HAL
open science

Large margin classifiers based on affine hulls

Hakan Cevikalp, Bill Triggs, Hasan Serhan Yavuz, Yalçin Küçük, Mahide Küçük, Atalay Barkana

To cite this version:

Hakan Cevikalp, Bill Triggs, Hasan Serhan Yavuz, Yalçin Küçük, Mahide Küçük, et al.. Large margin classifiers based on affine hulls. Neurocomputing, 2010, 73 (16-18), pp.3160-3168. 10.1016/j.neucom.2010.06.018 . hal-00564987

HAL Id: hal-00564987

https://hal.science/hal-00564987

Submitted on 10 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LARGE MARGIN CLASSIFIERS BASED ON AFFINE HULLS

Hakan Cevikalp[a], Bill Triggs[b], Hasan Serhan Yavuz[a], Yalcin Kucuk[c], Mahide Kucuk[c], Atalay Barkana[d]

[a]*Electrical and Electronics Engineering Department of Eskisehir Osmangazi University, Meselik 26480 Eskisehir, Turkey*
[b]*Laboratoire Jean Kuntzmann, Grenoble, France*
[c]*Mathematics Department of Anadolu University, Eskisehir, Turkey*
[d]*Electrical and Electronics Engineering Department of Anadolu University, Eskisehir, Turkey*

Abstract

This paper introduces a geometrically inspired large-margin classifier that can be a better alternative to the Support Vector Machines (SVMs) for the classification problems with limited number of training samples. In contrast to the SVM classifier, we approximate classes with affine hulls of their class samples rather than convex hulls. For any pair of classes approximated with affine hulls, we introduce two solutions to find the best separating hyperplane between them. In the first proposed formulation, we compute the closest points on the affine hulls of classes and connect these two points with a line segment. The optimal separating hyperplane between the two classes is chosen to be the hyperplane that is orthogonal to the line segment and bisects the line. The second formulation is derived by modifying the ν-SVM formulation. Both formulations are extended to the nonlinear case by using the kernel trick. Based on our findings, we also develop a geometric

Email addresses: hakan.cevikalp@gmail.com (Hakan Cevikalp), Bill.Triggs@imag.fr (Bill Triggs), hsyavuz@ogu.edu.tr (Hasan Serhan Yavuz), ykucuk@anadolu.edu.tr (Yalcin Kucuk), mkucuk@anadolu.edu.tr (Mahide Kucuk), atalaybarkan@anadolu.edu.tr (Atalay Barkana)

interpretation of the Least Squares SVM classifier and show that it is a special case of the proposed method. Multi-class classification problems are dealt with constructing and combining several binary classifiers as in SVM. The experiments on several databases show that the proposed methods work as good as the SVM classifier if not any better.
Keywords: Affine hull, classification, convex hull, kernel methods, large margin classifier, quadratic programming, support vector machines.

---

1. Introduction

The Support Vector Machine (SVM) classifier is a successful binary classification method that simultaneously minimizes the empirical classification error and maximizes the geometric margin, which is defined as the distance between the separating hyperplane and closest samples from the classes [2, 8]. To do so, SVM first approximates each class with a convex hull and finds the closest points in these convex hulls [1]. Then, these two points are connected with a line segment. The hyperplane, orthogonal to the line segment that bisects the line, is chosen to be the separating hyperplane. From the geometrical point of view, in the separable case, the two closest points on the convex hulls determine the separating hyperplane, and the SVM margin is merely equivalent to the minimum distance between the convex hulls that represent classes. However, convex hull approximations tend to be unrealistically tight in high-dimensional spaces since the classes typically extend beyond the convex hulls of their training samples. For example, a convex hull constructed by randomly sampled points from a high-dimensional hypersphere can include only a negligible fraction of the volume of the sphere even if the chosen samples are well spaced and close to the surface of the sphere

[3]. This situation may also be observed when the low-dimensional data samples are mapped to a higher-dimensional feature space through kernel mapping during estimation of the nonlinear decision boundaries between classes.

As opposed to the convex hulls, affine hulls (i.e., spanning linear subspaces that have been shifted to pass through the centroids of the classes) give rather loose approximations to the class regions, because they do not constrain the positions of the training points within the affine subspaces. Therefore, they may be better alternatives to convex hulls for some pattern classification problems especially when the data samples lie in high-dimensional spaces. In the context of classification, affine hulls are first used as global classifiers of isolated word and hand-written digits giving good classification performance [11, 14]. In these methods, each class is approximated with an affine hull constructed from its training samples, and the label of a test sample is determined based on the distance to the nearest affine hull. Vincent and Bengio [26] used affine/convex hulls in a local sense by constructing them using k-nearest neighbors of a test sample for classification problems with complex nonlinear decision boundaries. They report that affine hulls usually give higher classification accuracy than convex hulls and that using both models for classification significantly improves the k-nearest neighbor classification performance [26]. We extended local linear affine/convex hull classifiers to the nonlinear case in [4]. More recently, we compared different convex class models for high-dimensional classification problems, then found that affine hull approximations are typically more accurate than convex hull approximations [3]. These results are not surprising due to the fact that high-dimensional approximations tend to be simple: For a fixed sample size, the amount of geometric details that can be resolved usually decreases rapidly as the dimensionality in-

creases. Therefore, affine hulls tend to be better models for high-dimensional data approximations.

Besides the classification, approximations based on affine hulls have also been used for dimensionality reduction. Mixtures of Principal Component Analyzers [12] use local affine hulls to estimate nonlinear data manifolds. Similarly, Locally Linear Embedding [18] approximates the nonlinear structure of the high-dimensional data by exploiting local affine hull reconstructions. Verbeek [25] combined several locally valid affine hulls to obtain a global nonlinear mapping between the high-dimensional sample space and low-dimensional manifold. Many applications of affine hulls in the context of classification and dimensionality reduction can be attributed in part to their simplicity and computational efficiency. Finding distances from test samples to affine hulls requires only simple linear algebra. On the other hand, computing distances to nonlinear complex models can be problematic. Even if the models are restricted to being convex hulls, distance computations require the solution of a quadratic optimization problem.

