IÉSEG SCHOOL OF MANAGEMENT

OPTIMIZATION TECHNIQUES GROUP PROJECT

Support Vector Machine

FEBRUARY 20, 2022

BY:
MOHAMMAD HADI ALIPOUR MOTLAGH
WAJIH ARFAOUI
HARIKRISHNAN GOPALAKRISHNAN
AHMAD OMAR NAKIB
ENITA OMUVWIE
INDERPREET RANA

Table of Contents

INTRODUCTION

In this project we have chosen to implement the Support Vector Machine (SVM) model on the “Iris” data set to classify the data. The Support Vector Machine algorithm is an ML algorithm that finds a hyperplane in the N-dimensional space where N is the number of features to classify data points. We are aiming to extract two different classes clearly and distinctively from our dataset. In this report we will walk through the motivation behind the model and its application, we will also walk through the implementation of the model itself, our decision variables, parameters and constraints, our objective function and finally the outcomes of the model while discussing its accuracy and what it means in the context of our data.

MOTIVATION OF THE MODEL OVERVIEW OF SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used mainly for binary classification problems, even though it can also be used for regression as well. The objective of SVM is to find a N- dimension hyperplane that distinctly classifies data points.

The problem that arises is that we have infinite hyperplanes that can be applied, but our aim is to come out with the hyperplane that has the maximum distance from data points in the two classes, so we can ensure that the new data points can be classified with more confidence.

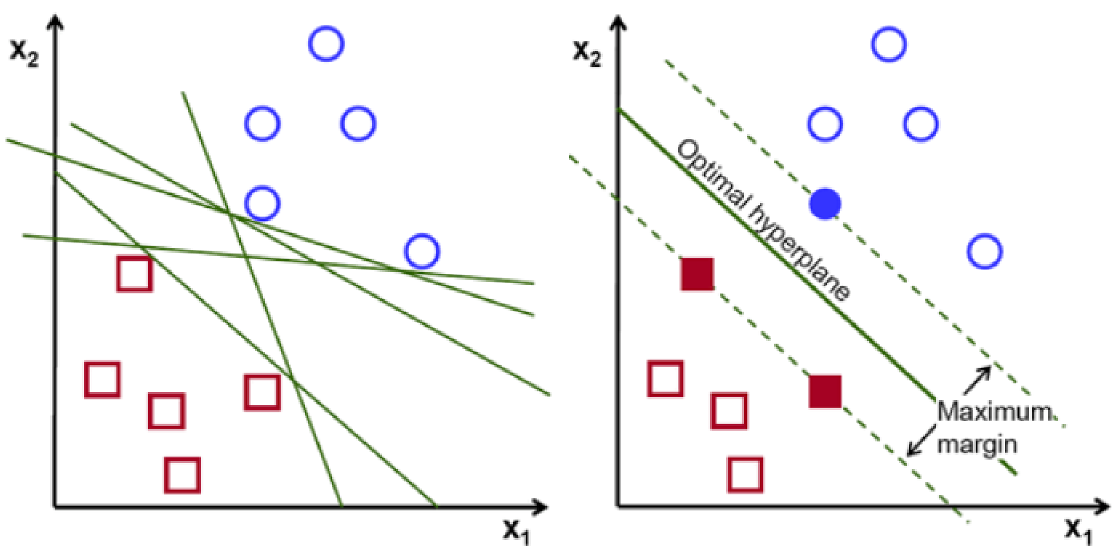


Figure 1: Shows how to find the hyperplane

In to find the most appropriate hyperplane, which means maximize the margin of the classifier, we rely on support vectors that are the closest data points to the hyperplane and can influence its position and orientation.

MATHEMATICAL CONCEPT OF SVM

As what we already discussed, we are looking to maximize the margin from the Support Vectors to the hyperplane, while minimizing the normal vector: $\boldsymbol{w}^T\boldsymbol{w} = \|\boldsymbol{w}\|$.

.

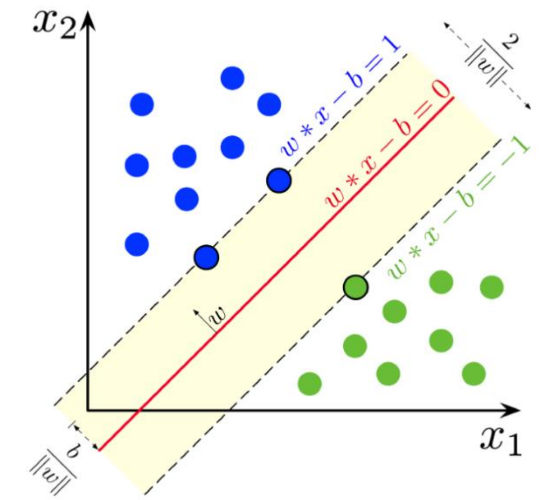


Figure 2: Shows how to maximize the margin

To put it another way, our objective can be attained by finding the optimal values of $\boldsymbol{w}$ and $\boldsymbol{b}$ yielding to the highest accuracy. But, when looking at real life examples, we can notice that the data can barely be separated linearly due to the noise.

