Application of Quadratic Programming (QP) in Support Vector Machines (SVMs) for Supervised Classification

Abu Saleh Mosa Faisal
New Mexico Institute of Mining and Technology
abusalehmosa.faisal@student.nmt.edu

Abstract

The purpose of this project is to explore how Quadratic Programming (QP) is used in Support Vector Machines (SVMs). In this project there will be brief discussion on Quadratic Programming, SVMs and Kernel Tricks and illustration of the techniques by implementing them on data set. The Support Vector Machines (SVMs) maps the the data into high dimensional feature space and then constructs an optimal separating hyperplane in the feature space. To find the optimal solution of parameters related to equation of optimal hyperplane requires to solve a quadratic programming problem. Often data can not be separated by a linear or straight line. Nonlinear SVMs use kernel trick to classify data into appropriate groups. This project will give explicit idea how this optimal classifier can be estimated from data.

Keywords

SVMs, Quadratic Programming, Kernel Function, Hyperplane

1 Introduction

The Support Vector Machines (SVMs) was first introduced by the Vladimir Vapnik and Alexey Chervonenkis in 1963 [1]. Before that in 1936 RA Fisher proposed an algorithm for pattern recognition [2]. In 1974 the concept of "statistical learning theory" introduced by Vapnik and Chervonenkis [3]. After that in 1992 Berhard E. Boser, Isabelle M. Guyon and Vladimir Vapnik introduce the concept of kernel tricks to classify non linear data [8]. SVMs algorithm is providing state-of-art solutions in several problems in machine learning and statistical pattern recognition [5]. Quadratic programming is an important part of this classification technique [9]. The concepts of Support Vector Machines classification rely on estimating the optimal hyperplane [6]. This optimal hyperplane is estimated by solving a quadratic programming problem. Often data can not be separated by drawing just a line in between them. In such cases non linear Support Vector Machines technique use to classify the data set. The Concept of kernel function plays vital role in non linear SVMs [4]. A hyperplane can be defined by the following form of equation

$$w^T x + b = 0$$

Where, w is a weight vector, x is input vector and b is the bias. To get an optimal hyperplane margin (distance between hyperplane and closest data point for a given weight vector w and bias b) needs to be maximized. To maximize margin we need to minimize the $\frac{1}{2}||w||^2$. This is a quadratic programming optimization problem. For linearly inseparable data one needs to choose a proper kernel function for better performance of SVMs classifier. The kernel is way of computing the dot product of two vectors x and y in some (very high dimensional) feature space, which is why kernel functions sometimes called "generalized dot product" [10].
More formally, if we have data $x^i, x^j \in X$ and a map ϕ: X→ R^n then

$$k(x^i, x^j) = \langle \phi(x^i), \phi(x^j) \rangle$$

is a kernel function. Input vector (x) need to be replaced by $\phi(x)$ where ϕ is the feature map corresponding to the original kernel k. Some popular kernel functions are Polynomial kernel, Gaussian Radial Basis Function, Laplace RBF Kernel, Linear Kernel etc. In this project, appropriate kernel function will be used in the illustration part.

2 Hyperplane

In Support Vector Machines classification the goal is to produce an optimal classifier that could classify the unseen training sample with minimum classification error [7]. There could be many possible linear classifiers(hyperplane) that can separate data but there is only one hyperplane that maximizes the margin (distance between hyperplane and closest training data points). This is called optimal separating hyperplane.

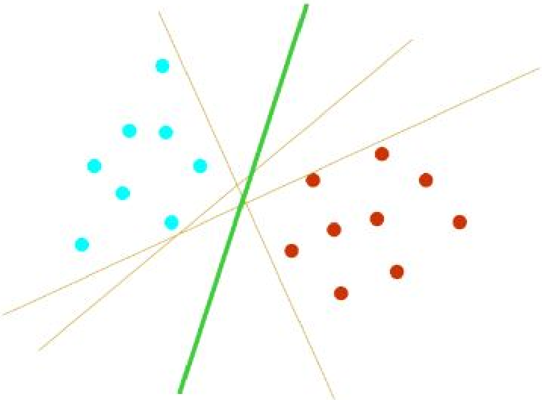

Figure 1: Optimal Separating Hyperplane

2.1 Optimal Separating Hyperplane with Support Vectors

Let we have a data set $\mathcal{D}$ such that (x^i, y_i) Where $x^i \in R^n$ is the training samples associated with class labels $y_i \in \{+1, -1\}$. There exist a hyperplane $\{x|w^T x + b = 0\}$ with, $w \in R^n$, $w \neq 0$, $b \in R$ that separates the training sample. Here, w is a weight vector and b is a bias. The side of the margin can be defined as hyperplane. The marginal hyperplanes can be expressed as