The classification methods using affine hulls or other convex sets described above are "nearest convex model" classifiers and they are instance-based in nature. In other words, decision boundaries are not explicitly created during a training phase. Instead, the decision boundaries remain implicit, and new examples are classified online based on the distances to the nearest convex class models. This paper investigates an alternative "margin between convex model" strategy that is based on explicitly building maximum margin separators between pairs of affine hulls. As a first example of the power of this approach, note that the SVM itself is the maximum margin separator between the convex hulls of the training samples of the two classes. One motivation for replacing nearest-convex-

model approaches with margin-based ones is that for all of the above cited nearest-convex-model classifiers, the 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 generalization 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, whereas they do typically change piece-wise quadratic decision surface of the equivalent nearest convex hull classifier. Another motivation for studying margin-between-affine hulls approach is their potential flexibility and compactness. In linear case, affine 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 simple linear algebra.

In our preliminary work [5] we showed how to construct maximum margin classifier that separates linear affine hulls. Another study addressing the same problem was independently given in [29]. Here we extend the method such that it can be used when the class samples lie on nonlinear manifolds that cannot be modeled with linear affine hulls. To this end, we map the samples in each class into a much higher-dimensional feature space through kernel mapping and then construct the linear affine hulls in the mapped space. Since the problem is cast in a much higher-dimensional space, it is more likely that class regions can now be approximated with linear affine hulls. Although the constructed linear affine

hulls in the mapped space corresponds to nonlinear manifolds in the input space, finding the maximum separating hyperplane between these nonlinear manifolds is still straightforward because of their linear nature in the mapped space. In case of outliers, to allow soft margin solutions, we first reduce affine hulls in order to alleviate the effects of those outliers and then search for the best separating hyperplane between these reduced robust models.

The rest of the paper is organized as follows: In Section II, we introduce the proposed method. Section III describes the experimental results. Concluding remarks are given in Section IV.

2. Method

Consider a binary classification problem with the training data given in the form $\{\mathbf{x}_i, y_i\}$, $i = 1, ..., n$, $y_i \in \{-1, +1\}$, $\mathbf{x}_i \in \mathbb{R}^d$. To separate classes, SVM classifier finds a separating hyperplane that maximizes the margin, which is defined as the distance between the hyperplane and closest samples from the classes. To do so, SVM first approximates each class with a convex hull [1]. A convex hull consists of all points that can be written as a convex combination of the points in the original set, and a convex combination of points is a linear combination of data points where all coefficients are nonnegative and sum up to 1. More formally, the convex hull of samples $\{\mathbf{x}_i\}_{i=1,...,n}$ can be written as

$$H^{convex} = \left\{ \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i | \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0 \right\}. \tag{1}$$

Convex hulls of two classes are illustrated in Fig. 1. Following this approximation, SVM finds the closest points in these convex hulls. Then, these two points are

connected with a line segment. The plane, orthogonal to the line segment that bisects the line, is selected to be the separating hyperplane as shown in Fig. 1.

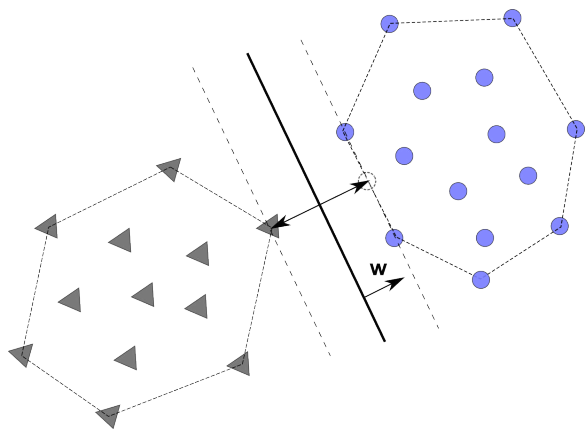

Figure 1: Two closest points on the convex hulls determine the separating hyperplane.

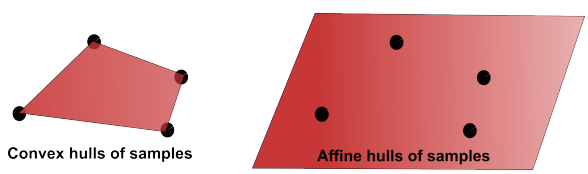

Figure 2: Comparison of convex and affine hulls of samples.

In contrast to the SVM classifier, the proposed method approximates each class (positive and negative classes) with an affine hull of its training samples. An affine hull of a class is the smallest affine subspace containing them. This is an unbounded, and hence typically rather loose model for each class, thus affine hull modeling can be a better choice than convex hull modeling for high-dimensional data. Affine and convex hulls of four samples are illustrated in Fig. 2. 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 $\sum_{i=1}^{n} \alpha_i = 1$. More formally affine hull of a class with samples $\{\mathbf{x}_i\}_{i=1,\ldots,n}$ can be written as

$$H^{aff} = \left\{ \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \mid \sum_{i=1}^{n} \alpha_i = 1 \right\}. \tag{2}$$

Our goal is to find the maximum margin linear separating hyperplane between affine hulls of classes. The points $\mathbf{x}$ which lie on the separating hyperplane satisfy $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$, where $\mathbf{w}$ is the normal of the separating hyperplane, $|b|/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin, and $||\mathbf{w}||$ is the Euclidean norm of $\mathbf{w}$. For any separating hyperplane, all points $\mathbf{x}_i$ in the positive class satisfy $\langle \mathbf{w}, \mathbf{x}_i \rangle + b > 0$ and all points $\mathbf{x}_i$ in the negative class satisfy $\langle \mathbf{w}, \mathbf{x}_i \rangle + b < 0$ so that $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ for all training data points. Finding the best separating hyperplane between affine hulls can be solved by computing the closest points on them. The optimal separating hyperplane will be the one that bisects perpendicularly the line segment connecting the closest points as in SVM classifier. The offset (also called threshold), b, can be chosen as the distance from the origin to the point halfway between the closest points along the normal $\mathbf{w}$. Once the best separating hyperplane is determined, a new sample $\mathbf{x}$ is classified based on the sign of the decision function, $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.

