Clustering Based Large Margin Classification: A Scalable Approach using SOCP Formulation

J. Saketha Nath
saketha@csa.iisc.ernet.in

C. Bhattacharyya
chiru@csa.iisc.ernet.in

M. N. Murty
mnm@csa.iisc.ernet.in

Department of Computer Science and Automation,
Indian Institute of Science,
Bangalore, Karnataka, INDIA.

ABSTRACT

This paper presents a novel Second Order Cone Programming (SOCP) formulation for large scale binary classification tasks. Assuming that the class conditional densities are mixture distributions, where each component of the mixture has a spherical covariance, the second order statistics of the components can be estimated efficiently using clustering algorithms like BIRCH. For each cluster, the second order moments are used to derive a second order cone constraint via a Chebyshev-Cantelli inequality. This constraint ensures that any data point in the cluster is classified correctly with a high probability. This leads to a large margin SOCP formulation whose size depends on the number of clusters rather than the number of training data points. Hence, the proposed formulation scales well for large datasets when compared to the sate-of-the-art classifiers, Support Vector Machines (SVMs). Experiments on real world and synthetic datasets show that the proposed algorithm outperforms SVM solvers in terms of training time and achieves similar accuracies.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier Design and Evaluation*

General Terms

Performance

Keywords

Gaussian Mixture Models, BIRCH, Scalability, large margin classification

1. INTRODUCTION

In recent times, there has been an explosive growth in the amount of data that is being collected in the business and scientific arena. As a result, many real-world binary classification applications involve analyzing millions of data points. Intrusion detection, web page classification and spam filtering applications are a few of them. Classification of such large datasets is a challenging task, as they may not fit into memory. Most of the existing classification algorithms are not attractive as they perform multiple passes over data.

Support Vector Machines [15] (SVMs) are one of the most successful classifiers that achieve good generalization in practice. SVMs (soft-margin SVMs) pose the classification problem as a convex quadratic optimization problem of size $m + n + 1$, where m is the number of training data points and n is their dimensionality. The optimization problem has a quadratic objective function and $2m$ linear inequalities. The main contribution of the present paper is to pose the classification problem as a convex optimization problem, whose size is not dependent on the training set. SVMs have emerged as useful tools for classification in practice, primarily because of the availability of efficient algorithms like SMO [13] and chunking [7], which solve the dual of the SVM formulation. However, these algorithms are known to be atleast $O(m^2)$ in running time (see [13, 16]) and hence not scalable to large datasets.

Clustering before computing the classifier is an interesting strategy for large scale problems. CB-SVM [16] is an iterative, hierarchical clustering based SVM algorithm, which handles large datasets. The algorithm thrives on the fact that the SVM optimization solution depends only on a small set of data points called support vectors that lie near the optimal classification boundary. The authors, in their paper, show that the algorithm gives accuracies comparable to SMO with a very small run-time. The proposed classification method also uses clustering to classify data points. However, the method does not proceed in an iterative fashion and does not require hierarchical clustering of the training set. The proposed classifier scales well for very large datasets and gives accuracies comparable to that of SVMs. The class conditional densities are assumed to be modeled using mixture models with spherical covariances. A scalable clustering algorithm is employed to estimate the second order moments of components of the mixture models. Using Chebyshev's inequality and the moments of component densities, the misclassification error on each of the components incurred by the hyperplane classifier is bounded. Introducing slack variables, the bounds can be relaxed to allow for non-separable cases. The relaxation is penalized by minimiz-

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.

ing the sum of the slack variables. Additionally an upper bound on $\|\mathbf{w}\|_2$ is put in order to maximize the generalization of the classifier. The resulting optimization turns out to be a Second Order Cone Programming (SOCP) problem, which can be efficiently solved using fast interior points algorithms [12].

The SOCP problem formulated as above has k linear inequalities and one SOC constraint, where $k = k_1 + k_2$. k_1, k_2 are the number of components in the mixture model of the i^{th} class. Note that the number of inequalities is independent of m, unlike the case of SVMs, where the number of linear inequalities is $2m$. Thus, the size of the optimization problem does not increase with the number of data points. Estimation of the moments of the component distributions is done using an efficient clustering scheme, such as BIRCH [17]. BIRCH, in a single pass over the data constructs a CF-tree (Cluster Feature tree), given a limited amount of resources. CF-tree consists of the sufficient statistics for the hierarchy of clusters in the data. BIRCH also handles outliers effectively as a by-product of clustering.

