NUWC-NPT Technical Report 10,663
30 June 1994

Procrustes: A Feature Set Reduction Technique

Stephen G. Greineder
Tod E. Luginbuhl
Submarine Sonar Department

Roy L. Streit
Combat Control Systems Department

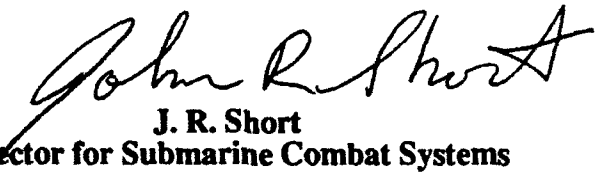
Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

The work presented in this report was sponsored by the Office of Naval Research; Program Manager, T. Goldsberry (Code 321W).

The technical reviewer for this report was D. A. Abraham (Code 2121).

Reviewed and Approved: 30 June 1994



J. R. Short
Director for Submarine Combat Systems

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 June 1994	Final

4. TITLE AND SUBTITLE

Procrustes: A Feature Set Reduction Technique

5. FUNDING NUMBERS

PR A60070

6. AUTHOR(S)

S. G. Greineder, T. E. Luginbuhl, and R. L. Streit

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center Detachment
39 Smith Street
New London, Connecticut 06320-5594

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 10,663

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Chief of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. .

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report explores the effectiveness of a new method for feature reduction and interpretation called Procrustes ordering. The investigation is performed using real data from eleven acoustic signal classes; hold-out studies are used to establish confidence in the conclusions reached. A significance test of the Procrustes angles based on a feature generation model is proposed. Additionally, an experimental statistical methodology is introduced to evaluate varying feature orderings derived from multiple trials using the same data set. Procrustes ordering is used in conjunction with a new variation of Fisher's method called "smoothed" Fisher. The variation is obtained by using a recently developed maximum likelihood trained probabilistic neural network, called Streit's Probabilistic Neural Network (SPNN), to provide smoothed estimates of the parameters defining the Fisher projection space. The results show that, on the given data set, Procrustes ordering used in conjunction with smoothed Fisher is an excellent method for feature reduction and interpretation. In addition, it is shown that Procrustes ordering is suitable for *in situ* application because it is fast and easy to compute on serial computers.

14. SUBJECT TERMS

Procrustes Ordering
Probabilistic Neural Network

15. NUMBER OF PAGES

40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

M	number of distinct signal classes
C	pooled labeled training set
C_j	class j training set
T_j	number of samples in class j
PDF	probability density function
$g_j(X\|\lambda_j)$	probability density function (PDF) for class j
λ_j	parameters defining the PDF of class j
λ	set of parameters for mixture of mixtures PDF
j	index of classes
α_j	*apriori* probability of class j
G_j	number of Gaussian components in class j
π_{ij}	mixing proportion of Gaussian component i in class j
μ_{ij}	mean vector of Gaussian component i in class j
$N(\cdot)$	normal density function
Σ_{kernel}	covariance matrix common to all Gaussian PDF mixture components
$\overline{X}_j$	exact mean of the estimated PDF for class j
Σ_j	exact covariance matrix of the estimated PDF for class j
$L(X\|\lambda)$	estimated PDF of the pooled training set
i	component index
$\overline{X}_{pool}$	exact mean of the estimated PDF, $L(X\|\lambda)$

Σ_{pool}	exact covariance of the estimated PDF, $L(X\|\lambda)$
$\Sigma_{components}$	sum of the covariances of each Gaussian component mean about the class mean
$\Sigma_{classes}$	sum of the covariances of the class means about the pooled mean
$J(w)$	Rayleigh quotient
w	n dimensional vector
ϕ_j	Procrustes angle of feature j
$\tilde{W}$	matrix whose columns are the non-zero eigenvectors that define the FPS
x	n dimensional feature vector
k	dimension of Fisher Projection Space; $M-1$
L	cholesky factor of the matrix, $\Sigma_{kernel} + \Sigma_{components}$
$P_{n,k}(\phi)$	PDF of the Procrustes angle, ϕ
$\Gamma(\cdot)$	Gamma function
n	number of features
β	level of significance of a significance level test

PROCRUSTES: A FEATURE SET REDUCTION TECHNIQUE

INTRODUCTION AND BACKGROUND

The primary objective of this technical report is to introduce a new method, called Procrustes ordering, for feature reduction and interpretation. In this report, Procrustes ordering is used in conjunction with a new variation of Fisher's method called "smoothed" Fisher; however, it can be used in conjuction with any feature-based classification pattern recognition method.

The canonical form for an automatic pattern recognition system contains three major components: a measurement system to convert the input into a form for further processing, a feature extractor that represents characteristic information and compresses the input, and a classifier that categorizes the input data based on computed features. When the different event catagories (or classes) have known unique measurable characteristics, the categorization (or classification) problem is straightforward. Typically, features that separate the classes are unknown and the canonical procedure for implementing a pattern recognition system is usually undertaken in the following (supervised) manner:

1. A collection of exemplars of each event is compiled. (Note: The set of available exemplars from all classes is called the *design set.*)

2. Features (i.e., real-valued functions of the data) are defined to measure class specific properties of each exemplar. (Note: The set of all features is called the *feature set*, and the values of the features extracted from one exemplar is called a *feature vector.*)

3. A classifier is trained on the feature vectors of the design set.

4. Unknown events are classified using the trained classifier.

This procedure is clearly imperfect, but it is the method of choice when adequate event models are not available. Poor feature sets cause a number of difficulties for automatic classification. In the pattern recognition literature, it is well known that too many features will decrease overall classification accuracy. The presence of this limitation is determined by the ratio of the number of features to the number of samples in the training set. Theoretical studies based on idealized Gaussian class assumptions (see, e.g., Foley [1], Jain and Waller [2], or Streit [3]) show that this counter-intuitive "performance peaking" phenomenon is due to the "curse of dimensionality." Moreover, empirical studies support the occurrence of peaking in many diverse applications where Gaussian assumptions do not hold. Peaking affects all classifiers, whether neural network or classical. Identifying features that do not enhance classification performance is another important problem in feature set design. Superfluous features contribute "opportunities" for misclassification and should be eliminated to improve system robustness. Small numbers of exemplars for one or more of the signal event classes greatly exacerbate these problems. The complexity and cost of feature measurement systems is directly related to the number of computed features. Consequently, from both a performance and economic perspective, it is important to have effective feature reduction algorithms.

In some applications the measured events exhibit variations due to differences in generating mechanisms, changing noise backgrounds, and measurement system performance. Feature sets that work well in one environment and fail miserably in another cannot form the basis for a robust classification system. It seems inevitable that robust systems will require adaptive *in situ* feature set selection and the continual compilation of event exemplars. Given a list of features that are known to be useful in certain situations, feature selection

for a current design set is indistinguishable from feature reduction as it is understood in this report. Therefore, adaptive feature selection is feature reduction on an evolving design set. Clearly, feature reduction algorithms must be computationally fast if adaptive feature selection is to be undertaken *in situ*.

FEATURE REDUCTION AND INTERPRETATION

Overview of Feature Reduction Algorithms

A commonly used feature reduction method is a classical method attributed to R. A. Fisher that dates to the 1930s (see[4]). It does not linearly order the individual features in terms of their relative importance to classification, but it is very fast computationally. Fisher's method derives a new set of features that are linear combinations of the original features. These new features are optimal for Bayesian classification in the case of homoscedastic, Gaussian distributed classes. In the statistical literature, Gaussian mixtures that have a common covariance matrix are called *homoscedastic* mixtures. The span of the derived features is called the multiclass Fisher projection space (FPS). The FPS maximally separates the class means relative to the class variances. This geometric interpretation greatly facilitates intuition and strongly indicates that the FPS is a good space for feature reduction. If the classes are linearly separable in the FPS, then Fisher's linear discriminator, defined on the FPS, can be used for classification. The use of the FPS does not guarantee linear separability; however, the maximal separation property of the FPS suggests that it is a good reduced feature space for nonlinear classification problems. The distinction between the traditional use of the FPS for linear discrimination and the use of the FPS for feature reduction followed by nonlinear discrimination is fundamental. The FPS is unlikely to contain any of the original features in its span, and methods for selecting subsets of the original features for classification by exploiting their relationship to the FPS do not appear to be discussed in the literature. The feature reduction method described in this report, Procrustes ordering, chooses a subset of the original feature set that best approximates (in the least squares sence)

the FPS.