Next, we will first show how to find the best separating hyperlane for linearly separable affine hulls and then extend the idea for inseparable case. After that, we explain kernelization process. This is followed by introducing a second equivalent formulation based on a variation of ν-SVM [20]. Lastly we show the relation between the proposed method and the Least Squares SVM (LS-SVM) [24, 23] and derive a geometric intuition for LS-SVM.

2.1. Linearly Separable Case

Suppose that affine hulls belonging to the positive and negative classes are linearly separable. The affine hulls of two classes do not intersect, i.e., they are linearly separable, if the affine combinations of their samples satisfy the rule

$$\sum_{i:y_i=+1} \alpha_i \mathbf{x}_i \neq \sum_{j:y_j=-1} \alpha_j \mathbf{x}_j \quad \text{for} \quad \sum_{i:y_i=+1} \alpha_i = \sum_{j:y_j=-1} \alpha_j = 1. \tag{3}$$

It should be noted that linear separability of data points does not necessarily guarantee the separability of corresponding affine hulls of classes. For linearly separable case, it is more convenient to write an affine hull as

$$\{\mathbf{x} = \mathbf{U}\mathbf{v} + \mu \,|\, \mathbf{v} \in \mathbb{R}^l\}, \tag{4}$$

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 $[\mathbf{x}_1 - \boldsymbol{\mu}, ..., \mathbf{x}_n - \boldsymbol{\mu}]$. 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. This subspace estimation process is essentially orthogonal least squares fitting. Discarding near-zero singular values corresponds to discarding directions that appear to be predominantly "noise". As an alternative, samples can be fitted with some other more robust subspace estimation processes such as L1 norm based subspace fitting procedures described in [10, 13]. But, we will consider only the least squares fitting (L2 norm) in this study.

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). A closest pair of points between the two hulls can be found by solving

$$\min_{\mathbf{v}_+,\mathbf{v}_-} \; ||(\mathbf{U}_+\mathbf{v}_+ + \boldsymbol{\mu}_+) - (\mathbf{U}_-\mathbf{v}_- + \boldsymbol{\mu}_-)||^2. \tag{5}$$

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

$$\min_{\mathbf{v}} \; ||\mathbf{U}\,\mathbf{v} - (\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)||^2. \tag{6}$$

If we take the derivative of the objective function (6) with respect to $\mathbf{v}$ and equate it to zero, then we obtain

$$\mathbf{U}^\top\mathbf{U}\mathbf{v} - \mathbf{U}^\top(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+) \;=\; \mathbf{0} \tag{7}$$

Subsequently, we get the solution of the problem as $\mathbf{v} = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)$. Taking the decision boundary $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle + b$,

$$\mathbf{w} \;=\; \frac{1}{2}(\mathbf{x}_+ - \mathbf{x}_-) \;=\; \frac{1}{2}(\mathbf{I} - \mathbf{P})(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) \tag{8}$$

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, $\mathbf{I} - \mathbf{P}$ is the corresponding projection onto the orthogonal complement of this span[1], and $\mathbf{x}_+$ and $\mathbf{x}_-$ denote the closest points on the positive and negative classes, respectively. Note that $\mathbf{w}$ lies along the line segment joining the two closest points and it is half the line segment's size. The offset b of the separating hyperplane is given by

$$b \;=\; -\mathbf{w}^\top(\mathbf{x}_+ + \mathbf{x}_-)/2. \tag{9}$$

[1]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 $\mathbf{P} = \tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}$.

2.2. *Inseparable Case*

A problem arises if the affine hulls of classes intersect, i.e., affine hulls are not linearly separable. If the affine hulls of classes are close to being linearly separable and they overlap because of a few outliers, we can restrict the influence of outlying points by reducing affine hulls. Note that ignoring directions corresponding to the overly small singular values during affine hulls constructions reduces the effects of noise and outliers up to some degree. But, we will use a different approach here in order to cope with the outliers. To this end, we use the initial affine hull formulation (2) and introduce upper and lower bounds on coefficients α_i to reduce affine hulls inspired by the idea that is introduced to reduce convex hulls in [1]. It should be noted that the reduced affine hulls are not uniformly scaled versions of the initial complete affine hulls. One may go further and choose different lower and uper bounds, or define a different interval for every sample in the training set if a-priori information is available. For instance, if the lower bound is set to zero, then the method will be equivalent to the SVM classifier. Finding the closest points on the reduced affine hulls can be written as a quadratic optimization problem

$$\begin{aligned} &\min_{\alpha} \; || \sum_{i:y_i=+1} \alpha_i \mathbf{x}_i - \sum_{i:y_i=-1} \alpha_i \mathbf{x}_i ||^2 \\ &s.t. \quad \sum_{i:y_i=+1} \alpha_i = 1, \quad \sum_{i:y_i=-1} \alpha_i = 1, \;\; -\tau \leq \alpha_i \leq \tau, \end{aligned} \tag{10}$$

where τ is the user-chosen bound. This optimization problem (10) can be written in a more compact form as

$$\begin{aligned} &\min_{\alpha} \; \sum_{ij} \alpha_i \alpha_j y_i y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \\ &s.t. \; \sum_i \alpha_i y_i = 0, \; \sum_i \alpha_i = 2, \;\; -\tau \leq \alpha_i \leq \tau. \end{aligned} \tag{11}$$

This is a quadratic programming problem that can be solved using standard optimization techniques. Note that the Hessian matrix, $\mathbf{G} = [G_{ij}] = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, is a positive semi-definite matrix, thus the objective function is convex and a global minimum exists as in SVM classifier. Moreover, if the Hessian matrix is strictly positive definite, the solution is unique and it is guaranteed to be the global minimum.

