Large Margin Classifiers Based On Convex Class Models

Hakan Cevikalp
Eskisehir Osmangazi University
Meselik Kampusu, 26480, Eskisehir, Turkey

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

Abstract

We propose large margin classifiers that are sometimes better than Support Vector Machines (SVMs) for high-dimensional classification problems with limited numbers of training samples. The basic idea is to approximate each class with a convex model of some form based on its training samples. For any pair of models of this form, there is a corresponding linear classifier that maximizes the margin between the models, and this can be found efficiently by solving a convex program that finds the closest pair of points in the two sets. If the classes are modeled with the convex hulls of their samples the result is the standard SVM, but many other convex models are possible. As examples we investigate maximum margin classifiers based on affine hull and bounding hyperdisk models. These methods can also be kernelized by working in orthonormal coordinates on the subspace of feature space spanned by the training samples. We compare the resulting margin-between-convex-model methods to SVM and to the corresponding nearest-convex-model classifiers on several data sets, showing that they are often competitive with these well-established approaches.

1. Introduction

High-dimensional classification problems with limited numbers of training samples are becoming increasingly common in various fields including computer vision, text classification and genetic microarrays. Simple locality based classifiers such as Nearest Neighbors (NN) tend to perform poorly in such problems because the available training samples do not suffice to cover the high-dimensional class regions densely. The resulting sparse and irregular distributions of samples leave many "holes", leading to classification errors when the nearest sample happens to have the wrong class [16].

Two kinds of strategies for dealing with this have been very successful. The first seeks low-dimensional projections that suffice to separate the data. The prototypical method of this kind is the two-class Support Vector Machine (SVM) [1,4], which seeks a 1D projection that maximizes the separation ("margin") between the boundaries of the projected class samples. Intuitively this works because in low dimensions, a comparatively small number of samples suffices to densely fill, and hence characterize, the regions spanned by the classes.

The second strategy attempts to "fill in the holes" in the original high dimensional space by using the training samples to build a simple geometric model of each class that approximates the region spanned by it better than the isolated samples themselves. The most successful approaches of this kind have used various kinds of convex models including linear subspaces [18,12], affine hulls [16,7,6], convex hulls [11], bounding hyperspheres [15] and bounding hyperdisks [3][1]. Convex models are natural choices because they intrinsically "fill in the holes" and because the computations that are needed to classify new examples such as point-model or model-model distances are efficient owing to convexity.

The methods just cited are "nearest neighbor" ones – or more precisely "nearest convex model" ones – in the sense that they classify new examples to the class whose convex model is nearest to the example. This paper investigates an alternative "margin between convex model" strategy that is based on explicitly building linear maximum margin separators between pairs of convex models. As a first example of the power of this approach, note that the SVM itself is the maximum margin linear separator between the convex hulls of the training samples of the two classes [1].

One motivation for replacing nearest-convex-model approaches with margin-based ones is that for all of the above cited approaches, the pairwise decision boundaries (surfaces equidistant from the two convex models) are generically at least quadratic or piecewise quadratic in complexity. For example for affine hulls they are generically hyperboloids. Such decision boundaries are more flexible than linear ones, but in high dimensions when the training data is scarce this may lead to overfitting, thus damaging gen-

[1] The affine hull is the smallest affine (shifted linear) subspace containing the training samples, the convex hull is the smallest convex set containing them, the bounding hypersphere is the smallest spherical ball containing them, and the bounding hyperdisk is the disk shaped region formed by the intersection of the affine hull and the bounding hypersphere.

eralization to unseen examples. Linear margin based approaches have fewer degrees of freedom, so they are typically less sensitive to the precise arrangement of the training samples. For example for an SVM classifier, motions of the SVM support vectors parallel to the SVM decision surface do not alter the margin and hence do not invalidate the classifier (although they might allow an even better one to be found), whereas they do typically change the piecewise quadratic decision surface of the equivalent nearest-convex-hull classifier.

There are also reasons for thinking that for some problems, SVM may not be the best way to formulate margin based classification. The convex hull model on which it is based is the tightest possible convex approximation to the training samples. For classes with more generous convex forms, it is typically a substantial under-approximation. For example for classes that are ellipsoids or boxes in high dimensions and for any placement of any number of samples sub-exponential in the dimension, the volume of their convex hull is exponentially smaller than the volume of the class. Similarly, for Gaussians the convex hull of any probable placement of a sub-exponential number of samples contains exponentially little probability mass, while for classes supported by affine hulls with long-tailed distributions of samples within the hull, the extent of the class can be almost infinitely underestimated by any finite number of training samples.

Natural classes almost always "bulge beyond the samples" in this sense, leading to fitted margins that sometimes substantially over-estimate the true inter-class margin. This does not invalidate the usual SVM performance bounds – which essentially guarantee that the volumes or probabilities of the "scalps" cut off by the over-estimated margins are almost always exponentially smaller than the corresponding class volumes or probabilities – but it does suggest that models that take better account of the true form of the classes might provide somewhat better inter-class separators. Indeed, particularly for problems with small numbers of samples in high dimensions, the experiments below confirm that the affine hull and bounding hyperdisk margin machines often compare favourably with the equivalent SVM (convex hull margin machine). A similar conclusion was reached for the corresponding nearest-convex-model classifiers in [3,16] – convex hulls of samples are often outperformed by simpler convex models such as affine hulls or hyperdisks.

