Margin-Based Discriminant Dimensionality Reduction for Visual Recognition

Hakan Cevikalp
Eskisehir Osmangazi University
Meselik Kampusu 26480 Eskisehir Turkey
Hakan.Cevikalp@gmail.com

Bill Triggs
Laboratoire Jean Kuntzmann
B.P. 53, 38041 Grenoble Cedex 9, France
Bill.Triggs@imag.fr

Frédéric Jurie
University of Caen
Université de Caen - F-14032 Caen, France
Frederic.Jurie@unicaen.fr

Robi Polikar
Rowan University
201 Mullica Hill Road, Glassboro NJ USA
polikar@rowan.edu

Abstract

Nearest neighbour classifiers and related kernel methods often perform poorly in high dimensional problems because it is infeasible to include enough training samples to cover the class regions densely. In such cases, test samples often fall into gaps between training samples where the nearest neighbours are too distant to be good indicators of class membership. One solution is to project the data onto a discriminative lower dimensional subspace. We propose a gap-resistant nonparametric method for finding such subspaces: first the gaps are filled by building a convex model of the region spanned by each class – we test the affine and convex hulls and the bounding disk of the class training samples – then a set of highly discriminative directions is found by building and decomposing a scatter matrix of weighted displacement vectors from training examples to nearby rival class regions. The weights are chosen to focus attention on narrow margin cases while still allowing more diversity and hence more discriminability than the 1D linear Support Vector Machine (SVM) projection. Experimental results on several face and object recognition datasets show that the method finds effective projections, allowing simple classifiers such as nearest neighbours to work well in the low dimensional reduced space.

1. Introduction

Although high-dimensional feature spaces often contain the information needed for effective multi-class classification, it can be difficult to exploit this with simple nearest neighbour classifiers or related kernel methods [27, 21]. This is particularly true when the classes have high intrinsic dimensionality: it is seldom feasible to cover such classes densely with training samples and the resulting sparse scatters of samples tend to have many 'holes' – regions that have few or no nearby training samples from the class. When such regions lie close to inter-class boundaries[1] the nearest training samples may lie in the wrong class, thus leading to classification errors. One way to control such effects is to find a small set of particularly discriminant features or directions and project the feature space onto these, thus reducing the dimensionality while still preserving sufficient inter-class separability for classification. Margin-based classifiers [7, 5, 16] such as SVM's and perceptrons are an extreme case in which just one discriminant direction is retained. However two class problems that are not linearly separable and multi-class problems require multi-dimensional projections to provide the necessary separability, which in turn implies the use of non-trivial classifiers within the reduced subspace.

In this paper we study an approach that uses margin-based sample reweighting to find a good set of linear discriminant projection directions. The method begins by 'filling in the holes' by approximating each high-dimensional class with a convex set containing its training samples. This is a reasonable strategy because high-dimensional approximations tend to be simple: for a fixed sample size, the amount of geometric detail that can be resolved usually decreases rapidly as the dimensionality increases [11, 14]. Here we test methods based on the affine hull of the training samples, their convex hull, and their bounding hyper-disk (the intersection of their affine hull and their minimum volume bounding hyper-sphere). For each training sample, we calculate the displacement vector linking it to the closest point on the convex approximation of each other class. The vectors are reweighted to focus attention on the hard-

[1] Inter-class margin $\ll$ hole radius $\approx$ local separation of same-class training samples.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

to-classify narrow margin cases, and informative projection directions are chosen by extracting the dominant directions of the resulting weighted set using singular value decomposition of the set itself or eigendecomposition of the corresponding scatter matrix. Samples are classified by projecting them into the discriminant subspace and applying a conventional classifier: here we used nearest neighbour classifiers for simplicity but any other nonlinear classifier could be used.

To illustrate the possible applications of the proposed method, we provide experimental results on several high-dimensional visual recognition tasks including face recognition. This is a good illustration because face images vary along many dimensions, even for a single person, and there are typically only a few training images for each person. The results show that the proposed method finds highly discriminant low dimensional projections that compare favourably to existing methods.

1.1. Related Work

Many dimensionality reduction methods have been applied to face recognition. Principal Component Analysis (PCA) [26] and Fisher Linear Discriminant Analysis (FLDA) [2] are both popular, but they have their deficiencies. The PCA optimizes for reconstruction rather than discrimination, and FLDA is constrained both by its Gaussian-with-equal-covariance construction and by its inability to handle degenerate class covariances (prior projection onto the span of the covariances is possible but it tends to remove much of the discriminant information).

