Sequential Feature Extraction Using Information-Theoretic Learning

Kenneth E. Hild II, Deniz Erdogmus, Kari Torkkola and Jose C. Principe

Abstract -- A classification system typically includes both a feature extractor and a classifier. The two components can be trained either sequentially or simultaneously. The former option has an implementation advantage since the extractor is trained independently of the classifier, but it is hindered by the sub-optimality of feature selection. Simultaneous training has the advantage of minimizing classification error, but it has implementation difficulties. Certain criteria, such as Minimum Classification Error, are better suited for simultaneous training, while other criteria, such as Mutual Information, are amenable for training the extractor either sequentially or simultaneously. Herein, an information-theoretic criterion is introduced and is evaluated for sequential training, in order to ascertain its ability to find relevant features for classification. The proposed method uses non-parametric estimation of Renyi's entropy to train the extractor by maximizing an approximation of the mutual information between the class labels and the output of the extractor. The proposed method is compared against seven other feature reduction methods and, when combined with a simple classifier, against the Support Vector Machine and Optimal Hyperplane. Interestingly, the evaluations show that the proposed method, when used in a *sequential* manner, performs at least as well as the best *simultaneous* feature reduction methods.

Index Terms -- Feature extraction, Information theory, Classification, Nonparametric statistics.

1 INTRODUCTION

Feature extraction is commonly employed as a pre-processor for applications including: visualization, classification, detection, and verification. Herein, feature reduction, which in the linear case is also known as subspace projection, is investigated as it applies to classification. Fig. 1 shows a block diagram of the major components used in a classification system. In this figure, $s_j(k)$, $x_j(k)$ and $y_j(k)$ are the size (N_I x 1) input features, (N_O x 1) output (transformed) features and the (N_C x 1) outputs of the classifier at time k and having class j (j = 1, 2, ..., N_C), respectively. For classification, optimality can be considered as the condition for which the probability of correct classification, 1 - P[$e(k)$ = 1], is maximized, where $e(k)$ = 1 if an error occurred at time k, and is 0 otherwise. An empirical estimation of the error probability will be used as the figure of merit.

Feature reduction methods attempt to improve generalization by reducing the variance in classification performance [32]. For certain classifiers, the improved generalization may occur since a reduction in the number of features causes a reduction in the number of free parameters (relative to the number of data points) required for classification. In other classifiers, the improved generalization may occur as a result of reducing the dimensionality of the required probability density function (pdf) estimation. However, classification performance does not improve indefinitely as the number of features is reduced, due to classifier bias [32]. At some point, the loss of information (about the class) inherent in reducing the number of features overwhelms any benefit gained from reducing, for example, the number of adaptable parameters. Obviously, this *bias-variance dilemma* is highly dependent upon the specific algorithms used for both the feature extractor and the classifier, and it is the trade-off between the two that provides the motivation for the present

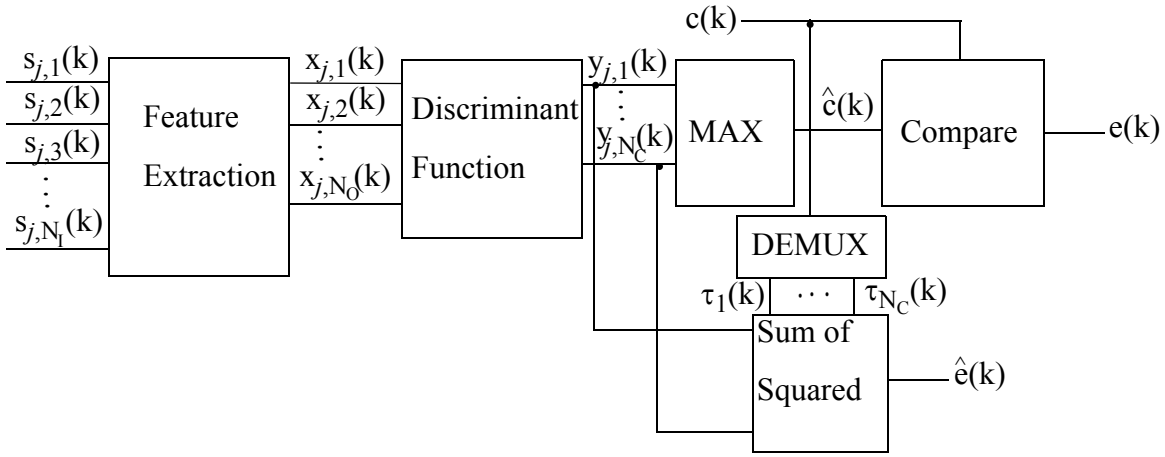

Fig. 1. Block diagram of feature extraction for classification. Included are blocks associated with both the operation and the training of the system.

study. Another method, Vapnik's Structural Risk Minimization (and its embodiment, the Support Vector Machine (SVM) [4]-[10]), has recently been determined to provide more explicit control of generalization through regularization. As such, it has become the obvious methodology with which to compare the performance of any feature reduction method.

Feature reduction methods may be categorized based on whether the projector and the classifier are trained sequentially or simultaneously. Sequential methods adapt the projector based on optimizing a criterion at the output of the *projector*. On the other hand, simultaneous methods adapt the projector based on optimizing a criterion at the output of the *classifier*. The former is independent of the classifier, while the latter is trained "through" the classifier. This gives sequential methods an implementation advantage. Not only does it take longer to train simultaneously due to the increased computational complexity, but also the cost function landscapes may become more difficult to search. Moreover, a new set of update equations must be derived and the projector must be re-trained if it is desired to evaluate a new classifier. On the other hand, simultaneous methods have the obvious (theoretically speaking) performance advantage in that they optimize the projector and the classifier together, that is they tune the projector to the classifier discriminant functions. It is expected that, every thing else remaining constant, simultaneous training will produce superior results compared to training sequentially. Consequently, the onus is on sequential training methods to demonstrate that there exists an easily implementable and general-purpose feature extraction algorithm that provides, with the combination of a suitable classifier, commensurate classification error.

Another difference between sequential and simultaneous systems is the choice of possible criteria for use in training the extractor. Criteria such as Minimum Classification Error (MCE) [1]-[3] and Mean Square Error (MSE) [11], for example, are well suited for training the extractor simultaneously. However, they are not appropriate for sequential training since both of these criteria are based on an error signal. In order to use the MSE criterion for sequential training, a set of N_O-dimensional targets (one for each class) must be defined in the output feature space, $y_j(k)$. There is no principled method known to the authors for selecting these targets in the feature space, combined with the expectation that the classification performance will vary considerably depending on the choice of targets used. Similarly, the MCE criterion is based on the assumption that the relative values of the output of the (in this case) projector, are related to the a posteriori probabilities of each class, to wit, it is based on the assumption that the values are the output of a classifier. Other criteria, on the other hand, are well suited for either sequential or simultaneous training. For example, an information-theoretic method that makes use of mutual information (MI) could be used. In this case, the extractor can be trained either

sequentially or simultaneously by maximizing the MI between the class labels and the outputs of the extractor or the classifier, respectively. Here we are particularly interested in measuring the performance of IT methods with respect to the preservation of discriminative information; hence, the information-theoretic methods considered in this paper are limited to sequentially-trained systems.

Since the theoretical performance advantage of a simultaneously-trained system, as compared to a sequentially-trained system, can not be guaranteed if the two use different criteria, this gives rise to an interesting question:

• *How does the performance of information-theoretic, sequential feature reduction methods compare to error-based, simultaneous methods?*

Furthermore, in order to put the performance results of the above-mentioned comparison in proper perspective, it may be asked:

• *How does the performance of the best feature reduction method considered here, when combined with a simple classifier, compare to the SVM and the closely related Optimal Hyperplane?*

In order to answer the first question, a common platform needs to be defined. Section 2 lists and describes the two classifiers used to compare the feature reduction algorithms, while Section 3 gives details on the constraints placed on the structure of the subspace projector. This is followed by Section 4, which gives an introduction to the proposed information-theoretic algorithm, MRMI-SIG. Section 5 then provides a brief description of all the competing feature reduction algorithms. These eight algorithms are then compared in Section 6 in order to address the first question. The results of the SVM and the Optimal Hyperplane (OH) are then given in order to address the second question.

2 CLASSIFIERS

The two Bayes (maximum likelihood) classifiers that are used for the comparisons are the Bayes-G and the Bayes-NP classifiers, both of which generate nonlinear decision surfaces. Notice that these classifiers are used to compare the feature reduction methods, and in a later section, are combined with the proposed feature reduction method in order to compare its performance with the SVM and the OH.

