Volume 3, Issue 10, October 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Chock-Full Survey on Support Vector Machines

P Indira Priyadarsini
Dept of Computer Science & Engg.
Acharya Nagarjuna University
Guntur,A.P.,India

Nagaraju Devarakonda
Dept of Computer Science & Engg.
Acharya Nagarjuna University
Guntur,A.P.,India

I Ramesh Babu
Dept of Computer Science,&Engg.
Acharya Nagarjuna University
Guntur,A.P.,India

Abstract -Support vector machines (SVMs) are nothing but machines that build support vectors for classification process. SVMs can formulate both linear and non-linear decision boundaries with good generalization ability and they are based on Statistical Learning Theory (SLT).Generally classification using SVMs is same as solving an optimization problem because strength of SVMs lies in solving optimization techniques. At present SVMs have become excellent areas for research which are also powerful tools for most of the machine learning tasks. These optimization problems deal with not only convex problems but also non convex such as semi infinite programming, bi-level programming and integer programming. The goal of this paper is to thoroughly review SVMs from the optimization point of view. Examining the many aspects of SVM optimization problems, it is useful to understand and apply this popular data mining technique more easily. There are two themes given, first theme specifies the SVM models and algorithms based on standard classification till now regarding optimization point of view. The main reason in constructing several optimization problems is to increase the SVM generalize well and reduce over fitting. And then second theme gives the models concerning enhancements to make SVM more accurate and build better, more rapid and easier to understand learning machines. Since research in SVMs and research in optimization problems have become increasingly coupled so present new challenges are systematically explored to construct new optimization models using SVM.

Key words-Support Vector Machines (SVM), generalization, Statistical Learning Theory (SLT), Optimization, machine learning, data mining, over fitting.

1. INTRODUCTION

Support Vector Machines (SVMs) have gained a great deal of attention from a machine learning community. The Support Vector Machine is a powerful, state-of-the-art classification technique introduced in the early1990's [9], [13] .These SVMs are very proficient in classifying both simple and highly complex data and based on structural risk minimization (SRM) principle. They employ sophisticated mathematical principles to avoid over fitting. There is no Support vector machine classification without Statistical Learning Theory (SLT) introduced by Vapnik & Chervonenkis (VC). Optimization techniques are the strength of SVMs. There are three factors that make SVM's more successful namely maximal margin, dual theory and kernel trick. In SVM, optimization methods are the most powerful tools to solve the problems of machine learning with finite training data. It is able to overcome the problems like curse of dimensionality, over fitting. They have numerous elegant features like excellent generalization abilities, pretty mathematical equations, geometrical illustrations and efficient empirical performances.

To use SVMs successfully, requires an understanding of how they work. When training an SVM the practitioner must be able to make decisions based on how to pre-process the data, what kernel to use and setting the parameters of the SVM. Another factor that made SVMs more successful is VC- dimension [92] for a given classification task. Because of its attractive features a large and diverse community works are done on SVMs from optimization, statistics, neural networks, machine learning, functional analysis etc.

Mainly SVMs reduce machine learning problems to optimization problems; especially the convex optimization problems plays vital roles in SVMs.Because convex problems are good in enough algorithmically and theoretically. Many SVM algorithms involve solving convex problems such as convex quadratic programming [66], semi-definite programming [55], second order cone programming [2], and etc. However, in addition to convex there is also non-convex optimization problems appeared in SVMs such as integer or discrete optimization considers non-convex problems with integer constraints, semi-infinite programming [38], and bi-level optimization [7] and so on. Particularly in the process of model construction, these optimization problems may be solved many times. Up to now, SVMs are applied in various areas. They were applied for text categorization [48], face detection, verification and recognition ([51],[61],[89]),Speech recognition [36],Bioinformatics [39],bankruptcy prediction [80],remote sensing image analysis [64], Information and image retrieval ([27], [60], [80]).information security [65], time series forecasting [54] etc.

Basically there are two majors themes in the interaction of SVMs and quadratic Programming. The first theme specifies the models that are based on the optimization point of view for binary classification of SVMs up to now. These are obtained by making changes to the standard SVM (*C*-SVM) ([13],[91]) which are constructed as powerful new models. They are to be precise *v*- SVM [81],Least squares SVM(LSSVM)[45] , Fuzzy SVM [57], Proximal SVM

[31],Twin SVM (TWSVM)[53],Multi kernel SVM [83], Crammer-Singer SVM [20],Cost sensitive SVM [1],AUC maximizing SVM ([3],[11]), robust classification([37],[100]),transductive classification [47], Knowledge based classification([31], [33],), privileged classification [94],Multi-instance classification [62],Multi-label classification [90],multi-view classification [30],and semi supervised and unsupervised classification ([98],[108],[109]),And subsequently the optimization algorithms for SVMs which act as most excellent for present research in any area are given as Sequential Minimal Optimization(SMO) [113],efficient methods for solving large-scale SVM ([15],[42],[50],[52]), parallel methods for solving large-scale SVM [112] and etc.

The second theme provide models regarding enhancements to make SVM more perfect including feature selection ([19],[88]), model selection [8], probabilistic outputs [72], rule extraction from SVMs [63],Leave-One-Out(LOO) error bounds on SVMs [93], concept boundary detection (CBD) method [70].

Examining the many aspects of SVM optimization problems, a systematic survey is necessary and useful to understand and apply this popular data mining technique more easily. The SVMs also perform well and useful for large data sets. The goal of this paper is to thoroughly review SVMs from the optimization point of view. Section 2 of the paper takes standard SVM as an example to summarize and explain the texture of SVMs. Section 3 will describe SVM optimization models with different ranges. Section 4 describes the standard algorithms where many researchers have been proposed, are categorized into three types. Namely Sequential Minimal Optimization (SMO), efficient methods, and Parallel methods. Section 5 describes the Models for better implementation of SVM. Section 6 will provide conclusion.

2. SUPPORT VECTOR MACHINE

Support vector machines are used for classification of data items. Support vector machines are nothing but machines that build support vectors for classification process .There are several methods for finding hyper planes but Support vector machines find an optimal one. These SVM's create hyper planes based on support vectors. These support vectors are decisive points nearer to decision boundary. The hyper plane is taken to separate two classes. There can be any number of hyper planes possible between these two classes. In order to separate the two classes more precisely we choose the hyper plane with larger margin. This is called Maximal marginal hyper plane (MMH) (as shown in Figure:1).In order to identify MMH the decision boundary lines are extended until they touch the closest data points. Accordingly MMH is chosen. The distance stuck between two hyper planes is called Margin of the classifier. Support vectors are points that touch the hyper plane at two ends. Once the support vectors are identified we can easily draw the hyper plane. For constructing large marginal hyper plane several techniques are applied to the data. Mainly quadratic optimization techniques are applied. It is easy to apply numerical data to the SVM idea. The SVM can be used to separate linearly separable data, linearly inseparable case, and non-linearly separable case. In linearly separable case once the support vectors are identified based on these support vectors maximum marginal hyper plane is drawn. In linearly inseparable case slack variables are added to the formulae in order to classify accurately using SVM. In non-linearly separable case kernel functions are introduced to build the model.

The model can be constructed by taking the following initial discussions.

2.1. Linear SVM: Separable case

Model is built based on parameters like lagragian multipliers, hyper plane parameters (w-weight vector, bias - b).Accordingly applied to train the data. This concept is used here to binary classification. The class labels are +1,-1[70].

Generally there are two margins. Both the margins are set in such a way that the hyper plane is maximum. In 3D space the equation is a plane where as in N dimensional case it is a hyper plane. The hyper plane can be constructed based on the line equation wx+b=0. It is called as separating hyper plane .where $W=\{w_1, w_2, \ldots, w_n\}$ is a weight vector and b a scalar (bias).For example 2-D it can be written as $w_0 + w_1 x_1 + w_2 x_2 = 0$.

The tuples that come under class label y=-1 are below the hyper plane .so it can be defined as wx+b<0.The tuples that come under class label y=1 are above the hyper plane .so it can be defined as wx+b>0.If we label all circles as class +1 and all squares as class -1.Then we can guess the class label y for any test sample "t" as if it is a square it should satisfy w.x+b<0 .if it is a circle it should satisfy w.x+b>0.

The two hyperplanes(margins) are expressed as

H1:w.x+b=-1

H2=w.x+b=1

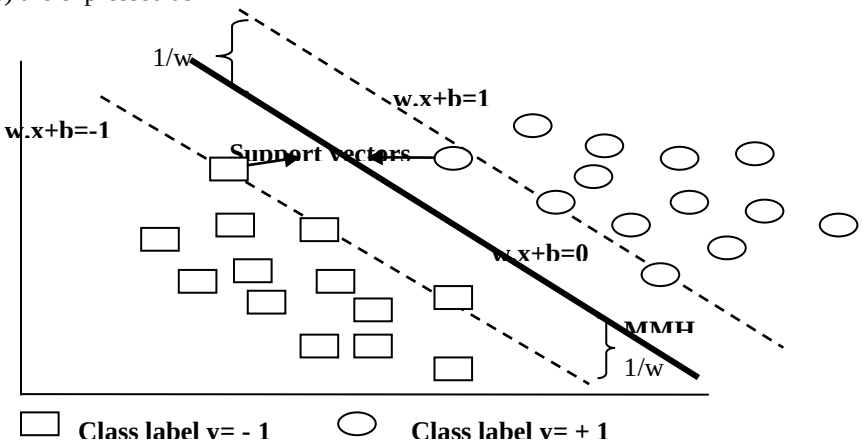


Fig. 1: Graphical representation of Linear SVM

Let data D be $(\mathbf{X}_1, y_1)\ldots (\mathbf{X}_{|D|}, y_{|D|})$, where $\mathbf{X}_i$ is the set of training tuples associated with the class labels y_i [41].

The distance between the two hyperplanes is called margin.This margin can be obtained by adding the distance between two hyperplanes.ie.(1/w+1/w=2/w).

Therefore d= $2/\|w\|$

For obtaining objective function, using the above formula then it is maximizing the margin. However maximizing the margin is same as minimizing the following objective function.

Min. $f(w)=\|w\|^2/2$

Then the constraints are obtained as follows: w,b are chosen in such a way that two conditions are met.

$wx_i+b \geq 1$ if $y_i=1$ --------(2.1)

$wx_i+b \leq -1$ if $y_i=-1$ -------(2.2)

Both inequalities are summarized as

$y_i(wx_i+b \geq 1)$ for i=1,2,3,…….N

Then the equation becomes

min. $f(w)= \|w\|^2/2$ subject to $y_i(wx_i+b \geq 1)$ for i=1,2,3,…….N

This equation can be solved by using langragian multiplier method.