Local neighbourhood based nonlinear dimensionality reduction techniques such as Isomap [25], Locally Linear Embedding [20], Stochastic Neighbor Embedding [13] and Laplacian Eigenmaps [3] have also been used, but these are essentially descriptive methods not discriminative ones. In fact, as initially defined they handle only training data. To incorporate test samples an additional local transformation matrix must be learned, *e.g.* as in Neighborhood Components Analysis (NCA) [9] or Locality Preserving Projections (LPP) [12], and the resulting projection is still not designed for optimal discrimination. Moreover, all of these methods typically rely on unlabelled nearest neighbours to construct their local neighbourhood graphs so they are subject to hole artifacts just like other nearest neighbour methods.

One way to reduce the influence of holes is to take account of known degrees of variability in the classes, either introducing a distance function that downweights the most variable directions (tangent distance in the local case, Mahalanobis distance in the global one) or explicitly generating and including additional training samples by deforming existing ones according to the expected variation model [22]. This works to some extent, but there is a limit to the number of such samples that can be handled, particularly when the expected variations are high-dimensional. If a well-adapted distance function is available it should certainly be used, but we still prefer to explicitly fill in the holes using a convex class model.

Some techniques learn a distance metric that ensures that samples with the same (or semantically similar) labels remain close while samples with different labels are pushed apart [29,6]. However, learning a general distance metric in high-dimensional spaces is impractical as the number of parameters to be estimated is the square of the dimensionality. Thus, in high-dimensional feature spaces these methods must be preceded by dimensionality reduction, which may remove a lot of the discriminant information. In fact, such methods are typically more suitable for retrieval purposes rather than classification.

A method that is more closely related to ours is Margin Maximizing Discriminant Analysis (MMDA) [16]. This attempts to preserve as much discriminant information as possible by projecting the dataset onto margin maximizing directions (separating hyperplane normals) found by an SVM algorithm. Like SVM, MMDA is intrinsically a two-class approach and the best strategy for generalizing it to multiple classes is unclear.

The above methods all select generic projection directions (linear combinations of input features). Another class of methods is designed to select useful subsets of input features (information gain, mutual information, odds ratio, *etc.*) [10,30]. Such variable selection methods usually require combinatorial search and we will not consider them here.

2. Method

Our approach is based on bounding each training class c with a convex set H_c and using the weighted displacement vectors between the training samples and these sets to choose suitable projection directions. We first describe how we estimate the projection given the bounding sets, then we consider several kinds of bounding sets. Let $\mathbf{P}(\mathbf{x}, H)$ denote the projection of a point $\mathbf{x}$ on a convex set H (*i.e.* the point in H that lies closest to $\mathbf{x}$), $\mathbf{dx}(\mathbf{x}, H) = \mathbf{x} - \mathbf{P}(\mathbf{x}, H)$ denote the displacement vector from $\mathbf{x}$ to H, $d(\mathbf{x}, H) = \|\mathbf{dx}(\mathbf{x}, H)\|$ denote the corresponding point-set distance and $\hat{\mathbf{d}}(\mathbf{x}, H) = \mathbf{dx}(\mathbf{x}, H)/d(\mathbf{x}, H)$ denote the corresponding unit-norm displacement direction. The basic intuition is that the displacement vectors from a class to the nearby training examples of other classes are useful projection directions because they allow the class to be separated from its nearby rivals. Compare this to SVM, which uses a single projection direction – the displacement between the convex hulls of the training examples of its two classes. This is problematic when the classes are not linearly separable

and also in the multiclass case. Our aims are more modest. We do not necessarily expect the classes to be linearly separable, but we would like to find a set of projection directions that is rich enough to allow the projected classes to be separated with a generic classifier (nearest neighbour, nonlinear SVM, *etc*.) while still remaining sufficiently low dimensional to avoid the gap effect.

To implement this we take our projection directions to be the largest few eigenvectors of the weighted scatter matrix of the normalized training displacements over all classes

$$\mathbf{S} \;=\; \sum_{c\neq c'=1}^{C} \sum_{i=1}^{N_c} \frac{w(\mathbf{x}_{ci},H_{c'})}{N_c}\, \hat{\mathbf{d}}(\mathbf{x}_{ci},H_{c'})\, \hat{\mathbf{d}}(\mathbf{x}_{ci},H_{c'})^\top \quad (1)$$

or equivalently the largest few singular vectors of the matrix of the weighted training displacements

$$\left[\ldots, \sqrt{\frac{w(\mathbf{x}_{ci},H_{c'})}{N_c}}\, \hat{\mathbf{d}}(\mathbf{x}_{ci},H_{c'}), \ldots\right]_{c\neq c'\in 1\ldots C, i=1\ldots N_c} \quad (2)$$