$$H_1: \quad \{x|w^T x^i + b \leq 1\} \; for \; y_i = -1 \tag{1}$$

$$H_2: \quad \{x|w^T x^i + b \geq 1\} \; for \; y_i = 1 \tag{2}$$

Above two can be combined as following:

$$y_i(w^T x^i + b) - 1 \geq 0 \qquad \forall i \tag{3}$$

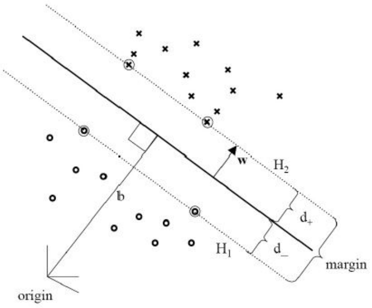

Figure 2: Marginal & Optimal hyperplane

In Figure 2 the points lie on the H_1 and H_2 are **Support Vectors**. Support vectors are the points closest to hyperplanes and can influence the position and orientation of hyperplane [8]. They are the most critical or tough points in training samples to classify accurately. By utilizing them we build the maximum margin of the classifier. Since only support vectors are sole influential training samples, the other training samples satisfying equation(3) falling either side of hyperplane can be attributed to different classes. d_+ and d_- representing shortest distance between optimal hyperplane and closest positive and negative points respectively.
The distance d(w,b;x^i) of a point x^i from hyperplane is following:

$$d(w, b; x^i) = \frac{|w^T x^i + b|}{||w||} \tag{4}$$

The margin can be derived from the following equation by taking two points x^i and x^j on the H_1 and H_2. H_1 plane can be expressed as $w^T x^i + b = -1$ and H_2 is $w^T x^j + b = 1$. So, the margin is the distance between marginal hyperplanes H_1 and H_2. The margin can be formulated by summing d_+ and d_- which is following:
$l(w, b)$=$d_+(w, b; x^i) + d_-(w, b; x^j)$
$= \frac{|w^T x^i + b|}{||w||} + \frac{|w^T x^j + b|}{||w||}$
$= \frac{|-1|}{||w||} + \frac{|1|}{||w||}$
$= \frac{1}{||w||} + \frac{1}{||w||}$
$= \frac{2}{||w||}$
So the margin of the hyperplane is $l(w, b) = \frac{2}{||w||}$. We want as big as margin for our training sample classification. To maximize the margin we need to minimize the $||w||$. A hyperplane is optimal if it minimize the loss function $\Psi(w, b) = \frac{1}{2}||w||^2$. The constraints are $y_i(w^T x^i + b) - 1 \geq 0 \quad \forall i$. This is a constrained(convex) optimization problem with quadratic objective function and linear constrains.

3 SVMs Classification when data are linearly separable

The primal optimization problem is

$$\begin{array}{ll} \min\limits_{w} & (\frac{1}{2})||w||^2 \\ \text{s.t.} & y_i(w^T x^i + b) - 1 \geq 0 \quad i = 1, ..., n \end{array} \tag{5}$$

The solution of the primal problem involves construction of dual problem where Lagrangian multipliers α_i is associated with every constraint in the primal. Lagrangian of this problem can be written as

$$\mathcal{L} = (\frac{1}{2}) w^T w - \sum_{i=1}^{n} \alpha_i [y_i(w^T x^i + b) - 1]; \qquad \alpha_i \geq 0 \tag{6}$$

