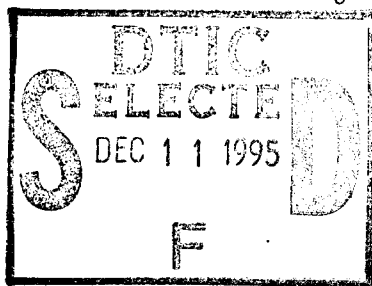


Serial No. 502,741

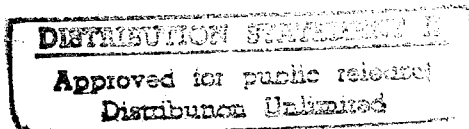
Filing Date 30 June 1995

Inventor Stephen G. Greineder
Tod E. Luginbuhl
Roy L. Streit



NOTICE

The above identified patent application is available for licensing. Requests for information should be addressed to:



OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
CODE OCCC3
ARLINGTON VA 22217-5660

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

19951208 026

DTIC QUALITY INSPECTED 1

30 Jun 95

2
3 SYSTEM AND METHOD FOR FEATURE SET REDUCTION

4
5 STATEMENT OF GOVERNMENT INTEREST

6 The invention described herein may be manufactured and used
7 by or for the Government of the United States of America for
8 governmental purposes without the payment of any royalties
9 thereon or therefor.

10
11 BACKGROUND OF THE INVENTION

12 (1) Field of the Invention

13 The present invention relates to a system and method for
14 feature reduction and interpretation for pattern recognition
15 systems. More specifically, the invention provides a system for
16 ranking features in order of importance and for selecting the
17 features that are important for classification.

18 (2) Description of the Prior Art

19 The use of automatic pattern recognition systems for rapid
20 identification and analysis of patterns in input data and for
21 accurate classification of input patterns into one of several
22 predetermined classes is well known in the art. Feature based
23 pattern recognition systems typically use an array of values or
24 measurements defining properties of the input pattern called a
25 feature vector. An input feature vector is compared to a
26 reference set of feature vectors representing the known classes

1 to determine which of the known class feature vectors has the
2 highest similarity to the input feature vector.

3 When the different event classes have known unique
4 measurable characteristics and features, the classification
5 problem is straightforward. However, for many applications the
6 characteristics of the classes and features that separate the
7 classes are unknown and the feature set designer must determine
8 the features that capture the class differences. This set of
9 available features is the feature set. Selecting the proper
10 feature set is necessary to obtain the most robust classification
11 performance.

12 Poor feature sets cause a number of difficulties for
13 automatic classification. The use of too few features results in
14 poor classification accuracy. However, using too many features
15 also decreases overall classification accuracy. This counter-
16 intuitive "performance peaking" phenomenon is due to the "curse
17 of dimensionality," and affects all classifiers, whether neural
18 network or classical. Thus, feature reduction, identifying and
19 removing features that do not enhance classification performance,
20 plays an important part in feature set design. Superfluous
21 features contribute "opportunities" for misclassification and
22 should be eliminated to improve system robustness. Furthermore,
23 the complexity and cost of feature measurement systems and
24 pattern recognition systems are directly related to the number of
25 computed features. Consequently, from both a performance and
26 economic perspective, it is important to have effective feature
27 reduction algorithms.

1 Furthermore, in some applications the pattern recognition
2 systems used exhibit large variations in performance due to
3 differences in input systems, measurement system performance, and
4 the environment (changing noise, light, temperature). Feature
5 sets that work well in one environment may fail miserably in
6 another and cannot form the basis for a robust classification
7 system. Recognition systems used in changing environments or
8 with changing collection mechanisms often require adaptive, *in*
9 *situ*, selection of features from a global feature set. Given a
10 list of features that are known to be useful in certain
11 situations, adaptive feature selection from the list of features
12 is indistinguishable from feature reduction as it is used in this
13 application. Clearly, feature reduction algorithms must be
14 computationally fast if adaptive feature selection is to be
15 undertaken *in situ*.

16 Although several feature reduction techniques have been
17 developed, they generally suffer from one or more disadvantages
18 which limit their use in many applications. For example, a
19 direct algorithm for obtaining the feature set with the lowest
20 classification error rate (the optimal feature set) is the
21 exhaustive combination method (ECM). ECM examines all possible
22 combinations of features to find the best feature set. Although
23 systems employing ECM will obtain the optimal feature set, they
24 are computationally complex and clearly impractical for most
25 applications unless the number of features is small because the
26 number of possible combinations grows exponentially with the
27 number of features. For example, finding the optimal feature set

1 from a set of 35 features requires examining $2^{35} \cong 3.4 \times 10^{10}$
2 feature sets, while a set of 70 features requires examining $2^{70} \cong$
3 1.2×10^{21} feature sets.

4 Another technique, single feature classification performance
5 ordering (SFCPO), linearly orders individual features by
6 classification performance when each feature is used alone. This
7 ordering is easily thresholded for various purposes, including
8 feature reduction. SFCPO is good at optimizing classification
9 performance, and it is not limited by severe computational
10 complexity or overhead. However, SFCPO does not provide any
11 intuitive interpretations that facilitate understanding of or
12 provide insight to the reduction or classification problem.

13 Another commonly used method of feature reduction is
14 attributed to R. A. Fisher. Fisher's method derives a new set of
15 features that are linear combinations of the original features.
16 The span of these newly derived features is called the multiclass
17 Fisher projection space (FPS). The FPS maximally separates the
18 class means relative to the class variances. This geometric
19 interpretation greatly facilitates intuition and strongly
20 indicates that the FPS is a good space for feature reduction.
21 Additionally, if the classes are linearly separable in the FPS,
22 Fisher's linear discriminator, defined on the FPS, can be used
23 for classification. However, the use of the FPS does not
24 guarantee linear separability.

25 Although Fisher's method is computationally fast, it does
26 not linearly order the individual features in terms of their

1 relative importance to classification. Additionally, the FPS is
2 unlikely to contain any of the original features in its span, and
3 thus, features that have natural interpretations may not be
4 readily interpreted if they have been modified.

5 Thus, what is needed is a system for feature reduction that
6 linearly ranks features in terms of their importance to
7 classification based on the original features relationship to the
8 FPS. Such a system would provide intuitive interpretations that
9 facilitate problem understanding and insight while maintaining
10 the natural interpretation of the original features.

11

12 SUMMARY OF THE INVENTION

13 Accordingly, it is a general purpose and object of the
14 present invention to provide a system and method for linearly
15 ranking features in order of importance.

16 Another object of the present invention is the provision of
17 a system and method for feature reduction.

18 A further object of the present invention is to provide a
19 feature ranking and reduction system that supports adaptive
20 optimization of an automatic classification system.

21 Yet another object of the present invention is to provide a
22 system for feature ranking and/or reduction which does not call
23 for relatively complex and/or extensive computations or
24 relatively large storage requirements.

