A BI-OBJECTIVE MODEL FOR SVM WITH AN INTERACTIVE PROCEDURE TO IDENTIFY THE BEST COMPROMISE SOLUTION

Mohammed Zakaria Moustafa[1], Mohammed Rizk Mohammed[2], Hatem Awad Khater[3], Hager Ali Yahia[2]

[1]Department of Electrical Engineering (Power and Machines Section)
ALEXANDRIA University, Alexandria, Egypt

[2]Department of Communication and Electronics Engineering,
ALEXANDRIA University, Alexandria, Egypt

[3]Department of Mechatronics, Faculty of Engineering, Horus University, Egypt

ABSTRACT

A support vector machine (SVM) learns the decision surface from two different classes of the input points, there are misclassifications in some of the input points in several applications. In this paper a bi-objective 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 numerical examples, give evidence of the effectiveness of the weighting parameters on reducing the misclassification between two classes of the input points. An interactive procedure will be added to identify the best compromise solution from the generated efficient solutions.

KEYWORDS

Support vector machine (SVMs); Classification; Multi-objective problems; weighting method; Quadratic programming; interactive approach.

1. INTRODUCTION

Support Vector Machines (SVMs) are a classification technique 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 [5]. Since then the technique has been deeply improved, being applied in many different contexts.

In many applications, SVM has been shown to provide higher performance than traditional learning machines [5]. SVMs are known as maximum margin classifiers, since they find the optimal hyperplane between two classes as shown in figure1, defined by a number of support vectors [3].

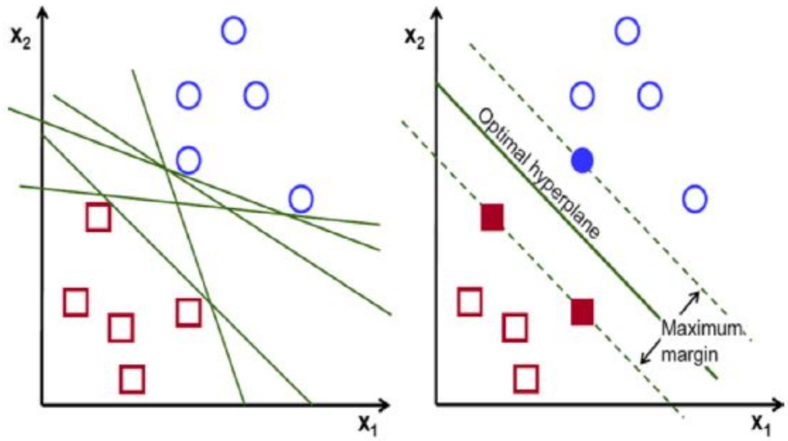

Figure 1: maximization of the margin between two classes

The well-known generalization feature of the technique is mainly due to the introduction of a penalty factor, named C that allows us 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 remaining sections are organized as follows. An abstraction of SVM is covered in section 2. Section 3 describes the proposed multi-objective model for the Support Vector Machine. NEXT, section 4 presents three numerical examples. Section 5 provides our general conclusions and future work.

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 a category.

The best hyperplane is the one that maximizes the margin, i.e., the distance from the nearest training points.

SVM has penalty parameters, and kernel parameters that have a great influence on the performance of SVM [2]. We review the basis of the theory of SVM in classification problems [6].

Let a set S of labelled training points

$$(y_1, x_1) \ldots (y_l, x_l) \qquad (1)$$

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,…, l.