The optimization is equivalent to find a saddle point of Lagrangian function $\mathcal{L}$ [8]. Finding a saddle point requires minimization of $L(w, b, \alpha_i)$ with respect to w and b maximization of $\mathcal{L}$ with respect to α_i. To minimize the Lagrangian function, calculation of gradient or partial derivative with respect to w and b is required. The gradient with respect to w is following:

$$\frac{d\mathcal{L}}{dw} = (\frac{1}{2}) \frac{dw^T w}{dw} - \frac{d \sum_{i=1}^{n} \alpha_i [y_i(w^T x^i + b) - 1]}{dw} \tag{7}$$

$$= \frac{1}{2} \times 2w - \frac{d \sum_{i=1}^{n} \alpha_i y_i w^T x^i}{dw} - \frac{d \sum_{i=1}^{n} \alpha_i y_i b}{dw} + \frac{d \sum_{i=1}^{n} \alpha_i}{dw} \tag{8}$$

$$= w - \sum_{i=1}^{n} \alpha_i y_i x^i \tag{9}$$

After equating with zero we get our optimal w^*

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x^i \tag{10}$$

The gradient with respect to b is

$$\frac{d\mathcal{L}}{db} = (\frac{1}{2})\frac{dw^Tw}{db} - \frac{d\sum_{i=1}^{n}\alpha_i[y_i(w^Tx^i + b) - 1]}{db} \tag{11}$$

$$= 0 - \frac{d\sum_{i=1}^{n}\alpha_i y_i w^T x^i}{db} - \frac{d\sum_{i=1}^{n}\alpha_i y_i b}{db} + \frac{d\sum_{i=1}^{n}\alpha_i}{db} \tag{12}$$

$$= 0 - \sum_{i=1}^{n}\alpha_i y_i \tag{13}$$

After equating with zero

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Let C= $\alpha_i[y_i(w^Tx^i + b) - 1]$. To find a maximum value of $\mathcal{L}$, C needs to be zero where both $\alpha_i \geq 0$ and $f_i(w,b) = [y_i(w^Tx^i + b) - 1] \geq 0$. So there are two conditions $\alpha_i = 0$ and $f_i(w^*, b^*) > 0$ and $\alpha_i > 0$ and $f_i(w^*, b^*) = 0$. The equation of hyperplane allows us to identify if a point is the support vector or correctly classified point. Each nonzero α_i indicates that corresponding x^i is a support vector. Each α_i taking zero value indicates that corresponding x^i is correctly classified. So, in the equation $w^* = \sum_{i=1}^{n}\alpha_i y_i x^i$ all the points who lies on marginal hyperplanes take $\alpha_i > 0$ value and rest of the points who correctly classified take $\alpha_i = 0$. Each of the points lie on marginal hyerplanes are support vectors. So, w^* is depended on only support vectors. Now, substitute the w^* and $\sum_{i=1}^{n}\alpha_i y_i = 0$ into original Lagrangian equation formulates an expression only depending on α_i.

$$\mathcal{L}(w^*, b, \alpha) = (\frac{1}{2})w^Tw - \sum_{i=1}^{n}\alpha_i y_i w^T x^i - \sum_{i=1}^{n}\alpha_i y_i b + \sum_{i=1}^{n}\alpha_i \tag{14}$$

$$\mathcal{L}(w^*, b, \alpha) = (\frac{1}{2})(\sum_{i=1}^{n}\alpha_i y_i x^i)^T(\sum_{i=1}^{n}\alpha_i y_i x^i) - \sum_{i=1}^{n}\alpha_i y_i(\sum_{i=1}^{n}\alpha_i y_i x^i)^T x^i - \sum_{i=1}^{n}\alpha_i y_i b + \sum_{i=1}^{n}\alpha_i \tag{15}$$

After simplification

$$\mathcal{L}(w^*, b, \alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x^i)^T x^j \tag{16}$$

$\mathcal{L}$ needs to be maximize to find the optimal Lagrangian multipliers α_i . This $\mathcal{L}$ is dual form of initial $\mathcal{L}$. Note here, in dual form we need to find only the dot product of each of the combination of training samples x^i and x^j. So, our dual problem is

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x^i)^T x^j \\ &\text{s.t.} \sum_{i=1}^{n}\alpha_i y_i = 0 \\ &\qquad \alpha_i \geq 0 \qquad\qquad i = 1, ..., n \end{aligned} \tag{17}$$

This dual problem is a convex quadratic programming problem. Solving dual problem is equivalent to solving primal problem under KKT(Krush-Khun- Tucker) conditions. We need to estimate the optimal value of α by solving this problem.To solve this problem there are several algorithms. More efficient algorithm is sequential minimal optimization(SMO) [11]. SMO algorithm works by picking some pairs of α_i, α_j by holding the other α_i's fixed. Now update the dual objective function with this pair of value. Repeat the process until it converges to a value with minute tolerance limit [11]. Optimal value of α^* substitute into the equation of w^*. The remaining parameter b can be found by using the hyperplane equation.