25 Yet a further object of the present invention is the
26 provision of a system and method for feature ranking and/or
27 reduction which preserves the natural interpretation of the

1 original features and supports intuitive interpretations
2 facilitating problem understanding and insight.

3 Still another object of the present invention is the
4 provision of a system and method for feature ranking and/or
5 reduction that reconciles multiple feature rankings.

6 These and other objects made apparent hereinafter are
7 accomplished with the present invention by providing a system for
8 ranking features by exploiting their relationship to the Fisher
9 projection space. The system ranks the n features in a feature
10 set using a set of exemplars wherein each exemplar corresponds to
11 one of the M event classes of an associated feature-based
12 classification system. The system uses a feature extractor to
13 produce an n -element feature vector for each exemplar and build a
14 design set comprising the n -element feature vectors. A training
15 set compiler creates a training set by randomly sampling feature
16 vectors from the design set. A projection space processor then
17 generates the smoothed Fisher projection space (SFPS) for the
18 training set. A feature ranking processor uses the (SFPS) to
19 generate a Procrustes angle for each feature in said feature set
20 and linearly rank the features by numerical size of their
21 respective Procrustes angles. A feature reduction processor
22 eliminates features which are not important for classification
23 based on the linear ranking of the features.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of the functional units of a system for feature ranking and reduction in accordance with the present invention;

FIG. 2 is a block diagram illustrating a system for feature ranking and reduction in accordance with the present invention;

FIG. 3 illustrates a sample performance evaluation plot for a feature set ranked in accordance with the present invention;

FIG. 4A graphically represents a ranking count for each feature in the feature set; and

FIG. 4B graphically illustrates a thresholded ranking count for the ranking count of FIG. 4A.

DESCRIPTION OF THE PREFERRED EMBODIMENT

In the embodiments described herein, the method for feature ranking and reduction in accordance with the present invention will be referred to as "Procrustes ordering." In a Procrustes ordering method, individual features are ranked by exploiting their relationship to the Fisher projection space (FPS). The maximal separation property of the FPS provides a good reduced feature space for nonlinear classification problems. However, the FPS is derived from linear combinations of the original features and is unlikely to contain any of the original features in its span. Procrustes ordering maintains the original features by choosing a subset of the original feature set that best approximates (in the least squares sense) the FPS. As will be readily apparent to those skilled in the art, Procrustes ordering

1 is data based and can be used in conjunction with any feature-
2 based classification pattern recognition method or system.
3 However, Procrustes ordering requires that the classification
4 system have $M \geq 1$ event classes and that the feature set
5 comprise $n \geq 2$ features.

6 Referring now to FIG. 1, there is shown a diagram
7 illustrating the functional units of a system for ranking
8 features in order of importance and for selecting the optimal
9 feature set in accordance with the present invention. The system
10 operates on the feature set and a set of sample input patterns
11 (exemplars) to produce a reduced feature set. The reduced
12 feature set is the subset of the original feature set that
13 provides the most robust classification performance. The
14 features (real-valued functions of the data defined to measure
15 class specific properties) comprising the feature set can be
16 defined using any conventional method that is consistent with the
17 classification pattern recognition system for which the feature
18 set is designed.

19 In FIG. 1, a feature extraction processor 10 operates to
20 compile and collect the data necessary for Procrustes ordering.
21 Feature extraction processor 10 acquires exemplars (samples) for
22 each event class and extracts a feature vector from each of the
23 exemplars. The feature vectors are compiled into a design set.
24 Each feature vector comprises n feature values, one for each of
25 the n features in the feature set.

1 Training/evaluation set compiler 20 creates a training set
2 and, optionally, an evaluation set by sampling the design set
3 compiled by feature extraction processor 10. A projection space
4 processor 30 produces a reduced feature space that is a subset of
5 the original n -dimensional feature space. The reduced feature
6 space can be generated using several known methods such as
7 principal component analysis, factor analysis, Fisher's method,
8 or the like. Similarly, a variation of Fisher's method (referred
9 to herein as "Smoothed Fisher") in which the sample means and the
10 within-class sample covariance matrix of Fisher's method are
11 replaced with mean vectors and covariance matrices derived from
12 event class probability density function estimates can be used to
13 derive a reduced feature space (referred to herein as "Smoothed
14 Fisher Projection Space" (SFPS)) from the training set.

15 Feature ranking processor 40 calculates the Procrustes
16 angle, the angle between a given feature and the reduced feature
17 space, for each feature in the feature set. Processor 40 then
18 generates a Procrustes ranking by ordering the features by
19 increasing numerical size of their Procrustes angles. An
20 optional classification performance evaluator 50 evaluates the
21 performance of the classification system under the Procrustes
22 ranking using the evaluation set built by compiler 20. However,
23 classification performance evaluator 50 has a broken line in FIG.
24 1 because the evaluator is an optional element of the present
25 invention and be omitted if a desired application does not
26 require its use.

1 Counter 60 counts the number of ranking trials performed to
2 ensure that a sufficient number of ranking trials have been
3 performed to satisfy a confidence interval criteria. A trial is
4 defined as a ranking of features by feature ranking processor 40
5 for a single training set.

6 Feature reduction processor 70 operates on the feature
7 ranking(s) derived by Procrustes feature ranking processor 40 to
8 select the subset of features that are important for
9 classification (the Procrustes reduced feature set). Feature
10 reduction processor 70 combines the Procrustes rankings from
11 processor 40 and generates a Procrustes reduced feature set by
12 identifying the subset of features that are consistently highly
13 ranked. Alternatively, a reduced feature set can be created by
14 eliminating those features having a Procrustes angle greater than
15 a threshold (decision confidence) angle.

16 The present invention is shown more particularly in FIG. 2,
17 in which is shown a block diagram illustrating a system for
18 feature ranking and reduction in accordance with the present
19 invention. In FIG. 2, feature extraction processor 10 acquires
20 the feature set to be reduced and a set of exemplars for the
21 different event classes. Preferably, the exemplars are obtained
22 from the measurement system (not shown) of the automatic pattern
23 classification system.

24 The measurement system provides a means for sampling the
25 input data to be classified and converting the data into a form
26 for further processing. For example, in a recognition system for
27 classifying acoustic signals, the measurement system (e.g.,

1 transducer array and acoustic signal processor) receives the
2 acoustic signal and converts it into a digital representation of
3 the signal. In such a system, the features used to define event
4 classes may include frequency, signal to noise ratio, coherence,
5 beam pattern, beam width, array gain, pulse length, or noise
6 spectral density. Similarly, in a character recognition system,
7 the measurement system, which can be a digital scanner, images
8 the unknown (input) characters and stores the image in a digital
9 format.

10 Feature extraction processor 10 is programmed to extract an
11 n -element feature vector from each of the exemplars received.
12 The n feature values of the feature vector are generated
13 according to the features defined within the feature set.
14 Feature extraction processor 10 compiles the feature vectors into
15 a design set stored in storage unit 12. Storage unit 12 which
16 can be random access memory, a magnetic storage device, or the
17 like is a shared storage device accessible to both feature
18 extraction processor 10 and training/evaluation set compiler 20.

