Université Jean Monnet

Fundamentals of Machine Learning

Practical Session-2 Report

Support Vector Machines

Team Members:
Karthik Bhaskar
Rohil Gupta

Supervisor:
Amaury Habrard

March 13, 2018

Introduction

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Tasks Completed

The following sections explains the tasks completed with regard to SVM's and knowledge gained by completing these tasks.

Task-1 : Little Warm-Up

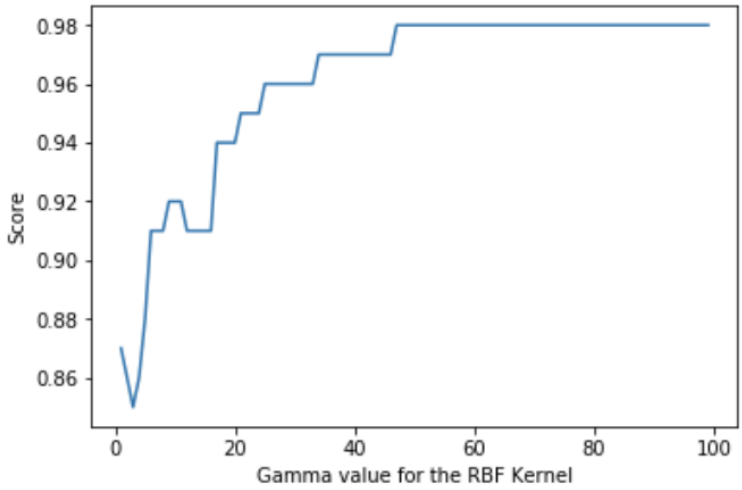
The first demo `http://cs.stanford.edu/people/karpathy/svmjs/demo/` allowed us to compare the effect of RBF kernel and linear kernel. It lets us set the value of sigma that can be tuned to a large and a small value . The large value of sigma limits the SVM decision boundary to learn something very complex and the small value has an opposite effect. The linear kernel is not able to learn non-linearly separable set of examples, so using RBF kernel becomes a good choice for non-linearly separable examples.

The second demo `https://www.csie.ntu.edu.tw/~cjlin/libsvm/` lets us compare different kernels than can be tuned by various option bars like (-t) lets us specify the kernel function, (-d) lets us give the degree of the polynomial kernel etc. It gives a similar effect on the on the SVM decision boundaries as it was seen in the first demo.

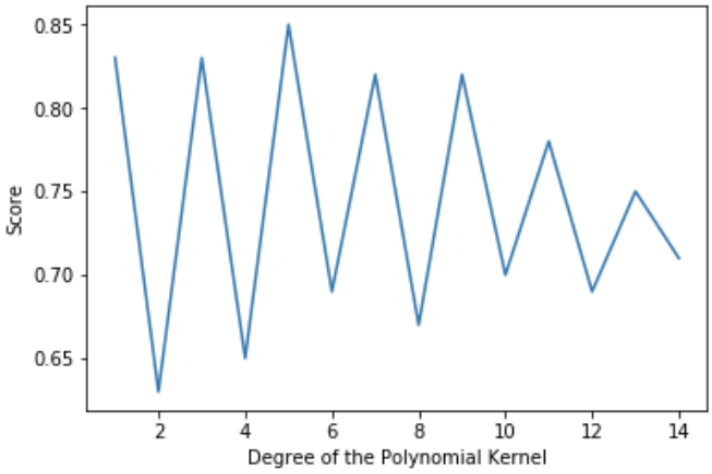
Task-2 : SVM with Scikit-Learn

In this section we learned the various parameters available in Scikit-Learn library that can be tuned for a SVM setting.

