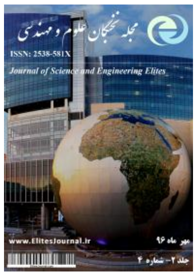
www.ElitesJournal.ir

مجله نخبگان علوم و مهندسی

Journal of Science and Engineering Elites

ISSN 2538-581X

جلد ۲- شماره ۴- سال ۱۳۹۶



An Overview on Second Order Cone Programming Formulations for Support Vector Machine Algorithms

Saeideh Roshanfekr[1]*

1- Master of Computer Engineering, University of Zanjan, Iran

*S.Roshanfekr@Znu.ac.ir

Received: September 2017   Accepted: October 2017

Abstract

In this tutorial we give an overview of the basic ideas second order cone programming (SOCP) formulations machines for support vector machine (SVM) algorithms. Furthermore, we include a summary of currently used algorithms for training SVM algorithms, covering both the quadratic (or convex) programming part and advanced methods for dealing with imbalanced and multiclass datasets. Finally, we mention some modifications and extensions that have been applied to the standard SOCP algorithm, and discuss the aspect of regularization from a SOCP-SVM perspective.

Keywords: Second order cone programming, support vector machine, imbalanced dataset, multi-class.

1. Introduction

The classification is one of the main topics in data mining, since it is used in different of fields such as computer vision [1], Credit Scoring [2], medical diagnosis [3] and so on [4-7]. Among the available proposed methods, SVM [8-10] has many advantages, including the ability to generalize the new sample and the low number of parameters [11]. The main idea of SVM is to find an optimal hyperplane between positive and negative samples of class. The optimal hyperplane, with a maximized margin between the two parallel hyperplanes, is obtained which minimizes a Quadratic Programming Problem (QPP).

TwinSVM is one of the extensions of SVM. This method makes two classifiers, one of them is very close to samples one of the classes and the second hyperplane is the maximum distance [12,13]. In recent years, in SVM community, SOCP (Second Order Cone Programming) methods has is highly regarded because performance of these methods is usually better than the other initial proposed methods. these methods in the papers [14,15] have a lot of large number of constraints, which increases the time complexity. In front of, the method proposed in the paper [16] describes that by considering all possible choices of class-conditional densities in a way that with a given mean and covariance and also for each training pattern considered to be a different constraint. There have been many reforms in the field of SVM [17-19]. In the paper [17], SOCP-SVM method was implemented in

the form of soft margin. In this way, the complexity of the classifier was controlled by adding a slack variable to the conic constraint. In the paper [18], NHSVM (Nonparallel Hyperplane SVM) [20] was implemented as SOCP. In the paper [19], the combination of TBSVM [21] and SOCP-SVM was presented. In the same formulation, a method to classify multi-class data was also provided [22].
This paper is categorized as follows. In Section 2, we review the theory and the algorithm thought of three representative SVM algorithms, Section 3 gives the research progress of them. Then, in Section 4, we list specific applications of SOCPSVM algorithms in recent years. Finally, we provide concluding remarks and discuss the direction of future research.

2. Review of representatives SOCP SVM algorithms

In this section, we explain the formulas of SVM [4,6], TwinSVM [12]. subsequently, the NH-SVM formulation [20] is presented. In these methods, consider a set of samples with their labels (x_k,y_k) that for $x_i \in R^n$ i = 1…m and $y_i \in \{-1,+1\}$ and the number of elements of the positive and negative class by m1 and m2 respectively, by $A \in R^{m1\times n}$ a data matrix for the positive class and $B \in R^{m2\times n}$ a data matrix for the negative class.
For binary data classification problems, SOCP SVM algorithms aim to find a hyperplane for each class, such that each hyperplane is proximal to the data points of one class and far from the data points of the other class. SOCP-SVM, ξ-SOCP-SVM, SOCP-TWSVM and RNH-SVM are four representative algorithms of SOCP SVM, and all the other methods are improved versions based on them. So, we firstly introduce the algorithm thought of the four representative SOCP SVM algorithms in this Section.

2.1. Support vector machine

The SVM is a binary classification method that determines the optimal hyperplane that separates the convex hulls of both classes. SVM build a classifier of the form $W^T x + b$ that maximizes the distance from it to the nearest sample in each class.