Here, $\mathbf{x}_{ci} \in \mathbb{R}^d$ are the training samples, $c = 1,\ldots,C$ indexes the C classes and $i = 1,\ldots,N_c$ indexes the N_c samples of class c. $w(\mathbf{x},H)$ is a heuristic weighting function that focuses attention on the training examples that are most relevant for the separation. It is typically a decreasing function of the point-class distance $d(\mathbf{x},H)$. By default we will use a simple exponential

$$w(\mathbf{x},H) \;=\; \exp\left(-d(\mathbf{x},H)/q\right) \quad (3)$$

where q is a global scale parameter that needs to be set by cross-validation. We have also experimented with more scale-independent local weighting functions of the form

$$w(\mathbf{x}_{ci},H_{c'}) \;=\; 2\,\frac{\min\{d(\mathbf{x}_{ci},H_c^{\text{knn}})^\alpha, d(\mathbf{x}_{ci},H_{c'}^{\text{knn}})^\alpha\}}{d(\mathbf{x}_{ci},H_c^{\text{knn}})^\alpha + d(\mathbf{x}_{ci},H_{c'}^{\text{knn}})^\alpha}, \quad (4)$$

where H_c^{knn} and $H_{c'}^{\text{knn}}$ are respectively the convex bounding sets of $\mathbf{x}_{ci}$'s k nearest neighbors in classes c and c' and α is a sharpness parameter. These functions typically have sigmoid-like behaviour, remaining close to 1 within c and near the decision boundary and dropping fairly rapidly to zero as we move away from the boundary.

Given $\mathbf{S}$, we take its largest few eigenvectors as the projection directions, *i.e*. we find a rectangular orthogonal matrix $\mathbf{U}$ that maximizes $J(\mathbf{U}) = \text{trace}(\mathbf{U}^\top\mathbf{S}\mathbf{U})$. We either take sufficiently many projection directions to ensure that a given fraction (typically 90–98%) of the overall energy (sum of the eigenvalues) is retained, or set the number to optimize a performance metric such as cross-validated classification error.

2.1. Affine Hull (AH) Case

We now specialize to the case where each class is approximated by the affine hull of its training examples

$$H_c^{\text{aff}} \;=\; \left\{\mathbf{x} = \textstyle\sum_{i=1}^{N_c} \alpha_i\, \mathbf{x}_{ci} \;\middle|\; \sum_i \alpha_i = 1\right\} \quad (5)$$

We suppose that the affine hulls have dimension less than d, so they are proper subsets of $\mathbb{R}^d$. This necessarily holds for small training sets in high dimensions, $N_c \ll d$, and in practice we find that the affine hull method often works surprisingly well in this case despite the fact that it provides only a rather loose bound on the training samples. The projection of a point $\mathbf{x}$ onto H_c^{aff} gives a displacement vector of the form $\mathbf{dx}(\mathbf{x},H_c) = \mathbf{P}_c^\perp(\mathbf{x}-\boldsymbol{\mu}_c) = (\mathbf{I}-\mathbf{P}_c)(\mathbf{x}-\boldsymbol{\mu}_c)$ where $\mathbf{P}_c$ projects along the directions spanned by the vectors within H_c^{aff}, $\mathbf{P}_c^\perp$ is the complementary orthogonal projection and $\boldsymbol{\mu}_c$ is the mean of the class c training samples (or any other reference point within H_c^{aff}). Note that the projection vector $\boldsymbol{\mu}_c^\perp = \mathbf{P}_c^\perp\boldsymbol{\mu}_c$ of $\boldsymbol{\mu}_c$ is not zero in general – it encodes the orthogonal displacement of the affine subspace H_c from the origin.

Numerically, $\mathbf{P}_c = \mathbf{Q}_c\,\mathbf{Q}_c^\top$ where $\mathbf{Q}_c$ is the U matrix of the thin SVD (or equivalently the Q matrix of the thin QR decomposition) of the matrix of centered class-c training examples $[\mathbf{x}_{c1}-\boldsymbol{\mu}_c,\ldots,\mathbf{x}_{cN_c}-\boldsymbol{\mu}_c]$. This allows point projections to be computed on the fly using $\mathbf{Q}_c$ without explicitly evaluating and storing the $d\times d$ projection matrices $\mathbf{P}_c$ and $\mathbf{P}_c^\perp$.