The ideal feature reduction and interpretation method should have several properties. These incude the following:

1. It should preserve the natural interpretation of the original features. Features that have natural signal interpretations (e.g., bandwidth, duration, spectral level, etc.) may not be readily interpreted if they are modified. The FPS fails in this regard because the derived features are linear combinations of the original features.

2. The computational complexity and storage requirements of the feature reduction method should be small enough to enable fast computation for *in situ* applications.

3. The feature reduction method should be compatible with nonlinear and non-Gaussian discrimination problems.

4. The reduction method should provide intuitive interpretations that facilitate problem understanding and insight. The FPS is very successful when measured by this criterion.

5. The reduction method should satisfy an optimality criterion of some kind in specialized problems. For instance, linear discrimination in the FPS is optimal in homoscedastic Gaussian multiclass problems.

The optimal feature set is the set with the lowest classification error rate. The direct algorithm for solving for this optimal set is called the exhaustive combination method (ECM) because it examines all possible combinations of features. To find the best feature set from n features by the ECM requires examining all 2^n possible feature subsets. The ECM is clearly impractical for *in situ* application unless the number of features is small because

the number of possible feature combinations grows exponentially with n. For instance, the example presented in this report contains 70 features, and to find the best 15 features, the ECM requires examination of approximately 7.2×10^{14} feature sets. Finding the best subset (of any size) requires examining $2^{70} \cong 1.2 \times 10^{21}$ feature sets. A branch and bound (BAB) technique, developed by Narendra and Fukunaga [5], also yields an optimal feature set choice, and is more efficient than the ECM because it does not examine directly all possible feature sets. Many alternative methods of feature subset selection have also been studied and reported in the literature (see [6] - [11]). The ECM, BAB and the other techniques cited here were not investigated in this report because of limited resources.

Linearly ordering the individual features by some measure of their importance to correct classification is a natural approach to feature reduction. Such orderings are easily thresholded for various purposes, including feature reduction. One readily available ordering is the single feature classification performance ordering (SFCPO). The SFCPO ranks the features by the classification performance when each feature is used alone. The classification method employed for these one dimensional problems can be any suitable multiclass classifier, including probabilistic neural networks trained by maximum likelihood methods. This ordering is quite good at optimizing classification performance, as the example presented later shows, and it does not have severe computational overhead. When measured against the above five ideal properties, the SFCPO satifies the first three criteria but not the last two. The SFCPO is used in this report as a benchmark algorithm for comparison purposes.

The selective addition method (SAM) chooses a feature order in the following way. The first feature is the feature with the lowest classification error, when only singleton feature sets are used for classification. The second feature is the feature that, in combination with the

first feature already selected, yields the lowest classification error. The third and subsequent features are selected similarly. The SAM is significantly less computationally intensive than the ECM, but it is still not fast enough for *in situ* application, as shown by the example. Hold-out studies of the SAM to establish confidence limits on classification performance were not undertaken in this report because the required computation time was prohibitive.

A relative of the SAM is the selective deletion method (SDM), that proceeds by deleting poor features one at a time in a manner analogous to the SAM. Although the SDM is not studied in this report, its computational requirements are very similar to the SAMs. Unlike the ECM, both the SAM and the SDM result in a linear ordering of the features from best to poorest in terms of their relative contribution to classification. These two linear orderings are not, in general, the same.

The Procrustes ordering is a linear ordering of the individual features that requires less computation and provides improved classification performance relative to the other techniques examined in this report. The Procrustes ordering satisfies the five criteria presented previously. In particular, Procrustes ordering provides a natural geometrical connection between feature order and the FPS which allows geometrical insight. The Procrustes ordering is obtained from the Procrustes angles between the original features and the FPS. The Procrustes angle is defined to be the smallest angle between a given feature and any non-zero vector in the FPS (see equation (12)). It is a measure of linear independence between a feature and the FPS. If the angle of a particular feature is near zero, the feature is nearly in the span of the FPS; however, if the angle is near 90 degrees, the feature is nearly orthogonal to the FPS. Intuitively, features with small Procrustes angles are good features for classification, whereas, features with large Procrustes angles are poor features for classification.

Further discussion of the Procrustes ordering and angles is provided elsewhere in this section.

The remainder of this section is devoted to several topics. First, a review of probabilistic neural networks (PNNs) and maximum likelihood (ML) training algorithms is provided. Next, a detailed description of a variation of Fisher's method called "smoothed" Fisher is discussed. Finally, the Procrustes ordering is described in detail.

Review of SPNN and Maximum Likelihood Training

Probabilistic neural networks are based on kernel, or Parzen window (see [4] Section 4.3), estimates of probability density functions (PDFs). Nonlinear discriminant functions for classification are derived from Parzen window estimates of the class PDFs by substituting the estimated class PDFs directly into a Bayesian classifier. Parzen window PDF estimators are readily interpretable in statistical terms, and remarkably, can be mapped onto a feed-forward neural network structure. The neural network interpretation of Parzen window estimators was first discussed and named PNNs by Specht [12]. The primary virtue of Specht's PNN is that it trains almost immediately with little computational effort. Its primary drawback is that it requires as many neural network nodes as training data.

The use of maximum likelihood methods to train PNNs, to significantly reduce the PNN size, was first discussed by Streit [13]. ML training of PNNs is extremely fast compared with the standard back-propagation method for training feed-forward neural networks. PNNs that use radially symmetric kernels are Radial Basis Function (RBF) networks. Maximum likelihood trained PNNs are as efficient as RBF networks and may represent the sample data better than RBF networks trained by nonprobabilistic methods. Consequently, ML training is important because it enables small sized, statistically robust PNNs to be rapidly

trained on large data sets. Streit's PNN is used in this report and it is referred to herein as SPNN. For a full discussion of SPNN, complete with a mathematical derivation of the training algorithm, the reader is referred to [14].

SPNN assumes that the training samples are statistically independent and that the class labels are known and correct (i.e., SPNN uses supervised training). The objective of ML training is to estimate the parameters in a mixture Gaussian approximation to the class PDFs. There is no loss of generality in using mixture Gaussians to approximate class PDFs since every continuous PDF can be approximated arbitrarily closely by such a mixture. SPNN's ML training algorithm estimates class PDFs simultaneously for all classes by exploiting cross-class data pooling ideas that originate in Fisher's work (see [14]). Simultaneous PDF approximation is made possible by requiring a common covariance matrix across all classes.

Automatic pattern recognition systems are often plagued by data poverty problems. Typically, one or more event classes have too few exemplars in the training set to enable satisfactory event models to be trained. SPNN mitigates these problems by cross-class pooling of the training data. Cross-class pooling enables sparsely represented events to "borrow" structure from well represented classes. Done properly, cross-class pooling has its greatest effects on the most sparsely represented classes and has very little effect on well represented classes. SPNN's training algorithm is a robust data-sensitive method for treating the data poverty problem.

Let $M \geq 1$ denote the number of classes, let $C_j = \{X_{kj}\}$ denote the training set available for class j, and let C_j comprise T_j samples. Let $C = C_1 \cup \ldots \cup C_M$ denote the pooled labeled

training set. The appropriate likelihood function for C is derived in [14] and is given by

$$L(C|\lambda) = \prod_{j=1}^{M} \prod_{k=1}^{T_j} \alpha_j g_j(X_{kj}|\lambda_j), \tag{1}$$

where $g_j(X|\lambda_j)$ denotes the PDF for class j, α_j denotes the *a priori* probability of class j, λ_j denotes the parameters defining the PDF of class j, and $\lambda = \lambda_1 \cup \ldots \cup \lambda_M$. Because $g_j(X|\lambda_j)$ is a homoscedastic mixture Gaussian PDF it takes the form

$$g_j(X|\lambda_j) = \sum_{i=1}^{G_j} \pi_{ij} N(X, \mu_{ij}, \Sigma_{kernel}), \tag{2}$$

where G_j is the number of Gaussian mixture components in class j, π_{ij} is the mixing proportion of the i^{th} Gaussian component in class j, $N(\cdot)$ represents a normal density function, μ_{ij} is the mean vector of the i^{th} component in class j, and Σ_{kernel} is the covariance matrix common to all class PDF mixture components. Consequently, the parameter sets to be trained are given by

$$\lambda \equiv \bigcup_{j=1}^{M} \{\alpha_j, \lambda_j\} \text{ where } \lambda_j \equiv \bigcup_{j=1}^{G_j} \{\pi_{ij}, \mu_{ij}, \Sigma_{kernel}\}. \tag{3}$$

Training SPNN is equivalent to estimating the parameter vector λ using maximum likelihood methods. The ML algorithm for SPNN is derived using the Expectation-Maximization (EM) method. The details are provided in [14], together with references to the general literature.

