A Fast Iterative Algorithm for Fisher Discriminant using Heterogeneous Kernels

Glenn Fung GLENN.FUNG@SIEMENS.COM
Murat Dundar MURAT.DUNDAR@SIEMENS.COM
Jinbo Bi JINBO.BI@SIEMENS.COM
Bharat Rao BHARAT.RAO@SIEMENS.COM
Computer Aided Diagnosis & Therapy Solutions, Siemens Medical Solutions, 51 Valley Stream Parkway, Malvern PA 19355

Linear Fisher Discriminant, Heterogeneous Kernels, Mathematical Programming, Binary Classification.

Abstract

We propose a fast iterative classification algorithm for Kernel Fisher Discriminant (KFD) using heterogeneous kernel models. In contrast with the standard KFD that requires the user to predefine a kernel function, we incorporate the task of choosing an appropriate kernel into the optimization problem to be solved. The choice of kernel is defined as a linear combination of kernels belonging to a potentially large family of different positive semidefinite kernels. The complexity of our algorithm does not increase significantly with respect to the number of kernels on the kernel family. Experiments on several benchmark datasets demonstrate that generalization performance of the proposed algorithm is not significantly different from that achieved by the standard KFD in which the kernel parameters have been tuned using cross validation. We also present results on a real-life colon cancer dataset that demonstrate the efficiency of the proposed method.

1. Introduction

In recent years, kernel based methods have been proved to be an excellent choice to solve classification problems. It is well known that the use of an appropriate nonlinear kernel mapping is a critical issue when nonlinear hyperplane-based methods such as Kernel Fisher Discriminant (KFD) are used for classification. Typically, kernels are chosen by predefining a kernel model (Gaussian, polynomial, etc.) and adjusting the kernel parameters by means of a tuning procedure. The selection is based on the classification performance on a subset of the training data that is commonly referred to as the validation set. This kernel selection procedure can be computationally very expensive and is particularly prohibitive when the dataset is large; furthermore, there is no warranty that the predefined kernel model is an optimal choice for the classification problem. In recent years, several authors (Hamers et al., 2003; Lanckriet et al., 2003; Bennet et al., 2002; Bach et al., 2004) have proposed the use of a linear combination of kernels formed by a family of different kernel functions and parameters; this transforms the problem of choosing a kernel model into one of finding an "optimal" linear combination of the members of the kernel family. Using this approach there is no need to predefine a kernel; instead, a final kernel is constructed according to the specific classification problem to be solved without sacrificing capacity control. By combining kernels, we make the hypothesis space larger (potentially, but not always), but with appropriate regularization, we improve prediction accuracy which is the ultimate goal for classification.

The drawback of using a linear combination of kernels is that it leads to considerable more complex optimization problems . We propose a fast iterative algorithm that transforms the resulting optimization problem into several relatively computationally less expensive strongly convex optimization problems.

At each iteration, our algorithm only requires to solve a simple system of linear equations and a relatively small quadratic programming problem with nonnegativity constraints, which makes the proposed algorithm easy to implement. In contrast with some of the

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the authors.

previous work, the complexity of our algorithm does *not* depend directly on the number of kernels in the kernel family.

We now outline the contents of the paper. In Section 2, we formulate the linear classification problem as a Linear Fisher Discriminant (LFD) problem. In section 3, using the result (Mika et al., 2000), we show how the classical Fisher discriminant problem can be reformulated as a convex quadratic optimization problem. Using this equivalent mathematical programming LFD formulation and using mathematical programming duality theory, we proposed a kernel Fisher discriminant formulation similar to the one proposed in (Mika et al., 1999), our formulation. Then we introduce a new formulation that incorporates both the KFD problem and the problem of finding an appropriate linear combination of kernels into an quadratic optimization problem with nonnegativity constraints on one set of the variables. In Section 4, we propose an algorithm for solving this optimization problem and we discuss the complexity and convergence of the proposed algorithm. In Section 5, we give some computational results including those for a real life colorectal cancer dataset as well as five other publicly available datasets.