In practice the training data is often somewhat noisy. This can lead to the inclusion of spurious 'noise' dimensions within the affine hulls, which can harm inter-class discriminability. To reduce this we suppress dimensions of the SVD (and hence of $\mathbf{Q}_c$) that correspond to overly small singular values.

2.2. Convex Hull (CH) Case

The affine hull gives a rather loose approximation to the class region because it does not constrain the position of the training points within the affine subspace. Alternatively we can take a maximally tight bound by approximating the class region with the convex hull of the training samples. To do this we need to include non-negativity constraints $\alpha_i \geq 0$, $i = 1,\ldots,N_c$ in (5) and replace the affine displacement and distance computations with convex hull based ones. Given a query point, finding the closest point on a convex hull in general requires the solution of a quadratic programming problem [27]. However for classes whose training samples are affinely independent, the hull is a simplex and we can compute the closest point with a simple elimination procedure: successively project the input point onto the affine span of the training samples (simplex vertices), write the result as a linear combination of these with weights α_i, and eliminate the vertex with the most

negative weight. Continue until all vertices have positive weights α_i. The final projection point is the desired output.

In the two class case our convex method has some similarities to a classical linear SVM. The SVM finds the minimum distance vector between the convex hulls of the two classes [8,4] and projects the dataset onto this unique direction. All of the weight is concentrated on the points of closest approach of the hulls (expressed as convex combinations of support vectors) and only one projection direction is used. In contrast, our convex method finds distance vectors from the convex hull of each class to the training samples (not the convex hull) of the other classes and uses a weighting that grades off more gradually with distance to find a number of promising projection directions using all of the data points, not just the closest points on the convex hulls as in SVM.

2.3. Bounding Disk

If the affine hull gives a rather loose approximation to the class, the convex hull method often gives an over-tight one because training examples are necessarily rather sparse in high dimensions and the class region is likely to extend well beyond their convex hull. As a third alternative we develop an approximation based on intersecting the affine hull of the training points with their bounding hypersphere (the smallest hypersphere enclosing all of them). This gives hyper-disk shaped class regions that can be computed economically and that support rapid nearest point computations. We are not aware of any other machine learning methods based on bounding hyper-disks, but bounding hyperspheres have often been used for both outlier detection [24] and classification [28].

The minimum bounding hypersphere of a point set $\{\mathbf{x}_i \,|\, i = 1, \ldots, N\}$ is characterized by its center $\mathbf{c}$ and radius r. These can be found by solving the following quadratic optimization problem

$$\min_{\mathbf{c}, r^2, \xi} \left(r^2 + \gamma \sum_i \xi_i \right) \quad \text{s.t.} \quad \forall i \quad ||\mathbf{x}_i - \mathbf{c}||^2 \leq r^2 + \xi_i, \tag{6}$$

or its dual

$$\begin{aligned} \min_{\alpha} & \left(\sum_{i,j} \alpha_i \alpha_j \, \mathbf{x}_i \cdot \mathbf{x}_j - \sum_i \alpha_i \, \|\mathbf{x}_i\|^2 \right) \\ \text{s.t.} & \quad \sum_i \alpha_i = 1, \quad \forall i \quad 0 \leq \alpha_i \leq \gamma. \end{aligned} \tag{7}$$

Here, the α_i are Lagrange multipliers and $\gamma \in [0, 1]$ is a ceiling parameter that can be set to a finite value to eliminate over-distant points as outliers [24]. The center of the hypersphere is then $\mathbf{c} = \sum_i \alpha_i \, \mathbf{x}_i$ and its radius is $r = \|\mathbf{x}_i - \mathbf{c}\|$ for any $\mathbf{x}_i$ with $0 < \alpha_i < \gamma$.

Given an arbitrary point $\mathbf{x}$, the closest point on the hyper-disk can be computed by projecting $\mathbf{x}$ onto the affine subspace supporting the disk, then, if the projected point $\mathbf{x}_{\text{aff}} = \mathbf{P}(\mathbf{x}, H^{\text{aff}})$ is not already within the disk, moving along the line from the projection to the disk center until the boundary of the disk is crossed, *i.e.* $\mathbf{P}(\mathbf{x}, H^{\text{disk}})$ is $\mathbf{x}_{\text{aff}}$ if $\|\mathbf{x}_{\text{aff}} - \mathbf{c}\| \leq r$ and $\mathbf{c} + \frac{r}{\|\mathbf{x}_{\text{aff}} - \mathbf{c}\|}(\mathbf{x}_{\text{aff}} - \mathbf{c})$ otherwise.

