $\boldsymbol{\omega}(n)$, by a quadratic function thus:

$$\begin{aligned}\ell_p\left(\boldsymbol{\omega}\right) = -\rho N\|\boldsymbol{\omega}\|_q^q &\geq -\frac{\rho}{2}N\sum_{i=1}^{i=d}\left(\frac{q\omega_i^2}{|\omega_i(n)|^{2-q}} + (2-q)|\omega_i(n)|^q\right) \\ &= -\frac{\rho}{2}N\left(\boldsymbol{q}\boldsymbol{\omega}^{\mathsf{T}}\mathrm{B}\left(\boldsymbol{\omega}(n)\right)\boldsymbol{\omega} + (2-q)\|\boldsymbol{\omega}(n)\|_q^q\right)\end{aligned} \tag{4}$$

with $\mathrm{B}\left(\boldsymbol{\omega}(n)\right) = \mathrm{diag}\left\{|\omega_i(n)|^{q-2}\right\}$. The result arises from the relationship $g(x) \geq g(y) + dg(y)(x-y)\forall x, y$ e.g. [10] and is ascribed to [11]. The function, $\ell\left(\boldsymbol{\omega}\right)$, in equation (3) is therefore minorized when the RHS is replaced by the upper bound given in (4) giving a quadratic surrogate having the ascent property:

$$S\left(\boldsymbol{\omega}|\boldsymbol{\omega}(n)\right) = \boldsymbol{\omega}^{\mathsf{T}}\tilde{\mathrm{K}}^{\mathsf{T}}\hat{\boldsymbol{y}} - \frac{1}{2}\boldsymbol{\omega}^{\mathsf{T}}\left(\tilde{\mathrm{K}}^{\mathsf{T}}\tilde{\mathrm{K}} + \rho N q\mathrm{B}\left(\boldsymbol{\omega}(n)\right)\right)\boldsymbol{\omega}$$

(omitting constant terms), which can be maximized analytically w.r.t $\boldsymbol{\omega}$. Identifying $\boldsymbol{\omega}(n+1)$ with $\boldsymbol{\omega}$ and assuming $\boldsymbol{\omega}(0) \neq \mathbf{0}$ gives the following iteration

$$\boldsymbol{\omega}(n+1) = \left(\tilde{\mathrm{K}}^{\mathsf{T}}\tilde{\mathrm{K}} + \rho N q\mathrm{B}\left(\boldsymbol{\omega}(n)\right)\right)^{-1}\tilde{\mathrm{K}}^{\mathsf{T}}\hat{\boldsymbol{y}} \tag{5}$$

A potential problem arises when the elements of $\boldsymbol{\omega}(n)$ approach zero, as expected when a sparse solution arises and $S(\boldsymbol{\omega}|\boldsymbol{\omega}(n))$ is no longer defined. Hunter and Li [9] deal with this formally but here we avoid the difficulty by re-writing $\mathrm{B}(\boldsymbol{\omega}(n)) = \Psi_n^{-2}$ with $\Psi_n = \mathrm{diag}\left\{|\omega_i(n)|^{\frac{2-q}{2}}\right\}$ [12] leading to

$$\boldsymbol{\omega}(n+1) = \Psi_n \left(\Psi_n \tilde{\mathrm{K}}^{\mathsf{T}} \tilde{\mathrm{K}} \Psi_n + \rho N q \mathrm{I}_N\right)^{-1} \Psi_n \tilde{\mathrm{K}}^{\mathsf{T}} \hat{\boldsymbol{y}}; \boldsymbol{\omega}(0) \neq \mathbf{0} \tag{6}$$

Convergence is declared when the relative change in the norm of the coefficient vectors is less than some threshold, $\epsilon$, and a coefficient is deemed to equal zero if its magnitude, relative to the largest, is less than $\eta$. We denote the resulting classifiers $\mathrm{kFDA}_q$.

## 4 Performance Comparison with Previous Methods

To evaluate the performance of $\mathrm{kFDA}_q$ extensive experimentation has been carried out on 13 datasets (http://ida.first.fraunhofer.de/projects/bench/) that have been used to benchmark numerous machine learning techniques [3, 4, 13]. The methodology outlined in [4] was followed to allow direct comparisons with [3, 4, 13]. We examine two situations: classifiers are selected based on minimum misclassification rate (MCR) and on number of retained samples (NS) from amongst the first five realizations. Each is then applied to all 100 test partitions.