The Bayes-G classifier is a parametric classifier that uses only second-order statistics of the output features. It has a single output for each class, which gives an estimate of the (a posteriori) probability of the associated class given the data (more specifically, it gives an estimate of the likelihood function multiplied by the prior, which is proportional to the a posteriori) [11]. The likelihoods are estimated by assuming that, for every class j, the set of output features, x_j, is multi-variate Gaussian distributed. The estimated class for each output is determined as the one that maximizes the weighted likelihood,

$$y_j(k) = \frac{P(j)}{(\sqrt{2\pi})^{N_O}|C_j|^{1/2}} e^{\frac{-1}{2}(x(k)-\mu_j)^T C_j^{-1}(x(k)-\mu_j)} \quad (1)$$

where (j = 1, 2, ..., N_C), N_C is the number of classes, $x(k)$ is the (N_O x 1) data point at time k that is to be classified, μ_j is the (N_O x 1) mean vector of class j, C_j is the (N_O x N_O) covariance matrix of class j, and $P(j)$ is the prior probability of class j, i.e. N_j / N_T, where N_j is the number of data points contained in the training set belonging to class j, and N_T is the total size of the training set.

The Bayes-NP is a non-parametric classifier that uses Parzen Windows [12] to estimate each of the a posteriori distributions (once again, it actually estimates the likelihood multiplied by the associated prior probability, which, for classification, is tantamount to determining the a posteriori). The N_O-dimensional likelihood for class j is estimated by placing a Gaussian kernel at each point (one for each data point in the training set belonging to class j) in the input space (input to the classifier, which is the output feature space), summing these together, multiplying by the prior and then normalizing by N_j. Once the likelihood functions are determined, a new point is classified by evaluating the weighted likelihood of each class at the location of the data point in question. The class that produces the maximum value is then determined to be the correct class,

where the weighted likelihood of each class is given by,

$$y_j(k) = \frac{P(j)}{N_j} \sum_{i=1}^{N_j} G(x(k) - x_j(i), 2\sigma^2) \qquad (2)$$

$x_j(i)$ is the i^{th} data point of the training set of class j (j = 1, 2, ..., N_C), σ is a user-defined kernel size, and $G(\boldsymbol{x},\sigma^2)$ is,

$$G(x, \sigma^2) = \frac{1}{(\sqrt{2\pi})^{N_O}\sigma} e^{\left(\frac{-1}{2\sigma^2}x^T x\right)} \qquad (3)$$

A nice feature of the Bayes-NP classifier is that there are no implicit assumptions on the distribution of the output features. Therefore, this method is able to take into account higher-order statistics of the output features, including multiple-modality, unlike the Bayes-G classifier. It does, however, have a user-defined parameter, σ, whose value must be determined.

Other classifiers could also have been used, for example, the SVM, k-NN [49] or one of several different static artificial neural networks (ANN's); viz., the Multi-Layer Perceptron (MLP) [13], Radial Basis Function (RBF) [13], or Polynomial Network [14]. The MLP classifier, the most popular of the ANN's just listed, was considered briefly because it can be considered to encompass both the projector and the classifier in one functional unit. However, results from several papers indicate that the MLP classifier performs quite poorly for two of the three data sets used in the upcoming comparison [15], [16]. More importantly, the two classifiers chosen have the nice feature that either the requisite training is trivial (i.e. estimation of second-order statistics for the Bayes-G classifier) or non-existent (for the Bayes-NP, which is a memory-based classifier). This is to be compared to an MLP classifier, the performance of which is subject to local minima and other convergence issues. As such, its use in comparing multiple feature extractors would reduce the level of certainty that the measured performance difference is due to the type of feature extractor and not due to imperfect training of the classifier.

3 PROJECTION ARCHITECTURE

To simplify the exposition, the projection architecture is limited to the set of linear transformations. The equation for a completely general linear transformation is $x = Rs + b$, where R is a (N_O x N_I) matrix of coefficients and b is a (N_O x 1) vector of coefficients. Notice that the block diagram in Fig. 1 does not include the b vector since the two classifiers are invariant under a change in the mean. In addition, it is trivial to show that both classifiers are invariant under an invertible, linear transform. Therefore it is assumed, without loss of generality, that the original features have been shifted, rotated and scaled so that the resulting (N_I x 1) *input* features, $s(k)$, are zero-mean, (spatially) uncorrelated and have unit variance. The invariance of the classifiers to invertible transforms can also be used to reduce the number of free parameters. This is done by constraining the R matrix to be a pure rotation. In this case, the R matrix is formed using the first N_O rows of the product of (N_I x N_I) Given's rotation matrices (one for each pair of outputs) [17]. This reduces the number of parameters from $N_O N_I$ to $N_O(N_I - N_O)$ without unnecessarily restricting the possible set of decision surfaces that can be produced by a linear projection. Notice that rotations between retained output features have no effect on classification, nor do rotations between rejected outputs. Only rotations between a feature that is retained and a feature that is rejected has any effect. Due to its generality, unless stated otherwise, the subspace projector of each feature reduction algorithm will be constrained to be a rotation matrix.

4 INFORMATION-THEORETIC EXTRACTION

Whereas methods that use second-order statistics compare the *linear* relationship of (commonly) *two* random variables, e.g. one output feature and the class label, information-theoretic methods compare the *nonlinear* relationships of *multiple* random variables, i.e. a vector of features and the class label. This is accomplished by maximizing $I(X;C)$, the MI between the output features and the class labels. In words, $I(X;C)$ may be described as the amount of information the random (input feature) vector X carries about the class, C (where, realizations of X and C are given by the (N_O x 1) vector $x(k)$ and the scalar $c(k)$, respectively).

A number of different definitions of MI exist, but the one having (arguably) the greatest theoretical importance is the one named in honor of Claude Shannon, which is given by [18],

$$I(X;C) = H(X) - H(X|C) \tag{4}$$

where H(X) is Shannon's (differential) entropy,

$$H(X) = -\int_{-\infty}^{\infty} f_X(x)\log f_X(x)dx \tag{5}$$

and $f_X(x)$ is the pdf of X. First, maximizing I(X;C) minimizes the amount of information (about the class) lost due to the projection, where an intuitive definition of the amount of information loss is I(S;C) - I(X;C), i.e. the amount of information the input features have about the class minus the information that the projected features have about the class. A projection of $s(k)$ cannot add information so that I(X;C) is always less than or equal to I(S;C) and the difference is zero only if $x(k)$ is an invertible (linear or nonlinear) function of $s(k)$. Since I(S;C) is constant with regards to the projector, the loss is minimized by maximizing I(X;C). A second, more principled, argument for using Shannon's MI comes from the consideration of the upper and lower bounds it places on the Bayes error rate. A lower bound for the Bayes rate is given by the Fano inequality [19], and an upper bound is given by Hellman and Raviv [63]. Both of these bounds are minimized by maximizing I(X;C), or equivalently, by minimizing H(C|X). While these are good arguments for using Shannon's MI, it is not in common use due to the difficulty of estimating (5). Part of the difficulty stems from the integral, which may be avoided by discretization of the variables [20]-[24]. However, discretization requires the bin sizes for each region of the multi-dimensional output feature space to be appropriately selected. Too small and a zero probability occurs due to having a finite data set, too large and details of the shape are lost. In addition, the determination of the appropriate bin sizes is time consuming and methods that rely on discretization require a large amount of training data for satisfactory results [21], due to the "curse of dimensionality" [39].

On the other hand, Alfred Renyi introduced a definition of entropy, namely Renyi's (quadratic) entropy [25], that can be used as an alternative to Shannon's definition. This alternative entropy measure is given by,

$$H_R(X) = -\log \int_{-\infty}^{\infty} f_X(x)^2 dx \tag{6}$$

When the pdf is estimated using Parzen windows with Gaussian kernels, there is no need for discretization. This is due to the fact that the integral of the product of two Gaussians is a Gaussian evaluated at a single point [26]. As a result, there are no truncations or approximations required, outside of the implicit pdf estimation using Parzen windows. This motivates an approximation for mutual information, MI, using Renyi's entropy, namely,

$$I_R(X, C) \cong H_R(X) - H_R(X|C) \tag{7}$$

which is similar to what has been referred to as mutual α-entropy difference by Hero *et al*. [59] (who use a direct estimation of Renyi's entropy by means of minimal spanning trees rather than the "plug-in" density estimator [64] used here). This approximation of MI is invariant to a change in scale, is much simpler than the one suggested by Renyi [25] and is similar in form to Shannon's MI, except that each of the entropies is substituted with the non-parametric estimator for Renyi's entropy given by [27],

$$H_R(X) \cong -\log \frac{1}{N^2} \sum_{k=1}^{N} G(x(k) - x(k-1), 2\sigma^2) \tag{8}$$

where $G(x,\sigma^2)$ was previously defined in equation (3). The combination of equations (7) and (8) results in the proposed information-theoretic criterion for sequential feature extraction, which is referred to as Minimum/Maximum Renyi's Mutual Information using the Stochastic Information Gradient, or MRMI-SIG [28]. The overall cost function is given by equation (9), where the influence of R is from the relation, $x(k) = Rs(k)$. The SIG modification is an approximation that is used to reduce the complexity from $O(N_T^2)$ of the original algorithm to $O(N_T)$ of the present algorithm. It turns out that, if the training data is presented multiple times and if the time indices are randomized for each presentation, the SIG algorithm converges in the limit to the original algorithm [28]. Several other methods may also be used to reduce the complexity, including importance sampling [29], random sampling [30], clustering [30], sliding window [31], and GMM, Gaussian

Mixture Models [30]. If gradient ascent is used as the optimization method, the tap weight update equation becomes,

$$R(n+1) = R(n) + \eta \nabla_R (I(X;C)) \quad (10)$$

where the second term is the gradient of I(X;C) with respect to R and η is the step size. Notice that the class labels are used only to determine which set of training data is included in the second term of equation (9), so that there is no need to convert the (nominal) class labels into ratio values (where all values may be categorized as either nominal, ordinal, interval, or ratio [60]).

