# SVM Decision Boundary Based Discriminative Subspace Induction

Jiayong Zhang and Yanxi Liu

CMU-RI-TR-02-15

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

## Abstract

Dimensionality reduction is widely accepted as an analysis and modeling tool to deal with high-dimensional spaces, although researches from different disciplines have different interpretations of what properties should be preserved in the reduction process. We study the problem of linear dimension reduction for classification, with a focus on sufficient dimension reduction, i.e., inducing subspaces without loss of discriminative information. Decision boundary analysis (DBA), originally proposed by Lee & Landgrebe (1993), can directly find the smallest subspace with such property. However, existing DBA implementations are computationally expensive and sensitive to sample size. In this paper, we first formulate the problem of sufficient dimension reduction for classification in parallel terms as for regression. Disclosures of these connections lead to several meaningful observations. Then we present a novel space reduction algorithm that combines SVM and DBA, thus inheriting several appealing properties from kernel machines such as good generalization, weak assumption, and efficient computation. In addition, the proposed method provides a natural way to reduce the complexity, and even improve the accuracy, of SVM itself. We demonstrate its superiority by comparative experiments on one simulated and four real-world benchmark datasets.

# 1 Introduction

Dimensionality reduction is widely accepted as an analysis and modeling tool to deal with high-dimensional spaces. There are several reasons to keep the dimensionality as low as possible, such as to reduce system complexity, to alleviate "curse of dimensionality" [3, 6], and to enhance understanding of the data. In general, dimensionality reduction can be defined as the search for a low-dimensional linear or nonlinear subspace that preserves some intrinsic properties of the original high-dimensional data. However, what "properties" should be preserved in the reduction process is interpreted differently in different research disciplines. At least we can identify three cases:

1. *Visualization and exploration*, where the challenge is to embed a set of high-dimensional observations into a low-dimensional Eucledian space that preserves as closely as possible their intrinsic global/local metric structure [43, 71, 65].
2. *Regression*, in which the goal is to reduce the dimension of the predictor vector with the minimum loss in its capacity to infer about the conditional distribution of the univariate response variable [53, 54, 17].
3. *Classification*, where we should seek reductions that minimize the lowest attainable classification error in the transformed space, i.e., the Bayes error [13].

Such disparate interpretations might thereby cast strong influence on both the design and the choice of an appropriate dimensionality reduction algorithm for a given problem as far as the reduction optimality is concerned.

In this paper we study the problem of dimensionality reduction for classification, which is commonly referred to in pattern recognition literatures as feature extraction [28, 44]. Particularly, we restrict ourselves to linear dimension reduction, i.e., inducing subspaces through linear mappings. Linear mapping is mathematically tractable and computationally simple, with certain regularization ability that sometimes makes it outperform nonlinear models. In addition, it may be nonlinearly extended, for example, through global coordinations of local linear models (e.g. [15, 29]) or kernel mapping (e.g. [69, 60]).

For recognition systems, inducing subspaces without any loss of discrimination power is especially attractive. We call this process *sufficient dimension reduction*, borrowing terminology from classical statistics. Among existing linear reduction algorithms (see Appendix A for a review), few has formally incorporated the notion of sufficiency. Recently, several methods were proposed trying to optimize some projection indices that either approximate or bound the Bayes error [56, 67, 72], yet they involve time-consuming iterations and require a given output dimension. As an exception, Lee & Landgrebe [51] found that, "the necessary feature vectors to achieve the same classification accuracy as in the original space" can be uniquely determined by the Bayes optimal decision boundary. Thus they suggested a general procedure, which we call *decision boundary analysis* (DBA), to predict these necessary feature vectors through some estimation of the decision boundary. At first glance, this approach seems questionable because one of the motivations for dimension reduction is just that we cannot accurately estimate the boundary in high-dimensional space. Later studies (e.g. [45, 49]) also confirmed that DBA is expensive in time and sensitive to sample size.

However, recent discovery and elaboration of kernel methods for classification and regression seem to suggest that learning in very high dimensions is not necessarily a terrible mistake. Several feature selection methods [10, 77, 31], as special cases of linear dimension reduction, have been shown to be successful which directly depend on the intrinsic generalization ability of kernel machines in high dimensional spaces. In the same spirit, we will demonstrate that, the marriage of DBA and kernel methods may lead to superior performance dimension reduction algorithms that share the appealing properties of both.

Our contribution in this work is twofold. First, we formulate the sufficient reduction problem for classification in parallel terms as for regression in recent statistics literatures. This is not surprising since regression and discrimination are inherently similar and can be seen as particular cases of function approximation. Disclosures of such connections lead to several meaningful observations, and there are interesting possibilities for cross-fertilization. Second, we present a novel dimension reduction method called SVM-DBA. The maximum margin boundary provided by SVM has been proven to be optimal in a structural risk minimization sense, which enables SVM-DBA to readily deal with sparse data in high-dimensional space. As the boundary is represented in a closed form by a few support vectors, the computational cost and estimation errors of DBA can be significantly reduced. Instead of requiring a given output dimension, our method generates a series of nested subspaces that enables easy explorations of the tradeoff between classification accuracy and subspace compactness. In addition, experiments show that the accuracy of SVM itself may be significantly improved via subspace induction when irrelevant and/or redundant features exist.

The rest of the paper is organized as follows. We first formulate the problem of sufficient dimension reduction for classification in Section 2. Its correspondence to the similar problem in context of regression and the basic ideas of DBA are covered. We then describe in detail the proposed SVM-DBA algorithm in Section 3, followed by empirical comparative results in Section 4 and some comments on related works in Section 5. We conclude the paper in Section 6, with discussions on some possible extensions. A short review on supervised linear dimension reduction for classification and the detailed information of the benchmark datasets employed in the empirical study are given in the appendices.

## 2 Discriminative subspaces for classification

Consider a $Q$-class classification problem. We are given some joint distribution $P(x, y)$, where $x$ is a $d$-dimensional random vector, $x \in \mathbb{R}^d$, and $y$ is a discrete-valued random variable, $y \in K = \{k\}_{k=1}^{Q}$. Therefore, $P(x, y) : \mathbb{R}^d \times K \to R$. Let $U$ denote a fixed $d \times m$, $(m < d)$, matrix, $\mathcal{S}(U)$ denote the subspace of $\mathbb{R}^d$ spanned by the $m$ column vectors of $U$. The notation $u \perp\!\!\!\perp v|z$ means that $P(u, v|z) = P(u|z)P(v|z)$, i.e., random vectors $u$ and $v$ are independent given any value for the random vector $z$.

We are interested in finding the linear mapping $U$ such that $\mathcal{S}(U) \times K$ contains the same amount of discriminative information as $\mathbb{R}^d \times K$. From Bayes decision theory we know that, given a loss matrix $C = [c_{jk}]_{Q\times Q}$, this discriminative information can be characterized by the expected Bayes risk $r$:

$$r = \int \left[ \min_{1\le j\le Q} r_j(y|x) \right] P(x)\, dx\,, \tag{1}$$

where

$$r_j(y|x) = \sum\nolimits_{1\le k\le Q, k\ne j} c_{jk} P(k|x)\,. \tag{2}$$

It is easy to show that the expected Bayes risk in the original space is the same, with arbitrary loss matrix, as in the projected subspace if and only if the projection preserves the *a posteriori* probability distribution $P(y|x)$ at any point $x$. In other words, all the points that are mapped into a point in $\mathcal{S}(U)$ should have the same *a posteriori* probability distribution. This condition can be formalized by the following definition.

**Definition 1** *If $y \perp\!\!\!\perp x|U^T x$, then $\mathcal{S}(U)$ is a sufficient discriminative subspace (SDS) for the $P(x, y)$ classification problem.*

This statement is equivalent to say that $P(y|x)$ is the same as $P(y|U^T x)$ for all values of $x$ in its marginal sample space according to the next proposition.

**Proposition 2.1** *The following two statements are equivalent:*

*(i)* $y \perp\!\!\!\perp x|U^T x$,

*(ii)* $P(y|U^T x) = P(y|x)$, $\forall x \in \mathbb{R}^d$ *and* $P(x) > 0$.

**Proof** According to the symbol definition of $\perp\!\!\!\perp$, (i) means $P(y, x|U^T x) = P(y|U^T x)P(x|U^T x)$. On the other hand, we know from the chain rule that $P(y, x|U^T x) = P(y|x, U^T x)P(x|U^T x)$. The equivalence between (i) and (ii) follows immediately considering the fact that $P(y|x, U^T x) = P(y|x)$. $\blacksquare$

Thus the definition of SDS implies that the $d \times 1$ feature vector $x$ can be replaced by the $m \times 1$ vector $U^T x$ without any loss of discriminative information. The concept of SDS is potentially useful because it represents a "sufficient" reduction in the dimension of the feature vector. Knowledge of the smallest SDS would be clearly valuable for characterizing the intrinsic properties of the classification problem. Let $\mathcal{S}_{y|x}$ denote the smallest SDS. We call $\mathcal{S}_{y|x}$ *intrinsic discriminative subspace* (IDS), and the dimension $d = \dim(\mathcal{S}_{y|x})$ *intrinsic discriminative dimension* (IDD).

In many applications we are only concerned with the Bayes error $\epsilon$, which equals to the Bayes risk with zero-one loss function:

$$\epsilon = \int [1 - \max_{1\le k\le Q} P(k|x)]P(x)\,dx\,. \tag{3}$$

It can be shown that, in order to leave Bayes error unchanged in a transformed space, only the Bayes-rule assignment at each point need to be preserved. Let $f(x)$ denotes the Bayes minimum error decision rule

$$f(x) = \arg\max_{1\le k\le Q} P(k|x). \tag{4}$$

We assume that $f(x)$ exists almost everywhere.

**Definition 2** *If $y \perp\!\!\!\perp f(x)|U^T x$, then $\mathcal{S}(U)$ is a max sufficient discriminative subspace (MSDS) for the $P(x,y)$ classification problem.*

The next proposition gives equivalent conditions for the conditional independence used in Definition 2.

**Proposition 2.2** *The following statements are equivalent:*

*(i)* $y \perp\!\!\!\perp f(x)|U^T x$,

*(ii)* $P(y|f, U^T x) = P(y|U^T x)$,

*(iii)* $f(x)$ *is a function of* $U^T x$, *or in other words,* $Var(f|U^T x) = 0$.

**Proof** That (i) implies (ii) is immediate. That (iii) implies (i) is also immediate, because, if $f(x)$ is a function of $U^T x$, then, given $U^T x$, $f(x)$ is a constant and hence independent of any other random variable. Now let's prove that (ii) implies (iii).