Table 1 shows percentage mean MCR and number of retained samples (below) calculated for the test sets for each of the 13 domains. We report $\mathrm{kFDA}_q$ for $q \in \{1, 0.5\}$, the better of the two methods proposed in [4] referred to as $\mathrm{kFDA}_{\mathrm{OLS}}$, and the current best results in [3, 13]. From Table 1, the $\mathrm{kFDA}_1$ classifier (selected on MCR) is more accurate than $\mathrm{kFDA}_{\mathrm{OLS}}$ and the best other reported technique across all 13 domains. However, its sparseness is relatively poor in 10 out of 13 cases. Choosing the $\mathrm{kFDA}_1$ classifier for sparseness gives best MCR in 7 out of 13 cases and outperforms $\mathrm{kFDA}_{\mathrm{OLS}}$ in 9 out of 13. This gives comparable sparseness to $\mathrm{kFDA}_{\mathrm{OLS}}$ in many cases but is much worse in a few.

To encourage further sparseness, $q$ is reduced to 0.5. Selecting on MCR, $\mathrm{kFDA}_{0.5}$ now exhibits highest accuracy in 8 out of 13 cases while achieving comparable sparseness with $\mathrm{kFDA}_{\mathrm{OLS}}$ (excepting "Splice"). In the other five cases performance is comparable with $\mathrm{kFDA}_{\mathrm{OLS}}$ but with greater levels of sparsity. Selecting for sparsity even simpler models are frequently found with no substantial loss of performance. Performance on "Splice" is now comparable, for instance.

While differences are not great, it is fair to say that $\mathrm{kFDA}_q$ offers convincingly competitive performance across a range of classification tasks.

## 5 Application to Virtual Screening

Here 11 different activity classes – domains in which molecules have been assayed for activity – from the MDL Drug Data Report (MDDR) database are

**Table 1.** Comparison of mean misclassification rate and sparsity for the proposed algorithm, kFDA$_q$, kFDA$_{\mathrm{OLS}}$ (★) [4], and the best algorithm from[3, 13]: Support Vector Machine (▲) , Regularized AdaBoost (▼), conventional kFDA (■), Sparse kFDA (□) and Sparse kFDA with Linear Loss (◊). Bold – best, italic – sample size.

| Database | Published | kFDA$_{\mathrm{OLS}}$ | kFDA$_1$ | | kFDA$_{0.5}$ | |
|---|---|---|---|---|---|---|
| | Best (%) | (%) | MCR (%) | NS (%) | MCR (%) | NS (%) |
| Banana | 10.6±0.4 | 10.7±0.5 | **9.74±0.00** | 13.05±0.15 | 10.48±0.12 | 11.60±0.12 |
| *400* | 8.00 ◊ | 7.25 | 14.25 | 5.75 | 5.75 | **2.25** |
| B. Cancer | 25.2±4.4 | 25.3±4.1 | 21.21±3.71 | 25.55±4.05 | **20.81±3.73** | 25.35±4.02 |
| *200* | 12.00 □ | 3.50 | 11.00 | 2.00 | 4.00 | **1.00** |
| Diabetes | 23.1±1.8 | 23.1±1.8 | **22.93±1.68** | 24.04±1.71 | 23.96±1.69 | 25.50±1.89 |
| *468* | 2.14 ★ | 2.14 | 2.14 | 1.28 | 0.85 | **0.43** |
| German | 23.6±2.3 | 24.0±2.1 | **19.38±1.87** | 23.43±2.04 | 24.55±2.13 | 24.08±2.00 |
| *700* | 2.00 □ | 1.14 | 11.86 | 4.00 | 0.57 | **0.43** |
| Heart | 15.8±3.4 | 15.8±3.4 | **14.63±3.36** | **14.63±3.36** | 15.46±3.22 | 15.46±3.22 |
| *170* | 1.76 ★ | 1.76 | 4.71 | 4.71 | **1.76** | **1.76** |
| Image | 2.7±0.6 | 2.8±0.6 | **1.10±0.56** | 2.19±0.58 | 1.95±0.44 | 1.95±0.44 |
| *1300* | 100.00 ▼ | 21.54 | 46.15 | 22.46 | **16.69** | **16.69** |
| Ringnorm | 1.5±0.1 | 1.6±0.1 | **1.41±0.03** | 1.52±0.03 | 1.70±0.04 | 1.70±0.04 |
| *400* | 6.00 ◊ | 1.75 | 9.00 | 3.75 | **1.00** | **1.00** |
| S. Flare | 32.4±1.8 | 33.5±1.6 | 31.90±1.90 | 32.69±1.96 | **31.07±1.88** | 32.60±1.82 |
| *660* | 91.00 ▲ | 1.36 | 56.21 | 44.09 | 4.85 | **2.27** |
| Splice | 9.5±0.7 | 11.7±0.6 | **7.03±0.79** | 7.46±0.80 | 7.32±0.75 | 8.22±0.91 |
| *1000* | 100.00 ▼ | 33.00 | 78.20 | 64.40 | 72.90 | **34.70** |
| Thyroid | 4.2±2.1 | 4.5±2.4 | 2.88±1.59 | 3.72±1.54 | **1.53±1.16** | 3.83±1.77 |
| *140* | 100.00 ■ | 16.43 | 10.00 | 9.29 | 8.57 | **1.43** |
| Titanic | 22.4±1.0 | 22.4±1.0 | **21.10±0.23** | 22.09±0.24 | **21.10±0.23** | 22.72±0.26 |
| *150* | 7.33 ★ | 7.33 | 40.67 | 34.00 | 6.00 | **1.33** |
| Twonorm | 2.6±0.2 | 2.7±0.2 | **2.32±0.00** | **2.32±0.00** | 2.87±0.04 | 2.87±0.04 |
| *400* | 100.00 ■ | 2.50 | 3.75 | 3.75 | **1.25** | **1.25** |
| Waveform | 9.8±0.8 | 10.0±0.4 | **9.46±0.13** | 10.01±0.13 | 10.10±0.15 | 12.23±0.13 |
| *400* | 100.00 ▼ | 3.50 | 7.75 | 5.00 | 1.50 | **0.75** |