The ML estimates of the *a priori* class probabilities are given by

$$\alpha_j = \frac{T_j}{T_1 + T_2 \ldots + T_M}. \tag{4}$$

Note that no iteration is required by the estimates in equation (4). This expression implicitly conveys a very important message for exemplar (data) screening because it shows that the *a priori* class probabilities are proportional to their representation in the pooled training set. If the exemplars are so heavily screened that proportional representation is inadequate, the

estimates in equation (4) should be neglected and *a priori* probabilities estimated in some other way. The iteration for the remaining parameters is independent of the parameters $\{\alpha_j\}$. The details of the recursions for π_{ij}, μ_{ij}, and Σ_{kernel} are also given in [14].

Smoothed Version of Fisher's Method

A variation of Fisher's method, referred to here as "Smoothed Fisher," is used to derive the FPS. The reason for the word smoothed is simply that SPNN is used to estimate class PDFs from the available feature vector samples, and the exact expressions for the parameters (mean vectors and covariance matrices) of these estimated class PDFs are then used in Fisher's method. In contrast, the usual formulation of Fisher's method uses the class sample means and (pooled) within-class sample covariance matrix. Smoothing the feature vector sets using SPNN reduces the effects of outliers on the FPS.

After ML training, SPNN yields an estimated mixture Gaussian PDF for each class. It is readily shown that the expression for the mean, $\overline{X}_j$, of the j^{th} estimated class PDF, $g_j(X|\lambda_j)$, is given by

$$\overline{X}_j = \sum_{i=1}^{G_j} \pi_{ij}\mu_{ij}, \tag{5}$$

and that the expression for the covariance matrix of $g_j(X|\lambda_j)$ is

$$\Sigma_j = \Sigma_{kernel} + \sum_{i=1}^{G_j} \pi_{ij}(\mu_{ij} - \overline{X}_j)(\mu_{ij} - \overline{X}_j)^t, \tag{6}$$

where π_{ij}, μ_{ij}, and Σ_{kernel} are estimated by SPNN training. Similarly, the estimated PDF, $L(X|\lambda)$, of the pooled training set is the "mixture of mixtures" given by

$$L(X|\lambda) = \sum_{j=1}^{M} \alpha_j \sum_{i=1}^{G_j} \pi_{ij} N(X, \mu_{ij}, \Sigma_{kernel}). \tag{7}$$

The expression for the pooled training set's mean is

$$\overline{X}_{pool} = \sum_{j=1}^{M} \alpha_j \sum_{i=1}^{G_j} \pi_{ij} \mu_{ij}, \tag{8}$$

and its covariance matrix is

$$\begin{aligned} \Sigma_{pool} &= \Sigma_{kernel} + \sum_{j=1}^{M} \alpha_j \sum_{i=1}^{G_j} \pi_{ij} (\mu_{ij} - \overline{X}_j)(\mu_{ij} - \overline{X}_j)^t + \sum_{j=1}^{M} \alpha_j (\overline{X}_j - \overline{X}_{pool})(\overline{X}_j - \overline{X}_{pool})^t \\ &\equiv \Sigma_{kernel} + \Sigma_{components} + \Sigma_{classes}. \end{aligned} \tag{9}$$

From the previous equation, it is seen that there are three major contributors to the "pooled" or global covariance matrix. From the discussion in [4], the sum of the first two terms, Σ_{kernel} and $\Sigma_{components}$, is a measure of the "within-class" scatter. The third term, $\Sigma_{classes}$, is a measure of the "between-class" scatter and will be referred to as the "spread-of-the-means" matrix. By inspection of the last equation it is clear (c.f., [4]) that maximizing the between-class to within-class "smoothed" variations requires maximizing the Rayleigh quotient given by

$$J(w) = \frac{w^t \Sigma_{classes} w}{w^t (\Sigma_{kernel} + \Sigma_{components}) w} \tag{10}$$

where $w \in R^n$. The quotient, $J(w)$, is the same as the formula given by Duda and Hart [4, Section 4.11] except that the smoothed mean and covariance matrix estimates given by equations 5, 6, 8, and 9 replace sample means and covariance matrices. Maximizing the Rayleigh quotient, $J(w)$, is equivalent to solving the following generalized eigenproblem,

$$\Sigma_{classes} w = \lambda (\Sigma_{kernel} + \Sigma_{components}) w. \tag{11}$$

The matrix $\Sigma_{kernel} + \Sigma_{components}$ is full rank because SPNN provides a full rank estimate of Σ_{kernel}. Consequently, the solution of the generalized eigenproblem (11) poses no conceptual

difficulties; however, the numerical solution should be carried out using the QR algorithm [15, Chapter 9, page 9.10] to avoid the numerical ill-conditioning that typically accompanies covariance matrix formation. Because SPNN is coded using the QR algorithm, equation (11) is solved without the need to form covariance matrices directly.

Fisher's feature reduction method cannot yield more than M-1 derived features, due to the rank of $\Sigma_{classes}$. The "spread-of-the-means" matrix, $\Sigma_{classes}$, in (9) has at most rank M because it is the sum of M outer products. However, one degree of freedom is lost because the global mean, $\overline{X}_{pool}$, is estimated; hence, the rank of $\Sigma_{classes}$ is at most M-1. Therefore, there are at most M-1 nonzero eigenvalues of the generalized eigenproblem (11). The span of the eigenvectors corresponding to the largest k, $1 \leq k \leq M-1$, nonzero eigenvalues is the Fisher projection space of dimension k, denoted FPS(k). The rank of the FPS(k) is exactly k because the eigenvectors spanning FPS(k) are linearly independent. If the context is clear, FPS(k) will be written simply as FPS.

Procrustes Ordering

The Procrustes ordering of the feature set is defined for every FPS(k), for $k = 1, 2, \ldots, (M-1)$. The ordering of most interest is the one resulting from the largest dimension $FPS \equiv FPS(M-1)$. The cosine of the angle, ϕ, between an arbitrarily specified nonzero vector, $x \in R^n$, and the FPS(k) can be defined relative to the original coordinate axes or the coordinate axes defined by the FPS(k). The two methods differ by a linear transformation, L^t, where L is the Cholesky factor of the "within-class" scatter matrix (see appendix). The difference is important for determining a null hypothesis for significance testing. For this

reason the angle relative to the FPS was chosen and is defined by

$$\phi_j = \cos^{-1}\left\{\frac{\| \tilde{W}^t L^t x_j \|_2}{\| L^t x_j \|_2}\right\}, \tag{12}$$

where $\tilde{W}$ is a matrix whose columns are the nonzero eigenvectors that define the FPS. The angle, ϕ, is uniquely defined if it is restricted to lie between 0 and 90 degrees, and it is the same angle for all vectors in the subspace spanned by x. When x is taken to be the j^{th} feature, that is, $x = f_j \equiv (0,\ldots,0,1,0,\ldots,0)^t$, the angle is defined to be the Procrustes angle, ϕ_j, of the j^{th} feature. The dependence of ϕ_j on the dimension k of the FPS is suppressed in the notation. A depiction of the geometry of the Procrustes angle is given in figure 1. As is clear from this figure, the Procrustes angle is related to least squares approximation. The least squares approximation to the j^{th} feature vector, f_j, is the orthogonal projection of f_j onto the FPS(k), and the Procrustes angle is the angle between f_j and its orthogonal projection.

The Procrustes ordering of the feature set is defined by ranking the features by increasing numerical size of their Procrustes angles. The first feature in the Procrustes ordering, therefore, has the smallest Procrustes angle, and the last feature has the largest angle. Intuitively, features with "small" Procrustes angles are "nearly" in the linear span of the FPS, and features with "large" angles are "nearly" orthogonal to the FPS. Because the FPS is a good space for feature reduction, it is natural to think that features with small Procrustes angles are "better" for classification than features with larger angles.