The remainder of the paper is organized as follows. In section 1.1, a brief review of the classification formulations of SVMs is provided. Main contributions of the paper are presented in section 2. Section 3 presents the experiments on synthetic and real world datasets. Section 4 concludes the paper by discussing some future directions of work.

1.1 Review of SVMs

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$, be the training set. $\mathbf{x}_i$ represents a data point whereas, y_i represents the corresponding class label. The original SVM formulation [15] solves the problem of linear binary classification. It uses $\mathbf{w}^\top \mathbf{x} - b = 0$ as the discriminating hyperplane between the two classes. The idea is to minimize the training set error, by constraining most of the data points to lie on either side of the set of canonical hyperplanes $\mathbf{w}^\top \mathbf{x} - b = \pm 1$, and upper bounding the complexity of the classifier — which in machine learning terms corresponds to achieving good generalization [15]. Bounding complexity, in case of such linear classifiers, gives the constraint $\|\mathbf{w}\|_2 \leq W$, where W is some positive real number. Geometrically this corresponds to having a lower bound on the distance between the set of canonical hyperplanes, called margin ($= \frac{2}{\|\mathbf{w}\|_2}$). Thus SVM formulation can be written as (ξ_j are slack variables):

$$\begin{aligned} \min_{\mathbf{w}, b, \xi_j} \quad & \textstyle\sum_{j=1}^m \xi_j \\ \text{s.t.} \quad & y_j(\mathbf{w}^\top \mathbf{x}_j - b) - 1 + \xi_j \geq 0,\ \xi_j \geq 0\ \forall\ j, \\ & \|\mathbf{w}\|_2 \leq W \end{aligned} \tag{1}$$

The problem (1) is an instance of Second Order Cone Programming problem. An SOCP problem is a convex optimization problem with a linear objective function and second order cone constraints (SOC). An SOC constraint on the variable $\mathbf{x} \in \mathbb{R}^n$ is of the form $\mathbf{c}^\top \mathbf{x} + d \geq \|\mathbf{A}\mathbf{x} + \mathbf{b}\|_2$ where $\mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{m \times n}$ are given. SOCP problems can be efficiently solved by interior point methods for convex non-linear optimization [12]. As a special case of convex non-linear optimization, SOCPs have gained much attention in recent times. For a discussion of further efficient algorithms and applications of SOCP see [9].

The problem (1) can be equivalently written as the following convex quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi_j} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C \textstyle\sum_{j=1}^m \xi_j \\ \text{s.t.} \quad & y_j(\mathbf{w}^\top \mathbf{x}_j - b) - 1 + \xi_j \geq 0,\ \xi_j \geq 0,\ \forall j \end{aligned} \tag{2}$$

(2) is the famous SVM soft-margin formulation. The parameters C and W are related. However, W has the elegant geometric interpretation as a lower bound on the margin.

2. CLUSTERING BASED LARGE MARGIN CLASSIFIER

This section presents the clustering based classifier. Let Z_1 and Z_2 represent the random variables that generate the data points of the positive and negative classes respectively. Assume that the distributions of Z_1 and Z_2 can be modeled using mixture models, with component distributions having spherical covariances. Let k_1 be the number of components in the mixture model of positive class and k_2 be that in the negative class. Let $k = k_1 + k_2$. Let $X_j,\ j = 1, \ldots, k_1$ represent the random variable generating the j^{th} component of the positive class and $X_j,\ j = k_1 + 1, \ldots, k$ represent that generating the $(j - k_1)^{th}$ component of the negative class. Let X_j have the second order moments $(\mu_j, \sigma_j^2 \mathbf{I})$. The probability density functions (pdfs) of Z_1 and Z_2 can be written as $f_{Z_1}(\mathbf{z}) = \sum_{j=1}^{k_1} \rho_j f_{X_j}(\mathbf{z}),\ f_{Z_2}(\mathbf{z}) = \sum_{j=k_1+1}^{k} \rho_j f_{X_j}(\mathbf{z})$ where, ρ_j are the mixing probabilities ($\rho_j \geq 0, \sum_{j=1}^{j=k_1} \rho_j = 1$ and $\sum_{j=k_1+1}^{j=k} \rho_j = 1$). Any good clustering algorithm will correctly estimate the second order moments of the components. BIRCH is one such clustering algorithm, that scales well for large datasets. Given these estimates of second order moments, an optimal classifier that generalizes well must be built.