2.1.1. Linear SVM

In paper [8], by adding misclassification by minimization of the Euclidean norm and the misclassification error by introducing slack variables ξ_i, i = 1, ..., m for a sample xi and a penalty parameter C, that controls this tradeoff, the following formula is introduced:

$$\min_{w,b} \ \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{m}\xi_i \tag{1}$$

$$\text{subject to} \quad y_i(W^T x_i + b) \ge 1 - \xi_i . \quad i = 1 \dots m.$$
$$\xi_i \ge 0 \quad i = 1 \dots m.$$

2.1.2. Nonlinear SVM

A nonlinear function can be achieved via the Kernel Trick on the formulation (1) [8].

$$\max_{\alpha} \ \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i.s=1}^{m}\alpha_i\alpha_s y_i y_s \mathcal{K}(x_i, x_s) \tag{2}$$

$$\text{subject to} \quad \sum_{i=1}^{m}\alpha_i y_i = 0 .$$
$$0 \le \alpha_i \le C \quad i = 1 \dots m.$$

where $\alpha \in R^m$ is dual variables corresponding to the constraints in (1) and $\mathcal{K}(x_i . x_s)$ is a kernel function. One of the kernel function is Gaussian kernel, which has the following form:

$$\mathcal{K}(x_i . x_s) = \exp\left(\frac{\|x_i - x_s\|^2}{2\mathfrak{z}^2}\right) \quad (3)$$

where $\mathfrak{z}$ is a free parameter.

2.2. Twin support vector machine

TwinSVM is a classifier in which samples are separated by creating two nonparallel hyperplanes instead of a hyperplane. These hyperplanes are solved by using two smaller QPPs. In fact, each of the two QPPs has the formulation of a SVM; the only difference is that instead of all the samples in constrains appear to be the only samples of the same class.

2.2.1. Linear Twin SVM

TwinSVM formulation [12] includes two hyperplanes in the following form:

$$W_1^T x + b_1 = 0 \qquad W_2^T x + b_2 = 0 \quad (4)$$

In this case, each hyperplane to your class samples is very closely and samples of the other class are the maximum distance. The linear TwinSVM formulation will be solved using the following QPP equations:

$$\min_{W_1 . b_1 . \xi_2} \frac{1}{2}\|AW_1 + e_1 b_1\|^2 + \frac{c_1}{2}(\|W_1\|^2 + b_1^2) + c_3 e_2^T \xi_2 \quad (5)$$

$$\text{subject to} \quad -(BW_1 + e_2 b_1) \geq e_2 - \xi_2, \quad \xi_2 \geq 0.$$

And;

$$\min_{W_2 . b_2 . \xi_1} \frac{1}{2}\|BW_2 + e_2 b_2\|^2 + \frac{c_2}{2}(\|W_2\|^2 + b_2^2) + c_4 e_1^T \xi_1 \quad (6)$$

$$\text{subject to} \quad (AW_2 + e_1 b_2) \geq e_1 - \xi_1, \quad \xi_1 \geq 0.$$

where c_1, c_2, c_3, c_4 are penalty parameters, and e_1 and e_2 are vectors of ones of appropriates dimensions.

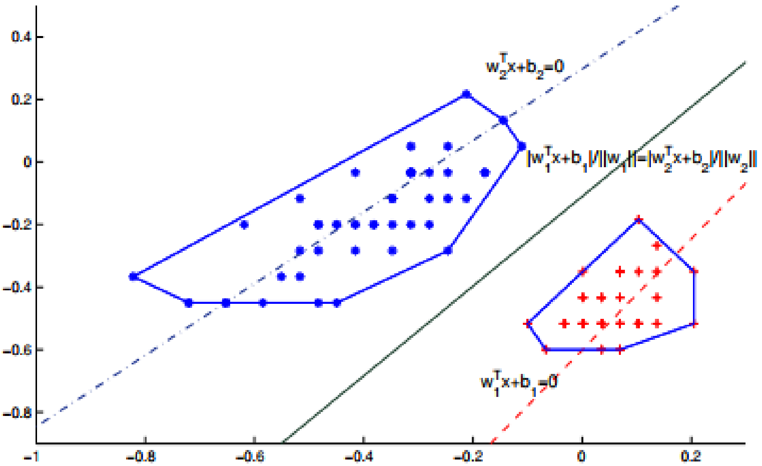


Figure 1- Geometric interpretation for Twin SVM

2.2.2. Nonlinear TwinSVM

The nonlinear TwinSVM is formulated as

$$\min_{u_1, b_1, \xi_2} \frac{1}{2}\|\mathcal{K}(A^T, \mathbb{X})u_1 + e_1 b_1\|^2 + \frac{c_1}{2}(\|u_1\|^2 + b_1^2) + c_3 e_2^T \xi_2 \tag{7}$$

$$\text{subject to} \quad -(\mathcal{K}(B^T, \mathbb{X})u_1 + e_2 b_1) \geq e_2 - \xi_2 \qquad \xi_2 \geq 0$$