First, we briefly describe our notation. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$, the 2-norm of x will be denoted by $\|x\|$. The 1-norm and ∞-norm will be denoted by $\|\cdot\|_1$ and $\|\cdot\|_\infty$ respectively. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e and the identity matrix of arbitrary order will be denoted by I. For $A \in R^{m\times n}$ and $B \in R^{n\times l}$, the kernel $K(A,B)$ (Vapnik, 2000; Cherkassky & Mulier, 1998; Mangasarian, 2000) is an arbitrary function which maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. In particular, if x and y are column vectors in R^n then, $K(x', y)$ is a real number, $K(x', A')$ is a row vector in R^m and $K(A, A')$ is an $m \times m$ matrix.

2. Linear Fisher's Discriminant (LFD)

We know that the probability of error due to the Bayes classifier is the best we can achieve. A major disadvantage of the Bayes error as a criterion, is that a closed-form analytical expression is not available for the general case. However, by assuming that classes are normally distributed, standard classifiers using quadratic and linear discriminant functions can be designed.

The well-known Fisher's Linear Discriminant (LFD) (Fukunaga, 1990), arises in the special case when the classes have a common covariance matrix. LFD is a classification method that projects the high dimensional data onto a line (for a binary classification problem) and performs classification in this one-dimensional space. This projection is chosen such that either the ratio of the scatter matrices (between and within classes) or the so called *Rayleigh quotient* is maximized.

More specifically, let be $A \in R^{m\times n}$ a matrix containing all the samples and let $A_c \subseteq A \in R^{l_c\times n}$ be a matrix containing the l_c labeled samples for class c, $x_i \in R^n$, $c \in \{\pm\}$. Then, the LFD is the projection u, which maximizes,

$$J(\alpha) = \frac{u^T S_B u}{u^T S_W u} \tag{1}$$

where

$$S_B = (M_+ - M_-)(M_+ - M_-)^T \tag{2}$$

$$S_W = \sum_{c\in\{\pm\}} \frac{1}{l_c}\left(A_c - M_c e_{l_c}^T\right)\left(A_c - M_c e_{l_c}^T\right)^T \tag{3}$$

are the between and within class scatter matrices respectively and

$$M_c = \frac{1}{l_c} A_c e_{l_c} \tag{4}$$

is the mean of class c and e_{l_c} is an l_c dimensional vector of ones. Traditionally, the LFD problem has been addressed by solving the generalized eigenvalue problem associated with Equation (1).

When classes are normally distributed with equal covariance, α is in the same direction as the discriminant in the corresponding Bayes classifier. Hence, for this special case LFD is equivalent to the Bayes optimal classifier. Although LFD relies heavily on assumptions that are not true in most real world problems, it has proven to be very powerful. Generally speaking when the distributions are unimodal and separated by the scatter of means, LFD becomes very appealing. One reason why LFD may be preferred over more complex classifiers is that as a linear classifier it is less prone to overfitting.

For most real world data, a linear discriminant is clearly not complex enough. Classical techniques tackle these problems by using more sophisticated distributions in modeling the optimal Bayes classifier, however these often sacrifice the closed form solution and are computationally more expensive. A relatively new approach in this domain is the kernel version of

Fisher's Discriminant (Mika et al., 1999). The main ingredient of this approach is the kernel concept, which was originally applied in Support Vector Machines and allows the efficient computation of Fisher's Discriminant in the kernel space. The linear discriminant in the kernel space corresponds to a powerful nonlinear decision function in the input space. Furthermore, different kernels can be used to accommodate the wide-range of nonlinearities possible in the data set. In what follows, we derive a slightly different formulation of the KFD problem based on duality theory which does not require the kernel to be positive semidefinite or what is equivalent,does not require the kernel to comply with Mercer's condition (Cristianini & Shawe-Taylor, 2000).

3. Automatic heterogeneous kernel selection for the KFD problem

As shown in (Xu & Zhang, 2001) and similar to (Mika et al., 2000), with the exception of an unimportant scale factor, the LFD problem can be reformulated as the following constrained convex optimization problem:

$$\begin{array}{rrcl} \displaystyle\min_{(u,\gamma)\in R^{m+1}} & \nu\frac{1}{2}\|y\|^2 & + & \frac{1}{2}(u'u) \\ \text{s.t.} & y & = & d-(Au-e\gamma). \end{array} \tag{5}$$

where $m = l_+ + l_-$ and d is an m-dimensional vector such that:

