Examining Of Support Vector Machines Algorithm And Modeling It With Monte-Carlo Simulation

Hüsamettin EKEN
Electric and Electronic Engineering
Ankara University
Ankara, Turkey
husamettineee@gmail.com

***Abstract*—Examining of support vector machines algorithm and modeling of different kernels of support vector machines on one data set using Monte Carlo Simulation.**

***Keywords*—vectors, support vectors, algorithm, Monte Carlo, kernels, lagrange function, MATLAB, simulation**

I. INTRODUCTION

In this study, we will talk about SVM (Support Vector Machines) algorithm, which is one of the artificial intelligence classification algorithms. The SVM algorithm, whose main purpose is to classify two classes of data, is also used to predict. In the basic logic, it was created to separate data using a line or plane, but different kernels were used for more complex data.

II. MATHEMATICAL EXPLANATION AND METHODS

The algorithm is divided into 3 steps to make it easier to understand. There are no margins of error on the data. There are some margins of error on the data and after that, we use kernel methods. The algorithm starts to be analyzed with data without any margin of error. Then we think there's some margin of error. After that, when we could not separate data with a hyperplane, we use different kernels and try to separate data more efficient.

A. Data Which Can Separable Without Error

We start with a question first. What do we want? We find the answer to this question as the widest range and correctly separated data. These answers evoke a function. Lagrange function. The Lagrange function consists of two parts. The first part contains the function which will be maximum or minimum and the second part contains the conditions that will be fulfilled when performing these operations. After realizing these parts, we must express these two parts mathematically.

As shown in Figure-1, our goal is to find the largest range and classify the data correctly.

First of all, we will express our data mathematically. My x values in the equation represent the points in my dataset and the y values represent their labels. Since the SVM algorithm is a binary classification algorithm, I only have two tags. I will separate my two data groups with a line. So I'm going to use a straight equation.You can see equations at the bottom.

$$\left(x_i, y_i \right), \quad i = 1, 2, \ldots, n$$

$$x_i \in \mathbb{R}^p \quad y_i \in \{-1, +1\}$$

$$\left\{ X \in \mathbb{R}^p : \beta^\top X + \beta_0 = 0, \beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R} \right\}$$

Eq-1

I'll use two more lines parallel to the line above. These line equations will be my basic elements in finding the widest range.You can see equations at the bottom.

$$\beta^\top X + \beta_0 - \Delta = 0 \qquad \beta^\top X + \beta_0 + \Delta = 0$$

Eq-2

Since we have two equations, we are trying to make these equations easier to solve. To do this we multiply the equations by tags. As a result, we get an equation. This equation is our condition in the lagrange function.You can see the last equation at Eq-3.

$$\begin{gathered} y_i = 1 \longrightarrow \beta^\top x_i + \beta_0 - 1 \geq 0 \\ y_i = -1 \longrightarrow \beta^\top x_i + \beta_0 + 1 \leq 0 \\ y_i \left(\beta^\top x_i + \beta_0\right) \geq 1 \end{gathered}$$

Eq-3

Since we have two equations, we are trying to make these equations easier to solve. To do this we multiply the equations by tags. As a result, we get an equation. This equation is our condition in the lagrange function.

We are trying to find the length shown in Figure-3.You can see length formula at Eq-4

$$\|\bar{x}\| \cos\theta = \frac{\beta^\top \bar{x}}{\|\beta\|} = \frac{1-\beta_0}{\|\beta\|}$$

Eq-4

We are trying to find the length shown in Figure-4.You can see length formula at Eq-5

$$\|\tilde{x}\| \cos\theta = \frac{\beta^\top \tilde{x}}{\|\beta\|} = \frac{-1-\beta_0}{\|\beta\|}$$

Eq-5

As shown in Figure-5, we obtain the longest distance by subtracting the two equations we found earlier.You can see latest form of distance formula at Eq-6. We want to maximize it.

$$\frac{1-\beta_0}{\|\beta\|} - \frac{-1-\beta_0}{\|\beta\|} = \frac{2}{\|\beta\|}$$

Eq-6

We've got all the necessary formulas, but we're gonna have to sort it out. We have an absolute function. We're making it into a form that we can solve by arranging it. You can see arranged function at Eq-8.