3. Experiments

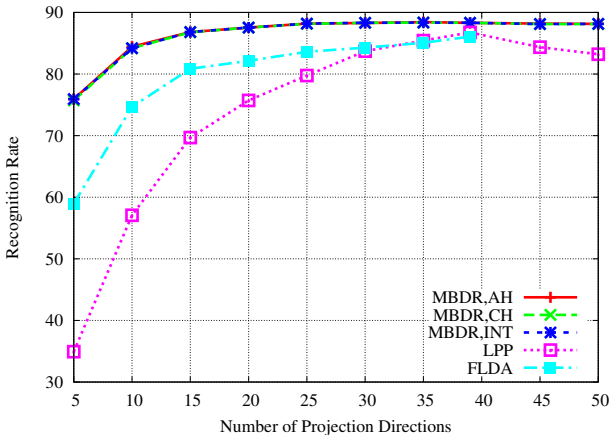
We illustrate the proposed affine hull ('MBDR-AH'), convex hull ('MBDR-CH'), and hyper-disk ('MBDR-INT') methods with experiments on four high-dimensional supervised classification tasks from visual recognition, comparing them to Fisher Linear Discriminant ('FLDA'), Locality Preserving Projections ('LPP') [12] and linear Support Vector Machines ('SVM'). For all four tasks the within-class scatter matrices have rank less than d owing to the high dimensionality of the visual feature space. Both FLDA and LPP require nonsingular scatter matrices so before applying them we used the PCA of the total scatter to project the training data onto a subspace in which the within-class scatter had full rank. LPP used the heat kernel method to compute its weights and the k-nearest neighbors from each class to construct its adjacency graph. We tested both the exponential and the local weighting functions (3,4) and they yielded similar results. We report the results for the exponential weighting function here. The outlier detection in the hyper-disk method was disabled by setting the ceiling parameter γ to 1. For multi-class SVM we used the one-against-all approach. For each method we optimized the algorithm parameters using a global coarse-to-fine search, using random partitions of the training data into training and validation sets unless the data set designates a validation set for this purpose.

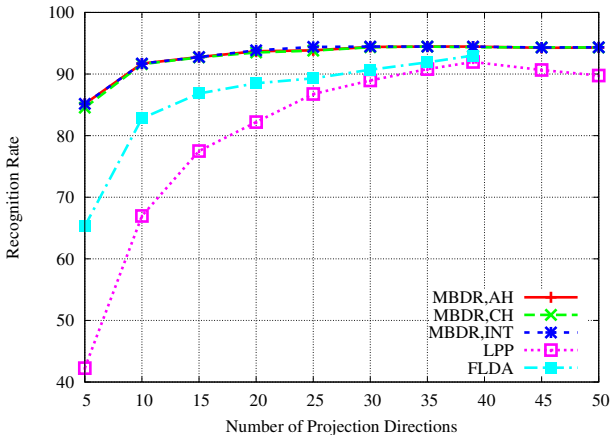
To demonstrate how effective our method is at 'filling in the holes' we used a very simple classifier – nearest neighbours (NN) – in the reduced space. We also ran some tests using SVM's and distances to the affine or convex hulls of the training samples in the reduced space, but for brevity we do not report on these here.

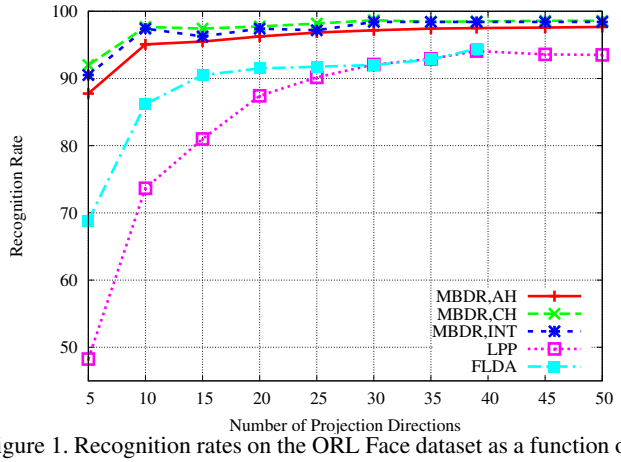
3.1. ORL Face Dataset

The Olivetti-Oracle Research Lab (ORL) face dataset[2] contains 10 upright frontal face images per person of $C = 40$ individuals. The images are 92×112. They were taken at different times and they have slightly different lighting conditions, image positions, facial expressions and facial details. For this experiment we used the raw image pixels as input features without applying any visual preprocessing. For training we randomly selected $k = 3, 5, 7$ images of each individual, keeping the remaining $10 - k$ for testing.

