

# BUILDING A BI-OBJECTIVE QUADRATIC PROGRAMMING MODEL FOR THE SUPPORT VECTOR MACHINE

Mohammed Zakaria Moustafa<sup>1</sup>, Mohammed Rizk Mohammed<sup>2</sup>, Hatem Awad Khater<sup>3</sup>, Hager Ali Yahia<sup>4</sup>

<sup>1</sup>Department of Electrical Engineering (Power and Machines Section) Alexandria University, Alexandria, Egypt

<sup>2</sup>Department of Communication and Electronics Engineering, Alexandria University, Alexandria, Egypt

<sup>3</sup>Department of Mechatronics, Faculty of Engineering, HORAS University, Damietta, Egypt

<sup>4</sup>Department of Communication and Electronics Engineering, Alexandria University, Alexandria, Egypt

## ABSTRACT

*A support vector machine (SVM) learns the decision surface from two different classes of the input points, in many applications there are misclassifications in some of the input points. In this paper a biobjective quadratic programming model is utilized and different feature quality measures are optimized simultaneously using the weighting method for solving our bi-objective quadratic programming problem. An important contribution will be added for the proposed bi-objective quadratic programming model by getting different efficient support vectors due to changing the weighting values. The experimental results, give evidence of the effectiveness of the weighting parameters on reducing the misclassification between two classes of the input points.*

## KEYWORDS

*Support vector machine (SVMs), Classification, Multi-objective problems, weighting method, Quadratic programming.*

## 1. INTRODUCTION

Support Vector Machines (SVMs) are classification techniques developed by Vapnik at the end of '60s [1]. The theory of support vector machines (SVMs) is a new classification technique and has drawn much attention on this topic in recent years [6]. So, the SVMs have been deeply improved to be applied in many different applications.

In many applications, SVM has been shown to provide higher performance than traditional learning machines [6]. SVMs are known as maximum margin classifiers, since they find the optimal hyperplane between two classes as shown in Fig.1, defined by a number of support vectors [4].

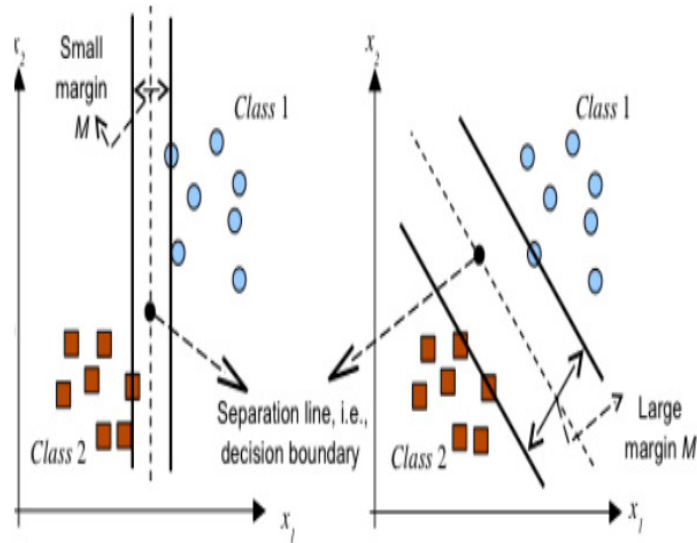


Figure 1. maximization of the margin between two classes

The well-known generalization feature of the technique is mainly due to the introduction of a penalty factor, named  $C$  that allows to prevent the effects of outliers by permitting a certain amount of misclassification errors.

In this paper, the idea is to apply the multi-objective programming technique for developing the set of all efficient solutions for the classification problem with minimum errors. The weighting method is used to solve the proposed multi-objective programming model. The remainder of this paper is organized as follows. Section 2 describes a brief review for the SVM. Section 3 describes the proposed multiobjective model for the Support Vector Machine. NEXT, section 4 presents three numerical examples. Section 5 provides our general conclusions.

## 2. SUPPORT VECTOR MACHINES

**SVM** is an efficient classifier to classify two different sets of observations into their relevant class as shown in figure 2 where there are more than straight line separates between the two sets. SVM mechanism is based upon finding the best hyperplane that separates the data of two different classes of the category. The best hyperplane is the one that maximizes the margin, i.e., the distance from the nearest training points [2]. Support vector machine has been utilized in many applications such as biometrics, chemoinformatics, and agriculture. SVM has penalty parameters, and kernel parameters that have a great influence on the performance of SVM [3]. We review the basis of the theory of SVM in classification problems [7].

