Short Papers

Two Variations on Fisher's Linear Discriminant for Pattern Recognition

Tristrom Cooke

Abstract—Discriminants are often used in pattern recognition to separate clusters of points in some multidimensional "feature" space. This paper provides two fast and simple techniques for improving on the classification performance provided by Fisher's linear discriminant for two classes. Both of these methods are also extended to nonlinear decision surfaces through the use of Mercer kernels.

Index Terms—Linear discriminant, classification.

1 Introduction

THERE are many methods available for characterizing patterns. For instance histograms, co-occurrence matrix measures, and fractal dimensions have all been used as measures of texture. Hence, each pattern can be represented as a point in some multidimensional "feature" space. Automatic categorization of patterns based on these features can be accomplished by using a discriminant to partition this feature space. The partition is based on a set of points having known class, which is referred to as the training set. Points belonging to the same partition will then be categorized as being produced by the same pattern. The performance of the discriminant can then be measured by using the same partitions to categorize an independent test of points, referred to as the test set. A higher percentage of correct classifications in the test set indicates a better discriminant.

Linear discriminants may be used to discriminate any number of classes of patterns, but are perhaps most commonly used when there are only two classes. An example of such a problem is in detection, where it is required that a target pattern, such as a vehicle in a radar image, is detected from among the uninteresting background patterns. Many detection problems are specified so that the classifier must produce either a particular detection rate or an upper bound for the rate at which false detections are produced. Each of these specifications will be referred to as an "operating point" for the classifier.

The Fisher discriminant [6] is the benchmark for the linear discrimination between two classes in multidimensional space. It is extremely quick to calculate since it is based only on the first and second moments of each distribution. Also, it may be shown to maximize a measure of the separation which is not specific to a particular distribution type. This makes the Fisher discriminant extremely robust. It is not optimal, however, since the discriminant may not give the minimum classification error unless the two classes are Gaussian with equal covariance.

In the quest for a linear discriminant which gives better classification errors for more general distribution types, numerous methods have been considered such as those outlined in Duda and Hart [5]. Gradient descent techniques and SVMs [2] have met with good success, often producing much better solutions than could be obtained by the Fisher discriminant. Unfortunately both of these techniques can be slow and it is often difficult to specify that the classifier should be trained for a particular operating point.

This paper provides two fast and simple techniques for improving on the linear discrimination provided by Fisher's discriminant and gives numerical examples showing favorable results compared with more complex methods. Both techniques rely on the robustness and speed of the Fisher discriminant, while incorporating some of the ideas from other discriminant methods. Due to their dependence only on dot products, these methods are also shown to be easily extended to nonlinear discrimination problems through the use of Mercer kernels.

2 Formal Description and Terminology

This paper considers two heuristics for solving the detection, or binary classification problem. In this problem, it is assumed that two classes of points are distributed in some multidimensional feature space, with unknown probability densities $f_i(\mathbf{x})$ for $i = 1 \ldots 2$. For each class i, a set of N_i points is sampled from the appropriate class distribution. These points are refered to as the training set. A binary classifier, or discriminant, is a procedure which partitions the feature space into disjoint regions Ω_1 and Ω_2 based only on the training set information. A point belonging to Ω_i may then be labelled by the classifier as belonging to class i.

The detection problem is to find a discriminant which minimizes some error criterion related to the above unknown density functions $f_i(\mathbf{x})$. Perhaps the most commonly used error criterion is the classification error, defined by

$$\text{Classification Error} = 1 - \frac{1}{2}\sum_{i=1}^{2}\int_{\Omega_i} f_i(\mathbf{x})d\mathbf{x}.$$

Another common error minimization principle is to minimize the classification error of one of the classes, while fixing that of the other class. For instance, minimizing $\int_{\Omega_1} f_2(\mathbf{x})d\mathbf{x}$ subject to $\int_{\Omega_2} f_1(\mathbf{x})d\mathbf{x} = \alpha$ for some constant α. Constraining a class error rate in this fashion is refered to as setting the operating point of the classifier.