To solve this problem consider the example. Given $f(x_1, x2, ...x_d)$ subject to equality constraints of the form $g_i(x)=0$, i=1,2,--------p

1. Then langragian is defined as $L(x,\lambda)= f(x)-$ where λ_i is a dummy variable called langragian multiplier.
2. $\partial L/\partial x_I = 0 \ \forall i = 1, - - - - d$
3. $\partial L/\partial \lambda_i = 0 \ \forall i = 1, - - - -$p

Solve (d+p) equations in steps 2 and 3 to obtain x and λ

Given Karush Kuhn Tucker(KKT) conditions as

$\lambda_i \geq 0$,

$g_i(x) \geq 0$,

$\lambda_i g_i = 0$,$\forall i$=1,2,….p

It is similar in the case of inequality constraints also.Now the given objective function becomes

$L_p=\frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda i(yi(w.xi + b) - 1)$ ------ (2.3)

Where λ_i is called lagrangian multiplier**.**

To minimize the lagrangian,we must take the derivative of L_p w r.t. w and b

$\frac{\partial Lp}{\partial w} = 0$ implies $w=\sum_i^N \lambda_i y_i x_i$------(2.4)

$\frac{\partial Lp}{\partial b} = 0$ implies $-\sum_{i=1}^{N} \lambda_i y_i = 0$,i.e. $\lambda_i y_i = 0$ ------(2.5)

According to KKT conditions (from above)

$\lambda_i \geq 0$ ------ (2.6)

$\lambda_i (y_i(w. x_i +b) -1) =0$ -------- (2.7)

Substituting equations (2.6) and (2.7) in eq. (2.3**)** we get

$L_D = \sum_i^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i\lambda_j x_i x_j y_i y_j$

Then it becomes,

Maximize $\sum_i^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i\lambda_j x_i x_j y_i y_j$ constrained to $\lambda_i \geq 0 \ \forall$i and $\sum_i^n \lambda_i y_i = 0$

The solution is to find the optimal setting of the Lagrange's multipliers λ_i by maximizing. $\sum_i^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i\lambda_j x_i x_j y_i y_j$

Only λ_i's corresponding to "support vectors" will be non-zero. We can solve Dual Lagrangian (L_D) easily because the parameters are reduced when compared to Primary Lagrangian (L_p).The Dual Lagrangian contain lagrangian multipliers and the training data while the Primary Lagrangian involves lagrangian multipliers as well as w and b(parameters of the decision boundary).And we can say that solutions for both optimization problems are same.

The solution involves to the dual problem:

$w =\Sigma\lambda_i y_i x_i$ and $b= y_k - wx_k$ for any x_k such that $\lambda_k \neq 0$

Each non-zero λ_i indicates that corresponding $\mathbf{x_i}$ is a support vector. Then the classifying function will have the form: $f(x) = \Sigma\lambda_i y_i x_i^{\cdot} x + b$**.** Observe that it relies on an inner product between the test point x and the support vectors x_i .Also keep in mind that solving the optimization problem involved computing the inner products $x\, x_j$ between all pairs of training points.

Once the parameters of the objective function are obtained, a test instance z is classified as follows.

$f(z)=sign(w.z + b) = sign(\Sigma\lambda_i y_i x_i .z+b)$ --------(2.8)

If f(z)= 1 then the test instance is said to be a positive class, otherwise it is said to be negative class.

2.2 Linear SVM: In-separable case

If the data is not separable with a good Maximum Marginal Hyper plane (MMH) then soft margin approach is applied. Since the MMH used in separable case do not effort when the problem arises like when there is a noise or error. An Over fitting error will occur. Then the solution is to consider the width of the margin and the number of training errors to construct the linear decision boundary. While the objective function used in Separable case (Linear SVM) do not support the non separable case because it do not satisfy the inequality constraints. To do this positive valued slack variables (ε_i) are introduced into the constraints of the optimization problem. Assign some points to be moved where they belong, at a cost**.** Let some points be moved to where they belong, at a cost [70].

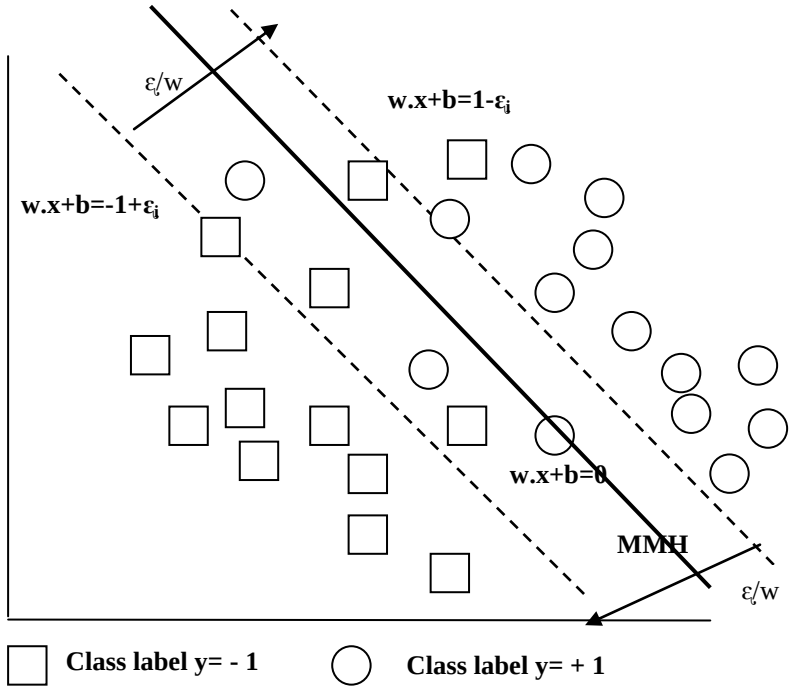


Figure: 2 Decision boundaries shown with soft margin method

$wx+b \geq 1-\varepsilon_i$ if $y_i = 1$

$wx+b \leq -1+\varepsilon_i$ if $y_i = -1$ where $\forall i: \xi i > 0$ --------(2.9)

The modified objective function is $f(w)= \frac{1}{2}\|w\|^2 + c \sum_i^N (\xi)^k$ where c and k are user specified parameters representing the penalty of misclassifying the training instances.(let's assume k=1).The Parameter *C* is taken as a way to control overfitting .

$L_p = \frac{1}{2}\|w\|^2 + C\sum_i^N \varepsilon_i - \sum_{i=1}^N \lambda_i \{ y_i(w. x_i + b) -1+\varepsilon_i\} - \sum_i^N \varepsilon_i \mu_i$ ------ (2.10)

KKT conditions are $\varepsilon_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0$

$\lambda_i \{ y_i(w. x_i + b) -1+\varepsilon_i\} = 0$

$\mu_i \varepsilon_i = 0$

Performing first order derivative of L w.r.t w,b and ε_i to zero

$\frac{\partial L}{\partial wj} = w_j - \sum_{i=1}^N \lambda_i y_i x_{ij} = 0 \rightarrow w_j = \sum_{i=1}^N \lambda_i y_i x_{ij}$ ----- (2.11)

$\frac{\partial L}{\partial b} = -\sum_{i=1}^N \lambda_i y_i = 0 \rightarrow \sum_{i=1}^N \lambda_i y_i = 0$ -------- (2.12)

$\frac{\partial L}{\partial \xi i} = C-\lambda_i-\mu_i = 0 \rightarrow \lambda_i+\mu_i = C$ -------- (2.13)

Substituting these equations (2.11), (2.12) (2.13) in the eq.(2.10),

We get, $L_D = \sum_{i=1}^N \lambda_i - \frac{1}{2}\sum_{i,j}^N \lambda_i\lambda_j x_i x_j y_i y_j$ ---------(2.14)

Where L_D is the dual formulation of Soft margin approach.The slack variables ε_i and lagranges multiplier do not appear in the dual formula. And x_i with nonzero λ_i will be taken as support vectors. Solution for the dual problem is $w = \Sigma\lambda_i y_i x_i$ and $b = y_k(1- \xi_k) - wx_k$ where $k = \text{argmax } \lambda_{k'}$. Then the objective function is given as $f(x) = \Sigma\lambda_i y_i x_i x + b$.

2.3 Non linear SVM

In the case where data is not linearly separable Nonlinear SVM is considered. It means that the datasets have nonlinear decision boundaries. With this non linear SVM curse of dimensionality problem is avoided and optimal decision boundary is set. The idea is to map the original feature space into some higher-dimensional feature space where the training set is separable. The approach is to transform the data from its original coordinate space in x into a new space $\phi(x)$ so that linear decision boundary can be used to separate the instances in the transformed space. After performing the transformation the methodology that was applied in the linear SVM case is applied [70].

The optimization problem is min. $\|w\|^2/2$

subject to $y_i(w.\phi(x_i)+b) \geq 1$, for i=1,2,3,…….N

Following the approach applied in Linear SVM, the dual Lagrangian for the constrained optimization problem is

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \phi(x_i)\, \phi(x_j) \text{ -------(2.15)}$$

Once the λ_i are obtained using quadratic programming, the parameters w and b are derived using the following equations:

$$w=\sum_i \lambda_i y_i \phi(x_i) \text{ -------(2.16)}$$

$$\lambda_i\{ y_i (\sum_j \lambda_i y_j \phi(x_j).\phi(x_i) + b\} = 0 \text{ ---------(2.17)}$$

Finally a test instance z can be classified using the equation:

$f(z)= \text{sign}(w.\phi(z) + b)=\text{sign}(\sum_i \lambda_i y_i \phi(x_i).\phi(z) + b)$ -------(2.18)

Except for the eq.(2.16) the remaining equations contain dot product between pairs of vectors in the transformed space, $\phi(x_i)$, $\phi(x_j)$ .It is a bit cumbersome and it may suffer from curse of dimensionality problem. To avoid that a method called kernel trick is introduced. If every datapoint is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes: $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(x_i) .\varphi(x_j)$.

$Q(\boldsymbol{\lambda}) = \Sigma\lambda_i - \frac{1}{2}\Sigma\lambda_i\lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and

(1) $\Sigma\lambda_i y_i = 0$

(2) $\lambda_i \geq 0$ for all λ_i

A kernel function is a function that is similar to an inner product in some feature space. There are various kernel functions that are applied for transformation.

Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$

Mapping Φ: $x \rightarrow \varphi(x)$, where $\varphi(x)$ is x itself

Polynomial of power *p*: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$

Mapping Φ: $x \rightarrow \varphi(x)$, where $\varphi(x)$ has $\binom{d+p}{p}$ dimensions

Gaussian (radial-basis function): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$

Mapping Φ: $x \rightarrow \varphi(x)$, where $\varphi(x)$ is infinite-dimensional: every point is mapped to a function (a Gaussian); combination of functions for support vectors is the separator [70].

3. OPTIMIZATION MODELS

3.1. *C*-SVM

The standard SVM is developed based on support-vector networks ([13], [91]).The support-vector network is interchangeably called *C*-SVM. It is a learning machine for binary classification problems. In Separable case optimal margin for good generalization was given through identifying support vectors. In Non separable case the soft margin was discussed with minimizing the errors. The Machine's idea is to map non linear input vectors into very high dimension feature space and then separate the data linearly. In this feature space a linear decision boundary is constructed. Special feature of the decision boundary is its high generalization ability of the learning machine. Mainly *C*-SVM was developed to consider non separable data. High generalization ability of support-vector networks is to employ polynomial input transformations. *C*-SVM has done most excellent work at utilizing quadratic optimization problem.

To control the generalization ability of a learning machine one has to control two different factors: the error-rate on the training data and the capacity of the learning machine as measured by its VC-dimension [92]. There exists a bound for the probability of errors on the test set of the following form: with probability $1 - \eta$ the inequality

Pr (test error) < Frequency (training error) + Confidence Interval -------- (3.1)

is valid. In the bound (3.1) the confidence interval depends on the VC-dimension of the learning classifier, the number of elements in the training set, and the value of η .The two factors in (3.1) specify a trade-off: the smaller the VC-dimension, the smaller the confidence interval, but larger the value of the error frequency.

However, when training data is separable we can obtain better generalization by minimizing the confidence term in (3.1) even there are errors in the training set. With the soft margin classifier method this can be done by choosing appropriate values for the parameter C. In this *C*-SVM one can manage the exchange between complexity of decision rule and frequency of error by changing the parameter C, even in the more regular case where there exists no solution with zero error on the training set. As a result, the support-vector network can control both factors for generalization ability of the learning machine. The quadratic programming equations in the previous section are all developed using *C*-SVM itself.