19 In a preferred embodiment, feature extraction processor 10
20 builds a labeled design set (the event class for each exemplar is
21 known) having at least two exemplars for each event class.
22 Building a labeled design set permits evaluation of individual
23 Procrustes ranking trials and comparison of reduced feature sets.
24 Additionally, having at least two exemplars for each event class
25 enables disjoint training and evaluation sets to be built.

1 After feature extraction processor 10 produces a feature
2 vector for each of the exemplars, the processor supplies a
3 control signal to training/evaluation set compiler 20. When
4 training/evaluation set compiler 20 receives the control signal
5 from feature extraction processor 10, compiler 20 builds a
6 training set and, optionally, an evaluation set by sampling the
7 exemplars in the design set compiled by feature processor 10.
8 Compiler 20 builds an evaluation set if the performance of the
9 Procrustes ranking for that training set will be evaluated by
10 performance evaluator 50.

11 In a preferred embodiment, the design set is labeled and the
12 training and evaluation sets are created by uniformly randomly
13 sampling a subset of the exemplars from each of the event classes
14 to create disjoint training and evaluation sets. The training
15 and evaluation sets should contain at least one exemplar from
16 each of the event classes. If the design set is not labeled, a
17 training set is compiled by sampling the entire design set.
18 After compiling the training set and, if necessary, the
19 evaluation set, compiler 20 notifies projection space processor
20 30 that a training set has been built.

21 Projection space processor 30 comprises a probability
22 density function (PDF) generator 32, a covariance matrix
23 generator 34, and an eigenvector generator 36. Preferably,
24 processor 30 produces a reduced feature space by deriving a
25 Smoothed Fisher Projection Space (SFPS) for the training set
26 built by compiler 20. The SFPS is based on a variation of
27 Fisher's method, a well known feature reduction method attributed

1 to R. A. Fisher. Fisher's method derives a new set of features
2 that are linear combinations of the original features. The span
3 of these derived features is defined as the multiclass Fisher
4 projection space. The FPS maximally separates the class means
5 relative to the class variances. This geometric interpretation
6 facilitates intuition and the maximal separation property of the
7 FPS suggests that it is a good reduced feature space for
8 nonlinear classification problems. A detailed description of
9 Fisher's method can be found in R. O. Duda and P. E. Hart, PATTERN
10 CLASSIFICATION AND SCENE ANALYSIS, (Wiley & Sons 1973) pp. 114-123
11 incorporated herein by reference.

12 The usual formulation of Fisher's method uses sample means
13 for each class and a (pooled) within-class sample covariance
14 matrix. To derive a SFPS, a probability density function (PDF)
15 is estimated for each of the event classes using the feature
16 vectors compiled in the training set. Expressions for mean
17 vectors and covariance matrices of these estimated class PDFs are
18 then used in Fisher's method to define a SFPS. The SFPS is
19 preferred over the FPS because smoothing the feature vectors in
20 the manner described reduces the effects of outliers on the SFPS.

21 PDF generator 32 generates an estimated PDF for each of the
22 event classes using the feature vectors from the training set
23 built by compiler 20. Any valid PDF estimation technique can be
24 used to obtain the estimated PDFs. In a preferred embodiment,
25 PDF generator 32 is programmed to generate the PDFs by estimating
26 the unknown PDFs as a mixture of Gaussian PDFs. This generation
27 technique is preferred because it is applicable both for event

1 classes that are well represented and those that are poorly
2 represented. A detailed discussion of this estimation technique
3 can be found in R.L. Streit and T.E. Luginbuhl, *Maximum*
4 *Likelihood Training of Probablistic Neural Networks*, IEEE
5 *Transactions on Neural Networks*, vol. 5, no. 5, September 1994,
6 pp. 764-783, incorporated herein by reference.

7 It should be noted that the above technique uses a
8 homoscedastic mixture (that is, common covariance for all mixture
9 components) of Gaussian PDFs to estimate the unknown class PDFs.
10 However, if there is enough data in the training set to estimate
11 class specific covariance matrices, the above technique can be
12 extended to use a heteroscedastic mixture (that is, different
13 covariance matrices for each component in each class).

14 Additionally, it should be noted that when an unlabeled
15 training set is used, an overall likelihood function is estimated
16 rather than estimating PDFs for each of the event classes. The
17 overall PDF can be estimated using common clustering techniques.

18 After the estimated PDFs are generated, covariance matrix
19 generator 34 generates the within-class scatter matrix, Σ_w , and
20 the between-class scatter matrix, Σ_b , using the estimated PDFs
21 generated by PDF generator 32. If $p_j(X)$ denotes the estimated PDF
22 for the j^{th} class, covariance matrix generator 34 can be
23 programmed to generate the within-class scatter matrix, Σ_w , and
24 the between-class scatter matrix, Σ_b , using the known general
25 formulas:

$$\Sigma_w = \sum_{j=1}^M \alpha_j \int_{-\infty}^{\infty} (X - \mu_j)(X - \mu_j)^t p_j(X) dX \quad (1)$$

and

$$\Sigma_b = \sum_{j=1}^M \alpha_j \int_{-\infty}^{\infty} (X - \mu)(X - \mu)^t p_j(X) dX \quad (2)$$

where μ_j is the mean of the estimated PDF, $p_j(X)$, for class j and is given by

$$\mu_j = \int_{-\infty}^{\infty} X p_j(X) dX \quad (3)$$

and μ is the global mean defined by

$$\mu = \sum_{j=1}^M \alpha_j \int_{-\infty}^{\infty} X p_j(X) dX. \quad (4)$$

In the equations, α_j represents the mixing proportion of the mixture associated with class j , superscript t denotes the vector/matrix transpose, and X represents the feature vector.

Having obtained the smoothed mean vectors and covariance matrices, the SFPS can be defined. By Fisher's method, maximizing the between-class to within-class scatter matrices requires maximizing the Rayleigh quotient given by

$$J(w) = \frac{w^t \Sigma_b w}{w^t \Sigma_w w} \quad (5)$$

where $w \in R^n$ (R^n denoting the n -dimensional set of real numbers). Maximizing $J(w)$ is equivalent to solving the generalized eigenvalue problem

$$\Sigma_b w = \lambda \Sigma_w w \quad (6)$$

where λ and w denote the eigenvalue and the eigenvectors in the generalized eigenvalue problem of equation (6).