$$\begin{gathered} \frac{2}{\|\beta\|} \\ y_i \left(\beta^\top x_i + \beta_0\right) \geq 1, \quad i = 1, \ldots, n \end{gathered}$$

Eq-7

$$\begin{gathered} \frac{1}{2}\beta^\top \beta \\ y_i \left(\beta^\top x_i + \beta_0\right) \geq 1, \quad i = 1, \ldots, n \end{gathered}$$

Eq-8

B. Data With Some Error

We looked data without error and now we will some error. You can see errored value at Figure-6. We create new equations by including the margin of error in the old line equations. We add the sum of the error margin values to the function we want to minimize. As a result, we get the equations seen in Eq-10. The c value here is used to optimize our accepted error margin.

$$\begin{aligned} y_i \left(\beta^\top x_i + \beta_0\right) &\geq 1 - \varepsilon, \quad \varepsilon = 0 \\ y_i \left(\beta^\top x_i + \beta_0\right) &\geq 1 - \varepsilon, \quad 0 < \varepsilon < 1 \\ y_i \left(\beta^\top x_i + \beta_0\right) &\geq 1 - \varepsilon, \quad \varepsilon \geq 1 \end{aligned}$$

Eq-9

$$\begin{gathered} \frac{1}{2}\beta^\top \beta + c \sum_{i=1}^{n} \varepsilon_i^d \\ y_i \left(\beta^\top x_i + \beta_0\right) \geq 1 - \varepsilon_i, \quad i = 1, \ldots, n \\ \varepsilon_i \geq 0, \quad i = 1, \ldots, n \end{gathered}$$

Eq-10

When we solve these equations by using the lagrange function, we find the most appropriate separator line for the data set with the wrong values.

C. Kernels

So far, we have examined the data sets that we can solve with line help. Where our data cannot be resolved with line help, we use kernels. Kernels are actually separators that use polynomial, gaussian or sigmoid functions instead of line equations.You can see a dataset that you can not separate with a line equation at Figure-7

In the Eq-11 you see the lagrange function. For different kernels, our equations or conditions change here.

$$\mathcal{L}(\beta, \beta_0; \alpha) = \frac{1}{2}\beta^\top \beta - \sum_{i \in A} \alpha_i \left(y_i \left(\beta^\top x_i + \beta_0\right) - 1\right)$$

Eq-11

You can see output of different kernels output at Figure-8.

D. Multiclass SVM

With the SVM algorithm, we can also separate data with multiple data sets. Here are two approaches. The first method is to classify the data set by comparing the data set in binary with the others after processing as much as the binary combination of the data set. The second method is to select a data set and classify the two data sets after accepting the rest as a set. In the second method, we perform the calculation process as much as the number of data sets.

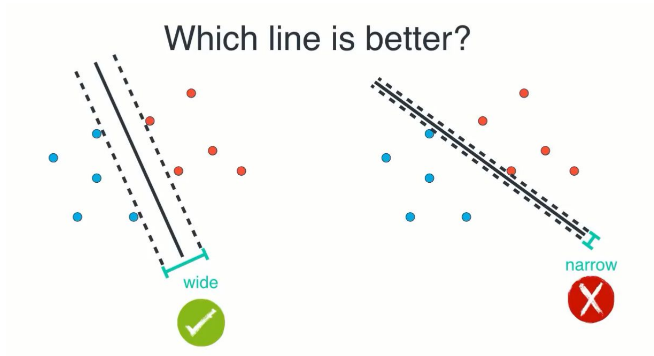

FIGURE 1. Hyperplanes correctly separate data sets, but the widest range is the best choice for the SVM algorithm.