3.2. Least Squares SVM

Least Square Support Vector Machines (LSSVM)[45] are also used to find a hyper plane for separating two classes like a traditional SVM, but using a different primal equation. Its features are sparseness and robustness. It gives new formulations for kernel PCA, Kernel CCA and kernel PLS. It is a kernel based classifier. It uses a spiral classification problem. The advantage of LSSVM is it is taken as dynamic case while the standard SVM is used for static estimation problem. The primal equation is

Min $\frac{1}{2} w^2 + \frac{\gamma}{2} \sum_{k=1}^{N} e^2_k$ ---------- (3.2)

Subject to the equality constraints $y_k [w.\phi(x_k) + b] = 1 - e_k, k = 1,2,\ldots\ldots N$

The minimizing function $1/2 \|w\|^2$ realizes the maximal margin between the straight lines

$(w \cdot x) + b = 1$ and $(w \cdot x) + b = -1$, --------(3.3)

While minimizing $\sum_{i=1}^{N} e_k^2$ making the straight lines (3.3) is proximal to all inputs of positive points and negative points respectively.

Dual formulation to be solved in LSSVM is

Max $-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (k(x_i x_j) + \delta_{ij}/\gamma) + \sum_{i=1}^{n} \alpha_i$ ---------(3.4)

Subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$ where $\delta_{ij} = 1, i=j$

$= 0, i \neq j$

In Standard SVM, the error is measured by the soft margin approach; in turn it depends on the support vectors. While in LSSVM, almost all training points add to the decision function, so the sparseness is reduced. Therefore LSSVM solves linear equations instead of quadratic programming problem. Therefore, it is simpler and faster than traditional SVM.

3.3 Fuzzy support vector machines

A traditional support vector machine (SVM) learns the decision boundary from two distinct classes of the input points. Each input point is treated equally in the traditional SVM. In some cases each input point such as an outlier may not be fully assigned to one of these two classes. This means that some inputs are not exactly decided whether they belong to either positive class or negative class. It may 90% belong to one class and 10% be meaningless, and it may 20% belong to one class and 80% be meaningless. In other words, each data point in the training data is assigned a membership function and reformulates the SVMs such that different input points can make different contributions to the learning of decision boundary. If any data point is detected as an outlier, it is assigned with a lower membership, so its need to total error term decreases. Therefore Fuzzy support vector machines (FSVM) [57] reduce the sensitivity of less important data points.

There is a fuzzy membership $0 < s_i \leq 1$ associated with each training point .This fuzzy membership can be regarded as the attitude of the corresponding training point toward one class in the classification problem and the value can be regarded as the attitude of meaningless. The primal equation is

Minimize $\frac{1}{2} w.w + c \sum_{i=1}^{l} s_i \varepsilon_j$ ----------(3.5)

Subject to $y_i(w.z_i)+b) \geq 1$, $i=1,2,3,\ldots\ldots l$

$\varepsilon_i \geq 0$, $i=1,2,3,\ldots\ldots l$

Where c is a constant. It is noted that a smaller s_i reduces the effect of the parameter in problem (3.5) such that the corresponding point is treated as less important.

Its dual problem is given as a convex quadratic programming

Maximize $W(\alpha) = \sum_{i=1}^{l} \alpha i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i x_j)$ ------(3.6)

Subject to $\sum_{i=1}^{l} \alpha_i y_i = 0, 0 \leq \alpha_i \leq s_i c, i=1\ldots.l$

3.4. Twin support vector machines

Twin support vector machine (TWSVM)[53] is a binary classifier based on the standard SVM classifier,TWSVM solves two smaller quadratic programming problems(QPP) instead of one large QPP. In SVM all data points exists in the constraints but in TWSVM they are distributed such that the patterns of one class determines the constraint of the other QPP and vice-versa. New patterns are assigned to one of the classes depending on its distance to the two hyper planes. Each of the two QPP in the TWSVM pair has the typical formulation of SVM but not all data patterns appear in the constraint of either problem at the same time. The improvements to the TWSVM were made to obtain low

computational complexity [79]. The two non parallel hyper planes are the positive hyper plane and the negative hyper plane:

$(w_+x)+b_+ =0$ -------(3.7)

$(w_-x)+b_- =0$ -------(3.8)

The primal problems for finding these two hyper planes are two convex quadratic programming problems [79].

Min $\frac{1}{2} c_1 (\|w_+\|^2 + b_+^2) + 1/2 \sum_{i=1}^{p} ((w x_i) + b_+)^2 + c_2 \sum_{j=p+1}^{p+q} \rho_j$ ------ (3.9)

w_+, b_+, ρ_-

subject to $(w_+.x_j) + b_+ \leq -1+\varepsilon_j, j=p+1 \ldots\ldots p+q$ ----- (3.10)

$\rho_j \geq 0, j=p+1 \ldots\ldots p+q$ ------ (3.11)

and

Min $\frac{1}{2} c_3 (\|w_-\|^2 + b_-^2) + 1/2 \sum_{i=1}^{p+q} ((w x_i) + b_-)^2 + c_2 \sum_{j=1}^{p} \rho_j$ ------ (3.12)

w_-, b_-, ρ_+

subject to $(w_-.x_j) + b_- \geq 1-\varepsilon_j, \ j=1 \ldots\ldots p$ ------ (3.13)

$\rho_j \geq 0 \ j=1 \ldots\ldots p$ ------ (3.14)

where $x_i, i= 1,...,p$ are positive inputs ,

$x_i, i=p+1,...,p+q$ are negative inputs,

$c1>0, c2>0, c3>0, c4>0$ are parameters,

$\rho_- =(\rho_{p+1}, \ldots\ldots \rho_{p+q})^T$

$\rho_+ =(\rho_1 \ldots\ldots \rho_p)^T$

For both of the primal problems above an interpretation can be offered in the same way.

TWSVM is obtained by solving two dual problems of the above primal problems separately. The simplification of TWSVM has been shown to be significantly better than standard SVM for both linear and nonlinear kernels. It has become more popular in machine learning techniques because of its low computational complexity. On an average, it is about four times faster than the standard SVMs in the training phase.

3.5 Proximal SVM

Proximal Support Vector Machine classification [31] idea is to classify by proximity to planes while the standard SVMs classify by half spaces. It is a fast new support vector machine classifier. It is very simple to implement. There is a classifying plane central between the parallel proximal planes. Thus, PSVM classifies points on the basis of proximity to two parallel planes: $x^1w=\varepsilon_v +1$and $x^1w=\varepsilon_v -1$. And the plane $x^1w-\varepsilon_v =0$ acts as a middle way between and parallel to the above proximal planes. It is a separating plane that roughly separates A+ and A- as given in the figure: 3. PSVM can also be considered as an especially unique case of regularization networks [29].

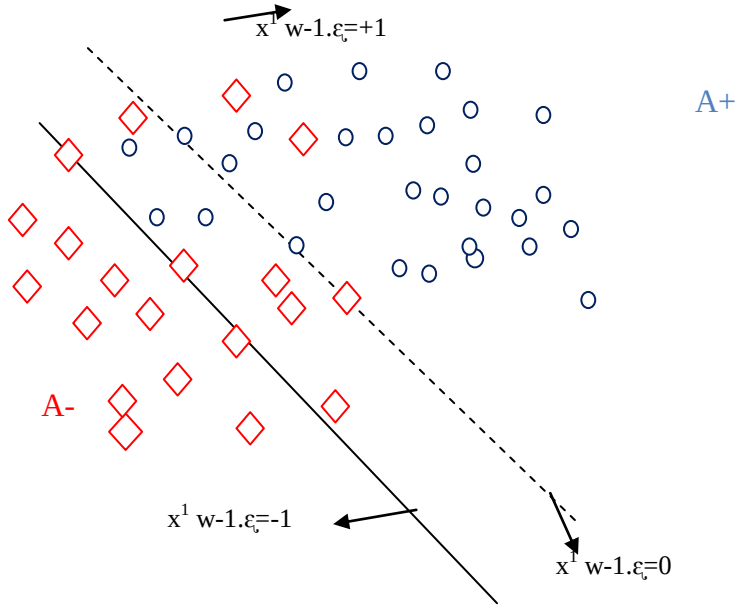


Figure 3: shows the proximal support vector machine classifier: The proximal planes $x^1w - \gamma = \pm 1$ around which points of the sets $A+$ and $A-$ and which are pushed apart by the optimization problem (3.15).

The problem in figure: 3 is considered classifying l points in the n-dimensional real space R^n, represented by the $l \times n$ matrix A, according to membership of each point A_i in the class $A+$ or $A-$ as specified by a given $l \times l$ diagonal matrix D with plus ones or minus ones along its diagonal. For this problem, the proximal support vector machine [31] with a linear kernel is given by the quadratic programming problem with parameter $v > 0$ and linear equality constraint:

Min $\underset{(w,\varepsilon,y)\in R^{n+1+m}}{v/2\ \|y\|^2 + 1/2\ \left\| \begin{pmatrix} w \\ \varepsilon \end{pmatrix} \right\|^2}$ ----------(3.15)

Subjected to $D(A\ w-e\ \varepsilon)+y=e$ ----------(3.16)

Where e is a vector of ones. $\begin{pmatrix} w \\ \varepsilon \end{pmatrix}$ is normal to the proximal planes:

$$\left.\begin{aligned} x^1\ w-1.\varepsilon &= +1 \\ x^1\ w+1.\varepsilon &= -1 \end{aligned}\right\} \quad \text{---------(3.17)}$$

$x^1 w-\varepsilon=0$ --------- (3.18)

These are proximal to points belonging to the sets *A*+ and *A*− respectively. The error variable *y* in (3.16) is a measure of the distance from the plane $x\_w - 1\cdot\varepsilon = +1$ of points of class *A*+ points and from the plane $x\_w - 1.\varepsilon = -1$ of points of class *A*−. The approximate separating plane (3.18) as depicted in figure 1, acts as a linear classifier as follows:

$x^1 w-\varepsilon$ { >0 then xЄA+ ; <0 then xЄA- ; =0 then xЄA+ or xЄA-

3.6. *V*-SVM

The ν-support vector machine (ν-SVM) [81] for classification has the advantage of using a parameter ν on controlling the number of support vectors. The soft margin SVM's and *nu* SVM are equivalent with the ν-SVM .The ν-SVM is a soft margin SVM which is implemented as a quadratic model and it is able to improve the generalization of a SVM on datasets that contain outliers. It has given a different formulation for solving SVMs in which the parameter c is transformed to a meaningful parameter *v*, that approximately represent the part of support vectors.

The soft margin SVMs [70] discussed previously, have soft margins that can take any positive value. It leads to the difficulty in testing because the range of soft margins has to be known previously. The ν - SVM specifies that the soft margin lies between the range of zero and one. The parameter ν is not controlling the switch between the training error and the generalization error. It is taken as an upper bound on the fraction of margin errors and is the lower bound on the fraction of support vectors.

The primal equation is

Min $\frac{1}{2}\ w^T w- v\rho + 1/l \sum_{i=1}^{l} \epsilon_i$ ------- (3.19)

Subjected to $y_i\ (w^T \phi(x_i) +b)\geq\rho-\epsilon_i$

$\epsilon i \geq 0$,i=1,2,----l , $\rho\geq 0$

The dual equation is Min $\frac{1}{2}\ Q\alpha$ ---------- (3.20)

Subjected to $y^T \alpha, e^T\alpha \geq v$

Where e is the vector of all ones ,Q is a positive semidefinite matrix $Q_{ij}=y_i y_j K(x_i,x_j)$ and $K(x_i,x_j) = \phi(x_i)^T\phi(x_j)$ is the kernel.

3.7 Tranductive Support vector machines

The Transductive support vector machine (TSVM) [47] presented a learning task from small training model by taking a transductive, as an alternative of an inductive approach. Generally labelled instances are more difficult, time consuming to obtain and expensive, where unlabeled instances are easy to collect. Standard SVMs use only labeled training samples, so using large amount of unlabeled data many semi-supervised SVMs (S^3 VM) were developed. TSVM is a proficient method for finding labelling of the unlabeled data, so that maximizing margin is built between both the original labelled and unlabeled data. Mainly transduction means labelling a test set. The decision function gives the smallest generalization error bound on unlabeled data.