Eigenvector generator 36 uses the within-class scatter matrix, Σ_w , and the between-class scatter matrix, Σ_b , generated by covariance matrix generator 34 to generate the eigenvectors that define the SFPS. In a preferred embodiment eigenvector generator 36 is programmed to generate the eigenvectors defining the SFPS by forming the Cholesky decomposition of Σ_w given by

$$\Sigma_w = LL^t \quad (7)$$

where L is a lower triangular matrix and superscript t denotes the vector/matrix transpose. Substituting the Cholesky decomposition into equation (6) gives

$$\Sigma_b w = \lambda LL^t w \quad (8)$$

which expands to

$$\Sigma_b L^{-t}(L^t w) = \lambda L(L^t w). \quad (9)$$

By defining $y = L^t w$ (i.e., forward transform or rotation of the original eigenvectors) and $C = L^{-1} \Sigma_b L^{-t}$, equation (9) reduces to familiar eigenproblem given by

$$Cy = \lambda y. \quad (10)$$

From equation (10) the singular value decomposition of C , $C = U \Sigma V^t$, can then easily be computed. The eigenvectors of C are the columns of V . Assume $p \geq 1$ singular values are non-zero, if W_i

1 denotes column i of V , then a $n \times p$ matrix, $\tilde{W}$, whose columns
2 are the non-zero eigenvectors that define the SFPS, is given by
3
$$\tilde{W} = [W_1 W_2 \dots W_p] \in \mathbb{R}^{n \times p}. \quad (11)$$

4 It should be noted that the $p \times p$ matrix, $\tilde{W}^t \tilde{W}$, is the identity
5 matrix, $I^{p \times p}$, because the columns of $\tilde{W}$ are orthonormal.

6 It should be apparent to those skilled in the art that there
7 are at most $M-1$ non-zero eigenvalues of the generalized
8 eigenproblem given in equation (6). The span of the eigenvectors
9 corresponding to the largest p , $1 \leq p \leq M-1$, non-zero
10 eigenvalues is the smoothed Fisher projection space of dimension
11 p , denoted $SFPS(p)$. The rank of the $SFPS(p)$ is exactly p because
12 the eigenvectors spanning $SFPS(p)$ are linearly independent.
13 Although the preferred $SFPS(p)$ is the one resulting from the
14 largest dimension, $SFPS(M-1)$, the Procrustes ordering can be
15 defined for every $SFPS(p)$, for $p = 1, 2, \dots, (M-1)$. Thus, the
16 dimension p of the $SFPS(p)$ will be suppressed, and $SFPS(p)$ will
17 be written as $SFPS$.

18 Having generated the eigenvectors that define the reduced
19 feature space, processor 30 passes the eigenvectors to feature
20 ranking processor 40. When processor 40 receives the
21 eigenvectors, the processor generates a Procrustes angle for each
22 feature in the feature set then ranks the features by increasing
23 numerical size of their Procrustes angles.

1 The cosine of the angle, ϕ , between an arbitrarily specified
2 non-zero vector, $x \in R^n$, and the SFPS can be defined relative to
3 the original coordinate axes or to the coordinate axes defined by
4 the SFPS. The two methods differ by a linear transformation, L^t ,
5 where L is the Cholesky factor of the "within-class" scatter
6 matrix, Σ_w , (recall the forward transform performed in generating
7 the eigenvectors defining the SFPS). Preferably, the angle
8 relative to the SFPS is used because this angle allows for
9 determining a threshold angle for significant features. A
10 complete description of the determination of a threshold angle is
11 described below in reference to feature reduction processor 70.

12 The Procrustes angles for each feature can be calculated by
13 projecting the features onto the p -dimensional SFPS and measuring
14 the angle between the feature and the projection. To determine
15 the Procrustes angle, the original features axis must be rotated
16 since it was necessary to rotate (forward transform) the
17 eigenvectors, w , to solve the generalized eigenvalue problem
18 given in equation (6). The original features axis can be rotated
19 by multiplying by the Cholesky factor of the within-class scatter
20 matrix defined in equation (7). Let x_j be the j^{th} feature, that
21 is, $x_j = f_j \equiv (0, \dots, 0, 1, 0, \dots, 0)^t$. Rotating the feature axis yields

$$22 \qquad \tilde{x}_j = L^t x_j \qquad (12)$$

23 where L^t is obtained from the Cholesky decomposition given in
24 equation (7).

Using a least squares approach, the projection of $\tilde{x}_j$ onto the column space, $\tilde{W}$, defining the SFPS is given by

$$\text{proj}_{\tilde{W}} \tilde{x}_j = \tilde{W}(\tilde{W}^t \tilde{W})^{-1} \tilde{W}^t \tilde{x}_j. \quad (13)$$

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

$$\text{proj}_{\tilde{W}} \tilde{x}_j = \tilde{W} \tilde{W}^t \tilde{x}_j. \quad (14)$$

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

$$\cos \phi = \frac{a^t b}{\|a\|_2 \|b\|_2}. \quad (15)$$

Equation (15) can be used to provide the Procrustes angle for the j^{th} feature, x_j , if a denotes the feature vector and b denotes the projection of the feature vector onto the SFPS, that is,

$$a = \tilde{x}_j \quad (16)$$

and

$$b = \tilde{W} \tilde{W}^t \tilde{x}_j. \quad (17)$$

Using equations (16) and (17) allows the following reductions:

$$\|a\|_2 = (\tilde{x}_j^t \tilde{x}_j)^{\frac{1}{2}} \quad (18)$$

$$= ((L^t x_j)^t (L^t x_j))^{\frac{1}{2}} \quad (19)$$

$$= \|L^t x_j\|_2 \quad (20)$$

and

$$\|b\|_2 = ((\tilde{W} \tilde{W}^t \tilde{x}_j)^t (\tilde{W} \tilde{W}^t \tilde{x}_j))^{\frac{1}{2}} \quad (21)$$

$$1 \quad = (\tilde{x}_j^t \tilde{W} \tilde{W}^t \tilde{W} \tilde{W}^t \tilde{x}_j)^{\frac{1}{2}} \quad (22)$$

$$2 \quad = \|\tilde{W}^t L^t \tilde{x}_j\|_2 \quad (23)$$

3 Substituting equations (16), (17), (20), and (23) into equation
 4 (15) gives the expression for the Procrustes angle, ϕ_j , between
 5 the j^{th} feature, x_j , and any non-zero vector in the SFPS. This
 6 angle, relative to the SFPS, is defined as

$$7 \quad \phi_j = \cos^{-1} \left\{ \frac{\|\tilde{W}^t L^t \tilde{x}_j\|_2}{\|L^t x_j\|_2} \right\}. \quad (24)$$

8 The Procrustes angle, ϕ_j , will be uniquely defined if it is
 9 restricted to lie between 0 and 90 degrees, and will be the same
 10 angle for all vectors in the subspace spanned by x_j .

11 As previously described, the Procrustes ordering of the
 12 feature set is defined by ranking the features by increasing
 13 numerical size of their Procrustes angles. Procrustes ordering
 14 assumes that the SFPS is a good space for feature reduction.
 15 Procrustes ordering exploits this property of the SFPS by
 16 selecting a subset of the original features that best
 17 approximates the SFPS. The Procrustes angle is a measure of
 18 linear independence between a feature and the SFPS. If the angle
 19 of a particular feature is small (near zero), the feature is
 20 nearly in the span of the SFPS; however, if the angle is large
 21 (near 90 degrees), the feature is nearly orthogonal to the SFPS.
 22 Intuitively, features with small Procrustes angles are good
 23 features for classification, whereas, features with large

1 Procrustes angles are poor features for classification. The
2 first feature in the Procrustes ordering, therefore, has the
3 smallest Procrustes angle, and the last feature has the largest
4 angle.

