Support Vector Machine

Subham Roy
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
subhamro@buffalo.edu

1 Overview

The Support Vector Machine (SVM) is a generalization of a simple and intuitive classifier called the Maximal Margin Classifier (MMC). Though MMC is elegant and simple, it unfortunately is not applicable to most data sets, since it requires that the classes be separable by a linear boundary. So, we introduce the concept of Support Vector Classifier, an extension of the MMC that can be applied in a broader range of cases. Support vector machines are intended for the binary classification setting in which there are two classes and also, SVM has a close connection with other statistical methods such as logistic regression.

2 About Support Vector Machine

The support vector classifier, sometimes called a soft margin classifier is based on a hyperplane that does not perfectly separate the two classes, in the interest of -

- Greater robustness to individual observations, and
- Better classification of most of the training observations.

It could be worthwhile to misclassify a few training observations in order to do a better job in classifying the remaining(or test) observations.
The support vector classifier classifies a test observation depending on which side of a hyperplane it lies. The hyperplane is chosen to correctly separate most of the training observations into the two classes, but may misclassify a few observations.

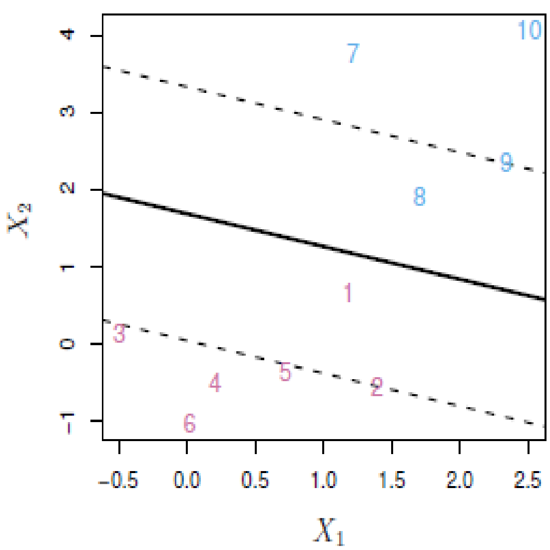

Fig: 1

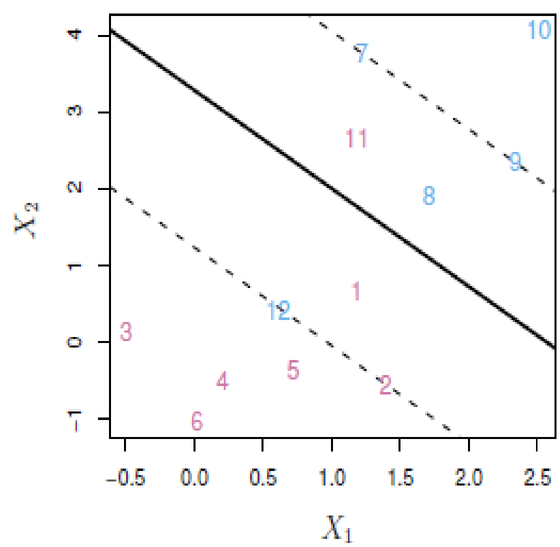

Fig: 2

Fig: 1 - A support vector machine was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Pink observations: Observations 3, 4, 5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observations: Observations 7 and 10 are on the correct side of the margin, observation 9 is on the margin, and observation 8 is on the wrong side of the margin. No observations are on the wrong side of the hyperplane.
Fig: 2 - Same as left panel with two additional points, 11 and 12. These two observations are on the wrong side the hyperplane and the wrong side of the margin.

Details of the Support Vector –

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,\, M}{\text{maximize}} \quad M$$

$$\text{subject to} \ \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C,$$

It is the solution to the optimization problem where C is a nonnegative tuning parameter, M is the width of the Margin; we seek to make this quantity as large as possible, and lastly, $\epsilon 1, \ldots, \epsilon n$ are slack variables that allow individual observations to be on slack the wrong side of the margin or the hyperplane.
Once we have solved these equations and inequalities, we then classify a test observation x^*, by simply determining on which side of the hyperplane it lies. That is, we classify the test observation based on the sign of $f(x^*) = \beta_0 + \beta_1 x^*_1 + \cdots + \beta_p x^*_p$.

3 Experiment

In this experiment, I used the IRIS dataset. Initially, we start with loading the Dataset and checking the dimensions and the number of variables within the Dataset. In this case, the dataset has 5 features and 150 rows.