At the earlier, TSVM algorithms are incapable of dealing with a huge number of unlabeled samples. The first one is [6], used integer programming which yielded difficulty in handling large problems. Later [47] proposed which is based on combinatorial optimization problem for training the data. This gave rise to famous and widely used software, SVM^{light} which was used for thousands of examples. The key idea of the algorithm is that it begins with a labelling of the test data based on the classification of an inductive SVM. Then it upgrades the solution by switching the labels of test examples so that the objective function will decrease.

Given a training set, $T=\{(x_1,y_1),\ldots. (x_l,y_l)\}\cup\{(x_{l+1},\ldots\ldots x_{l+q}\}$ -------- (3.21)

where $x_i \in R^n$, $y_i \in \{-1,+1\}$,i=1,…..*l* , $x_i \in R^n$,i=*l*+1,…..*l*+q and the set $\{(x_{l+1},\ldots\ldots x_{l+q}\}$ is a collection of unlabeled inputs. The primal problem in TSVM is built as the following (partly) combinatorial optimization problem

$$\min_{w,b,\xi,y^*} \quad \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=1}^{l}\xi_i^* \qquad \text{-------(3.22)}$$

subjected to $y_i((w.x_i)+b) \geq 1-\xi_i,\ i=1,\ldots..l$ -------(3.23)

$$y_i^*((w.x_i^*)+b) \geq 1-\xi_i,\ i=l+1,\ldots\ldots l+q \qquad \text{-------(3.24)}$$

$$\xi_i \geq 0, i=1\ldots..l \qquad \text{-------(3.25)}$$

$$\xi_i^* \geq 0, i=l+1,\ldots..l+q \qquad \text{-------(3.26)}$$

Where $y^* = (y_{l+1}^*, \ldots y_{l+q}^*), C>0, C^*>0$ are parameters. But finding the exact solution to this problem is NP-hard. Major effort has focused on efficient approximation algorithms. Among these approximation algorithms some relax the TSVM training problem with semi-definite programming (SDP) [98],[108],[109]).

3.8 Knowledge based SVM

In Knowledge based SVMs [32] we are given not only the training set but also some prior knowledge such as advised classification rules. Using the prior knowledge, we can improve the predictive training accuracy of learning algorithms or reduce the amount of training data needed.

Now the problem can be given in the following way: the single input points in the training points are now considered as input sets, called knowledge sets. The basic principle of their approach is that in the case of a polyhedral set of constraints it is possible to formulate constrained optimization problems such that in addition to finding a large margin solution the estimate also satisfies the prior knowledge constraints. Later on [62] studied a reformulation of linear and nonlinear support vector classifiers that can contain prior knowledge in the form of various polyhedral sets, each belonging to one of two groups.

Like classical SVMs, they learn a linear classifier ($w'x = b$) given data $(x^t, y_t)_{t=1}^T$ with $x^t \in R^n$ and labels $y_t \in \{\pm 1\}$. In addition, they are also given prior knowledge specified as follows: all points that satisfy constraints of the polyhedral set $D^1x \leq d^1$ belong to class +1. That is, the advice specifies that $\forall x, D^1x \leq d^1 \Rightarrow w'x - b \geq 0$. Advice can also be given about the other class using a second set of constraints: $\forall x, D_2x \leq d2 \Rightarrow w'x - b \leq 0$. Combining both cases using advice labels, $z = \pm 1$, advice is

Given by specifying (D, d, z), which denotes the implication

$Dx \leq d \Rightarrow z(w'x - b) \geq 0.$ ---------- (3.27)

We assume that m advice sets $(D_i, d^i, y_i)_{i=1}^m$ are given in addition to the data,

And if the i-th advice set has l_i constraints, we have $D_i \in R^{li \times n}$, $d_i \in R^{li}$ and $y_i = \pm 1$.

Given a Training set $T=\{(\chi_1,y_1),\ldots(\chi_p,y_p),(\chi_{p+1},y_{p+1}),\ldots(\chi_{p+q},y_{p+q})\}$----- (3.28)

where χ_i is a polyhedral in R_n defined by $\chi_i=\{x|\ D_ix \leq d_i\}$ $D_i \in R^{li \times n}$, $d_i \in R^{li}$ and $y_1 = \ldots = y_p = 1$, $y_{p+1} = \ldots = y_{p+q} = -1$.Then find a real valued function g(x) in R_n such that the value of y for any x can be predicted by the decision function.

$$f(x)=\text{sgn}(g(x)) \qquad \text{------- (3.29)}$$

The primal problem to be the following semi infinite programming problem

$$\min_{w,b,\xi} \quad \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{p+q}\xi_i \qquad \text{-------- (3.30)}$$

subjected to $(w.x)+b \geq 1$,for $x \epsilon \chi_i$.$i=1,\ldots\ldots p,$ -------(3.31)

$(w.x)+b \leq -1$,for $x \epsilon \chi_i$.$i=p+1,\ldots\ldots p+q$ -------(3.32)

$$\xi_i \geq 0, i=1,\ldots.. \quad p+q \qquad \text{-------- (3.33)}$$

3.9 Robust Classification

In traditional SVM, the parameters obtained are assumed to be known completely in optimization problems. But there can be quandaries in the training data because it may contain measurement errors. The solution to the optimization problem is to consider discriminants robust to measurement noise. The Robust SVM (RSVM) ([37], [100]) makes use of semi definite programming and considers uncertainty sets that allow this class of problems to be formulated as second order cone programming (SOCP). An important factor in robust optimization problem is how to choose the parameters that define the uncertainty structures. It is mainly concerned with an ellipsoidal model of uncertainty for classifying noisy data [16]. For example, when the training data is subjected to measurement errors, it is considered with uncertainty sets $X_i \in R_n$, $i = 1\ldots l$.Therefore the standard problem becomes to be the following robust classification problem.

Given a training set $T = \{(X_1,Y_1),\ldots,(X_l,Y_l)\}$, where X_i is a set in R_n , $Y_i \in \{-1,1\}$. Obtain a real function $g(x)$ in R_n , such that the value of y for any x can be predicted by the decision function

$$f(x) = \text{sgn}(g(x)) \qquad \text{------(3.34)}$$

The figure 4 traced out shows the robust problem with circle perturbations where the circles with "+" and "-" are positive and negative input sets respectively, the maximal marginal hyper plane $(w^*\cdot x)+b^*=0$ constructed by robust SVM (RSVM). The primal problem for RSVM for such case is a semi-infinite programming problem

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \qquad \text{-------(3.35)}$$

where C is a penalty for misclassification

$$\text{subjected to } y_i(w.(x_i.r_i.u_i))+b) \geq 1-\xi_i, \forall \|ui\| \leq 1, i = 1, \ldots . l \qquad \text{------(3.36)}$$

$$\xi_i \geq 0, i=1,\ldots . l \qquad \text{-------(3.37)}$$

where the set X_i is a supers phere obtained from deviation of a point x_i

$$X_i = \{x \mid \|x - x_{i|}\| \leq r_i\}. \qquad \text{------(3.38)}$$

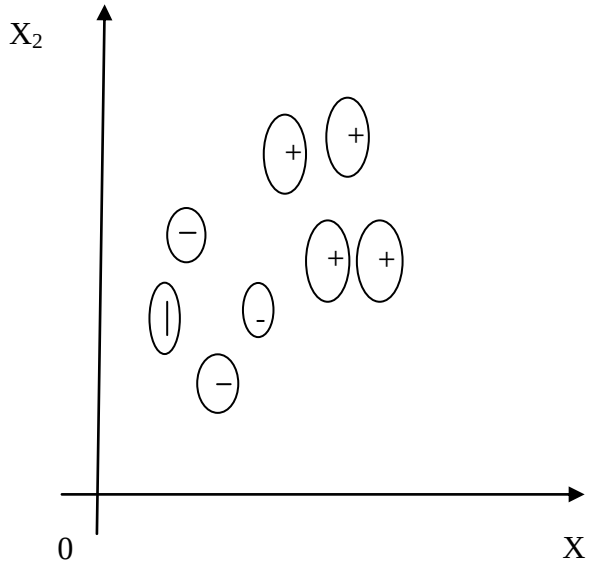


Fig: 4 Pictorial interpretation of robust classification problem

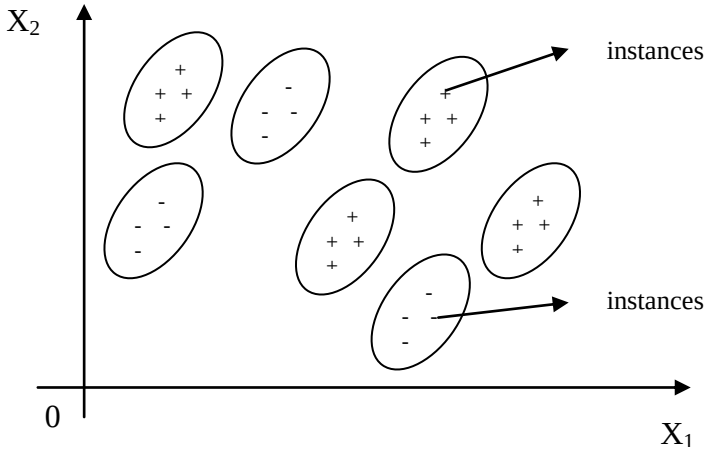
3.10 Multi instance classification

Multi-instance classification [62] is given using a non integer real space subjected to linear and bilinear constraints. It mainly used for drug activity prediction. Here an efficient linearization algorithm is proposed through which a local solution is obtained by solving a sequence of fast linear programs in a few steps. A typical feature of linear classifier is the sparse number of features it utilizes. It is similar to both the robust and knowledge-based classification problems; it can be formulated as follows.

Suppose that there is a training set $T = \{(X_1,Y_1),\ldots,(X_l,Y_l)\}$, where $X_i = \{x_{i1},\ldots\ldots,x_{ili}\}$, $x_{ij} \in R^n$ $j= 1,\ldots,l_i$, $Y_l \in \{-1,1\}$. Find a real function $g(x)$ in R_n, such that the label y for any instance x can be predicted by the decision function

$f(x) = \text{sgn}(g(x))$. ------- (3.39)

The set X_i is known as a bag containing a number of instances. The attractive point of this problem is that the label of a bag is related with the labels of the instances in the bag and determined by the following way. An unseen bag is positive if and only if there is at least one instance in the bag is positive side of the decision surface and a bag is negative if and only if all instances in the bag are negative side of the decision surface. A pictorial representation of multi-instance classification problem is shown in Figure 5, where every enclosure represents a bag; a bag with "+" is positive and a bag with "-" is negative.



"+" represent positive bag "-" represent negative bag

Figure: 5 Pictorial representation of multi-instance classification problem

For a linear classifier, a positive bag is classified correctly if and only if some convex combination of points in the bag lies on the positive side of a separating plane. Therefore the primal formulation in the multi-instance SVM (MISVM) is con

min ½ ‖w‖² +C1 $\sum_{i=1}^{r}$ ζi +C2 $\sum_{i=r+1}^{p}$ ζi ---------(3.40)
w,b,v,ξ

subjected to (w.$\sum_{j\epsilon I(i)}$ v_j^i (x_j)+b≥1-ξ_i, i=1,…p ---------(3.41)

(w.(x_i)+b≤-1+ξ_i, i=1,…r+1,…….r+s --------(3.42)

ξ_i≥0, i=1,…p,r+1,…….r + s , -------- (3.43)

v_j^i ≥ 0,jϵ I(i),i=1,….p --------(3.44)

$\sum_{j\epsilon I(i)}$ v_j^i=1,i=1,…..p --------(3.45)

Where r and s are number of instances in all positive bags and all negative bags respectively and p is the number of positive bags.