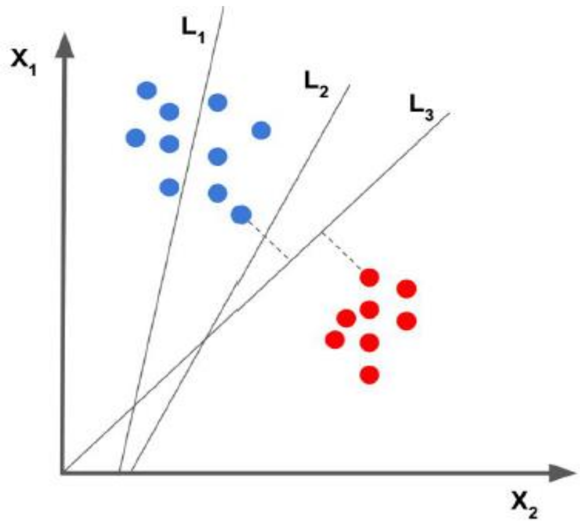

Figure 2: Data classification using support vector machine

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

$$w.z + b = 0 \qquad (2)$$

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

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

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

$$\begin{cases} (w.z_i + b) \geq 1, & if y_i = 1 \\ (w.z_i + b) \leq -1, & if y_i = -1, \end{cases} i = 1, \ldots, l \qquad (4)$$

For the linearly separable set S, we can find a unique optimal hyperplane 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 figure 3, the previous analysis can be generalized by introducing some nonnegative variables $\xi_i \geq 0$ then,

$$y_i\,(w.\,z_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l. \tag{5}$$

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

$$\begin{gathered} minimize \frac{1}{2} w.w + C \sum_{i=1}^{l} \xi_i \\ subjecttoy_i(w.z_i + b) \geq 1 - \xi_i \;, \\ i = 1, \ldots, l \\ \xi_i \;\geq 0, \quad i = 1, \ldots, l \end{gathered} \tag{6}$$

where,C is a constant. The parameter C can be regarded as a regularization parameter [4]. 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. For example, linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

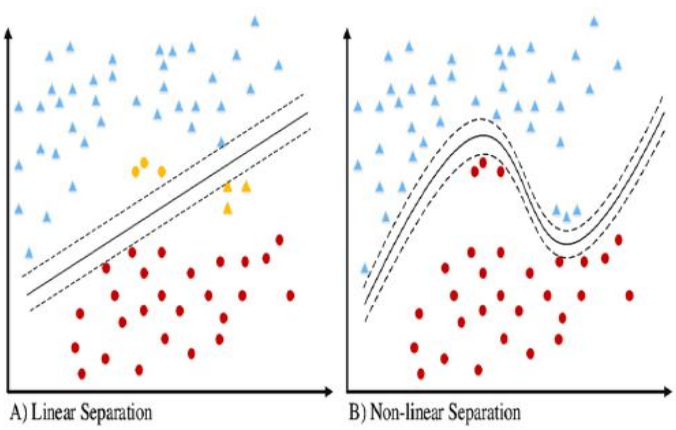

Figure 3: linearly separable and nonlinearly separable

3. The BI-Objective Quadratic Progamming Model Formulation of SVM

In this section, we make a detail description about the idea and formulation of the bi-objective programming model for the SVM. 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 form

$$\text{Min } \| w \|^2,$$

$$\text{Min } \sum_{i=1}^{l} \xi_i$$

$$\text{Subject to} \qquad (7)$$

$$y_i(w.x_i + b) \geq 1 + \xi_i \quad ,i = 1,2,\ldots,l$$

$$\xi_i \geq 0 \quad ,i = 1,2,\ldots,l$$

This problem is a bi-objective quadratic programming problem. For the first objective, maximizing the gap between the two hyperplanes which used to classify the input points. For the second objective, minimizing the errors in measuring the amount of misclassification in case of nonlinearity separable input points [11].

The previous problem 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 [9].

3.1. The Weighting Method

In this method each objective $f_i(X)$, $i = 1,2,\ldots, k$, is multiplied by a scalar weigh $w_i \geq 0$ *and* $\sum_{i=1}^{k} w_i = 1$

1. Then, the k weighted objectives are summed to form a weighted-sums objective function [7].

$$Assume\ W \text{ as} \begin{Bmatrix} w \in R^k : w_i \geq 0, \\ i = 1,2, \dots, k \\ and \sum_{i=1}^{k} w_i = 1 \end{Bmatrix} \tag{8}$$

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

$$P(W): Min \sum_{i=1}^{k} w_i f_i$$
$$\text{Subject to } M = \begin{Bmatrix} X \in R^n : g_r(X) \leq 0, \\ r = 1,2, \dots, m \end{Bmatrix}. \tag{9}$$