Another motivation for studying margin-between-convex-model approaches is their potential flexibility and compactness. The affine, hypersphere and hyperdisk models allow each class to be fitted individually and represented compactly, following which the linear separator between any two classes can be found quickly by an efficient convex calculation. This allows classes to be easily added, removed or modified. In contrast, updating SVM based representations potentially requires the storage of every sample on the convex hull of each class, with SVM retraining following each modification. This can be very expensive.

1.1. Related Work

Our work generalizes the SVM [1,4] maximum margin approach to convex class models that differ from the convex hull of the samples. In particular we develop the affine hull [16] and bounding hyperdisk [3] margin classifiers and provide direct experimental comparisons with the corresponding nearest-convex-model methods tested in [3].

Cevikalp et al. [2] discuss a related margin based feature extraction method in which weighted separation vectors from samples to convex models are used to find a set of projection directions (features) that provide good multi-class discrimination under nearest neighbour classification.

There are many other linear discriminant based classifiers such as perceptrons, decision trees, *etc*., but we will not consider these here as they are not based on (simple) underlying geometric models of the classes. Our models are compromises chosen to provide reasonable representational power with modest and well-controlled complexity, thus avoiding the full combinatorial complexity (and potential for overfitting) of generic high-dimensional geometric models. Similarly, in high dimensional problems there is seldom enough data to do more than roughly delimit the region occupied by the class, so we typically prefer a geometric (region based) approach to a probabilistic (density modelling) one.

2. General Approach

Basically, our method consists of using the training samples of each class to build a convex model of the given type (affine hull, bounding hyperdisk, *etc*.) for it, then finding the maximum margin linear separator for the two models, *i.e*. the vector $\mathbf{w}$ with minimal norm such that $\mathbf{w}^\top \mathbf{x} + b \geq 1$ for all points $\mathbf{x}$ in the first class and $\mathbf{w}^\top \mathbf{x} + b \leq -1$ for all points $\mathbf{x}$ in the second class. The final decision rule assigns test points to the first class iff $\mathbf{w}^\top \mathbf{x} + b > 0$.

$\mathbf{w}$ can be estimated by finding a closest pair of points between the two models – *i.e*. points $\mathbf{x}_+$ in the first model and $\mathbf{x}_-$ in the second such that $\|\mathbf{x}_+ - \mathbf{x}_-\|$ is as small as possible – and taking $\mathbf{w} = 2(\mathbf{x}_+ - \mathbf{x}_-)/\|\mathbf{x}_+ - \mathbf{x}_-\|^2$ and $b = (\|\mathbf{x}_-\|^2 - \|\mathbf{x}_+\|^2)/\|\mathbf{x}_+ - \mathbf{x}_-\|^2$. The closest point problem reduces to the convex program minimizing $\|\mathbf{x}_+ - \mathbf{x}_-\|^2$ under within-class-model constraints for $\mathbf{x}_+$ and $\mathbf{x}_-$. In degenerate cases the solution for $\mathbf{x}_+$ and $\mathbf{x}_-$ may not be unique, however the corresponding minimal-norm $\mathbf{w}$ is always unique. In this paper we will always assume that the class models do not intersect one another, so that such pairs of points always exist and separation with a positive margin

is possible. (If not, the notion of margin can be extended to negative values by defining $\mathbf{w}$ to be the smallest-norm translation vector that just separates the two classes. But we will not pursue this here).

The above construction is for the two-class case. To handle multi-class problems, any of the standard strategies for extending SVM's can be used. Below we test the two most popular approaches: *One-Against-Rest* (OAR) and *One-Against-One* (OAO). For a c-class problem, the OAR strategy trains c binary classifiers, one to separate each class from the remaining $c-1$ classes. Each classifier is trained on the entire training set. Test samples are classified to the class whose classifier gives the highest output in the ensemble. In contrast, the OAO strategy constructs $c(c-1)/2$ classifiers, one for each possible pair of classes, in each case using only the training data for that particular pair. New examples are classified by a majority vote algorithm, with each OAO classifier casting a vote for its preferred class for the example. Other notable approaches include Directed Acyclic Graphs [13] and Binary Decision trees [17], but we will not test these here.