Let a set  $S$  of labeled training points

$$S = (y_1, x_1), (y_2, x_2), \dots, (y_l, x_l) \quad (1)$$

Where  $x_i \in \mathcal{R}^N$  belongs to either of two classes and is given a label  $y_i = \{-1, 1\}$  for  $i = 1, \dots, l$ .

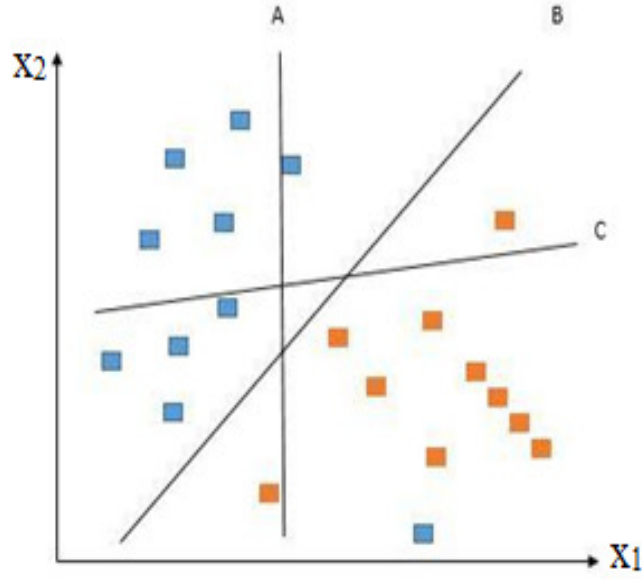


Figure 2. Data classification using support vector machine

In some cases, to get the suitable hyperplane in an input space, mapping the input space into a higher dimension feature space and searching the optimal hyperplane in this feature space.

Let  $z = \varphi(x)$  denotes the corresponding feature space vector with mapping  $\varphi$  from  $\mathcal{R}^N$  to a feature space  $z$ . We wish to find the hyperplane

$$w \cdot z + b = 0 \quad (2)$$

defined by the pair  $(w, b)$  according to the function

$$f(x_i) = \text{sign}(w \cdot z_i + b) = \begin{cases} 1, & \text{if } y_i = 1 \\ -1, & \text{if } y_i = -1 \end{cases} \quad (3)$$

where  $w \in z$  and  $b \in \mathcal{R}$ . For more precisely, the equation will be

$$\begin{cases} (w \cdot z_i + b) \geq 1, & \text{if } y_i = 1 \\ (w \cdot z_i + b) \leq -1, & \text{if } y_i = -1, \end{cases} \quad i = 1, \dots, l \quad (4)$$

For the linearly separable set  $S$ , a unique optimal hyperplane is determined for which the margin between the projections of the training points of two different classes is maximized.

For the data that are not linearly separable as shown in Fig. 3, the previous analysis can be generalized by introducing some non-negative variables  $\xi_i \geq 0$  then,

$$y_i(w \cdot z_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, l. \quad (5)$$

The term  $\sum_{i=1}^l \xi_i$  can be thought of as some measure of the amount of misclassifications. The optimal hyperplane problem is then regarded as the solution to the problem

$$\text{minimize } \frac{1}{2} w \cdot w + C \sum_{i=1}^l \xi_i$$

$$\text{subject to } y_i(w \cdot z_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, l \quad (6)$$

$$\xi_i \geq 0, \quad i = 1, \dots, l$$

Where  $C$  is a constant. The parameter  $C$  can be regarded as a regularization parameter [5]. SVM algorithms use a set of mathematical functions that are defined as the kernel.

The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions (i.e. linear, nonlinear, polynomial, radial basis function (RBF) and sigmoid).

Basically, the training part consists in finding the best separating plane (with maximal margin) based on specific vector called support vector. If the decision is not feasible in the initial description space, the space dimension is increased using the kernel functions and a hyperplane that will be the decision separator is determined.