One way to decide whether or not the Procrustes angle of a given feature is significant is to apply a statistical significance test. In a significance test, one is, in effect, testing a single hypothesis against all other hypotheses, with no particular alternatives in mind. To formulate an appropriate hypothesis, consider the process of feature generation. In most

complex classification problems, unique class characteristics are unknown; therefore, it is left to the feature set designer to determine the set of features that captures the class differences. The resultant feature set will be a union of two subsets; the *knowledge-based* set and the *intuition-based* set. The knowledge-based feature set is defined to be those features that are derived from known measurable class differences, whereas the set of intuition-based features is the "intelligent guess" set. For most complex classification problems the cardinality of the knowledge-based set is small compared to that of the intuition-based set; therefore, the underlying model should be dominated by the intuition-based set. Because Procrustes ordering is independent of feature vector length, the model adopted in this report is that the feature vectors are random with a uniform distribution on the unit sphere in R^n. The feature evaluation/selection process then becomes the process of determining the subset of these "randomly" generated features that "happen" to best approximate the FPS. In keeping with the idea of Procrustes, the set of features that approximates this subspace is the set determined by those features with the smallest angle with that space. Assuming this is an accurate model of the feature generation process, thresholding the upper tail of the resulting PDF will enumerate those features that are poor for classification.

Denote the PDF of the Procrustes angle, ϕ, between a fixed k dimensional subspace of R^n and a uniformly distributed random variable on the unit sphere in R^n, by $P_{n,k}(\phi)$. It can be shown [16] that the random variable $t \equiv cos^2\phi$ is beta distributed, with parameters $\frac{n}{2}$ and $\frac{(n-k)}{2}$, so that after a change of variables $P_{n,k}(\phi)$ is given explicitly by

$$P_{n,k}(\phi) = \frac{2\Gamma(\frac{n}{2})}{\Gamma(\frac{k}{2})\Gamma(\frac{(n-k)}{2})} \cos^{k-1}\phi \sin^{n-k-1}\phi, \tag{13}$$

where $0 \leq \phi \leq \frac{\pi}{2}$. Therefore, under the above feature generation model, a feature is

considered to be poor for classification at a significant level β if it lies on the upper β% tail of $P_{n,k}(\phi)$. A plot of $P_{n,k}(\phi)$ for the example considered in this report is given in figure 2. As is clear from figure 2, for these particular choices of n and k, (n = 70, k = M-1 = 10), the PDF is approximately Gaussian. Unfortunately, for the example in this report, the data was no longer available at the time the model for the hypothesis was completed. Although as an academic exercise, simple synthetic problems were generated to confirm the utility of this hypothesis, the practical utility of this test can only be determined by its application to real problems.

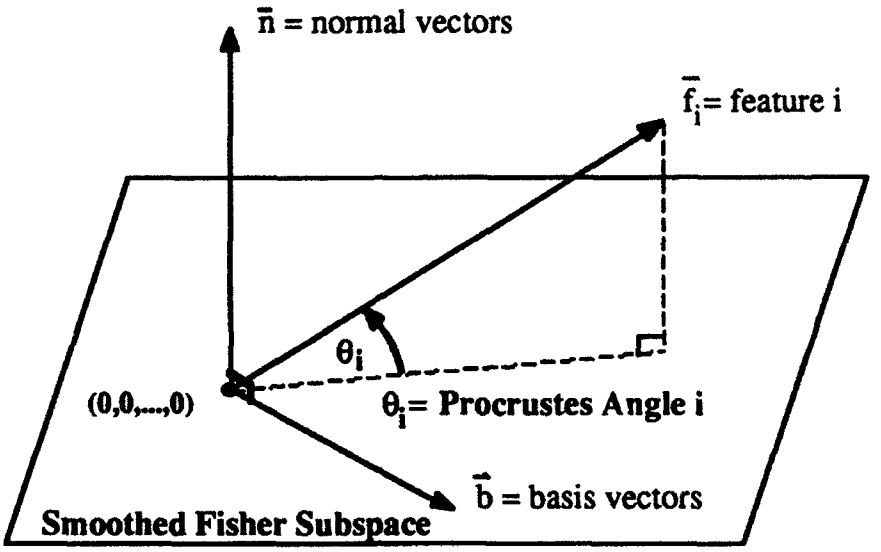


Figure 1. Geometric Interpretation of Procrustes Angles

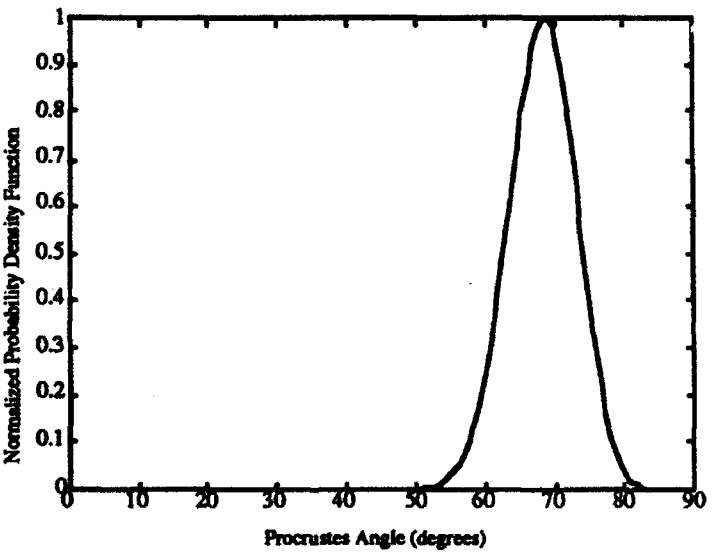


Figure 2. PDF of the Procrustes Angles (n=70, k=10)

APPLICATION OF PROCRUSTES ORDERING TECHNIQUE

This section describes the application of Procrustes linear ordering to a multiclass, multidimensional, acoustic signal classification problem. A comparison of the classification results and the computational requirements associated with each of the methods is analyzed. The data set is comprised of eleven event classes on which 70 feature measurements were available. There were a total of 249 exemplars available for this experiment. The distribution of class training and testing exemplars is provided in table 1.

Table 1: Distribution of Training and Testing Exemplars

Class Number	Training Samples	Testing Samples
1	16	15
2	15	14
3	21	21
4	15	15
5	10	10
6	10	09
7	10	09
8	06	06
9	04	04
10	07	06
11	13	13
Total	127	122

Statistical Hold-Out Study

As discussed previously (refer to Feature Reduction and Interpretation section), small data sets often pose severe difficulties on automatic classification. Additionally, when exploring the utility of new techniques, or measuring the performance of existing algorithms, careful consideration must be given to the conclusions that can be drawn from small data sets. That is, given the reality of limited data, every effort must be made to statistically quantify the results and to resist the temptation to generalize these conclusions beyond those supported by the data. There are a number of accepted resampling techniques that attempt to extend the utility of small data sets by exploiting the variability of different subsets of the data. One such method is that of performing "hold-out" studies; this technique is used to assess the performance of the linear ordering feature reduction methods considered in this report.

Typically, .he data is divided into disjoint training and testing sets by randomly sampling the original data according to a uniform distribution. A trial is defined as the assessment of the classifier performance based on a training and testing set pair. By resampling the data, a number of trials are generated and an average performance is observed. The purpose of this hold-out procedure is to reduce the bias and variance associated with performance estimates based on a small data set, and therefore, provide a better method of comparing reduction techniques.

Feature Selection Across Multiple Trials

A high level diagram of the test methodology that was used to perform the multiple trials for Smoothed Fisher and Procrustes is provided in figure 3. The performance curves for the Procrustes ordering, Smoothed Fisher, and SFCPO are plotted in figures 4, 5 and 6, respectively. Figure 7 shows the relative performance of all the techniques considered in this report. These plots resulted from twenty independent trials of different training and test sets derived from the original data. The output of each trial was a prioritized list of the candidate features. In general, the list changed as a function of the data used for each trial. The underlying motivation behind these reduction techniques is to select the subset of features that provides the most robust discrimination capability. Therefore, reconciling these conflicting linear orderings is necessary to obtain the desired subset of features.

The first step in addressing this issue is to examine the consistency of the orderings across trials. For this part of the study, 100 trials were examined. Intuitively, the number of times a particular feature is highly ranked (i.e., has a small Procrustes angle) across trials should be an indication of the relative importance of this feature. Histograms were plotted to display the features that were consistently ranked in the top m (m=1, 2, ... 35) positions. For m=1, feature 23 occurred 98 times which suggests that this feature is always important. The number of "active" features (i.e., features that appear at least once in the top m rankings) increases nonlinearly with m. For m=1, there are only 2 active features; however, 27 features become active for m=5. Figures 8(a), 8(c) and 8(e) are histograms for values of m=12, 25, and 35, respectively. Note that for m=12, approximately 90% of the features occurred at least once over the 100 trials. This result is somewhat surprising; however, it supports the

notion of increasing the utility of the data set by resampling to exploit the variability within the data.