3. Affine Hull Model

Now consider the case where each class is modelled by the affine hull of its training samples, *i.e.* the smallest affine subspace containing them. This is an unbounded, and hence typically rather loose, model for the class. The affine hull of samples $\{\mathbf{x}_i\}_{i=1,\ldots,n}$ contains all points of the form $\sum_{i=1}^n \alpha_i \mathbf{x}_i$ with $\sum_{i=1}^n \alpha_i = 1$. Below it will be more convenient to write this explicitly as $\{\mathbf{x} = \mathbf{U}\mathbf{v} + \mu \,|\, \mathbf{v} \in \mathbb{R}^l\}$, where $\boldsymbol{\mu} = (1/n)\sum_i \mathbf{x}_i$ is the mean of the samples (or any other reference point in the hull) and $\mathbf{U}$ is an orthonormal basis for the directions spanned by the affine subspace. The vector $\mathbf{v}$ contains the reduced coordinates of the point within the subspace, expressed with respect to the basis $\mathbf{U}$. Numerically, $\mathbf{U}$ can be found as the U-matrix of the 'thin' Singular Value Decomposition (SVD) of $[\mathbf{x}_1 - \boldsymbol{\mu}, \ldots, \mathbf{x}_n - \boldsymbol{\mu}]$. Here, 'thin' indicates that we take only the columns of U corresponding to "significantly non-zero" singular values λ_k. l is the number of such non-zero singular values.

The above subspace estimation process is essentially orthogonal least squares fitting. Discarding near-zero singular values corresponds to discarding directions that appear to be predominantly "noise". Unfortunately, least squares estimates are sensitive to outliers in the training samples, which will typically force unwanted directions to be added to the hull. As an alternative, the samples can be fitted with some other more robust subspace estimation process. In the experiments we test both 'L2' methods based on conventional least squares and 'L1' methods based on the L1-norm Rotationally Invariant PCA of Ding et al. [5].

Now suppose that we have two affine hulls with point sets $\{\mathbf{U}_+ \mathbf{v}_+ + \boldsymbol{\mu}_+\}$ and $\{\mathbf{U}_- \mathbf{v}_- + \boldsymbol{\mu}_-\}$. (These can be estimated with either L2 or L1 fitting and they may have different numbers of dimensions l, but we assume that the hulls are non-intersecting). A closest pair of points between the two hulls can be found by solving

$$\min_{\mathbf{v}_+,\mathbf{v}_-} ||(\mathbf{U}_+\mathbf{v}_+ + \boldsymbol{\mu}_+) - (\mathbf{U}_-\mathbf{v}_- + \boldsymbol{\mu}_-)||^2. \quad (1)$$

Defining $\mathbf{U} \equiv \begin{pmatrix}\mathbf{U}_+ & -\mathbf{U}_-\end{pmatrix}$ and $\mathbf{v} \equiv \begin{pmatrix}\mathbf{v}_+ \\ \mathbf{v}_-\end{pmatrix}$, this can be written as the standard least squares problem

$$\min_{\mathbf{v}} ||\mathbf{U}\,\mathbf{v} - (\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)||^2 \quad (2)$$

whose solution is $\mathbf{v} = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)$, *i.e.* $\mathbf{w} \propto \mathbf{x}_+ - \mathbf{x}_- = (\mathbf{I} - \mathbf{P})(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$ where $\mathbf{P} = \mathbf{U}\,(\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top$ is the orthogonal projection onto the joint span of the directions contained in the two subspaces and $\mathbf{I} - \mathbf{P}$ is the corresponding projection onto the orthogonal complement of this span[2].

4. Hyperdisk Case

The hyperdisk model of a class [3] is the intersection of the affine hull and the smallest bounding hypersphere of its training samples. It thus consists of points of the form $\sum_{i=1}^n \alpha_i \mathbf{x}_i$ where $\sum_{i=1}^n \alpha_i = 1$ as before and in addition $\|\sum_{i=1}^n \alpha_i \mathbf{x}_i - \mathbf{c}\|^2 \le r^2$. Here, $\mathbf{c}$ is the center of the bounding hypersphere and r is its radius. The hyperdisk can found by solving a quadratic program that minimizes r under the above constraints, with $\mathbf{c}$ as free variables. Alternatively, given an orthonormal basis $\mathbf{U}$ for the affine hull of the points $\{\mathbf{x}_i\}$ as above, the hyperdisk can be written $\{\mathbf{x} = \mathbf{U}\,\mathbf{v} + \mathbf{c} \,|\, \|\mathbf{v}\|^2 \le r^2\}$, where each training point $\mathbf{x}_i$ must have an expansion $\mathbf{v}_i$ of this form. The disk can again be found by solving a quadratic program that minimizes r – see *e.g.* [3].