There are two weaknesses of this approach. First, there is no guarantee that maximizing equation (7) using Renyi's definition, is equivalent to maximizing (4) using Shannon's definition. Second, due to the difficulty of estimating pdf's in high dimensional spaces, the number of dimensions should be kept small. Notice, however, the dimensionality of the pdf estimation is not determined by the number of input features, but by the (smaller) number of output features.

MRMI-SIG was previously used in an unsupervised fashion to minimize the mutual information between a set of outputs for the application of blind source separation (BSS) [34]. There are several differences between the formulation above and that used for BSS. The criterion for subspace projection involves maximization instead of minimization, only N_O of the outputs of the rotation matrix are kept, mutual information is measured between the output feature set and the class label (instead of between the outputs) and the conditional (or joint) entropy does not disappear. Additional details on the entropy estimator given in equation (8), including proof of convergence when it is used for error entropy minimization, is given by Erdogmus *et al*. [28].

5 FEATURE REDUCTION ALGORITHMS

The previous section described the proposed information-theoretic algorithm, which uses the MRMI-SIG criterion. This section will briefly describe a second sequential method that makes use of an information-theoretic criterion, two benchmark methods, and four simultaneous feature reduction methods. All eight of these feature reduction algorithms are listed in Table 1. Note that when no name is available for a given feature reduction method, in order to prevent the inclusion of more terminology than is necessary, the algorithm will be referred to by the name of the criterion which it uses.

The second information-theoretic sequential method, ED-QMI-SIG, is a slight variation of a criterion published by Principe *et al*. [19] and used by Torkkola for subspace projections [36]-[38]. It is similar to MRMI-SIG in that it uses Parzen windows with Gaussian kernels for the pdf estimation. The difference between the two is the choice of the distance measure between the two relevant pdf's, which for MI are: (1) the joint pdf of X and C, and (2) the product of the two marginal pdf's. The distance measure for MRMI-SIG is loosely based on an approximation of Kullback-Leibler divergence [18], while ED-QMI-SIG uses Euclidean distance. The criterion for this method is given in equation (11).

The two benchmark methods are the well-known Principal Components Analysis (PCA) [13], and Linear Discriminant Analysis (LDA, which is an extension of Fisher's Discriminant Ratio [39]). These methods use only second-order statistics, and the transformation matrix R for both methods consists of a set of N_O eigenvectors of the appropriately defined matrix. As a result, the solution may be obtained analytically (i.e. non-iteratively). The main difference between these two methods is that PCA is unsupervised and LDA is supervised. In addition, the number of output features for LDA is restricted to be less than or equal to N_C, the number of classes.

There are a total of four simultaneous feature reduction algorithms considered. The first two of these are Feature Ranking (FR) and Feature Selection (FS). The difference is that FR evaluates each feature independently of the others, while FS con-

$$\arg\max_R \; -\log \frac{1}{N_T} \sum_{k=1}^{N_T} G(x(k) - x(k-1), 2\sigma^2) + \sum_{j=1}^{N_C} \left(\frac{N_j}{N_T} \log \frac{1}{N_j} \sum_{k=1}^{N_j} G(x_j(k) - x_j(k-1), 2\sigma^2) \right) \quad (9)$$

Table 1: Description Of The Feature Reduction Algorithms (excluding the classifier).

Algorithm	Type	Extraction Architecture	Extraction Criterion	Extraction Optimization
MRMI-SIG (Minimum/Maximum Renyi's Mutual Information - Stochastic Information Gradient)	Sequential	Linear Constrained (rotation)	Supervised Sequential Information theoretic Nominal classes	Iterative Gradient ascent
ED-QMI-SIG (Quadrature Mutual Information using Euclidean Distance - Stochastic Information Gradient)	Sequential	Linear Constrained (rotation)	Supervised Sequential Information theoretic Nominal classes	Iterative Gradient ascent
PCA (Principal Components Analysis)	Benchmark	Linear Constrained (rotation)	Unsupervised Sequential Second-order statistics N/A	Non-iterative (analytic)
LDA (Linear Discriminant Analysis)	Benchmark	Linear Constrained (rotation)	Supervised Sequential Second-order statistics Nominal classes	Non-iterative (analytic)
FR (Feature Ranking)	Simultaneous	Linear Constrained (0,1; a single 1 per row/column)	Supervised Simultaneous N/A Nominal classes	Iterative Brute force (by rank)
FS (Feature Selection)	Simultaneous	Linear Constrained (0,1; a single 1 per row/column)	Supervised Simultaneous N/A Nominal classes	Iterative Brute force (exhaustive search)
MSE (Mean Square Error)	Simultaneous	Linear Constrained (rotation)	Supervised Simultaneous Second-order statistics Assigns targets	Iterative Gradient descent
MCE (Minimum Classification Error)	Simultaneous	Linear Constrained (rotation)	Supervised Simultaneous Higher-order statistics Nominal classes	Iterative Gradient descent

siders all possible combinations of the N_I inputs, taken N_O at a time. Consequently, FS is expected to produce superior classification performance; however, the computational complexity suffers from combinatorial explosion. These methods produce at each output of the projection, a single, unweighted input feature. This differs from feature extraction, where each output feature is a weighted sum of all input features. This difference is completely characterized by the architecture, which for FR and FS is constrained to: (1) have only elements that are 0 or 1, and (2) have only one non-zero element in each row and in each column. Notice that, when this architecture is combined with either classifier, it is not possible to span the full space of linear projections. As a result, the performance may be needlessly reduced. In fact, it is trivial to construct a data set that will cause very poor performance for either method. The criterion for both is to minimize the (empirical) classification error.

The third simultaneous feature reduction method considered is the Mean Square Error (MSE) method, which trains the extractor "through" the classifier by using the MSE criterion at the output of the classifier. Notice that the outputs of both the Bayes-G and the Bayes-NP classifiers are ratios, unlike the classes which are necessarily nominal values, e.g. "person has diabetes" and "person does not have diabetes". Therefore, to compare the classifier outputs with the class labels, either the ratios (classifier outputs) need to be converted to nominal values (class labels) or vice versa. The former case requires training through the MAX operator of Fig. 1 and involves a *non-arbitrary* process. This was the method used for FR and FS and is emulated by the MCE method, to be described next. The latter case, which applies to MSE, requires an *arbitrary* assignment. More specifically, if the training does not take place through the MAX operator, then each class label must be converted to a rather arbitrarily chosen (N_C x 1) vector of ratios. This process is indicated in Fig. 1 by the DEMUX (de-multiplexer) operator.

The resulting criterion for this method may be expressed as a function of these targets as,

$$\arg\min_{R} \frac{1}{N_C N_T} \sum_{j=1}^{N_C} \sum_{k=1}^{N_T} (y_j(k) - \tau_j(k))^2 \qquad (12)$$

where $\tau_j(k)$ is the user-defined target for the j^{th} output of the classifier at time k and $y_j(k)$ is given by either equation (1) or (2). The classification performance for this method depends on how well the subspace projector/classifier combination can approach the targets, which is itself dependent on how well the targets are chosen. Unfortunately, it is not clear how the targets should be chosen to optimize performance of the overall system (although it is much less difficult than for defining targets in the feature space). The 1 of N_O scheme is commonly used, which is defined as setting the target associated with the correct class to 1, while all other targets are 0. If the MSE criterion were used to train the classifier, then this particular choice of targets produces the optimal (in the mean square sense) approximation of the a posteriori probabilities [11], [43], [62]. Note that the *classifiers* used here are not trained using MSE; instead, their outputs are determined using either equation (1) or (2). In this particular case, the concern is how to train the extractor. Since it also appears to be a reasonable choice of targets for training the extractor, the 1 of N_O scheme is used.