In many applications, it is desirable to know the performance of the classifier at a number of different operating points. The usual method for showing this is a Receiver Operating Characteristic (ROC) curve. This is a plot of the probability of correct classification of one class ($\int_{\Omega_2} f_2(\mathbf{x})d\mathbf{x}$) against the probability of incorrect classification of the other ($\int_{\Omega_2} f_1(\mathbf{x})d\mathbf{x}$). Since the density functions $f_i(\mathbf{x})$ are unknown, neither the classification error nor points on the ROC curve can be known precisely. Instead, they are usually estimated from a set of points sampled from the two classes but independent of the training set. This set is called the test set and when it is sufficiently large, points on the ROC curve may be estimated to arbitrary precision.

The following section describes two classifiers which partition the feature space using a hyperplane. The performance of these discriminants are tested in Section 5.

3 The Two Linear Discriminants

In this section, two methods are presented for partitioning a multidimensional space based on training samples. It is claimed that, for reasonably dense training data (defined here to mean that the number of training samples of each class is large compared with the number of dimensions), the expected classification error

• *The author is with the Center for Sensor Signal and Information Processing, SPRI Building, 1 Warrendi Rd., Mawson Lakes, South Australia 5096. E-mail: tcooke@cssip.edu.au.*

Manuscript received 23 Feb. 2000; revised 04 Feb. 2001; accepted 16 July 2001.
Recommended for acceptance by U. Sethi.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111547.

of an independent test set using these partitions will often be significantly lower than when using the standard Fisher discriminant. In fact, the numerical results of Section 5 indicate that the second method gives similar performance to a support vector machine with a linear kernel.

It should be stressed that the following methods are heuristic. Although, both are based strongly on intuitive arguments, it has not been possible for the author to derive any analytical results concerning the performance of the methods. Since the numerical simulations described in Section 5 were so successful, the details of the methods are presented here in the hope that they will inspire further research.

3.1 1D Parameter Search

The only way to guarantee an optimal solution to the linear discriminant problem would be a multidimensional search through all possible normals for the partitioning hyperplane. This is usually infeasible for real problems which contain large numbers of variables. If this search could be reduced to a lower-dimensional space of directions in which it was more likely to find the required solution, then a faster but no longer necessarily optimal method could be obtained.

Anderson and Bahadur [1] showed that the optimal normal to the decision hyperplane separating two multidimensional Gaussians will be given by $(\mathbf{C}_1 + \gamma\mathbf{C}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$. Here, $\boldsymbol{\mu}$ and $\mathbf{C}$ are the mean and covariance, the indices correspond to the class and γ is a real constant, which may range from $-\infty$ to $+\infty$ depending on the desired operating point. It was further shown in Cooke and Peake [4] that this direction is the best that can be possibly chosen, regardless of the distribution type, when only the first and second moments of each distribution are known. Varying γ produces a one dimensional set of directions which can be searched for the optimal solution.

When $\gamma = 1$ in the above formula for the normal, this discriminant corresponds to the Fisher discriminant, so this will be the lower limit on the training performance. For the case when both distributions are only functions of their Mahalanobis distance (normal or student-t distributions for instance), this method should produce an optimal result.

3.2 Recursive Fisher

The method known as a Support Vector Machine (SVM) [2] for discriminating two classes, minimizes a particular error functional which is only slightly related to the training error. This functional is based on the minimum distance between the decision surface and each class, and the squared distance error of a number of poorly or incorrectly classified points which are termed "support vectors." The final solution which minimizes this error functional is dependent only on these support vectors, which are either close to the final decision surface or are misclassified by it.

The SVM technique generally has the advantage of producing much better classification errors than the Fisher discriminant. It also has disadvantages however. First, it is slow compared with Fisher's discriminant, especially when classifying large numbers of points. Second, the standard SVM formulation is dependent on a regularization parameter, which cannot be set a priori. Setting this parameter correctly often requires an SVM to be run for several estimates of the parameter and cross-validated with an independent subset of the training set (although implementations do exist which do not require a cross-validation set, e.g. [3]). Finally, it is difficult to specify that the resulting classifier should be trained for a particular operating point.