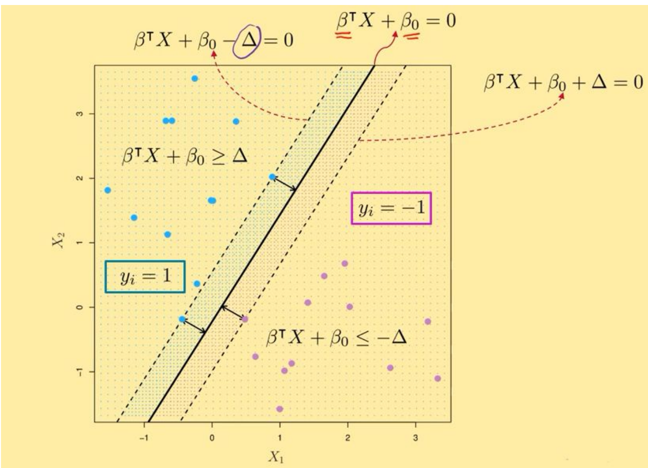

FIGURE 2. It visualize Eq-1 and Eq-2

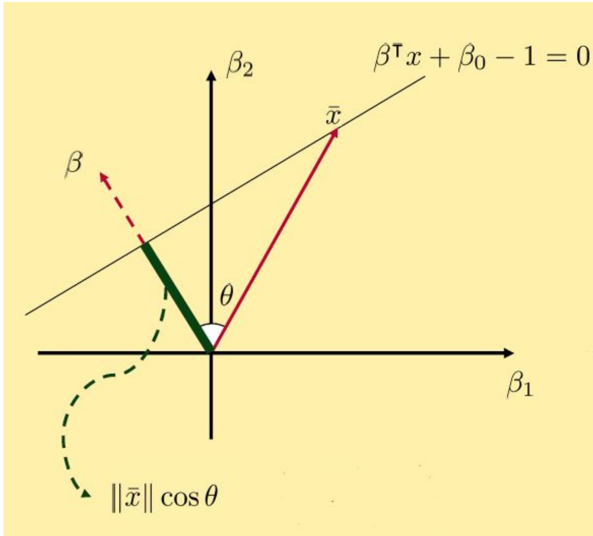

FIGURE 3. Upper parallel line of two parallel line

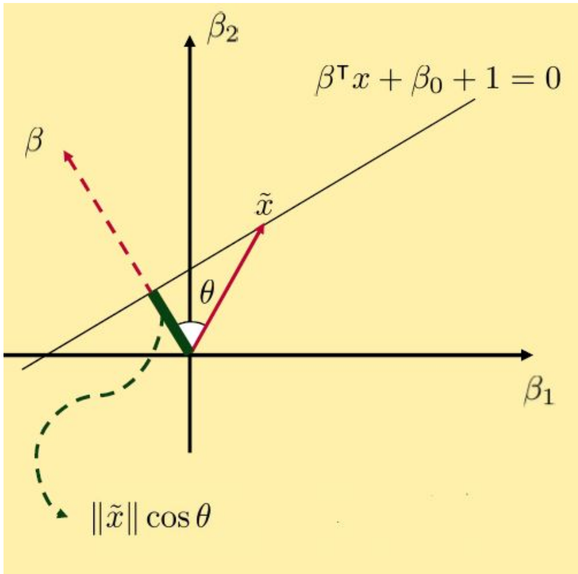

FIGURE 4. Lower parallel line of two parallel line

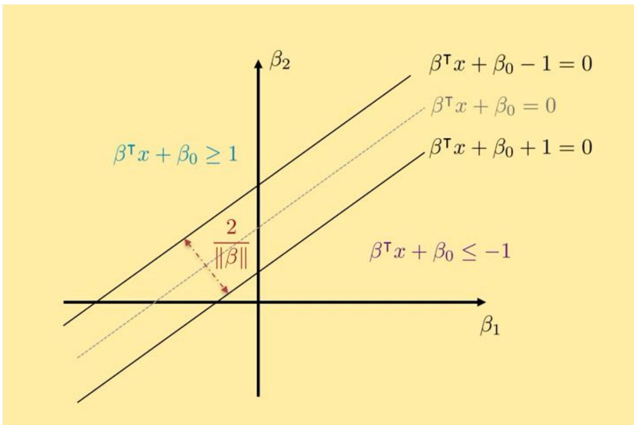

FIGURE 5. We find largest length

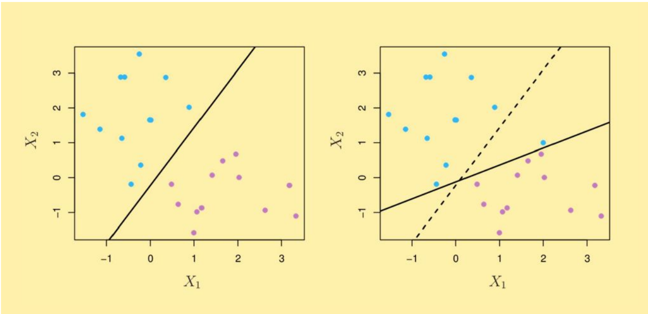

FIGURE 6. There is errored value at second image bu we want old line like image 1 and we accept some errors.

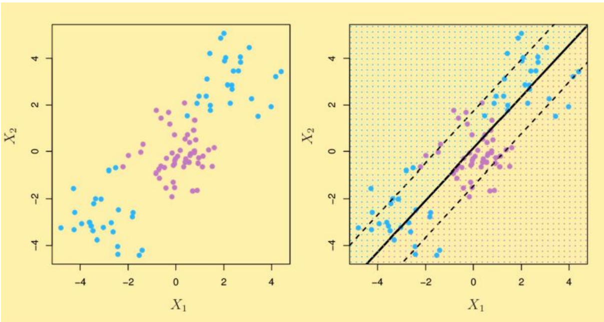

FIGURE 7. There is dataset that can not separable with a line equation.

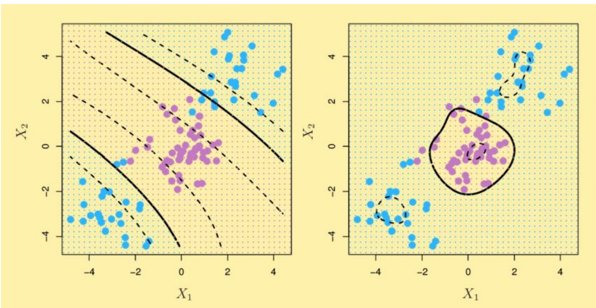

FIGURE 8. There is dataset that classified with different kernels.

III. MATLAB SIMULATION

In this section, I have to add something about my data set before I begin. When I first spoke to my teacher, he told me that I could analyze the sound of numbers from zero to nine, but I failed. Then I chose one and two tones by reducing this data set. When I got the data ready for the SVM algorithm, I realized that the size of my data set was very large. When I tried