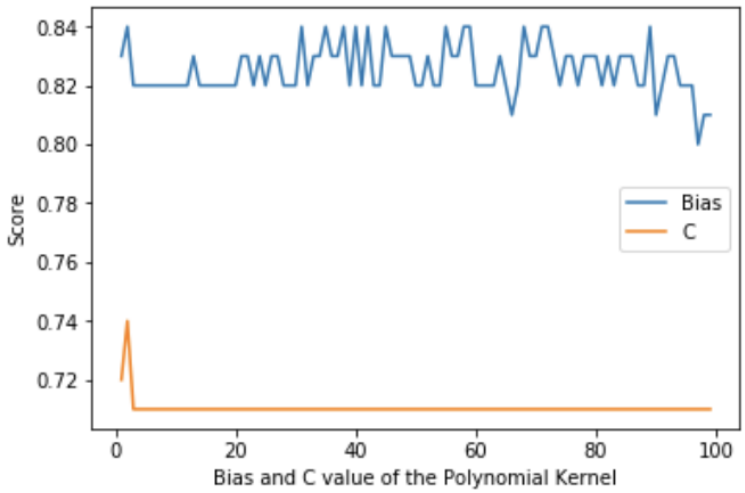
- At first, we randomly generated a sample of points using make classification function in Scikit-Learn which gives us many in-built setting to get a desired set of points or dataset for further analysis. The points generated are non linearly separable.
- SVM model is built with the default parameters and as shown in the Jupyter notebook learns fairly good decision boundary with an accuracy of 0.82
- When a linear kernel is introduced the accuracy drops and the decision boundary misclassify many examples.
- When an RBF kernel is used with a very low value of gamma, the accuracy of the model drops because we are limiting SVM to learn something very complex, hence the decision boundary is simple. The graph below shows the effect of increasing value of gamma on the accuracy of the learned model.



- When a polynomial kernel is used with high degree the accuracy drops significantly because the high degree of polynomial kernel gives the freedom to the decision boundary of SVM to learn something complex, but it comes with cost of decrease in accuracy. The graph bellow shows the similar effect.



- The higher the degree the of the polynomial Kernel, the value of score decreases, the SVM performance decreases in this setting. The value of the Bias and C is good for low values in this setting too.



Task-3 : Dataset Generation

Task-3.1 : Random datasets

A random dataset is generated by using make_classification() function again.

Task-3.1.1 : Tuning the hyper parameters of different kernels

- Cross-validation technique is used on the dataset to find the best value gamma for RBF kernel in a range of 1 to 50 and it was found that for particular dataset the best value of gamma is equal to 9.
- Same procedure is followed for the polynomial kernel and the best value of degree is equal to 3 and for Bias is equal to 2.

Task-3.1.2 : Plotting the decision boundaries and displaying support vectors

In this task, the best value of gamma, degree and Bias are used and the decision boundaries for the dataset is plotted. In addition to this the number of support vectors are also plotted in the graph which tells us that how many examples from the dataset were really used to make the decision boundary.

Task-3.2 : Existing datasets

The procedure followed above is done for the existing dataset also (i.e. Moon and Iris dataset) and similar results are obtained as before. As it is shown in the Jupyter notebook.

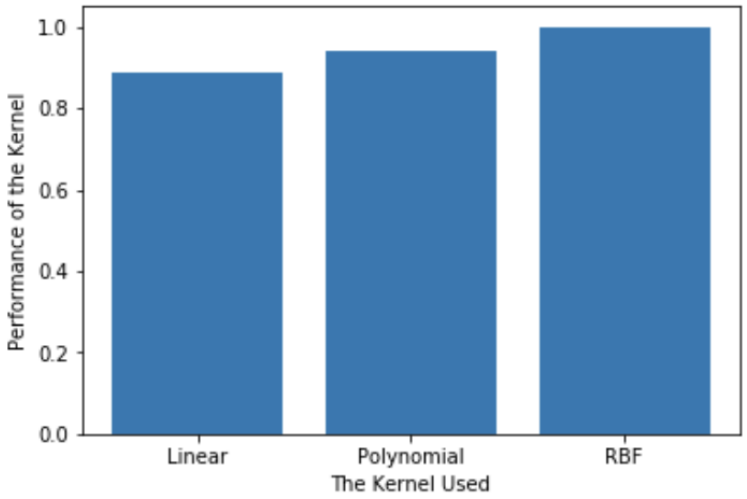
Task-4 : Real dataset

The ozone dataset provided is used in this section and at first, the number of stations are converted into a binary vector using get_dummies() function of pandas library. The observation column for the ozone concentration is converted into a binary column for which any value greater than 150 is equal to 1 and otherwise it is 0.