Let $\mathbf{w}^\top \mathbf{x} - b = 0$ be the discriminating hyperplane and $\mathbf{w}^\top \mathbf{x} - b = 1$, $\mathbf{w}^\top \mathbf{x} - b = -1$ be the corresponding set of supporting hyperplanes. As discussed in section 1.1, the constraints $\mathbf{w}^\top Z_1 - b \geq 1$ and $\mathbf{w}^\top Z_1 - b \leq -1$ ensure that training set error is low. Since Z_1 and Z_2 are random variables, the constraints cannot be always satisfied. Thus, we ensure that with high probability, the events $\mathbf{w}^\top Z_1 - b \geq 1$ and $\mathbf{w}^\top Z_1 - b \leq -1$ occur:[1]:

$$\begin{aligned} P(\mathbf{w}^\top Z_1 - b \geq 1) \geq \eta, \quad & P(\mathbf{w}^\top Z_2 - b \leq -1) \geq \eta \\ Z_1 \sim f_{Z_1}, \quad & Z_2 \sim f_{Z_2} \end{aligned} \tag{3}$$

where, η is a user defined parameter. η lower bounds the classification accuracy. Since the distribution of Z_i is a mixture model, in order to satisfy (3), it is sufficient that each of the components satisfy the following constraints:

$$\begin{aligned} P(\mathbf{w}^\top X_j - b \geq 1) \geq \eta, \quad & j = 1, \ldots, k_1 \\ P(\mathbf{w}^\top X_j - b \leq -1) \geq \eta, \quad & j = k_1 + 1, \ldots, k \\ X_j \sim f_{X_j}, \quad & j = 1, \ldots, k \end{aligned} \tag{4}$$

It can be easily seen that the constraints (4) are consistent only if the means of the components are linearly separable. Thus, in order to handle the case of outliers and almost linearly separable datasets, the constraints in (4) can be relaxed using some slack variables (ξ_i) and suitably penalizing the relaxation. This leads to the following large margin clas-

[1] $Z_i \sim f_{Z_i}$ means Z_i has the pdf given by f_{Z_i}

sification formulation (similar to (1)):

$$\begin{aligned}\min_{\mathbf{w},b,\xi_j} \quad & \textstyle\sum_{j=1}^{k} \xi_j \\ \text{s.t.} \quad & P(\mathbf{w}^\top X_j - b \geq 1 - \xi_j) \geq \eta,\ j = 1,\ldots,k_1, \\ & P(\mathbf{w}^\top X_j - b \leq -1 + \xi_j) \geq \eta,\ j = k_1+1,\ldots,k, \\ & \|\mathbf{w}\|_2 \leq W,\ \xi_j \geq 0,\ j = 1,\ldots,k \\ & X_j \sim f_{X_j}, j = 1,\ldots,k \end{aligned} \tag{5}$$

The constraints in the optimization problem (5) are probabilistic. In order to solve the optimization problem (5), the constraints need to be written as deterministic constraints. To this end, consider the following multivariate generalization of Chebyshev-Cantelli inequality [3, 10, 2].

THEOREM 1. *Let* $\mathbf{X}$ *be an* n *dimensional random vector. The mean and covariance of* $\mathbf{X}$ *be* $\mu \in \mathbb{R}^n$ *and* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$. *Let* $\mathcal{H}(\mathbf{w}, b) = \{\mathbf{z} | \mathbf{w}^\top \mathbf{z} < b, \mathbf{w}, \mathbf{z} \in \mathbb{R}^n, b \in \mathbb{R}\}$ *be a given half space, with* $\mathbf{w} \neq 0$. *Then*

$$Prob(\mathbf{X} \in \mathcal{H}) \geq \frac{s^2}{s^2 + \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w}} \tag{6}$$

where $s = (b - \mathbf{w}^\top \mu)_+,\ (x)_+ = max(x, 0)$.

Applying theorem 1 (see also [8]), the constraints for positive class can be handled by setting $P(\mathbf{w}^\top X_j - b \geq 1 - \xi_j)$:

$$\geq \frac{(\mathbf{w}^\top \mu_j - b - 1 + \xi_j)_+^2}{(\mathbf{w}^\top \mu_j - b - 1 + \xi_j)_+^2 + \mathbf{w}^\top \sigma_j^2 \mathbf{I} \mathbf{w}} \geq \eta$$

which results in the constraint

$$\mathbf{w}^\top \mu_j - b \geq 1 - \xi_j + \kappa\ \sigma_j \|\mathbf{w}\|_2 \tag{7}$$

where, $\kappa = \sqrt{\frac{\eta}{1-\eta}}$. Similarly the set of constraints on the negative class can be obtained.