5 Processor 40 generates the Procrustes angle for each feature
6 (ϕ_j , for $j= 1$ to n) using equation (24). After generating the
7 Procrustes angle for a feature, the feature and its Procrustes
8 angle are positioned in the Procrustes ranking in accordance with
9 the size of the feature's Procrustes angle.

10 Classification performance evaluator 50 uses the evaluation
11 set built by compiler 20 to evaluate the performance of the
12 classification system under the Procrustes ranking generated by
13 feature ranking processor 40. Classification evaluator 50 can be
14 programmed to simulate the classification system. Alternatively,
15 classification evaluator 50 can be programmed to initiate the
16 classification system and transfer the appropriate feature set
17 and input vectors to the classification system. Any conventional
18 performance evaluation technique can be used. Preferably,
19 evaluator 50 is used to generate a performance curve based on
20 linear combinations of the features ranked by feature ranking
21 processor 40 to determine the subset that provides the best
22 performance. That is, the Procrustes ordering is sequentially
23 tested (i.e., feature rankings (1), (1,2), ... (1,2,...j), ...
24 (1,2,...n)) and the performance (probability of correct
25 classification) is plotted as a function of j , the feature index.

1 FIG. 3 shows a sample performance plot (probability of
2 correct classification vs. number of features). In FIG. 3, the
3 classification performance peaks when 23 features are used and
4 begins to decline when more than 27 features are used; indicating
5 that the reduced feature set should contain between 23 and 27
6 features. This performance evaluation can be used for comparison
7 against other Procrustes orderings or for comparison with the
8 number of features in the Procrustes reduced feature set derived
9 by feature reduction processor 70. Evaluating classification
10 performance is useful if only a small number of trial rankings
11 will be performed. Evaluating classification performance need
12 not be performed for every Procrustes ranking trial produced by
13 feature ranking processor 40, and the evaluator can be omitted
14 entirely if a desired application does not require it.

15 Referring again to FIG. 2, counter 60 determines if another
16 Procrustes ordering, based on a different training set, will be
17 performed. Preferably, multiple Procrustes ordering trials are
18 performed to increase the utility of a small design set by
19 exploiting the variability within the design set. Using multiple
20 trials can reduce the bias and variance associated with
21 performance estimates based on a small design set. In choosing
22 the number of ordering trials to perform, consideration should be
23 given to the size of the design set and the processing time
24 available to perform the trials.

25 Counter 60 receives the Procrustes ranking created by
26 processor 40 and stores the ranking in a Procrustes ranking
27 storage unit 62. Storage unit 62 which can be random access

1 memory, a magnetic storage device, or the like is accessible to
2 both counter 60 and feature reduction processor 70. After
3 receiving the Procrustes ranking for processor 40, counter 60
4 increments a trial counter and compares the number of ranking
5 trials performed with a predetermined trials run number that
6 identifies the total number of ranking trials to be performed.
7 If more trials are to be performed, counter 60 supplies a control
8 signal 64 to training/evaluation set compiler 20 to initiate the
9 compilation of another training set and, if required, an
10 evaluation set. Preferably, the predetermined trials run number
11 is calculated to ensure that a sufficient the number of ranking
12 trials are performed to satisfy a statistical criteria such as
13 confidence or tolerance intervals.

14 Feature reduction processor 70 comprises a feature reducer
15 72 and, optionally, a feature filter 74. Feature reduction
16 processor 70 receives the number of Procrustes ranking trials
17 performed from counter 40. If multiple Procrustes trials were
18 performed, feature reduction processor 70 initiates feature
19 reducer 72. However, if only one Procrustes ranking trial was
20 performed, reduction processor 70 initiates feature filter 74.

21 When multiple Procrustes ranking trials are performed,
22 feature ranking processor generates a separate Procrustes feature
23 ranking for each trial. These feature rankings may vary from
24 trial to trial. Feature reducer 72 reconciles the multiple
25 trials and generates the Procrustes reduced feature set. The
26 across-trial feature ordering is based on the assumption that the
27 number of times a particular feature is highly ranked is an

1 indication of the relative importance. Feature reducer 72
2 generates the Procrustes reduced feature set by first combining
3 the multiple Procrustes rankings produced by feature ranking
4 processor 40 and then identifying the features that are
5 consistently highly ranked across the multiple trials.

6 Feature reducer 72 combines the ranking of features across
7 multiple trials by counting the number of times each feature is
8 ranked in the top m positions; where $m=1,2,\dots,n$ (number of
9 features). That is, for $m=1$, feature processor 72 builds a
10 ranking count which indicates a ranking count number for each
11 feature. For $m=1$ the ranking count number equals the number of
12 times the feature was ranked first. For $m=2$, feature reducer 72
13 builds a ranking count in which the ranking count number for each
14 feature indicates the number of times the respective feature was
15 ranked first or second. Feature reducer 72 continues to build
16 these ranking counts until $m=n$ (that is n individual ranking
17 counts are built). Referring to FIG. 4A, there is shown a
18 graphical representation of a ranking count for $m=25$ calculated
19 from 100 Procrustes ranking trials of a feature set having 70
20 features. As can be seen in FIG. 4A, eight features (features 1,
21 3, 6-8, 23, 26, and 28) were ranked in the top 25 positions in
22 each of the 100 trials while 56 features were ranked in the top
23 25 at least once.

24 The features that are consistently highly ranked are of most
25 importance for classification. Feature reducer 72 identifies the

1 features which are consistently highly ranked by constructing a
2 thresholded version of each ranking count built. A thresholded
3 ranking count is constructed by varying a threshold, T , between 0
4 and the total number of trials and, for each threshold value,
5 counting the number of features whose ranking count number
6 exceeds the threshold. FIG. 4B shows the thresholded ranking
7 count for the ranking count of FIG. 4A. As can be seen in FIG.
8 4B, at $T=1$, the thresholded ranking count indicates that 56
9 features were ranked in the top 25 at least once, while at $T=100$,
10 the thresholded ranking count indicates that only eight features
11 were ranked in the top 25 for all 100 trials.

12 After calculating the thresholded ranking counts, feature
13 reducer 72 separates the features which are consistently highly
14 ranked from those which are only occasionally ranked by examining
15 each thresholded ranking count to determine the longest series of
16 threshold values over which the number of features whose ranking
17 count numbers exceed the threshold values remains constant. This
18 series is illustrated in FIG. 4B by the flat portion of the
19 curve, identified as 40, over a wide range of threshold values
20 (between 40 and 68). Portion 40 of the thresholded ranking curve
21 separates the features consistently highly ranked (feature with a
22 ranking count over 68) from those that are not consistently
23 ranked (ranking count less than 40). It should be noted that it
24 is not a threshold value (40 or 68 in FIG. 4B) that determines
25 whether a feature is considered to be consistently ranked.
26 Rather, it is the flat portion 40 of the curve, which occurs for
27 the same number of features, but over different threshold values,

1 in the other thresholded ranking counts, that indicates the
2 breakpoint between features. This breakpoint defines the
3 features that are important for classification.