Of the active features, those that appear consistently in the top m positions are thought to be of most importance for classification. To examine this, a thresholded version of the histogram is constructed. This is accomplished by varying a threshold between 0 and the total number of trials (100 in this example) and simply counting the number of times the threshold is exceeded on the histogram. Figures 8(b), 8(d), and 8(f) are the thresholded versions of the histograms in figures 8(a), 8(c) and 8(e), respectively. Recall from figure 4 that the peak of the classification curve based on the Procrustes ordering occurred for 26 features; however, because of the variability introduced by the application of multiple trials, this number is only an estimate. It was hoped that because 90% of the features are active in the top 12 positions, thresholding the histogram would enumerate a set of desirable features on the order of 26. From figure 8(a) and 8(b), it is clear that although the features are adequately represented, a distinct feature subset is not observable. Increasing m to 25, we notice the original feature set is now divided into two subsets. This is indicated by the flat portion of the curve in figure 8(d). Because we currently lack the theoretical tools to support the significance of this apparent "breakpoint," we looked for confirmation by examining the case for m=35. Figures 8(e) and 8(f) show that 23 features occur 98% of the time. This flat characteristic, which is present at roughly the same number of features that lead to the maximum classification performance, appears to point to the "breakpoint" between features. This breakpoint defines features that are important for classification and those that can be ignored for this data set. Additionally, the fact that the flat portion of the curve occurs over a wide range of threshold values (between 40 and 65 for m=25, and between 88 and 98 for

m=35) supports the idea of a distinct separation between the feature subsets.

The 23 "best" features can be identified from figure 8(e) as those features that were ranked in the top 35 positions greater than 90 times in the 100 trials. The feature indices are as follows: 1-8, 10, 11, 17, 22-26, 28, 29, 37, 41, 42, 44, and 45. The average classification performance based on this feature set for 20 trials is 86.6% with a standard deviation of 2.37 (86.5% +/- 2.74 for 50 trials). Comparing this to the performance based on the Procrustes ordering for the entire 70 dimensional feature set and 20 trials (85.8% +/- 2.08), we see an insignificant difference in the mean performance level and only a slight increase in the variance. A small increase in variance is a reasonable tradeoff for a reduction in feature set size from 70 to 23 dimensions. This favorable comparison validates the multi-trial, Procrustes based, feature selection process.

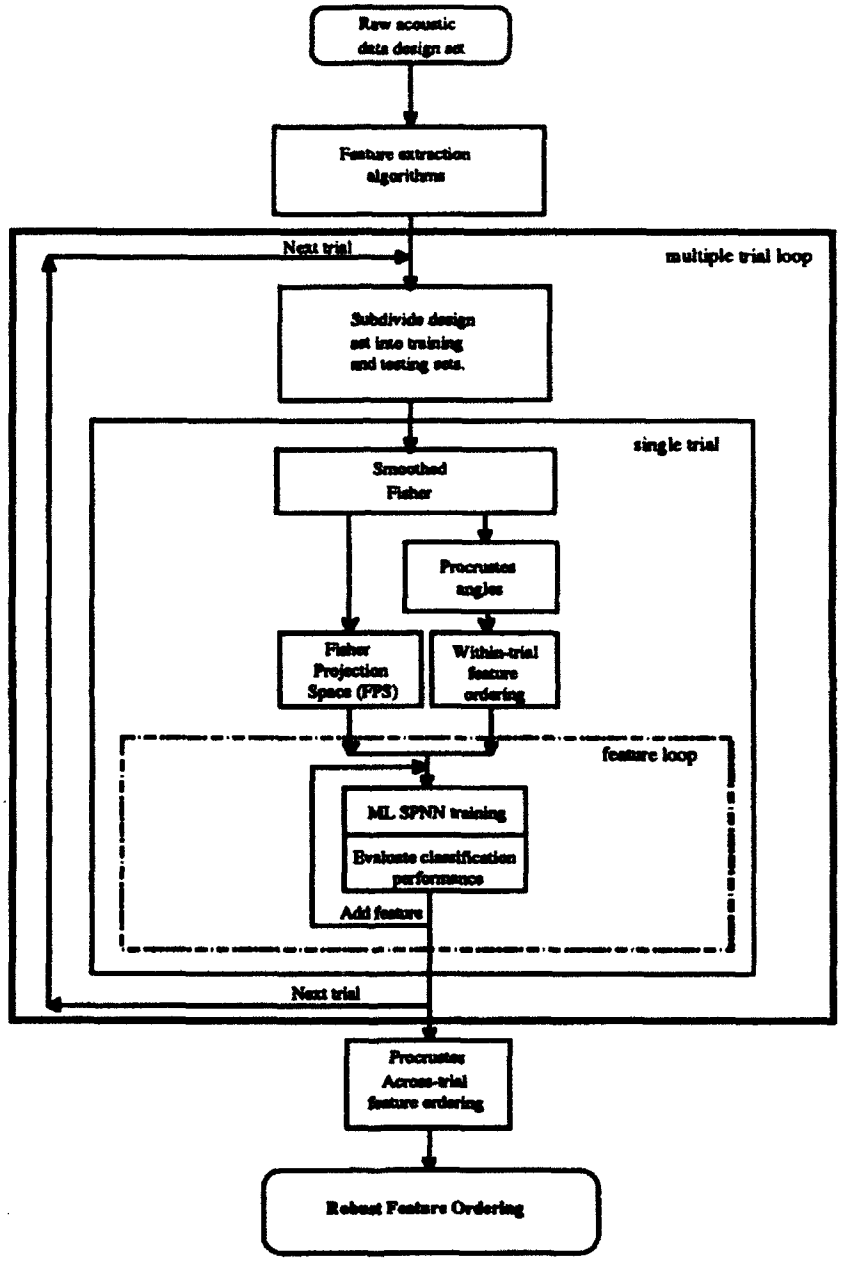


Figure 3. Test Methodology for New Reduction Techniques (Smoothed Fisher, Procrustes)

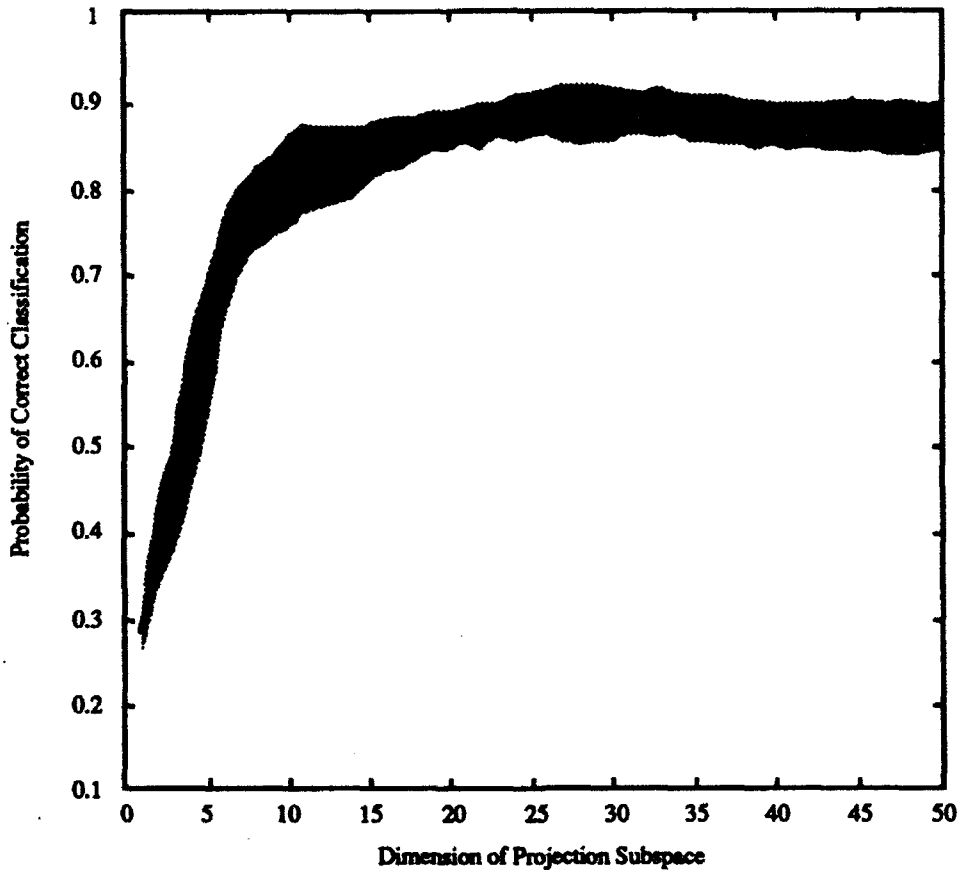


Figure 4. Average Probability $(+/-1\sigma)$ of Correct Classification vs. Projection Subspace Dimension Using Procrustes Ordering