[2] From *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*.

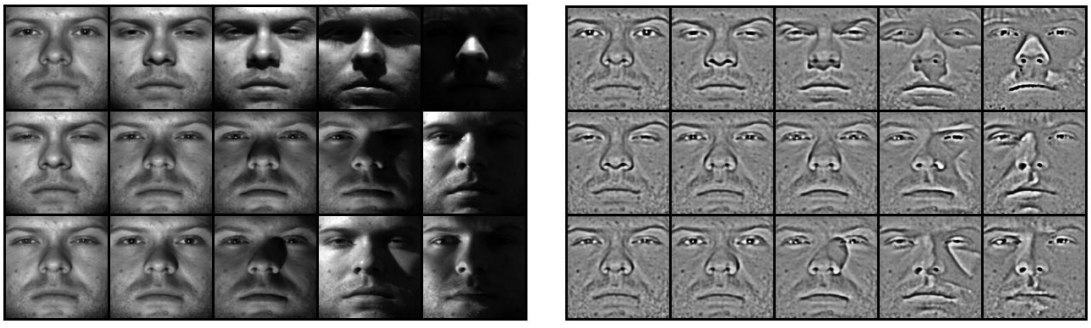






Methods	$k=3$	$k=5$	$k=7$
MBDR-AH	**88.43%**	94.45%	97.67%
MBDR-CH	88.36%	94.45%	**98.67%**
MBDR-INT	**88.43%**	94.45%	98.42%
SVM	87.11%	**95.20%**	97.58%
FLDA	86.07%	93.00%	94.42%
LPP	86.75%	91.95%	94.08%

Figure 1. Recognition rates on the ORL Face dataset as a function of number of projection directions for $k = 3$ (top left), $k = 5$ (top right) and $k = 7$ (bottom left) training examples per subject, with testing on the remaining $10 - k$. (Bottom right) The best computed recognition accuracies.



(a) (b)

Figure 2. Some image samples from extended Yale-B face database: (a) original images; (b) the corresponding illumination normalized images.

We tested the methods with various numbers of projection directions, in all cases using a Euclidean NN classifier in the reduced space for classification. The reported recognition rates are averages over 10 random test/training splits.

The results are shown in figure 1. As can be seen, our proposed methods outperform the other linear dimension methods in all cases, particularly for low-dimensional projections. For $k = 3, 5$ all three methods gave very similar results with the affine hull based methods having a slight lead. For $k = 7$ the convex hull method was preferred. Our proposed methods have similar performance to a classical SVM classifier – all of the proposed methods slightly out-

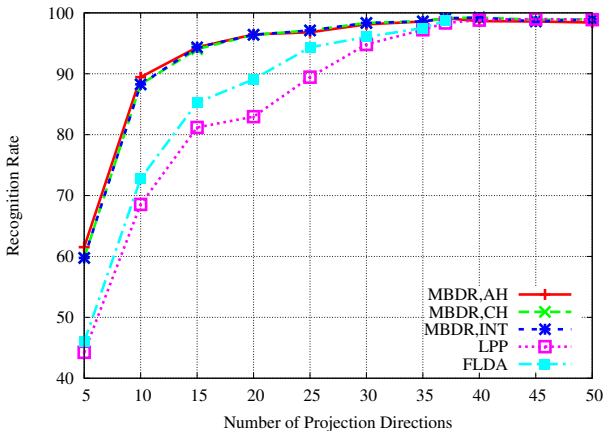


Figure 3. Recognition rates on the extended Yale-B face dataset as a function of the number of projection directions.

perform SVM for $k = 3, 7$ while SVM wins for $k = 5$.

3.2. Extended Yale-B Face Dataset

To study the performance of the methods in a more challenging and realistic scenario, we also tested them on the Extended Yale Face dataset[3]. This includes 38 subjects, each seen from 9 camera poses under 64 illumination conditions that range from straightforward to very challenging. The images are divided into five subsets according to the angle between the light source direction and the central camera axis. Our experiments used the frontal images, with subsets 1 and 2 (frontal and near frontal lighting) for training and subsets 3-5 (increasingly severe tangential lighting) for testing. A robust visual feature set is necessary for acceptable results on this dataset. After aligning and scaling the images so that the centers of the two eyes always fall at fixed coordinates and cropping the results to 120×120, we preprocessed the images to reduce the effects of illumination variations using the method of [23]. This involves strong gamma compression, difference of Gaussian (DoG) filtering with well-chosen inner and outer scales, robust normalization of the resulting range of output variations, and strong sigmoid-based compression to reduce the effects of any remaining isolated peaks such as specularities. These steps greatly reduce the influence of illumination variations, local shadowing and highlights while still preserving the essential elements of visual appearance that are needed for recognition. Some pre-processed images are shown in figure 2. Finally, Local Binary Histogram features [1] were computed to give a robust but very high-dimensional visual feature set for recognition.