Minimum Classification Error (MCE) is the final feature reduction method considered. Notice that the term "minimum classification error", as before, is used for both the name of the method and for the name of the criterion which it employs. In addition, the expression is used to characterize several criteria, such as FR and FS, since they attempt to minimize the empirical classification error. The MCE criterion has also been referred to as Minimum Classification Error using Generalized Probabilistic Descent (MCE/GPD) [1], [3], Minimum Error Rate (MER) [33], and Discriminative Feature Extraction (DFE) [2], [44], [16]. It is also related to the Bayesian Back Propagation algorithm of Nedeljkovic [45]. The motivation behind this criterion is to approximate the decision process (the maximization in Fig. 1) using a function that is continuously differentiable. This allows a straightforward application of the stochastic gradient-based optimization method. This criterion can be described in the following manner. It modifies the parameters in an attempt to make the classifier output associated with the correct class have as much margin as possible with respect to the single classifier output having the largest value of all the outputs associated with the incorrect class. Mathematically, the criterion is given by equation (13) where,

$$1(x_j(k) \in c(k)) \qquad (14)$$

has a value of 1 when the correct class for $\boldsymbol{x}_j(k)$ is j and is 0 otherwise. There are two user-definable parameters, α and ν, for this criterion. As both approach infinity, the criterion becomes precisely the classification error.

There are many other options for feature reduction that are not included in the following comparison. For example, there is an extension to LDA that removes the restriction on the number of possible output features [46]. Projection pursuit could be used to find interesting projections [47], as could unsupervised clustering methods [39]. The maximization of the mutual information could be performed in a simultaneous manner, i.e. through the classifier. Either an SVM or MI could also be used to rank or select features [20]-[23], [48], [58]. For feature selection, greedy selection can be used to reduce the

$$\arg\max_{R} \frac{1}{N_T} \sum_{j=1}^{N_C} \sum_{k=1}^{N_j} G(x_j(k) - x_j(k-1), 2\sigma^2) + \frac{1}{N_T} \left(\sum_{j=1}^{N_C} \left(\frac{N_j}{N_T} \right)^2 \right) \sum_{k=1}^{N_T} G(x(k) - x(k-1), 2\sigma^2) + \frac{1}{N_T} \sum_{j=1}^{N_C} \frac{N_j}{N_T} \sum_{k=1}^{N_j} G(x_j(k) - x(k-1), 2\sigma^2) \qquad (11)$$

$$\arg\min_{R} \frac{1}{N_T} \sum_{k=1}^{N_T} \sum_{j=1}^{N_C} 1(x_j(k) \in c(k)) \left(1 + e^{-\alpha \left(-y_j(k) + \left(\frac{1}{N_C - 1} \sum_{\substack{i=1 \\ i \neq j}}^{N_C} y_i(k)^{\nu} \right)^{\frac{1}{\nu}} \right)} \right)^{-1} \qquad (13)$$

computational burden of an exhaustive search, producing a computational complexity between that of FR and FS. Greedy algorithms, which are suboptimal, come in three forms; viz., forward selection, backward elimination, and stepwise regression [20]. Another option, which is not always applicable, is a method known as "branch and bound". This approach is able to reduce the size of the exhaustive search space for FS without loss of optimality [49]. Other techniques include the construction of an MI matrix on which eigenanalysis is applied [22], the use frequency domain transforms [39], or the use of an unsupervised maximum entropy method [50].

6 COMPARISONS

There are three data sets used for the comparisons, the important characteristics of which are listed in Table 2. These are the Pima, Landsat, and Letter Recognition data sets. All three of these may be found at the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). The data was first pre-processed. This included centering the data, sphering the data, and, for two of the data sets (Pima and Landsat), reducing the original dimension (to N_I). The dimension was reduced using PCA by removing the eigendirections that had associated eigenvalues smaller than 0.5% that of the maximum eigenvalue. This same pre-processing was used for all the results included in this paper. One of the differences between the data sets is that the Pima data was determined to have outliers, which for purposes of this paper is defined as missing data points, e.g. when a feature has a value of 0 even though a value of 0 is not meaningful or physically possible. These outliers are points in feature space that are statistically distant from the mean calculated using the remaining data (with the points in question removed). The Pima data set will be used with all invalid points included, which has the benefit of helping to identify which feature reduction methods are sensitive to outliers.

The Bayes-G classifier was determined to provide the best classification performance for the Pima and Landsat data sets, while the Bayes-NP classifier was found to produce the best results for the Letter Recognition data set. Therefore, the results shown are restricted to these combinations of data sets and classifiers. It should be noted that, for the other combinations of data sets and classifiers, there is virtually no change in the order of the performance of the eight algorithms. When the Bayes-NP classifier is used, the kernel size, σ, is always set to 0.25, which was experimentally determined (using resubstitution) to be at or near the optimal value. The training was performed using the first N_T samples of the overall data set, and then tested on the remaining data. For the gradient-based methods, the number of Monte Carlo runs is set to 10. The initial conditions for the projector for each Monte Carlo repetition were initialized randomly (zero for the first run and, thereafter, normally distributed with a variance of 1). Randomizing the initial conditions was done instead of randomly selecting the training set in hopes that it will facilitate other researcher's attempts to duplicate the results and/or compare these results with their own work. This has the added benefit of simplifying the task of discerning which algorithms are susceptible to local minima. The step size for each method was chosen small enough so that, after convergence, the variance of the parameter values have no, or very little, effect on classification performance (convergence of every adaptation was verified manually). As a result, the variation of classification performance for each feature reduction method can be expected to be caused by local maxima/minima.

The following user-defined parameters were determined using resubstitution. The kernel size, σ, for

Table 2: Descriptions Of The Three Data Sets Used In The Comparison (after pre-processing)

Data Set	Input Dimension (N_I)	Classes	Training Size (N_T)	Test Size	Outlier %
Pima Indians Diabetes	4	2	500	268	49%
Landsat Satellite Image (Statlog)	8	6	4435	2000	0%
Letter Recognition	16	26	16,000	4000	0%

MRMI-SIG is set to 0.25, 0.35, and 0.5 when N_O is 1, 2-4, 5-8, respectively. Notice that the optimal value of σ increases as the number of output dimensions increases, partially offsetting the sparsity of data in high dimensions. Also notice that a fixed value of σ is used during training, which is in contrast to approaches by other researchers [51], [37], [19]. For ED-QMI-SIG, the resubstitution results suggest using a kernel size of 0.5, independent of N_O. For MCE, the ν parameter was somewhat arbitrarily set to 10 and the optimal smoothing parameter, α, was determined to be 2. It should be mentioned that several combinations of α and N_O were plagued by local minima. In particular, out of the 20 Monte Carlo runs associated with $N_O = 1$, $\alpha = 1$ and $N_O = 1$, $\alpha = 4$, the final classification performance (using resubstitution) was between 25-28% for 8 runs and between 64-65% for the remaining 12. Incidentally, local minima of the MCE algorithm can (if performed properly) be avoided by scheduling the value of the smoothing parameter, α. This can also be accomplished with the proposed information-theoretic criteria by annealing the kernel size, σ [52]. Since it is not known precisely how the scheduling should be done, and in order to simplify the experimental procedure, this is not done for either MCE or MRMI-SIG.

For the OH, the free parameter (which is referred to as the ν parameter by Mangasarian *et al*. [7]) was set to 0.1. For the SVM, pairwise training (as opposed to one vs. all) is used to combine the binary decisions for multi-class data. Unlike the other methods, the user-defined parameters for the SVM are not determined using resubstitution, since the resulting generalization performance was very poor. This is particularly true for the Pima data set, where the resubstitution results are very *negatively* correlated with the generalization results. This is possibly due to the known weakness of the SVM, for the case that there is missing data [9], [10]. The usual solution is to use cross-validation [39] in place of resubstitution. However, since the SVM is used as a benchmark, and to ensure that it cannot be claimed that the training of the SVM was insufficient, the parameters are chosen using the best *generalization* performance. This resulted in the following parameters: penalty = 0.8, $\sigma = 8$, for the Pima dataset, and penalty = 0.5, $\sigma = 2$ for Landsat (no results are given for the Letter Recognition data due to its size). The selection of the parameters by means of the generalization error gives the SVM a definite performance advantage over the other methods.

Not only is the criterion for each method optimized, but it is also "de-optimized" by maximizing instead of minimizing, or vice versa. This is done to see how well the criterion can manipulate the error rate in both directions and to give an indication of the worst performance possible, for sake of perspective. Random projections were also used, for which the coefficients were chosen uniformly in [-1,1]. These are always indicated by a dashed line in the following plots. Ideally, the results of the optimized criterion should be well above the results for random projections, and the results of the de-optimized criterion (signified in the plots by placing an asterisk after the name of the algorithm) well below the results for random projections. For PCA and LDA, the de-optimized results are found by selecting the eigenvectors associated with the minimum eigenvalues. For MCE, this was accomplished by maximizing the classification error rate and by changing ν from 10 to -10. The latter was done so that the margin for the output pertaining to the correct class was minimized with respect to the *smallest* output (of all incorrect classes). Figs. 2-4 show the plots of correct classification percentage, one for each data set, for both the optimized and de-optimized versions of each criterion. In each case, the left subplot shows the results for the PCA, LDA, ED-QMI-SIG and MRMI-SIG algorithms, as well as for the random projection. The right subplot shows the results for the FR, FS, MSE and MCE algorithms. Also shown on the right subplot, for sake of convenience, are the results for MRMI-SIG and the random projection. In all cases, when $N_O = N_I$, the results for all algorithms are the same. This is because both classifiers are invariant under full-rank linear transforms.