and

$$\min_{u_2, b_2, \xi_1} \frac{1}{2}\|\mathcal{K}(B^T. \mathbb{X})u_2 + e_2 b_2\|^2 + \frac{c_2}{2}(\|u_2\|^2 + b_2^2) + c_4 e_1^T \xi_1 \tag{8}$$

$$\text{subject to} \quad (\mathcal{K}(A^T, \mathbb{X})u_2 + e_1 b_2) \geq e_1 - \xi_1, \qquad \xi_1 \geq 0.$$

where $\mathcal{K}(x_i . x_s)$ is a kernel function. One of the kernel function identify according to Equation (3). ξ is a slack variable. c_1, c_2 are penalty parameters and e_1 and e_2 are vectors of ones of appropriates dimensions. u_1 and u_2 are line equations.

2.3. Nonparallel Hyperplane SVM (NH-SVM)

The NH-SVM method constructs two nonparallel hyperplanes by solving a single QPP. The linear NH-SVM formulation finds two hyperplanes that each classifier is close to one of the training patterns and far away from the other class.

2.3.1. Linear NH-SVM

The linear NH-SVM formulation follows [20]:

$$\min_{\substack{W_k, b_k, \xi_k \\ k=1,2}} \frac{1}{2}(\|AW_1 + e_1 b_1\|^2 + \|BW_2 + e_2 b_2\|^2) + \frac{c_1}{2}(\|W_1\|^2 + b_1^2 + \|W_2\|^2 + b_2^2) + \frac{c_2}{2}(e_1^T \xi_1 + e_2^T \xi_2) \tag{9}$$

$$\text{subject to} \quad AW_1 + e_1 b_1 - AW_2 - e_1 b_2 \geq e_1 - \xi_1,$$

$$BW_2 + e_2 b_2 - BW_1 - e_2 b_1 \geq e_2 - \xi_2,$$

$$\xi_1 \geq 0, \qquad \xi_2 \geq 0.$$

where c1, c2 > 0 are regularization parameters. A point x in R^n is assigned to class k' by identifying the nearest hyperplane according to Equation (10) and e_1 and e_2 are vectors of ones of appropriates dimensions.

There are three terms in the objective function in (20). The second is the sum of squared distances from the two hyperplanes to points in the corresponding class, keeping the hyperplanes close to the points of the corresponding class. The third is the sum of error variables related with the constraints, requiring the distances from the two hyperplanes to points in the other class to be one or greater.

2.3.2. Nonlinear NH-SVM

The kernel-based formulation NH-SVM can be obtained via the kernel trick. This formulation is given by

$$\min_{\substack{u_k, b_k, \xi_k \\ k=1.2}} \frac{1}{2}(\|\mathcal{K}(A^T,\mathbb{X})u_1 + e_1b_1\|^2 + \|\mathcal{K}(B^T,\mathbb{X})u_2 + e_2b_2\|^2) + \frac{c_1}{2}(\|u_1\|^2 + b_1^2 + \|u_2\|^2 + b_2^2) + \frac{c_2}{2}(e_1^T\xi_1 + e_2^T\xi_2) \quad (10)$$

$$\text{subject to} \quad \mathcal{K}(A^T,\mathbb{X})u_1 + e_1b_1 - \mathcal{K}(A^T,\mathbb{X})u_2 - e_1b_2 \geq e_1 - \xi_1,$$

$$\mathcal{K}(B^T,\mathbb{X})u_2 + e_2b_2 - \mathcal{K}(B^T,\mathbb{X})u_1 - e_2b_1 \geq e_2 - \xi_2,$$

$$\xi_1 \geq 0, \qquad \xi_2 \geq 0.$$

where c1, c2 are positive parameters [20] and K(x,y) is kernel function and e_1 and e_2 are vectors of ones of appropriates dimensions.

2.4. Second order cone programming support vector machine

Let A and B be vectors that generate the samples of the positive and negative classes respectively, with mean μ_k and covariance $\sum_k$, k=1,2. In order to construct a classifier in a way that the probability of false-negative and false-positive errors does not exceed 1−η1 and 1−η2, with η1, η2 ∈ (0, 1), respectively [16].