Let S is the subset of training samples who are support vectors. Each support vector $x^s \in S$ will satisfy the following equation

$$y_s((w^*)^T x^s + b) = 1$$

Substitute the value of w^* into the equation

$$y_s(\sum_{p \in S} \alpha_p y_p (x^p)^T x^s + b) = 1$$

Multiplying both side with y_s we get

$$b^* = y_s - \sum_{p \in S} \alpha_p y_p (x^p)^T x^s$$

Instead of using one single support vector b should be calculated from average of all support vectors in S. So the updated equation for b is

$$b^* = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{p \in S} \alpha_p^* y_p (x^p)^T x^s) \tag{18}$$

Now we have optimal estimate of w^*, b^* to find the optimal orientation of separating hyperplane. To classify a new point x^{test} we need to evaluate $sign((w^*)^T x^{test} + b^*)$. Since only a very small subset of training samples(support vectors) can fully specify the decision function, $w^* = \sum_S \alpha_s y_s x^s$ should use to evaluate the sign of new testing point. So, the equation will be

$$y(x) = sign(\sum_S \alpha_s^* y_s (x^s)^T x^{test} + b^*) \tag{19}$$

The testing point belongs to which class will be defined by the sign of this equation.

4 SVMs Classification when data are not fully linearly separable

In order to handle the data that is not perfectly linearly separable slack variables need to incorporate in the equation for hyperplane to allow misclassification of difficult or noisy training points. This is called soft margin classification [13].

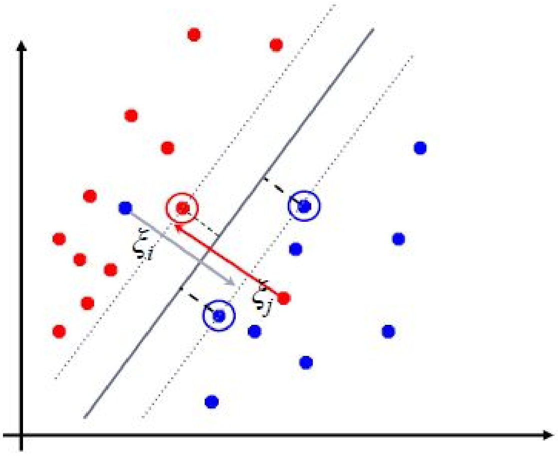

Figure 3: Soft Margin Classification: Classification error is allowed here for some points. Then try to minimize the error.

$$\begin{aligned} w^T x^i + b \geq +1 - \xi_i \qquad & for \quad y_i = +1 \\ w^T x^i + b \leq -1 + \xi_i \qquad & for \quad y_i = -1 \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{20}$$

By combining both of the equation we get

$$\begin{aligned} y_i(w^T x^i + b) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0 \quad \forall i \end{aligned} \tag{21}$$

The variable ξ_i are slack variables use to balance the misclassification in the set of inequalities. The formation of the optimization problem is

$$\min_{w,b,\xi} \quad (\frac{1}{2})w^T w + C\sum_{i=1}^{n} \xi_i \tag{22}$$

Subject to

$$\begin{aligned} y_i(w^T x^i + b) \geq 1 - \xi_i \quad i = 1, ..., n \\ \xi_i \geq 0 \quad i = 1, ..., n \end{aligned} \tag{23}$$

C is a positive constant worked as a tuning parameter which controls the trade of between size of margin and penalty of slack variable. Lagrangian of this primal problem is following

$$\mathcal{L}(w, b, \xi_i, \alpha_i, \nu_i) = (\frac{1}{2})w^T w + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i[y_i(w^T x^i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \nu_i \xi_i \tag{24}$$

The Lagrangian multipliers $\alpha_i, \nu_i \geq 0$. Minimize $\mathcal{L}$ by taking partial derivative with respect to w, b and ξ_i and equate to zero provides

$$\frac{d\mathcal{L}}{dw} = \frac{1}{2} \times 2w - \sum_{i=1}^{n} \alpha_i y_i x^i \tag{25}$$

after equating with zero we get $w^* = \sum_{i=1}^{n} \alpha_i y_i x^i$.