Let $y_j, j = 1,\ldots,k$ represent the labels of the components (clusters). Note that $y_i = 1$ for k_1 components and $y_i = -1$ for the other k_2 components. Using this notation, (5) can be written as the following deterministic optimization problem:

$$\begin{aligned}\min_{\mathbf{w},b,\xi_j} \quad & \textstyle\sum_{j=1}^{k} \xi_j \\ \text{s.t.} \quad & y_j(\mathbf{w}^\top \mu_j - b) \geq 1 - \xi_j + \kappa\ \sigma_j \|\mathbf{w}\|_2,\ j = 1,\ldots,k \\ & \|\mathbf{w}\|_2 \leq W,\ \xi_j \geq 0,\ j = 1,\ldots,k \end{aligned} \tag{8}$$

One can derive tighter bounds on the probabilities in (5), by assuming that the component distributions in mixture model are Gaussian. In other words, assume that the distributions of Z_1 and Z_2 are modelled using Gaussian Mixture Models (GMMs). With such an assumption, one can write the constraints in (5) as deterministic constraints using:

$$\begin{aligned} P(\mathbf{w}^\top X_j - b \geq 1 - \xi_j) &= \Phi\left(\frac{\mathbf{w}^\top \mu_j - b - 1 + \xi_j}{\sigma_j \|\mathbf{w}\|_2}\right), \\ \Phi(z) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\left(-s^2/2\right)\ ds \end{aligned}$$

where Φ is the distribution function of univariate normal distribution with mean 0, unit variance. Thus, the constraints in (4) can be written as:

$$y_j(\mathbf{w}^\top \mu_j - b) \geq 1 - \xi_j + \kappa\ \sigma_j \|\mathbf{w}\|_2,\ j = 1,\ldots,k \tag{9}$$

where, $\kappa = \Phi^{-1}(\eta)$. Note that, the final form of the constraints with (7) or without (9) the assumption of Gaussian components are the same. In the following text, κ is assumed to be $\Phi^{-1}(\eta)$ if Gaussian components are assumed and $\sqrt{\frac{\eta}{1-\eta}}$ otherwise.

The constraints in (8) involving $\|\mathbf{w}\|_2$ can be written as:

$$\begin{aligned} \frac{y_j(\mathbf{w}^\top \mu_j - b) - 1 + \xi_j}{\kappa\ \sigma_j} &\geq \|\mathbf{w}\|_2,\ j = 1,\ldots,k \\ W &\geq \|\mathbf{w}\|_2 \end{aligned}$$

Thus, the optimization problem (8) can be written in the following equivalent form:

$$\begin{aligned}\min_{\mathbf{w},b,\xi_j} \quad & \textstyle\sum_{j=1}^{k} \xi_j \\ \text{s.t.} \quad & y_j(\mathbf{w}^\top \mu_j - b) \geq 1 - \xi_j + \kappa\ \sigma_j W,\ j = 1,\ldots,k \\ & W \geq \|\mathbf{w}\|_2,\ \xi_j \geq 0,\ j = 1,\ldots,k \end{aligned} \tag{10}$$

The classification formulation (10) is an SOCP problem. This problem can be solved to obtain the optimal values of $\mathbf{w}$ and b. The classification algorithm employed is summarized as follows:

- Using a scalable clustering algorithm cluster the positive and negative data points.
- Compute the second order moments of all the clusters.
- Solve the optimization problem (10), using SOCP solvers. This gives optimum values of $\mathbf{w}$ and b.
- The label of a new data point $\mathbf{x}$ is given by $\text{sign}(\mathbf{w}^\top \mathbf{x} - b)$.

Observe that when $\sigma_{ij} = 0$, the SVM formulation (1) and the present formulation are same. In other words, if each data point is considered to be a cluster, then both the formulations are same. Also, note that the number of linear inequalities in (1) is $2m$, whereas in the proposed formulation it is k. Thus, the proposed formulation is expected to scale very well to large datasets. The time-complexity of clustering algorithm like BIRCH is $O(m)$ and that of the optimization is independent of m. Thus, the overall algorithm is expected to have a training time of $O(m)$.