Given two (non-intersecting) hyperdisks $(\mathbf{c}_+, \mathbf{U}_+, r_+)$ and $(\mathbf{c}_-, \mathbf{U}_-, r_-)$, we can find their maximum margin separator by using Lagrange multipliers to find points $\mathbf{v}_+, \mathbf{v}_-$ that minimize the inter-disk distance

$$\arg\min_{\mathbf{v}_+,\mathbf{v}_-} \|(\mathbf{U}_+\,\mathbf{v}_+ + \mathbf{c}_+) - (\mathbf{U}_-\,\mathbf{v}_- + \mathbf{c}_-)\|^2 \quad (3)$$

subject to $\|\mathbf{v}_+\|^2 \le r_+^2$ and $\|\mathbf{v}_-\|^2 \le r_-^2$. Introducing Lagrange multipliers $\lambda_+ - 1$ and $\lambda_- - 1$ for the two inequality constraints, this reduces to solving the linear system

$$\begin{pmatrix}\lambda_+\mathbf{I} & -\mathbf{U}_+^\top\mathbf{U}_- \\ -\mathbf{U}_-^\top\mathbf{U}_+ & \lambda_-\mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{v}_+ \\ \mathbf{v}_-\end{pmatrix} = \begin{pmatrix}-\mathbf{U}_+^\top \\ \mathbf{U}_-^\top\end{pmatrix}(\mathbf{c}_+ - \mathbf{c}_-) \quad (4)$$

[2] If the two subspaces share common directions, $\mathbf{U}^\top\mathbf{U}$ is not invertible and the solution for $(\mathbf{v}_+, \mathbf{v}_-)$ and $(\mathbf{x}_+, \mathbf{x}_-)$ is non-unique, but the orthogonal complement remains well defined, giving a unique minimum norm separator $\mathbf{w}$. Numerically all cases can be handled by finding $\tilde{\mathbf{U}}$, the U matrix of the thin SVD of $\mathbf{U}$, and taking $\mathbf{P} = \tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}$.

subject to $\lambda_+ \geq 1$, $\lambda_- \geq 1$. (Here λ_+, λ_- are one plus the usual Lagrange multipliers, with the ones coming from the inclusion of $\mathbf{U}_+^\top \mathbf{U}_+ = \mathbf{I}$ and $\mathbf{U}_-^\top \mathbf{U}_- = \mathbf{I}$ terms on the diagonal of the matrix).

We now need to choose (λ_+, λ_-) (≥ 1) so that the solution of this system satisfies $\|\mathbf{v}_+\|^2 \leq r_+^2$, $\|\mathbf{v}_-\|^2 \leq r_-^2$. This can be done efficiently by a 2D Newton root-finding process analogous to the 1D one used by eigenvalue finders. First, note that by changing coordinates to the principal angle basis between the supporting affine subspaces of the hyperdisks, and hence diagonalizing the off-diagonal blocks of the matrix, the linear system can be reduced to a set of decoupled 2×2 subsystems. The SVD of $\mathbf{U}_+^\top \mathbf{U}_-$ gives the necessary linear bases, with the singular values being the cosines of the corresponding principal angles. (In cases where there is no principal angle or the angle is 90 degrees, the equations separate further to sets of two 1×1 ones). For any given (λ_+, λ_-), we can solve these equations in closed form to find $(\mathbf{v}_+, \mathbf{v}_-)$.

To find (λ_+, λ_-) we run a Newton root finding iteration on the 2D equation $(\mathbf{v}_+^\top \mathbf{v}_+, \mathbf{v}_-^\top \mathbf{v}_-) = (r_+^2, r_-^2)$ with $(\mathbf{v}_+, \mathbf{v}_-)$ as functions of (λ_+, λ_-). This holds for $(\lambda_+, \lambda_-) > 1$. If either λ_+ or λ_- becomes 1 (indicating that the solution is interior to the corresponding disk so that the norm constraint on $\mathbf{v}$ is inactive), we use the Newton method to solve the corresponding 1D equation for the remaining variable. In either case the process converges rapidly, usually within about 3-5 iterations.

Finally, given $(\mathbf{v}_+, \mathbf{v}_-)$ it is trivial to find the closest pair of points in the two disks and the maximum margin separator $\mathbf{w} \propto (\mathbf{U}_+ \mathbf{v}_+ + \mathbf{c}_+) - (\mathbf{U}_- \mathbf{v}_- + \mathbf{c}_-)$ with $\mathbf{w}^\top \mathbf{x} + b \geq 1$ for disk 1, ≤ -1 for disk 2.

5. Kernelization

The above methods are based on convex models and linear separation in the input feature space. If more nonlinear classifiers are needed – *e.g*. because the feature space is too low-dimensional to allow the linear separation of all classes – it is straightforward to kernelize them. Basically this is just a matter of using the training samples to construct an explicit orthogonal basis for the finite dimensional subspace of the (possibly infinite dimensional and implicit) kernel feature space that they span. This allows all of the required (intrinsic Euclidean geometry) constructions to be performed explicitly in finite dimensional subspace coordinates. Test samples are projected orthogonally into the same subspace by kernel evaluation against the training samples, so their orthogonal deviations from the subspace are irrelevant. The bases need to be constructed using the training samples of all participating classes, *i.e*. of the pair of classes being separated for OAO methods and of all classes for OAR methods. Additional samples (subspace dimensions) can be included if desired, although this makes the computation more expensive and potentially less well conditioned.

