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

May 1, 2020

Overview

1. Introduction
   - Hyperplane
   - Support Vectors
2. Optimal hyperplane estimation when data are linearly separable(Hard Margin)
3. SVMs Classification when data are not fully linearly separable(Soft Margin)
4. Kernel Tricks for non linear data classification
5. Data
6. Data analysis Results
7. Conclusion

Background

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 kernel tricks to classify non linear data [3].

Goal

The goal of this project is to explain the application of quadratic programming in support vector machines(SVMs). Later on supervised classification of couple of data sets by using SVMs.

Hyperplane

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. In Support Vector Machines classification the goal is to produce an optimal classifier that could classify the unseen training sample with minimum classification error [2].

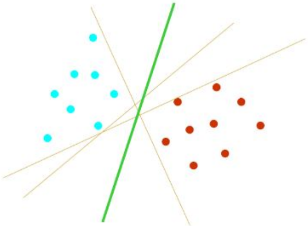

Figure: Optimal Separating Hyperplane

Finding an Optimal Hyperplane

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 $y_i \in \{+1, -1\}$. 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:

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

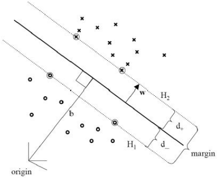

Figure: Marginal Hyperplanes

Support vectors

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 [3]. Lets take two points x^i and x^j on the H_1 and H_2.

H_1: $w^T x^i + b = -1$

H_2: $w^T x^j + b = 1$

We know the distance of a point (x_1, y_1) from a line ax+by+c is $\frac{|ax_1+by_1+c|}{\sqrt{a^2+b^2}}$. The margin of the hyperplane is

$$
\begin{aligned}
l(w,b) &= d_+(w,b;x^i) + d_-(w,b;x^j) \\
&= \frac{|w^T x^i + b|}{||w||} + \frac{|w^T x^j + b|}{||w||} \\
&= \frac{|-1|}{||w||} + \frac{|1|}{||w||} \\
&= \frac{1}{||w||} + \frac{1}{||w||} \\
&= \frac{2}{||w||}
\end{aligned}
$$

To find a optimal hyperplane margin needs to be maximized. To maximize margin $||w||$ needs to be minimized.

Optimization Problem

The primal optimization problem

$$\begin{aligned} \min_{w} \quad & (\tfrac{1}{2})||w||^2 \\ \text{s.t.} \quad & y_i(w^T x^i + b) - 1 \geq 0 \quad i = 1, ..., n \end{aligned} \tag{4}$$

Lagrangian Dual Problem

$$\mathcal{L} = (\frac{1}{2}) w^T w - \sum_{i=1}^{n} \alpha_i [y_i(w^T x^i + b) - 1]; \qquad \alpha_i \geq 0 \tag{5}$$

The optimization is equivalent to find a saddle point of Lagrangian function $\mathcal{L}$ [3]. 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.

Minimization of Lagrangian

Gradient with respect to w

The gradient with respect to w is following:

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

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

After equating with zero we get our optimal w^*

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

Gradient with respect to b

The gradient with respect to b is

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

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

After equating with zero

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

Plugging in the value into Lagrangian

Substituting the value of w^* and $\sum_{i=1}^{n} \alpha_i y_i = 0$

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

$$\mathcal{L}(w^*, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j \tag{6}$$

Dual of the problem

Dual optimization problem

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x^i)^T x^j & \\ \text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0 & \\ \alpha_i \geq 0 & \qquad i = 1, ..., n \end{aligned} \tag{7}$$

This dual problem is a convex quadratic programming problem. Solving dual problem is equivalent to solving primal problem under KKT(Krush-Khun- Tucker) conditions. With the estimated optimal value of α by solving this problem w^* can be explicitly determined. 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.

Classification of testing data

Estimation of b^*

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^* and multiply with y_s

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

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

$$b^* = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{p \in S} \alpha_p^* y_p (x^p)^T x^s) \tag{8}$$

Classification of testing data

Estimation of classifier

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

$$y(x) = sign(\sum_S \alpha_s^* y_s (x^s)^T x^{test} + b^*) \tag{9}$$

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

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 [3].