To capitalize on an advantage of the SVM, while avoiding most of the disadvantages, a recursive Fisher algorithm was developed based on the concept of using support vectors. The procedure consists of the following four steps:

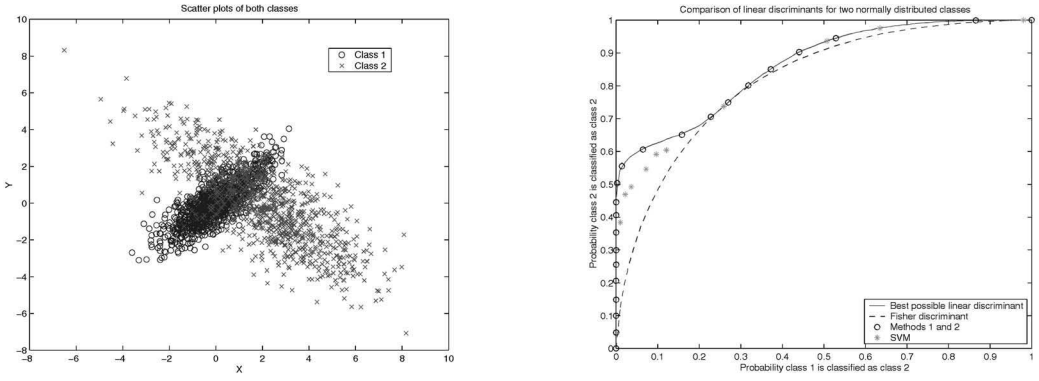
1. **Initialization.** Set the percentage of support vectors $S = S_1$. Then, calculate the initial hyperplane decision surface $\mathbf{n}.\mathbf{x} = c$. The normal to the hyperplane $\mathbf{n}$ may be calculated by using the Fisher discriminant, while the constant c should be chosen in an attempt to satisfy the required optimality condition (for instance, to minimize the classification error, c might be chosen to minimize the training error).
2. **Choosing support vectors.** Generate two new distributions by keeping the closest S percent of points from each class to the decision surface $\mathbf{n}.\mathbf{x} = c$.
3. **Fisher discriminant.** Calculate the new decision surface by finding the Fisher discriminant of these two new distributions to determine $\mathbf{n}$ and again choose c to satisfy the optimality condition.
4. **Loop termination condition.** Decrease the percentage of support vectors S by some amount ΔS. If S is below some threshold T, then end the loop, otherwise go to step 2.

An intuitive explanation for the above method can be seen by a comparison with how a human might accomplish the same task. First, one might use the overall shape of the distribution to arrive at a rough estimate of the discriminant with the best training error. After this, one might make small adjustments to the decision surface until the best result is obtained. Since small adjustments will not affect the way in which points that are far from the decision surface are classified, then these have essentially become redundant. Making smaller and smaller adjustments thus corresponds to shrinking the number of support vectors in the above algorithm. Eventually, the algorithm might be expected to converge to a local minimum in the training error. It should be noted that this explanation is not specifically dependent on the discriminant being used and perhaps the method might be suitable for other classifiers. This has not been investigated however.

For unimodal distributions, the best result seems to be usually obtained for the lowest percentage of support vectors. This is true even when both classes are Gaussian with equal covariance. For this case the original Fisher estimate is optimal so removing the first percentage of support vectors actually reduces the classifier performance. As further support vectors are removed however, the resulting discriminant seems to converge towards the original optimal estimate.

As would be expected from the intuitive explanation, for multimodal classes decreasing the percentage of support vectors can occasionally significantly degrade discrimination performance. This is likely due to the local minimum training error being very far from the global minimum. For this reason, the training error should be measured each time through the loop and the intermediate (or the final) decision surface which gives the best training error should be used. In this way, since the initial discriminant is the Fisher discriminant, this technique can only improve the training error. When the training set is reasonably dense, it is claimed that this should also correspond to a reduced test error.