In detail – *c.f*. KPCA [14] – let $\boldsymbol{\phi}(\cdot)$ be the implicit feature space embedding and $k(\mathbf{x}, \mathbf{y}) = \boldsymbol{\phi}^\top(\mathbf{x})\, \boldsymbol{\phi}(\mathbf{y})$ be the corresponding kernel function. Suppose that we want to project points $\mathbf{x}$ onto the affine hull of a given set of samples $\{\mathbf{x}_i\}_{i=1,\ldots,m}$. Let $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_m)]$ be their feature space embedding matrix, $\mathbf{K} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi} = [k(\mathbf{x}_i, \mathbf{x}_j)]$ be their $m \times m$ kernel matrix and $\mathbf{k}_\mathbf{x} = \boldsymbol{\Phi}^\top \boldsymbol{\phi}(\mathbf{x}) = [k(\mathbf{x}_i, \mathbf{x})]$ be the $m \times 1$ kernel vector of $\mathbf{x}$ against the samples. The feature space mean of the samples is $\boldsymbol{\mu} = \frac{1}{m} \boldsymbol{\Phi}\, \mathbf{1}_m$ (where $\mathbf{1}_m$ is an m-vector of 1's). So the matrix of centered sample features is $[\boldsymbol{\phi}(\mathbf{x}_1) - \boldsymbol{\mu}, \ldots, \boldsymbol{\phi}(\mathbf{x}_m) - \boldsymbol{\mu}] = \boldsymbol{\Phi}\, \boldsymbol{\Pi}$, where $\boldsymbol{\Pi} = \mathbf{I} - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^\top$ is the orthogonal projection in sample space that implements subtraction of the mean on $\boldsymbol{\Phi}$. If $\boldsymbol{\phi}$ were an explicit embedding, the thin SVD $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ of $\boldsymbol{\Phi}\, \boldsymbol{\Pi}$ would yield an orthogonal basis $\mathbf{U} = \boldsymbol{\Phi}\, \boldsymbol{\Pi}\, \mathbf{V}\, \mathbf{D}^{-1} = \boldsymbol{\Phi}\, \mathbf{A}^\top$ for the affine subspace, where $\mathbf{A} = \mathbf{D}^{-1} \mathbf{V}^\top \boldsymbol{\Pi}$. (We use a 'thin' SVD containing only the significantly non-zero singular values, so $\mathbf{D}$ is invertible). Although we can not calculate this SVD explicitly, we can get the same result by taking the corresponding thin eigendecomposition $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$ of the centred kernel matrix $\widetilde{\mathbf{K}} = (\boldsymbol{\Phi}\, \boldsymbol{\Pi})^\top (\boldsymbol{\Phi}\, \boldsymbol{\Pi}) = \boldsymbol{\Pi}\, \mathbf{K}\, \boldsymbol{\Pi}$ and defining $\mathbf{D} = \boldsymbol{\Lambda}^{1/2}$, *i.e*. $\mathbf{A} = \boldsymbol{\Lambda}^{-1/2} \mathbf{V}^\top \boldsymbol{\Pi}$. In either case, the projection of a new sample $\mathbf{x}$ onto $\mathbf{U}$ coordinates in the affine hull is then simply $\mathbf{U}^\top \boldsymbol{\phi}(\mathbf{x}) = \mathbf{A}\, \mathbf{k}_\mathbf{x}$.

Alternatively we can work in terms of linear spans rather than affine ones, when necessary subtracting the mean explicitly after reduction to $\mathbf{U}$ coordinates. The same formulae apply with $\boldsymbol{\Pi}$ omitted.

6. Experiments

We tested linear and kernelized versions of the proposed Affine Hull Margin Classifier (AHMC) and Hyperdisk Margin Classifier (HDMC) on a number of data sets, comparing them to SVM (*i.e*. the Convex Hull Margin Classifier) and to the corresponding nearest-convex-set classifiers from [3]: Nearest Affine Hull (NAH), Nearest Hyperdisk (NHD), and Nearest Convex Hull (NCH). For the AHMC and HDMC methods, we compared two methods of estimating the underlying affine subspaces, standard least squares ('-L2' variants) and L1 norm Rotationally Invariant PCA [5] ('-L1' variants). In each case we estimated the subspace dimension by retaining enough leading eigenvectors to account for 95–98% of the total energy in the eigendecomposition. For the multi-class margin classifiers we tested both One-Against-Rest (OAR) and One-Against-One (OAO) approaches, reporting the results of whichever was best.

The linear versions of the above methods were tested on an instructive toy problem and on visual recognition problems from the AR Face, Birds, Coil-100 and Xerox-10 data sets. The kernelized versions were tested on five low-dimensional problems from the UCI repository.

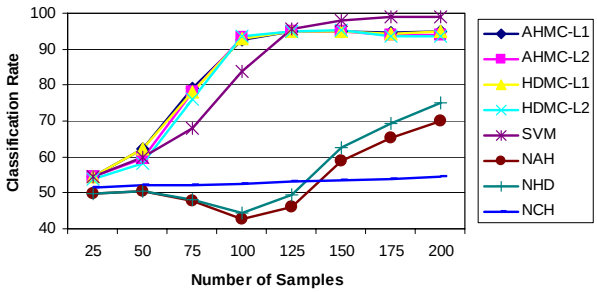

Figure 1. Classification rates as a function of number of training samples for the 'Pancake' data set.