Then, in this paper the weighting method takes the form

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

$$\text{Subject to}$$

$$y_i(w.x_i + b) \geq 1 + \xi_i \quad , i = 1,2, \dots, l$$

$$\xi_i \geq 0 \quad , i = 1,2, \dots, l$$

$$w_1 > 0, w_2 \geq 0 \tag{10}$$

$$w_1 + w_2 = 1$$

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

3.2. An Interactive Procedure to Identify the Best Compromise Solution

3.2.1. Introduction

By solving a multi-objective optimization problem, we get a set of efficient solutions. The efficient set in many cases may contain infinite number of points. Now, the decision maker problem is, how to choose a point from the efficient set?

Because of the difficulty in choosing one of this set, the decision maker needs a specific technique to do this. One of such techniques is the interactive programming approach. The use of interactive algorithm for multicriteria optimization has been proposed by several authors

.

The purpose of such interactive algorithms is to present to the decision maker, in a series of meetings, a choice of efficient alternatives which are in some sense representative of all those available.

Over these series of meetings, he must explore his preferences amongst presented alternatives and finally choose one which is to be admitted as satisfactory [10].

The interactive approaches are characterized by the following procedures:

Step 1: Generate a solution or group of solutions (preferably feasible and efficient)
Step 2: Interact with decision maker to obtain his reaction to the solution. Then the decision maker inputs information to the solution procedures.
Step 3: Repeat step 1&2 until termination either by the algorithm itself or by the decision maker.

For the version of our bi-objective (SVM) model which applies to determine the best compromise solution,

we need the following hypothesis (after the interaction with the decision maker):

The best compromise solution for the set of the generated efficient solution is that efficient one corresponding to

$$\min_{w_1,w_2} N^- \leq \min_{w_1,w_2} N^+$$

Where, N^- is the number of support vectors of the negative class,
N^+ is the number of support vectors of the positive class.

We must notice that this hypothesis can be reversed according to the preference of the decision maker (see Yaochu Jin,2006) [8].

4. EXPERIMENTAL RESULTS

The previous problem is solved by using the python program. The used dataset in these examples is consisted of 51 input points and each point has two features X1&X2 as shown in table 1. These examples show the effect of the different values of the weighting parameters.

Table 1: Part of datasets used in this study.

Feature1 X1	Feature2 X2	Y
4.015	3.1937	1
3.3814	3.4291	1
3.9113	4.1761	1
2.7822	4.0431	1
2.4482	2.6411	0
2.7938	1.9656	0
2.091	1.6177	0
2.5403	2.8867	0
2.5518	4.6162	1
3.3698	3.9101	1
3.1048	3.0709	1
1.9182	4.0534	1