Though the above problem is nonlinear, it is easy to say that among all its constraints, only the first one is nonlinear. Then a local solution to this problem has got by solving a series of fast linear programs in a few iterations. Alternatively, hold one set of variables which make the bilinear terms constant while varying the other set. For a nonlinear classifier, a similar statement can be obtained to the higher dimensional space by the kernel function.

3.11 Multi view classification

In improving the performance of SVM, a pre-processing step Kernel Canonical Correlation Analysis (KCCA) is combined with standard SVM.This gave rise to Multi view SVM [30].The technique is to first reduce the dimensions, then train SVM classifiers and then combine the two steps together.

The same set of objects can be described in multiple different views. Here it is considered two views. Features are naturally separated into *K* sets. So the features are split into two sets. The two views are redundant but not completely correlated.

The KCCA algorithm considers the two feature spaces such that when the training data is projected onto those directions the two vectors (one for each view) of values obtained are maximally correlated. Making use of the Rademacher complexity theory [4], the generalization error bound of the supervised SVM-2K algorithm is analyzed**.**

A traditional SVM is considered as a one-dimensional projection followed by thresholding, while Multi view SVM combines the two steps by introducing the constraint of similarity between two one-dimensional projections identifying two distinct SVMs one in each of the two feature spaces. The extra constraint is chosen slight differently from the 2-norm that characterizes KCCA. Then an ϵ-insensitive 1-norm is used to measure the amount by which points fail to meet ϵ similarity:

$$| <w_A,\phi_A(x_i)>+b_A-<w_B,\phi_B(x_i)>-b_B | \leq \eta_i + \epsilon \quad \text{--------(3.46)}$$

Where w_A,b_A(w_B,b_B) are the weight and threshold of the first(second) SVM. Combining this constraint with the usual 1-norm SVM constraints and allowing different regularization constants gives the following optimization function.

$$\min L=1/2 \| w_A\|^2 +1/2 \| w_B\|^2+ C^A\sum_{i=1}^{l} \xi^A+ C^B\sum_{i=1}^{l} \xi^B +D\sum_{i=1}^{l} \eta_i$$

such that $| <w_A,\phi_A(x_i)>+b_A-<w_B,\phi_B(x_i)>-b_B | \leq \eta_i + \epsilon$ --------- (3.47)

y_i ($<w_A,\phi_A(x_i)>+b_A$)≥ 1-ξ_i^A ---------(3.48)

y_i ($<w_B,\phi_B(x_i)>+b_B$) ≥ 1-ξ_i^B ---------(3.49)

ξ_i^A≥ 0, ξ_i^B≥ 0, η_i≥ 0 for all 1≤i≤*l* ---------(3.50)

Let w_A, $b_A w_B$, b_B be the solution to this optimization problem. The final SVM 2K decision function is then

$h(x)= sign(f(x))$,where ----------(3.51)

$f(x)=0.5(<w_A,\phi_A(x_i)>+b_A+<w_B,\phi_B(x_i)>+b_B)$

$=0.5(f_A(x)+ f_B(x))$ ----------(3.52)

SVM-2K was later extended to multi-view semi-supervised learning (Szedmak et al.2007).

4. STANDARD ALGORITHMS

4.1 SMO algorithm

The Sequential Minimal Optimization algorithm (SMO) is an efficient training algorithm for SVMs. It serves as a classical algorithm for training SVMs .The main agenda in solving the Quadratic Programming (QP) problem is to speed up the training of SVM. The SMO [113] is able to find efficient solution to the QP problem. The solution is to split a large QP problem into an order of smallest possible QP sub problems and solving the smallest possible optimization problem, involving two Lagrange multipliers, at each step. And therefore the amount of memory required for computation is reduced.SMO exhibit good performance for linear SVMs because the evaluation of linear SVM is a single dot product rather than a summation of linear kernels. SMO, it does excellent job for SVMs with sparse inputs, even for non-linear SVMs, because the kernel computation time can be minimized. It can be considered a special case of Osuna's algorithm [60].

When compared to other methods, SMO has proven best in terms of speed. The similar methods are projected conjugated gradient chunking algorithm [12] and Osuna's algorithm [60]. A recent improvement in this direction is an online learning method called LASVM [113]. LASVM is an approximate SVM solver that uses online approximation. It can achieve accuracy similar to that of a SMO after performing a single sequential pass through the training examples. Further benefits can be achieved using selective sampling techniques to choose which example should be considered next.

The algorithm is as follows:

Step 1:Input: C, kernel, kernel parameters, epsilon

Step 2: Initialize b and all λ's to 0

Step 3:Repeat until KKT is satisfied (to within epsilon):

(i) Find an example *e1* that violates KKT (prefer unbound examples here, choose randomly among those)

(ii)Choose a second example *e2*. Prefer one to maximize step size (in practice, faster to just maximize $|E_1 – E_2|$). If that fails to result in change, randomly choose unbound example. If that fails, randomly choose example. If that fails, re-choose e1.

(iii)Update α_1 and α_2 in one step

(iv)Compute new threshold b

SMO mainly concerned with solving the smallest possible optimization problem at every step. For the standard SVM QP problem, it involves two Lagrange multipliers, because the Lagrange multipliers must satisfy a linear equality constraint. At every step, SMO chooses two Lagrange multipliers to collectively optimize, finds the optimal values for these multipliers, and updates the SVM to copy the new optimal values.

SMO does not require any extra matrix storage. Therefore, very large SVM training problems can be saved inside of the memory of an ordinary personal computer. Because no matrix algorithms are used in SMO, it is less exposed to numerical precision problems. There are mainly two methods in SMO: an analytic method for solving for the two Lagrange multipliers, and a heuristic for choosing which multipliers to optimize. The SMO algorithm is slow for non-linear SVMs than linear SVMs, because the time is subjected by the evaluation of the SVM.

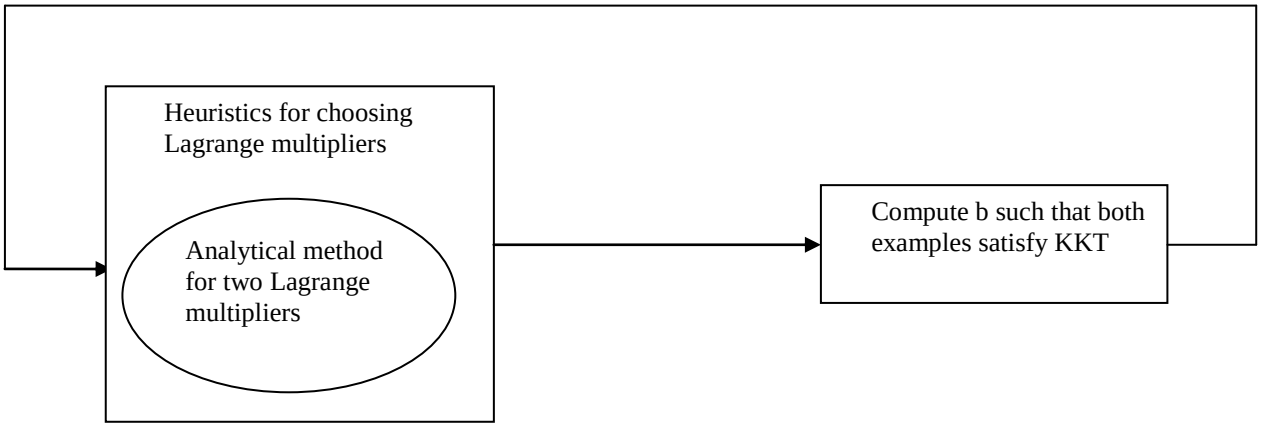


Fig 6: Do this until entire training set satisfies the KKT conditions

The method can be combined with an active selection of training samples to yield faster training, higher accuracy rates, and simple models. Another improvement in this direction is a method called maximum-gain working set selection [115] which is considerably faster than SMO on large training sets.

An outstanding characteristic of SMO is that it is easy to implement and does not require a QP Library.Comparative testing against other algorithms done by Platt shown that SMO is drastically faster and has better scaling properties.

4.2 Efficient methods for large Scale SVMs

The methods given here are efficient for large scale SVM's where other methods degrade. Recently a coordinate descent method for solving primal L2-SVM [15] is proposed. According to the survey on related algorithms, show that their approach obtain a useful model more quickly. Coordinate descent, a popular optimization technique, updates one variable at a time by minimizing a single-variable sub-problem. If one can efficiently solve this sub-problem, then it can be a competitive optimization method. Due to the non-differentiability of the primal L1-SVM, Chang et al's work is restricted to L2-SVM. Moreover, as primal L2-SVM is differentiable but not twice differentiable, certain considerations are needed in solving the single-variable sub-problem.

The other one is a simple Cutting-Plane Algorithm [50] for training linear SVMs that is shown to converge in time O(sn) for classification. It is based on an alternative formulation of the SVM optimization problem that exhibits a different form of sparsity compared to the conventional formulation. The algorithm is empirically very fast and has an intuitively meaningful stopping criterion. The algorithm opens several areas for research. Since it takes only a small number of sequential iterations through the data, it is promising for parallel implementations using out-of-core memory. Also, the algorithm can in principle be applied to SVMs with Kernels. While a straightforward implementation is slower by a factor of n, matrix approximation techniques and the use of sampling might overcome this problem.

Another algorithm which is more efficient for large linear SVM was given namely Dual coordinate decent method [42].It is very simple to implement, and possesses sound optimization properties. Experiments show that this method is faster than state of the art implementations. It presents a novel dual coordinate descent method for linear SVM with L1- and L2- loss functions. This method is simple and reaches an ϵ-accurate solution in O (log (1/ϵ)) iterations. Experiments indicate that this method is much faster than state of the art solvers such as Pegasos [78], TRON [59], SVM^{perf} [50] and a recent primal coordinate descent implementation. SVM^{perf} uses a cutting plane technique. And other one [82] apply bundle methods, and view SVM^{perf} as a special case. A trust region Newton method TRON) [59] is proposed for logistic regression and L2-SVM. These algorithms focus on different aspects of the training speed.

The other approach in [52] presents a fast dual method for the SVM training. The main idea is to sequentially traverse through the training set and optimize the dual variables associated with one example at a time. The speed of training is improved by shrinking and cooling heuristics. Experiments indicate that this method is much faster than current solvers such as cutting plane, bundle and exponentiated gradient methods.

4.3 Parallel methods

Mainly for training large training sets, SMO and Chunking [116] become infeasible. To overcome this difficulty a large number of algorithms were developed. The Zanghirati and Zanni [112] achieved an efficient sub problem solution by a gradient projection-type method. Their idea is to box constraints and to a single linear equality constraint. Thus, approaches based on explicit storage of the matrix of the quadratic form are not practicable. It is a parallelizable approach that splits the problem into a sequence of smaller quadratic programming sub problems. These sub problems are solved by a variable projection method that is well suited to a parallel implementation and is very effective in the case of Gaussian support vector machines. Then it achieves good trade-off between convergence rate and cost per iteration. By using this method as inner solver, an implementation that works efficiently with sub problems to produce few iterations of the decomposition scheme is developed. This was the first approach suited for an effective implementation on multiprocessor systems. Of course, several issues must be addressed to achieve good performance, such as limiting the overhead for kernel evaluations and choosing a suitable inner QP solver.

The promising results given by this parallel scheme can be further given credit to some more studies on both the gradient projection QP solvers ([76], [22]) and the selection rules for large sized working sets [77].Therefore on the basis of these studies a new parallel gradient projection-based decomposition technique (PGPDT) [107] is developed and implemented in software.