The technique described above has three arbitrary parameters, S_1, ΔS, and T. Obviously, for best results, ΔS and T should be as low as possible, but this results in increased computation time. Some testing has indicated that $S_1 = 90\%$ and $\Delta S = T = 10\%$ provides a very good compromise and these are the parameters used later in



(a) (b)

Fig. 1. Comparison of linear discriminant results for Gaussians. (a) Distribution of the classes. (b) Comparison of performance.

the numerical simulations. This setting means that the new method need only 10 evaluations of Fisher's discriminant, which still makes it considerably faster than most other methods.

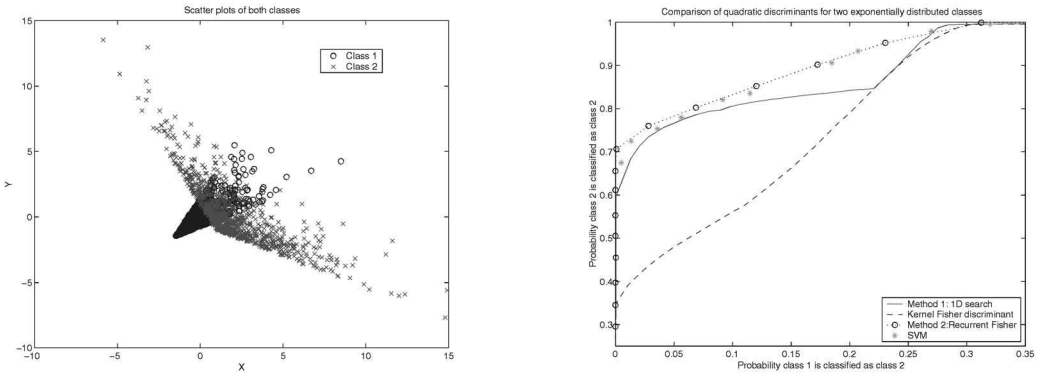
4 Extension to Nonlinear Discrimination

The standard technique for allowing SVMs (which are principally linear discriminants) to solve nonlinear discrimination problems involves the use of Mercer kernels. Mika et al. [8] shows that the same procedure can be applied to the Fisher discriminant, yielding a nonlinear discriminant having an accuracy comparable to that of SVMs. They refer to this new discriminant as the Kernel Fisher Discriminant (KFD).

The basic idea behind the kernel method is that a nonlinear decision surface can be exactly the same as a linear decision surface in a higher-dimensional space. For instance, a quadratic discriminant in coordinates (x_1, x_2) can be obtained by constructing a linear discriminant in the five-dimensional space having coordinates $(x_1, x_2, x_1^2, x_1x_2, x_2^2)$. For higher order discriminants however, the number of features required quickly becomes unmanageable. Suppose that $\mathbf{x}$ is a point in the lower-dimensional space and that $\mathbf{\Phi}(\mathbf{x})$ is a mapping of this point into a higher-dimensional space. Then, by using Mercer kernels, which are a set of functions $k(\mathbf{x}, \mathbf{y}) = \mathbf{\Phi}(\mathbf{x}).\mathbf{\Phi}(\mathbf{y})$ which express the dot product of the higher-dimensional space in terms of the lower-dimensional coordinates, it is often not necessary to perform the mapping $\mathbf{\Phi}$ directly. For instance, the polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}.\mathbf{y})^n$ would correspond to a mapping of the data into the space of all monomials with degree less than or equal to n. Another commonly used Mercer kernel is the Gaussian radial basis function $k(\mathbf{x}, \mathbf{y}) = \exp(-|\mathbf{x} - \mathbf{y}|^2/c)$ for some positive constant c. More information concerning Mercer kernels can be obtained in Vapnik [10].