$$d_i = \begin{cases} +m/l_+ & \text{if} \quad x_i \in A_+ \\ -m/l_- & \text{if} \quad x_i \in A_- \end{cases} \tag{6}$$

and the variable ν is a positive constant introduced in (Mika et al., 2000) to address the problem of ill-conditioning of the estimated covariance matrices. This constant can also be interpreted as a capacity control parameter. In order to have strong convexity on all variables of problem (5) we can introduce the extra term γ^2 on the corresponding objective function. In this case, the regularization term is minimized with respect to both orientation u and relative location to the origin γ. Extensive computational experience, as in (Fung & Mangasarian, 2003; Lee & Mangasarian, 2001) and other publications, indicates that in similar problems (Fung & Mangasarian, 2001) this formulation is just as good as the classical formulation, with some added advantages such as strong convexity of the objective function. After adding the new term to the objective function of the problem (5) the problem becomes

$$\begin{array}{rrcl} \displaystyle\min_{(u,\gamma,y)\in R^{m+1+m}} & \nu\frac{1}{2}\|y\|^2 & + & \frac{1}{2}(u'u+\gamma^2) \\ \text{s.t.} & y & = & d-(Au-e\gamma). \end{array} \tag{7}$$

The Lagrangian of (7) is given by

$$L(u,\gamma,y,v) = \nu\frac{1}{2}\|y\|^2+\frac{1}{2}\|\begin{bmatrix} u \\ \gamma \end{bmatrix}\|^2 - v'((Au-\gamma e)+y-d) \tag{8}$$

Here $v \in R^m$ is the Lagrange multiplier associated with the equality constrained problem (7). Solving for the gradient of (8) equal to zero, we obtain the Karush-Kuhn-Tucker (KKT) necessary and sufficient optimality conditions (Mangasarian, 1994, p. 112) for our LFD problem with equality constraints as given by

$$\begin{array}{lcl} u - A'v & = & 0 \\ \gamma + e'v & = & 0 \\ \nu y - v & = & 0 \\ Au - e\gamma + y - d & = & 0 \end{array} \tag{9}$$

The first three equations of (9) give the following expressions for the original problem variables (u,γ,y) in terms of the Lagrange multiplier v:

$$u = A'v, \quad \gamma = -e'v, \quad y = \frac{v}{\nu}. \tag{10}$$

Replacing these equalities in the last equality of (9) allows us to obtain an explicit expression involving v in terms of the problem data A and d, as follows:

$$AA'v + ee'v + \tfrac{v}{\nu} - d \quad = \quad \left(HH' + \tfrac{I}{\nu}\right)v - d = 0 \tag{11}$$

where H is defined as:

$$H = [A \quad (-e)]. \tag{12}$$

From the two first equalities of (10) we have that

$$\begin{bmatrix} u \\ \gamma \end{bmatrix} = H'v \tag{13}$$

Using this equality and pre-multiplying by H' in (11) we have

$$\left(H'H + \frac{I}{\nu}\right)\begin{bmatrix} u \\ \gamma \end{bmatrix} = H'd \tag{14}$$

Solving the linear system of equations (14) gives the explicit solution $\begin{bmatrix} u \\ \gamma \end{bmatrix}$ to the LFD problem (7). To obtain our "kernelized" version of the LFD classifier we modify our equality constrained optimization problem (7) by replacing the primal variable u by its dual equivalent $u = A'v$ from (10) to obtain:

$$\begin{array}{rrcl} \displaystyle\min_{(v,\gamma,y)\in R^{m+1+m}} & \nu\frac{1}{2}\|y\|^2 & + & \frac{1}{2}(v'v+\gamma^2) \\ \text{s.t.} & y & = & d-(AA'v-e\gamma). \end{array} \tag{15}$$

where the objective function has also been modified to minimize weighted 2-norm sums of the problem variables. If we now replace the linear kernel AA' by a nonlinear kernel $K(A,A')$ as defined in the Introduction, we obtain a formulation that is equivalent to the kernel Fisher discriminant described in (Mika et al., 1999):

$$\begin{array}{rrcl} \displaystyle\min_{(v,\gamma,y)\in R^{m+1+m}} & \nu\frac{1}{2}\|y\|^2 & + & \frac{1}{2}(v'v+\gamma^2) \\ \text{s.t.} & y & = & d-(K(A,A)'v-e\gamma). \end{array} \tag{16}$$