Since the coefficients are bounded between $-\tau$ and $+\tau$, the solution is determined by more points and no extreme point or noisy point can excessively influence the solution for well-chosen τ. Once we compute the optimal values of coefficients α_i, the normal and the offset of the separating hyperplane can be computed as in the linearly separable case

$$\mathbf{w} = \frac{1}{2}\Big(\sum_{i:y_i=+1} \alpha_i \mathbf{x}_i - \sum_{i:y_i=-1} \alpha_i \mathbf{x}_i\Big), \tag{12}$$

$$b = -\frac{1}{2}\mathbf{w}^\top\Big(\sum_{i:y_i=+1} \alpha_i \mathbf{x}_i + \sum_{i:y_i=-1} \alpha_i \mathbf{x}_i\Big), \tag{13}$$

We call this method Large Margin Classifier of Affine Hulls (LMC-AH) since it uses affine hulls to approximate class regions and finds the optimal separating hyperplane yielding the largest margin between the affine hulls.

If the underlying geometry of the classes is highly complex and nonlinear, and approximating classes with linear affine hulls is not appropriate, we can map the data into a higher-dimensional space where the classes can be approximated with linear affine hulls. Note that the objective function of (11) is written in terms of the dot products of samples, which allows the use of the kernel trick. Thus, by using kernel trick, – i.e., replacing $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ where $\phi : \mathbb{R}^d \rightarrow \Im$ is the mapping function from the input space to the mapped space $\Im$ – we can find the best seperating hyperplane parameters

in the mapped space. As a result, more complex nonlinear decision boundaries between classes can be approximated by using this trick.

2.3. An Equivalent Formulation Based on Variation of ν-SVM Classifier

The ν-SVM formulation has been proposed as an alternative to the classical SVM formulation [20]. A new parameter ρ' is introduced in this formulation, and error penalty term C that appears in classical SVM formulation is removed. Here, we introduce an alternative formulation to find the best separating hyperplane between affine hulls based on a variation of ν-SVM formulation. A major advantage of new formulation is that one can relate the parameter τ in (11) with the expected error bounds and this may help us to find a more sophisticated procedure for choosing unknown parameters that appear in both formulations.

The ν-SVM optimization is formulated as

$$\begin{aligned} \min_{\mathbf{w}',b',\xi',\rho'} \ & \frac{1}{2}||\mathbf{w}'||^2 - \nu\rho' + \frac{1}{n}\sum_i \xi'_i \\ s.t. \ & y_i(\langle \mathbf{w}', \mathbf{x}_i\rangle + b') \geq \rho' - \xi'_i, \ \ \xi'_i \geq 0, \ \ \rho' \geq 0, \end{aligned} \tag{14}$$

where $\mathbf{w}'$ represents the normal of the separating hyperplane, b' is the offset, ν is a user-chosen parameter between 0 and 1, and ξ'_i, $i = 1, ..., n$, are the positive slack variables. In this formulation, for linearly separable case, there exist two parallel supporting hyperplanes positioned such that all points in the positive class satisfy $\langle \mathbf{w}', \mathbf{x}\rangle + b' \geq \rho'$ and all points in the negative class satisfy $\langle \mathbf{w}', \mathbf{x}\rangle + b' \leq -\rho'$ as shown in Fig. 3. Therefore, classes are separated by the margin $2\rho'/||\mathbf{w}'||$ and it is shown that ν acts as an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors [20]. Moreover, the decision function produced by ν-SVM can also be produced by classical SVM for appropriate choice of error penalty term C.

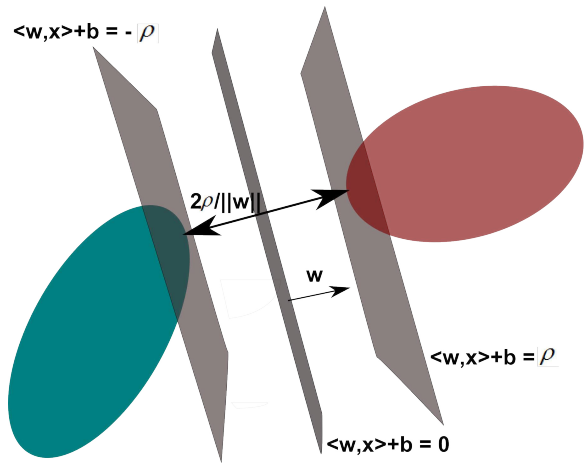

Figure 3: Illustration of the supporting and the best separating hyperplanes in linearly separable case for ν-SVM classifier.

The ν-SVM formulation can be interpreted as a maximal separation between the reduced convex hulls of classes [9]. Since we use affine hulls to model classes, we need to revise the optimization problem to accommodate this change. To this end, we first divide the objective function by $\nu^2/2$, the constraints by ν, and make the following substitutions as in [9]

$$\tau = \frac{2}{\nu n}, \;\; \mathbf{w} = \frac{\mathbf{w}'}{\nu}, \;\; b = \frac{b'}{\nu}, \;\; \rho = \frac{\rho'}{\nu}, \;\; \xi_i = \frac{\xi_i'}{\nu} \tag{15}$$

These modifications yield the equivalent formulation[2]

$$\begin{aligned} \min_{\mathbf{w},b,\xi,\rho} \;\; & ||\mathbf{w}||^2 - 2\rho + \tau \sum_i \xi_i \\ s.t. \;\; & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i, \;\; \xi_i \geq 0, \end{aligned} \tag{16}$$

with the new decision function $f(\mathbf{x}) \equiv f'(\mathbf{x})/\nu$.

[2]Crisp and Burges [9] showed that the constraint $\rho' \geq 0$ in (14) is redundant and hence it can be removed.