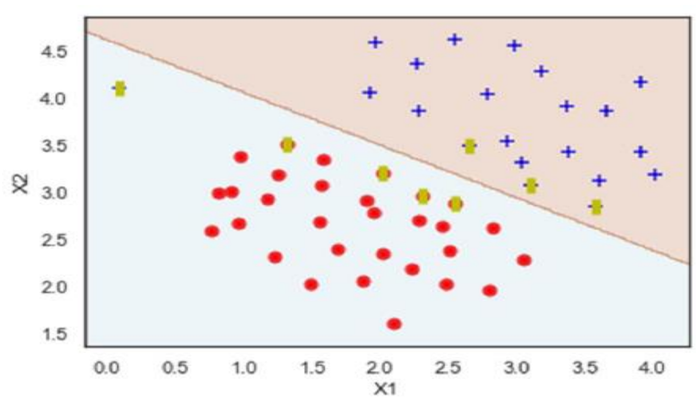

Figure 4: $w_2 = \frac{3}{4}, w_1 = \frac{1}{4}$, number of support vectors = 8

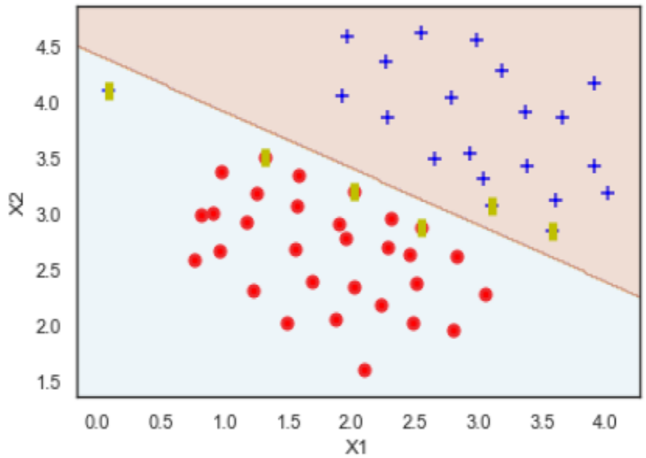

Figure 5: $w_2 = \frac{9}{10}, w_1 = \frac{1}{10}$, number of support vectors = 6

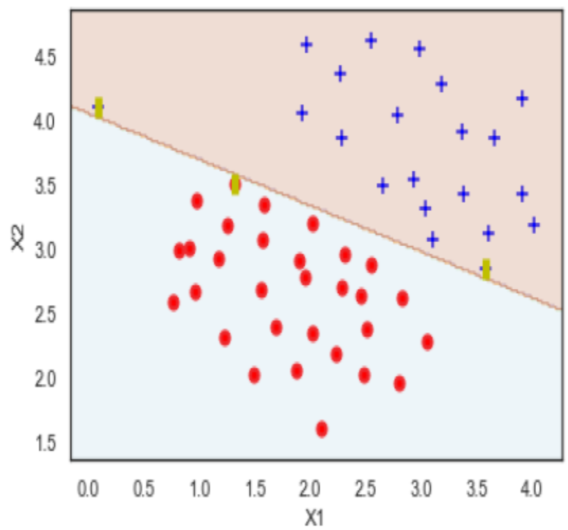

Figure 6: $w_2 = \frac{89}{90}, w_1 = \frac{1}{90}$, number of support vectors=3

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 $\boldsymbol{w_1}$& $\boldsymbol{w_2}$ 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 $\boldsymbol{w_1}$&$\boldsymbol{w_2}$, this point can be joined to its set. So, when the weighting parameter $\boldsymbol{w_2}$ is increased the misclassification and the number of support vectors will be reduced as shown in figures 5&6.

There are good reasons to prefer SVMs with few support vectors (SVs). In the hard-margin case, the number of SVs (#SV) is an upper bound on the expected number of errors made by the leave-one-out procedure [8].

According to our hypothesis that presented in section 3.2, the best compromise solution is that corresponding to $w_2 = \frac{89}{90}, w_1 = \frac{1}{90}$

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. An interactive procedure is added to identify the best compromise hyperplane from the generated efficient set.

Our future work is to build 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] Alaa Tharwat1;_, Thomas Gabel1, Aboul Ella Hassanien2;_ 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.

[3] 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).

[4] 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.

[5] C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, vol.2, no.2,1998.

[6] C. Cortes and V. N. Vapnik, “Support vector networks,” Machine Learning, vol.20,pp.273-297,1995.

[7] Chankong V. and Haimes Y.Y., Multi-objective Decision-Making: Theory and Methodology (North Holland Series in System Science and Engineering, 1983).

[8] Yaochu Jin (Ed.), Multi-objective Machine Learning Studies in Computational Intelligence, Vol. 16, pp. 199-220, Springer-Verlag, 2006.

[9] 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.

[10] Zoints S.,S. Y. Prasad and M. H. Karwan , Use of Convex Cones in Interactive Multiple Objective Decision Making, Management Science, 43 (5) (May 1997) 723-734.

[11] Mohammed Zakaria Moustafa, Mohammed Rizk Mohammed, Hatem Awad Khater and Hager Ali Yahia, A bi-objective Model for SVM with an interactive procedure to identify the best compromise solution, 8th International Conference on Artificial Intelligence, Soft Computing (AISC 2020), DOI:10.5121/csit.2020.100208.