Recent SVM formulations with least squares loss (Suykens & Vandewalle, 1999) are much the same in spirit as the problem of minimizing $\nu\frac{1}{2}\|y\|^2+\frac{1}{2}w'w$ with constraints $y=d-(Aw-e\gamma)$. Using a similar duality analysis to the one presented before, and then "kernelizing" they obtain the objective function

$$\nu\frac{1}{2}\|y\|^2+\frac{1}{2}v'K(A,A')v. \tag{17}$$

The regularization term $v'K(A,A')v$ determines that the model complexity is regularized in a reproducing kernel Hilbert space (RKHS) associated with the specific kernel K where the kernel function K has to satisfy Mercer's conditions and $K(A,A')$ has to be positive semidefinite.

By comparing the objective function (17) to problem (16), we can see that problem (16) does not regularize in terms of RKHS. Instead, the columns in a kernel matrix are simply regarded as new features $K(A,A')$ of the classification task in addition to original features A. we can construct then, classifiers based on the features introduced by a kernel in the same way as how we build models using original features in A. More precisely, in a more general framework (regularized networks (Evgeniou et al., 2000a)) our method could produce linear classifiers (with respect to the new kernel features $K(A,A')$) which minimize the cost function regularized in the span space formed by these kernel features. Thus the requirement for a kernel to be positive semidefinite could be relaxed, at the cost in some cases, of an intuitive geometrical interpretation. In this paper, however, since we are considering a Kernel fisher discriminant formulation, we will require the kernel matrix to be positive semidefinite. This requirement allows to conserve the geometrical interpretation of the KFD formulation since the kernel matrix can be seen as a "covariance" matrix on the higher dimensional space induced implicitly by the kernel mapping.

Next, Let us suppose that instead of the kernel K being defined by a single kernel mapping (i.e Gaussian, polynomial, etc.), the kernel K is instead composed of a linear combination of kernel functions $K_j, j=1,\ldots,k$, as below

$$K(A,A')=\sum_{j=1}^{k}a_jK_j(A,A'), \tag{18}$$

where $a_j\geq 0$. As it is pointed out in (Lanckriet et al., 2003), the set $\{K_1(A,A'),\ldots,K_k(A,A')\}$ can be seen as a predefined set of initial "guesses" of the kernel matrix. Note that the set $\{K_1(A,A'),\ldots,K_k(A,A')\}$ could contain very different kernel matrix models, e.g., linear, Gaussian, polynomial, all with different parameter values. Instead of fine tuning the kernel parameters for a predetermined kernel via cross-validation, we can optimize the set of values $a_i\geq 0$ in order to obtain a PSD linear combination $K(A,A')=\sum_{j=1}^{k}a_jK_j(A,A')$ suitable for the specific classification problem. Replacing equation (18) in equation (16) and solving for y in and replacing it on the objective function in (16), we can reformulate the KFD problem optimization for heterogeneous linear combinations of kernel as follows

$$\begin{array}{rl} \displaystyle\min_{(v,\gamma,a\geq 0)\in R^{m+1}} & \nu\frac{1}{2}\|d-((\sum_{j=1}^{k}a_jK_j)v-e\gamma)\|^2 \\ & +\frac{1}{2}(v'v) \end{array} \tag{19}$$

where $K_j=K_j(A,A')$. When considering linear combinations of kernels the hypothesis space may become larger, making the issue of capacity control an important one. It is known that if two classifiers have similar training error, a smaller capacity may lead to better generalization on future unseen data (Vapnik, 2000; Cherkassky & Mulier, 1998). In order to reduce the size of the hypothesis and model space and to gain strong convexity in all variables, an additional regularization term $a'a=\|a\|^2$ is added to the objective function of problem (19). The problem then becomes,

$$\begin{array}{rl} \displaystyle\min_{(v,\gamma,a\geq 0)\in R^{m+1}} & \nu\frac{1}{2}\|d-((\sum_{i=1}^{k}a_iK_i)v\ \ -e\gamma)\|^2 \\ & +\frac{1}{2}(v'v+\gamma^2+a'a) \end{array} \tag{20}$$