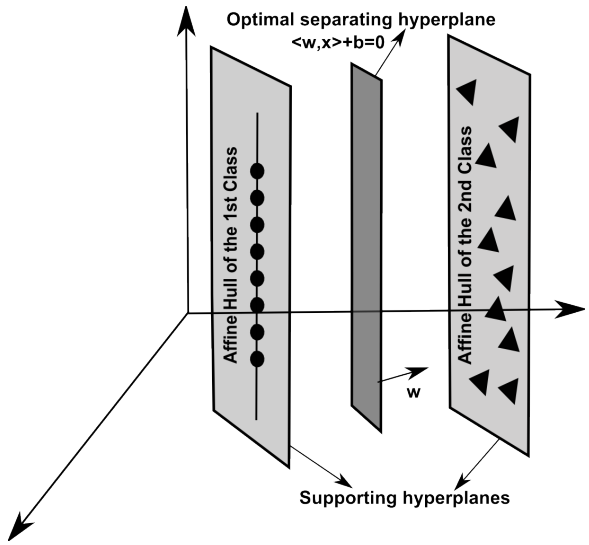

Figure 4: Optimal separating hyperplane between affine hulls of two classes. Note that affine hulls lie on the supporting hyperplanes.

Note that two affine hulls are linearly separable if they lie parallel to each other in the given input space since affine hulls extend to infinity in all directions. In this case, the supporting hyperplanes yielding the largest margin between affine hulls will entirely include them so that all affine combinations of samples belonging to the positive class satisfy $\left\langle \mathbf{w}, \mathbf{x}_{+}^{aff} \right\rangle + b = \rho$ and all affine combinations of samples belonging to the negative class satisfy $\left\langle \mathbf{w}, \mathbf{x}_{-}^{aff} \right\rangle + b = -\rho$ as illustrated in Fig. 4. Fig. 4 illustrates affine hulls of two classes where the affine hull of the first class is a line and the affine hull of the second class is a plane. Note that affine hulls are linearly separable if they lie parallel to each other. Therefore, all samples of classes and their affine combinations lie on the supporting hyperplanes, which yield the largest margin between the affine hulls. In case of outliers, we must construct reduced compact affine hulls that will fit the data robustly. Therefore, we

should allow errors for outlier samples from all over the input space, not just the ones near the decision boundary as illustrated in Fig. 5. To do so, the inequality constraints in (16) is replaced with equality constraints $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) = \rho - \delta_i \xi_i$, where δ_i is a term which takes values +1 or -1 based on the location of outliers with respect to the supporting hyperplanes. This leads to the new optimization problem

$$\begin{aligned} &\min_{\mathbf{w}, b, \xi, \rho} \; ||\mathbf{w}||^2 - 2\rho + \tau \sum_i \xi_i \\ &s.t. \;\; y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) = \rho - \delta_i \xi_i, \;\; \xi_i \geq 0. \end{aligned} \tag{17}$$

To derive the dual, we consider the Lagrangian

$$L(\mathbf{w}, b, \xi, \rho, \alpha, \beta) = ||\mathbf{w}||^2 - 2\rho + \tau \sum_i \xi_i - \sum_i \alpha_i \left[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - \rho + \delta_i \xi_i \right] - \sum_i \beta_i \xi_i, \tag{18}$$

where $\beta_i \geq 0$. The Lagrangian L has to be maximized with respect to α_i, β_i and minimized with respect to $\mathbf{w}, b, \xi$, and ρ. The optimality conditions yield

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}} = \mathbf{0} \rightarrow \mathbf{w} = \frac{1}{2} \sum_i \alpha_i y_i \mathbf{x}_i, \\ &\frac{\partial L}{\partial b} = 0 \rightarrow \sum_i \alpha_i y_i = 0, \\ &\frac{\partial L}{\partial \rho} = 0 \rightarrow \sum_i \alpha_i = 2, \\ &\frac{\partial L}{\partial \xi_i} = 0 \rightarrow \alpha_i = \frac{\tau - \beta_i}{\delta_i} \rightarrow -\tau \leq \alpha_i \leq \tau. \end{aligned} \tag{19}$$

Thus, the dual of the optimization problem becomes

$$\begin{aligned} &\min_{\alpha} \; \frac{1}{4} \sum_{ij} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &s.t. \;\; \sum_i \alpha_i y_i = 0, \;\; \sum_i \alpha_i = 2, \;\; -\tau \leq \alpha_i \leq \tau. \end{aligned} \tag{20}$$

This optimization is equivalent to the one given in (11) - here $1/4$ appears in the objective function, but rescaling objective function with a positive constant does not change the solution. Therefore, the new formulation based on modified ν-SVM is in fact equivalent to finding the best separating hyperplane between the reduced affine hulls that represent classes. We call this method ν-LMC-AH. Due to the Karush-Kuhn-Tucker (KKT) conditions, slack variables can occur only

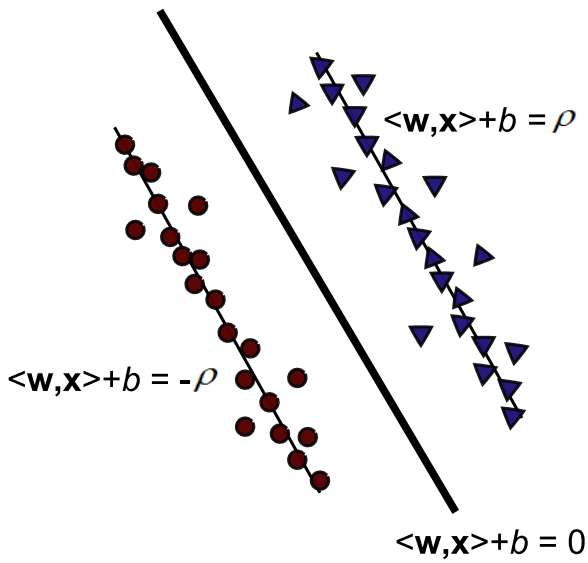

Figure 5: To obtain better separating hyperplanes between affine hulls, we should allow errors for outlier samples from all over the input space.

when $\alpha_i = \pm\tau$. To compute offset b, we use the primal constraints and take equal number of samples with coefficients $\alpha_i \neq \pm\tau$ from positive and negative classes. Assume that there are l selected samples. By using KKT conditions, we know that $\xi_i = 0$ for the samples with $\alpha_i \neq \pm\tau$. Thus, the offset will be

$$b \;=\; -\frac{1}{2l}\sum_{i=1}^{l}\sum_{j=1}^{n}\alpha_j y_j \left\langle \mathbf{x}_j, \mathbf{x}_i \right\rangle. \tag{21}$$

This offset is not necessarily equivalent to the one given in (13). Therefore, using geometrically inspired formulation and ν-LMC-AH formulation create separating hyperplanes with the same normal, but the positions (perpendicular distances from the origin) of these hyperplanes may be different. It is not a-priori evident that which offset is the best and one can use other principled methods to determine the best b for a given problem, e.g., given $\mathbf{w}$, b can be computed as value yielding the smallest classification error on a validation set. As in the previous case, extension to the nonlinear case can be done by using the kernel trick.