Recognition rates as a function of the number of projection directions are given in figure 3. Again our methods give the best results. The proposed methods and SVM all had an asymptotic error rate of 0.07% on this dataset. To the best of our knowledge these are state of the art results.

3.3. Visual Object Recognition Problems

We also tested the proposed methods on two object recognition datasets: Pascal Challenge 2006[4] and Birds [17]. The Pascal dataset contains images of 10 object categories. For each category there is a set of natural images containing objects of the category, taken from assorted viewpoints and scales under various lighting conditions and often with partial occlusion. There is substantial intra-class variation in all of the categories. The goal is to predict whether or not objects from the category are present in each test image. We tested the methods on the first four categories – bicycles, buses, cars, and cats. In all there were 2618 training images and 2686 test images. As before, the nearest neighbour method was used for classification in the reduced subspace. For SVM we trained a binary one-against-all classifier for each category, *i.e.* the training images from the category being learned were positives and all of the remaining training images were negatives. Since this is a binary classification problem, we also tested the MMDA method.

The Birds dataset contains six categories, each with 100 images. It is a challenging dataset with highly cluttered backgrounds and large intra-class, scale and viewpoint variations. We used 5-fold cross validation to test the generalization performance. As before a nearest neighbor classifier is used in the reduced space.

We used a "bag of features" image representation for both datasets as they are too diverse to allow simple geometric alignment of their objects. In bag of features methods, patches are sampled from the image at many different positions and scales, either densely, randomly or based on the output of some kind of salient region detector. Each patch is described using the robust visual descriptor SIFT [18] and vector quantized using nearest neighbour assignment against a visual dictionary learned from the complete set of training patches [15]. The "visual words" of each image are then histogrammed to make a representation analogous to the bag-of-words one used in document analysis. Despite the fact that they only encode image geometry rather weakly, bag of features representations turn out to be very effective in content based image classification tasks. Experience shows that the most critical factor is the number of patches sampled not the sample selection method [19], so for simplicity we randomly sampled thousands of patches per image. The size of the dictionary was 5000 words for the Pascal Challenge dataset and 2000 for the Birds one.

Our experimental results on the Pascal Challenge 2006 dataset are summarized in table 1. Recognition rates are

[3] From *http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html*.

[4] From *http://www.pascal-network.org/challenges/VOC/voc2006/index.html*.

Methods	Bicycles	Buses	Cars	Cats
MBDR,AH	**90.3**, [9]	94.6, [9]	**95.8**, [9]	**89.2**, [9]
MBDR,CH	89.8, [7]	94.4, [13]	94.0, [7]	85.2, [10]
MBDR,INT	**90.3**, [9]	**94.8**, [10]	**95.8**, [11]	**89.2**, [9]
FLDA	82.3, [1]	91.3, [1]	89.0, [1]	78.7, [1]
LPP	77.8, [10]	87.7, [15]	92.1, [20]	80.0, [15]
MMDA	86.7, [10]	92.1, [13]	94.5, [10]	85.4, [8]
SVM	88.6	93.6	95.0	86.6

Table 1. Recognition Accuracies (%) in terms of AUC on the Pascal Challenge Dataset.

Methods	Recognition Accuracies (%)
MBDR,AH	**93.83**, $\sigma = 1.62$, [6]
MBDR,CH	92.99, $\sigma = 2.38$, [6]
MBDR,INT	**93.83**, $\sigma = 1.62$, [6]
SVM	93.46, $\sigma = 3.47$
FLDA	91.50, $\sigma = 2.59$, [5]
LPP	92.66, $\sigma = 2.32$, [15]

Table 2. Recognition Accuracies and Standard Deviations for the Birds Dataset.

given in terms of areas under the ROC curves (AUC). The number of selected projection directions is given in square brackets. The affine hull and hyper-disk methods significantly outperform the other methods tested including SVM, particularly for low-dimensional projections. The convex hull is less good but still outperforms SVM on the Bicycles and Buses categories.