4 The breakpoint identifies a constant number of features
5 whose ranking count number exceeds the threshold value (referred
6 to as the Procrustes number). The Procrustes number indicates
7 the size of the reduced feature set. The features comprising the
8 reduced feature set can be easily identified from the ranking
9 count. For example, the Procrustes number for the thresholded
10 ranking count shown in FIG. 4B is 23. The 23 features that
11 comprise the reduced feature set can easily be identified from
12 the ranking count (FIG. 4A) as the 23 features having the highest
13 ranking count numbers. It should be noted that the Procrustes
14 number may vary from one thresholded ranking count to another.
15 Thus, the Procrustes number that occurs consistently over a
16 number of thresholded ranking counts is used to identify the
17 number of features in the Procrustes reduced feature set.

18 When multiple trials have not been performed, feature filter
19 74 generates the Procrustes reduced feature set by eliminating
20 those features having a Procrustes angle greater than a threshold
21 angle. A threshold angle is determined by applying a statistical
22 significance test under an appropriate null hypothesis. To
23 formulate an appropriate null hypothesis, a model for the feature
24 generation process is defined. The model assumes the feature set
25 is comprised of two subsets; a *knowledge-based* set and an
26 *intuition-based* set. The knowledge-based set is defined as those
27 features that are derived from known measurable class

1 differences, whereas the intuition based set is comprised of
 2 features which are believed to define class differences. For
 3 most complex classification problems, the size of the knowledge-
 4 based set is small compared to the size of the intuition-based
 5 set; therefore, the underlying null hypothesis should be
 6 dominated by the intuition based set. Because Procrustes
 7 ordering is independent of vector length, the model adopted is
 8 that the feature set vectors are uniformly randomly distributed
 9 on the unit sphere in R^n . With this model, the feature selection
 10 process becomes the process of determining the subset of these
 11 "randomly" generated features that "happen" to best approximate
 12 the SFPS. If these assumptions regarding the feature set are
 13 accurate, thresholding the upper tail of the resulting PDF will
 14 enumerate those features that are poor for classification.

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

$$21 \quad P_{n,p}(\phi) = \frac{2\Gamma(\frac{n}{2})}{\Gamma(\frac{p}{2})\Gamma(\frac{(n-p)}{2})} \cos^{p-1} \phi \sin^{n-p-1} \phi \quad (25)$$

22 where $0 \leq \phi \leq \pi/2$ and $\Gamma(\cdot)$ represents the Gamma function.

23 Therefore, under a Procrustes ordering method, the angle of a

1 feature is significant at the $\alpha\%$ significance level if it lies
2 on the upper $\alpha\%$ tail of $P_{np}(\phi)$.

3 Feature filter 74 eliminates those features having a
4 Procrustes angle greater than a predetermined threshold wherein
5 the threshold is chosen to provide a predetermined significance
6 level (decision confidence) in accordance with equation (25).
7 The remaining features comprise the Procrustes reduced feature
8 set.

9 In addition to single trials, feature filter 74 is useful
10 for creating a reduced feature set when a small number of trials
11 have been run. Additionally, filter 74 may be used for
12 comparison against the reduced feature set generated by feature
13 reducer 74 or against the reduced feature set suggested by
14 performance evaluator 50.

15 What has thus been described is a system and method for
16 ranking features and reducing the number of features used in a
17 real-time feature based classification system. The present
18 invention provides a novel approach for ranking features in order
19 of importance and for reducing the size of a feature set and
20 offers several significant advantages over the prior art. First,
21 Procrustes ordering is fast and computationally simple enabling
22 its use for real-time, *in situ* applications. Second, it provides
23 geometric insight into the problem of feature selection while
24 maintaining the original interpretation of the given features.

25 Obviously many modifications and variations of the present
26 invention may become apparent in light of the above teachings.

1 For example, various feature space reduction techniques
2 including, but not limited to, principal component analysis or
3 factor analysis can be used to generate the reduced feature
4 space. The features can then be linearly ranked and reduced
5 based on the angle between the feature and the reduced feature
6 set. Similarly, other statistical models and analysis methods
7 may be used to combine the individual trial rankings into the
8 single across-trial reduced feature set generated by feature
9 reducer 72.

10 The elements in the embodiment of the system shown in FIG. 2
11 can be implemented using a combination of computer-readable
12 memory (e.g., EPROM) and combinatorial logic to rank features
13 and/or generate a reduced feature set. Alternatively, the system
14 can comprise software modules of a digital processing program
15 stored in computer-readable memory under control of a digital
16 processor that can be used to direct the processor to generate a
17 reduced feature set and/or rank feature.

18 In light of the above, it is therefore understood that
19 the invention may be
20 practiced otherwise than as specifically described.

2
3 SYSTEM AND METHOD FOR FEATURE SET REDUCTION

4
5 ABSTRACT OF THE DISCLOSURE

6 A system and method for ranking features by exploiting their
7 relationship to the Fisher projection space. The system ranks n
8 features in a feature set using a design set comprising exemplars
9 from each of M possible event classes of an associated feature-
10 based classification system. A training set is created by
11 randomly selecting exemplars from each of the M classes in the
12 design set. A "smoothed" Fisher projection space for the
13 training set is created by replacing the sample means and the
14 within-class sample covariance matrix normally used in deriving a
15 Fisher projection space with expressions for the mean vectors and
16 covariance matrices derived from event class probability density
17 function estimates. The angle between a given feature and the
18 smoothed Fisher projection space is calculated for each feature
19 in the feature set, and the features are then ordered by
20 increasing numerical size of this angle. The system produces a
21 reduced feature set by eliminating those features which are not
22 important for classification based on the linear ranking of the
23 features.