The first plot, given in Fig. 2, pertains to the Pima data set. There are several important items to notice in this figure. For example, it is noteworthy that the performance improves by *decreasing* the dimensionality. Since the projections to a given dimension include as a subset all projections to lower dimensions, the cause of this must be due to inaccurate estimation of the parameters, either of the projector

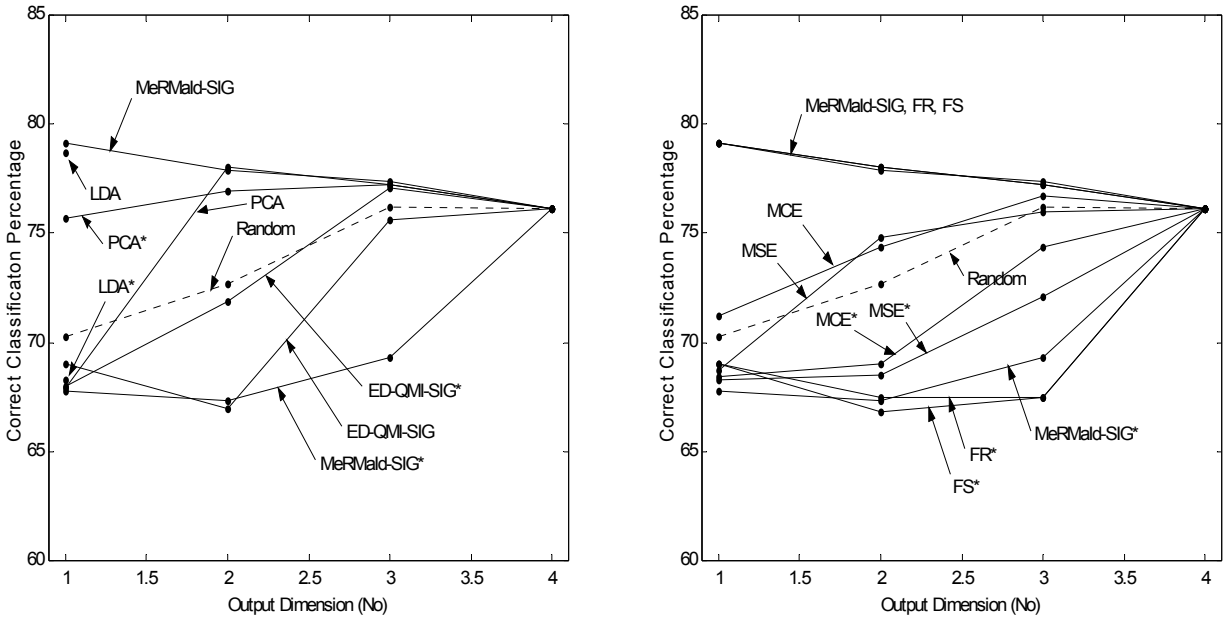

Fig. 2. Classification performance versus output dimension, N_O, for Pima data set.

or the classifier. This anomaly only occurred for the Pima data set, therefore it seems likely that the underlying cause is the existence of the outliers. Another possible explanation concerns the generalization of the results for the Pima data set since it is the smallest of the three; however, the ratio of free parameters to amount of training data lies between that of the other two data sets (where the number of free parameters is based on the Bayes-G classifier and $N_O = N_I/2$). Another unexpected result was that ED-QMI-SIG* (the "de-optimized" result) outperforms ED-QMI-SIG for $N_O = 2$ and $N_O = 3$. It could be that ED-QMI-SIG has local maxima whose performance is worse than that associated with one or more local minima, or that the "convergence" of several of the Monte Carlo runs was to a saddle point. In either case, the fact that this only occurred for the Pima data set, suggests the possibility that the ED-QMI-SIG algorithm is sensitive to outliers (an additional test, not included here, was performed that seems to verify this claim). Also notice that the result for both LDA and LDA* is only a single point, due to the limitation of the criterion as mentioned in Section 5. PCA* outperforms PCA for $N_O = 1$, which is not too peculiar considering that PCA is an unsupervised method. The Landsat data set is shown in Fig. 3. Since the number of classes for this data is 6, LDA can be calculated for $N_O \leq 5$. The results from these two plots indicate that the dimension can be reduced from 8 to 2 without loss of performance. Results for the Letter Recognition data set are shown in Fig. 4, for which a slight improvement is provided by reducing the dimensionality by one-half. Due to the size of this data set, the projections were trained only for $N_O = 1$, 2, 4, and 8. Notice that no results are shown for the FS algorithm for $N_O \geq 4$. This is due to the combinatorial explosion associated with this algorithm.

For all three data sets, as N_O approaches N_I, it becomes less and less important how the projection is chosen. This is shown in the figures in two different ways. First, the result for the random projection approaches the best result as N_O approaches N_I. Second, there is a tendency for the difference between the highest and lowest performance (for optimizing and de-optimizing, respectively) to approach zero as N_O approaches N_I. Consequently, the projection to $N_O = 1$ dimension is likely the most important data point in determining which algorithm performs the best. For generalization purposes, it is also important that an algorithm performs well for all output dimensions. Keep in mind, though, that when results are given that average over all possible output dimensions, the performance difference between the different algorithms

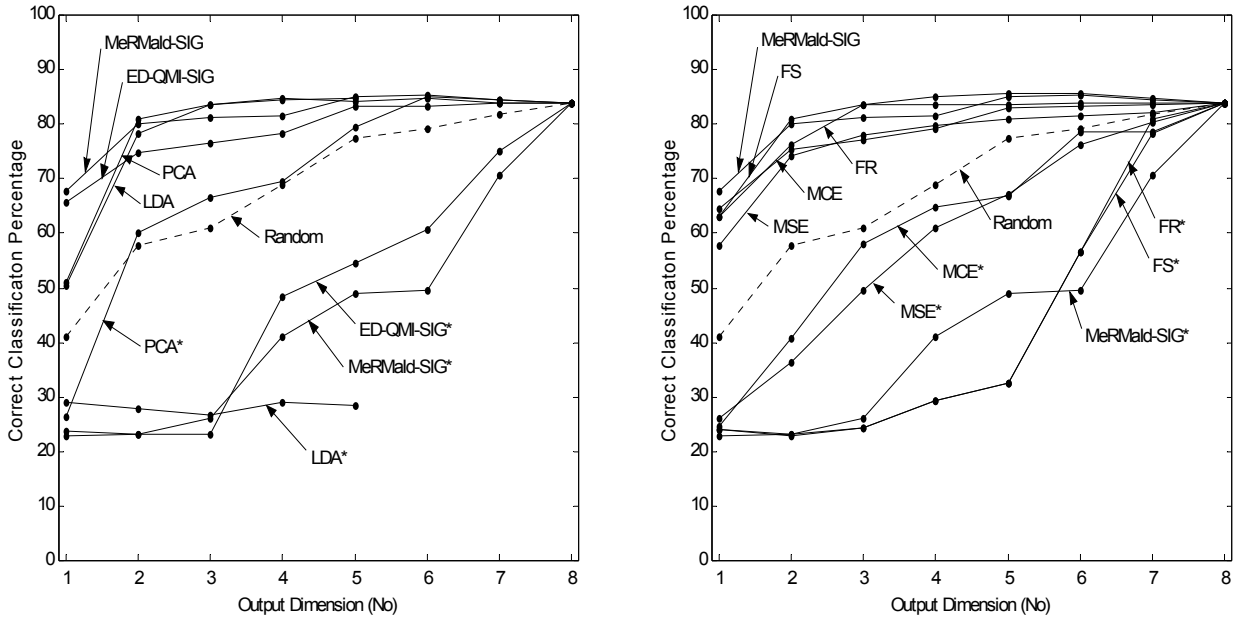

Fig. 3. Classification performance versus output dimension, N_O, for Landsat data set.

is necessarily deflated for the reason given above. Consequently, the overall results are shown in two different ways. The upper subplot in Fig. 5 shows the overall results for $N_O = 1$, averaged over all three data sets, where the mean value is shown in parentheses (results for de-optimization are not shown). The lower subplot of Fig. 5, on the other hand, shows the results for each algorithm averaged over all data sets and all output dimensions. In the former bar graph, FR and FS can be seen to have identical results. This is because the algorithms become identical for $N_O = 1$. In the latter bar graph, no results are shown for LDA or FS since they have missing data points (LDA because of the limitation on the number of outputs and FS because of the