to process this dataset on my computer, I got an error on my computer. Therefore, I could not use the data set that I initially agreed with my teacher. After that, I used a dataset called fisheriris which was available in MATLAB. This dataset is a dataset containing the lengths and widths of the petals and petals of 3 different flowers. I first separated this dataset using a linear kernel. Then I separated it using gaussian and polynomial kernels. And then I noticed something. You can see the separation results in Figure 9 10 11. What I noticed is that all of them have a correct separation rate of one hundred percent. I decided using Monte Carlo simulation which would be a better choice for this data set. I added incorrect and random data to twenty percent of my dataset and tested it two hundred times for each kernel. As a result, I obtained the same results as shown in Figure 12 13 14. I understand that the gaussian kernel gives more accurate results for this dataset.

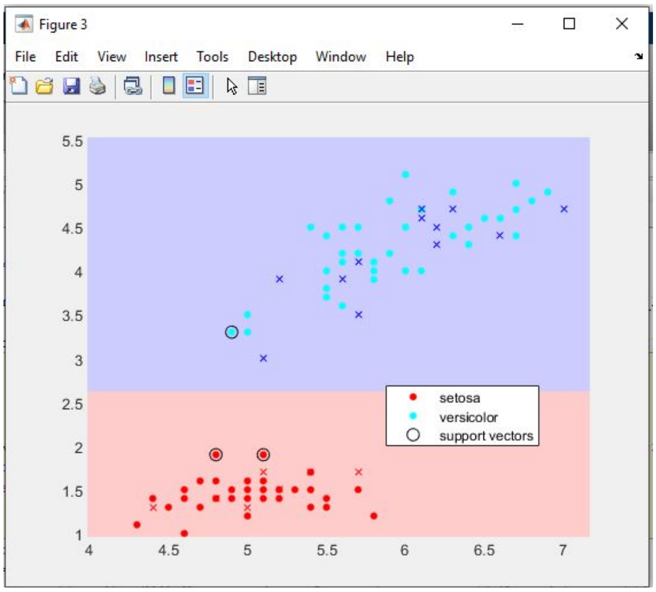

FIGURE 9. Matlab fisheriris dataset is separated with linear kernel and visualized