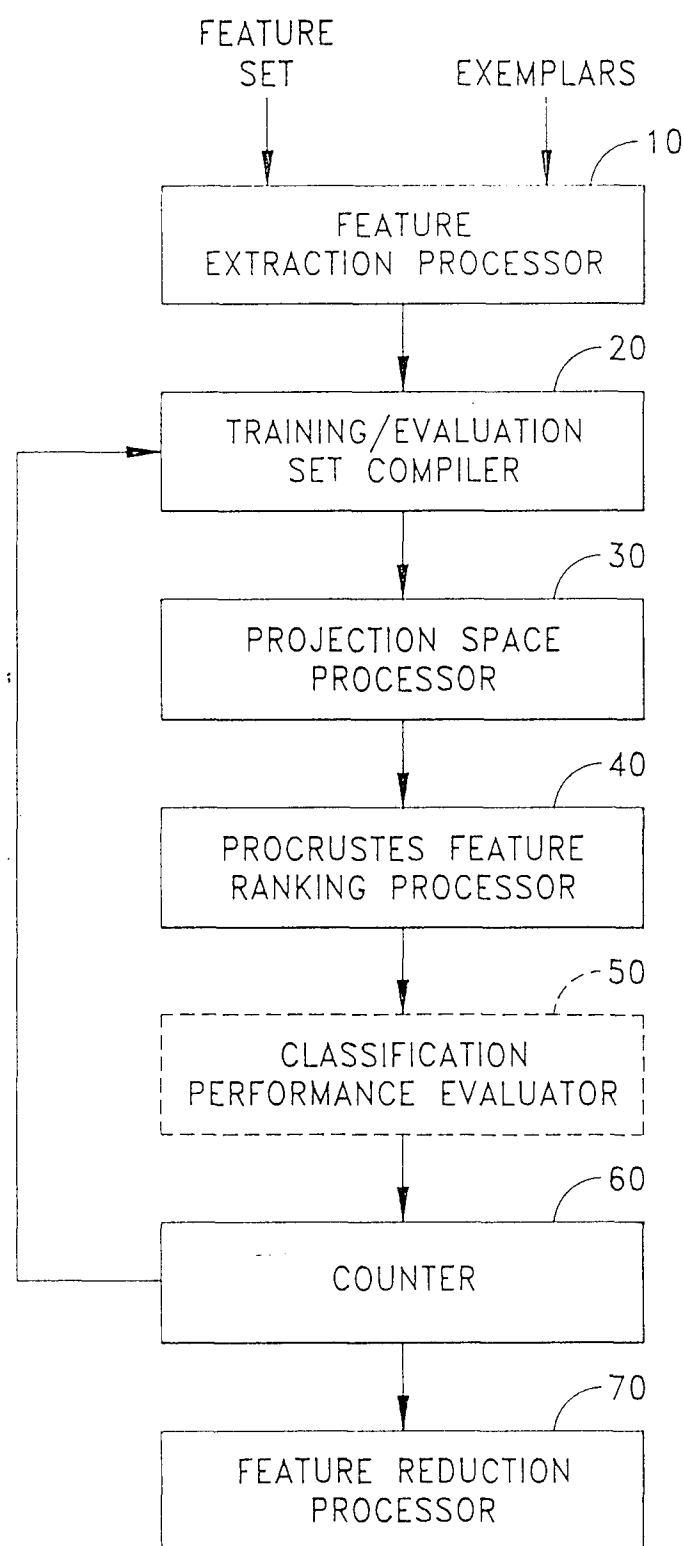


FIG. 1

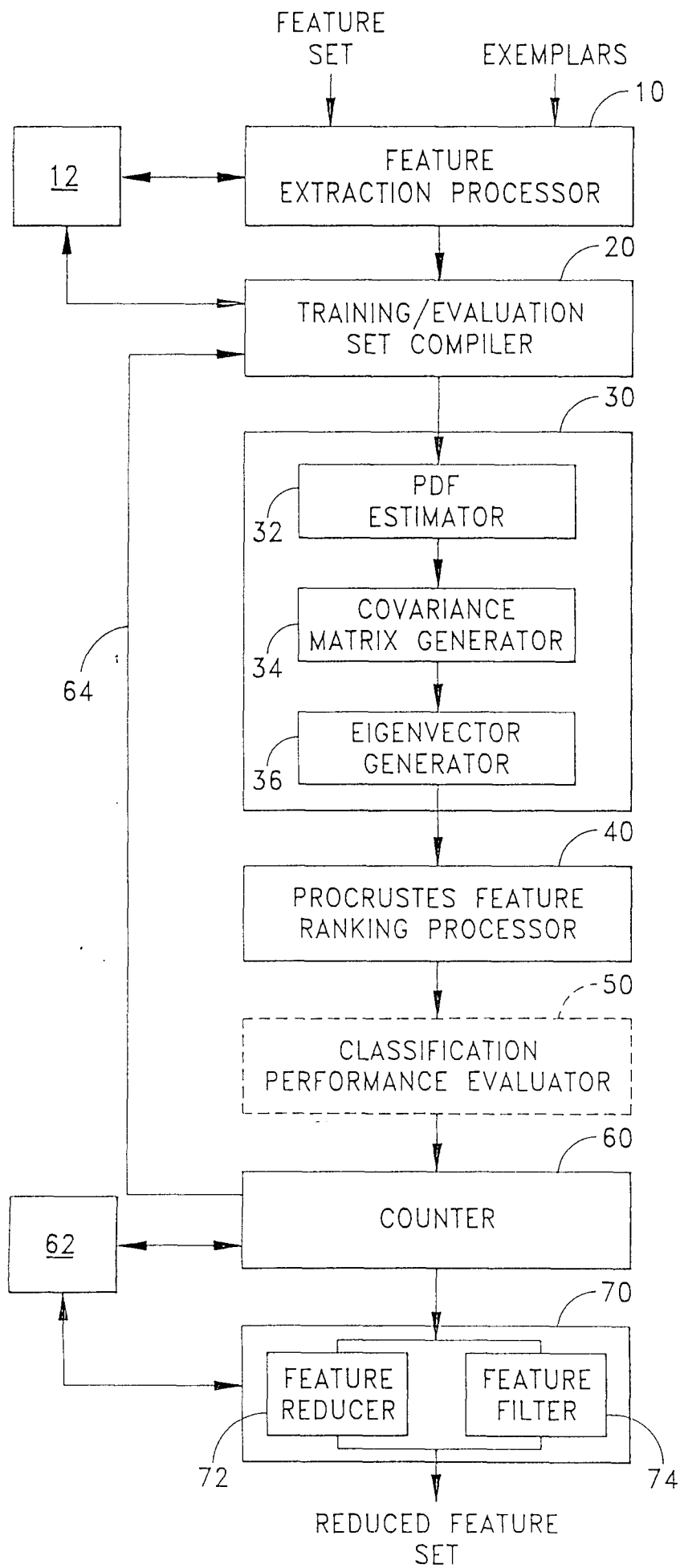


FIG. 2