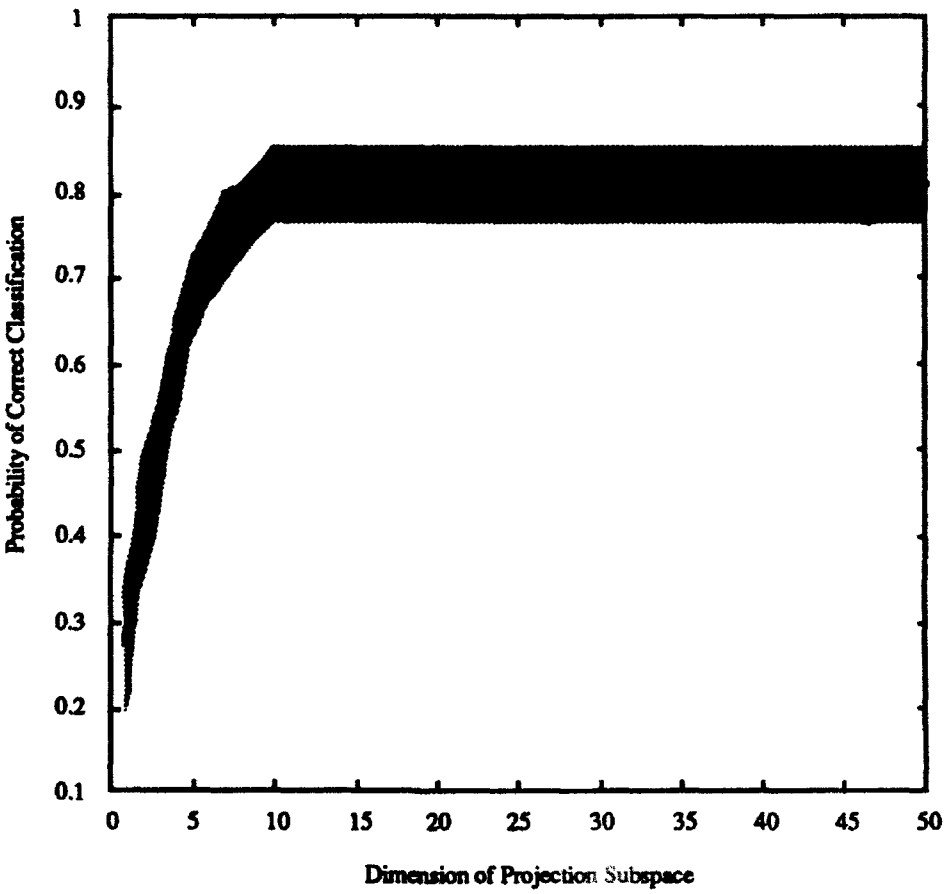


Figure 5. Average Probability $(+/-1\sigma)$ of Correct Classification vs. Projection Subspace Dimension Using Smoothed Fisher Ordering

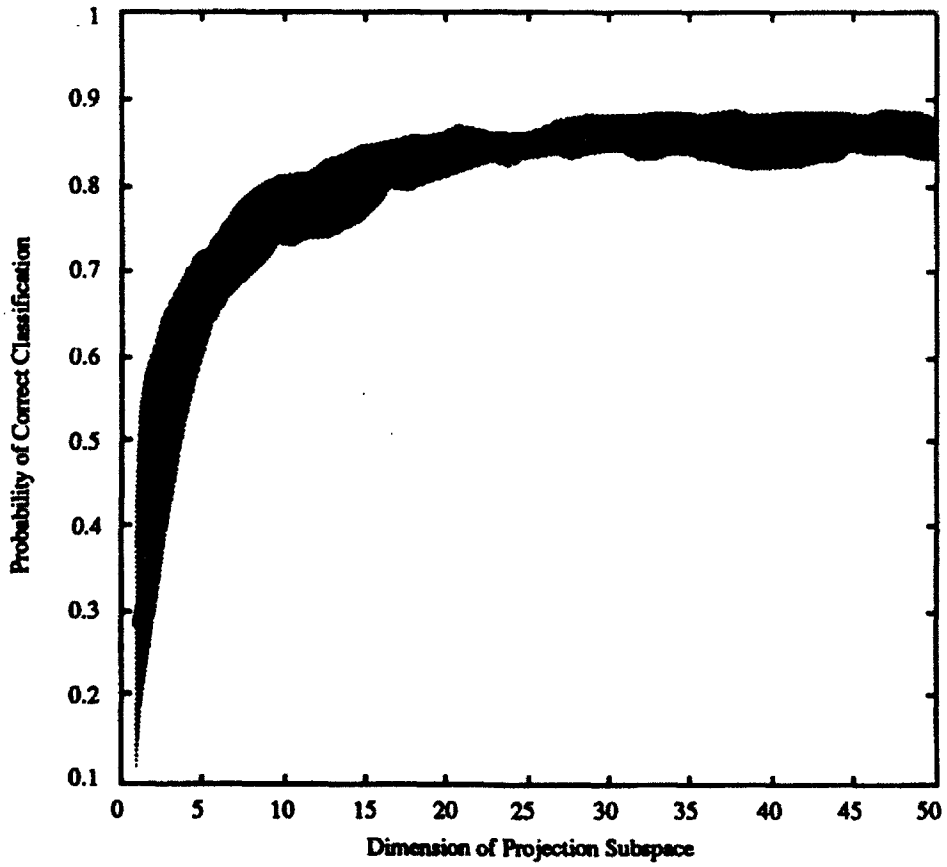


Figure 6. Average Probability $(+/-1\sigma)$ of Correct Classification vs. Projection Subspace Dimension Using SFCPO

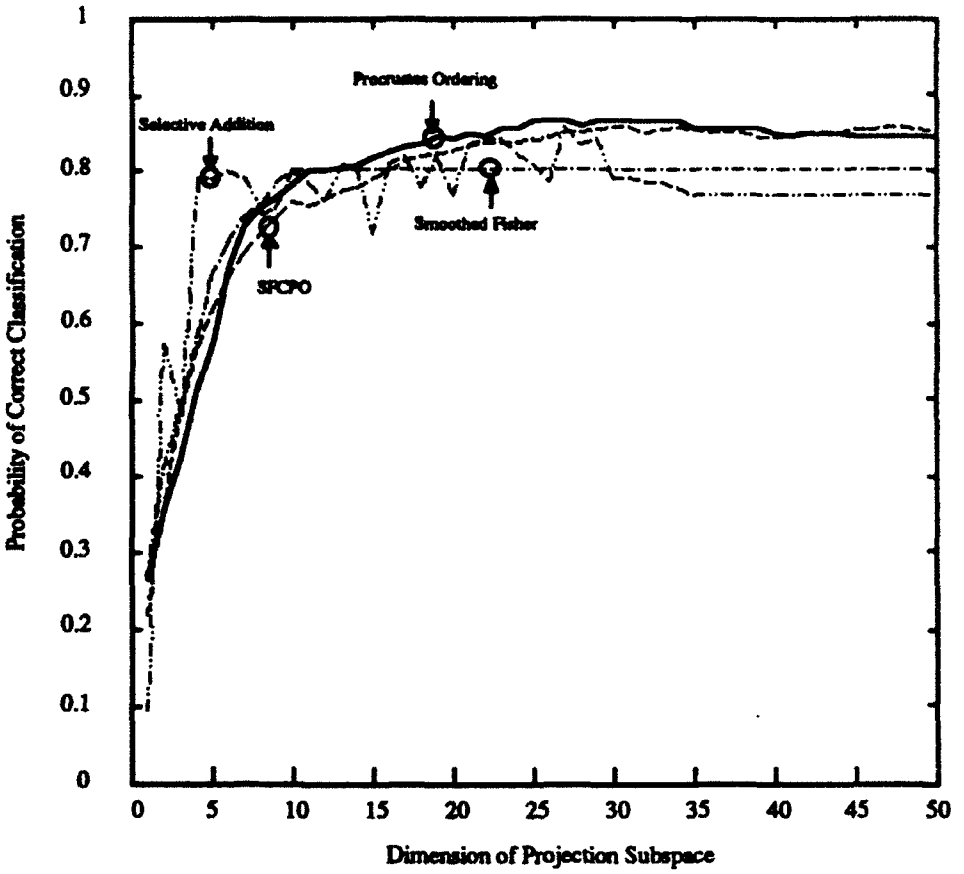


Figure 7. Comparison of Average Probability of Correct Classification vs. Projection Subspace Dimension for all Methods