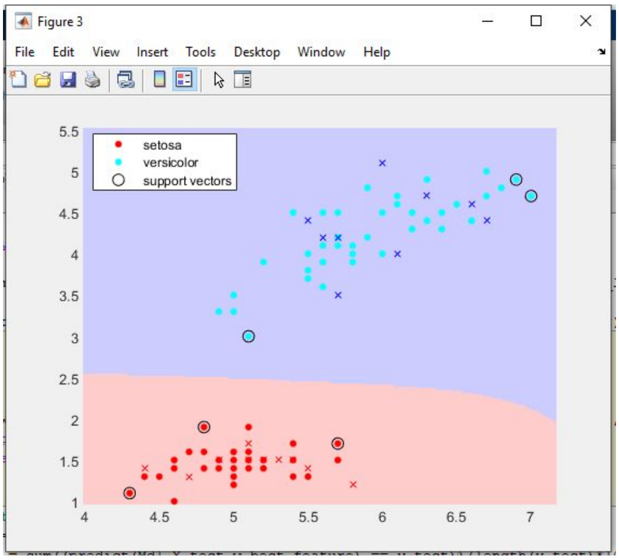

FIGURE 10.Matlab fisheriris dataset is separated with gaussian kernel and visualized

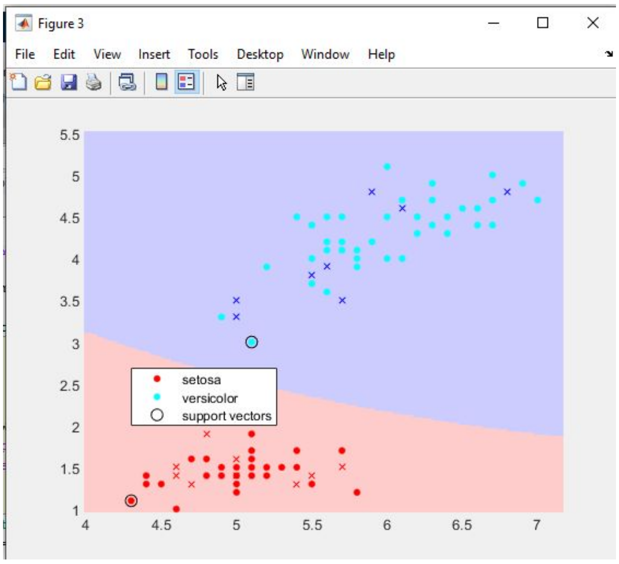

FIGURE 11. Matlab fisheriris dataset is separated with polynomial kernel and visualized

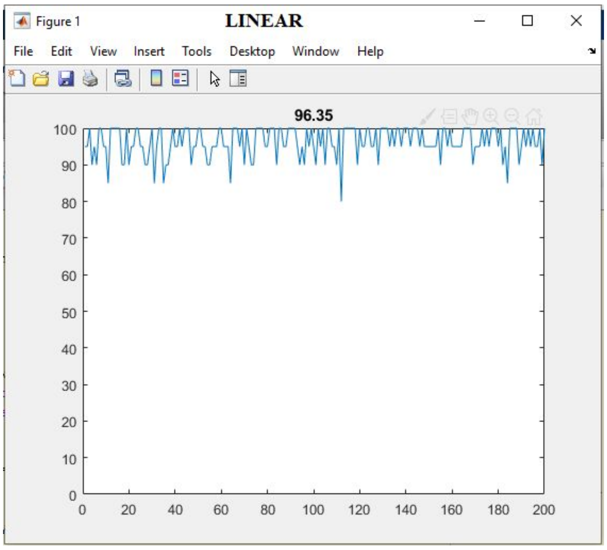

FIGURE 12. Some errored values added to fisheriris dataset and Monte Carlo simulation is applied to SVM algoirthm outputs. The success of linear kernel on the dataset is %96.35