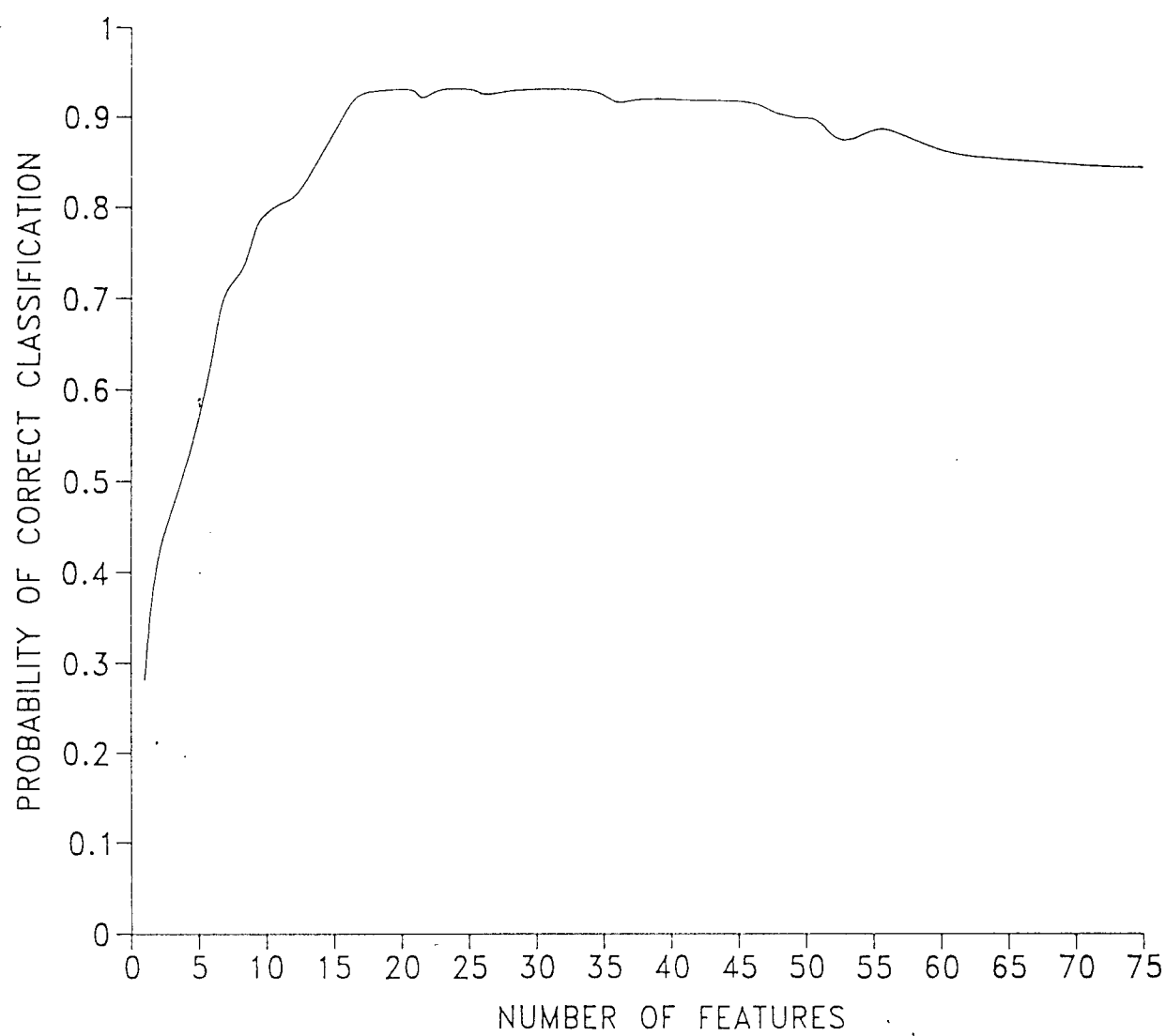


FIG. 3

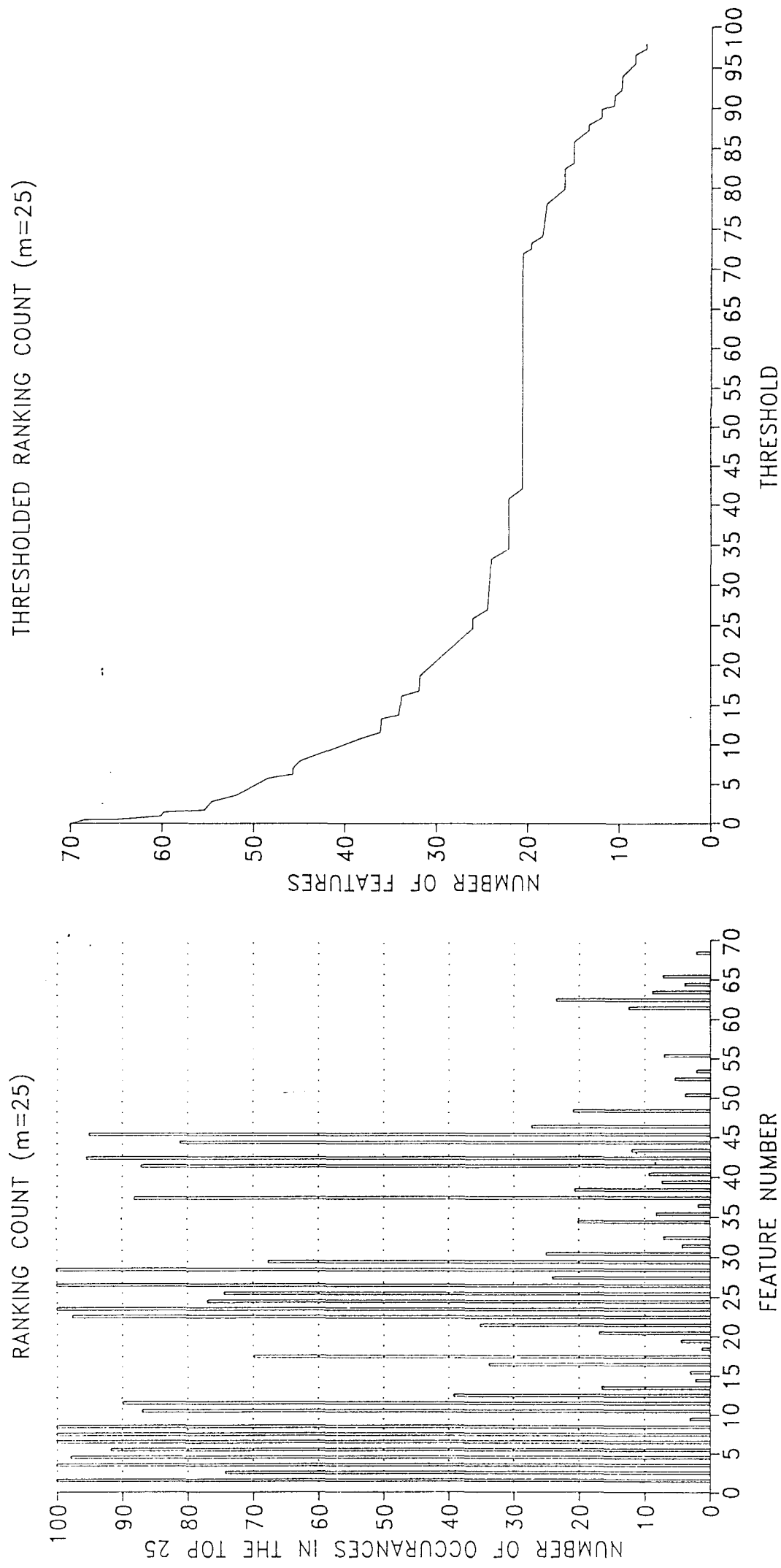


FIG. 4A

FIG. 4B