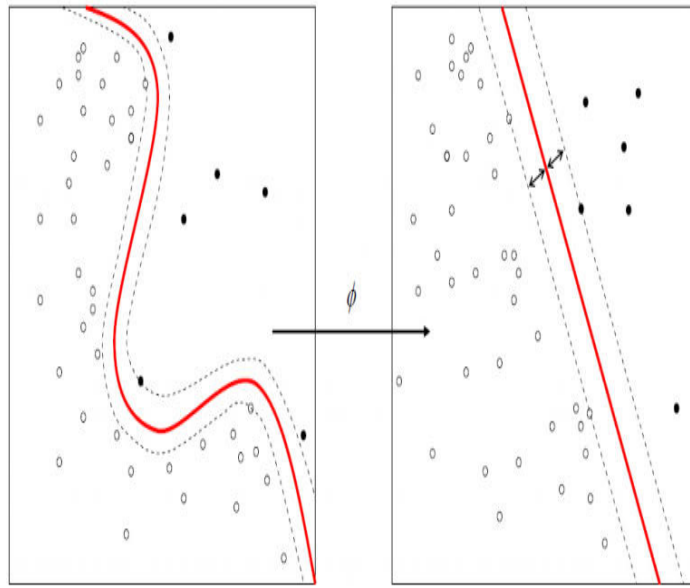


Figure 3. linearly separable and nonlinearly separable

### 3. FORMULATION OF THE BI-OBJECTIVE QUADRATIC PROGRAMMING MODEL OF SVM

In this section, a detailed description about the idea and formulation of the bi-objective programming model for the SVM are introduced. SVM is a powerful tool for solving classification problems, but due to the nonlinearity separable in some of the input data, there is an error in measuring the amount of misclassification. This leads us to add another objective function for the previous model in section 2 to be in the following form

$$\begin{aligned}
 & \text{Min } \| w \|^2, \\
 & \text{Min } \sum_{i=1}^l \xi_i \\
 & \text{Subject to} \quad (7) \\
 & y_i(w \cdot x_i + b) \geq 1 + \xi_i \quad , i = 1, 2, \dots, l \\
 & \xi_i \geq 0 \quad , i = 1, 2, \dots, l
 \end{aligned}$$

This problem is a bi-objective quadratic programming problem. The first objective is to maximize the gap between the two hyperplanes which used to classify the input points. The second objective is to minimize the errors in measuring the amount of misclassification in case of nonlinearity separable input points. The problem 7 can be solved by the weighting method to get the set of all efficient solutions for the classification problem. The right choice of weightage for each of these objectives is critical to the quality of the classifier learned, especially in case of the class imbalanced data sets. Therefore, costly parameter tuning has to be undertaken to find a set of suitable relative weights [10].

### 3.1 The Weighting Method

In this method each objective  $f_i(X), i = 1, 2, \dots, k$ , is multiplied by a scalar weigh  $w_i \geq 0$  and  $\sum_{i=1}^k w_i = 1$ . Then, the k weighted objectives are summed to form a weighted-sums objective function [8].

$$\text{Assume } W \text{ as } \left\{ \begin{array}{l} w \in R^k: w_i \geq 0, \\ i = 1, 2, \dots, k \\ \text{and } \sum_{i=1}^k w_i = 1 \end{array} \right\} \quad (8)$$

be the set of nonnegative weights. Then the weighting problem is defined as:

$$P(W): \text{Min } \sum_{i=1}^k w_i f_i$$

$$\text{Subject to } M = \left\{ \begin{array}{l} X \in R^n: g_r(X) \leq 0, \\ r = 1, 2, \dots, m \end{array} \right\}. \quad (9)$$

Then, in this paper the weighting method takes the form

$$\text{Inf } z = w_1 \|w\|^2 + w_2 \sum_{i=1}^l \xi_i$$

Subject to

$$\begin{aligned} y_i(w \cdot x_i + b) &\geq 1 + \xi_i, \quad i = 1, 2, \dots, l \\ \xi_i &\geq 0, \quad i = 1, 2, \dots, l \\ w_1 &> 0, w_2 &\geq 0 \\ w_1 + w_2 &= 1 \end{aligned} \quad (10)$$

Here we use “Inf” instead of “Min” since the set of constraints is unbounded, where  $w_1 \neq 0$ . Also, we avoid the redundant solutions by adding the constraint  $w_1 + w_2 = 1$ .