5. MODELS FOR BETTER IMPLEMENTATION

5.1 Probabilistic outputs for support vector machine

The decision function that is used in the standard SVM is used to predict the label of any test input *x* in a binary classification. But there is no guarantee that the estimation is absolutely correct. Sometimes it is necessary to know how much confidence we have, i.e probability of the input x belonging to the positive class. So the value of g(x) is used to estimate the probability $P(y = 1|g(x))$ of the input *x* which belongs to the positive class. Here $g(x)$ is given with the probability with interval between [0, 1] namely the sigmoid function. [72].Then the SVM output transform to a posterior probability of a class using a sigmoid function.

$$P(g)=1/1+\exp(Ag+B) \qquad \text{-------(5.1)}$$

Where A and B are parameters to be determined. By using sigmoid function it is clear that good posterior estimates are produced. While using sigmoid function one can use a maximum likelihood method on a training set (f_i,y_i). It can be given as

$$\text{Max} \prod_{y_i=1} P_i \prod_{y_i=-1} (1- P_i) \qquad \text{-------(5.2)}$$

$$\text{where } P_i(c_1,c_2)=1/1+\exp(c_1(g(x_i))+c_2),\ i=1,\ldots\ldots l \qquad \text{---------(5.3)}$$

This is a two-parameter maximization problem; therefore it can be performed using any number of optimization algorithms. An improved algorithm for better implementation of solving problem (5.2), also meet and avoid numerical difficulties was also given [58].

5.2 Model selection

The initial work in model selection with cross validation [17], [28]) involves solving a convex optimization problem with one or more hyper parameters: the penalty parameter C and any kernel parameters with lowest test error. While cross validation is frequently in use for selecting these parameters, its implementation by a grid-search technique is used in the parameter space effectively in a model. It is simple to implement and wide spread. Its limitation is high computation exponential due to the combinatorial explosion of grid points in high dimensions. In order to remove the drawbacks in the above approach, bi level optimization was developed.

The second one bi-level optimization [7], that provides a systematic search of the hyper parameters. It is based on support vector based training. Specifically, the model is optimized with respect to the ε-insensitive loss function with 2-norm regularization and with a trade-off parameter C. There are two levels in solving bi level optimization with the outer-level leader problem selects the hyper parameters γ to perform well on a validation set and the follower problem trains an optimal inner-level model for the given hyper parameters and returns a weight vector w for validation as shown in Figure 7.

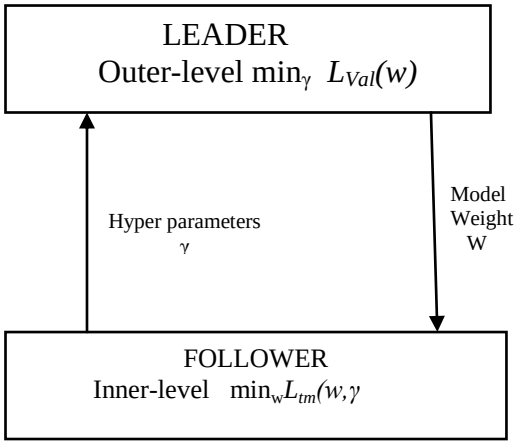


Figure 7: Bi level optimization

The inner-level problems minimize the regularized training error to decide the best function for the given hyper parameters for each fold. The hyper parameters are the outer-level control variables. The objective of the outer-level is to minimize the validation error based on the optimal parameters (***w***) returned for each fold. Therefore the bi level programming approach offers a ample framework in which new regularization methods can be developed, valid bounds on the test set errors can be achieved, and most significantly, improved model selection can be produced.

5.3 LOO error bounds on SVMs

Leave-one-out (LOO) method [93] is the brawny approach for support vector machine model selection. Minimizing bounds of Leave-one-out errors is an almost unbiased estimate of the generalization error. The success of SVMs depends on modifying the several parameters which affect the generalization error. The approach is to estimate the generalization error and then search for parameters so that this estimator is minimized. This requires that the estimators are computationally efficient. However, one limitation of the LOO method is that it needs more time. An effective idea is to approximate the LOO error by its upper bound, that is computed by running a actual classification algorithm only once on the original training set *T* of size *l*. This approach has successfully been developed for both support vector machine for classification [40],[43],[49],[93], support vector machine for regression ([14],[87]) and support vector ordinal regression [101]. Then we can search for parameter so that this upper bound is minimized.

These bounds which are tighter than the one defined in Vapnik [91] and valid for hyper planes not necessarily passing through the origin depend on a new concept called the span of support vectors. The bounds obtained show that the generalization ability of SVM depends on more complex geometrical constructions than large margin.

This procedure is usually used to estimate the probability of test error of a learning algorithm. The leave-one-out procedure consists of removing from the training data one element, building the decision rule on the basis of the remaining training data and then testing the removed element. In this way one tests all ℓ elements of the training data (using ℓ different decision rules). The number of errors in the leave-one-out procedure is denoted by $L(x_1, y_1, ..., x_\ell, y_\ell)$. It is known that the leave-one-out procedure gives an almost unbiased estimate of the probability of test error : the expectation of test error for the machine trained on $\ell - 1$ examples is equal to the expectation of $1/\ell\ L(x_1, y_1, ..., x_\ell, y_\ell)$. Leave-one-out procedure is conducted only for support vectors where as non support vectors will be recognized correctly since removing a point which is not support vector does not change the decision function. The major idea lies in the fact that the upper bounds on the number of errors is made by span of support vectors. This will depend on the span of the support vectors, which gives vigorous bounds that depending on the diameter of the training points.

Later on, provoked by the Leave-one-out error bound, approaches were proposed by directly minimizing the expression given by the bound in an attempt to minimize leave-one-out error([87],[96]) and these approaches are called LOO support vector machines (LOOSVM). LOOSVMs mainly involve solving convex optimization problems, and one of which is a linear programming problem.

$$\text{Min}_{\alpha,\xi} \sum_{i=1}^{l} \xi_i \quad \text{----------(5.4)}$$

subjected to $y_i f(x_i)) \geq 1-\xi_i+\alpha_i K(x_i,x_j), i=1,\ldots l$ --------(5.5)

$\alpha_i \geq 0, \xi_i \geq 0, i=1,\ldots l$ --------(5.6)

where $f(x)=\text{sign}(\sum_{i=1}^{l} \alpha i \; y_i \; K(x_i,x_j))$ --------(5.7)

and $K(x,x^1)$ is the kernel function.

LOOSVMs possess many of the same properties as standard SVMs. The main Prominence of these algorithms is that apart from the choice of kernel, they are parameter less. The selection of the number of training errors is inherent in the algorithms and not chosen by an extra free parameter as in SVMs.

5.4 Using Feature selection in SVMs

It is indispensable to build SVMs which are better and easier to understand learning machines even with thousands to millions of low level features**.** Selecting the most relevant one is the concept of feature selection. In many supervised learning problems feature selection [19] is an important task for a variety of reasons: increasing generalization performance, shorter training time requirements, and improved model interpretability. When we consider Standard SVMs there is no guarantee in getting important features which contribute to classification process. By using feature selection we can identify subset of features. So feature selection when combined with standard SVM we can achieve classification with more perfectness. Generally there are several feature selection strategies used namely filters: ranks features or feature subsets independently of the classifier, Wrapper method**:** uses a classifier to assess features or feature subsets, like F-score, Random forest, etc. The main judging criterion is the balanced error rate (BER). It is given as

$$\text{BER}= 1/2 \left[\frac{\#\text{positive instances predicted wrong}}{\#\text{positive instances}} \right] + \left[\frac{\#\text{negative instances predicted wrong}}{\#\text{negative instances}} \right]$$

Suppose assume a test data set contains 90 positive and 10 negative instances. If all instances are predicted as positive, then BER is 50% since the first term of above formula is 0/90 but the second is 10/10. There are other judging criteria such as the number of features and probes; mainly the smallest BER is used.

In machine learning, there are several views for feature selection in SVM [10],[39],[56],[96],[111]).Some of them are applied to optimization problems like l_0 norm, $l_1$norm [110], l_∞ norm SVM and achieved good performance.

Naturally, we expect that using the l_p -norm ($0 < p <1$) in SVM can find more sparse solution than using l_1 -norm and more algorithmic advantages. Through combining standard SVM and feature selection strategy by introducing the l_p-norm ($0 < p <1$) , the primal problem in l_p-support vector machines (l_p -SVM) is [26].

$$\min_{w,b,v,\xi} \sum_{i=1}^{n} v_i^{\,p}, \; +C \sum_{i=1}^{l} \xi_i \quad \text{----------} \; (5.8)$$

subjected to $y_i((w.x_i) +b) \geq 1-\xi_i$ ---------- (5.9)

$\xi_i \geq 0, i=1,\ldots\ldots l$ ---------- (5.10)

$-v \leq w \leq v$ ---------- (5.11)

This problem can be solved by using successive approximation algorithm ([10],[26]).

5.5 Rule extraction from support vector machines

Although SVM is a state-of-the-art classification technique in data mining, they are as strong as their weak, as the non linear models are considered as incomprehensible black box models.Then opening the black box by extracting rules from SVM models that replicate their activities as they are validated before being implemented. They became more useful in medical diagnosis [63].Rule extraction is performed for the following two reasons: (1) to understand the classifications made by the underlying non-linear black-box model, thus to open up the black box and (2) to improve the performance of rule induction techniques by removing idiosyncrasies in the data. The classification scheme for rule extraction techniques that can easily be extended to SVMs is based on the following criteria: Translucency, Expressive power, Specialized training regime, Quality, Algorithmic complexity.

Two main kinds of rule extraction methods are known as pedagogical [75] and decompositional ([34],[67]). Pedagogical techniques are those which consider the trained model as a black box and directly extract rules which relate the inputs and outputs of the SVMs. On the other hand, decompositional approach is closely related to the internal mechanism of the SVMs and their created hyper plane.

An algorithm was presented which extract propositional classification rules from linear classifiers [34].The method is considered to be decompositional because it is only applicable when the underlying model provides a linear decision boundary. The algorithm is iterative and extracts the rules by solving a constrained optimization problem that is computationally inexpensive to solve. While the mathematical details are relatively complex and the principal idea is

rather straightforward to explain. Therefore different optimal rules extracted according to different criteria will maximize the log volume of the region, which lead to solving the following optimization problem

$$\max \prod_{yi=1} x_i \quad \text{-------(5.12)}$$

$$x \epsilon R^n$$

$$\text{subjected to } (w.x) + b = 0, \quad \text{-------(5.13)}$$

$$0 \leq x \leq 1 \quad \text{------- (5.14)}$$

However, there are limitations for existing rule extracted algorithms in real applications with high dimensions. So the integration of the feature selection into the rule extraction problems is another possibility to be explored, and there are previously some papers related to this topic [102].

5.6. Concept Boundary detection

The main idea in concept boundary detection (CBD) method [69] is to reduce the dimensionality of the dataset and consider only support vectors and discard non support vectors. This can be done by K nearest neighbour method. To reduce the high dimensional datasets by avoiding curse of dimensionality problem is the key feature.

To speed up processing, at the pre-processing step itself data is reduced. Identifying instances is done by using a method close to the boundary between classes. This boundary detection algorithm aims to eliminate instances that are supposed to be non support vectors. After identifying the boundary, SVM was applied for classification which contains only support vectors.

The boundary detection algorithm consists of mainly two steps: 1.Concept independent pre-processing 2.Concept specific sampling. The first step is described as identifying the nearest neighbours of every instance with higher accuracy. Here class labels are not considered. By using efficient algorithms like Locality sensitive Hashing (LSH) [117] is applied to the dataset to obtain nearest neighbours. Locality sensitive hashing is a technique for grouping points in space into buckets based on distance measure operating on the points. Points that are close to each other under the chosen metric are placed in the identical bucket with high probability.LSH can obtain approximate nearest neighbours with high accuracy.

The second step is concept specific sampling where class labels are given. The output from the first step is selected as input for next step. A scoring function is calculated for every instance. The objective of the scoring function is to accord higher scores to instances closer to the boundary between the positive and negative classes. Based on the score obtained instances are considered, and then the SVM is applied to the dataset for classification. The accuracy has increased drastically since only the boundary points are identified.