2.4.1. Linear SOCP-SVM

The SOCP-SVM formulation provides a robust and efficient framework for classification since it considers all possible choices of class-conditional densities with a given mean and covariance matrix, achieving great predictive results under different conditions of the data sets. Let X1 and X2 be random vectors that generate the samples of the positive and negative classes respectively, with means and covariance matrices. This problem is formulated as follows:

$$\min_{w,b} \frac{1}{2}\|W\|^2 \quad (11)$$

$$\text{subject to} \quad W^T\mu_1 + b \geq 1 + \kappa_1\|S_1^TW\|$$

$$-(W^T\mu_1 + b) \geq 1 + \kappa_2\|S_2^TW\|$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ and $\sum_k = S_kS_k^T$ for k=1,2 [16]. The probability of false-negative and false-positive errors does not exceed 1−η1 and 1−η2, with η1, η2 ∈ (0, 1), respectively.

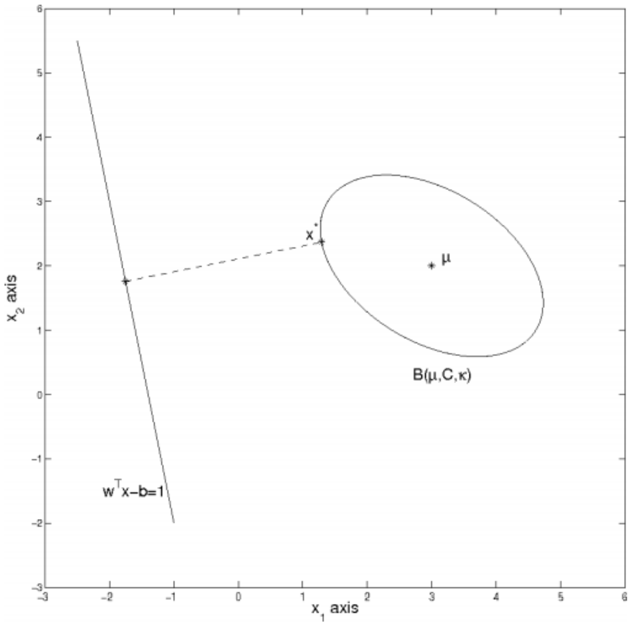


Figure 2- Illustration showing SOCP-SVM [16]

2.4.2. Nonlinear SOCP-SVM

A nonlinear version can be obtained from formulation (11) by rewriting weight vector $W \in R^n$ as $W = \mathbb{X}s + Mr$, where M is a matrix whose columns are orthogonal to samples; and $\mathbb{X} = [A^T\ B^T] \in R^{n\times m}$ is the data matrix containing both positive and negative classes; s and r are vectors with the appropriate dimension and is obtained from combining coefficients.
The mean μ_i and covariance $\sum_i$ are given by:

$$\hat{\mu}_1 = \frac{1}{m_1}A^T e_1,\quad \hat{\mu}_2 = \frac{1}{m_2}B^T e_2,\quad \hat{\sum}_k = S_k S_k^T,\ \ k = 1,2 \qquad (12)$$

$$S_1 = \frac{1}{\sqrt{m_1}}(A^T - \hat{\mu}_1 e_2{}^T),\quad S_2 = \frac{1}{\sqrt{m_2}}(B^T - \hat{\mu}_2 e_1{}^T). \qquad (13)$$

thus, there is the following equality for each class k

$$W^T\mu_k = s^T g_k \quad W^T{\sum}_k w = s^T \Xi_k s \quad k = 1.2.$$

where

$$g_k = \frac{1}{m_k}\begin{bmatrix} K_{1k}e_k \\ K_{2k}e_k \end{bmatrix},\qquad \Xi_k = \frac{1}{m_k}\begin{bmatrix} K_{1k} \\ K_{2k} \end{bmatrix}\left(I_{m_k} - \frac{1}{m_k}e_k e_k^T\right)\left[K_{1k}^T\ \ K_{2k}^T\right].$$

with $K_{11} = AA^T$ ،$K_{12} = K_{21}{}^T = BA^T$ ،$K_{22} = BB^T$ matrices whose elements are inner products of data points.

$$AW_1 = [K_{11}\ \ K_{12}]s_1 = K_1 \circ s_1.\quad BW_2 = [K_{21}\ \ K_{22}]s_2 = K_2 \circ s_2$$

Therefore, the nonlinear formulation is given by:

$$\min_{s_1.b_1.\xi_2} \frac{1}{2}s^T K s \qquad (14)$$

$$\text{subject to}\quad s^T g_1 - b \geq 1 + \kappa_1\sqrt{s^T\Xi_1 s}$$

$$-s^T g_2 - b \geq 1 + \kappa_2 \sqrt{s^T \Xi_2 s}$$

where $K = [K_{11}, K_{12}; K_{21}, K_{22}] \in R^{m\times m}$ [16].