Dataset description –

	Sepal.Length	Sepal.Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Fig: Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Sepal.Length  150 non-null    float64
 1   Sepal.Width   150 non-null    float64
 2   Petal_Length  150 non-null    float64
 3   Petal_Width   150 non-null    float64
 4   Species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Fig: Dataset Summary – Column Datatypes

Plot of the Dataset-

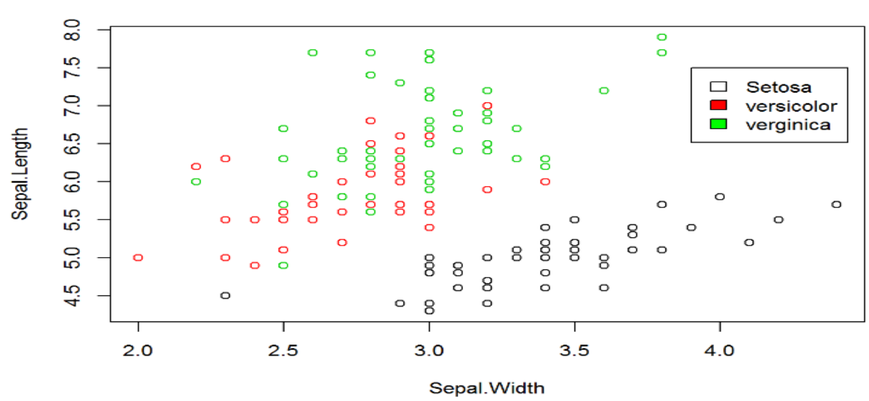

Fig: 3

Next step is we perform data preprocessing. We perform the following steps before we can be certain that we can apply the classifier model. This is almost similar for all the techniques we apply. First, we check for missing data, then check for the spread of the data that means where the Mean, Median of the data, are there any outliers, we take care of that.

In this case study, the dataset had no Null values or outliers. One thing we noticed; our dataset had variables of similar scales. So, we are not implementing any scaling features over the dataset. We now plot the data w.r.t. two different features i.e., Sepal.Length and Sepal.Width. We assigned three different colors for three different species class – Setosa, Versicolor and Verginica. Now, we implement the SVM model to linearly classify 3 different class. Let us analyze the result, the outcome of the model on the training and test set.

4 Analysis

We find out a lot of information about our prediction from this model. Firstly, the model accuracy for all the predictions is close to 80%, which is good considering the data was difficult to linearly classify. The below plot depicts the linear classification on the dataset. Later we implemented non-linear kernel SVM i.e., radial basis function, we gave high accuracy above 90%, but resulted into overfitting which according to my understanding is not ideal–

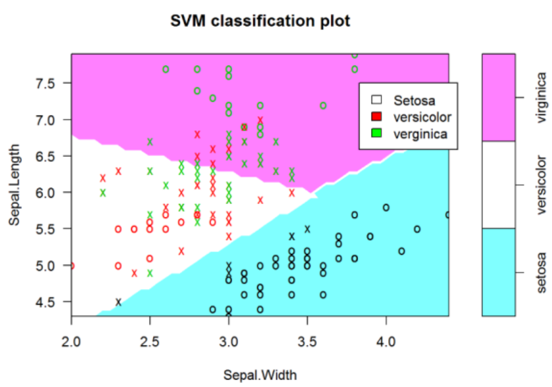

Fig: 4.a. (Linear Boundary)

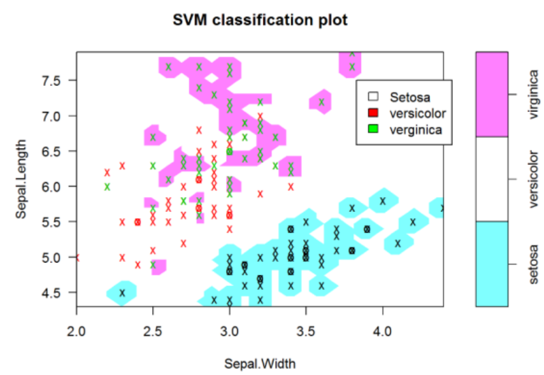

Fig: 4.b. (Radial Boundary)

References

[1] Gareth James [at], Daniela Witten [aut], Trevor Hastie [aut, cre], Rob Tibshirani [aut], Balasubramanian Narasimhan [ctb], *Introduction to Statistical Learning, Second Edition*.

[2] Tom M. Mitchell, *Machine Learning: A multistrategy approach, 1997 Edition*.