Figure 2. Aligned images of one subject from the AR Face data set.

6.1. Experiments on Synthetic 'Pancake' Data Set

We begin by illustrating some properties of the proposed methods with experiments on the 4-class synthetic data set from [3]. This was produced by creating four unit-radius spheres (one for each class) in 300 dimensions with centres $(\pm 0.2, \pm 0.2, 0, \ldots, 0)$, sampling test and training points uniformly within each sphere, then compressing 200 of the dimensions including the second one by a factor of 10. This produces a high dimensional data set containing two aligned stacks of two 100D-pancake like classes, with many irrelevant variables.

Beginning with $n = 25$ samples per class, we gradually increased the number of training samples up to 200, using the remaining $1000 - n$ for testing. The results are shown in Fig. 1, using OAR for the margin based classifiers. With these numbers of samples, the nearest convex set classifiers perform poorly compared to the margin based ones. All of the margin based methods have similar accuracy for $n = 25, 50$, but the affine hull and hyperdisk ones dominate for $n = 75, 100$, only to be surpassed by SVM for $n \geq 150$ (at which point the samples start to become dense enough to fill out the convex hulls of the classes to a significant extent). The L1 and L2 subspace estimates yield similar performance.

6.2. Experiments on the AR Face Data Set

The AR Face data set [10] contains 26 frontal images with different facial expressions, illumination conditions and occlusions for each of 126 subjects, recorded in two 13-image sessions spaced by 14 days. For this experiment we randomly selected 20 male and 20 female subjects. The images were down-scaled (from 768×576), aligned so that the centers of the two eyes fell at fixed coordinates, then cropped to size 105×78. Fig. 2 shows some of the resulting images of one subject. Raw pixel values were used as features. For training we randomly selected $n = 7, 13, 20$ samples for each individual, keeping the remaining $26 - n$ for testing. This process was repeated 10 times, with the final classification rates being obtained by averaging the 10 results.

AR	$n = 7$	$n = 13$	$n = 20$
AHMC-L1	95.12 ± 0.5	98.88 ± 0.2	99.62 ± 0.3
AHMC-L2	95.19 ± 0.6	$\mathbf{98.95} \pm 0.3$	99.62 ± 0.3
HDMC-L1	$\mathbf{95.32} \pm 0.7$	98.86 ± 0.5	99.74 ± 0.2
HDMC-L2	95.20 ± 0.7	98.80 ± 0.4	$\mathbf{99.76} \pm 0.2$
SVM	94.54 ± 0.6	98.66 ± 0.2	99.58 ± 0.3
NAH	76.45 ± 1.8	95.96 ± 1.8	98.85 ± 0.9
NHD	75.62 ± 1.6	95.05 ± 2.1	98.65 ± 0.7
NCH	60.73 ± 0.9	80.00 ± 2.3	92.08 ± 1.8

Table 1. Classification Rates (%) for the linear methods on the AR Face data set.

The results are shown in Table 1. The performance differences are most apparent for $n = 7$. As in the previous experiment, the large margin classifiers outperformed the nearest convex set ones, with the OAR variants being preferred for the margin based methods. Among the nearest convex set classifiers, NAH and NHD clearly outperformed NCH. Similarly, the AHMC (affine hull) and HDMC (hyperdisk) margin classifiers outperformed SVM (convex hull margin classifier), with the L1 and L2 variants of AHMC and HDMG all giving similar results. Overall these results support our claims, suggesting that affine hulls and hyperdisks can be better models for representing classes in high-dimensional spaces when the number of samples is limited.

6.3. Experiments on the Birds Data Set

The Birds data set contains 100 images each of six species of birds in the wild [8]. It is a challenging visual recognition task with the birds appearing against highly cluttered backgrounds and the images having large intra-class, scale, and viewpoint variability. We use a "bag of features" representation for the images as they are too diverse to allow simple geometric alignment of their objects. In this method, patches are sampled from the image at many different positions and scales (densely, randomly or based

Birds	$n = 25$	$n = 50$	$n = 75$
AHMC-L1	88.80 ± 1.3	92.00 ± 1.1	91.93 ± 2.0
AHMC-L2	88.78 ± 1.3	91.94 ± 1.2	91.73 ± 2.1
HDMC-L1	$\mathbf{89.62} \pm 1.4$	92.27 ± 1.3	92.53 ± 1.8
HDMC-L2	89.60 ± 1.2	92.50 ± 1.1	$\mathbf{92.60} \pm 1.7$
SVM	89.53 ± 1.3	$\mathbf{92.53} \pm 1.3$	92.20 ± 1.3
NAH	87.24 ± 1.6	90.43 ± 1.1	91.67 ± 1.7
NHD	87.24 ± 1.6	90.43 ± 1.1	91.67 ± 1.7
NCH	87.22 ± 1.5	90.77 ± 1.3	92.27 ± 1.8

Table 2. Classification Rates (%) for the linear methods on the Birds data set.