2.5. ξ-Second order cone programming support vector machine

Maldonado et al. proposed an improved SOCP-SVM also named as ξ-SOCP-SVM. This method extends the soft margin SVM approach [8] for training data. The main issue is to provide a relaxation of the conic constraints by including a slack variable.

2.5.1. Linear ξ-SOCP-SVM

This formulation extends the ideas of the soft-margin SVM approach for training data that are not linearly separable to SOCP-SVM. The main idea is to provide a relaxation of the conic constraints by including a slack variable, penalizing it in the objective function. The structural risk is then controlled by the trade-off between the Euclidean norm minimization and a margin variable ξ. This problem can now be stated as the following SOCP problem:

$$\begin{aligned} \min_{w.b}\ & \frac{1}{2}\|W\|^2 + C\xi \\ \text{subject to}\ & W^T\mu_1 + b \geq 1 - \xi + \kappa_1 \|S_1^T W\| \\ & -(W^T\mu_1 + b) \geq 1 - \xi + \kappa_2 \|S_2^T W\| \\ & \xi \geq 0 \end{aligned} \tag{15}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ and $\sum_k = S_k S_k^T$ for k=1,2. The kernel-based for ξ-SOCP-SVM method has not been implemented [17].

2.6. Second order cone programming Twin SVM

SOCP-TwinSVM is to develop a new classification method which is formulated as second order cone programming presented in [19]. SOCP-TwinSVM builds two nonparallel hyperplanes that each plane is closer to one of the two classes and is as far as possible from the other. In this method, instead of the convex hulls, ellipsoids are used to characterize each class.

2.6.1. Linear SOCP-TwinSVM

The reasoning behind this approach is developing two nonparallel classifiers in such a way that each hyperplane is closest to one of the two classes and as far as possible from the other class. However, ellipsoids are used to characterize each training pattern instead of the convex hulls. The SOCP-TwinSVM problem has been achieved as the SOCP problem:

$$\begin{aligned} \min_{W_1.b_1.\xi_2}\ & \frac{1}{2}\|AW_1 + e_1 b_1\|^2 + \frac{\theta_1}{2}(\|W_1\|^2 + b_1^2) \\ \text{subject to}\ & -W_1^T\mu_2 - b_1 \geq 1 + \kappa_2 \|S_2^T W_1\| \end{aligned} \tag{16}$$

And

$$\begin{aligned} \min_{W_2.b_2.\xi_1}\ & \frac{1}{2}\|BW_2 + e_2 b_2\|^2 + \frac{\theta_2}{2}(\|W_2\|^2 + b_2^2) \\ \text{subject to}\ & W_2^T\mu_1 + b_2 \geq 1 + \kappa_1 \|S_1^T W_2\| \end{aligned} \tag{17}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$ and $\sum_k = S_k S_k^T$ for k=1,2.

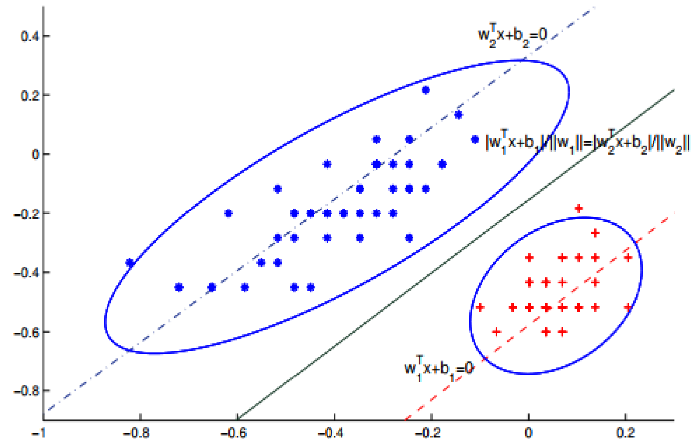


Figure 3- Geometric interpretation for Twin SOCP-SVM [19]

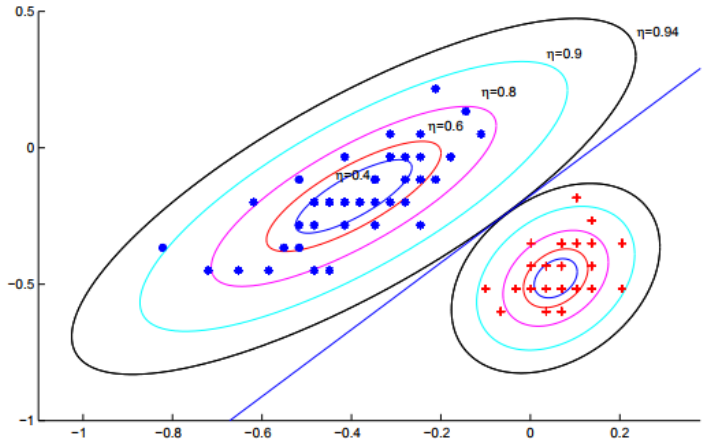


Figure 4- Geometric interpretation for Twin SOCP-SVM, influence of parameter η

Figure 3 observe that η controls the size of the ellipsoid of the respective class: higher values of η imply bigger ellipsoids. Uneven η values allow the method to manage the classification performance for each class, favoring the one with higher error costs, for sample.