$$\tfrac{d\mathcal{L}}{db} = 0 - \textstyle\sum_{i=1}^{n} \alpha_i y_i$$

After equating with zero

$$\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0$$

Now taking partial derivative with respect to ξ_i

$$\tfrac{d\mathcal{L}}{d\xi_i} = 0 \quad \Rightarrow C = \alpha_i + \nu_i \Rightarrow 0 \leq \alpha_i \leq C \quad i = 1, ..., n$$

However, $\nu_i \geq 0 \quad \forall i$ implies $\alpha_i \leq C$. Now substituting these into Lagrangian equation to get the Dual Lagrangian with constraints

$$\mathcal{L}(w, b, \xi_i, \alpha_i, \nu_i) = (\frac{1}{2})(\sum_{i=1}^{n} \alpha_i y_i x^i)^T \sum_{i=1}^{n} \alpha_i y_i x^i + (\alpha_i + \nu_i)\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i[y_i((\sum_{i=1}^{n} \alpha_i y_i x^i)^T x^i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \nu_i \xi_i \tag{26}$$

$\mathcal{L}(w, b, \xi_i, \alpha_i, \nu_i) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j + \alpha_i \sum_{i=1}^{n} \xi_i + \nu_i \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \nu_i \xi_i$

$$\mathcal{L}(w, b, \xi_i, \alpha_i, \nu_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j \tag{27}$$

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j$$

Subject to

$$\begin{aligned} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1, ..., n \end{aligned} \tag{28}$$

From KKT conditions there may have couple of conditions. First condition is $\nu_i > 0$ and $\xi_i = 0$ and the second condition is $\nu_i = 0$ and $\xi_i > 0$. As we know $\alpha_i = C - \nu_i$ for the first scenario $\alpha_i < C$ and $\xi_i = 0$ so we get $y_i[w^T x^i + b] = 1$. This means training samples x^i is on the margin. For the second condition where $xi_i > 0$ and $\nu_i = 0$ $\alpha_i = C$ so we get $y_i[w^T x^i + b] = 1 - \xi_i$. Which means training sample x^i is correctly classified. b^* will be calculated in similar manner as equation(18).

$$b^* = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{p \in S} \alpha_p^* y_p (x^p)^T x^s) \tag{29}$$

One important feature is the set of Support Vectors used to calculate b is determined by finding the indices i where $0 < \alpha_i \leq C$. The soft margin SVMs classifier will be

$$y(x) = sign(\sum_{S} \alpha_s^* y_s (x^s)^T x^{test} + b^*) \tag{30}$$

The testing point belongs to which class will be defined by the sign of this equation.

5 Kernel Function for non linear SVMs

For nonlinear classification kernel function is used to transform training samples to a high dimensional space where they can be separated by the hyperplane.

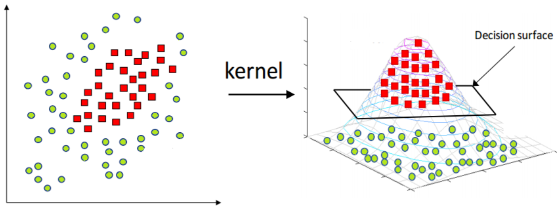

Figure 4: Kernel trick is mapping the training samples into high dimensional space where they are separated by a hyperplane

Kernel tricks are popular in different machine learning algoritms specially in Support Vector Machines [12]. It is useful when data set is not linearly separable. A Kernel function maps the data set to a high dimensional space where they can be linearly separable. Remember our dual Lagrangian function where inner product of training samples were used to solve dual of the problem.

$$\mathcal{L}_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j \tag{31}$$

The linear classifier depends only on the dot product of $(x^i)^T x^j$. If we map the every training points into high dimensional space via some transformation $\Phi : x \rightarrow \phi(x)$ the inner product of dual function becomes $K(x^i, x^j)$=$((\phi(x^i))^T.\phi(x^j))$ Here we just need to replace the dot product of with dot product of mapping functions. Only dot product of mapped inputs in the feature space need to be determined without explicit calculation of ϕ. The dual optimization problem will be

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j ((\phi(x^i))^T.\phi(x^j)) \tag{32}$$

Subject to the constraints

$$\begin{gathered} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1, ..., n \end{gathered} \tag{33}$$

The optimal value of w is $w^* = \sum_{i=1}^{n} \alpha_i^* y_i \phi(x^i)$. The optimal value of b will calculate as

$$b^* = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{p \in S} \alpha_p^* y_p (\phi(x^p))^T \phi(x^s)) \tag{34}$$