Suppose $\exists\, U^T x = \eta \in \mathcal{S}(U)$ with $P(U^T x = \eta) > 0$ such that $Var(f|U^T x = \eta) \neq 0$, then, $\exists\, \alpha,\beta \in K$ and $\alpha \neq \beta$, such that $P(f = \alpha|U^T x = \eta) \cdot P(f = \beta|U^T x = \eta) > 0$, or $P(f = \alpha, U^T x = \eta) \cdot P(f = \beta, U^T x = \eta) > 0$. According to the definition of $f(x)$, the inequality

$$P(y = \alpha|x, f(x) = \alpha, U^T x = \eta) > P(y = \beta|x, f(x) = \alpha, U^T x = \eta) \tag{5}$$

consistently holds when $P(x|f(x) = \alpha, U^T x = \eta) > 0$. It follows that

$$P(y = \alpha|f(x) = \alpha, U^T x = \eta) > P(y = \beta|f(x) = \alpha, U^T x = \eta)\,. \tag{6}$$

Similarly we have

$$P(y = \alpha|f(x) = \beta, U^T x = \eta) < P(y = \beta|f(x) = \beta, U^T x = \eta)\,. \tag{7}$$

Combining the above two inequalities, we get

$$P(y|f(x) = \alpha, U^T x = \eta) \neq P(y|f(x) = \beta, U^T x = \eta)\,, \tag{8}$$

and thus

$$P(y|f(x), U^T x = \eta) \neq P(y|U^T x = \eta)\,, \tag{9}$$

which contradicts to (ii). ∎

It follows from this definition that a SDS is necessarily a MSDS, because $y \perp\!\!\!\perp x|U^T x$ implies $y \perp\!\!\!\perp f(x)|U^T x$. Let $\mathcal{S}_{f(x)|x}$ denote the smallest MSDS. We call $\mathcal{S}_{f(x)|x}$ *max intrinsic discriminative subspace* (MIDS), and the dimension $d = \dim(\mathcal{S}_{f(x)|x})$ *max intrinsic discriminative dimension* (MIDD). Obviously we have $\mathcal{S}_{f(x)|x} \subseteq \mathcal{S}_{y|x}$.

Now the following two problems arise naturally: 1) How to determine the optimal discriminative dimension? 2) How to find the corresponding optimal subspace?

Decision boundary analysis (DBA) is a technique that can directly estimate MIDS and MIDD. For a two-class problem, define a discriminant function

$$h(x) = P(C_1|x) - P(C_2|x)\,, \tag{10}$$

thus $h(x) = 0$ represents the Bayes optimal decision boundary $\mathcal{B}$. Since any surface normal $N(s) = \nabla h(s)$, $s \in \mathcal{B}$, belongs to MIDS of the problem [51], it is natural to apply PCA for estimating this subspace. We can compute the so-called *decision boundary scatter matrix* (DBSM) as

$$M = \int_{\mathcal{B}} N(s) N^T(s) p(s)\, ds\,. \tag{11}$$

The eigen decomposition of $M$, $M = \Psi^T \Lambda \Psi$ where $\Psi$ is an orthonormal matrix and $\Lambda$ is a diagonal matrix with decreasing eigenvalues, delivers several important information:

1. the number of non-zero eigenvalues, $m = \dim(\Lambda)$, corresponds to MIDD,
2. the first $m$ eigenvectors of $M$ (i.e. the first $m$ columns of $\Psi$) provide an orthonormal basis of MIDS,
3. the non-zero eigenvalues show how much $h$ varies in each direction.

When given finite samples, $M$ can be replaced by the estimate

$$\hat{M} = \sum_{i=1}^{l} \hat{N}(\hat{s}_i)\, \hat{N}(\hat{s}_i)^T / l\,, \tag{12}$$

where $\{\hat{s}_i\}_{i=1}^{l}$ are $l$ points sampled from the estimated decision boundary. DBA can be readily extended to multi-category problems by repeating the procedure in either one-vs-all or pairwise mode. For the latter case, we may employ the following weighted average DBSM:

$$M = \sum_{j=1}^{Q} \sum_{k=1, k\neq j}^{Q} w_{jk} M_{jk} \tag{13}$$

Table 1: A one-to-one correspondence between concepts of sufficient dimension reduction for discrimination and those for regression, where $y$ denotes a continuous response variable.

| Discrimination | Regression | Description |
|---|---|---|
| *sufficient discriminative subspace* (SDS) | *dimension-reduction subspace* | Any subspace $\mathcal{S}(U)$ such that $y \perp\!\!\!\perp x \mid U^T x$. |
| *intrinsic discriminative subspace* (IDS) | *central subspace* | Intersection of all dimension-reduction subspaces. |
| *intrinsic discriminative dimension* (IDD) | *structural dimension* | Dimension of the central subspace. |
| *max sufficient discriminative subspace* (MSDS) | *mean dimension-reduction subspace* | Any subspace such that $y \perp\!\!\!\perp E(y\mid x) \mid U^T x$. |
| *max intrinsic discriminative subspace* (MIDS) | *central mean subspace* | Intersection of all mean dimension-reduction subspaces. |

where $M_{jk}$ is the DBSM between class $j$ and $k$.

The key problem of DBA is the accurate estimate of decision boundary. Several estimators have been used in previous DBA implementations, including Quadratic Discriminant [51], Parzen window density estimator [50] and BPNN [52]. However, later studies (e.g. [45, 49]) show that DBA is expensive in time and sensitive to sample size.

We note that the subspace definitions described above have neat counterparts in recent literatures on regression [53, 17, 18]. This is not surprising since regression and discrimination are inherently similar and can be seen as particular cases of function approximation. Such correspondences are summarized in Table 1. We also note that the DBA estimator is in essence similar to the so-called average derivative estimation (ADE) which were first introduced in [70] and [63]. ADE has been used to estimate the central mean subspace of a single-index or multi-index regression model [66, 38]. The only difference between ADE and DBA is that, in ADE, surface normals are replaced by the gradient $F(x) = \nabla f(x)$ of the regression function at every point $x$. The main issue of ADE is again how to more accurately estimate the derivatives. Various other methods have been developed to estimate the central (mean) subspace, including ordinary least squares (OLS), sliced inverse regression (SIR, [53]), principle Hessian directions (pHd, [54, 16]), graphical regression (GREG) and sliced average variance estimation (SAVE, [17]). Readers may refer to, e.g. [17], for a comprehensive introduction.

Two observations immediately follow discussions in this section:

1. The difference between IDS and MIDS is significant only when the loss matrix is arbitrary. To our knowledge, no existing discriminative dimension reduction algorithms is considered under the arbitrary loss matrices assumption. DBA can be extended to estimate IDS, for example, by averaging DBFMs associated with different loss matrices;

2. Since SVM is also very good at estimating regression functions, adapting SVM-DBA to central mean subspace estimation should be straightforward.

# 3 SVM-DBA algorithm

SVM combines three important ideas into a very successful whole: quadratic programming for convex optimization, kernel representation, and maximum-margin. For a complete coverage, readers are referred to, e.g. [76, 19, 35].

The decision function of a two-class problem derived by SVM can be written as

$$h(x) = w \cdot \Phi(x) + b = \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\,, \tag{14}$$

where $x_i \in \mathbb{R}^d$ is the training sample, and $y_i \in \{\pm 1\}$ is the class label of $x_i$. A transformation $\Phi(\cdot)$ maps the data points $x$ of the input space $\mathbb{R}^d$ into a higher dimensional feature space $\mathbb{R}^D, (D \geq d)$. The mapping $\Phi(\cdot)$ is performed by a kernel function $K(\cdot,\cdot)$ which defines an inner product in $\mathbb{R}^D$. The parameters $\alpha_i \geq 0$ are optimized by finding the hyperplane in feature space with maximum distance to the closest image $\Phi(x_i)$ from the training set, which reduces to solving the following linearly constrained convex quadratic program:

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i \tag{15}$$

$$s.t. \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{16}$$

$$0 \leq \alpha_i \leq C,\ i = 1, \ldots, n \tag{17}$$

where $C$ is the weight that penalizes misclassifications.

In the general case of nonlinear mapping $\Phi$, SVM generates a nonlinear boundary $h(x) = 0$ in the input space. Given any two points $z_1, z_2 \in \mathbb{R}^d$ such that $h(z_1)\,h(z_2) < 0$, a surface point $s = \xi z_1 + (1-\xi) z_2, \xi \in [0,1]$, can be found by solving the following equation with respect to $\xi$:

$$h(s) = h\left(\xi z_1 + (1-\xi) z_2\right) = 0\,. \tag{18}$$

The unit normal vector $N(s)$ at the boundary point $s$ is then given by

$$N(s) = \frac{\nabla h(s)}{\|\nabla h(s)\|}\,, \tag{19}$$

where

$$\nabla h(s) = \frac{\partial h(s)}{\partial s} = \sum_{i=1}^{n} \alpha_i y_i \frac{\partial K(s, x_i)}{\partial s}\,. \tag{20}$$

Computations of $\partial K(s, x_i)/\partial s$ with different kernel functions are tabulated in Table 2. In the empirical study, we mainly employ polynomial kernels of various degrees $p$, where we have

$$\frac{\partial h(s)}{\partial s} = \sum_{i=1}^{n} \alpha_i y_i (s \cdot x_i + 1)^{(p-1)} \cdot x_i = \sum_{i=1}^{n} \frac{\alpha_i y_i K(s, x_i)}{s \cdot x_i + 1} \cdot x_i\,. \tag{21}$$

We summarize the SVM-DBA algorithm in Table 3. Several issues in the algorithm need some explanation. First, we prune those training samples far away from the decision boundary in locating the boundary points. This helps to reduce computational cost and suppress the negative influence of outliers. Second, we adopt the one-vs-all approach for solving $Q$-class problems with SVMs. Totally $Q$ SVMs need to be trained, each of which separates a single class from all remaining classes. Third, the complexity of SVM-DBA can be controlled by several parameters including $l$, the number of boundary points to be sampled, and $\delta$, the accuracy of the root to equation (18). Our experience seems to suggest that SVM-DBA is not very sensitive to the choice of these parameters. Finally, we have used $p$-degree polynomial kernels in our experiments.

In theory we can directly estimate the problem's MIDD via the rank of $M$. However, inevitable errors in our estimation prevent us from doing so. A more practical reason is that, the contributions of boundary normals to discrimination are non-even so that the subspace dimension may be significantly reduced while achieving *almost* the same accuracy. Therefore other model selection technique is required in order to decide an appropriate subspace. This problem will not be discussed in this paper, but we want to point out that nested subspaces generated by SVM-DBA can easily facilitate such explorations.

# 4 Experiments

We evaluate the proposed linear dimension reduction algorithm by one simulated and four real-world datasets drawn from the UCI Machine Learning Repository [7]. Their basic information is summarized in Table 4. On each dataset, we compare the goodness of the subspaces induced by SVM-DBA to those induced by PCA and LDA. Our comparative study differs from prior works in two aspects:

1. All the datasets we selected have been extensively benchmarked (see Appendix B for further details). Benchmark error rates have been listed in Table 4, that is, the Bayes error for WAVE [11], the lowest error rates of more than 20 classifiers for PIMA, VEHICLE and LETTER reported in the StatLog Project [59], and the median result of 25 classifier combination schemes for MFEAT [44]. To facilitate the comparison, we have kept our setups as close as possible to the benchmark experiments.
2. We employed $p$-polynomial SVM classifiers to evaluate the induced subspaces and tried our best to optimize the choice of parameters. Since SVM is presently one of the best-known classification techniques, it perhaps provides a better index for the discrimination power of the subspace. In our experiments, the degree of polynomial kernel $p$ and the penalty weight $C$ are chosen through cross validation. This tuning is by no means exhaustive, but is accurate enough not to affect any conclusions we have drawn.