6. CONCLUSION

This paper has offered Support vector machines in every aspect: Linear Case, Non separable case, Non Linear case with their relevant optimization problems and how to solve them to achieve classification. It gives wide review of optimization models of SVMs, including C-SVM ,v-SVM, least squares SVM, twin SVM, fuzzy SVM, Proximal SVM, Transductive SVM, Knowledge based SVM, Robust SVM, Multi instance SVM, Multi view SVM and also standard algorithms including SMO, Parallel methods ,Efficient methods for large scale SVM. Then SVM for Model selection, feature selection, LOOSVM based on minimizing LOO error bound, probabilistic outputs for SVM, rule extraction from SVM and Concept Boundary Detection method gives the issues regarding enhancements to make SVM more accurate. These models have been used in many real-life applications, such as text categorization, bio-informatics, bankruptcy prediction, remote sensing image analysis, network intrusion and detection, information security, and credit assessment management.

Research in SVMs and research in optimization problems have become increasingly coupled. In this paper, we can see optimization models including linear, nonlinear, second order cone, and semi-definite, integer or discrete, semi-infinite programming. Apparently, there are still many optimization models of SVMs not discussed in this paper, and novel practical problems remaining to be explored will become new challenges to SVM to construct new optimization models.

REFERENCES

[1] Akbani, R.; Kwek1, S.; Japkowicz, N..,"Applying support vector machines to imbalanced datasets", in Proceedings of European Conference on Machine Learning, Lecture Notes in Computer Science 3201,pp. 39–50, 2004.

[2] Alizadeh, F.,Goldfarb D.,"Second-order cone programming", Mathematical Programming, Series *B* 95: pp.3–51, 2003. http://dx.doi.org/10.1007/s10107-002-0339-5

[3] Ataman, K.; Street, W. N.," Optimizing area under the ROC curve using ranking SVMs", in Proceedings of International Conference on Knowledge Discovery in Data Mining., 2005.

[4] Bartlett and S. Mendelson,"Rademacher and Gaussian complexities: risk bound and structural results", Journal of Machine Learning Research, 3,pp.463–482, 2002

[5] Bartlett, P., Schölkopf B., Schuurmans D., (Eds.),"Advances in Large Margin Classifiers",MIT Press, Cambridge, MA,1999.

[6] Bennett K. and A. Demiriz. "Semi-supervised support vector machines", Advances in Neural Information Processing Systems 12,. MIT Press, Cambridge, MA, pp.368–374 1998**.**

[7] Bennett K., Ji, X.,Hu, J., Kunapuli, G., Pang, J. S. "Model selection via bilevel optimization", in Proceedings of IEEE World Congress on Computational Intelligence, pp.1922–1929,2006.

[8] Bennett K., Parrado-Hernández E., "The interplay of optimization and machine learning research", Journal of Machine Learning Research 7: pp.1265–1281, 2006.

[9] Boser, Guyon, and Vapnik, "A training algorithm for optimal margin classifiers",Proceedings of the fifth annual workshop on Computational learning theory.pp.144-152, 1992.

[10] Bradley, P.; Mangasarian, O., "Feature selection via concave minimization and support vector machines", in Proceedings of International Conference on Machine Learning, Morgan Kaufmann,pp.82–90, 1998.

[11] Brefeld, U., Scheffer, T, "Auc maximizing support vector learning", in Proceedings of the 22nd International Conference on Machine Learning, Workshop on ROC Analysis in Machine Learning,2005.

[12] Burges C Joachims, B. Sch¨olkopf and A. Smola, "Making large-scale SVM learning practical", Chapter 11 in Advances in Kernel Methods MIT Press, Cambridge, 1998.

[13] Cortes C.,Vapnik V.,"Support vector networks, in Proceedings of Machine Learning 20: pp.273–297, 1995.

[14] Chang M. W., Lin C. J., "Leave-one-out bounds for support vector regression model selection",Neural Computation 17(5):pp.1188–1222,2005.

[15] Chang K. W.,Hsieh C. J.,Lin C. J., "Coordinate descent method for large-scale L2-loss linear SVM", Journal of Machine Learning Research 9: pp.1369–1398, 2008.

[16] Chiranji b B hattacharyya, "Robust Classification of noisy data using Second Order Cone Programming approach", IClSlP IEEE, 2004.

[17] Chapelle O, V. Vapnik, O. Bousquet and S. Mukherjee, "Choosing multiple parameters for support vector machines", Machine Learning 46, pp.131–159, 2002.

[18] Chapelle O, V. Vapnik, "Model selection for support vector machines",Advances in neural information rocessing systems.cambridge MA.MIT Press,2000

[19] Chen W. J.,Tian, Y. J.,"l*p*-norm proximal support vector machine and its applications",Procedia Computer Science 1(1):pp. 2417–2423, 2010. http://dx.doi.org/10.1016/j.procs.2010.04.272

[20] Crammer, K.; Singer, Y.,"On the algorithmic implementation of multi-class kernel based vector Machines", Journal of Machine Learning Research, 2: 265–292, 2001.

[21] Cristianini, N., Shawe-Taylor, J,"An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.

[22] Dai Y. and Roger Fletcher,"New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds", Mathematical Programming, 106(3):pp.403–421, 2006.

[23] Dai Y. and Roger Fletcher, "Projected Barzilai-Borwein methods for large-scale boxconstrained quadratic programming" Numerische Mathematik, 100(1):pp.21–47, 2005.

[24] Deng, N. Y., Tian, Y. J., "New Method in Data Mining: Support Vector Machines" ,Science Press, Beijing,China, 2004.

[25] Deng, N. Y.; Tian, Y. J. ,"Support Vector Machines: Theory, Algorithms and Extensions"Science Press ,Beijing, China, CRC Press, 2009.

[26] Deng, N. Y., Tian, Y. J., Zhang, C. H., "Support Vector Machines: Optimization Based Theory, Algorithms and Extensions" ,CRC Press, 2012.

[27] Druker, H., Shahrary, B., Gibbon, D. C., "Support vector machines: relevance feedback and information retrieval", Information Processing and Management 38(3),pp. 305–323,2001.

[28] Duan K., S. Keerthi,A. Poo.," Evaluation of simple performance measures for tuning SVM hyper parameters", Neurocomputing 51,pp. 41–59, 2003.

[29] Evgeniou, T., Pontil, M., & Poggio, T. "Regularization networks and support vector machines", Advances in Computational Mathematics ,13, 1, pp. 1-50 ,2000.

[30] Farquhar J. D. R., Hardoon D. R., Meng H. Y.; Taylor J. S.,Szedmák S., "Two view learning: SVM-2K, theory and practice", Advances in Neural Information Processing Systems 18:pp.355–362**.** 2005.

[31] Fung, G.; Mangasarian, O. L.; Shavlik, J., "Proximal Support Vector Machine Classifiers", Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001.

[32] Fung, G.; Mangasarian, O. L.; Shavlik, J.,"Knowledge-based support vector machine classifiers", Advances in Neural Information Processing Systems 15, pp.537–544, 2001.

[33] Fung, G.; Mangasarian, O. L.; Shavlik, J.,"Knowledge-based nonlinear kernel classifiers", Learning Theory and Kernel Machines, Lecture Notes in Computer Science 2777,pp.102–113, 2003. http://dx.doi.org/10.1007/978-3-540-45167-9_9

[34] Fung, G.; Sandilya, S.; Rao, R. B.," Rule extraction from linear support vector machines", in Proceedings of International Conference on Knowledge Discovery in Data Mining, 32–40, 2005.

[35] Gaetano Zanghirati and Luca Zanni,"A parallel solver for large quadratic programs in trainingsupport vector machines",Parallel Computing, 29,pp.535–551, 2003.

[36] Ganapathiraju A., Hamaker J., Picone J.,"Applications of support vector machines to speech recognition", IEEE Transaction on Signal Process 52(8),pp.2348–2355, 2004.. http://dx.doi.org/10.1109/TSP.2004.831018

[37] Goldfarb, D., Iyengar, G.,"Robust convex quadratically constrained programs", Mathematical Programming, Series *B* 97, 495–515,2003. http://dx.doi.org/10.1007/s10107-003-0425-3

[38] Goberna, M. A.,López, M. A., "Linear Semi-Infinite Optimization", New York: John Wiley, 1998.

[39] Guyon I., Weston J., Barnhill, S., Vapnik V., "Gene selection for cancer classification using support vectormachines",MachineLearning46,pp.389–422,2001. http://dx.doi.org/10.1023/A:1012487302797

[40] Gretton, A.,Herbrich, R., Chapelle O., "Estimating the leave-one-out error for classification learning with SVMs",2001 Available from Internet: http://www.kyb.tuebingen.mpg.de/publications/pss/ps1854.

[41] Han J, Micheline Kamber, Jian Pei "Textbook:Data Mining, Second Edition: Concepts and Techniques"

[42] Hsieh, C. J., Chang, K. W., Lin, C. J.,Keerthi, S. S., Sundararajan, S., "A dual coordinate descent method for large-scale linear SVM", in Proceedings of the 25th International Conference on Machine Learning (ICML'08),pp. 408–415, 2008.

[43] Jaakkola, T. S.,Haussler, D., "Exploiting generative models in discriminative classifiers", Advances in Neural Information Processing Systems 11. MIT Press, 1998.

[44] Jaakkola, T. S., Haussler, D. "Probabilistic Kernel regression models", in Proceedings of the 1999 Conference on AI and Statistics. Morgan Kaufmann, 1999.

[45] Johan, A. K. S., Tony, V. G., Jos, D. B.; Bart, D. M., Joos, V., "Least Squares Support Vector Machines", World Scientific Singapore, 2002. ISBN 981-238-151-1

[46] Joachims T, Making large-scale SVM learning practical, in: B. Schoolkopf, C.J.C. Burges, A. Smola €(Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.

[47] Joachims, T., "Transductive inference for text classification using support vector machines", in Proceedings of 16th International Conference on Machine Learning. Morgan Kaufmann, San Francisco,CA, pp.200–209,1999b.

[48] Joachims, T,"Text categorization with support vector machines: learning with many relevant features", in Proceedings of 10th European Conference on Machine Learning,pp. 137–142, 1999a.

[49] Joachims, T.,"Estimating the generalization performance of an SVM efficiently", in Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Franscisco, California,pp.431–438, 2000.

[50] Joachims, T., "Training linear SVMs in linear time", in Proceedings of International Conference on Knowledge Discovery in Data Mining, pp.217–226, 2006.

[51] Jonsson, K.; Kittler, J.; Matas, Y. P.,. "Support vector machines for face authentication", Journal of Image and Vision Computing 20(5):pp.369–3752002. http://dx.doi.org/10.1016/S0262-8856(02)00009-4

[52] Keerthi, S. S., Sundararajan S., Chang, K. W.; Hsieh C. J., Lin, C. J.," A sequential dual method for large scale multi-class linear SVMs", in Proceedings of the International Conference on Knowledge Discovery and Data Mining,pp. 408–416. 2008.

[53] Khemchandani, J. R.; Chandra, S.,"Twin support vector machines for pattern classification", IEEE Transaction on Pattern Analysis and Machine Intelligence 29(5):pp. 905–910., 2007.http://dx.doi.org/10.1109/TPAMI.2007.1068

[54] Kim, K. J. "Financial time series forecasting using support vector machines," Neurocomputing 55(1),pp.307–319,2003. http://dx.doi.org/10.1016/S0925-2312(03)00372-2

[55] Klerk, E. "Aspects of Semidefinite Programming", Kluwer Academic Publishers, Dordrecht. 2002.

[56] Li J. P., Chen, Z. Y., Wei, L. W., Xu W. X., Kou G., "Feature selection via least squares support feature machine", International Journal of Information Technology and Decision Making 6(4),pp. 671–686, 2007,http://dx.doi.org/10.1142/S0219622007002733.