The corresponding *nonlinear* classifier to this nonlinear separating surface is then:

$$\left(\sum_{j=1}^{k}(a_jK_j(x',A'))\right)v-\gamma=\left\{\begin{array}{ll} >0, & \text{then } x\in A_+, \\ <0, & \text{then } x\in A_-, \\ =0, & \text{then } x\in A_+\cup A_-. \end{array}\right. \tag{21}$$

Furthermore, problem (20) can be seen as a biconvex program of the form,

$$\min_{(S,T)\in(R^{m+1},R^k)}F(S,T) \tag{22}$$

where $S=\begin{bmatrix} v \\ \gamma \end{bmatrix}$ and $T=a$

When $T = \hat{a}$ is fixed, problem (22) becomes:

$$\begin{aligned} \min_{(S)\in(R^{m+1})} \quad F(S,\hat{a}) = & \\ \min_{(v,\gamma)\in R^{m+1}} \quad & \nu\tfrac{1}{2}\|d-(\hat{K}v \ - e\gamma)\|^2 \\ & +\tfrac{1}{2}(v'v+\gamma^2) \end{aligned} \tag{23}$$

where $\hat{K} = \sum_{j=1}^{k} \hat{a}_j K_j$. This is equivalent to solve (16) with $K = \hat{K}$. On the other hand when $\hat{S} = \begin{bmatrix} \hat{v} \\ \hat{\gamma} \end{bmatrix}$ is fixed, problem (22) becomes:

$$\begin{aligned} \min_{T\geq 0\in(R^k)} \quad F(\hat{S},T) = & \quad \min_{a\geq 0\in(R^k)} F(\hat{S},a) = \\ \min_{a\geq 0\in R^k} \quad & \nu\tfrac{1}{2}\left\|d-((\textstyle\sum_{j=1}^{k}\Lambda_j a_j)-e\hat{\gamma})\right\|^2 \\ & +\tfrac{1}{2}(a'a) \end{aligned} \tag{24}$$

where $\Lambda_j = K_j v$. Subproblem (23) is an unconstrained strongly convex problem for which a unique solution in close form can be obtained by solving a $(m+1)\times(m+1)$ system of linear equations. On the other hand, subproblem (24) is also a strongly convex problems with the simple nonnegativity constraint $a \geq 0$ on k variables (k is usually very small) for which a unique solution can be obtained by solving a very simple quadratic programming problem. We are ready now to describe our proposed algorithm.

4. Automatic kernel selection KFD Algorithm

Algorithm 4.1 Automatic kernel selection KFD Algorithm (A-KFD)
Given m data points in R^n represented by the $m \times n$ matrix A and vector L of ± 1 labels denoting the class of each row of A , the parameter μ and an initial $a^0 \in \Re^k$, we generate the nonlinear classifier (21) as follows:

(0) Calculate $K_1, \ldots, K_k$, the k kernels on the kernel family, where for each i, $K_i = K_i(A, A')$. Define the vector d as in (7).

For each iteration i do:

(i) given an $a^{(i-1)}$ calculate the linear combination $K = \sum_{j=1}^{k} a_j^{(i-1)} K_j$.

(i) Solve subproblem (23) to obtain $(v^{(i)}, \gamma^{(i)})$.

(ii) Calculate $\Lambda_l = K_l v^{(i)}$ for $l = 1, \ldots, k$.

(iii) Solve subproblem (24) to obtain a^i.

Stop when a predefined maximum number of iterations is reached or when there is sufficiently little change of the objective function of problem (20) evaluated in successive iterations.

Let N_i be the number of iterations of algorithm 4.1, when $k << m$ that is usually the case, this is when the number of kernels functions considered on the kernel family is much smaller than the number of data points, then the complexity of the Algorithm 4.1 is approximately $N_i(O(m^3)) = O(m^3)$, since N_i is bounded by the maximum of iterations and the cost of solving the quadratic programming problem (24) it is "dominated" by the cost of solving problem (23). In practice, we found that Algorithm 4.1 typically converges in 3-4 iterations (3 or 4) to a local solution of problem (20).