Table 2: Commonly used kernel functions and their derivatives, where $u$ and $v$ are $d$ dimensional vectors.

| Kernel Type | $K(u, v)$ | $\partial K(u, v)/\partial u$ |
|---|---|---|
| Linear | $u \cdot v$ | $v$ |
| Polynomial | $(u \cdot v + 1)^p$ | $\frac{K(u, v)}{u \cdot v + 1} v$ |
| Gaussian Radial Basis | $\exp\left(-\frac{\|u - v\|^2}{2\sigma^2}\right)$ | $-\frac{K(u, v)}{\sigma^2}(u - v)$ |
| Sigmoid | $\frac{1}{1 + \exp[-\eta(u \cdot v) + \theta]}$ | $-\eta K(u, v)(1 - K(u, v))\, v$ |

Table 3: The proposed SVM-DBA algorithm.

**Input:** $n$ sample pairs $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{k\}_{k=1}^{Q}$.

**Output:** $d$ nested linear subspaces $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}_d$ such that $\dim(\mathcal{S}_m) = m$.

**SVM-DBA Algorithm:**

**S1** Repeat **S2** to **S5** for $k = 1$ to $Q$.

**S2** Divide the $n$ samples into two subsets $T^+ = \{x_i | y_i = k\}$ and $T^- = \{x_i | y_i \neq k\}$. Learn a SVM decision function $h(x)$ using $T^+$ and $T^-$.

**S3** Sort the $n$ samples in ascending order by the absolute function output values $|h(x_i)|$. Denote the subset consisting of the first $r$ samples as $T'$.

**S4** Select $l$ pairs of points $\{(z_1^j, z_2^j)\}_{j=1}^{l}$ from $T'$ randomly such that $h(z_1^j)\, h(z_2^j) < 0$. For each pair solve equation (18) to an accuracy of $\delta$, and thus get $l$ estimated boundary points $\{\hat{s}_j\}_{j=1}^{l}$.

**S5** Compute the unit surface norm $\hat{N}(\hat{s}_j)$ at $\hat{s}_j$ according to equation (19), and estimate the decision boundary scatter matrix as $\hat{M}_k = \sum_{j=1}^{l} \hat{N}(\hat{s}_j)\hat{N}(\hat{s}_j)^T$.

**S6** Compute the average scatter matrix $\hat{M} = \sum_{k=1}^{Q} \hat{M}_k / Q$, and its eigen decomposition $\hat{M} = \Psi^T \Lambda \Psi$, where $\Psi$ is an orthonormal matrix and $\Lambda$ is a diagonal matrix with decreasing eigenvalues.

**S7** Let $\mathcal{S}_m = \mathcal{S}(\Psi_m)$, where $\Psi_m$ is a $d \times m$ matrix that consists of the first $m$ columns of $\Psi$.

Table 4: A summary of dataset information.
CV – Cross Validation, $\varepsilon_0\%$ – Benchmark Error Rates.

| Dataset | #Classes | #Attributes | #Samples | Test Mode (#Train,#Test) | $\varepsilon_0$ |
|---|---|---|---|---|---|
| WAVE-21 | 3 | 21 | - | (10-300,5000)×10 | 14 |
| WAVE-40 | 3 | 21+19 | - | (100-1500,5000)×50 | 14 |
| PIMA | 2 | 8 | 768 | 12-fold CV | 22.3 |
| VHICLE | 4 | 18 | 846 | 9-fold CV | 15.0 |
| LETTER | 26 | 16 | 20,000 | (15000,5000) | 6.4 |
| MFEAT | 10 | 649 | 2,000 | (500,1500) | 2.3 |

In fact, all the SVM error curves we got by the proposed method have been, at some subspace dimensions, lower than or close to the benchmark values. We believe that evaluating the "extreme" performance of the induced subspace is significant provided the reduction sufficiency is concerned, and casually chosen evaluators may be insensitive to the loss of subtle discriminative information.

We first illustrate the robustness of SVM-DBA to small sample size on the WAVE dataset, which are modified versions of the well-known simulated example from the CART book [11]. There are two versions of the dataset, denoted as WAVE-21 and WAVE-40 respectively. WAVE-21 is a three class problem. Each class is generated from a combination of two of three "base" waves sampled at 21 points, and each instance is generated by adding Gaussian noise ($\sigma^2$=1) to the signal. WAVE-40 is constructed from WAVE-21 by appending at the end 19 pure noise attributes ($\sigma^2$=9). Thus its first 21 attributes are highly correlated and strongly relevant for discriminating the classes, while the extra 19 are completely irrelevant. It can be proven that the Bayes errors of both versions are about 14%, and their MIDDs equal two. Hence we can generate training sets of different sizes, and directly evaluate the quality of the MIDS estimates (i.e. the induced 2D subspaces). The average SVM error in the estimated MIDS over a number of simulations (10 for WAVE-21 and 50 for WAVE-40) is plotted in Figure 1 as a function of the size of training data. It can be observed that the superiority of SVM-DBA becomes more obvious as the sample size decreases. For example, on WAVE-40 the gap between SVM-DBA and LDA estimates increases from 1.6% to 11.3% (corresponding to an increase in relative improvements over LDA performance from 11% to 31%) when sample size is reduced from 1500 to 100. To give an intuitive impression, we show in Figure 2 the scatter plots in the 2D MIDSs estimated via different methods on WAVE-40.

It is interesting to note the diverse behaviors of PCA on the short and long versions of WAVE data. For WAVE-21 whose signal-to-noise ratio is large, it can be proved that the

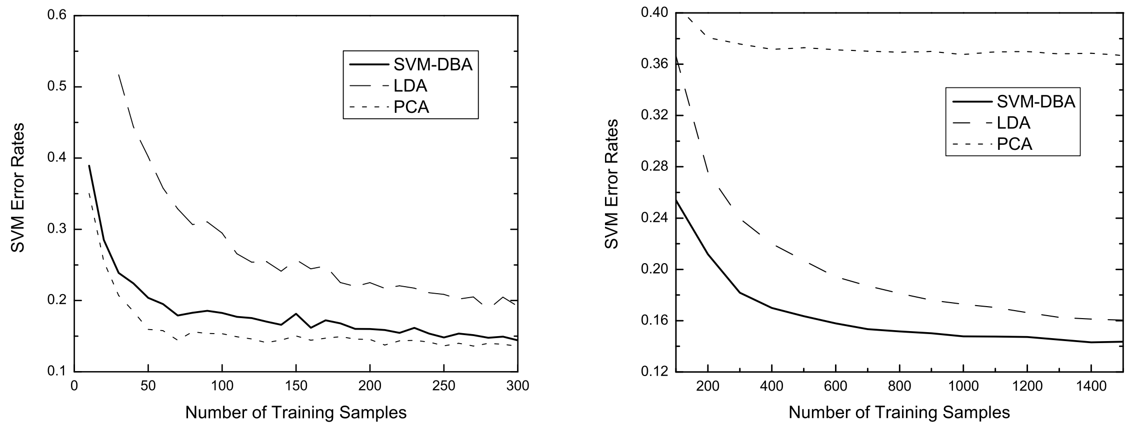
2D subspace with least mean square error (MSE) coincides with MIDS, i.e., the hyperplane spanned by the three "base" waves. In this case, PCA seems to show more effective usage of the available samples, thus marginally outperforming SVM-DBA. However, the optimality of MSE criterion disappears in WAVE-40, where the magnitudes of the 19 irrelevant noise attributes are comparable to that of the signal. As a result, the performance of unsupervised dimension reduction algorithms like PCA could be surprisingly poor in sense of discrimination, as has been demonstrated in Figure 1(b).

Figure 3 gives the comparative results over all output dimensions. We observe that SVM-DBA is consistently superior on all datasets over a dominant range of output dimensions. Failures at low dimensions are reasonable, since the optimality of DBA does not hold when output dimensions get less than MIDD of the data. This seems not to be a problem as far as sufficient reduction is concerned. They also suggest a complement between SVM-DBA and those iterative approaches that optimize complex projection indices in the projected subspace. The significant drops in SVM error on WAVE-40 and PIMA suggest that SVM-DBA introduces further regularizations into SVM when irrelevant and/or redundant features exist. However, similar effects cannot be observed with LDA and PCA.

One may argue that the "good" performance of SVM-DBA can be attributed to the use of the evaluator (i.e. the SVM classifier) itself, and it may work "badly" for other evaluators. To the best of our knowledge, almost all existing linear dimension reduction algorithms can be labeled as so-called *filters* [46, 8], and we believe further study on the coupling effect between reduction methods and types of classifiers deserves attention. On the other hand, we believe that there are some common regularities in most real-world datasets that distinguish them from pure random sets, as has been confirmed by the results of other authors [20, 37]. Hence a subspace that allows high performance of one classifier should also facilitate high performance of a different classifier. As a prelimenary attempt, we replace the SVM evaluator with a 1-NN classifier and repeat the comparison of three reduction algorithms, with results given in Figure 4. Although the best performances of 1-NN are much worse than SVM on all datasets except LETTER, the superiority of SVM-DBA is still held. On two datasets (VEHICLE and MFEAT) LDA becomes comparable to SVM-DBA. This can be explained by its "whitening" effect that favors nearest neighbor classification [28].

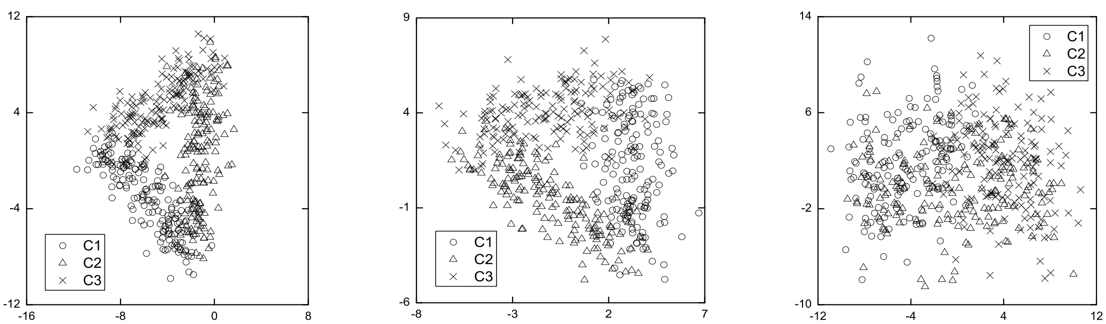
# 5 Related work

Guyon et al. [31] proposed a feature ranking scheme by linear SVMs. The basic idea is to use the magnitude of the weights of a linear discriminant classifier as an indicator of feature relevance. This approach can be readily generalized to nonlinear cases via SVM-DBA based feature screening (see [78] for further details), in which the feature relevance is evaluated by

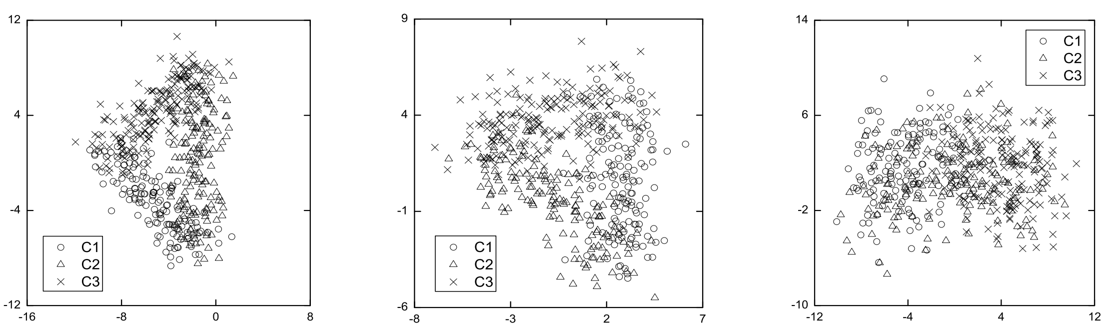



(a) WAVE-21 (b) WAVE-40

Figure 1: Quality of MIDS estimates as functions of the sample size on the WAVE dataset. Both WAVE-21 and WAVE-40 have the same MIDD(=2).