[57] Lin C. F.,Wang, S. D.,"Fuzzy support vector machine", IEEE Transaction on Neural Network 13(2),pp.464–471, 2002,http://dx.doi.org/10.1109/72.991432

[58] Lin H. T.,Lin, C. J., Weng, R. C., "A note on Platts probabilistic outputs for support vector machines", *Machine Learning* 68, 267–276, 2007, http://dx.doi.org/10.1007/s10994-007-5018-6

[59] Lin C.-J., Weng, R. C., & Keerthi, S. S."Trust region Newton method for large-scale logistic regression", JMLR, 9, pp.627-650, 2008.

[60] Liu, Y.; Zhang, D.; Lu, G.; Ma, W. Y.. "A survey of content-based image retrieval with high-level semantics", *Pattern Recognition* 40(1): 262–282, 2007 http://dx.doi.org/10.1016/j.patcog.2006.04.045

[61] Lu, J. W.; Plataniotis, K. N.; Ventesanopoulos, A. N.,"Face recognition using feature optimization and υ-support vector machine", in Proceedings of the 2001 IEEE Signal Processing Society Workshop, 373–382, 2001.

[62] Mangasarian, O. L.; Wild, E. W.,"Multiple instance classification via successive linear programming", Journal of Optimization Theory and Application 137(1),pp. 555–568, 2008.

[63] Martens, D.; Huysmans, J.; Setiono, R.; Vanthienen, J.; Baesens B.,"Rule extraction from support vector machines: an overview of issues and application in credit scoring", Studies in Computational Intelligence (SCI) 80: 33–63,2008. http://dx.doi.org/10.1007/978-3-540-75390-2_2

[64] Melgani, F.; Bruzzone, L.,"Classification of hyperspectral remotesensing images with support vector machines", IEEE Transactions on Geoscience and Remote Sensing 42(8): 1778–1790, 2004. http://dx.doi.org/10.1109/TGRS.2004.831865

[65] Mukkamala, S.; Janoski, G.; Sung, A. H.,"Intrusion detection using neural networks and support vector machines", in Proceedings of IEEE International Joint Conference on Neural Network, 1702–1707, 2002.

[66] Nash, S. G.; Sofer, A., "Linear and Nonlinear Programming". McGraw-Hill Companies, Inc. USA,1996.

[67] Núñez, H.; Angulo, C.; Català, A. "Rule extraction from support vector machines", in European Symposium on Artificial Neural Networks (ESANN), 107–112, 2002.

[68] Osuna E., R.Freund, and F.Girosi, "Training support vector machines: an application to face detection", Proceedings CVPR'97, 1997b.

[69] Panda N., E. Y. Chang, and G. Wu, "Concept boundary detection for speeding up SVMs",Proceedings of the 23 International Conference on Machine Learning, Pittsburgh, PA, 2006

[70] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 2006, "Introduction to data mining", Addison-Wesley,769.

[71] Peng, Y.; Kou, G.; Wang, G. X., *et al*. "Empirical evaluation of classifiers for software risk management"., International Journal of Information Technology and Decision Making 8(4): 749–767,2009.http://dx.doi.org/10.1142/S0219622009003715

[72] Platt J., Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in Smola, A.; Bartlett, P.; Schölkopf, B.; Schuurmans, D. (Eds.). Advances in Large Margin Classifiers. MIT Press, Cambridge, MA; 2000.

[73] Robert Andrews, Joachim Diederich, and Alan B. Tickle. "Survey and critique of techniques for extracting rules from trained artificial neural networks", Knowledge-Based Systems, 8(6):pp.373–389, 1995.

[74] Schölkopf, B.; Smola, A. J. 2002. *Learning with Kernels–Support Vector Machines, Regularization, Optimization*

[75] Setiono, R.; Baesens, B.; Mues, C., "Risk management and regulatory compliance: a data mining framework based on neural network rule extraction", in Proceedings of the International Conference on Information Systems (ICIS'06). 2006.

[76] Serafini T,Gaetano Zanghirati, and Luca Zanni,"Gradient projection methods for quadraticprograms and applications in training support vector machines", Optimization Methods and Software, 20,pp.353–378, 2005

[77] Serafini T and Luca Zanni. "On the working set selection in gradient projection-based decomposition techniques for support vector machines" Optimization Methods and Software, 20:583–596, 2005.

[78] Shalev-Shwartz, S., Singer, Y., & Srebro, N. "Pegasos: primal estimated sub-gradient solver for SVM." ICML, (2007).

[79] Shao, Y.; Zhang, C. H.; Wang, X. B.; Deng, N. Y.," Improvements on twin support vector machines", IEEE Transactions on Neural Networks 22(6),pp.962–968, 2011. http://dx.doi.org/10.1109/TNN.2011.2130540.

[80] Shin, K. S.; Lee, T. S.; Kim, H. J. "An application of support vector machines in bankruptcy prediction model", Expert Systems with Applications 28(1),pp.127–135, 2005. http://dx.doi.org/10.1016/j.eswa.2004.08.009

[81] Smola, P. Bartlett, B. Sch¨olkopf, & D. Schuurmans (Eds.), Advances in large margin classifiers ,pp. 171–203.Cambridge, MA: MIT Press

[82] Smola, A. J., Vishwanathan, S. V. N., & Le, Q.,Bundle methods for machine learning. NIPS,2008.

[83] Sonnenburg, S.; Rätsch, G.; Schäfer, C.; Schölkopf, B.," Large scale multiple kernel learning", Journal of Machine Learning Research 7,pp.1–18, 2006.

[84] Szedmak S, Shawe-Taylor J "Synthesis of maximum margin and multiview learning using unlabeled data", Neurocomputing 70,pp. 1254–1264, 2007.

[85] Tian, Q.; Hong, P.; Huang, T. S.. "Update relevant image weights for content based image retrieval using support vector machines", in Proceedings of IEEE International Conference on Multimedia and Expo 2, pp.1199–1202, 2000.

[86] Tian, Y. J. 2005. "Support Vector Regression and Its Applications": PhD Thesis. China Agricultural University

[87] Tian, Y. J.; Deng, N. Y.. "Leave-one-out bounds for support vector regression", in Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies

[88] Tian, Y. J.; Yu, J.; Chen, W. J. "lp-norm support vector machine with CCCP", in Proceedings of 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery,pp.1560–1564, 2010.

[89] Tefas, A.; Kotropoulos, C.; Pitas, I. "Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication", IEEE Transaction on Pattern Analysis and Machine Intelligence 23(7),pp.735–746, 2001. http://dx.doi.org/10.1109/34.935847

[90] Tsoumakas, G.; Katakis, I. "Multi-label classification: an overview", International Journal of Data Warehousing and Mining 3(3),pp.1–13, 2007 http://dx.doi.org/10.4018/jdwm.2007070101

[91] Vapnik, V. N..,"Statistical Learning Theory", New York: John Wiley and Sons**,** 1998.

[92] Vapnik, V.N. "Estimation of Dependences Based on Empirical Data", Addendum 1, New York: Springer-Verlag, 1982.

[93] Vapnik, V. N.; Chapelle, O. Bounds on error expectation for SVM, in Advances in Large-Margin Classifiers, Neural Information Processing. MIT Press, pp.261–280, 2000.

[94] Vapnik, V. N.; Vashist, A.,"A new learning paradigm: learning using privileged information", NeuralNetworks 22(5).pp. 544–577., 2009.

[95] Vanderbei, R. J.. "Linear Programming: Foundations and Extensions*",* Second edition. Kluwer Academic Publishers, 2001.

[96] Weston, J. "Leave-one-out support vector machines", in Proceedings of the International Joint Conference on Artificial Intelligence,pp. 727–731, 1999.

[97] Weston, J.; Mukherjee, S.; Vapnik, V. "Feature selection for svms", Advances in Neural Information Processing Systems 13,pp. 668–674, 2001.

[98] Xu, L.; Schuurmans, D. "Unsupervised and semi-supervised multi-class support vector machines", in Proceedings of the 20th National Conference on Artificial Intelligence, 2005.

[99] Yang, Q.; Wu, X. D.,"Challenging problems in data mining research", International Journal of Information Technology and Decision Making 5(4),pp.567–604, 2006. http://dx.doi.org/10.1142/S0219622006002258

[100] Yang, Z. X. "Support Vector Ordinal Regression and Multi-class Problems": PhD Thesis. China Agricultural University, 2007..

[101] Yang, Z. X.; Tian, Y. J.; Deng, N. Y. "Leave-one-out bounds for support vector ordinal regression machine", Neural Computing and Applications 18(7): 731–748, 2009. http://dx.doi.org/10.1007/s00521-008-0217

[102] Yang, S. X.; Tian, Y. J., "Rule extraction from support vector machines and its applications", in Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 221–224, 2011. http://dx.doi.org/10.1109/WI-IAT.2011.132

[103] Yi-Wei Chen and Chih-Jen Lin .,"Combining SVMs with Various Feature Selection Strategies", Feature Extraction Studies in Fuzziness and Soft Computing Volume 207, pp 315-324, 2006

[104] Zanni L. "An improved gradient projection-based decomposition technique for support vectormachines",. Computational Management Science, 3(2),pp.131–145, 2006

[105] Zanni L, V. Ruggiero, "A modified projection algorithm for large strictly convex quadratic programs",J. Optim. Theory Appl. 104 ,281–299, 2000.

[106] Zanni L, V. Ruggiero, "Variable projection methods for large convex quadratic programs", in: D.Trigiante (Ed.), Recent Trends in Numerical Analysis, vol. 3, Nova Science Publishers, pp. 299-313,2000.

[107] Zanni L ,Thomas Serafini , Gaetano Zanghirati ,"Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems", Journal of Machine Learning Research 7 ,pp. 1467–1492 ,2006.

[108] Zhao, K.; Tian, Y. J.; Deng, N. Y.,"Unsupervised and semi-supervised two-class support vector machines", in Proceedings of the 6th IEEE International Conference on Data Mining Workshops, 813–817, 2006.

[109] Zhao, K.; Tian, Y. J.; Deng, N. Y.,"Unsupervised and semi-supervised lagrangian support vector machines", in Proceedings of the 7th International Conference on Computational Science Workshops,Lecture Notes in Computer Science 4489,pp.882–889., 2007.http://dx.doi.org/10.1007/978-3-540-72588-6_140

[110] Zhu, J.; Rosset, S.; Hastie, T.; Tibshirani, R.."1-norm support vector machines", Advances in Neural Information Processing Systems 16,pp.49–56, 2004.

[111] Zou, H.; Yuan, M. "The F∞-norm support vector machine", Statistica Sinica 18: 379–398., 2008.http://dx.doi.org/10.1093/bioinformatics/btm036

[112] Zanghirati G and Luca Zanni.,"A parallel solver for large quadratic programs in training support vector machines".,Parallel Computing, 29,pp.535–551, 2003

[113] Platt J., "Fast training of support vector machines using sequential minimal optimization", in Schölkopf, B.; Burges, C. J. C.; Smola, A. J. (Eds.). Advances in Kernel Methods Support Vector Learning.Cambridge, MA: MIT Press,pp. 185–208,1999.

[114] Bordes A., S. Ertekin, J. Weston, and L. Bottou,"Fast kernel classifiers with online and active learning",In Journal of Machine Learning Research, 6,pp.1579-1619, 2005.

[115] Glasmachers T., C. Igle.,"Maximum-gain working set selection for SVMs", In Journal of Machine Learning Research, 7,pp.1437–1466, 2006.

[116] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition". In Data Mining and Knowledge Discovery, 2(2),pp. 1-47, 1998.

[117] Slaney, M.,Casey M.,"Locality Sensitive Hashing for finding nearest neighbours", Signal Processing Magazine, IEEE, 2008.