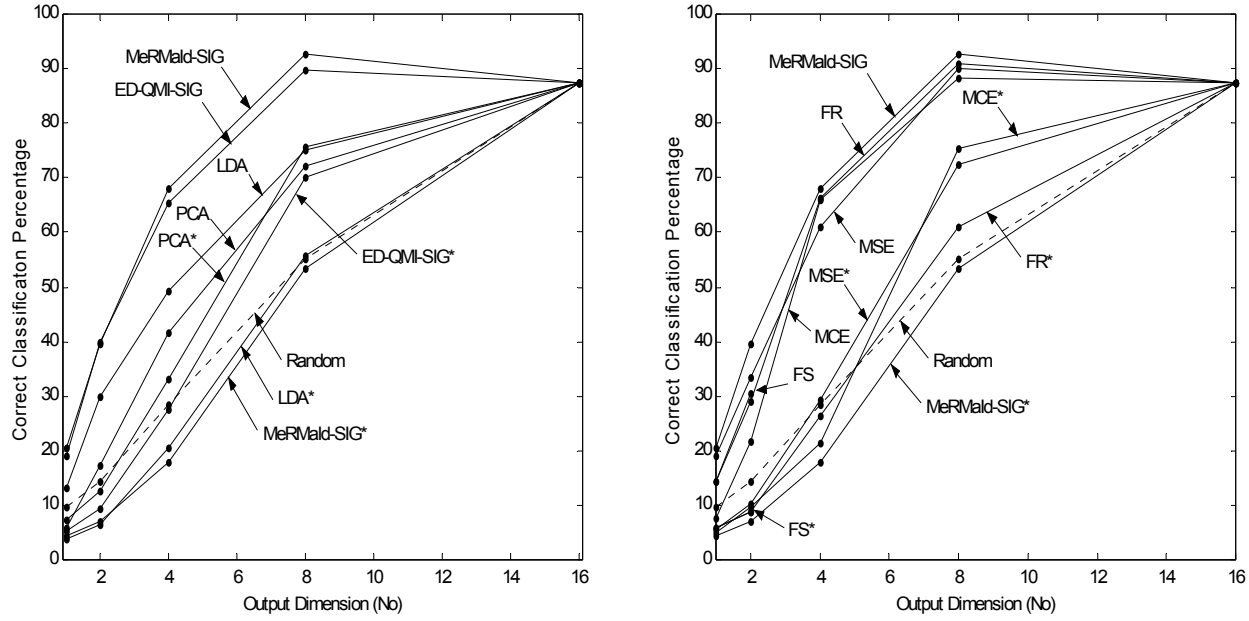

Fig. 4. Classification performance versus output dimension, N_O, for Letter Recognition data set.

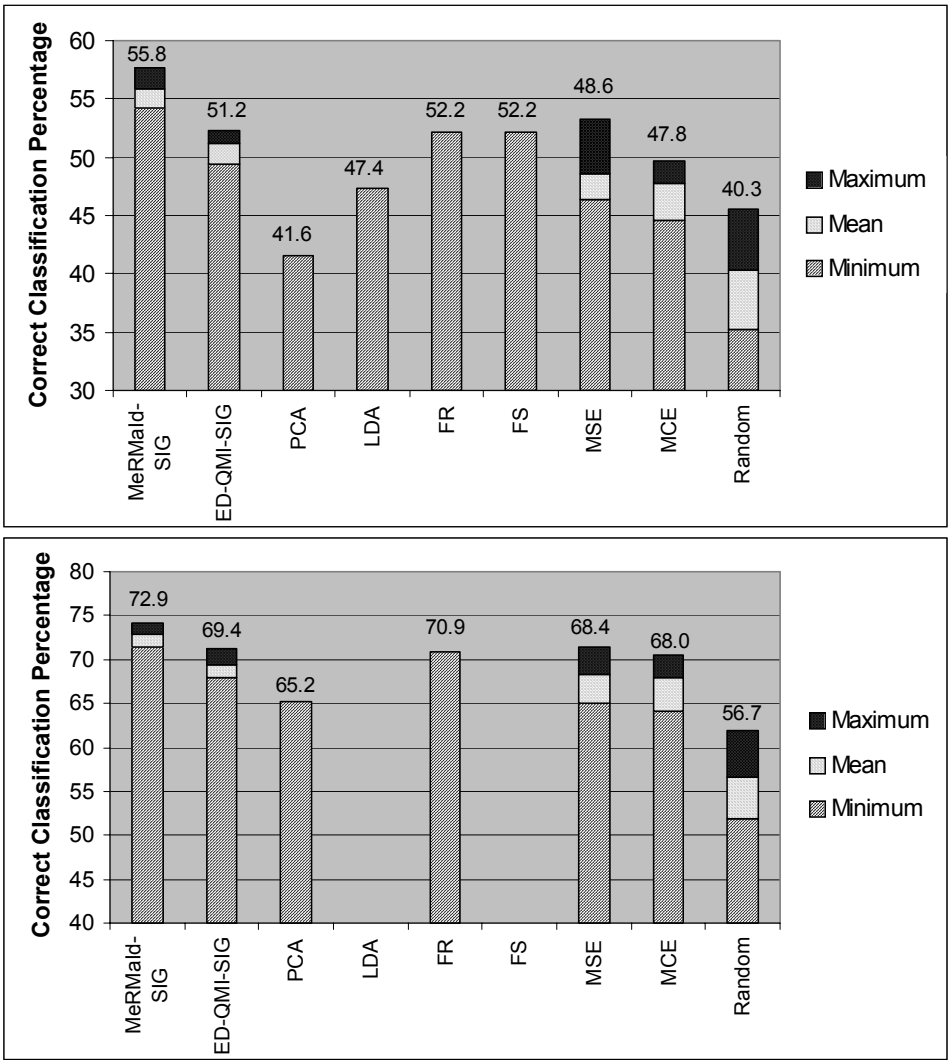

Fig. 5. Classification performance for all 3 data sets. The upper and lower subplots show results for $N_O = 1$ and the average over all values of N_O, respectively. The value above each bar is the mean classification.

training time involved), indicating that the two algorithms are not generally applicable.

For each of the gradient-based algorithms, Fig. 5 shows the minimum value, the mean value, and the maximum value of the 10 Monte Carlo runs. The performance for each of the gradient-based algorithms is not Gaussian distributed, otherwise $\pm$ 3 sigma values would be given. The reason that they are not Gaussian distributed is due to the existence of local minima/maxima. The largest difference between the minimum and maximum results is for the random projection, as expected. The algorithms having the next largest difference is the MSE and MCE algorithms, while the two MI methods are fairly consistent. These results indicate that the performance of the MSE and MCE algorithms are more sensitive to local optima than the MI methods. For the case that $N_O = 1$, the MRMI-SIG algorithm outperformed all other feature reduction methods by 4.6% to 14.2% (a relative increase of 9% to 34%). For the case that the results are averaged over all data sets and all output dimensions, MRMI-SIG outperformed all others by 2.0% to 7.7% (a relative increase of 3% to 12%).

It is interesting that the proposed sequential method has a mean classification error less than that produced by any of the simultaneous feature reduction methods, three of which use a criterion that *spe-*

cifically minimizes the classification error. This is, however, possible since the minimum classification error methods minimize the error on the finite *training* set, not on the disjoint finite *test* set. In order to prove that the error-based methods minimize the probability of error on the disjoint test set, "infinite training" is required, as acknowledged by Watanabe *et al*. [2] and Katagiri *et al*. [3]. This is essentially equivalent to knowing the underlying distributions, which would allow any of a number of methods to produce the optimum (Bayes) solution. While it seems perfectly reasonable to use this as a criterion when the data is finite, it is no longer guaranteed to be optimal. In fact, as suggested by the results presented here, it may very well be suboptimal because these methods focus only on training data that is near the decision boundary. This means that much of the already finite data is ignored, so that the results can be expected to be sensitive to outliers.

In order to see how the proposed feature reduction method compares with the SVM and the OH, these results are now given. Mixed in with these results are the best classification performances recorded in each of several different articles in the published literature (results shown without an associated reference are those obtained by the authors). For feature reduction methods, the value of N_O is also included. Keep in mind that the results for the MRMI-SIG algorithm are found using very simple classifiers, especially for the Pima and Landsat data, while the results from the literature are often the result of using sophisticated classifiers and/or training techniques, such as Boosting (which uses a committee of 3 learning machines) [53].