2.6.2. Nonlinear SOCP-TwinSVM

In this section, Twin SOCP-SVM to Kernel functions to obtain non-linear classifiers for the problems (16) and (17) are as following:

$$\min_{s_1.b_1.\xi_2} \frac{1}{2}\|K_1 \circ s_1 + e_1 b_1\|^2 + \frac{\theta_1}{2}(\|s_1\|^2 + b_1^2) \quad (18)$$

subject to $-s_1^T g_2 - b_1 \geq 1 + \kappa_2\|\Lambda_2^T s_1\|$

$$\min_{s_2.b_2.\xi_1} \frac{1}{2}\|K_2 \circ s_2 + e_2 b_2\|^2 + \frac{\theta_2}{2}(\|s_2\|^2 + b_2^2) \quad (19)$$

subject to $s_2^T g_1 + b_2 \geq 1 + \kappa_1\|\Lambda_1^T s_2\|$

The notations introduced in section 2.4.2 [19].

2.7. Robust Nonparallel Hyperplane SVM (RNH-SVM)

An approach for binary classification using second-order cones and nonparallel hyperplanes is presented in this section.

2.7.1. Linear RNH-SVM

In this Section, we introduce a formulation for the method, Nonparallel Hyperplane SVM. Its main contribution is the use of robust optimization techniques in order to construct nonlinear models with superior performance and appealing geometrical properties. This formulation extends the ideas of the NH-SVM approach [20] to second-order cones.

$$\min_{\substack{W_k.b_k\\ k=1.2}} \frac{1}{2}(\|AW_1+e_1b_1\|^2+\|BW_2+e_2b_2\|^2) + \frac{c_1}{2}(\|W_1\|^2+b_1^2+\|W_2\|^2+b_2^2) \tag{20}$$

$$\text{subject to} \quad (W_1-W_2)^T\mu_1+(b_1-b_2)\ge 1+\kappa_1\left\|S_1^T(W_1-W_2)\right\| .$$

$$-((W_1-W_2)^T\mu_2+(b_1-b_2))\ge 1+\kappa_2\left\|S_2^T(W_1-W_2)\right\|$$

A kernel-based formulation NH-SVM can be obtained via the kernel trick. Following the notation introduced in section 2.4.2. This formulation is given by

$$\min_{s_k.b_k} \frac{1}{2}(\|K_1s_1+e_1b_1\|^2+\|K_2s_2+e_2b_2\|^2) + \frac{c_1}{2}(\|s_1\|^2+b_1^2+\|W_2\|^2+b_2^2) \tag{21}$$

$$\text{subject to} \quad (s_1-s_2)^Tg_1+(b_1-b_2)\ge 1+\kappa_1\left\|\Lambda_1^T(s_1-s_2)\right\| .$$
$$-((s_1-s_2)^Tg_2+(b_1-b_2))\ge 1+\kappa_2\left\|\Lambda_2^T(s_1-s_2)\right\|$$

2.7.2. Nonlinear RNH-SVM

In this Section, we show the nonlinear RNH-SVM formulation. Consider the following kernel-generated surface [20]:

$$K(x^T,X^T)W_1+b_1=0, K(x^T,X^T)W_2+b_2=0$$

where $X^T=[X_1\ X_2]^T$ and K is a type of kernel. The above surfaces are determined by the primal problem

$$\min_{W_k.b_k,\xi_k} \frac{1}{2}\begin{pmatrix}\|W_1\|^2+b_1^2\\ +\|W_2\|^2+b_2^2\end{pmatrix}+\frac{c_1}{2}\begin{pmatrix}\|K(X_1^T,X^T)W_1+e_1b_1\|^2\\ +\|K(X_2^T,X^T)W_2+e_2b_2\|^2\end{pmatrix} + c_2(e_1^T\xi_1+e_2^T\xi_2) \tag{22}$$

$$\begin{aligned} &(K(X_1^T,X^T)W_1+e_1b_1)-(K(X_1^T,X^T)W_2+e_1b_2)\ge e_1-\xi_1,\\ \textit{subject to}\ &(K(X_2^T,X^T)W_2+e_2b_2)-(K(X_2^T,X^T)W_1+e_2b_1)\ge e_2-\xi_2,\\ &\xi_1\ge 0,\xi_2\ge 0\end{aligned}$$

Where c1, c2 are positive parameters [20] and K(x,y) is kernel function and e_1 and e_2 are vectors of ones of appropriates dimensions.