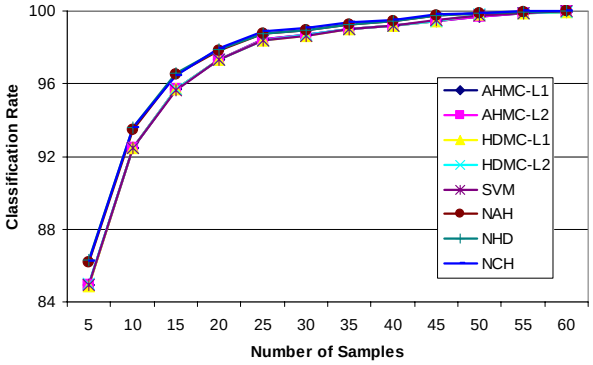

Figure 3. Classification rates as a function of number of training samples for the Coil data set.

on the output of some salient region detector), coded, and pooled to characterize the image. Here we used a dense grid of patches, with each patch being coded using the robust visual descriptor SIFT [9] and vector quantized using nearest neighbor assignment against a 2000 word visual dictionary learned from the complete set of training patches. For training we randomly selected $n = 25, 50, 75$ images of each class, keeping the remaining $100-n$ for testing. Table 2 summarizes the results, which are again averages of 10 random training/test splits. For the margin based methods, the OAR approach again gave the best accuracies. Overall, HDMC achieves the best classification rates for $n = 25, 75$ while SVM wins for $n = 50$. The margin based methods again outperform the nearest convex set ones, and HDMC outperforms AHMC, but the differences in performance are small, especially for $n = 75$.

6.4. Experiments on the Coil100 Object Data Set

The Coil100 data set[3] includes 72 views each of 100 small objects against a flat background, taken at 5° orientation intervals on a turntable. The size of each image is 128×128. We randomly chose 40 objects from the data set. We used raw grayscale pixel values as input features without applying any further visual preprocessing. For training we randomly selected $n = 5, 10, ..., 60$ images of each object, keeping the remaining $72 - n$ for testing. As before, we report averages over 10 random test/training splits – see Fig. 3. For the margin based methods, the OAO approach gave the best performance. All of the margin based classifiers gave almost identical results, as did all of the nearest convex set classifiers, with the latter being the better performers. (In contrast, for the OAR approach, SVM slightly outperformed HDMC and both significantly outperformed AHMC – presumably because the large 'Rest' classes are overapproximated unless a tightly fitting convex model is used).

[3] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

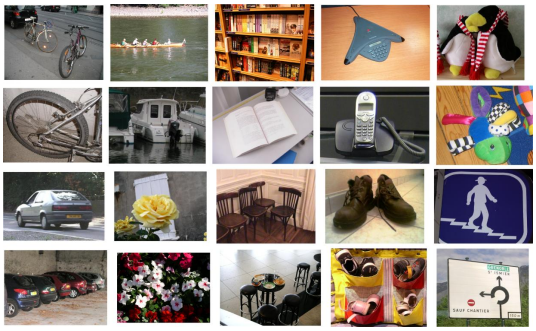

Figure 4. Some images from the Xerox10 data set.

6.5. Experiments on the Xerox10 Data Set

The Xerox10 data set[4] contains 10 objects classes (bikes, boats, books, cars, chairs, flowers, phones, road signs, shoes, and soft toys) and a total of 3105 images. Some examples are shown in Fig. 4. We removed 6 mislabeled images, leaving 3099 in the data set used.

The classification task is quite challenging as the backgrounds are cluttered and the objects have substantial intra-class variations, highly variable pose and lighting, and occasional occlusions. We again used a SIFT based "bag of features" representation, setting the dictionary size to 2000. We used $20n$ samples per class for training, for $n = 1, \ldots, 10$. The results are shown in Fig. 5. These are again averages over 10 random training/test splits. For the margin based methods, the best accuracies were obtained with the OAO approach. NCH was the best performer in all cases tested. Among the remaining methods, NAH and NHD are best until about $n = 100$ when SVM takes over. AHMC and HDMC are not the best performers, but they do achieve the same performance as SVM up to about $n = 80$.

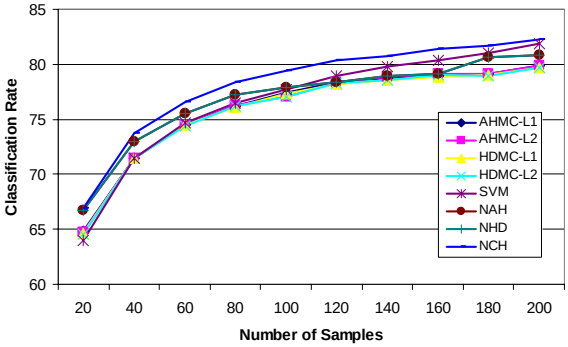

Figure 5. Classification rates as a function of number of training samples for the Xerox10 data set.