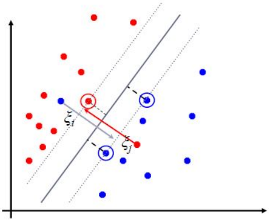

Figure: Soft Margin Classification: Classification error is allowed here for some points. Then try to minimize the error.

Hyperplanes

$$\begin{aligned} w^T x^i + b &\geq +1 - \xi_i \qquad \textit{for} \quad y_i = +1 \\ w^T x^i + b &\leq -1 + \xi_i \qquad \textit{for} \quad y_i = -1 \\ &\qquad\qquad \xi_i \geq 0 \quad \forall i \end{aligned} \tag{10}$$

By combining both of the equation we get

$$\begin{aligned} y_i(w^T x^i + b) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0 \quad \forall i \end{aligned} \tag{11}$$

The variable ξ_i are slack variables use to balance the missclassification in the set of inequalities.

Primal Problem

The formation of the optimization problem is

$$\min_{w,b,\xi} \quad (\frac{1}{2})w^T w + C\sum_{i=1}^{n} \xi_i \tag{12}$$

Subject to

$$\begin{aligned} y_i(w^T x^i + b) &\geq 1 - \xi_i \quad i = 1, ..., n \\ \xi_i &\geq 0 \quad i = 1, ..., n \end{aligned} \tag{13}$$

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 primal problem

Lagrangian function

Lagrangian of this primal problem is following $\mathcal{L}(w, b, \xi_i, \alpha_i, \nu_i)$=$(\frac{1}{2})w^T w + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i[y_i(w^T x^i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \nu_i \xi_i$ 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

$$\frac{d\mathcal{L}}{dw} = \frac{1}{2} \times 2w - \sum_{i=1}^{n} \alpha_i y_i x^i \tag{14}$$

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

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

However, $\nu_i \ge 0 \quad \forall i$ implies $\alpha_i \le C$.

Now substituting these into Lagrangian equation to get the Dual Lagrangian with constraints

plugging in the values

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

Simplification

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

Dual Lagrangian function

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

Dual Problem

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

Subject to

$$\begin{gathered} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1, ..., n \end{gathered} \tag{15}$$

Some Conditions

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 sample 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.

Calculation of b

b^* will be calculated in similar manner as hard margin SVMs. $b^* = \frac{1}{N_s} \sum_{s \in S}(y_s - \sum_{p \in S} \alpha_p^* y_p (x^p)^T x^s)$ One important feature is the set of Support Vectors used to calculate b is determined by finding the indices i where $0 < \alpha_i \leq C$.

SVMs classifier

The soft margin SVMs classifier will be $y(x) = sign(\sum_S \alpha_s^* y_s (x^s)^T x^{test} + b^*)$

Non linear data classification by kernel tricks

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

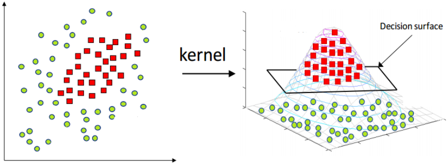

Figure: Kernel trick is mapping the training samples into high dimensional space where they are separated by a hyperplane

Kernel tricks are popular in different machine learning algoritms specially in Support Vector Machines[2]. 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.

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

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.

Kernalized Dual Problem

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j ((\phi(x^i))^T . \phi(x^j)) \tag{17}$$

Subject to the constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C \quad i = 1, ..., n \tag{18}$$

Estimation of parameters

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

$$b^* = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{p \in S} \alpha_p^* y_p (\phi(x^p))^T \phi(x^s)) \tag{19}$$

The SVMs classifier will be

$$y(x) = sign(\sum_{S} \alpha_s^* y_s (\phi(x^s))^T \phi(x^{test}) + b^*) \tag{20}$$

Some important features of Kernel Function

- 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.
- An appropriate kernel function can construct a classifier which is linear in feature space even though non-linear in original space [**?**].
- 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 $[\gamma = \frac{1}{2\sigma^2}]$

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 [4]. 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) [5].