Since each of the two optimization problems ((23) and (24)) that are required to be solved by the **A-KFD** algorithm are strongly convex and thus each of them have a unique minimizer, the **A-KFD** algorithm can also be interpreted as an Alternate Optimization (AO) problem (Bezdek & Hathaway, 2003). Classical instances of AO problems include fuzzy regression c-models and fuzzy c-means clustering.

The **A-KFD** algorithm then, inherits the convergence properties and characteristics of AO problems. As stated in (Bezdek & Hathaway, 2002), the set of points for which Algorithm 4.1 can converge can include certain type of saddle points (i.e. a point that behaves like a local minimizer only when projected along a subset of the variables). However, it also stated that is extremely difficult to find examples where converge occurs to a saddle point rather than to a local minimizer. If the initial estimate is chosen sufficiently near a solution, a local q-linear convergence result is also presented by Bezek et al in (Bezdek & Hathaway, 2002). A more detailed convergence study in the more general context of of regularization networks (Evgeniou et al., 2000a; Evgeniou et al., 2000b), including SVM type loss functions, is in preparation.

5. Numerical Experiments

We tested our algorithm on five publicly available datasets commonly used in the literature for benchmarking from the UCI Machine Learning Repository (Murphy & Aha, 1992): Ionosphere, Cleveland Heart, Pima Indians, BUPA Liver and Boston Housing. Additionally, a sixth dataset, the colon CAD dataset, relates to colorectal cancer diagnosis using virtual colonoscopy derived from computer tomographic images. We will refer to this dataset as the colon CAD dataset. The dimensionality and size of each dataset are shown in Table 1.

5.1. Numerical experience on five publicly available datasets

We compared our proposed **A-KFD** against standard KFD as described in equation (7) where the kernel model is chosen using a cross-validation tuning procedure. For our family of kernels we chose a family of 5 kernels: A linear kernel ($K = AA'$) and 4 Gaussians kernels with $\mu \in \{0.001, 0.01, 0.1, 1\}$:

$$(G_\mu)_{ij} = (K(A,B))_{ij} = \varepsilon^{-\mu\|A_i{}' - B_{\cdot j}\|^2} \quad i = 1\dots, m,\ j = 1\dots, n. \tag{25}$$

where $A \in R^{m\times n}$, $B = A' \in R^{n\times m}$. For all the experiments, in algorithm 4.1, we used an initial a^0 such that:

$$K = \sum_{j=1}^{k} a_j^{(i-1)} K_j = AA' + G_1 \tag{26}$$

That is, our initial kernel is an equally weighted combination of a linear kernel $A'A$ (the kernel with less fitting power) and G_1 (the kernel with the most fitting power). The parameter ν required for both methods was chosen to be on the following set $\{10^{-3}, 10^{-2}, \dots, 10^0, \dots, 10^{11}, 10^{12}\}$. To solve the quadratic programming (QP) problem (24) we used CPLEX 9.0 (CPL, 2004), although, since the problem to solve has nice properties and it is "small" in size ($k = 5$ in our experiments) any publicly available QP solver can be used for this task. Next, we describe the methodology used in our experiments:

1. Each dataset was normalized between -1 and 1.
2. We randomly splitted the dataset into two groups consisting of 70 % for training and 30% for testing. We called the training subset T_R and the training set T_E.
3. On the training set T_R we used a ten-fold cross-validation tuning procedure to select "optimal" values for the parameter ν in **A-KFD** and for the parameters ν and μ in the standard KFD. By "optimal" values, we mean the parameters values that maximize the ten-fold cross-validation testing correctness. A linear kernel was also considered as a kernel choice in the standard KFD.
4. Using the "optimal" values found in step 3, we build a final classification surface (21), and then evaluate the performance on the testing set T_E.

Steps 1 to 4 are repeated 10 times and the average testing set correctness is reported in Table 1. The average times over the ten runs are reported in Table 2. A paired t-test (Mitchell, 1997) at 95% confidence level was performed over the ten runs results to compare the performance of the two algorithms tested. In most of the experiments, the p-values obtained show that there is no significant difference between **A-KFD** and the the standard KFD where the kernel model is chosen using a cross-validation tuning procedure. Only on two of the datasets, ionosphere and housing there is a small statistically significant difference for the two methods, with the performance of **A-KFD** being the better of the two for the ionosphere dataset and the standar tunning being the best for the housing dataset. These results suggest that the two methods are not significantly different regarding generalization accuracy.