2.1 Geometric Interpretation and Dual

The constraints in (10) have an elegant geometric interpretation. In order to see this, consider the following problem. Suppose $B(\mathbf{c}, r) = \{\mathbf{x} | (\mathbf{x} - \mathbf{c})^\top (\mathbf{x} - \mathbf{c}) \leq r^2\}$ is the set of data points lying in the sphere B with center $\mathbf{c}$ and radius r. Assume that all points of set B belong to positive class. Consider the problem of classifying the points lying in $B(\mu, \kappa\sigma)$ correctly (allowing for slack variables):

$$\mathbf{w}^\top \mathbf{x} - b \geq 1 - \xi,\ \forall\ \mathbf{x}\ \in\ B(\mu, \kappa\sigma) \tag{11}$$

(11) has infinite number of constraints, but can be posed as a single constraint as shown below:

$$z \geq 1 - \xi,\ z = \min_{\mathbf{x}\ \in\ B(\mu,\kappa\sigma)} \mathbf{w}^\top \mathbf{x} - b \tag{12}$$

Geometrically, the constraints in (11) say that all points that belong to $B(\mu, \kappa\sigma)$ lie on the positive half space of the hyperplane $\mathbf{w}^\top \mathbf{x} - b = 1 - \xi$. This geometric picture (also see [4]) immediately shows that all the constraints (11) can be satisfied just by ensuring that the point in $B(\mu, \kappa\sigma)$ which is nearest to the hyperplane $\mathbf{w}^\top \mathbf{x} - b = 1 - \xi$ lies on the positive

half space. This idea is stated as equation (12). Finding the minimum distant point on a sphere to a given hyperplane is simple. Drop a perpendicular to the hyperplane from the sphere's center. The point at which the perpendicular intersects the sphere gives the minimum distant point ($\mathbf{x}^*$). Note that $\mathbf{x}^*$ is the optimum solution of (12). Using this geometrical argument, $\mathbf{x}^*$ can be calculated using: $\mathbf{x}^* - \mu = -\alpha\mathbf{w},\ \mathbf{x}^* \in B(\mu, \kappa\sigma)$. This gives $\mathbf{x}^* = \mu - \frac{\kappa\sigma\mathbf{w}}{\|\mathbf{w}\|_2}$. Now, (11) is satisfied if $\mathbf{w}^\top\mathbf{x}^* - b \geq 1 - \xi$. This says that[2],

$$\mathbf{w}^\top\mu - b \geq 1 - \xi + \kappa\ \sigma\|\mathbf{w}\|_2 \qquad (13)$$

Note that this equation is of the same form as (9). Hence, geometrical interpretation (see also [5]) of the constraints of (10) is to restrict the discriminating hyperplane to lie such that most of the spheres $B(\mu_j, \kappa\sigma_j)$ are classified correctly. Figure 1 shows this geometric picture. Note that in the figure except the sphere at $(5, 5)$, all the spheres satisfy the constraint with $\xi_j = 0$.

It is interesting to study the dual of the formulation (10). Using the dual norm characterization $\|\mathbf{w}\|_2 = \sup_{\|\mathbf{u}\|_2 \leq 1} \mathbf{u}^\top\mathbf{w}$ and the Lagrange multiplier theory, the dual can be written as:

$$\begin{aligned} \max_{\alpha_j, \lambda} \quad & \textstyle\sum_j \alpha_j + \kappa W \sum_j \alpha_j\sigma_j - \lambda W, \\ \text{s.t.} \quad & \|\textstyle\sum_j \alpha_j y_j \mu_j\|_2 \leq \lambda,\ \sum_j \alpha_j y_j = 0,\ 0 \leq \alpha_j \leq 1 \end{aligned} \qquad (14)$$

and the necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions can be written as:

$$\sum_j \alpha_j y_j \mu_j = \lambda\mathbf{u}, \quad \textstyle\sum_j \alpha_j y_j = 0, \quad \alpha_j + \beta_j = 1,$$

$$\alpha_j(1 - \xi_j + \kappa\sigma_j W - y_j(\mathbf{w}^\top - b)) = 0,$$

$$\beta_j\xi_j = 0,\ \lambda(\mathbf{w}^\top\mathbf{u} - W) = 0,\ \alpha_j \geq 0,\ \beta_j \geq 0,\ \lambda \geq 0 \qquad (15)$$

where $\alpha_j, \beta_j, \lambda$ are the Lagrange multipliers. Suppose $0 < \alpha_j < 1$ and $\lambda > 0$ then, from the KKT conditions it can be seen that $\xi_j = 0$, $\|\mathbf{w}\|_2 = W$ and $y_j(\mathbf{w}^\top\mu_j - b) = 1 + \kappa\sigma_j\|\mathbf{w}\|_2$. This says that the supporting hyperplanes are tangent to $B(\mu_j, \kappa\sigma_j)$. Extending the terminology used in case of SVMs, such spheres may be called as non-bound support spheres. Similarly one can define the bounded support spheres as spheres with $\alpha_j = 1$. Also, note that $\alpha_j = 1 \Rightarrow \xi_j > 0$. In figure 1, the spheres marked with 'o' are non-bound support spheres and hence are tangent to the supporting hyperplanes.