The outliers caused by this noise can affect the positioning of the hyperplane, so to avoid this, we need to find a hyperplane that tolerates at a certain extent misclassification. To do so, the used constraint in linearity separable case:

$$\boldsymbol{z_i(w^T x_i + b) \geq 1}$$

Is relaxed by inserting a slack $\boldsymbol{s_i} \geq \boldsymbol{0}$ for each observation $\boldsymbol{x_i}$. This $\boldsymbol{s_i}$ indicates the deviation of $\boldsymbol{x_i}$ from the corresponding hyperplane for $\boldsymbol{i} = \boldsymbol{1}, \ldots, \boldsymbol{n}$.

Based on the above considerations, the modified constraint is:

$$\boldsymbol{z_i(w^T x_i + b) + \; s_i \geq 1 \;\; for \; i = 1, \ldots, n}$$

Consequently, $\sum_{i=1}^{n} s_i$ represents the total deviation error, which is considered as the upper bound for the total number of misclassified data points. Simultaneously, we can add a hyperparameter $v > 0$ in order to regulate the influence of s_i, since it ensures achieving the best ratio between the margin with the highest width and the lowest number of misclassified data points.

Therefore, the optimization problem is represented as follows:

$$min_{(w,b,s)} \; v \sum_{i=1}^{n} s_i + \frac{1}{2} \|w\|^2$$

$$s.to: \quad z_i\left(w^T x_i + b\right) + s_i \geq 1$$

$$z_i \geq 0, \; s_i \geq 0, b \geq 0$$

$$for \; i = 1, \ldots, n$$

Where v denotes the regularization hyperparameter.

s_i denotes the distance of x_i from the separating hyperplane.

b denotes a bias parameter

$\|w\|^2 = w^T w$

z_i denotes a variable that equals 1 if x_i belongs to **class 1** and -1 if x_i belongs to **class 2**

OUR DATA SET OF CHOICE "IRIS"

We have decided to go with the "iris" dataset in order to implement Support Vector Machine since the "iris" dataset is a clean data set that creates an optimal environment for the implementation of SVM. "iris" is a data set with 150 cases (rows) and 5 variables (columns) named Sepal. Length, Sepal. Width, Petal. Length, Petal. Width, and Species.

We are using a hyper parameter and we have tried multiple values and the optimal result found was at the parameter value 2

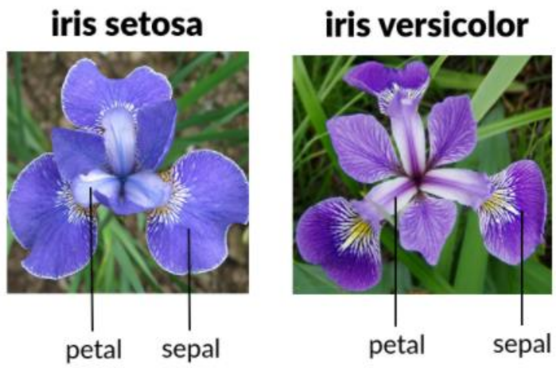
IMPLEMENTATION

The features are: **Sepal. Length**, **Sepal. Width**, **Petal. Length**, **Petal. Width**, and **Species**.

We are applying SVM in an attempt to predict the type of species which are Iris Setosa or Versicolor

CLASSES:

Iris Setosa, Versicolor



OUR .MOD FILE:

```
##Set variables
#VARS  = sepal length, speal width, petal lenth and petal width
#Species = observations
```

- Setting our variables
- Setting our observations (in this case the species of iris)

```
#Set assign parametres
param x {SPECIES,VARS} >= 0;
param z {SPECIES};
param c;

var y{SPECIES} >= 0;
var w{VARS} >= 0;
var g >= 0;
```

- We proceed by assigning our parameters

```
#equation to minimize the error
minimize Error:
      c*sum{i in SPECIES} y[i] + 0.5*(sum{j in VARS} w[j]^2);
```

- We define our equation to minimize the error

```
#constrain to split the species in 2 classes
#if greater then 1 then class 1 else class 2
subject to Type {i in SPECIES}:
   z[i]*(sum {j in VARS} x[i,j]*w[j]- g) + y[i] >= 1;

#normalized IRIS data
data;
```