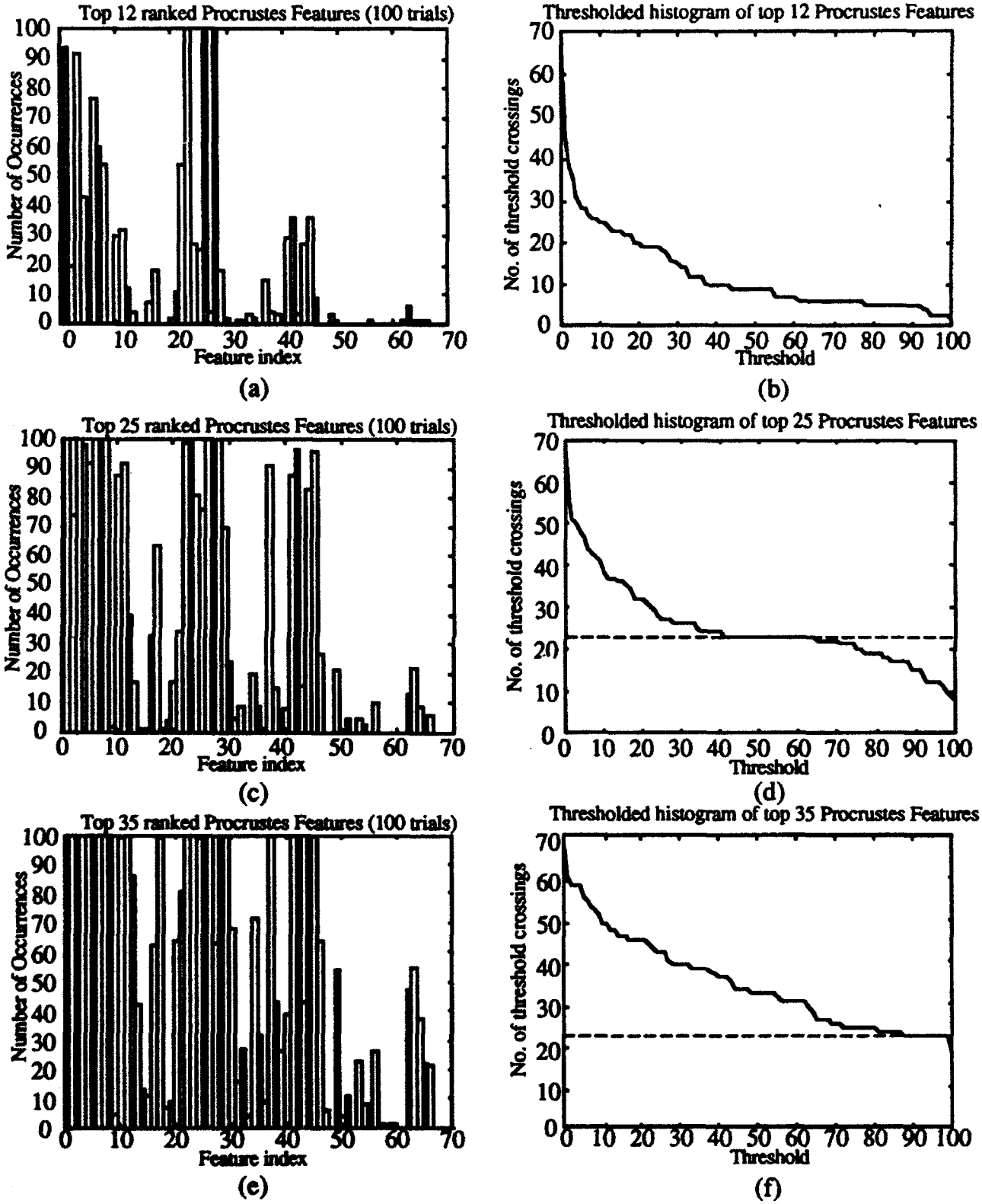


Figure 8. Histograms and Thresholded Histograms of Top 12, 25 and 35 Ranked Procrustes Features

Computational Complexity

As presented earlier, one of the most desirable properties of any feature reduction technique is low computational complexity. Table 2 presents the measured (and predicted) times associated with the three different stages of the feature reduction process. These stages are as follows:

1. Prioritization - which refers to providing an ordering of features based on the performance from a single, 70 dimensional trial.

2. Evaluation - which involves the generation of a performance curve based on linear combinations of the prioritized features to determine the subset that provides the best performance. In other words, the prioritized order is sequentially tested (i.e., feature rankings (1), (1,2),... (1,2,3,... k),... (1,2,3,... 70)) and the performance is plotted as a function of k, the feature index.

3. Statistical analysis - which involves performing multiple trials. In this example, 20 trials were performed.

The times associated with Smoothed Fisher are not as impressive as they may initially seem because the computational complexity increases as the cube of the feature dimension size, n, and since n is bounded by the number of classes minus one for Fisher, this result is misleading. Also, note that since Fisher forms a linear combination of all the features, a prioritized ranking of individual features is not available. The times associated with the SFCPO require each of the features to be evaluated independently to determine the prioritization. This difference, although present, is negligible for the evaluation stage (300 minutes

vs. 310 minutes). The times associated with the SAM were estimated from the computation for only 35 features. For the SAM, the result of the prioritization stage is also the final evaluation; therefore, a time of zero was recorded in the table. Based on this discussion, and the results in table 2, Procrustes provides the best performance at the lowest computational cost.

Table 2. Computational Comparison

Reduction Technique	Computational Time Required		Statistical Analysis
	Prioritization	Evaluation	
Smoothed Fisher	NA	12 min	4 hrs
SFCPO	10 min	5 hrs	99 hrs
SAM (estimated)	48 hrs	0	80 days
Procrustes	30 sec	5 hrs	96 hrs

CONCLUSIONS AND RECOMMENDATIONS

Procrustes ordering of the feature set is proposed as a feature reduction and interpretation method. Procrustes ordering is new and has not been previously proposed or studied in the pattern recognition and classification literature. Consequently, the central investigation in this report focuses on the effectiveness of Procrustes ordering for feature selection for nonidealized problems with real data. The conclusion of this investigation is that Procrustes ordering of the feature set significantly outperforms the commonly used and accepted alternative linear ordering, SFCPO, on the 11 class, 70 feature example presented in this report. This conclusion is considered statistically significant because this investigation is based on extensive and careful statistical trials using a hold-out methodology.

Procrustes ordering is defined in terms of the Procrustes angles between the features and the classical multiclass FPS. Because of the strong geometrical and analytic character of this relationship, Procrustes ordering is a natural extension of and complement to the fundamental ideas of the FPS. A significance test of the Procrustes angles based on a feature generation model was proposed. Unfortunately, because the original data was lost, this significance test was not applied to the example presented in this report.

The utility of Procrustes ordering for nonidealized real data is established only for the example presented. To establish that Procrustes ordering and the significance test are widely useful, the performance of the algorithm must be studied statistically in many different problems of considerable variation in character, size, and application domain. It is recommended that additional studies of Procrustes ordering be undertaken.

The statistical methodology proposed in the example section is an experimental method

of determining fixed feature set selection for a multi-trial statistical study. The difficulty is that each trial produces an ordered feature set, and the ordering varies from trial to trial. Reconciling these feature set orderings to find an effective fixed feature set is a subtle and easily underestimated task. The experimental methodology proposed for the example is intended to facilitate this task; however, it is not based on a theoretical statistical model. The experimental methodology seems sound and sensible, and it suggests that interesting theoretical models can be developed that will support the methodology. Unfortunately, theoretical models of this sort are unknown to the authors. It is recommended that a theoretical study of the experimental statistical methodology be undertaken. Such a study could develop useful analytical tools for the general feature reduction problem and would be applicable to any hold-out study resulting in conflicting feature orderings.

Finally, it is recommended that Procrustes ordering be studied in conjunction with SPNN and the smoothed FPS. Procrustes ordering is compatible with SPNN, as the discussion shows. Such an investigation should encompass a statistical hold-out trial methodology, as was done in the example, and should address the problems associated with fixed feature set selection from multiple trials, model order selection, and data poverty. These issues cause multiple and conflicting effects, and untangling them all poses interesting practical and theoretical problems.