UCI	Iris	IS	MF	Wine	WDBC
AHMC-L1	**96.7**	96.1	98.2	**98.8**	96.3
AHMC-L2	**96.7**	96.0	98.3	**98.8**	96.7
HDMC-L1	**96.7**	96.1	98.3	**98.8**	96.5
HDMC-L2	**96.7**	96.0	98.3	**98.8**	96.7
SVM	95.3	**97.1**	97.6	97.2	97.5
NAH	**96.7**	95.7	**98.4**	96.7	95.3
NHD	**96.7**	96.0	**98.4**	96.7	96.3
NCH	96.0	95.7	98.2	97.8	**97.7**

Table 3. Classification Rates (%) for the kernelized methods on the UCI data sets.

6.6. Experiments on the UCI Data Sets

The above experiments all use linear classifiers. We also tested the kernelized versions of the methods on five lower-dimensional data sets from the UCI repository: Iris; Image Segmentation (IS); Multiple Features pixel averages (MF); Wine; and Wisconsin Diagnostic Breast Cancer (WDBC). We used Gaussian kernels with widths set by 5-fold cross-validation for all experiments. OAO was used for multi-class problems. The results, shown in Table 3, are quite mixed. The L2 and L1 variants of AHMC and HDMC perform similarly on all of these experiments, achieving the best accuracies on two of the five data sets, with SVM winning on one of the remaining sets and NAH/NHD winning on the two others.

7. Summary and Conclusion

We investigated the idea of basing large margin classifiers on convex models of the participating classes, in particular studying margin classifiers based on affine hull and bounding hyperdisk models as alternatives to the SVM (convex hull large margin classifier) for high-dimensional classification tasks. We also contrasted these margin-between-convex-model approaches to the corresponding nearest-convex-model approaches tested in [3].

Given two convex models, their corresponding margin based classifier is easily determined by finding a closest pair of points on the two models and bisecting the displacement between them. Such classifiers can also be kernelized, and the extension to multi-class classification is straightforward using any of the standard approaches such as One-Against-One or One-Against-Rest.

The experimental results provided useful insights on the potential application areas of the proposed methods. Although the proposed affine (AHMC) and hyperdisk (HDMC) margin classifiers did not come first in every experiment, in the ensemble they appear to be competitive with SVM and the nearest-convex-model approaches (NAH, NHD, NCH), particularly for high dimensional problems with very limited amounts of training data. In general the hyperdisk method does at least as well as the affine one and often a little better, presumably owing to its tighter and somewhat more realistic convex model. (The same conclusion was reached for the nearest-convex-model approach in [3]).

At least in the tests reported here, replacing least squares subspace fitting with L1 distance based fitting made little difference. However, particularly for the affine hull methods, the performance can be sensitive to the estimated dimension of the subspaces so it may pay to test a range of these.

There was no clear winner between OAR and OAO for multi-class problems, but in OAR, if the 'Rest' class becomes too diverse its convex model can overlap the 'One' class, causing OAR to fail badly. For this reason, OAR tends to prefer tighter class representations such as convex hulls and hyperdisks rather than affine hulls.

[4] ftp://ftp.xrce. xerox.com/pub/ftp-ipc

Acknowledgment

This work is supported by the Young Scientists Award Programme (TÜBA-GEBİP/2009-10) of the Turkish Academy of Sciences.

References

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

[2] H. Cevikalp, B. Triggs, F. Jurie, and R. Polikar. Margin-based discriminant dimensionality reduction for visual recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[3] H. Cevikalp, B. Triggs, and R. Polikar. Nearest hyperdisk methods for high-dimensional classification. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 120–127, New York, NY, USA, 2008. ACM.

[4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[5] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 2006.

[6] M. B. Gulmezoglu, V. Dzhafarov, and A. Barkana. The common vector approach and its relation to principal component analysis. *IEEE Trans. Speech Audio Proc.*, 9:655–662, 2001.

[7] J. Laaksonen. *Subspace classifiers in recognition of handwritten digits*. PhD thesis, Helsinki University of Technology, 1997.

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

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] A. M. Martinez and R. Benavente. The AR face database. Technical report, Computer Vision Center, Barcelona, Spain, 1998.

[11] G. I. Nalbantov, P. J. F. Groenen, and J. C. Bioch. Nearest convex hull classification. Technical report, Econometric Institute and Erasmus Research Institute of Management, 2007.

[12] E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.

[13] J. C. Platt, N. Cristianini, and J. Shawe-taylor. Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems*, pages 547–553. MIT Press, 2000.

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

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

[16] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *NIPS*, 2001.

[17] V. Vural and J. G. Dy. A hierarchical method for multi-class support vector machines. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 105, New York, NY, USA, 2004. ACM.

[18] S. Watanabe, P. F. Lambert, C. A. Kullikowski, J. L. Buxton, and R. Walker. Evaluation and selection of variables in pattern recognition. *Comput. Inf. Sci. II*, pages 91–122, 1967.