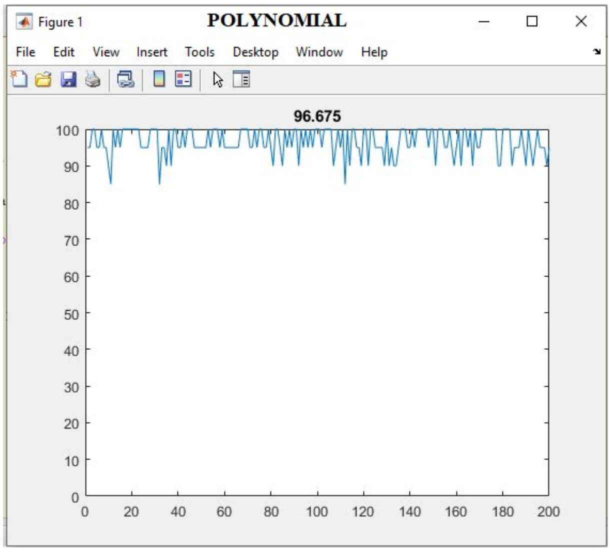

FIGURE 13. Some errored values added to fisheriris dataset and Monte Carlo simulation is applied to SVM algorithm outputs. The success of polynomial kernel on the dataset is %96.675

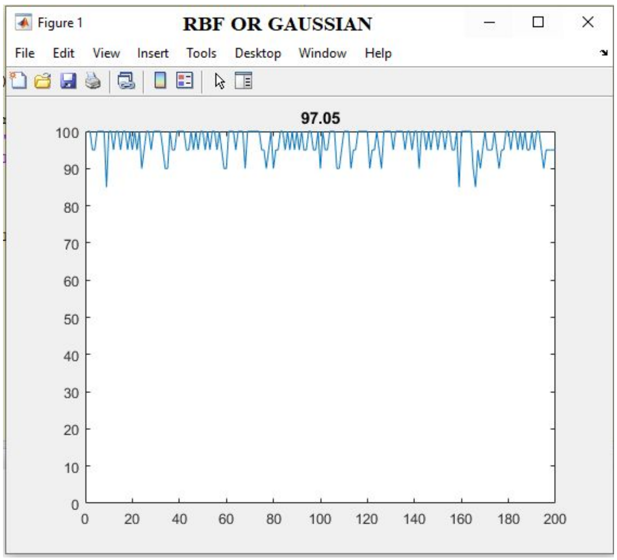

FIGURE 14. Some errored values added to fisheriris dataset and Monte Carlo simulation is applied to SVM algorithm outputs. The success of gaussian kernel on the dataset is %97.05

IV. Conclusion

As a result, in this study we understand the working logic of the SVM algorithm. We have examined how it is implemented on different datasets. We understand how an algorithm designed to separate two groups of data can be used to classify more than two groups. We have learned how to implement the SVM algorithm on MATLAB. We implemented the Monte Carlo simulation. During the presentation, I learned that I cannot process audio data directly and that I need to extract features.

References

[1] https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html

[2] https://www.mathworks.com/help/stats/fitcsvm.html

[3] https://www.youtube.com/watch?v=vJ_WL9aYfNI

[4] https://www.youtube.com/watch?v=VRGZm1wPesY

[5] https://www.youtube.com/watch?v=SIU_GWaEiIs

[6] https://www.turkhackteam.org/yapay-zeka/1658261-support-vector-machine-s-svm.html

[7] https://medium.com/@ankitnitjsr13/math-behind-support-vector-machine-svm-5e7376d0ee4d

[8] https://www.youtube.com/watch?v=Lpr__X8zuE8

[9] https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8

[10] https://www.youtube.com/watch?v=N1vOgolbjSc

[11] https://www.youtube.com/watch?v=efR1C6CvhmE

[12] http://www.veridefteri.com/2018/12/13/tahmin-ve-cikarim-9-destek-vektor-makineleri/

[13] https://www.coursera.org/learn/machine-learning