The two linear discriminants presented in Section 3 have a form similar to the Fisher discriminant. Hence, similar analysis to that in Mika et al. [8] may be used to extend these to produce nonlinear discriminants. They lose, however, many of the properties that made them attractive as linear discriminants. First, the computational requirements are increased. Mika et al. [9], reports that, for a training set of size N, a greedy approximation method may be used to implement a kernel Fisher discriminant with computational complexity $\mathcal{O}(N^{3/2})$ for large N. Since the new algorithms require the added calculation of the training error, the complexity of these methods should be roughly $\mathcal{O}(N^2)$ which is similar to that obtained for SVMs [7].

Second, new kernel parameters may be required to be set to prevent overfitting. These need to be set using cross-validation in the same way that the regularization parameter for SVMs is set. Nonlinear SVMs require both the kernel parameters and the regularization parameter to be set, so is still slightly worse in this respect. Finally, although the original training data may be dense, using a Mercer kernel is equivalent to increasing the dimensionality of the discrimination problem. The training data may no



(a) (b)

Fig. 2. Comparison of quadratic discriminant results for exponential distributions. (a) Distribution of the classes. (b) Comparison of performance.

TABLE 1
Results from Testing the New Discriminants on SAR Imagery

	Classification error (percent)			
No. Points	Fisher	1D Search	Recursive Fisher	SVM
500	16.58 ± 0.97	14.56 ± 0.39	13.78 ± 0.44	14.06 ± 0.47
200	17.18 ± 1.4	14.90 ± 0.67	14.13 ± 0.69	14.52 ± 0.87
100	17.20 ± 1.9	15.29 ± 1.2	14.30 ± 0.75	15.14 ± 1.5
50	18.25 ± 2.2	16.09 ± 1.7	15.03 ± 1.3	16.49 ± 1.8
20	19.29 ± 2.9	17.97 ± 2.8	16.82 ± 2.5	19.58 ± 3.5

longer be dense in this higher space, so minimizing the training error will be less likely to reduce the corresponding test error. Hence, for extremely high dimensionality kernels like the radial basis function, the two new methods may not yield a good performance and the SVM formulation may be more appropriate. The numerical examples in Section 5 show that the new methods may still be useful for other kernels such as the quadratic kernel $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}.\mathbf{y})^2$.

Following the analysis in Mika et al. [8], since the normal to the decision surface between two distributions should belong to the vector space spanned by the points in the distributions, we can write the normal vector of the linear discriminant in the higher-dimensional space as

$$\mathbf{n} = \sum_{i=1}^{N} \alpha_i \mathbf{\Phi}(\mathbf{x}_i), \tag{1}$$

where $N = N_1 + N_2$ is the total number of points in both classes and $\mathbf{x}_i$ is the ith point from the set of all points. Now, if the two classes in the high-dimensional space have means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and covariances $\mathbf{C}_1$ and $\mathbf{C}_2$, then the Fisher discriminant maximizes the expression for the separability given by

$$S = \frac{(\mathbf{n}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1))^2}{\mathbf{n}^T\mathbf{C}_1\mathbf{n} + \mathbf{n}^T\mathbf{C}_2\mathbf{n}}.$$

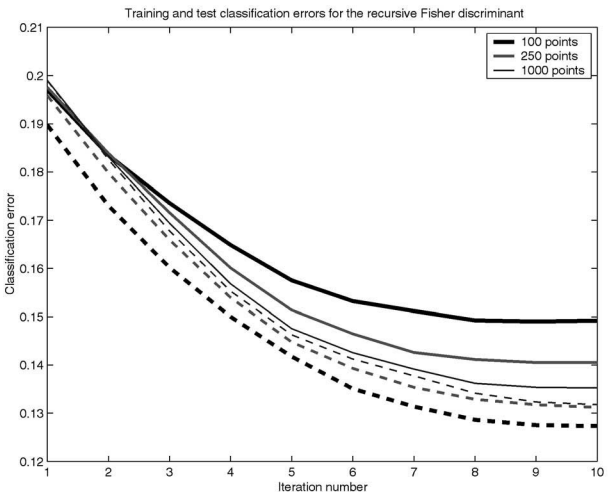


Fig. 3. Training and test errors as a function of iteration for the recursive Fisher discriminant.