Table 1. Ten-fold and testing set classification accuracies and p-values for five publicly available datasets (**best and statistical significant values in bold**).

Data set ($m \times n$)	A-KFD	KFD + kernel tuning	p-value
Ionosphere (351×34)	**94.7%**	92.73%	**0.03**
Housing (506×13)	**89.9%**	89.4 %	0.40
Heart (297×13)	79.7 %	**82.2 %**	**0.04**
Pima (768×8)	74.1%	**74.4 %**	0.7
Bupa (345×6)	**70.9%**	70.5%	0.75

In all experiments, the **A-KFD** algorithm converged in average on 3 or 4 iterations, thus obtaining the final classifier in a considerable faster time that the standard KFD with kernel tuning. Table 2 shows that **A-KFD** was up to 6.3 times faster in one of the cases.

5.2. Numerical experience on the Colon CAD dataset

In this section of the paper we performed experiments on the colon CAD dataset. The classification task associated with this dataset is related to colorectal cancer diagnosis. Colorectal cancer is the third most common cancer in both men and women. Recent studies (Yee et al., 2003) have estimated that in 2003, nearly 150,000 cases of colon and rectal cancer would be diagnosed in the US, and more than 57,000 people would die from the disease, accounting for about 10% of all cancer deaths. A polyp is an small tumor that projects from the inner walls of the intestine or rectum. Early detection of polyps in the colon is critical

Table 2. Average times in seconds for both methods: **A-KFD** and standard KFD where the kernel width was obtained by tuning. Times are the averages over ten runs. Kernel calculation time and ν tuning time are included in both algorithms(**Best in bold**).

Data set ($m \times n$)	A-KFD (secs.)	KFD + kernel tuning (secs.)
Ionosphere (351×34)	**55.3**	350.0
Housing (506×13)	**134.4**	336.9
Heart (297×13)	**39.7**	109.2
Pima (768×8)	**341.5**	598.4
Bupa (345×6)	**48.2**	81.7

because polyps can turn into cancerous tumors if they are not detected in the polyp stage.

The database of high-resolution CT images used in this study was obtained at NYU Medical Center. One hundred and five (105) patients were selected so as to include positive cases (n=61) as well as negative cases (n=44). The images are preprocessed in order to calculate features based on moments of tissue intensity, volumetric and surface shape and texture characteristics. The final dataset used in this paper is a balanced subset of the original dataset consisting of 300 candidates, 145 candidates are labeled as a polyp and 155 as non-polyps. Each candidate is represented by a vector of 14 features that have the most discriminating power according to a feature selection pre-processing stage. The non-polyp points were chosen from candidates that were consistently misclassified by an existing classifier that was trained to have a very low number of false positives on the entire dataset. This means that in the given 14 dimensional feature space, the colon CAD dataset is extremely difficult to separate.

For the tests, we used the same methodology described in subsection 5.1 obtaining very similar results. The standard KFD performed in an average time of 122.0 seconds over ten runs and an average test set correctness of 73.4 %. The **A-KFD** performed in an average time of 41.21 seconds with an average test set correctness of 72.4 %. As in Section 5.1, we performed a paired t-test (Mitchell, 1997) at 95% confidence level with a p-value of $0.32 > 0.05$, this indicates that there is no significant difference between both methods in this dataset at the 95% confidence level.

In summary, **A-KFD** had the same generalization capabilities and ran almost 3 times faster than the standard KFD.

6. Conclusions and Outlook

We have proposed a simple procedure for generating heterogeneous Kernel Fisher Discriminant classifier where the kernel model is defined to be a linear combination of members of a potentially larger pre-defined family of heterogeneous kernels. Using this approach, the task of finding an "appropriate" kernel that satisfactory suits the classification task can be incorporated into the optimization problem to solve. In contrast with previous works that also consider linear combination of kernels, our proposed algorithm requires nothing more sophisticated than solving a simple nonsingular system of linear equations of the size of the number of training points m and solving a quadratic programming problem that is usually very small since it depends on the predefined number of kernels on the kernel family (5 in our experiments). The practical complexity of the **A-KFD** algorithm does not explicitly depend on the number of kernels on the predefined kernel family.