- We add the constraint to split the species into 2 classes

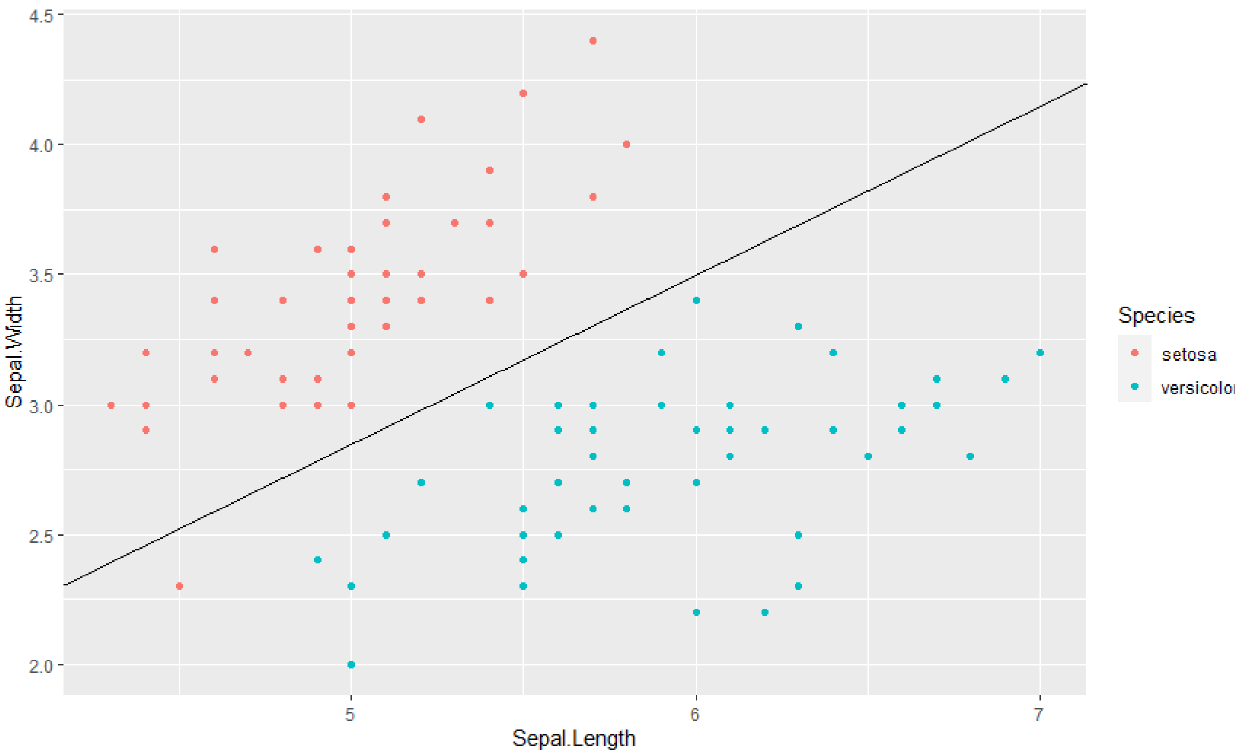
```
set SPECIES := a1 a2   a3   a4   a5   a6   a7   a8   a9   a10   a11   a12   a13   a14   a15   a16   a17   a18   a19   a20   a21   a22   a23   a24   a25   a26   a27   a28   a29   a3

set VARS := w1 w2 w3 w4;
```

- We set the “SPECIES” and vars “w1”,”w2”,”w3” and ”w4”

```
param x:
            w1      w2      w3      w4     :=
      a1    0.296296296 0.625 0.097560976 0.058823529
      a2    0.222222222 0.416666667 0.097560976 0.058823529
      a3    0.148148148 0.5   0.073170732 0.058823529
      a4    0.111111111 0.458333333 0.12195122  0.058823529
      a5    0.259259259 0.666666667 0.097560976 0.058823529
```

- Finally, we define parameter X. Note: this is the head of the “iris” data frame

GRAPHICAL CLASSIFICATION IN R:



CONCLUSION:

Applying SVM through 57 iterations resulted in an Objective function i104.2430651 Nonlinear evals: obj = 70, grad = 69.

REFERENCES:

- https://scikit-learn.org/stable/modules/svm.html
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://github.com/EgonFerri/SVM_in_ampl
  https://www.youtube.com/watch?v=efR1C6CvhmE
- https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/
- https://www.researchgate.net/publication/301876055_Multilevel_Weighted_Support_Vector_Machine_for_Classification_on_Healthcare_Data_with_Missing_Values
- https://ampl.com/BOOK/CHAPTERS/24-refman.pdf
- https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html
- Support vector machine with AMPL Research paper by Dr. Stefano Nasini