APPENDIX

Derivation of Procrustes Angle

Given the generalized eigenproblem

$$Aw = \lambda Bw, \tag{A-1}$$

where B is $n \times n$, positive definite, symmetric matrix, form the Cholesky decomposition of B, $B = LL^t$, and substitute

$$Aw = \lambda LL^t w$$

$$AL^{-t}(L^t w) = \lambda L(L^t w). \tag{A-2}$$

If we define $y = L^t w$ (i.e., forward transform or rotate the original eigenvectors) and $C \equiv L^{-1}AL^{-t}$, the result is the familiar eigenproblem given by

$$Cy = \lambda y. \tag{A-3}$$

Compute the singular value decomposition of C, $C = U\Sigma V^t$. Note that the eigenvectors of C are the columns of V. Let W_i denote column i of V. Suppose $p \geq 1$ singular values are $\neq 0$. Define the $n \times p$ matrix

$$\tilde{W} = [W_1 W_2 \ldots W_p] \in R^{n \times p}. \tag{A-4}$$

Note that the $p \times p$ matrix, $\tilde{W}^t\tilde{W}$, is the identity matrix, $I^{p \times p}$, because the columns of $\tilde{W}$ are orthonormal.

At this point, the problem of finding the Procrustes angles involves projecting the original features, $x_i \equiv e_i \in R^{n\times 1}$, where e_i denotes the standard basis, onto the p-dimensional Fisher projection subspace and measuring the angle between the original feature and the projection. First, to be consistent, the original features must be forward transformed, since it was necessary to "forward transform" the eigenvectors, w, to solve the generalized eigenproblem. This is given by

$$\tilde{x}_i = L^t x_i. \tag{A-5}$$

Using a least squares approach (see [17], pages 106-107), the projection of $\tilde{x}_i$ onto the column space $\tilde{W}$ is given by

$$proj_{\tilde{W}}\tilde{x}_i = \tilde{W}(\tilde{W}^t\tilde{W})^{-1}\tilde{W}^t\tilde{x}_i. \tag{A-6}$$

However, since the matrix, $\tilde{W}^{n\times p}$ is orthogonal, $\tilde{W}^t\tilde{W} = I$ and the projection reduces to

$$\begin{aligned} proj_{\tilde{W}}\tilde{x}_i &= \tilde{W}\tilde{W}^t\tilde{x}_i \\ &= \tilde{W}\tilde{W}^t L^t x_i. \end{aligned} \tag{A-7}$$

The angle between any two vectors, a and b, is given by

$$\cos\phi = \frac{a^t b}{\| a \|_2 \| b \|_2}. \tag{A-8}$$

Let $a = \tilde{x}_i$ and $b = \tilde{W}\tilde{W}^t\tilde{x}_i$ then

$$\begin{aligned} \| a \|_2 &= (\tilde{x}_i^t\tilde{x}_i)^{\frac{1}{2}} \\ &= ((L^t x_i)^t(L^t x_i))^{\frac{1}{2}} \\ &= \| L^t x_i \|_2 \end{aligned} \tag{A-9}$$

and

$$\begin{aligned} \| b \|_2 &= ((\tilde{W}\tilde{W}^t\tilde{x}_i)^t(\tilde{W}\tilde{W}^t\tilde{x}_i))^{\frac{1}{2}} \\ &= (\tilde{x}_i^t\tilde{W}\tilde{W}^t\tilde{W}\tilde{W}^t\tilde{x}_i)^{\frac{1}{2}} \\ &= (\tilde{x}_i^t\tilde{W}\tilde{W}^t\tilde{x}_i)^{\frac{1}{2}} \\ &= \| \tilde{W}^t L^t x_i \|_2 . \end{aligned} \tag{A-10}$$

Substituting into equation (A-8) gives the expression for the Procrustes angle

$$\phi_i = \cos^{-1}\left\{\frac{\| \tilde{W}^t L^t x_i \|_2}{\| L^t x_i \|_2}\right\}. \tag{A-11}$$

REFERENCES

[1] D. H. Foley, *Considerations of Sample Size and Feature Size*, IEEE Transactions on Information Theory, vol. IT-18, 1972, 618-626.

[2] A. K. Jain and W. G. Waller, *On the Optimal Number of Features in the Classification of Multivariate Gaussian Data*, Pattern Recognition, Vol. 10, 1978, 365-374.

[3] R. L. Streit, *An Upper Bound on Feature Vector Dimension as a Function of Design Set Size For Two Gaussian Populations*, NUWC-NL Technical Memorandum No. 921048, 10 March 1992.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley & Sons, New York, 1973.

[5] P. M. Narendra and K. Fukanaga, *A Branch and Bound Algorithm for Features Subset Selection*, IEEE Transactions on Computers, Vol. C-26, No. 9, September 1977, 917-922.

[6] A. N. Mucciardi and E. E. Gose, *A Comparison of Seven Techniques for Choosing Subsets of Pattern Recognition Properties*, IEEE Transactions Computer, Vol. C-20, September 1971, 1023-1031.

[7] J. Kittler and P. C. Young, *A New Approach to Feature Selection Based on the Karhunen-Loeve Expansion*, Pattern Recognition, Vol. 5, December 1973, 335-352.

[8] T. M. Cover and J. M. Van Campenhout, *On the Possible Orderings in the Measurement Selection Problem*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, September 1977, 657-661.

[9] A. K. Jain and R. Dubes, *Feature Definition in Pattern Recognition With Small Sample Size*, Pattern Recognition, Vol. 10, 1978, 85-96.

[10] C. Chang, *Dynamic Programming as Applied to Feature Subset Selection in a Pattern Recognition System*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 2, March 1973, 166-171.

[11] F. Z. Brill, D. E. Brown, and W. N. Martin, *Fast Genetic Selection of Features for Neural Network Classifiers*, IEEE Transactions on Neural Networks, Vol. 3, No. 2, March 1992, 324-328.

[12] D. F. Specht, *Probabilistic Neural Networks for Classification, Mapping, or Associative Memory*,in Proceedings of the IEEE International Conference on Neural Networks, Vol. I, San Diego, California, July 24-27, 1988, 525 - 532.

[13] R. L. Streit, *A Neural Network for Optimum Neyman-Pearson Classification*, in Proceedings of the International Joint Conference on Neural Networks, Vol. I, June 17-21,1990, San Diego, California, sponsored by IEEE and Neural Network Society, 685 - 690.

[14] R. L. Streit and T. E. Luginbuhl, *Maximum Likelihood Training of Probabilistic Neural Networks*, to be published in IEEE Transactions on Neural Networks. (See also NUSC-NL Technical Memorandum No. 911277, 30 December 1991).

[15] J. J. Dongarra, J. R. Bunch, C. B. Moler, G. W. Stewart, *LINPACK Users' Manual*, Society of Industrial and Applied Mathematics, Philadelphia, 1979.

[16] R. A. Vitali, *Personal Communication*, Department of Statistics, University of Connecticut, Storrs, 7 September 1992.

[17] G. Strang, *Linear Algebra and its Applications*, Academic Press, Inc., New York, 1976.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Office of Naval Research 321W (T. G. Goldsberry, L. Jacobi, N. Harned)	3
Defence Science and Technology Office, Sonar Sys. Div. (Dr. Ross Barrett)	1
Defence Science and Technology Office, Radar Tech. and Sys. (Dr. Douglas Kewley)	1
Cooperative Research Centre for Sensor Signal and Information Processing (Dr. Douglas Gray)	1
ARL/University of Texas (Dr. Gary Wilson)	1
ORICON (Dr. Joseph Presley)	1
Tracor Applied Sciences (Dr. Thomas Leih)	1
Nichols Research Corp. (Dr. Leonid Perlovsky)	1
Norden Systems (Dr. Israel Metal)	1
APL/Pennsylvania State University (Dr. David Hall)	1
Raytheon Corporation (Dr. Stanley Chamberlain)	1
University of Connecticut (Dr. R. Vitali)	1
University of Massachusetts, Dartmouth (Dr. C. Chen)	1
MIT Lincoln Labs. (Dr. Richard Lippmann)	1
Harry L. Hurd Associates, Inc. (Dr. Harry Hurd)	1
Naval Submarine Base (Dr. Thomas Hanna)	1
Science Applications International Corp. (Dr. Charles Katz)	1
DTIC	2