(a) SVM-DBA, Train (b) LDA, Train (c) PCA, Train




(d) SVM-DBA, Test (e) LDA, Test (f) PCA, Test

Figure 2: Scatter plots of training and test WAVE-40 samples in the 2D MIDSs estimated via different methods. The number of training samples $n = 500$.

$v^T M v$. Here $v$ denotes the feature direction, and $M$ denotes the SVM decision boundary scatter matrix.

SVM boundary has been used in locally adaptive metric techniques to improve $k$-NN performance [62, 23]. Measures of local feature relevance are computed from the surface normal near the query, from which a local full-rank transformation is derived. In [62] the surface normal is estimated by repeatedly learning linear SVMs from the neighbors of the query, while in [23] a nonlinear SVM boundary is precedently computed. Both of these two local approaches need to perform $k$-NN procedure multiple times in the original high-dimensional space. On the contrary, SVM-DBA tries to globally characterize the discriminative information embedded in the SVM decision boundary. It induces a single reduced-rank projection, and thus is computationally much more efficient.

Hristache et al. [40, 39] proposed an iterative improvement of ADE, where the smoothing windows are shrinked in the directions of current central space estimation and stretched in all orthogonal directions. Adaptive smoothing can relax some assumptions required by previous ADE algorithms, and get more accurate estimate of the derivatives. However, it does need *a prior* knowledge of the subspace dimension. SVM-DBA achieves the same goal while avoids this limitation, thanks to the good generalization ability of SVM.

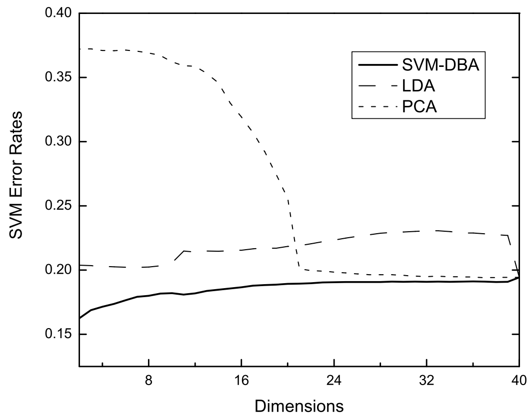
# 6 Conclusion and discussion

We have formulated the problem of sufficient dimension reduction for classification in parallel terms as for regression. A novel reduction algorithm is proposed which inherits the appealing properties of both SVM and DBA. The maximum margin boundary of SVM alleviates the sample-size and computational limitations in traditional DBA implementations. DBA in turn provides a natural way to reduce the complexity, and even improve the accuracy, of SVM itself. The proposed method generates nested subspaces that enable exploration of the tradeoffs between classification performance and subspace compactness, and facilitate further test for finding a proper working dimension. It only requires that SVM is suitable for the data.

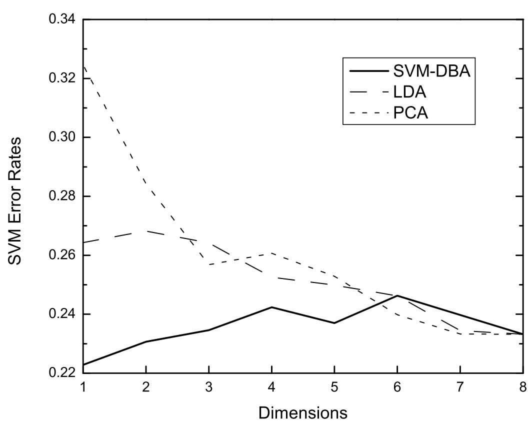
We plan to explore several extensions:

1. Combine SVM-DBA and iterative approaches that optimize complex projection indices in the projected subspace when the desired output dimension is much lower than the data's MIDD.

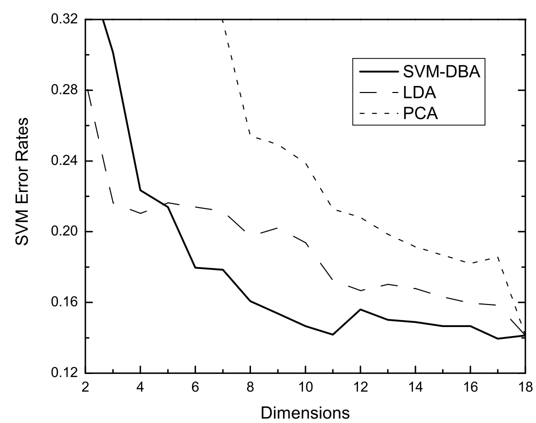
2. Incorporate priors on the linear mapping structures. For instance, suppose the original feature set is dominated by irrelevent noise, then it may be more effective to do feature selection, i.e., to constrain the search in those subspaces that are only spanned by the original feature directions. Feature selection is effective in dealing with irrelevant features, but is limited in its ability to model high redundancy. Hence, in more general cases,



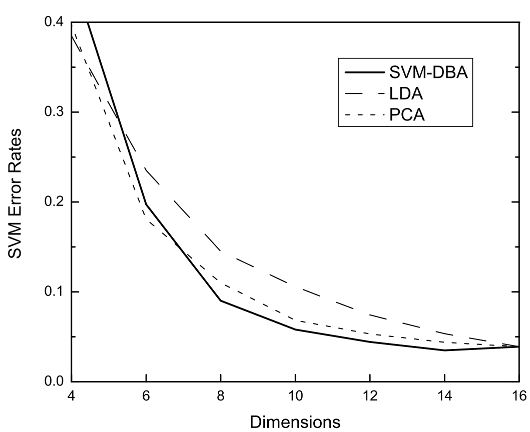

(a) WAVE-40, $n$ = 500



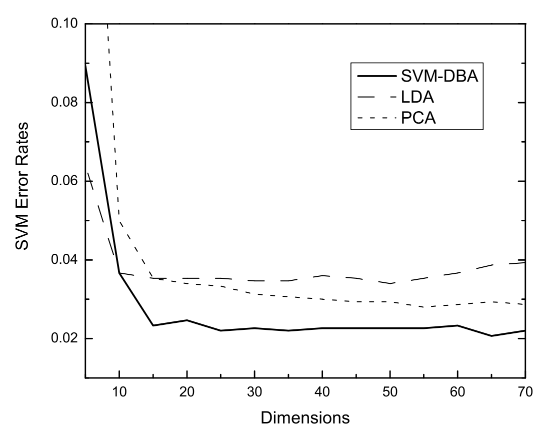

(b) PIMA




(c) VEHICLE

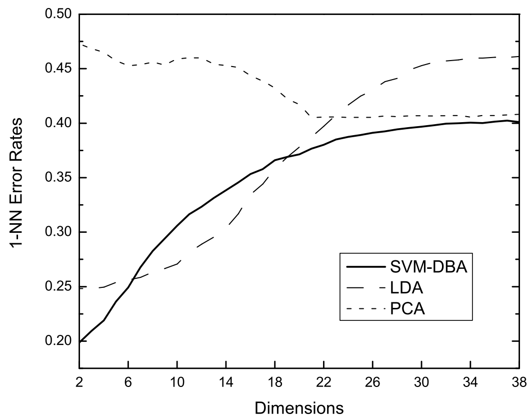



(d) LETTER

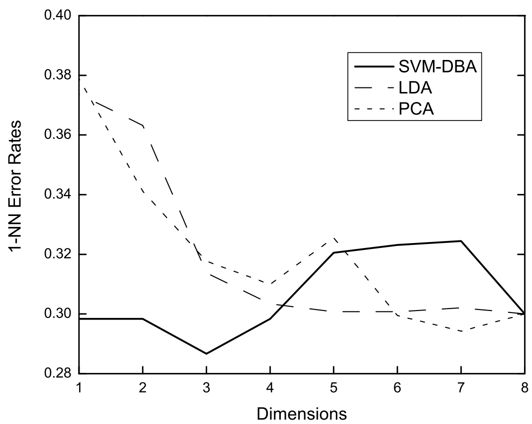



(e) MULTI-FEATURES

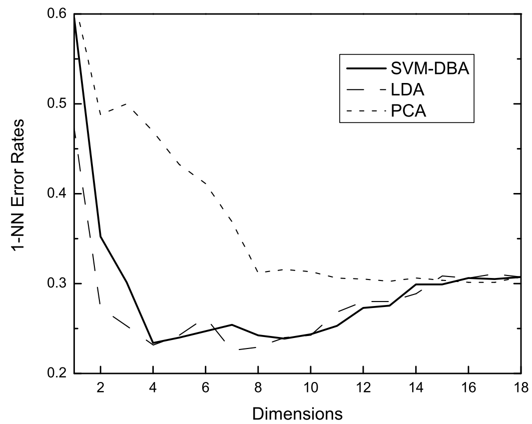
Figure 3: Comparison of subspaces over all output dimensions using SVM classifier on different datasets.



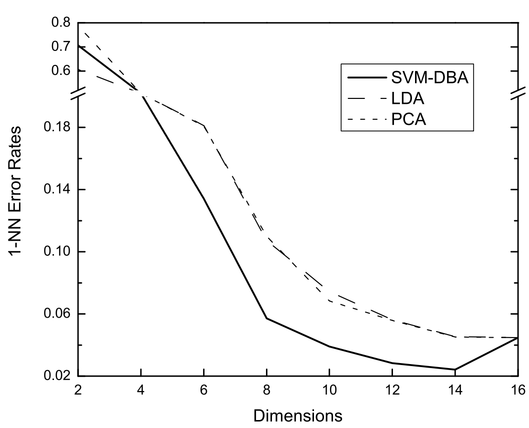

(a) WAVE-40, $n = 500$



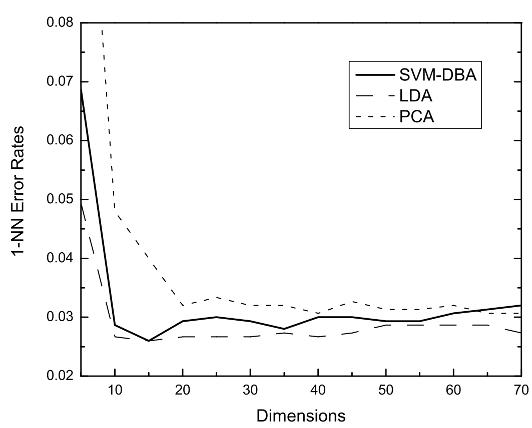

(b) PIMA




(c) VEHICLE




(d) LETTER




(e) MFEAT

Figure 4: Comparison of subspaces over all output dimensions using 1-NN classifier on different datasets.

hybrid strategies that concatenate selection and transformation should be desirable.

3. Introduce the subspace structure constraint into SVM's learning process by, e.g., either adaptively modifying the kernel or explicitly transforming the feature space [2, 77, 39], thus jointly optimize the boundary estimation and the discriminative subspace through an iterative process.
4. Investigate the performance of SVM-DBA for central mean subspace estimation in context of regression.