Table 2 gives the recognition rates for the Birds dataset. The number of projection directions that were selected is shown in square brackets. Again the affine hull and hyper-disk methods come equal first. SVM comes second and the convex hull method third. The identical performance of the affine hull and hyper-disk methods suggests that the hyper-sphere bounds are usually inactive, *i.e*. the projections of training samples onto the affine hulls of the other classes typically lie within those classes hyperspheres.

3.4. Discussion

Despite its simplicity, the affine hull method regularly outperformed the convex hull one in the experiments. This can be attributed to the high dimensionality of the input spaces. For classes that span general regions in high-dimensional feature spaces, most of their volume is typically outside of, and often quite far from, the convex hull of the available training examples. For example, a simplex spanned by points sampled from a high-dimensional sphere can include only a negligible fraction of the volume of the sphere, even if the vertices themselves are well spaced and close to the surface of the sphere. Filling out the complete boundaries of the support region of the class would typically require a number of training samples that is exponential in the classes affine dimension, which is infeasible in practice. Hence, in many cases the affine hull and the hyper-disk approximations are better guides to the region that might be spanned by the class than the convex hull. Similar comments apply to other convex hull based methods such as linear SVM: owing to sparse sampling in high dimensions, true class boundaries often extend well beyond the hyperplane of the observed support vectors into the margin.

4. Summary and Conclusion

We have proposed a discriminative linear dimensionality reduction method that attempts to preserve separability by choosing projection directions that are well-aligned with inter-class margins while suppressing directions orthogonal to these. The method works by approximating each class with a convex set containing its training samples, accumulating a weighted scatter matrix of the vectors separating each sample from the sets of its rival classes over all training samples and classes, using the dominant eigenvectors of the scatter matrix as projection directions and finally applying a simple classifier (nearest neighbours in our case) in the reduced space. The separation vector weights are designed to focus attention on samples that lie close to rival classes. We tested three methods of this type based respectively on the affine hull, the convex hull, and the bounding hyper-disk (the intersection of affine hull and the minimal bounding sphere) of the training samples. Experiments on four high-dimensional visual recognition problems suggested that the proposed methods – particularly the affine hull and hyper-disk ones – have better performance than most other linear projection based approaches and that they are competitive with SVM while allowing the easy integration of new classes.

Ongoing work. For high-dimensional feature spaces, the eigenvalue problem for the scatter matrix can be transformed into a smaller one with a matrix of size $M(C-1) \times M(C-1)$, where M denotes the size of the training set. However this may be still problematic if the training set size M is too large. One way to deal with this is to ignore any sample-class pairs whose separation is greater than some given threshold. Another approach would be to directly use the geometric nearest distance separation between manifolds in either a one-against-one or a one-against-all manner, which would result in 'easy' eigendecompositions of size at most $C(C-1) \times C(C-1)$ or $C \times C$. This should allow the proposed schemes to be scaled to problems involving many training samples. We are also working on incorporating the kernel trick into our framework. Given that affine hulls gave better results than convex hulls in many of our experiments, the kernelization of affine hull based

methods might yield more efficient classifiers than SVM in terms of both accuracy and computational complexity.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE TPAMI*, 28(12), 2006.

[2] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 2001.

[4] K. P. Bennett and E. J. Bredensteiner. Duality and geometry in SVM classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 57–64. Morgan Kaufmann, San Francisco, CA, 2000.

[5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discovery*, 2:121–167, 1998.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2005.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] A. Dax. The distance between two convex sets. *Linear Algebra and its Applications*, 416:184–213, 2006.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. *Advances in Neural Information Processing Systems*, 2005.

[10] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[11] P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society series B*, 67(3):427–444, 2005.

[12] X. He and P. Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 2003.

[13] G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, 2002.

[14] L. O. Jimenez and D. A. Landgrebe. Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transcations on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 28(1):39–54, 1998.

[15] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Int. Conf. Computer Vision*, Oct. 2005.

[16] A. Kocsor, K. Kovacs, , and C. Szepesvari. Margin maximizing discriminant analysis. *ECML/PKDD*, pages 227–238, 2004.

[17] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. *International Conference on Computer Vision*, 2005.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004.

[19] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, pages IV 490–503. Springer, 2006.

[20] S. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[21] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[22] P. Simard, Y. Le, J. Denker, and B. Victorri. Transformation invariance in pattern recognition - tangent distance and tangent propagation. *Lecture Notes in Computer Science*, 1524:239–274, 1998.

[23] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modelling of Faces and Gestures*, pages 168–182, Oct. 2007.

[24] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.

[25] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[26] M. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[27] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, 2001.

[28] J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. In *Discovery Science*, pages 241–252, 2005.

[29] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 2005.

[30] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, 1997.