3. Improvement on SOCP SVM algorithms

In this Section, we supply some improvements on SOCP SVM during the last few years from two aspects of multi class and imbalanced classification.

3.1. Multi class based algorithms

In paper [22] is presented novel second-order cone programming (SOCP) formulations that determine a linear multi-class predictor using Support Vector Machines (SVMs). The quadratic chance-constrained programming problem is proposed for each class k=1,…,K:

$$\min_{W_i.b_i} \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\left\|W_i - W_j\right\|^2 + \frac{1}{2}\sum_{i=1}^{K}\|W_i\|^2 \quad (22)$$

$$subject\ to\ (W_i - W_j)^T.\mu_i - (b_i - b_j) \geq 1 + k_{ij}\left\|S_i^T(W_i - W_j)\right\|$$
$$i,j = 1,\dots,K, i \neq j$$

Where $\Sigma_i = S_i S_i^T$ and $k_{ij} = \sqrt{\frac{\eta_{ij}}{1-\eta_{ij}}}$, for I,j=1,…,K, $i \neq j$.

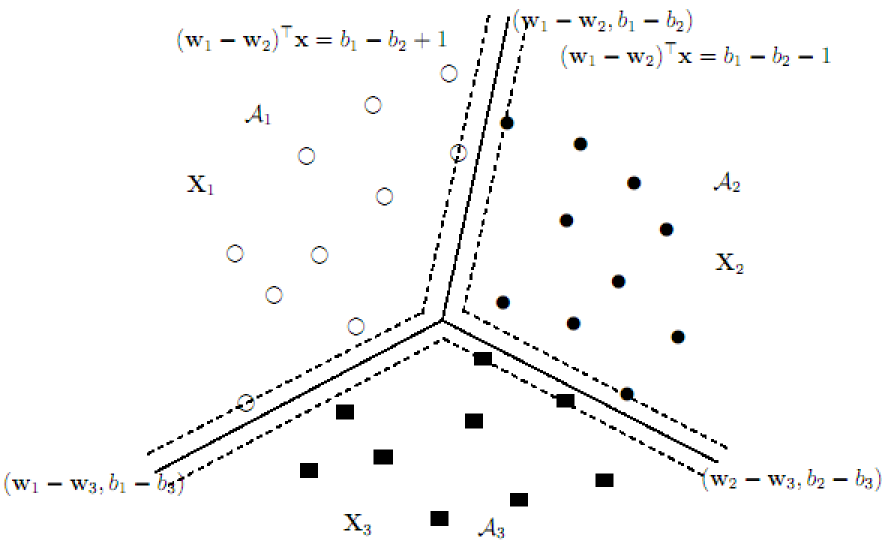


Figure 5- Linear separator for three classes [22]

3.2. Imbalanced datasets based algorithms

In this paper [23], a SOCP-SVM formulation for imbalanced dataset is presented. The main idea is to improve LP-SVM to a formulation based on second order cone programming. The linear SOCP programming problem is proposed:

$$\min_{w,b,r} -r \quad (23)$$

$$subject\ to\ \begin{aligned} W^T.\mu_1 - b &\geq r + \kappa_1\|S_1^T W\|,\\ b - W^T.\mu_2 &\geq r + \kappa_2\|S_2^T W\|,\\ -1 \leq W_j &\leq 1, j = 1,\dots,n.\\ r &\geq 0 \end{aligned}$$

For improve the previous formulation, margin variable r is then replaced by $r_1$and r_2, one for each conic constrain and matrixed in objective function. The trade-off between both values is managed by a positive parameter C:

$$\min_{w,b,r_1,r_2} -r_1 - Cr_2 \quad (24)$$
$$\text{subject to} \quad \begin{aligned} W^T.\mu_1 - b &\geq r_1 + \kappa_1\|S_1^T W\|, \\ b - W^T.\mu_2 &\geq r_2 + \kappa_2\|S_2^T W\|, \\ -1 \leq W_j &\leq 1, j = 1, \dots, n. \\ r_1, r_2 &\geq 0 \end{aligned}$$

In this work, we introduce a cost-sensitive approach for classifying imbalanced data via direct margin maximization, where this task is performed separately for each class, and which benefits the correct prediction of the target class.