In order to evaluate S without the need to evaluate the mapping $\mathbf{\Phi}$, (1) is applied to the expression containing the mean, yielding

$$\begin{aligned}\mathbf{n}^T\mu_1 &= \frac{1}{N_1}\mathbf{n}^T \sum_{j=1}^{N_1} \mathbf{\Phi}(\mathbf{x}_j^1) \\ &= \frac{1}{N_1}\sum_{i=1}^{N}\alpha_i \sum_{j=1}^{N_1} \mathbf{\Phi}(\mathbf{x}_i)^T\mathbf{\Phi}(\mathbf{x}_j^1) \\ &= \frac{1}{N_1}\sum_{i=1}^{N}\alpha_i \sum_{j=1}^{N_1} k(\mathbf{x}_i, \mathbf{x}_j^1) \\ &= \frac{1}{N_1}\boldsymbol{\alpha}^{\mathbf{T}}\mathbf{k_1},\end{aligned}$$

where $\mathbf{x}_j^1$ is the jth point of the first class. In a similar way, the expression containing the covariance may be written after some manipulation as

$$\begin{aligned}\mathbf{n}^T\mathbf{C}_1\mathbf{n} &= \frac{1}{N_1}\sum_{i=1}^{N_1}\left(\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_i^1, \mathbf{x}_j)\right)^2 - (\mathbf{n}^T\boldsymbol{\mu}_1)^2 \\ &= \frac{1}{N_1}\boldsymbol{\alpha}^{\mathbf{T}}\mathbf{K_1}\mathbf{K_1}^{\mathbf{T}}\boldsymbol{\alpha} - \frac{1}{N_1^2}\boldsymbol{\alpha}^{\mathbf{T}}\mathbf{k_1}\mathbf{k_1^T}\boldsymbol{\alpha}.\end{aligned}$$

The above expressions imply that the separability criterion to be maximized for the Fisher discriminant can be written as $S = (\alpha^{\mathbf{T}}\mathbf{A}\alpha)/(\alpha^{\mathbf{T}}\mathbf{B}\alpha)$ for some $N \times N$ matrices $\mathbf{A}$ and $\mathbf{B}$. The separability can be maximized by choosing α to be the eigenvector of $\mathbf{AB}^{-1}$ having the highest eigenvalue. Once α is known, the normal to the decision hyperplane in the higher-dimensional space can be calculated from (1) and a ROC curve can be drawn by examining the distance of each of the distributions from the hyperplane passing through the origin. For a point $\mathbf{\Phi}(\mathbf{x})$, this distance can be calculated using

$$\mathbf{n}\mathbf{\Phi}(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i k(\mathbf{x}, \mathbf{x}_i).$$

The two new linear discriminants described in this paper can be implemented in a similar fashion. There is a slight difference however for the 1D parameter search, since it requires the evaluation of $(\mathbf{C}_1 + \gamma\mathbf{C}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ for the calculation of the search directions. This is the equivalent of maximizing a separability of $S(\gamma) = (\mathbf{n}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1))^2/(\mathbf{n}^T\mathbf{C}_1\mathbf{n} + \gamma\mathbf{n}^T\mathbf{C}_2\mathbf{n})$, or finding the eigenvector of $\mathbf{AB}(\gamma)^{-1}$ having the highest eigenvalue, for an easily computable matrix $\mathbf{B}(\gamma)$.