2.4. Geometric Interpretation of Least Squares SVM Classifier

Least Squares Support Vector Machines (LS-SVM) was initially proposed by Suykens and Vandewalle [24] for classification and nonlinear function estimation and then new variants of this method have been introduced [7, 21, 22]. The basic motivation was to simplify the classical SVM formulation without losing much generalization performance. To this end, inequality constraints in the SVM classification formulation are heuristically replaced with equality constraints[3], and square of the slack variables are used in the objective function. More formally the optimization problem is defined as

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \ \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{2}\sum_i \xi_i^2 \\ &s.t. \ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) = 1 - \xi_i, \end{aligned} \tag{22}$$

where C is a user-chosen error penalty term as in classical SVM classifier. It is shown that the solution is obtained by solving a set of linear equations rather than

[3]In fact, using equality constraints for nonlinear function estimation was introduced earlier in [19].

solving a quadratic programming problem [24]. Note that, in this formulation, for the linearly separable case, there exist two parallel supporting hyperplanes positioned such that all points in the positive class satisfy $\langle \mathbf{w}, \mathbf{x} \rangle + b = 1$ and all points in the negative class satisfy $\langle \mathbf{w}, \mathbf{x} \rangle + b = -1$ where the margin between these hyperplanes is given by $2/||\mathbf{w}||$. As in ν-LMC-AH formulation, this corresponds to approximating each class with an affine hull instead of a convex hull since all samples and their affine combinations are forced to lie on the supporting hyperplanes. In LS-SVM, L2 norm (squares) of the slack variables are used in the optimization, but using L1 norm of the slack variables in the objective function may be more appropriate for some applications since it allows more robust fitting of data samples. If we use L1 norm of the slack variables, new optimization problem becomes

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \\ &s.t. \ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) = 1 - \delta_i \xi_i, \ \ \xi_i \geq 0, \end{aligned} \tag{23}$$

where δ_i is a term which takes values +1 or -1. The Lagrangian will be

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \delta_i \xi_i \right] - \sum_i \beta_i \xi_i, \tag{24}$$

under the constraint $\beta_i \geq 0$. The optimality conditions yield

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}} = \mathbf{0} \rightarrow \mathbf{w} = \frac{1}{2}\sum_i \alpha_i y_i \mathbf{x}_i, \\ &\frac{\partial L}{\partial b} = 0 \rightarrow \sum_i \alpha_i y_i = 0, \\ &\frac{\partial L}{\partial \xi_i} = 0 \rightarrow \alpha_i = \frac{C - \beta_i}{\delta_i} \rightarrow -C \leq \alpha_i \leq C. \end{aligned} \tag{25}$$

Thus, the dual of the optimization problem becomes

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2}\sum_{ij}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_i \alpha_i \\ s.t. \ & \sum_i \alpha_i y_i = 0, \ -C \leq \alpha_i \leq C. \end{aligned} \tag{26}$$

Similar to the previous cases, this is a convex quadratic optimization problem with a global minimum. Due to the Karush-Kuhn-Tucker (KKT) conditions, slack variables can occur only when $\alpha_i = \pm C$. To compute offset b, we use the primal constraints and take equal number of samples with coefficients $\alpha_i \neq \pm C$ from positive and negative classes as in ν-LMC-AH. Assume that there are l selected samples. By using KKT conditions, we know that $\xi_i = 0$ for the samples with $\alpha_i \neq \pm C$. Thus, the offset will be

$$b \ = \ -\frac{1}{2l}\sum_{i=1}^{l}\sum_{j=1}^{n}\alpha_j y_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle. \tag{27}$$

A new sample is classified based on the sign of the decision function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$. Nonlinearization can be done by replacing the dot products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. We call this method as C-LMC-AH since it uses error penalty term C. It follows from Proposition 6 of [20] that for appropriate choices of C, the C-LMC-AH algorithm will yield identical results to LMC-AH and ν-LMC-AH classifiers. More precisely, if ν-LMC-AH classification leads to $\rho \geq 0$, then C-LMC-AH method with C set a priori to $1/\rho' n$ (or $1/\rho\nu n$), leads to the same decision function as ν-LMC-AH [6, 20].

2.5. Extension to the Multi-Class Classification Problems

To use the proposed methods in multi-class classification problems, we can use most of the strategies adopted for extending binary SVM classifiers to the multi-class cases. Here we will discuss the most popular two strategies: one-against-one

(OAO) and one-against-rest (OAR). For a c-class classification problem, the OAR strategy trains c binary classifiers, in which each classifier separates one class from the remaining $c-1$ classes. All classifiers are needed to be trained on the entire training set, and the class label of a test sample is determined according to the highest output of the classifiers in the ensemble. On the other hand, the OAO strategy constructs all possible $c(c-1)/2$ binary classifiers out of c classes. The decision of the ensemble is decided by max wins algorithm: Each OAO classifier casts one vote for its preferred class, and the final decision is the class with the most votes. In addition to these we can also use Directed Acyclic Graphs [17] or Binary Decision Trees [28] for multi-class classification.

3. Experiments

We tested[4] the linear and kernelized versions of the proposed methods LMC-AH, ν-LMC-AH and C-LMC-AH (L1 norm based LS-SVM) on a number of data sets and compared them to the SVM classifier. For the linearly separable case, linear separator is determined by using affine subspace estimation formulation, and subspace dimensions are set by retaining enough leading eigenvectors to account for 95-98% of the total energy in the eigen-decomposition. For the inseparable and nonlinear cases, we used quadratic programming formulations. Both one-against-rest (OAR) and one-against-one (OAO) approaches are used for multi-class classification problems and we report the results of whichever yields the best.