4. Conclusion

In this paper, we review SVM approach, which extends the ideas of SVM, Twin SVM to second-order cones. SVM is an effective method for classification problem and second order approaches for SVM have the ability of generalizing training patterns effectively by considering a robust setting for data distribution. The design benefits the correct prediction of both classes. Although it extended to multi-class classification.

The approaches construct nonparallel classifier, and represents each training pattern by an ellipsoid characterized by the mean and covariance of each class, instead of the reduced convex hulls used in standard SVM. Originally developed for linear classifiers, the method is also adapted to construct nonlinear classification functions via the kernel trick. The use of ellipsoids for SVM modeling has been applied successfully in the context of expert systems.

In view of the above shortages, SOCP-SVM needs further improvement. The items presented in this study include: Improve the existing model, and propose TWSVM model with this formulation. SOCP-TWSVM has high performance and Extend the SOCP-SVM idea to multi-class classification. SOCP-SVM can work when there are three or more types of training samples. Also, Extend the SOCP-SVM idea to imbalanced data classification. SOCP-SVM can work when datasets are imbalanced.

5. Reference

1. Zhao, Ying, et al. "Image processing based recognition of images with a limited number of pixels using simulated prosthetic vision." Information Sciences 180.16 (2010): 2915-2924.

2. Thomas, Lyn C., David B. Edelman, and Jonathan N. Crook. Credit scoring and its applications. Siam, 2002.

3. Straszecka, Ewa. "Combining uncertainty and imprecision in models of medical diagnosis." Information Sciences 176.20 (2006): 3026-3059.

4. Meyer, David, Friedrich Leisch, and Kurt Hornik. "The support vector machine under test." Neurocomputing 55.1 (2003): 169-186.

5. Noble, William Stafford. "Support vector machine applications in computational biology." Kernel methods in computational biology (2004): 71-92.

6. Wu, Yu-Chieh, Yue-Shi Lee, and Jie-Chi Yang. "Robust and efficient multiclass SVM models for phrase pattern recognition." Pattern recognition41.9 (2008): 2874-2889.

7. Wang, Xiang-Yang, Ting Wang, and Juan Bu. "Color image segmentation using pixel wise support vector machine classification." Pattern Recognition44.4 (2011): 777-787.

8. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

9. Vapnik, Vladimir. "Statistical learning theory. 1998." (1998).

10. Cortes, Corinna, and Vladimir Vapnik. "Support vector machine." Machine learning 20.3 (1995): 273-297.

11. Khan, Naimul Mefraz, et al. "A novel SVM+ NDA model for classification with an application to face recognition." Pattern Recognition 45.1 (2012): 66-79.

12. Khemchandani, R., and Suresh Chandra. "Twin support vector machines for pattern classification." Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.5 (2007): 905-910.

13. Kumar, M. Arun, and Madan Gopal. "Application of smoothing technique on twin support vector machines." Pattern Recognition Letters 29.13 (2008): 1842-1848.

14. Goldfarb, Donald, and Garud Iyengar. "Robust convex quadratically constrained programs." Mathematical Programming 97.3 (2003): 495-515.

15. Zhong, Ping, and Masao Fukushima. "Second-order cone programming formulations for robust multiclass classification." Neural computation 19.1 (2007): 258-282.

16. Nath, J. Saketha, and Chiranjib Bhattacharyya. "Maximum Margin Classifiers with Specified False Positive and False Negative Error Rates." SDM. 2007.

17. Maldonado, Sebastián, and Julio López. "Alternative second-order cone programming formulations for support vector classification." Information Sciences 268 (2014): 328-341.

18. Carrasco, Miguel, Julio López, and Sebastián Maldonado. "A second-order cone programming formulation for nonparallel hyperplane support vector machine." Expert Systems with Applications 54 (2016): 95-104.

19. Maldonado, Sebastián, Julio López, and Miguel Carrasco. "A second-order cone programming formulation for twin support vector machines." Applied Intelligence (2016): 1-12.

20. Shao, Yuan-Hai, Wei-Jie Chen, and Nai-Yang Deng. "Nonparallel hyperplane support vector machine for binary classification problems." Information Sciences 263 (2014): 22-35.

21. Shao, Yuan-Hai, et al. "Improvements on twin support vector machines. "Neural Networks, IEEE Transactions on 22.6 (2011): 962-968.

22. López, Julio, and Sebastián Maldonado. "Multi-class second-order cone programming support vector machines." Information Sciences 330 (2016): 328-341.

23. Maldonado, Sebastián, and Julio López. "Imbalanced data classification using second-order cone programming support vector machines." Pattern Recognition47.5 (2014): 2070-2079.