Empirical results show that the proposed method compared to the standard KFD where the kernel is selected by a cross-validation tuning procedure, is several times faster with no significant impact on generalization performance.

The convergence of the **A-KFD** algorithm is justified as a special case of the alternate optimization algorithm described in (Bezdek & Hathaway, 2003). Future work includes a more general version of the proposed algorithm in the context of regularized networks, where the convergence results are presented in a more detailed manner. Future work also includes the use of sparse kernel techniques and random projections to improve further computational efficiency. We also plan to explore the use of weak kernels (kernels that depends on a subset of the original input features) for feature selection in this framework.

7. Citations and References

References

(2004). *CPLEX optimizer.* ILOG CPLEX Division, 889 Alder Avenue, Incline Village, Nevada. `http://www.cplex.com/`.

Bach, F., Lanckriet, J., & Jordan, M. (2004). *Fast kernel learning using sequential minimal optimization*Technical Report CSD-04-1307). Division of

Computer Science, University of California, Berkeley.

Bennet, K., Momma, M., & Embrechts, M. (2002). Mark: a boosting algorithm for heterogeneous kernel models. *Proceedings KDD-2002: Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, CA* (pp. 24–31). Asscociation for Computing Machinery.

Bezdek, J., & Hathaway, R. (2002). Some notes on alternating optimization. *Proceedings of the 2002 AFSS International Conference on Fuzzy Systems* (pp. 288–300). Springer-Verlag.

Bezdek, J., & Hathaway, R. (2003). Convergence of alternating optimization. *Neural, Parallel Sci. Comput.*, *11*, 351–368.

Cherkassky, V., & Mulier, F. (1998). *Learning from data - concepts, theory and methods.* New York: John Wiley & Sons.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines.* Cambridge: Cambridge University Press.

Evgeniou, T., Pontil, M., & Poggio, T. (2000a). Regularization networks and support vector machines. *Advances in Computational Mathematics*, *13*, 1–50.

Evgeniou, T., Pontil, M., & Poggio, T. (2000b). Regularization networks and support vector machines. *Advances in Large Margin Classifiers* (pp. 171–203). Cambridge, MA: MIT Press.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition.* San Diego, CA: Academic Press.

Fung, G., & Mangasarian, O. L. (2001). Proximal support vector machine classifiers. *Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA* (pp. 77–86). New York: Asscociation for Computing Machinery. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.

Fung, G., & Mangasarian, O. L. (2003). Finite Newton method for Lagrangian support vector machine classification. *Special Issue on Support Vector Machines. Neurocomputing*, *55*, 39–55.

Hamers, B., Suykens, J., Leemans, V., & Moor, B. D. (2003). Ensemble learning of coupled parmeterised kernel models. *International Conference on Neural Information Processing* (pp. 130–133).

Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. (2003). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Lee, Y.-J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine. *Computational Optimization and Applications*, *20*, 5–22. Data Mining Institute, University of Wisconsin, Technical Report 99-03. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.

Mangasarian, O. L. (1994). *Nonlinear programming.* Philadelphia, PA: SIAM.

Mangasarian, O. L. (2000). Generalized support vector machines. *Advances in Large Margin Classifiers* (pp. 135–146). Cambridge, MA: MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

Mika, S., Rätsch, G., & Müller, K.-R. (2000). A mathematical programming approach to the kernel fisher algorithm. *NIPS* (pp. 591–597).

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., & Müller, K.-R. (1999). Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX* (pp. 41–48). IEEE.

Mitchell, T. M. (1997). *Machine learning.* Boston: McGraw-Hill.

Murphy, P. M., & Aha, D. W. (1992). UCI machine learning repository. www.ics.uci.edu/~mlearn/MLRepository.html.

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

Vapnik, V. N. (2000). *The nature of statistical learning theory.* New York: Springer. Second edition.

Xu, J., & Zhang, X. (2001). Kernel mse algorithm: A unified framework for kfd, ls-svm and krr. *Proceedings of The International Joint Conference on Neural Networks 2001.* IEEE.

Yee, J., Geetanjali, A., Hung, R., Steinauer-Gebauer, A., wall, A., & McQuaid, K. (2003). Computer tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults. *Proceeding of NEJM-2003* (pp. 2191–2200).