The SVMs classifier will be

$$y(x) = sign(\sum_{S} \alpha_s^* y_s (\phi(x^s))^T \phi(x^{test}) + b^*) \tag{35}$$

There is a necessary and sufficient condition to be a kernel. The Mercer theorem tells us for any Kernel K: $R^n \times R^n \to R$ to be a valid kernel, it is necessary and sufficient that for any training sample set $\{x^1, ..., x^n\}$ the corresponding kernel matrix is symmetric positive semi-definite. Thus by choosing an appropriate kernel function we can construct a classifier which is linear in feature space even though non-linear in original space [16]. Some important and popular kernel functions are following:
Linear Kernel: $K(\mathbf{x}^i, \mathbf{x}^j) = \mathbf{x}^i.\mathbf{x}^j$
Polynomial Kernel: $K(\mathbf{x}^i, \mathbf{x}^j) = ((\mathbf{x}^i)^T.\mathbf{x}^j + 1)^d$ [d=degree of polynomial]
Gaussian Radial Basis Function(RBF): $K(\mathbf{x}^i, \mathbf{x}^j) = \exp^{\gamma|\mathbf{x}^i - \mathbf{x}^j|^2}$ mostly $[\gamma = \frac{1}{2\sigma^2}]$

6 Data

Couple of data sets are being used here to illustrate the concepts of quadratic programming, finding an optimal hyperplane and classification by using support vector machine. All of the data sets collected from the UCI machine learning data repository. One of the data set is Divorce predictors data set. In this data set participants completed the personal information form and divorce predictors scale. There are 54 attributes to classify the data into divorced or not divorced [14]. Another data set is Cryotherapy data set where data set contains wart treatment result of 90 patients using cryotherapy. There are seven attributes to classify the result of the treatment(recovered, not recovered) [15].

7 Analysis & Result

Our first data is divorce predictors data set. I have used my own algorithm and MATLAB built-in algorithm to build a proper model for this data set. As, this data set contains 54 attributes I have selected two attributes by using best feature selection process. Attribute 6(We don't have time at home as partners) and Attribute 11(I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other) came as significant pair. Here is the classification accuracy report of this data set for three SVMs classification.

Table 1: Accuracy Report for different classification methods

Classification	Accuracy(%)
Linear SVMs	41.18
Nonlinear SVMs	58.82
Nonlinear SVMs with kernel trick(RBF)	97.05

The nonlinear SVMs with kernel trick(RBF)is more preciously classify the data then other two. The optimal hyperplane is

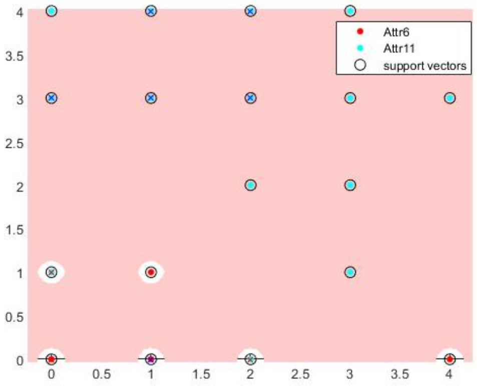

Figure 5: Optimal hyperplane of Divorce predictor data with attribute 6 and attribute 11

So, by using kernel trick more specifically Gaussian radial basis kernel function we can predict about 97% of divorced cases. The other data set is containing information about wart treatment of 90 patients using cryotherapy. By using the best feature selection process two variables age and time have been selected from 6 variables. After preparing the data set three SVMs classification techniques have been used to see which one works best. Here is the prediction accuracy report:
The optimal hyperplane for this model is

Table 2: Accuracy Report for different classification methods

Classification	Accuracy(%)
Linear SVMs	50
Nonlinear SVMs	55.56
Nonlinear SVMs with kernel trick(RBF)	88.88

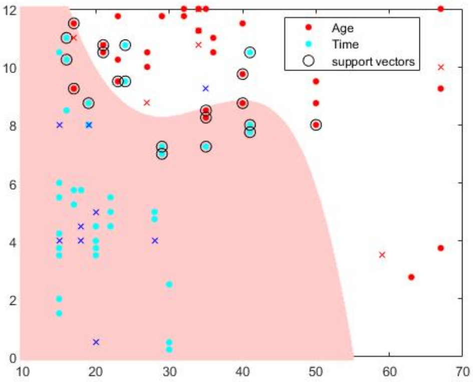

Figure 6: Optimal hyperplane of wart treatment result by using cryotherapy data with variables Age and Time

Data are not linearly separable. So Gaussian radial basis function or kernel is appropriate for classification of this data. It is showing pretty good accuracy in predicting the recovered patients.