Pima: MRMI-SIG result is **79.1%** using a Bayes-G classifier with $N_O = 1$

- **~76%** using LVQ with $N_O = 1$ [54]
- **76.1%** using regularized AdaBoost [42]
- **76.6%** using Adaptive Margin SVM [42]
- **78.7%** using LVQ with $N_O = 3$ [36]
- **78.7%** using OH
- **79.5%** using SVM
- **~81%** using LVQ with $N_O = 6$ [54]

Landsat: MRMI-SIG result is **85.3%** using a Bayes-G classifier with $N_O = 6$

- **~78%** using LVQ with $N_O = 1$ [54]
- **80.4%** using an MLP with $N_O = 3$ [22]
- **83.8%** using OH
- **88.8%** using AdaBoost [10]
- **89.5%** using LVQ with $N_O = 15$ [30]
- **90.5%** using SVM
- **93.3%** using Bayesian (200 hidden units) [55]

Letter Recognition: MRMI-SIG result is **92.7%** using a Bayes-NP classifier with $N_O = 8$

- **79.3%** using an MLP (7 hidden layer nodes) [15]
- **~80%** using Holland-style (1190 rules) [56]
- **80.3%** using an MLP with $N_O = 15$ [22]
- **80.5%** using a Bayesian network [57]
- **83.2%** using OH
- **88.6%** using LVQ with $N_O = 8$ [30]
- **89.9%** using k-NN [15]
- **92.9%** using AdaBoost [10]

Besides classification performance, there are implementation issues that are relevant in algorithm selection. Two such issues include algorithmic complexity and length of time required for training. Complexity, in terms of computational requirements to update the parameters of the projection, is addressed first. In this regard, the PCA, LDA, FR, and FS methods have trivial complexity. The first two due to the existence of an analytical solution, the latter two because all the possible solutions are enumerated a priori (however, the number of solutions for FS may be very large). A good indication of the complexity of the gradient-based systems, on the other hand, is obtained by examining the criteria given previously in equations (9), (11), (12), and (13). Be mindful, the criteria for MSE and MCE are deceptively simple looking compared to those of the two MI methods. This is not so once one of the classifiers of equation (1) or (2) is inserted into (12) and (13). Also keep in mind that MCE and MSE are not as generally applicable, in that it is very difficult to use them with certain classifiers, such as k-NN. The time required for training is another important implementation issue, which is difficult to address since the experiment was not designed to minimize adaptation time for a given performance level. Nevertheless, the results of adaptation speed are given for the interested reader. These give a first-order estimate of the relative speed of the different methods. In all cases, the training (for a single Monte

Carlo run) took on the order of minutes or seconds. For the Letter Recognition data set, however, several of the algorithms took significantly longer. For example, when transforming to $N_O = 8$ output features, MRMI-SIG and ED-QMI-SIG took on the order of 2 hours, while the MSE and MCE methods took on the order of 24 hours. In addition, it was estimated that FS would take 3,075 hours, or 128 days, to complete.

7 CONCLUSION

The MRMI-SIG method has very interesting implementation characteristics: it has small relative complexity/high adaptation speed (compared to MCE, MSE and sometimes FS); it allows the use of different dimensionality in the class labels and network outputs (unlike MSE); it is not as sensitive to local optima relative to the other gradient-based methods (unlike MCE and MSE); it is independent of the classifier (unlike FR, FS, MSE, MCE); and, it appears to be robust to outliers (compared to ED-QMI-SIG, MCE, MSE, SVM, OH). In addition, it can be used to train nonlinear projectors [54], or used to train a classification system in a simultaneous manner. Also, the proposed method is non-parametric, does not require discretization of the data, and is computed directly from the samples. On the other hand, there are several shortcomings. It was mentioned earlier that there is no guarantee that it will produce the same solution as Shannon's MI; however, results seem to indicate that this is not an issue (notice that it is not claimed that the results indicate that the proposed method produces the same solution as Shannon's, only that it produces a useful solution). It takes longer to train (for large data sets) than PCA, LDA, or FR. Although the sensitivity to local minima is small, it is larger than for the SVM and all the methods that are not gradient-based (as expected). Also, as previously mentioned, there is a potential loss of performance as N_O increases due to the required high-dimensional density estimation (but only when $N_O << N_I$, otherwise any method is sufficient).

The goal of this article was to answer two questions. The answer to the first question, "*How does the performance of information-theoretic, sequential feature reduction methods compare to error-based, simultaneous methods?"* is "*favorably*". This conclusion is based on the classification performance and indicates that, for real (finite) data sets, samples not near the boundary carry information vital to the proper placement of the decision boundaries. Consequently, since simultaneous methods have a theoretical performance advantage, it would be interesting to use an MI criterion in a simultaneously-trained system (although this may prove to be prohibitively complex for most classifiers). The answer to the second question, "*How does the performance of the best feature reduction method considered here, when combined with a simple classifier, compare to the SVM and the closely related Optimal Hyperplane?*" is "*reasonably well*", especially considering that the parameters of the SVM were chosen using the best generalization performance. In fact, the results here indicate that it would be interesting to combine an SVM with a feature reducer since the former appears to be robust to outliers and may help alleviate the lack of robustness of an SVM (see Weston *et al*. [9]).

Much more theoretical work is required to validate the mutual information approach for feature extraction in classification. The fundamental difficulty is related to the implicit link between mutual information and classification error, where the only known results are expressed in the form of upper and lower bounds on the Bayes classification error. However, the tightness of the bounds remains unknown. Probably a more productive approach with IT learning is to reverse the question of tuning the classifier topology to the feature extractor, and seek classifiers that will meet the minimization of the Bayes error with MI-derived features. Another important line of research, where some encouraging results using the Information Bottleneck method have been published by Tishby [61], is the optimization of the dimensionality of the feature space.

ACKNOWLEDGEMENTS

Work partially supported by NSF ECS #9900394.

REFERENCES

[1] Biing-Hwang Juang and Shigeru Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. on Signal Proc.*, Vol. 40, No. 12, pp. 3043-3054, Dec. 1992.

[2] Hideyuki Watanabe, Tsuyoshi Yamaguchi, and Shigeru Katagiri, "Discriminative metric design for robust pattern recognition," *IEEE Trans. on Signal Proc.*, Vol. 45, No. 11, pp. 2655-2662, Nov. 1997.

[3] Shigeru Katagiri, Biing-Hwang Juang, and Chin-Hui Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, Vol. 86, No. 11, pp. 2345-2373, Nov. 1998.

[4] Vladimir Naumovich Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., Springer-Verlag, New York, NY, 2000.

[5] Thilo-Thomas Frieb, Nello Cristianini, and Colin Campbell, "The Kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines," *Intl. Conf. on Machine Learning*, Madison, WI, pp. 188-196, July 1998.

[6] Davide Mattera, Francesco Palmieri, and Simon Haykin, "An explicit algorithm for training support vector machines," *IEEE Signal Proc. Letters*, Vol. 6, No. 9, pp. 243-245, Sept. 1999.

[7] O.L. Mangasarian and David R. Musicant, "Lagrangian support vector machines," *Journal of Machine Learning Research*, Vol. 1, pp. 161-177, Mar. 2001.

[8] Thorsten Joachims, "Making Large-Scale SVM Learning Practical," *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola ed., MIT Press, 1999.

[9] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," *Advances in Neural Information Proc. Systems (NIPS '00)*, MIT Press, Cambridge, MA, pp. 668-674, Nov. 2000.

[10] Erin L. Allwein, Robert E. Schapire, and Yoram Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research, Vol. 1, pp. 113-141, Dec. 2000.*

[11] Chris Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[12] Emanuel Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, Vol. 33, No. 3, pp. 1065-1076, Sept. 1962.

[13] Jose C. Principe, Neil R. Euliano, and W. Curt Lefabvre, *Neural and Adaptive Systems*, John Wiley & Sons, Inc., New York, NY, 1999.

[14] William M. Campbell, Kari Torkkola, and Sreeram V. Balakrishnan, "Dimension reduction techniques for training polynomial networks," *Intl. Conf. on Machine Learning (ICML '00)*, Stanford, CA, pp. 119-126, June 2000.

[15] Shailesh Kumar, Joydeep Ghosh, and Melba Crawford, "A Bayesian pairwise classifier for character recognition," *Cognitive and Neural Models for Word Recognition and Document Processing*, Nabeel Mursheed (ed.), World Scientific Press, River Edge, NJ, 2000.

[16] Alain Biem, Shigeru Katagiri, and Biing-Hwang Juang, "Discriminative feature extraction for speech recognition," *Neural Networks for Signal Proc. (NNSP '93)*, Linthicum, MD, pp. 392-401, Sept. 1993.

[17] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, MD, 1996.

[18] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., New York, NY, 1991.

[19] Jose C. Principe, Dongxin Xu, Qun Zhao, and John W. Fisher III, "Learning from examples with information theoretic criteria," *Journal of VLSI Signal Proc. Systems*, Vol. 26, No. 1/2, pp. 61-77, Aug. 2000.