used [14]. The MDDR database contains 1,024-dimensional fingerprints representing 102,514 known drugs and biologically relevant molecules collected from patent literature, journals, meetings and congresses. The classes used here were selected to reflect typical drug discovery projects for pharmaceutical companies.

A "binomial" kernel is used, $k_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j) = \lambda^{m-d(\boldsymbol{u}_i,\boldsymbol{u}_j)} (1-\lambda)^{d(\boldsymbol{u}_i,\boldsymbol{u}_j)}$ where $\lambda \in [0.5, 1.0]$ denotes the kernel "width" and $m = 1024$. $d(\boldsymbol{u}_i, \boldsymbol{u}_j)$ is a measure of the degree of dis-similarity between molecules $i$ and $j$. In [7] the Jaccard/Tanimoto (J/T) distance was found to offer substantial gains over the conventional Hamming distance (HD) when used in BKD. Experiments show that this is also the case for kFDA$_q$ so only results using this function are reported. The experiment was run five times with different random data splits. $\lambda$ is identified by five-fold cross validation on the basis of *sum of active rank position*, e.g. if all $n_1$ active compounds are ranked in the first $n_1$ positions, the rank sum is minimal.

In table 2, the mean self-similarity provides a measure of the homogeneity of each of the activity classes and is a useful way to compare design spreads and coverage. It is usual in chemoinformatics applications to report the percentage of the maximum possible number of active compounds ranked in the top 5% of the ranked database. These are shown in Table 2 along with the percentage of

**Table 2.** Comparison of maximum percentage actives retrieved in top 5% of sample.

| Index | Activity Class | Self-Similarity | | BKD | | $kFDA_1$ | $kFDA_{0.5}$ |
|---|---|---|---|---|---|---|---|
| | | Mean | S.D. | HD (%) | J/T (%) | J/T (%) | J/T (%) |
| 1 | 5HT3 Antagonists | 0.351 | 0.116 | 90.19 | **93.88** | 91.32 | 90.61 |
| | *150* | | | | | 76.00 | **57.87** |
| 2 | 5HT1A Agonists | 0.343 | 0.104 | 86.77 | **88.28** | 83.98 | 82.10 |
| | *166* | | | | | 66.87 | **51.21** |
| 3 | 5HTReuptake Inhibitors | 0.345 | 0.122 | 69.47 | **73.62** | 64.89 | 65.08 |
| | *72* | | | | | 81.39 | **71.67** |
| 4 | D2 Antagonists | 0.345 | 0.103 | 74.25 | **77.97** | 70.25 | 68.34 |
| | *80* | | | | | 78.50 | **59.00** |
| 5 | Renin Inhibitors | 0.573 | 0.106 | 98.84 | 99.10 | **99.25** | 99.23 |
| | *226* | | | | | 53.63 | **28.14** |
| 6 | Angiotensin II AT1 Antagonists | 0.403 | 0.101 | 98.77 | 97.43 | **99.27** | 99.22 |
| | *190* | | | | | 55.16 | **30.11** |
| 7 | Thrombin Inhibitors | 0.419 | 0.127 | **94.04** | 94.02 | 93.77 | 92.74 |
| | *162* | | | | | 59.63 | **41.11** |
| 8 | Substance P Antagonists | 0.399 | 0.106 | 91.86 | **93.70** | 90.74 | 90.38 |
| | *250* | | | | | 62.16 | **54.32** |
| 9 | HIV Protease Inhibitors | 0.446 | 0.122 | 94.37 | **94.70** | 92.89 | 93.45 |
| | *150* | | | | | 54.53 | **45.47** |
| 10 | Cyclo-oxygenase Inhibitors | 0.268 | 0.093 | 68.43 | **76.26** | 65.52 | 63.11 |
| | *128* | | | | | 86.25 | **71.09** |
| 11 | Protein Kinase C Inhibitors | 0.323 | 0.142 | 78.92 | **81.23** | 71.16 | 62.95 |
| | *92* | | | | | 84.57 | **39.57** |