#### 4. NUMERICAL EXAMPLES

Using python programming language, the previous problem is solved and the effect of different values of the weighting parameters is described. The data set that is used in these examples consist of 51 points and each point has two features, table 1 shows part of this data.

Table 1. Description of part of datasets used in our study.

X1	X2	Y
1.9643	4.5957	1
2.2753	3.8589	1
2.9781	4.5651	1
2.932	3.5519	1
3.5772	2.856	1
0.9044	3.0198	0
0.76615	2.5899	0
0.086405	4.1045	0

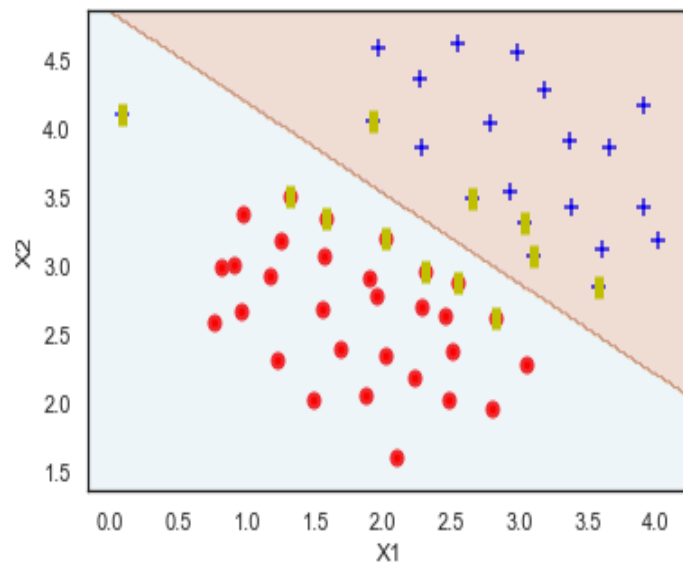


Figure 4.  $w_2 = \frac{1}{2}, w_1 = \frac{1}{2}$ , number of support vectors = 12

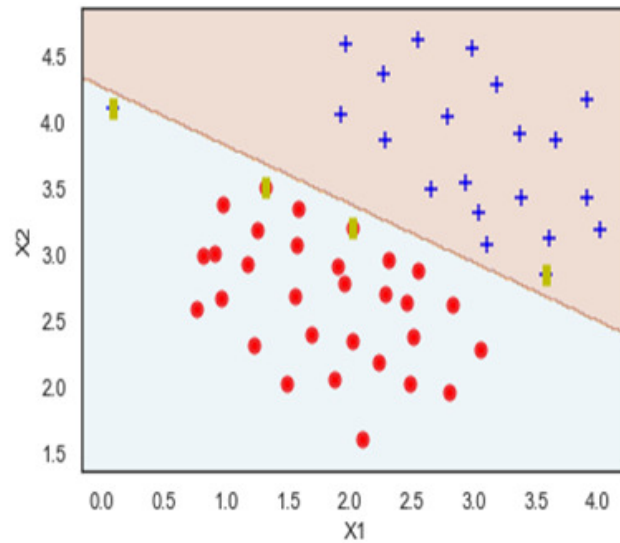


Figure 5.  $w_2 = \frac{20}{21}, w_1 = \frac{1}{21}$ , number of support vectors = 4

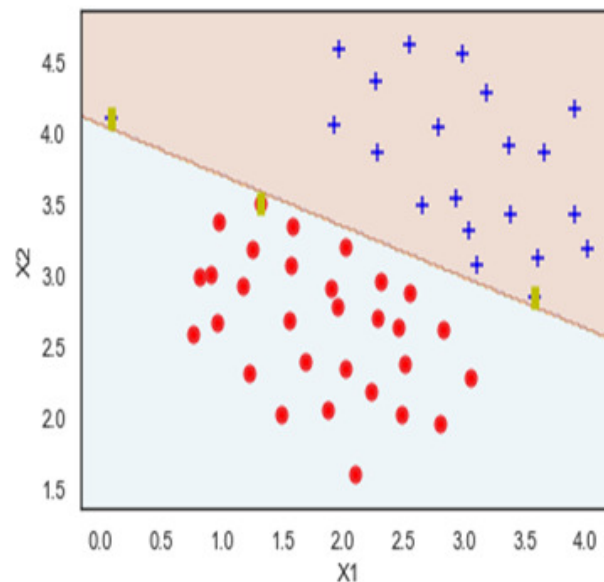


Figure 6.  $w_2 = \frac{99}{100}, w_1 = \frac{1}{100}$ , number of support vectors=3