# Appendix

# A Linear Dimension Reduction: A Selected Review

In this section we will briefly review some classification-driven linear dimension reduction methods, emphasizing recent developments. The main issue of dimension reduction is the choice of a criterion function. Bayes error is an ideal criterion for the least loss of classification information. However, the Bayes error itself cannot be reliably estimated when the ratio of sample size to the number of features is small. In addition, it is known to be notoriously difficult for mathematical treatment, even in the case of multivariate normal distributions. Consequently various suboptimal criteria have been used in practice. These can be generally divided into two groups: 1) various functions of scatter matrices or likelihood, and 2) criteria that approximate or place upper/lower bounds on the Bayes error.

Linear Discriminant Analysis (LDA) [28] is a time-honored, standard tool for dimension reduction, which maximizes the so-called Fisher criterion $J(\Psi) = \mathrm{tr}[(\Psi S_w \Psi^T)^{-1}(\Psi S_b \Psi)]$, where $S_b$ is the between-class scatter matrix, $S_w$ is the pooled within-class scatter matrix. The eigenvectors corresponding to largest eigenvalues of $S_w^{-1} S_b$ constitute the columns of the solution. LDA assumes unimodal equal-covariance Gaussian class distributions, with reasonable robustness to non-normality and even to mild non-homoscedasticity. On the down side, it is not directly related to the classification rate. It is too flexible (overfit) in situations with large number of highly correlated features, and too rigid (underfit) when the decision boundaries are complex and nonlinear. As the rank of $S_b$ is $(Q-1)$, the dimension of the projected subspace is limited by the number of categories. Some variants of LDA can avoid part of these problems. For example, besides the commonly used ML estimator of $S_w$, various regularization techniques are available to obtain robust estimates in situations of small sample-size (e.g. RDA [26]) or high collinearity (e.g. PDA [32]). Okada and Tomita [61] lifted the $(Q-1)$ limitation by selecting the projection axes one at a time under an orthogonality constraint.

Campbell [14] first shown that reduced-rank LDA is equivalent to the restricted maximum likelihood solution to a Gaussian model, subject to the equal within-class covariance constraint and the rank $m$ constraint on the class centroids. In light of this connection, several generalizations of LDA have been proposed under the maximum likelihood framework.

Kumar [48] removed the equal within-class covariance assumption when studying the case for diagonal covariance modeling, and proposed Heteroscedastic Discriminant Analysis (Kumar's HDA), which is a maximum likelihood solution to a Gaussian model with common covariances in the rejected subspace. A different version of HDA is proposed in [68], which directly maximizes the between-class separation in the projected subspace while keeping the product of individual with-class separations constant. It can be interpreted as a constrained maximum likelihood projection with the constraint being given by the maximization of the projected between-class scatter volume. If diagonal covariance modeling is adopted in the projected subspace, the so-called Maximum Likelihood Linear Transform (MLLT) [30] may be applied after HDA to minimize the loss in likelihood between the full and diagonal covariance Gaussian models, i.e., to make the diagonal constraint more valid as evidenced from the data. This is in accordance with the common understanding that the features generated should be suited to the form of classifier being used [46]. MLLT turns out to be a particular case of Kumar's HDA when the dimensions of the original and the projected space are the same.

Hastie and Tibshirani [33] assumed that each observed class is in fact a mixture of unobserved normally distributed subclasses with equal within-subclass covariance. The resultant problem, called Mixture Discriminant Analysis (MDA), is formulated as maximizing the Gaussian mixture log-likelihood subject to the rank $m$ constraint on the subclass means, which can be solved by EM iterations. MDA enjoys a number of favorable properties over LDA. It can deal with nonlinear decision boundaries. The number of discriminant variates can be up to one less than the number of subclasses in the mixture representation. In addition, a series of functionalities are added. By solving the M-step via weighted "optimal scoring", i.e. multiple linear regression of a blurred response matrix followed by an eigen analysis, both PDA [32] and FDA [34] adapt naturally to MDA. Another feature called subclass shrinkage is proposed to bias the decomposition, and the positions of the means themselves, towards classification. MDA can be viewed as a smooth version of LVQ.

Nonparametric Discriminant Analysis (NDA), first proposed by Fukunaga and Mantock [27], generalizes LDA by replacing the parametric between-class scatter $S_b$ with a distribution free version through $k$-nearest neighbor local density estimation. Here is roughly how NDA works for a two-class problem. For each point $x^1$ in class $C_1$, its $k$ nearest neighbors $\{x_i^2\}_{i=1}^k$ in class $C_2$ can be found, and the shift vector between $x^1$ and the mean of $\{x_i^2\}_{i=1}^k$ is then computed. The nonparametric $S_b$ is defined as the scatter matrix of all shift vectors. To de-emphasize samples far from the boundary, an adaptive weight controlled by a parameter $\alpha$ can be applied to each shift vector. When $k$ equals to the number of training samples in each class, NDA degenerates to LDA. Practical performance of NDA is usually difficult to predict because: 1) $k$ and $\alpha$ are decided by rules of thumb, 2) $S_w$ is still with a parametric form, 3) the shift vector estimation is noise sensitive.

The computation of $S_b$ in LDA weights contributions of individual class pairs according to the Euclidian distance of the respective class means. This tends to overweight large class distances since the resulting transform preserves the distance of already well-separated classes while causing a large overlap of neighboring classes. Approximate Pairwise Accuracy Criterion (aPAC) [55] is a computationally inexpensive method to tackle this problem, in which the weighting is derived from an attempt to approximate the Bayes error for pairs of classes in the projected subspace. aPAC still assumes unimodal equal-covariance Gaussian class-conditional distributions and retains the computational simplicity of LDA. An important property is that it can confine the influence of outlier classes on the final transform, thus is more robust than LDA. However, aPAC is no guaranteed to always lead to improved classification rate because of the various approximations introduced in order to arrive at a solution that is computationally simple. Various other weighting heuristics are available to tweak LDA performance (e.g., the iterative weighting in [57]). Lotlikar and Kothari [56] introduced an adaptive approach which is quite similar to aPAC, except that an exact pairwise Bayes error function without any approximation is minimized by iterative adjustment to the projection matrix. As a result, it is computationally much more demanding. The adaptive approach can be extended to more general distributions using Parzen window density estimation as a superposition of Gaussians.

Two probabilistic class separability measures were adopted in [67]: 1) Average pairwise Kullback divergence. For classes with uniform priors, if the divergence in the projected space is the same as in the original space, then the Bayes error is also preserved; 2) Bhattacharyya bound, an upper bound on the Bayes error. The practical implementations are accompanied by a parametric estimation (unimodal Gaussian) of the densities at hand.

Patrick-Fisher distance [22] can be defined nonparametrically using Parzen estimators with Gaussian kernels of the class-conditional densities of the projections. It induces an upper bound on Bayes error that is larger than those of other probabilistic class separability measures. Nevertheless it is more practical for its analytical simplification. Aladjem [1] studied the optimization of the highly nonlinear PF distance function in case of two-class problems. Stimulated by an idea of Friedman called "structure removal" [25], he proposed a recursive optimization procedure for searching the directions corresponding to several large local maxima of the PF distance. The main idea is to transform the data along a found direction into data with deflated maxima of the PF distance, and then iterate to obtain the next direction.

Shannon's mutual information determines a lower bound to the Bayes error according to Fano's inequality [24]. Hence the lower bound on error probability is minimized when the mutual information between feature vector and class label is maximized. However, the MMI criterion has not been in wide use due to computational difficulties. The probability density functions are required, and MI involves integrating functions of those, which in turn involves evaluating those on a dense set. Bollacker and Gosh [9] replaced the joint MI with the sum of numerically approximated pairwise MIs, which are independently computed between the class label and a set of orthogonal projections. A greedy search is employed to find the successive 1D projections. At each step, a single direction that maximizes the corresponding pairwise MI is found through polytope algorithm and then removed from the input space. The procedure is repeated in the residual subspace. "This method suffers from the problem", as mentioned by the authors themselves, "of overlapping mutual information contributed by each feature, which becomes worse as more features are extracted". Torkkola and Campbell [75, 73] combined Renyi's quadratic entropy and Parzen window density estimator into a differentiable and relatively simple optimization criterion for (non)linear transformation design. The criterion only depends on weighted pairwise interactions of all training samples, thus has computational complexity of $O(n^2)$. Later two techniques were incorporated to speed up the process, i.e., semi-parametric GMM [72] and online stochastic optimization [74]. Experiments show that this technique is especially useful for low dimensional mapping. On the down side: 1) Parzen estimator or GMMs limit the useful output dimension due to the density estimation difficulty in the high dimensional space. 2) Further tweaks such as better initialization, kernel smoothing or shrinking are needed to help escape from local maxima.

Several papers use as optimization criterion the $k$-nearest neighbor ($k$-NN) estimate of the Bayes error in the transformed space. Buturovic [13] proposed two criteria that are both average of leave-one-out and resubstitution $k$-NN estimates over an appropriate range of $k$, and adopted some improved estimation techniques such as Mahalanobis distance and optimal threshold. The simplex algorithm is used to minimize the criteria, which successively contract and expand the multidimensional simplex to bound the minimum value of the function. The initial simplex can be generated pseudo-randomly, or taken as a result of another dimension-reducing algorithm. As can be imagined, the algorithm is computationally expensive due to the high complexity of the error estimation procedure. Raymer et al. [64] used $k$-NN error as the fitness function in a genetic algorithm that simultaneously optimizes a real-valued scaling vector for feature transformation, a binary mask vector for feature selection, and the classifier parameter $k$. Note that feature extraction here is limited to the form of scaling, i.e., each output feature is based only on a single input feature. A more general form of feature extraction may be achieved by directly representing the entire transformation matrix on the chromosome, but it remains to be investigated whether the ability of GA to perform arbitrary transformations would lead to better classification accuracy or would be offset by the increase in the size of the search space to be explored.

Discriminative feature extraction (DFE) [4, 5] is one of the very few dimension reduction methods that explicitly examines the interactions between the front-end feature extractor and the back-end classifier. The formulation of DFE mainly relies on the minimum classification error/generalized probabilistic descent (MCE/GPD) framework. It has been shown in speech recognition tasks that improved recognition results can be obtained by using an integrated optimization of both the feature extractor and the classifier.

Stochastic methods are considered to possess some favorable properties over deterministic ones. It is not surprising that the combination of multiple random subset selections contains sufficient discriminative information [47, 36]. However, it seems less obvious that how much separability can be preserved in a single random projection. In fact there are two relevant theoretical results [21]: 1) Data from a mixture of $k$ Gaussians can be projected into $O(\log k)$ dimensions while still retaining the approximate level of separation between the clusters. The projected dimension is independent of the number of data points and the original dimension; 2) Even if the original clusters are highly eccentric, random projection will make them more spherical. These two benefits have made random projection the key ingredient in the first polynomial-time, provably correct (in a PAC-like sense) algorithm for learning mixtures of Gaussians. Although it must be possible to do better by actually taking the data into account, there have been no deterministic procedure which has such performance guarantees. Experiments (quite sparse) indicate that sometimes random projection can get results comparable to PCA with much less computational cost.