[20] Howard Hua Yang and John Moody, "Feature selection based on joint mutual information," *Advances in Intelligent Data Analysis (AIDA '99), Computational Intelligence Methods and Applications (CIMA), International Computer Science Conventions*, Rochester, NY, June 1999.

[21] Roberto Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. on Neural Networks*, Vol. 5, No. 4, pp. 537-550, July 1994.

[22] Kurt D. Bollacker and Joydeep Ghosh, "Mutual information feature extractors for neural classifiers," *Intl. Conf. on Neural Networks (ICNN '96)*, Washington DC, *pp. 1528-1533, June 1996.*

[23] Nojun Kwak and Chong-Ho Choi, "Improved mutual information feature selector for neural networks in supervised learning," *Intl. Joint Conf. on Neural Networks (IJCNN '99)*, Washington, DC, Vol. 2, pp. 1313-1318, July 1999.

[24] R. Rajagopal, K. Anoop Kumar, and P. Ramakrishna Rao, "An integrated approach to passive target classification," *Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '94)*, Adelaide, Australia, Vol. 2, pp. 313-316, Apr. 1994.

[25] A. Renyi, *Probability Theory*, North-Holland Publishing Company, Amsterdam, Netherlands, 1970.

[26] Dongxin Xu, Jose C. Principe, John Fisher III, Hsiao-Chun Wu, "A novel measure for independent component analysis (ICA)," *Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Seattle, WA, Vol. 2, pp. 1161-1164, May 1998.

[27] Kenneth E. Hild II, Deniz Erdogmus, and Jose C. Principe, "On-line Minimum Mutual Information Method For Time-varying Blind Source Separation," *Intl. Workshop on Independent Component Analysis and Signal Separation, (ICA '01)*, San Diego, CA, pp. 126-131, Dec. 2001.

[28] Deniz Erdogmus, Kenneth E. Hild II, and Jose C. Principe, "On-Line Entropy Manipulation: Stochastic Information Gradient," *accepted to IEEE Signal Processing Letters*, Oct. 2001.

[29] Keinosuke Fukunaga and Raymond R. Hayes, "The reduced parzen classifier," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 4, pp. 423-425, Apr. 1989.

[30] Kari Torkkola, "Learning discriminative feature transforms to low dimensions in low dimensions," *Advances in Neural Information Proc. Systems (NIPS '01)*, MIT Press, Cambridge, MA, Dec. 2001.

[31] Deniz Erdogmus, Kenneth E. Hild II, and Jose C. Principe, "Blind Deconvolution of Linear Channels by Minimizing or Maximizing Renyi's Entropy," submitted to *IEEE Trans. on Signal Proc.*, Jun 2002.

[32] Brian D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, UK, 1995.

[33] Qi Li and Biing-Hwang Juang, "A new algorithm for fast discriminative training," *Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '02)*, Orlando, FL, Vol. 1, pp. 97-100, May 2002.

[34] Kenneth E. Hild II, Deniz Erdogmus and Jose C. Principe, "Blind source separation using Renyi's Mutual Information," *IEEE Signal Proc. Letters*, Vol. 8, No. 6, pp. 174-176, June 2001.

[35] Deniz Erdogmus and Jose C. Principe, "Lower and upper bounds for misclassification probability based on Renyi's Information," accepted to *Journal of VLSI Signal Processing Systems Special Issue on Wireless Communications and Blind Signal Processing* (invited), Dec 2001.

[36] Kari Torkkola and William M. Campbell, "Mutual information in learning feature transformations," *Intl. Conf. on Machine Learning (ICML '00)*, Stanford, CA, pp. 1015-1022, June 2000.

[37] Kari Torkkola, "Visualizing class structure in data using mutual information," *Neural Networks for Signal Proc. (NNSP '00)*, Sydney, Australia, pp. 376-385, Dec. 2000.

[38] Kari Torkkola, "On feature extraction by mutual information maximization," *Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '02)*, Orlando, FL, pp. 821-825, May 2002.

[39] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition*, Academic Press, San Diego, CA, 1999.

[40] Colin Campbell, Thilo-Thomas Frieb, and Nello Cristianini, "Maximum margin classification using the KA algorithm," *Intelligent Data Engineering and Learning (IDEAL '98)*, Hong Kong, pp. 355-362, Oct. 1998.

[41] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, No. 3, pp. 273-297, Sept. 1995.

[42] Ralf Herbrich and Jason Weston, "Adaptive margin support vector machines for classification," *Intl. Conf. on Artificial Neural Networks (ICANN '99)*, Edinburgh, UK, pp. 880-885, Sept. 1999.

[43] Michael D. Richard and Richard P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, Vol. 3, No. 4, pp. 461-483, Winter 1991.

[44] Alain Biem, Shigeru Katagiri, and Biing-Hwang Juang, "Pattern recognition using discriminative feature extraction," *IEEE Trans. on Signal Proc.*, Vol. 45, No. 2, pp. 500-504, Feb. 1997.

[45] Vladimir Nedeljkovic, "A novel multilayer neural networks training algorithm that minimizes the probability of classification error," *IEEE Trans. on Neural Networks*, Vol. 4, No. 4, pp. 650-659, July 1993.

[46] T. Okadada and S. Tomita, "An optimal orthonormal system for discriminant analysis," *Pattern Recognition*, Vol. 18, No. 2, pp. 139-144, 1985.

[47] Jerome H. Friedman, "Exploratory projection pursuit," *Journal of the American Statistical Association*, Vol. 82, No. 397, pp. 249-266, March 1987.

[48] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik, "Gene selection for cancer classification using Support Vector Machines," *Machine Learning*, Vol. 46, No. 1-3, pp. 389-422, Jan.-Mar. 2002.

[49] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, Inc., Boston, MA, 1990.

[50] John W. Fisher III and Jose C. Principe, "A methodology for information theoretic feature extraction," *Intl. Joint Conf. on Neural Networks (IJCNN '98)*, Anchorage, AK, Vol. 3, pp. 1712-1716, May 1998.

[51] Qun Zhao and Jose C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 37, No. 2, pp. 643-654, Apr. 2001.

[52] Deniz Erdogmus and Jose C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," accepted to *IEEE Trans. on Neural Networks*, Sept. 2002.

[53] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee, "Boosting the Margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, Vol. 26, No. 5, pp. 1651-1686, May 1998.

[54] Kari Torkkola, "Nonlinear feature transforms using maximum mutual information," *Intl. Joint Conf. on Neural Networks (IJCNN '01),* Washington, DC, *pp. 2756-2761, July 2001.*

[55] Nello Cristianini, John Shawe-Taylor, and Peter Sykacek, "Bayesian classifiers are large margin hyperplanes in a Hilbert space," *Intl. Conf. on Machine Learning (ICML '98)*, Madison, WI, pp. 109-117, July 1998.

[56] P. W. Frey, and D. J. Slate, "Letter recognition using Holland-style adaptive classifiers", *Machine Learning,* Vol. 6, No. 2, 161–182, March 1991.

[57] Moninder Singh and Marco Valtorta, "Construction of Bayesian Belief Networks from Data: a Brief Survey and an Efficient Algorithm", *International Journal of Approximate Reasoning*, Vol. 12, No. 2, Feb. 1995, pp. 111-131.

[58] Dongming Xu and Jose C. Principe, "Feature evaluation using quadratic mutual information," *Intl. Joint Conf. on Neural Networks (IJCNN '01)*, Washington, DC, Vol. 1, pp. 459-463, July 2001.

[59] Alfred O. Hero, Bing Ma, Olivier Michel and John Gorman, "Alpha-divergence for classification, indexing and retrieval (revision 2)," *Technical Report CSPL-328*, Communications and Signal Processing Laboratory, The University of Michigan, pp. 1-25, June 2002.

[60] Jens Blauert, *Spatial Hearing, The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA, 1983.

[61] Noam Slonim and Naftali Tishby, "Agglomerative Information Bottleneck," *Advances in Neural Information Proc. Systems (NIPS '99)*, MIT Press, Cambridge, MA, pp. 617-623, Nov. 1999.

[62] Chanchal Chatterjee and Vwani Roychowdhury, "Statistical risk analysis for classification and feature extraction by multilayer perceptrons," *Intl. Conf. on Neural Networks (ICNN '96)*, Washington, DC, Vol. 3, pp. 1610-1615, June 1996.

[63] Martin E. Hellman and Josef Raviv, "Probability of error, equivocation, and the Chernoff Bound," *IEEE Trans. on Information Theory*, Vol. IT-16, No. 4, pp. 368-372, July 1970.

[64] J. Beirlant, E. J. Dudewica, L. Gyofi, and E. van der Meulen, "Non-parametric entropy estimation: An overview," *Intl. Journal of Math. Stat. Sci*., Vol. 6, No. 1, pp. 17--39, 1997.