So, the previous results, by using different values of weighting parameters, show how these parameters effect on the performance of SVM. For the first values of  $w_1$  &  $w_2$ , there is one point of the blue set can't be classified to its set, the second values make this point closes to its set and the third values of  $w_1$  &  $w_2$ , this point can be joined to its set. So, when the weighting parameter  $w_2$  is increased, the misclassification and the number of support vectors will be reduced as shown in Fig. 5 and Fig. 6. There are good reasons to prefer SVM<sup>s</sup> with few support vectors ( $SV^s$ ). In the hard-margin case, the number of  $SV^s$  is an upper bound on the expected number of errors made by the leave-one-out procedure [9].

So, we can control the performance of SVM according to the requirements by adjusting the values of the weighting parameters.

## 5. CONCLUSIONS

This paper introduced the multi-objective programming technique for developing the set of all efficient solutions for the classification problem with minimum errors and how to solve the proposed multi-objective programming model by using the weighting method. The experimental evaluation was carried out using 51 datasets, each one has two features. The experimental results show the effect of the weighting parameters on the misclassification between two sets.

The future work can include a construction of a utility function to select the best compromised hyperplane from the generated set of the efficient solutions and building a fuzzy bi-objective quadratic programming model for the support vector machine.

## REFERENCES

- [1] Cortes, Corinna; Vapnik, Vladimir N (1995) "Support vector networks" (PDF). *Machine learning*. 20 (3):273297. CiteSeerX 10.1.1.15.9362. DOI:10.1007/BF00994018.
- [2] Asmaa Hamad<sup>1,3(B)</sup>, Essam H. Houssein<sup>1,3</sup>, Aboul Ella Hassanien<sup>2,3</sup>, and Aly A. Fahmy<sup>2</sup>: Hybrid Grasshopper Optimization Algorithm and Support Vector Machines for Automatic Seizure Detection in EEG Signals. Faculty of Computers and Information, Minia University, Minya, Egypt. January 2018. DOI: 10.1007/978-3-319-74690-6\_9.
- [3] Alaa Tharwat<sup>1;\_</sup>, Thomas Gabel<sup>1</sup>, Aboul Ella Hassanien<sup>2;\_</sup> Parameter Optimization of Support Vector Machine using Dragon\_y Algorithm. Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany ,Faculty of Computers and Information, Cairo University, Egypt. January 2018 DOI: 10.1007/978-3-319-64861-3\_29.
- [4] Gray, D., Bowes, D., Davey, N., Sun, Y., Christianson, B.: Using the Support Vector Machine as a Classification Method for Software Defect Prediction with Static Code Metrics. In: Palmer Brown, D., Draganova, C., Pimenidis, E., Mouratidis, H. (eds.) EANN 2009. *Communications in Computer and Information Science*, vol. 43, pp. 223–234. Springer, Heidelberg (2009).
- [5] Chun-Fu Lin and Sheng-De Wang: Fuzzy Support Vector Machines. Article in *IEEE Transaction on neural networks* March 2002. DOI:10.1109/72.991432.
- [6] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol.2, no.2, 1998.
- [7] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol.20, pp.273-297, 1995.
- [8] Chankong V. and Haimes Y.Y., *Multi-objective Decision-Making: Theory and Methodology* (North Holland Series in System Science and Engineering, 1983).
- [9] Yaochu Jin (Ed.), *Multi-objective Machine Learning Studies in Computational Intelligence*, Vol. 16, pp. 199-220, Springer-Verlag, 2006.
- [10] Shounak Datta and Swagatam Das, Multiobjective Support Vector Machines: Handling Class Imbalance With Pareto Optimality, *IEEE Transactions on Neural Networks and Learning Systems*, 2019. DOI:10.1109/TNNLS.2018.2869298.