We summarize the algorithms introduced so far in Table 5. A quick scan of the table reveals that several problems are underestimated in previous research: 1) sufficient dimension reduction, 2) interactions between reduction methods and classifiers being used, 3) *a priori* knowledge of the mapping structure, and 4) generalization performance in situations of small sample set and very high dimension. The scope of our review is limited in several aspects:

1. The whole body of literature on dimension reduction for visualization, exploration and regression is omitted.

2. We emphasize supervised algorithms that utilize the category information associated with each sample, thus have ignored a bunch of well-known unsupervised methods such as PCA, Projection Pursuit [41] and ICA [42]. Although supervised methods generally have better performance than unsupervised methods in context of discrimination tasks, this is not always the case. For example, it was demonstrated [58] that, when training set was small, PCA could outperform LDA in appearance-based object recognition. Another example is our comparative results on the WAVE-21 dataset (see Figure 1(a)).

3. We restrict to linear mapping, which is less powerful than nonlinear methods. In fact, it has been proved in theory [12] that, for a $Q$-class classification problem with continuous class-conditional density functions, its worst case minimal dimension of continuous discriminative sufficient statistics is $\min(d, Q-1)$, where $d$ is the dimension of the orig-

inal feature space. However, this result offers little guidance in practical situations with unknown distribution and limited learning samples.

4. All the selected schemes exclusively generate a single "good" feature subspace. However, for some complex problems, multiple local feature sets (e.g. sample-dependent or class-specific) may be more appropriate than a single feature set. The idea of multiple subspace projections has been examined in several papers (e.g. [15, 29]) for HMM-based speech recognition, where the key issue is to handle multiple subspaces while still maintaining meaningful distance measure comparisons between classes.

# B Dataset Information

This section documents detailed information with benchmark results on the datasets employed in the study. Most materials are excerpted from [11, 59, 44] with little modification.

## B.1 Waveform Data

WAVE is a popular simulated example taken from [11] and used elsewhere. It is a three class problem. Each class is generated from a combination of two of three "base" waves, and each instance is generated by adding Gaussian noise in each attribute. The priors for three classes are chosen to be equal. There are two versions of the dataset, WAVE-21 and WAVE-40, with 21 and 40 variables respectively. WAVE-21 patterns are defined by

$$\begin{aligned}
x_i &= u\, h_1(i) + (1-u)\, h_2(i) + \epsilon_i && \text{Class 1} \\
x_i &= u\, h_1(i) + (1-u)\, h_3(i) + \epsilon_i && \text{Class 2} \\
x_i &= u\, h_2(i) + (1-u)\, h_3(i) + \epsilon_i && \text{Class 3}
\end{aligned}$$

where $1 \le i \le 21$, $u \sim U(0,1)$, $\epsilon_i \sim N(0,1)$, and $h_j$ are shifted triangular waveforms: $h_1(i) = \max(6 - |i-1|, 0)$, $h_2(i) = h_1(i-4)$, $h_3(i) = h_1(i+4)$. The standard WAVE-40 is constructed from WAVE-21 by appending 19 zero points at the end of $h_j$, which is equivalent to extending the signal with 19 pure noise attributes (standard normal).

For the version of short signal, all 21 attributes are highly correlated and strongly relevant for discriminating the classes. The problem is difficult, since the optimal Bayes error is only about 14% [11]. As we might have guessed, the classes appear to lie on the edges of a triangle in a two-dimensional hyperplane. This is because the $h_j$ are represented by 3 points in 21-dimensional space, thereby forming vertices of a triangle, and each class is represented as a convex combination of a pair of vertices, and hence lies on an edge. The signal generation process enables linear dimension reduction to be particularly effective in removing redundancy

Table 5: Selected classification-driven supervised linear dimension reduction methods. The methods are grouped into three sections: that can be unified under the maximum likelihood framework (top); that use class separability measures either approximating or placing upper/lower bounds on the Bayes error (middle); that remain uncovered (bottom).

| Method | Criterion | Optimization | Comments |
|---|---|---|---|
| Linear Discriminant Analysis (LDA) [28] | SCATTER | EIGEN | fast, good for equal-covariance Gaussian, limited to $(Q-1)$ outputs, overemphasize large class distances |
| Penalized Discriminant Analysis (PDA) [32] | SCATTER | EIGEN | impose spatial smoothness constraint on the coefficients, better than LDA for high collinearity |
| Heteroscedastic Discriminant Analysis (Kumar's HDA) [48] | LL | GD | diagonal but not necessarily equal covariance modeling |
| Heteroscedastic Discriminant Analysis (HDA) [68] | SCATTER | GD | Gaussian modeling without equal-covariance constraint |
| Maximum Likelihood Linear Transform (MLLT) [30] | LL | GD | special case of full-rank Kumar's HDA, good for diagonal covariance modeling |
| Mixture Discriminant Analysis (MDA) [33] | LL | EM | non-Gaussian, recommended to be combined with penalization and subclass shrinkage |
| Approximate Pairwise Accuracy Criterion (aPAC) [55] | "Bayes" error | EIGEN | fast, good for equal-covariance Gaussian, more robust to outlier classes than LDA |
| (Lotlikar and Kothari, 2000) | "Bayes" error | GD | (non-)parametric, without approximations in aPAC but computationally more expensive |
| (Saon and Padmanabhan, 2000) | KULLBACK | GD | conditionally sufficient, implemented for Gaussian |
| | BHATTA | GD | upper bound, implemented for Gaussian |
| (Aladjem, 1998) | PF | GD + structure removal | nonparametric, two-class, looser upper bound, search for multiple large local maxima |
| (Bollacker and Gosh, 1996) | MI | polytope | pairwise approximation to joint MI, sequential greedy search, good for low dimensional mapping |
| (Torkkola and Campbell, 2000) | MI | GD | nonparametric or semi-parametric, approximate Shannon's MI, esp. good for low dimensional mapping |
| (Buturovic, 1994) | $k$-NN error | simplex | non-Gaussian, computationally expensive, possibly poor generalization |
| (Raymer et al., 2000) | $k$-NN error | GA | limited in the form of mapping, combine scaling and selection |
| Discriminative Feature Extraction (DFE) [4, 5] | MCE | GPD | wrapper-style, approximate empirical error, classifier-dependent implementation |
| Nonparametric Discriminant Analysis (NDA) [27] | SCATTER | EIGEN | non-Gaussian, sample-size and noise sensitive, too many adjustable parameters |
| Decision Boundary Analysis (DBA) [51] | SCATTER | EIGEN | non-Gaussian, sample-size and noise sensitive, computationally expensive |
| Random Projection [21] | – | – | very fast, good for high dimensional linear mapping, sometimes comparable to PCA |

Key to abbreviations:

SCATTER – scatter-matrix based
LL – log-likelihood
MI – mutual information
PF – Patrick-Fisher distance
KULLBACK – Kullback divergence
BHATTA – Bhattacharyya distance
MCE – minimum classification error
EIGEN – eigen decomposition
GD – gradient descent
EM – expectation maximization
GA – genetic algorithm
GPD – generalized probabilistic descent

Table 6: The cross error rates (%) of some classification methods on WAVE-21 and the standard WAVE-40 datasets [11, 33].

| Algorithm | WAVE-21 | WAVE-40 |
|---|---|---|
| Optimal | 14 | 14 |
| CART[1] | 28 | 28 |
| 1-NN[1] | 22 | 62 |
| LDA[2] | 19.1(.6) | – |
| QDA[2] | 20.5(.6) | – |
| CART[2] | 28.9(.4) | – |
| MDA[2] | 15.5(.5) | – |

[1] (training, test) = (300, 5000)
[2] (training, test) = (300, 500) × 10
format = MEAN(STD)

Figure 5: Vehicle silhouettes prior to high level feature extraction. These are clockwise from top left: Double decker bus, Opel Manta 400, Saab 9000 and Chevrolet van.

in the origial attributes. In fact, it can be proved that the MIDD of WAVE-21 is two. Intuitively we can imagine that this 2D subspace should be spanned by the three "base" waves, *i.e.* the hyperplane spanned by the three points $h_j\,(j = 1, 2, 3)$. For the version of long signal, these analyses are still valid. However, the 19 extra completely irrelevant attributes make it more difficult either to classify signals or to identify MIDD, especially when the size of training samples is small. And to make it even worse, we have deliberately increased from 1 to 9 the variance of the 19 pure noise attributes in the standard WAVE-40.

Some previous results on WAVE-21 and standard WAVE-40 are given in Table 6.

## B.2 Pima Indians Diabetes Database

This dataset was originally donated by Vincent Sigillito, Applied Physics Laboratory, Johns Hopkins University and was constructed by constrained selection from a larger database held by the National Institute of Diabetes and Digestive and Kidney Diseases. All represented patients are females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona. The problem is to predict whether a patient would test positive for diabetes according to World Health Organization criteria (i.e. if the patients 2 hour post-load plasma glucose is at least 200 mg/dl) given a number of physiological measurements and medical test results. The eight attributes are: number of times pregnant, plasma glucose concentration in an oral glucose tolerance test, diastolic blood pressure (mmHg), triceps skin fold thickness, 2-hour serum insulin (muU/ml), body mass index (kg/m$^2$), diabetes pedigree function, and age (years). There are 500 examples of class 1 (tested positive) and 268 of class 2 (tested negative). Twelve-fold cross validation was used to estimate prediction accuracy.

The dataset is rather difficult to classify. The so-called "class" value is really a binarized form of another attribute which is itself highly indicative of certain types of diabetes but does not have a one-to-one correspondence with the medical condition of being diabetic. No algorithm performs exceptionally well, although ALLOC80 and k-NN seem to be the poorest. Automatic smoothing parameter selection in ALLOC80 can make poor choices for datasets with discrete valued attributes and k-NN can have problems scaling such datasets. Overall though, it seems reasonable to conclude that the attributes do not predict the class well. Cal5 uses only 8 nodes in its decision tree, whereas NewID, which performs less well, has 119 nodes. $AC^2$ and C4.5 have 116 and 32 nodes respectively and CN2 generates 52 rules, although there is not very much difference in the error rates here.

PIMA has been studied by Smith *et al.* (1988) using the ADAP algorithm. With 576 examples as a training set, ADAP achieved an error rate of 0.24 on the remaining 192 instances.

## B.3 Vehicle Silhouettes

A problem in object recognition is to find a method of distinguishing 3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects. This data was originally gathered at the Turing Institute in 1986-87 by J. P. Siebert. Four "Corgi" model vehicles were used for the experiment (see Figure 5). This particular combination of vehicles was chosen with the expectation that the bus, van and either one of the cars would be readily distinguishable, but it would be more difficult to distinguish between the cars. The vehicles were rotated and a number of image silhouettes were obtained from a variety of orientations and angles. All images were captured with a spatial resolution of $128 \times 128$ pixels quantized to 64 grey levels. These images were cleaned up, binarized and subsequently processed to produce 18 variables intended to characterize shape (e.g. circularity, radius ratio, compactness, scaled variance along major and minor axes). A total of 946 examples were obtained but 100 were retained in case of dispute, so the trials reported in Table 7(b) used only 846 examples and the algorithms were run using 9-fold cross-validation.