We first tested the linear LMC-AH method on multiple and single shot face

[4]For software see *http://www2.ogu.edu.tr/∼mlcv/softwares.html*.

recognition problems to demonstrate that affine hull approximations are more appropriate than convex hull approximations when the dimensionality of the input space is high. To assess the generalization performances of kernelized versions of the methods, we tested them on seven low-dimensional databases chosen from the UCI repository.

3.1. Experiments on the Honda/UCSD Database

Honda/UCSD database [15] has been collected for video-based face recognition and it consists of 59 video sequences belonging to 20 individuals. Each video consists of approximately 300-500 frames. It is a fixed database so that 20 of the videos are allocated for training and the remaining 39 for testing. Here, we consider face recognition based on multiple images. In this scenario, face recognition problem is defined as taking a set of face images from an unknown person and finding the most similar set among the database of labeled image sets. We used

Figure 6: Some detected face images from videos belonging to two subjects.

the cascade face detector of Viola and Jones [27] to detect faces in each video sequence and resized the detected face images to the gray images of size 40×40

Table 1: Classification Rates (%) on the Honda/UCSD Database.

Methods	Clean	Corrup. Training	Corrupted Test	Corrup. Training+Test
LMC-AH	**97.44**	**97.44**	**92.31**	**87.18**
SVM	94.87	92.31	**92.31**	82.05

followed by histogram equalization. Then, these images are used to construct image sets of individuals. Some of the detected face images are shown in Fig. 6. We used affine hulls and convex hulls to model image sets, and used the distances between these models as a similarity measure. In other words, computed margins between linear affine hulls and convex hulls are used to determine the label of the tested image sets[5]. We performed several experiments in order to test the robustness against outliers. In the first experiment, we computed classification rates based on the clean image sets. Then, we systematically corrupted training and test sets by adding images from other classes to each set. These images can be seen as outliers and the changes in classification rates reflect the robustness of the methods against these outliers. The results are given in Table 3.1. As can be seen from the table, affine hull approximations yield better results than convex hull approximations in all cases except for the corrupted test where both models give same results. Thus, affine hulls seem better and more robust models for approximating image sets.

[5]For the affine hull case we simply used the minimum distance between the estimated affine subspaces using equation (6) whereas soft margin linear SVM algorithm is used to determine the distances between convex hulls.

3.2. *Experiments on the AR Face Database*

The AR Face data set [16] 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 centers of the two eyes fell at fixed coordinates, then cropped to size 105×78. Some pre-processed images are shown in Fig. 7. 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 rate being obtained by averaging the 10 results. The results are shown in Table 3.2. Best results are obtained by OAR strategy for both tested methods. The proposed method gives better classification rates than soft-margin linear SVM classifier in all cases. The performance difference is more apparent for $n = 7$. These results support our claims, suggesting that affine hulls can be better models for representing classes in high-dimensional spaces when the number of samples is limited.

Figure 7: Aligned images of one subject from the AR Face database.

Table 2: Classification Rates (%) on the AR Face Database.

Methods	$n = 7$	$n = 13$	$n = 20$
LMC-AH	**95.19**$\pm$0.6	**98.95**$\pm$0.3	**99.62**$\pm$0.3
SVM	94.54$\pm$0.6	98.66$\pm$0.2	99.58$\pm$0.3

3.3. Experiments on the UCI Databases

In this group of experiments, we tested the kernelized versions of the methods (quadratic programming formulations) on seven lower-dimensional datasets from the UCI repository: Ionosphere, Iris, Letter Recognition (LR), Multiple Features (MF) - pixel averages, Pima Indian Diabetes (PID), Wine, and Wisconsin Diagnostic Breast Cancer (WDBC). The key parameters of these datasets are summarized in Table 3.3. We used the Gaussian kernels, and all design parameters are set based on random partitions of datasets into training and test sets. OAO strategy was used for multi-class problems. Reported classification rates given in Table 3.3 are computed by 5-fold cross-validation. Although being quite mixed, results indicate that generalization performances of LMC-AH and ν-LMC-AH methods compare favorably with SVM classifier whereas C-LMC-AH generally yields the worst classification accuracy.

4. Summary and Conclusion

We investigated the idea of basing large margin classifiers on affine hulls of classes as an alternative to the SVM (convex hull large margin classifier). Given two affine hull models, their corresponding large margin classifier is easily determined by finding a closest pair of points on these two models and bisecting the

Table 3: Low-Dimensional Databases Selected from UCI Repository

Databases	Number of Classes	Data Set Size	Dimensionality
Ionosphere	2	351	34
Iris	3	150	4
LR	26	20000	16
MF	10	2000	256
PID	2	768	8
Wine	3	178	13
WDBC	2	569	30

Table 4: Classification Rates (%) on the UCI Datasets.

UCI	LMC-AH	ν-LMC-AH	C-LMC-AH	SVM
Ionosphere	**93.7**$\pm$2.9	**93.7**$\pm$2.9	93.4$\pm$2.3	92.9$\pm$3.2
Iris	94.7$\pm$2.9	94.7$\pm$2.9	**95.3**$\pm$3.8	**95.3**$\pm$3.8
LR	**99.98**$\pm$0.02	**99.98**$\pm$0.02	99.89$\pm$0.13	99.64$\pm$0.12
MF	**98.4**$\pm$0.4	**98.4**$\pm$0.4	97.8$\pm$0.3	98.0$\pm$0.4
PID	**99.9**$\pm$0.3	**99.9**$\pm$0.3	**99.9**$\pm$0.3	**99.9**$\pm$0.3
Wine	**98.8**$\pm$1.6	**98.8**$\pm$1.6	94.8$\pm$2.6	98.2$\pm$1.6
WDBC	96.0$\pm$2.5	96.0$\pm$2.5	94.9$\pm$3.0	**97.6**$\pm$0.7

displacement between them. We also investigated another formulation obtained by revising the ν-SVM classifier. This formulation yields a separating hyperplane with the same normal as in our first formulation, but the offset is not necessarily the same. This suggests that for a fixed hyperplane normal $\mathbf{w}$ in a specific problem, there may be principled procedures to determine the best offset b. To allow soft margin solutions, we first reduce affine hulls to alleviate the effects of outliers and then find the best separating hyperplanes between these reduced models. Such classifiers can also be kernelized, and extension to the multi-class classification is straightforward using any of the standard approaches such as OAO or OAR.