Analysis results for Divorce predictor data set

The 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: Accuracy Report for different classification methods

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

So, by using kernel trick more specifically Gaussian radial basis kernel function we can predict about 97% of divorced cases.

Optimal hyperplane

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

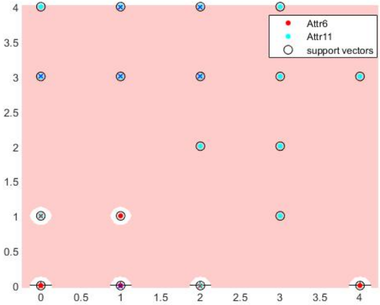

Figure: Optimal hyperplane of Divorce predictor data with attribute 6 and attribute 11

Analysis results for Cryotherapy data set

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:

Table: Accuracy Report for different classification methods

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

Optimal Hyperplane

The optimal hyperplane for this model is

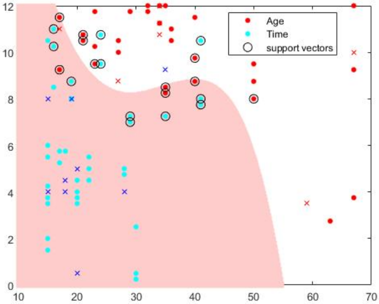

Figure: Optimal hyperplane of wart treatment result by using cryotherapy data with variables Age and Time

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.

Conclusion

Prelude

In this paper I have started with the brief literature review, discussion of basic concepts and definitions such as hyperplane, support vectors, margin etc.

Methodology

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).

Methodology

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.

Illustration

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. I have achieved an explicit and sound concept about how SVMs can be used in data classification and pattern recognition.

MATLAB code for linear classifier

Example (Code for linear SVMs)

```
function classifier = linearclassifier(X,y)
        n=size(X,1);
        %kernel or dot product of X's
        kernel=X*X'; % Linear kernel which is merely a
        %dot product of vector X's.
        cvx_begin
        cvx_precision high
        variable al(n)
        minimize ((0.5.*quad_form(y.*al, kernel))
                                   -ones(n,1)'*al);
        subject to
              y'*al == 0;
              al>=0;
        cvx_end
```

MATLAB code for linear classifier

Example (Code for linear SVMs)

```
alpha_star=al;

        tol=10^-5;
    alpha_star(alpha_star<tol)=0;
    X_support = X(alpha_star>0,:);
    y_support = y(alpha_star>0);
    alpha_star = alpha_star(alpha_star>0);
    kernel=X*X_support';
    b_star=mean(y-kernel*(alpha_star.*y_support));
    % New Model with only support vectors
    classifier = new_model(X_support, y_support, alpha_star, b
end
```

MATLAB code for nonlinear classifier

Example (Code for non linear SVMs)

```
function classifier = nonlinearclassifier(X,y,C,sigma)
      kernel=X*X';
      %
      n=size(X,1);
      %
      cvx_begin
      cvx_precision high
      variable al(n);
      minimize ((0.5.*quad_form(y.*al, kernel))
                               -ones(n,1)'*al);
      subject to
           al>=0;
           y'*al==0;
           al<=C; % Additional condition for non-linear case
      cvx_end
```

MATLAB code for nonlinear classifier

Example (Code for nonlinear SVMs)

```
alpha_star=al;
        % Removing data other than support vectors

        tol=10^-5;
    alpha_star(alpha_star<tol)=0;
    X_support = X(alpha_star>0,:);
    y_support = y(alpha_star>0);
    alpha_star = alpha_star(alpha_star>0);
    kernel=X*X_support';
    b_star=mean(y-kernel*(alpha_star.*y_support));
    classifier = new_model(X_support, y_support, alpha_star,
                                              b_star, sigma)
end
```

References

- 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).

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

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

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

- 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.

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

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

- 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.

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

- 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.

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

- 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.

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

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

- 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.

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

The End