The evaluation on the dataset is done on the normalized and normal dataset.

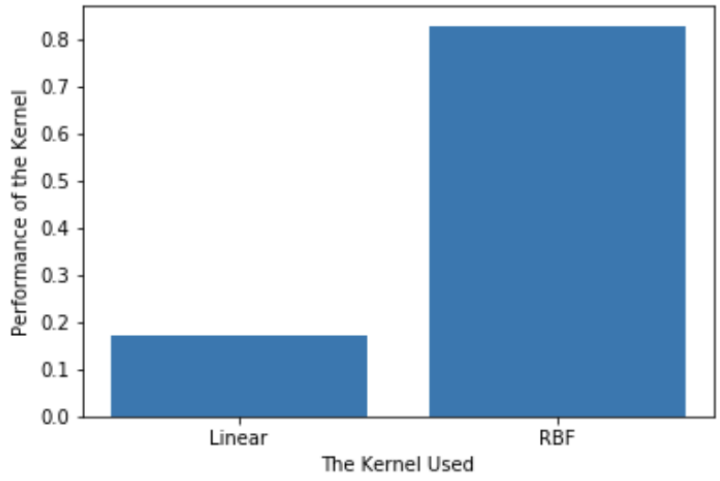
- **Evaluating using normalized dataset**

1. The value of Gamma, C, Degree and Bias are tuned using the same strategy as before.
2. The best values found are used to obtain the model accuracy and number of support vectors with different settings of kernels and the following graph is obtained as given below.



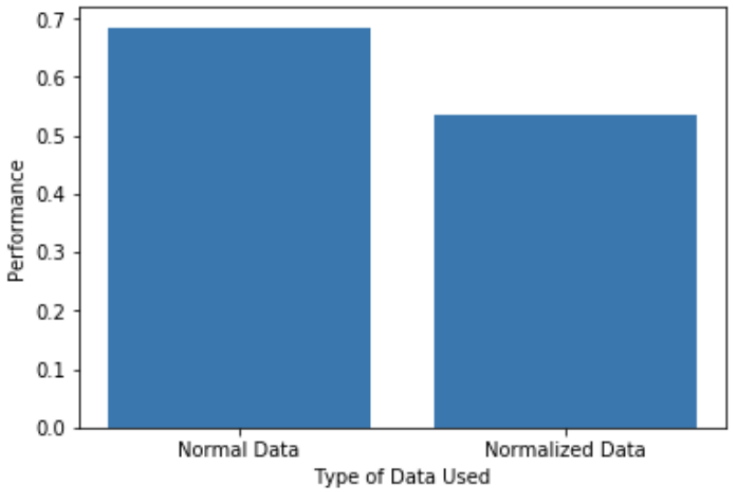
- **Evaluating using normal dataset**

1. The value of Gamma, C, Degree and Bias are tuned using the same strategy as before with normal dataset.
2. The hyper-parameters of polynomial Kernel cannot be tuned as it takes exponential time to learn the best hyper-parameters for normal dataset. So it cannot be analyzed in normal circumstances.
3. The best values found for Gamma and C are used to obtain the model accuracy and number of support vectors with different settings of kernels and the following graph is obtained as given below.



Task-5 : Regression problem: predicting the ozone concentration

The SVR is imported and used with normal setting with RBF kernel and is used to fit normalized and normal dataset and the following performance scores are obtained are shown through the graph below.



Conclusion

The practical session gave us an in-depth idea of how we can use SVM by tuning its hyper-parameters and its performance on different data setting. This is important for us because we get the better idea about the theoretical concepts in practical setting.