TABLE 2
Discriminant Performance on Benchmarks

	Classification error (percent)			
Data set	Fisher	1D Search	Recursive Fisher	SVM
Banana (linear)	40.3 ± 2.5	39.2 ± 1.9	38.8 ± 1.6	44.5 ± 3.6
Banana (quad)	33.8 ± 2.8	29.4 ± 1.6	28.4 ± 1.8	34.1 ± 2.9
Banana (cubic)	23.9 ± 2.8	22.9 ± 3.1	19.4 ± 1.8	23.6 ± 3.1
Banana (quartic)	14.4 ± 1.0	14.2 ± 1.0	11.7 ± 0.7	12.2 ± 0.9
Breast Cancer	35.5 ± 5.3	35.4 ± 5.7	38.0 ± 5.2	36.6 ± 5.5
Diabetis	26.1 ± 1.8	26.0 ± 1.8	26.5 ± 2.1	26.9 ± 2.1
Flare	33.9 ± 2.1	33.1 ± 1.5	33.0 ± 1.5	32.6 ± 2.0
Thyroid	18.0 ± 5.0	13.5 ± 3.8	12.4 ± 4.0	15.3 ± 5.2
Titanic	32.9 ± 5.2	30.7 ± 1.3	29.8 ± 1.0	29.9 ± 0.9

5 Numerical Results

To show the possible improvement in performance of the two new linear discriminants compared with the Fisher discriminant, four numerical examples were examined. The first two examples considered the linear discriminants and their nonlinear extensions applied to simulated data sets. The discriminants were trained for various operating points and the results from each separate discriminant were displayed on ROC curves. For completeness, the classifiers were also compared against a support vector machine (SVM) and, in the second example, a table was produced to show the effects of training data size on the classifier performance. The third example considers the performance of linear and nonlinear discriminants on some benchmark data sets. Finally, real data from a target detection problem in Synthetic Aperture Radar (SAR) imagery was used to compare the discrimination results of the two new methods against that of the standard Fisher discriminant.

The first example considered the two 2D Gaussian distributed classes, as shown in Fig. 1a. The means and covariances of the classes were given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{C}_1 = \begin{bmatrix} 1.25 & 1 \\ 1 & 1.25 \end{bmatrix},$$
$$\boldsymbol{\mu}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}.$$

A training set of $4,000$ points ($2,000$ from each class) and a test set of size $100,000$ were then prepared and used to create the ROC curves in Fig. 1b. In this example, both of the two new methods yielded the best possible solution for any linear discriminant. For the 1D parameter search, this is not surprising since the method was based on the optimal solution for two normally distributed classes [1]. The recursive Fisher method which was not based on any such assumptions gave equally good results. Both techniques give much stronger results than the Fisher discriminant.

The points in Fig. 1b corresponding to the SVM were obtained with SVMlight [7] for the linear kernel $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}.\mathbf{y}$. Also, half of the training set was used for cross-validation. In this example, the new methods gave a better performance than the SVM. This indicates that, for dense data sets, minimizing the training error may give a better test error than the SVM procedure which minimizes a measure of the empirical risk.

The second example compares the nonlinear variants of the new linear discriminants. The kernel used was $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}.\mathbf{y})^2$, which results in a quadratic discriminant. First, two independent 1D negative exponential distributions were generated for each class and then translated and scaled to give the same class means and covariances as for the first example. Some points from these two-dimensional classes are shown in Fig. 2a.

Fig. 2b shows the ROC curves obtained by using the same training, test, and cross-validation set sizes as in the first example. Again, the 1D parameter search and the recurrent Fisher still significantly outperform the standard Fisher. The SVM discriminant however now appears to do better than the 1D search, although it still lags a little behind the performance of the recurrent Fisher.

Table 1 shows the effect of training class size on the performance of the discriminants in the second example. Even for relatively small sample sizes, the two new methods show significant improvement over the Fisher discriminant. In fact for very small sample sizes, all of the methods seem to perform better than the SVM. This may be somewhat misleading though, since half of the SVM training set was used as a cross-validation set, while the other methods did not require cross-validation. Even taking this into account however, the recursive Fisher discriminant appears to give similar performance for all but the lowest training set size.

To further examine the behavior of the recursive Fisher discriminant for the second example, Fig. 3 shows the average training error (solid lines) and test error (dashed lines) as a function of the iteration number. For this plot, the training error of the second class was fixed at 20 percent. As expected, more overfitting occured for smaller numbers of points, which resulted in smaller training errors but higher test errors. Increasing the number of points gave greater consistency between the training and test errors.