retained features (below). Results from $kFDA_q$ and from BKD are presented. It is clear that $BKD_{J/T}$ is the leading contender in eight out of 11 cases but delivers no sparsity, while $BKD_{HD}$ is most accurate in one. As before, we see that $kFDA_1$ is generally more accurate (9/11) than $kFDA_{0.5}$ but is less sparse, as expected. However, $kFDA_q$ only displays best accuracy in two cases but is comparable ($< \pm 3\%$) to $BKD_{J/T}$ in four others. It is worth noting that $kFDA_q$ delivers its best accuracy in the classes that are most homogeneous.

## 6 Conclusion

We have introduced an algorithm for the solution of the kFDA problem through the application of the MM principle. We have demonstrated that it performs as well as or better than a number of leading machine learning algorithms in a substantial benchmark. We have then applied the method to a problem in chemoinformatics but found that performance is generally worse than an important recent development in this field, BKD. However, operationally, a sparse solution may still be of value since many commercial databases contain $\mathcal{O}\left(10^6\right)$ samples and speed of recall can be an issue. Given that, ultimately, both techniques rely on optimally chosen linear combinations of kernel functions, this failure seems puzzling. Qualitatively, there is a significant difference between the benchmark and the molecular data – the fingerprint samples suffer extremely from the small-sample-size problem and this fact may account for the drop in performance. This issue is to be addressed in the future.

**Acknowledgments.** Many thanks to thank Jerôme Hert for providing the self-similarity information and David Wood for preparing the MDDR data.

## References

1. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
2. B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.
3. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh Coefficients in kernel feature spaces. *IEEE T. Pattern Anal.*, 25, 2003.
4. S. A. Billings and K. L. Lee. Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks*, 15:263–270, 2002.
5. A. Leach and V. Gillet. *An Introduction to Chemoinformatics*. Kluwer, 2003.
6. G. Harper, J. Bradshaw, J. C. Gittins, D. Green, and A. R. Leach. Prediction of biological activity for high-throughput screening using binary kernel discrimination. *J. Chem. Inf. Comp. Sci.*, 41:1295–1300, 2001.
7. B. Chen, R. F. Harrison, K. Pasupa, P. Willett, D. J. Wilton, D. J. Wood, and X. Q. Lewell. Virtual screening using binary kernel discrimination: Effect of noisy training data and the optimization of performance. *J. Chem. Inf. Mod.*, 46:478–486, 2006.
8. K. C. Kiwiel. An exact penalty function algorithm for non-smooth convex constrained minimization problems. *IMA J. Numer. Anal.*, 5:111–119, 1985.
9. D. R. Hunter and R. Li. Variable selection using MM algorithms. *Ann. Stat.*, 33: 1617–1642, 2005.
10. K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *J. Comput. Graph. Stat.*, 9:1–59, 2000.
11. R. Dutter and P. J. Huber. Numerical methods for the nonlinear robust regression problem. *J. Stat. Comput. Sim.*, 13:79–113, 1981.
12. B. Krishnapuram, L. Carin, M. A. Figueiredo, and A. J. Hartemink. Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE T. Pattern Anal.*, 27:957–968, 2005.
13. G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Mach. Learn.*, 42:287–320, 2001.
14. MDL Information Systems Inc. *The MDL Drug Data Report Database*. http://www.mdli.com