One would expect this dataset to be nonlinear since the attributes depend on the angle at which the vehicle is viewed. Therefore they are likely to have a sinusoidal dependence, although this dependence was masked by issuing the dataset in permuted order. Quadisc does very well, and this is due to the highly nonlinear behavior of this data. One would have expected the Backprop algorithm to perform well on this dataset since, it is claimed, Backprop can successfully model the nonlinear aspects of a dataset. However, Backprop is not straightforward to run. Unlike discriminant analysis, which requires no choice of free parameters, Backprop requires essentially two free parameters: the number of hidden nodes and the training time. Neither of these is straightforward to decide. This figure for Backprop was obtained using 5 hidden nodes and a training time of four hours for the training time in each of the nine cycles of cross-validation. However, one can say that the sheer effort and time taken to optimize the performance for Backprop is a major disadvantage compared to Quadisc which can achieve a much better result with a lot less effort. DIPOL92 does nearly as well as Quadisc. As compared with Backprop it performs better and is quicker to run. It determines the number of nodes (hyperplanes, neurons) and the initial weights by a reasonable procedure at the beginning and doesn't use an additional layer of hidden units but instead a symbolic level. The poor performance of CASTLE is explained by the fact that the attributes are highly correlated. In consequence the relationship between class and attributes is not built strongly into the polytree. The same explanation accounts for the poor performance of Naive Bayes. k-NN, which performed so well on the raw digits dataset, does not do so well here. This is probably because in the case of the digits the attributes were all commensurate and carried equal weight. In the vehicle dataset the attributes all have different meanings and it is not clear how to build an appropriate distance measure. The attributes for the vehicle dataset, unlike the other image analysis, were generated using image analysis tools and were not simply based on brightness levels. This suggests that the attributes are less likely to be equally scaled and equally important. This is confirmed by the lower performances of k-NN, LVQ and Radial Basis functions, which treat all attributes equally and have a built in mechanism for normalizing, which is often not optimal. ALLOC80 did not perform well here, and so an alternative kernel method was used which allowed for correlations between the attributes, and this appeared to be more robust than the other three algorithms although it still fails to learn the difference between the cars. The original Siebert (1987) paper showed machine learning performing better than k-NN, but there is not much support for this in our results. The tree sizes for $AC^2$ and Cal5 were 116 and 156 nodes, respectively. The high value of fract2 = 0.8189 might indicate that linear discrimination could be based on just two discriminants. This may relate to the fact that the two cars are not easily distinguishable, so might be treated as one (reducing dimensionality of the mean vectors to 3D). However, although the fraction of discriminating power for the third discriminant is low ($1 - 0.8189$), it is still statistically significant, so cannot be discarded without a small loss of discrimination.

## B.4 Letter Recognition Database

The dataset was constructed by David J. Slate, Odesta Corporation. The objective here is to classify each of a large number of black and white rectangular pixel displays as one of the 26 capital letters of the English alphabet. The character images produced were based on 20 different fonts and each letter within these fonts was randomly distorted to produce a file of 20,000 unique images. For each image, 16 numerical attributes were calculated using edge counts and measures of statistical moments which were scaled and discretized into a range of integer values from 0 to 15. One-shot train and test was used for the classification. Perfect classification performance is unlikely to be possible with this dataset. One of the fonts used, Gothic Roman, appears very different from the others.

Table 7: StatLog results for PIMA, VEHICLE and LETTER datasets.

(a) PIMA (2 classes, 8 attributes, 768 observations, 12-fold cross-validation)

| Algorithm | Max Stor | Time (sec.) Train | Time (sec.) Test | Error Rate Train | Error Rate Test | Rk |
|---|---|---|---|---|---|---|
| LogDisc | 311 | 30.8 | 6.6 | 0.219 | 0.223 | 1 |
| DIPOL92 | 52 | 35.8 | 0.8 | 0.220 | 0.224 | 2 |
| Discrim | 338 | 27.4 | 6.5 | 0.220 | 0.225 | 3 |
| SMART | 780 | 3762.0 | – | 0.177 | 0.232 | 4 |
| RBF | 179 | 4.8 | 0.1 | 0.218 | 0.243 | 5 |
| ITrule | 60 | 31.2 | 1.5 | 0.223 | 0.245 | 6 |
| BackProp | 147 | 7171.0 | 0.1 | 0.198 | 0.248 | 7 |
| Cal5 | 137 | 236.7 | 0.1 | 0.232 | 0.250 | 8 |
| CART | 144 | 29.6 | 0.8 | 0.227 | 0.255 | 9 |
| CASTLE | 82 | 35.3 | 4.7 | 0.260 | 0.258 | 10 |
| QuaDisc | 327 | 24.4 | 6.6 | 0.237 | 0.262 | 11 |
| NaiveBay | 431 | 25.0 | 7.2 | 0.239 | 0.262 | 11 |
| C4.5 | 61 | 11.5 | 0.9 | 0.131 | 0.270 | 13 |
| IndCART | 596 | 215.6 | 209.4 | 0.079 | 0.271 | 14 |
| BayTree | 68 | 10.4 | 0.3 | 0.008 | 0.271 | 14 |
| LVQ | 69 | 139.5 | 1.2 | 0.101 | 0.272 | 16 |
| Kohonen | 62 | 1966.4 | 2.5 | 0.134 | 0.273 | 17 |
| $AC^2$ | 373 | 4377.0 | 241.0 | 0.000 | 0.276 | 18 |
| NewID | 87 | 9.6 | 10.2 | 0.000 | 0.289 | 19 |
| CN2 | 190 | 38.4 | 2.8 | 0.010 | 0.289 | 19 |
| ALLOC80 | 152 | 1374.1 | – | 0.288 | 0.301 | 21 |
| k-NN | 226 | 1.0 | 2.0 | 0.000 | 0.324 | 22 |
| Default | – | – | – | 0.350 | 0.350 | 23 |

(b) VEHICLE (4 classes, 18 attributes, 846 observations, 9-fold cross-validation)

| Algorithm | Max Stor | Time (sec.) Train | Time (sec.) Test | Error Rate Train | Error Rate Test | Rk |
|---|---|---|---|---|---|---|
| QuaDisc | 593 | 250.9 | 28.6 | 0.085 | 0.150 | 1 |
| DIPOL92 | 64 | 150.6 | 8.2 | 0.079 | 0.151 | 2 |
| ALLOC80 | 227 | 30.0 | 10.0 | 0.000 | 0.173 | 3 |
| LogDisc | 685 | 757.9 | 8.3 | 0.167 | 0.192 | 4 |
| Backprop | 186 | 14411.2 | 3.7 | 0.168 | 0.207 | 5 |
| Discrim | 231 | 16.3 | 3.0 | 0.202 | 0.216 | 6 |
| SMART | 105 | 2502.5 | 0.7 | 0.062 | 0.217 | 7 |
| CART | 158 | 24.4 | 0.8 | 0.284 | 0.235 | 8 |
| C4.5 | – | 174.0 | 2.0 | 0.065 | 0.266 | 9 |
| BayTree | 71 | 27.1 | 0.5 | 0.079 | 0.271 | 10 |
| k-NN | 104 | 163.8 | 22.7 | 0.000 | 0.275 | 11 |
| Cal5 | 171 | 23.3 | 0.5 | 0.068 | 0.279 | 12 |
| Cascade | 238 | 289.0 | 1.0 | 0.263 | 0.280 | 13 |
| LVQ | 77 | 229.1 | 2.8 | 0.171 | 0.287 | 14 |
| $AC^2$ | 776 | 3135.0 | 121.0 | – | 0.296 | 15 |
| IndCART | 296 | 113.3 | 0.4 | 0.047 | 0.298 | 16 |
| NewID | – | 18.0 | 1.0 | 0.030 | 0.298 | 16 |
| RBF | 716 | 1735.9 | 11.8 | 0.098 | 0.307 | 18 |
| CN2 | – | 100.0 | 1.0 | 0.018 | 0.314 | 19 |
| ITrule | 307 | 985.3 | – | – | 0.324 | 20 |
| Kohonen | 1441 | 5962.0 | 50.4 | 0.115 | 0.340 | 21 |
| CASTLE | 80 | 13.1 | 1.8 | 0.545 | 0.505 | 22 |
| NaiveBay | 56 | 5.4 | 0.6 | 0.519 | 0.558 | 23 |
| Default | – | – | – | 0.750 | 0.750 | 24 |

(c) LETTER (26 classes, 16 attributes, (train,test) = (15000,5000) observations)

| Algorithm | Max Stor | Time (sec.) Train | Time (sec.) Test | Error Rate Train | Error Rate Test | Rk |
|---|---|---|---|---|---|---|
| ALLOC80 | 758 | 39574.7 | – | 0.065 | 0.064 | 1 |
| k-NN | 200 | 14.8 | 2135.4 | 0.000 | 0.068 | 2 |
| LVQ | 377 | 1487.4 | 47.8 | 0.057 | 0.079 | 3 |
| QuaDisc | 80 | 3736.2 | 1222.7 | 0.101 | 0.113 | 4 |
| CN2 | – | 40458.3 | 52.2 | 0.021 | 0.115 | 5 |
| BayTree | 2516 | 275.5 | 7.1 | 0.015 | 0.124 | 6 |
| NewID | 376 | 1056.0 | 2.0 | 0.000 | 0.128 | 7 |
| IndCART | 3600 | 1098.2 | 1020.2 | 0.010 | 0.130 | 8 |
| C4.5 | 1042 | 309.0 | 292.0 | 0.042 | 0.132 | 9 |
| DIPOL92 | 189 | 1303.4 | 79.5 | 0.167 | 0.176 | 10 |
| RBF | 418 | – | – | 0.220 | 0.233 | 11 |
| LogDisc | 316 | 5061.6 | 38.7 | 0.234 | 0.234 | 12 |
| CASTLE | 1577 | 9455.3 | 2933.4 | 0.237 | 0.245 | 13 |
| $AC^2$ | 2033 | 2529.0 | 92.0 | 0.000 | 0.245 | 13 |
| Kohonen | 1204 | – | – | 0.218 | 0.252 | 15 |
| Cal5 | 1554 | 1033.4 | 8.2 | 0.158 | 0.253 | 16 |
| SMART | 881 | 400919.0 | 184.0 | 0.287 | 0.295 | 17 |
| Discrim | 78 | 325.6 | 84.0 | 0.297 | 0.302 | 18 |
| BackProp | 154 | 277445.0 | 22.0 | 0.323 | 0.327 | 19 |
| NaiveBay | 1464 | 74.6 | 17.9 | 0.516 | 0.529 | 20 |
| ITrule | 593 | 22325.4 | 69.1 | 0.585 | 0.594 | 21 |
| Default | – | – | – | 0.955 | 0.960 | 22 |

Quadisc is the best of the classical statistical algorithms on this dataset. This is perhaps not surprising since the measures data gives some support to the assumptions underlying the method. Discrim does not perform well although the logistic version is a significant improvement. SMART is used here with a 22 term model and its poor performance is surprising. A number of the attributes are nonlinear combinations of some others and SMART might have been expected to model this well. ALLOC80 achieves the best performance of all with k-NN close behind. In this dataset all the attributes are pre-scaled and all appear to be important so good performance from k-NN is to be expected. CASTLE constructs a polytree with only one attribute contributing to the classification which is too restrictive with this dataset. Naive Bayes assumes conditional independence and this is certainly not satisfied for a number of the attributes. NewID and $AC^2$ were only trained on 3000 examples drawn from the full training set and that in part explains their rather uninspiring performance. NewID builds a huge tree containing over 1760 nodes while the $AC^2$ tree is about half the size. This difference probably explains some of the difference in their respective results. Cal5 and C4.5 also build complex trees while CN2 generates 448 rules in order to classify the training set. ITrule is the poorest algorithm on this dataset. Generally we would not expect ITrule to perform well on datasets where many of the attributes contributed to the classification as it is severely constrained in the complexity of the rules it can construct. Of the neural network algorithms, Kohonen and LVQ would be expected to perform well for the same reasons as k-NN. Seen in that light, the Kohonen result is a little disappointing. In a previous study Frey and Slate (1991) investigated the use of an adaptive classifier system and achieved a best error rate of just under 0.2.