Note that the dual involves dot products of data points. This is because,

$$\|\sum_j \alpha_j y_j \mu_j\|_2 = \sqrt{\left(\sum_i \sum_j \alpha_i\alpha_j y_i y_j \mu_i^\top \mu_j\right)}$$

The estimate of σ_j^2 is $\frac{1}{m_j}\sum_{k=1}^{m_j}(\mathbf{x}_k - \mu_j)^\top(\mathbf{x}_k - \mu_j)$ where, $\mathbf{x}_k$ are the m_j data points that belong to j^{th} cluster. As the formulation (14) involves only the dot products of the data points, it can be extended to arbitrary feature spaces by using Mercer kernels [11].

Assuming that the given dataset is linearly separable, one can write an equivalent of the hard-margin classifier for the proposed formulation (10):

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2, \\ \text{s.t.} \quad & y_j(\mathbf{w}^\top\mathbf{x}_j - b) \geq 1 + \kappa\sigma_j\|\mathbf{w}\|_2\ \forall\ j \end{aligned} \qquad (16)$$

Interestingly, the dual of the problem (16) turns out to be the problem of finding distance between the convex hulls formed by the negative and positive spheres $(B(\mu_j, \kappa\sigma_j))$[3]. This is analogous to the case of SVMs, where dual is the problem of finding distance between the convex hulls formed by the negative and positive data points [1].

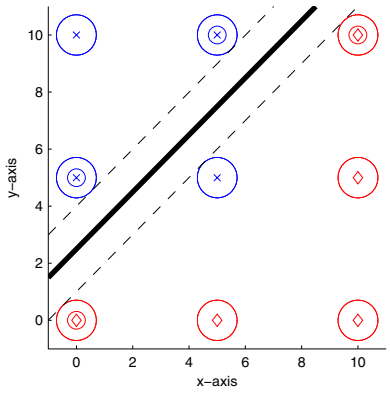

Figure 1: Illustration showing the geometric meaning of the constraints. Clusters marked with '×' have positive labels and those marked '◇' have negative labels. The radius of clusters is proportional to $\kappa\sigma_j$.

3. EXPERIMENTAL RESULTS

In this section, we present experimental results on synthetic and real world data sets. The results show that the accuracies achieved by SVM and the proposed classifier are comparable and that the proposed classification algorithm scales well for large datasets. In all cases, BIRCH was used to cluster the positive and negative training data points. The original BIRCH implementation by Zhang et.al. [17] was used for clustering. `SeDuMi` [14], a publicly available SOCP solver was used to solve the optimization problem (10) in all experiments. The performance of the proposed Clustering Based Classifier (**CB-SOCP**) was compared to that of SVM (using linear kernel) implemented by `LIBSVM` [6] (denoted by **SVM**)[4]. All experiments were carried on Pentium 4 2.4GHz machines with 1GB memory. A '×' in the tables 1,2 and 3 represents the failure of the corresponding classifier to complete training.

3.1 Parameter Setting

The parameter C (see (2)) of **SVM** was tuned for each dataset separately. The main parameters for BIRCH algorithm were chosen to be the default values as given in [17]. Since the values of k_1 and k_2 are not known for the real world datasets, they were chosen to be the number of leaf CF entries in the CF-tree for positive and negative data points respectively. However, in case of synthetic datasets since k_1

[2]The same constraint can be derived more rigorously using optimization theory

[3]Proof not provided due to space restrictions

[4]CB-SVM is not used for comparison of performance due to non-availability of its implementation

and k_2 are known, k_1 and k_2 were used as the number of clusters for positive and negative data points. The values of η and W were fixed to be 0.8 and 500. The values were not tuned for each dataset. However, in general, tuning of these parameters specifically for a dataset can give better results.