The experimental results provided useful insights on the potential application areas of the proposed method. The proposed method is much more efficient than SVM classifier in terms of classification accuracy and real-time performance (testing time) when the dimensionality of the sample space is high and affine hulls are linearly separable (in this case solution is easily determined based on subspace estimation which requires simple linear algebra whereas SVM formulation requires solving a quadratic programming). For the low-dimensional databases generalization performances of the proposed methods compare favorably with SVM classifier but SVM is more efficient in terms of testing time. This is because of the fact that all training data points contribute to the affine hull models (almost all computed α_i coefficients are nonzero), thus the proposed quadratic optimization solutions lack sparseness, and we need more computations to evaluate decision functions. Nevertheless, some pruning techniques can be employed to overcome this problem.

Acknowledgment

This work was supported by the Young Scientists Award Programme (TÜBA-GEBİP/2010) of the Turkish Academy of Sciences.

References

[1] K. P. Bennett and E. J. Bredensteiner, 2000. Duality and geometry in SVM classifiers, International Conference on Machine Learning.

[2] C. J. C. Burges, 1998. tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, vol. 2, pp. 121-167.

[3] H. Cevikalp and B. Triggs and R. Polikar, 2008. Nearest hyperdisk methods for high-dimensional classification, International Conference on Machine Learning.

[4] H. Cevikalp and D. Larlus and M. Neamtu and B. Triggs and F. Jurie, 2008. Manifold based local classifiers: linear and nonlinear approaches, Journal of Signal Processing Systems (published online).

[5] H. Cevikalp and B. Triggs, 2009. Large Margin Classifiers Based on Convex Class Models, International Conference on Computer Vision Workshops.

[6] C.C. Chang and C.J. Lin, 2001. Training ν-support vector classifiers: Theory and algorithms, Neural Computation, vol. 13, no 9, pp. 2119-2147.

[7] W. Chu and C. J. Ong and S. Keerthi, 2005. An improved conjugate gradient scheme to the solution of least squares SVM, IEEE Transactions on Neural Networks, vol. 16, pp. 498-501.

[8] C. Cortes and V. Vapnik, 1995. Support vector networks, Machine Learning, vol.20, pp. 273-297.

[9] D. J. Crisp and C. J. Burges, 1999. A geometric interpretation of ν-SVM classifiers, Neural Information Processing Systems.

[10] C. Ding and D. Zhou and X. He and H. Zha, 2006. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization, International Conference on Machine Learning.

[11] M. B. Gulmezoglu and V. Dzhafarov and A. Barkana, 2001. The common vector approach and its relation to principal component analysis, IEEE Trans. Speech Audio Proc., vol. 9, pp. 655-662.

[12] G. E. Hinton, P. Dayan, and M. Revow, 1997. Modeling the manifolds of images of handwritten digits, IEEE Trans. on Neural Networks, vol. 18, pp. 65-74.

[13] Q. Ke and T. Kanade, 2005. Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming, IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[14] J. Laaksonen, 1997. Subspace classifiers in recognition of handwritten digits, Technical Report.

[15] K. C. Lee and J. Mo and M. H. Yang and D. Kriegman, 2003. Video-based face recognition using probabilistic appearance manifolds, IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[16] A. M. Martinez and R. Benavente, 1998. The AR Face Database, Technical Report, Computer Vision Center, Barcelona, Spain.

[17] J. C. Platt and N. Cristianini and J. Shawe-taylor, 2000. Large margin dags for multiclass classification, Advances in Neural Information Processing Systems.

[18] S. T. Roweis and L. K. Saul, 2000. Nonlinear dimensionality reduction by locally linear embedding, Science, vol. 290, pp. 2323-2326.

[19] C. Saunders and A. Gammerman and V. Vovk, 1998. Ridge regression learning algorithm in dual variables, International Conference on Machine Learning.

[20] B. Schölkopf and A. J. Smola and R. C. Williamson, and P. L. Bartlett, 2000. New support vector algorithms, Neural Computation, vol. 12, pp. 1207-1245.

[21] J. A. K. Suykens and L. Lukas and J. Vandewalle, 2000. Sparse least squares support vector machine classifiers, in Proc. of the European Symposium on Artificial Neural Networks (ESANN'2000), Bruges, Belgium, pp. 37-42.

[22] J. A. K. Suykens and J. De Brabanter and L. Lukas and J. Vandewalle, 2002. Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing,vol. 48, no. 1-4, pp. 85-105.

[23] J. A. K. Suykens and T. Van Gestel and J. De Brabanter and B. De Moor and J. Vandewalle, 2002. Least Squares Support Vector Machines, World Scientific Publishing Co. Pte. Ltd.

[24] J. A. K. Suykens and J. Vandewalle, 1999. Least squares support vector machine classifiers, Neural Processing Letters, vol. 9, pp. 293-300.

[25] J. Verbeek, 2006. Learning non-linear image manifolds by global alignment of local linear models, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 28, pp. 1236-1250.

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

[27] P. Viola and M. Jones, 2004. Robust real-time face detection, International Journal of Computer Vision, vol. 57, pp. 137-154.

[28] V. Vural and J. G. Dy, 2004. A hierarchical method for multi-class support vector machines, International Conference on Machine Learning.

[29] Z. Xiaofei and S. Yong, 2009. Affine Subspace Nearest Points Classification Algorithm for Wavelet Face Recognition, 2009 WRI World Congress on Computer Science and Information Engineering.