## B.5 Multiple Features Dataset

This dataset consists of features of handwritten numerals (‘0’–‘9’) extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. These digits are represented in terms of six feature sets: 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Loève coefficients, 240 pixel averages in $2 \times 3$ windows, 47 Zernike moments, and 6 morphological features. Total number of attributes is 649. Table 8 gives the results of twelve classifiers and five classifier combination rules. First, different classifiers were trained on the same 500 ($10 \times 50$) training patterns from each of the 6 feature sets and tested on the same 1,000 ($10 \times 100$) test patterns. The resulting classification errors are reported in the top-left block. For each feature set, the best result over the classifiers is printed in bold. Next, the 12 individual classifiers for a single feature set were combined using five combining rules. It is underlined to indicate that this combination result is better than the performance of individual classifiers for this feature set. Finally the outputs of each classifier and each classifier combination scheme over all the six feature sets are combined using the five combining rules (last five columns).

Table 8: Error rates (%) of different classifiers and classifier combination schemes for the multi-feature digit dataset [44].

| Classifier / | Feature Set | | | | | | Combination Rule | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Combining Rule | No. 1 | No. 2 | No. 3 | No. 4 | No. 5 | No. 6 | Med. | Prod. | Vot. | NM | 1-NN |
| QuaDisc | 25.7 | 5.8 | 12.8 | 6.2 | 21.2 | 31.0 | 2.8 | 6.3 | 6.8 | 6.7 | 5.0 |
| LinDisc | 21.3 | **3.4** | 5.7 | 9.9 | **18.0** | 29.1 | 3.7 | 3.1 | 5.1 | 3.9 | 4.2 |
| Nearest Mean | 22.4 | 18.1 | 9.9 | 9.6 | 27.8 | 54.0 | 6.2 | 4.6 | 7.5 | 10.3 | 4.6 |
| 1-NN | 19.2 | 9.0 | 4.4 | **3.7** | 19.7 | 57.0 | 2.6 | 1.7 | 4.0 | 11.3 | 3.0 |
| k-NN | 18.9 | 9.2 | 4.4 | **3.7** | 19.3 | 51.0 | 5.4 | 4.2 | 5.1 | 3.6 | 2.6 |
| Parzen | **17.1** | 7.9 | **3.7** | **3.7** | 18.5 | 52.1 | 2.9 | 2.7 | 5.1 | 3.1 | 3.1 |
| Fisher | 24.8 | 4.7 | 8.2 | 15.3 | 21.0 | **28.2** | 3.2 | 5.2 | 5.7 | 3.5 | 3.6 |
| Decision Tree | 45.4 | 40.3 | 40.0 | 54.9 | 59.8 | 32.9 | 13.4 | 11.0 | 21.8 | 10.2 | 10.8 |
| ANN-20 | 90.0 | 4.6 | 14.6 | 85.2 | 90.0 | 32.8 | 17.7 | 90.0 | 32.7 | 2.6 | 2.1 |
| ANN-50 | 24.5 | 13.0 | 82.3 | 81.0 | 26.5 | 71.7 | 24.4 | 80.7 | 16.3 | 5.5 | 3.3 |
| SVC-Linear | 24.6 | 6.6 | 6.1 | 7.7 | 29.4 | 84.8 | 10.8 | 10.1 | 4.7 | 6.0 | 5.8 |
| SVC-Quadratic | 21.2 | 5.1 | 4.0 | 6.0 | 19.3 | 81.1 | 3.6 | 3.8 | 3.8 | 4.0 | 4.0 |
| Median | 19.0 | 4.3 | 3.6 | 4.5 | 17.4 | 28.7 | 2.3 | 2.5 | 5.0 | 1.9 | 5.0 |
| Product | 29.4 | 13.1 | 4.4 | 8.2 | 40.1 | 41.2 | 23.4 | 8.6 | 56.8 | 85.1 | 68.5 |
| Voting | 17.5 | 3.5 | 3.2 | 3.7 | 16.9 | 31.8 | 2.3 | 2.0 | 4.8 | 2.1 | 2.0 |
| Nearest Mean | 19.8 | 3.7 | 4.6 | 7.3 | 18.1 | 26.6 | 2.0 | 1.9 | 5.1 | 1.5 | 1.8 |
| 1-NN | 18.6 | 3.8 | 4.1 | 7.2 | 17.0 | 32.8 | 1.8 | 1.8 | 4.1 | 1.9 | 1.8 |

# References


[1] M. Aladjem. Nonparametric discriminant analysis via recursive optimization of Patrick-Fisher distance. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 28(2):292–299, 1998.

[2] S. Amari and S. Wu. Improving support vector machines by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[3] R. E. Bellman. *Adaptive Control Processes*. Princeton Unversity Press, Princeton, NJ, 1961.

[4] A. Biem and S. Katagiri. Filter bank design based on discriminative feature extraction. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 485–488, 1994.

[5] A. Biem, S. Katagiri, and B.-H. Juang. Pattern recognition using discriminative feature extraction. *IEEE Transactions on Signal Processing*, 45(2):500–504, 1997.

[6] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1995.

[7] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[8] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[9] K. D. Bollacker and J. Ghosh. Linear feature extractors based on mutual information. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 720–724, 1996.

[10] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10:209–217, 1998.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

[12] L. J. Buturovic. On the minimal dimension of sufficient statistics. *IEEE Transactions on Information Theory*, 38(1):182–186, 1992.

[13] L. J. Buturovic. Toward Bayes-optimal linear dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):420–424, 1994.

[14] N. Campbell. Canonical variate analysis — a general formulation. *Australian Journal of Statistics*, 26:86–96, 1984.

[15] R. Chengalvarayan and L. Deng. HMM-based speech recognition using state-dependent, discriminatively derived transforms on Mel-warped DFT features. *IEEE Transactions on Speech and Audio Processing*, 5(3):243–256, 1997.

[16] R. D. Cook. Principal Hessian directions revisited (with discussion). *Journal of the American Statistical Association*, 93(441):84–100, 1998.

[17] R. D. Cook. *Regression Graphics: Ideas for Studying Regressions through Graphics*. Wiley, New York, 1998.

[18] R. D. Cook and B. Li. Dimension reduction for conditional mean in regression. *Annals of Statistics*, 30(2), 2002.

[19] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[20] S. Das. Filters, wrappers and a boosting based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81. Morgan Kaufmann, 2001.

[21] S. Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 143–151, Stanford, CA, 2000.

[22] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, Englewood Cliffs, NJ, 1982.

[23] C. Domeniconi and D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[24] R. M. Fano. *Transmission of Information: A Statistical Theory of Communications*. MIT Press, Cambridge, MA, 1961.

[25] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1986.

[26] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[27] K. Fukunaga. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:671–678, 1984.

[28] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 2nd edition, 1990.

[29] M. J. F. Gales. Maximum likelihood multiple subspace projections for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 10(2):37–47, 2002.

[30] R. A. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 661–664, 1998.

[31] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[32] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

[33] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.

[34] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270, 1994.

[35] R. Herbrich. *Learning Kernel Classifiers*. The MIT Press, 2002.

[36] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[37] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.

[38] J. L. Horowitz and W. Hardle. Direct semiparametric estimation of single-index models with discrete covariates. *Journal of the American Statistical Association*, 91(436):1632–1640, 1996.

[39] M. Hristache, A. Juditski, J. Polzehl, and V. Spokoiny. Structure adaptive approach for dimension reduction. *Annals of Statistics*, 29(6):1537–1566, 2001.

[40] M. Hristache, A. Juditsky, and V. Spokoiny. Direct estimation of the index coefficient in a single-index model. *Annals of Statistics*, 29(3):593–623, 2001.

[41] P. J. Huber. Projection pursuit (with discussion). *Annals of Statistics*, 13(2):435–475, 1985.

[42] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

[43] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[44] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[45] L. O. Jimenez and D. A. Landgrebe. Hyperspectral data analysis and supervised feature reduction via projection pursuit. *IEEE Transactions on Geoscience and Remote Sensing*, 37(6):2653–2667, 1999.

[46] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, 1994.

[47] E. M. Kleinberg. An overtraining-resistant stochastic modeling method for pattern recognition. *Annals of Statistics*, 24(6):2319–2349, 1996.

[48] N. Kumar and A. G. Andreou. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communication*, 26(4):283–297, 1998.

[49] B.-C. Kuo and D. Landgrebe. Improved statistics estimation and feature extraction for hyperspectral data classification. Technical Report TR-ECE-01-6, School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, Dec. 2001.

[50] C. Lee and D. A. Landgrebe. Decision boundary feature extraction for nonparametric classification. *IEEE Transactions on Systems, Man and Cybernetics*, 23(2):433–444, 1993.

[51] C. Lee and D. A. Landgrebe. Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):388–400, 1993.

[52] C. Lee and D. A. Landgrebe. Decision boundary feature extraction for neural networks. *IEEE Transactions on Neural Networks*, 8(1):75–83, 1997.

[53] K.-C. Li. Sliced inverse regression for dimension reduction (with discussion). *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[54] K.-C. Li. On principal Hessian directions for data visualization and dimension reduction: Another application of Stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

[55] M. Loog, R. P. W. Duin, and R. Haeb-Umbach. Multiclass linear dimension reduction by weighted pairwise Fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[56] R. Lotlikar and R. Kothari. Adaptive linear dimensionality reduction for classification. *Pattern Recognition*, 33(2):185–194, Feb. 2000.

[57] R. Lotlikar and R. Kothari. Fractional-step dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):623–627, 2000.

[58] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[59] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, 1994.

[60] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[61] T. Okada and S. Tomita. An optimal orthonormal system for discriminant analysis. *Pattern Recognition*, 18(2):139–144, 1985.

[62] J. Peng, D. R. Deisterkamp, and H. K. Dai. LDA/SVM driven nearest neighbor classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 58–64, 2001.

[63] J. L. Powell, J. H. Stock, and T. M. Stoker. Semiparametric estimation of index coefficients. *Econometrica*, 57(6):1403–1430, 1989.

[64] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2):164–171, 2000.

[65] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2001.

[66] A. Samarov. Exploring regression structure using nonparametric functional estimation. *Journal of the American Statistical Association*, 88(423):836–847, 1993.

[67] G. Saon and M. Padmanabhan. Minimum Bayes error feature selection for continuous speech recognition. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2000.

[68] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1129–1132, 2000.

[69] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[70] T. M. Stoker. Consistent estimation of scaled coefficients. *Econometrica*, 54(6):1461–1481, 1986.

[71] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2001.

[72] K. Torkkola. Learning discriminative feature transforms to low dimensions in low dimentions. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[73] K. Torkkola. Nonlinear feature transforms using maximum mutual information. In *Proc. IJCNN*, pages 2756–2761, 2001.

[74] K. Torkkola. On feature extraction by mutual information maximization. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 821–824, 2002.

[75] K. Torkkola and W. M. Campbell. Mutual information in learning feature transformations. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1015–1022. Morgan Kaufmann, San Francisco, CA, 2000.

[76] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[77] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, pages 668–674, 2000.

[78] J. Zhang, Y. Liu, and T. Zhao. SVM based feature screening applied to hierarchical cervial cancer detection. In *International Conference on Diagnostic Imaging and Analysis (ICDIA 2002)*, 2002.