3.2 Synthetic Datasets

In this section, experiments on two large, almost linearly separable synthetic datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ are presented. $\mathcal{D}_1$ is a synthetic dataset with $m = 4,500,000$ and $n = 2$. $\mathcal{D}_1$ was generated using 9 Gaussian distributions with $\sigma = 0.5$ and centers on a 3×3 square grid. Each grid point is separated from the neighbor by 5 units. Equal number of points $(500,000)$ were generated from each cluster. The labels were assigned as shown in the figure 1. As seen from the figure, the dataset is linearly separable if the label of the cluster at the center of the grid is inversed. Along with the training set $\mathcal{D}_1$ a testset was also generated using the same Gaussian distributions. The size of testset was $450,000$ (10% of the training set size). $\mathcal{D}_2$ is a synthetic dataset with $m = 4,500,000$ and $n = 38$ such that the projection of $\mathcal{D}_2$ on plane formed by first two dimensions gives $\mathcal{D}_1$. Similarly the testset for $\mathcal{D}_2$ was also generated.

The results comparing the performance of the methods on $\mathcal{D}_1$ and $\mathcal{D}_2$ are shown in Table 1. In case of both datasets, the **SVM** classifier failed to complete training, whereas **CB-SOCP** gave high testset accuracy with small training time. In order to evaluate the growth of training time as a function of training set size, scaling experiments were performed on the datasets. Table 2,3 shows the scaling experiment results. The results show that the proposed algorithm is scalable and that the training time with **CB-SOCP** grows almost linearly with respect to sample size (see figure 2). In the tables, 'S-Rate' represents the fraction of training set, 'S-Size' represents the size of the sampled training set, t_1 represents the time for clustering the training data in seconds, t_2 represents the time for solving (10) in seconds and t represents the total time in seconds for training. Note that the time taken for solving the optimization problem (10) was 0.85 sec in all cases. This is as expected, since the complexity of the optimization problem grows with number of clusters k rather than with the number of data points m.

3.3 Real World Datasets

In this section, results on three large real world datasets — Web-Page, IJCNN1 and Intrusion detection are presented. The web-page dataset[5] has $49,749$ data points in 300 dimensions. The classification task is "Text categorization": classifying whether a web page belongs to a category or not. The IJCNN1 dataset[6] has $49,990$ data points in 22 dimensions. The intrusion detection dataset[7] has $4,898,430$ data points in 41 dimensions. The classification task is to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. This dataset has 7 categorical features and 3 of them take string values. Since the proposed classifier and the SVMs work for numerical data, these three features were removed from the training data. Hence, the final training data has 38 dimensions.

Table 2: Comparison of training times (t sec) with CB-SOCP and SVM on $\mathcal{D}_1$

S-Rate	S-Size	CB-SOCP			SVM
		t_1	t_2	t	t
0.01	45,000	0.36	0.85	1	214
0.05	225,000	2.27	0.85	3	4155
0.10	450,000	4.77	0.85	6	15279
0.30	1,350,000	14.3	0.85	15	×
0.50	2,250,000	24.79	0.85	26	×
0.70	3,150,000	35.61	0.85	36	×
0.90	4,050,000	46.39	0.85	47	×

Table 3: Comparison of training times (t sec) with CB-SOCP and SVM on $\mathcal{D}_2$

S-Rate	S-Size	CB-SOCP			SVM
		t_1	t_2	t	t
0.01	45,000	0.93	0.85	2	470
0.05	225,000	6.39	0.85	7	11576
0.10	450,000	12.2	0.85	13	52166
0.30	1,350,000	42.9	0.85	44	×
0.50	2,250,000	78	0.85	79	×
0.70	3,150,000	109.18	0.85	110	×
0.90	4,050,000	142.32	0.85	143	×

Table 4: Comparison of training times (t sec) with CB-SOCP and SVM on Intrusion dataset

S-Rate	S-Size	CB-SOCP			SVM
		t_1	t_2	t	t
0.10	494,020	12.84	8.08	21	3343
0.30	1,468,756	43.22	29.01	72	15652
0.50	2,449,224	75.19	9.02	84	44705
0.70	3,429,241	110.95	10.14	121	89101

The results comparing the performance of the methods on the real world datasets are shown in Table 1. In the case of web-page and IJCNN1 datasets, the accuracies obtained using **CB-SOCP** classifier are comparable to those obtained with **SVM** classifier. However, the proposed algorithm requires much less training time than the **SVM** classifier. The **SVM** classifier did not complete training with intrusion detection dataset. Whereas, **CB-SOCP** with small training time achieved high accuracy. Table 4[8] shows the scaling experiment results on the intrusion detection dataset. The results show that the proposed algorithm is scalable and that the training time with **CB-SOCP** grows almost linearly with respect to sample size (see figure 2).