The third example considers a handful of the benchmark databases available at http://ida.first.gmd.de/~raetsch/data/benchmarks.htm. The data is considered with 100 different training and test set splits. The results of applying the new discriminants, which are linear unless specified, are presented in

TABLE 3
Results from Testing the New Discriminants on SAR Imagery

Area Under ROC Curve

Features	Fisher	1D Search	Recursive Fisher
$\{1,2\}$	0.918	0.979	0.978
$\{1,3\}$	0.910	0.967	0.967
$\{2,3,4\}$	0.930	0.972	0.971
$\{3,4,5\}$	0.915	0.947	0.948
$\{3,4,5,6\}$	0.936	0.970	0.971

Table 2. The performance is measured as the mean classification error in percent, with the error term corresponding to the standard deviation over the splits.

From the table, it can be seen that both of the new algorithms generally perform better than the standard Fisher discriminant. While the 1D search gave consistent improvements on the Fisher discriminant, the recursive Fisher seemed to perform worse on the two sparsest data sets (Diabetes and Breast Cancer). Only the second of these was statistically significant however and it is worth noting that the SVM also performed worse than Fisher's discriminant on these same data sets. In the remaining data, the recursive Fisher discriminant was competitive with the best of the other classifiers considered.

The final example uses the 1D parameter search, recursive Fisher, and standard Fisher discriminants to classify 1m resolution SAR imagery. Six features were calculated for each image from a data set of 993 target and 157,816 background images. This set was then bisected into separate training and test sets and the three linear discriminants were applied to various combinations of the features. The results from each discriminant are shown in Table 3 as areas under the ROC curve and show that both of the new methods provide significant improvement over the standard Fisher method.

6 Conclusions

Two fast linear discriminants have been described. Both of these methods will allow better training performance for pattern detection problems than the standard Fisher discriminant.

The first of the two methods, the 1D parameter search, should provide optimal results for symmetrical feature distributions, but performs somewhat less well when this assumption is not satisfied. For dense training sets, the second method (recursive Fisher discriminant) has, in the examples tested, consistently provided a comparable classification performance to linear support vector machines, but for a much lower computational burden. At the expense of an increase in computational complexity and a possible decrease in generalization performance, both of these methods have been extended using Mercer kernels to provide nonlinear discrimination.

Acknowledgments

The author would like to thank DSTO (Defence Science and Technology Organization) for providing the INGARA SAR images of various targets and backgrounds.

References

[1] T.W. Anderson and R.R. Bahadur, "Classification into Two Multivariate Normal Distributions with Different Covariance Matrices," *Annals of Math. Statistics,* vol. 33, pp. 420-431, June 1962.
[2] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 1-47, 1998.
[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Kernel Parameters for Support Vector Machines," submitted to Machine Learning, 2000, http://www. ens-lyon.fr/~ochapell/kernel_params.ps.gz.
[4] T. Cooke and M. Peake, "The Optimal Classification Using a Linear Discriminant for Two Point Classes having Known Mean and Covariance," to appear in J. Multivariate Analysis.
[5] R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," New York: Wiley-Interscience, 1973.
[6] R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics,* vol. 7, part II, pp. 179-188, 1936.
[7] T. Joachims, "Making Large—Scale SVM Learning Practical," *Advances in Kernel Methods - Support Vector Learning,* B. Scölkopf, C. Burges, and A. Smola eds., MIT-Press 1999, http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_99a.pdf.
[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, "Fisher Discriminant Analysis with Kernels," *Neural Networks for Signal Processing,* vol. 9, pp. 41-48, 1999.
[9] S. Mika, A.J. Smola, and B. Schölkopf, "An Improved Training Algorithm for Fisher Kernel Discriminants," *Proc. Artifical Intelligence and Statistics, 2001, (AISTATS '01).* T. Jaakkaola and T. Richardson, eds., pp. 98-104, 2001.
[10] V.N. Vapnik, *Statistical Learning Theory.* New York: Wiley, 1998.