8 Conclusion

In this paper I have started with the brief literature review, discussion of basic concepts and definitions such as hyperplane, support vectors, margin etc. The formulation of primal problem, derivation of dual problem by using Lagrangian function, then optimal solution for different parameters of a hyperplane have been derived mathematically. Detail derivation of Lagrangian function, dual formulation have been covered for both linearly separable and linearly non separable data set. I have explained how by using support vectors only we can estimate our optimal hyperplane. Other than support vectors there is no need of other samples in classification process. That is the reason of calling it support vector machines (SVMs). For linearly inseparable data set kernel tricks has been introduced to show how by changing dimension of linearly non separable data can be classified linearly in higher dimensional space. For the illustration I have used both my own written algorithm and MATLAB built-in functions. I have used CVX for write simple program for linear SVMs and non linear SVMs. Then used them to analyze two real life data set. By analysis these two data set I have found in divorce predictor data set two attribute can produce classification accuracy upto 97% . On the other hand other data set 89% accuracy in predicting a patient recovery. I have achieved an explicit and sound concept about how SVMs can be used in data classification and pattern recognition.

References

[1] Stephen Boyd, Lieven Vandenberghe. *"Pairwise Multi-classification Support Vector Machines: Quadratic Programming (QP-PAMSVM) formulations."* 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp205-210).

[2] R. A. Fisher. *"The use of multiple measurements in taxonomic problems"* Annals of human genetics, vol. 7, no. 2, pp. 179–188, 1936.

[3] V. Vapnik. *"The nature of statistical learning theory"*. Springer science & business media, 2013.

[4] Schölkopf, Bernhard. *"The kernel trick for distances."* In Advances in neural information processing systems, pp. 301-307. 2001.

[5] Sha, Fei, Lawrence K. Saul, and Daniel D. Lee. *"Multiplicative updates for nonnegative quadratic programming in support vector machines."* In Advances in neural information processing systems, pp. 1065-1072. 2003.

[6] Bhuvaneswari, P., and J. Satheesh Kumar. *"Support vector machine technique for EEG signals."* International Journal of Computer Applications 63, no. 13 (2013).

[7] Gunn, Steve R. *"Support vector machines for classification and regression."* ISIS technical report 14, no. 1 (1998): 5-16.

[8] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. *"A training algorithm for optimal margin classifiers."* In Proceedings of the fifth annual workshop on Computational learning theory, pp. 144-152. 1992.

[9] Scheinberg, Katya. *"An efficient implementation of an active set method for SVMs."* Journal of Machine Learning Research 7, no. Oct (2006): 2237-2257.

[10] Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. *"Kernel principal component analysis."* In International conference on artificial neural networks, pp. 583-588. Springer, Berlin, Heidelberg, 1997.

[11] Platt, John. *"Sequential minimal optimization: A fast algorithm for training support vector machines." (1998).*

[12] Kwok, James T., and Ivor W. Tsang. *"Learning with idealized kernels."* In Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 400-407. 2003.

[13] Cortes, Corinna, and Vladimir Vapnik. *"Soft margin classifier."* U.S. Patent 5,640,492, issued June 17, 1997.

[14] Yöntem, Mustafa Kemal, Kemal Adem, Tahsin İlhan, and Serhat Kılıçarslan. *"Divorce prediction using correlation based feature selection and artificial neural networks."* Nevşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi 9, no. 1 (2019): 259-273.

[15] Khozeimeh, Fahime, Farahzad Jabbari Azad, Yaghoub Mahboubi Oskouei, Majid Jafari, Shahrzad Tehranian, Roohallah Alizadehsani, and Pouran Layegh. *"Intralesional immunotherapy compared to cryotherapy in the treatment of warts."* International journal of dermatology 56, no. 4 (2017): 474-478.

[16] Fradkin, Dmitriy, and Ilya Muchnik. *"Support vector machines for classification."* DIMACS series in discrete mathematics and theoretical computer science 70 (2006): 13-20.