4. CONCLUSIONS

A classification method which is scalable to very large datasets has been proposed, using SOCP formulations. Assuming that the class conditional densities of positive and negative data points can be modeled using mixture models, the second order moments of the components of mixture are estimated using a scalable clustering algorithm like BIRCH. Using the second order moments, an SOCP formulation is proposed which ensures that most of the clusters are classi-

[5]Training and testset available at `http://research.microsoft.com/~jplatt/smo.html`

[6]Training and testset available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`

[7]Training and testset available at `http://www.ics.uci.edu/~kdd/databases/kddcup99/kddcup99.html`

[8]Notation used is described in section 3.2

Table 1: Results on some large datasets, comparing the performance of CB-SOCP and SVM.

Dataset	m	Accuracy %		Total Time (sec)	
		CB-SOCP	SVM	CB-SOCP	SVM
Web-page	49,749	97.24	98.79	12	80
IJCNN1	35,000	90.52	91.64	2	71
Intrusion	4,898,430	91.71	×	176	×
$\mathcal{D}_1$	4,500,000	88.88	×	53	×
$\mathcal{D}_2$	4,500,000	88.88	×	161	×

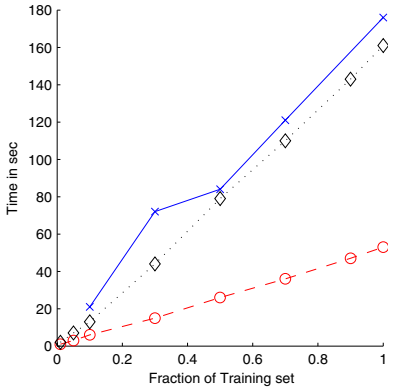

Figure 2: Graph showing that the training time of CB-SOCP grows almost linearly with m. Solid line, dashed line and dotted line represent Intrusion, $\mathcal{D}_1$ and $\mathcal{D}_2$ datasets respectively.

fied correctly. The geometric interpretation of the formulation, is to classify spherical clusters $B(\mu_j, \kappa\sigma_j)$ with as little error as possible. Experiments on synthetic and real world datasets show that the proposed method achieves good accuracy with $O(m)$ training time.

As pointed in section 2.1, the optimization formulation can be extended to non-linear classifiers. However, a scalable clustering algorithm that clusters data points in feature space needs to be built. In future, we would like to explore such clustering schemes. We would also like to explore the possibility of extending the SMO algorithm to solve the dual (14) of the proposed optimization problem. Another direction of future work is to explore the possibility of designing fast nearest point algorithms to solve the dual of the hard-margin formulation (16).

5. ACKNOWLEDGMENTS

The first author is supported by DST (Department of Science and Technology, Government of India) project DSTO /ECA/CB/660.

6. REFERENCES

[1] K. P. Bennett and E. J. Bredensteiner. Duality and geometry in SVM classifiers. In *Proceedings of the International Conference on Machine Learning*, pages 57–64, 2000.

[2] D. Bertsimas and J. Sethuraman. Moment problems and semidefinite optimization. *Handbook of Semidefinite optimization*, pages 469–509, 2001.

[3] C. Bhattacharyya. Second order cone programming formulations for feature selection. *Journal of Machine Learning Research*, 5:1417–1433, 2004.

[4] C. Bhattacharyya, P. K. Shivaswamy, and A. J. Smola. A second order cone programming formulation for classifying missing data. In *NIPS*, 2004.

[5] J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods—Support Vector Learning*, pages 169–184, Cambridge, MA, 1999.

[8] G. R. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2003.

[9] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1–3):193–228, 1998.

[10] A. W. Marshall and I. Olkin. Multivariate chebychev inequalities. *Annals of Mathematical Statistics*, 31(4):1001–1014, 1960.

[11] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, A 209:415–446, 1909.

[12] Y. Nesterov and A. Nemirovskii. *Interior Point Algorithms in Convex Programming*. Number 13 in Studies in Applied Mathematics. SIAM, 1993.

[13] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods—Support Vector Learning*, pages 185–208, Cambridge, MA, 1999.

[14] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[15] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

[16] H. Yu, J. Yang, and J. Han. Classifying large data sets using svm with hierarchical clusters. In *Proceedings of the ACM SIGKDD International Conference*, 2003.

[17] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference*, pages 103–114, 1996.