Support Vector Machines - HW7

Karthik Ravindra Rao and Praneet Kalluri

April 27, 2017

1 Algorithms

1.1 Linear SVM

Following functions are written for Linear SVM in file named svm_linear.py

prepare_data(*data*) : function to prepare the data in the required format for quadratic solver

plot_data_points(*coordinates*, *labels*, *plot*) : helper function to plot the data points

plot_separator(*plot*, *weight*, *bias*) : helper function to plot the seperator line

quadratic_solver(P, q, G, h, A, b) : function which performs quadratic programming

calculate_weight_and_bias(*alphas*, *labels*, *coordinates*) : function which calculates the weight and the bias

main() : Main function which calls all other functions

The following result was obtained which is also depicted in Figure 1:

```
The optimal Alphas are :  [33.738751924716915, 1.2946850552133398, 32.444067196391288]
The weights are :  [ 7.2500563  -3.86188924]
Bias is as follows :  -0.106987337628
The support vectors are :  [(0.24979413895365565, 0.18230306153352005),
(0.39178890015066026, 0.96675591132753536), (0.020664582042122071, 0.27003157791727084)]
```

1.2 Non-Linear SVM

This uses the same functions as the Linear SVM. However, the Kernel function is additional, which transforms the input coordinates into a different dimension.

This is implemented in file named svm_nonlinear.py

kernel(X) : Kernel function transforms the coordinates into a different dimension

The following result was obtained. The coordinates of points before transformation is shown in figure 2 and the final result after transformation of the coordinates and plotting the line is shown in figure 3.

```
The optimal Alphas are :  [0.020317189620194166, 0.021095697614131999, 0.041412887117296834]
The weights are :  [ 0.20161734  0.20536852]
Bias is as follows :  -21.1111001562
Support Vectors are :  [(105.28748483329353, 4.3011355142140193),
(1.7938083202366126, 105.90443854850511), (47.699405505786672, 51.098743215459734)]
Support Vectors in the original data set before application of kernel are :
[(-10.260969000698401, 2.0739179140491601),
```

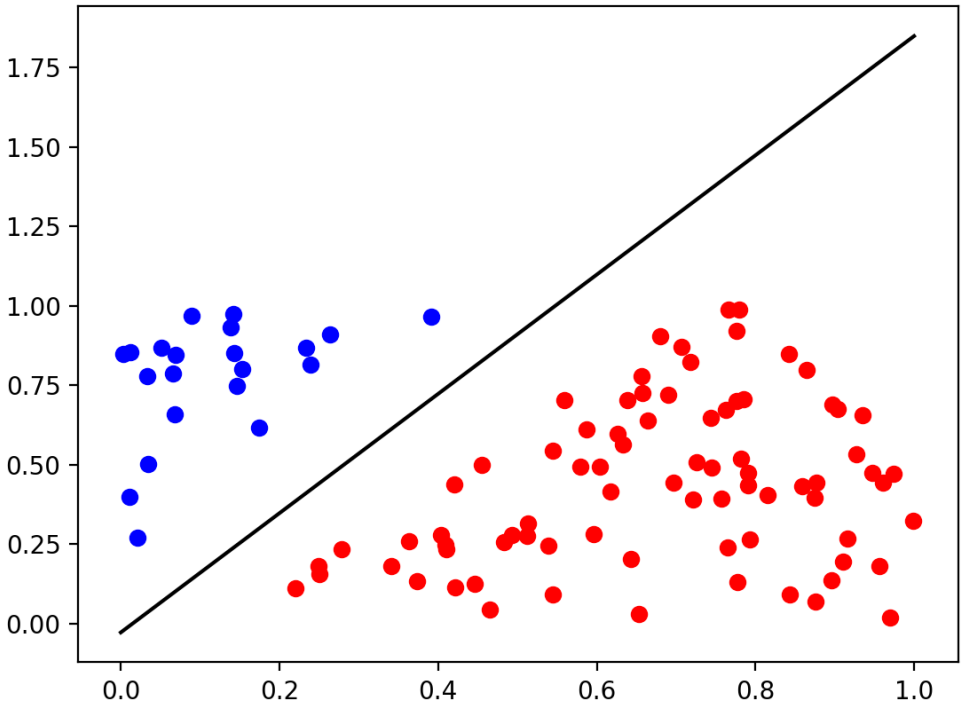

Figure 1: Linear SVM

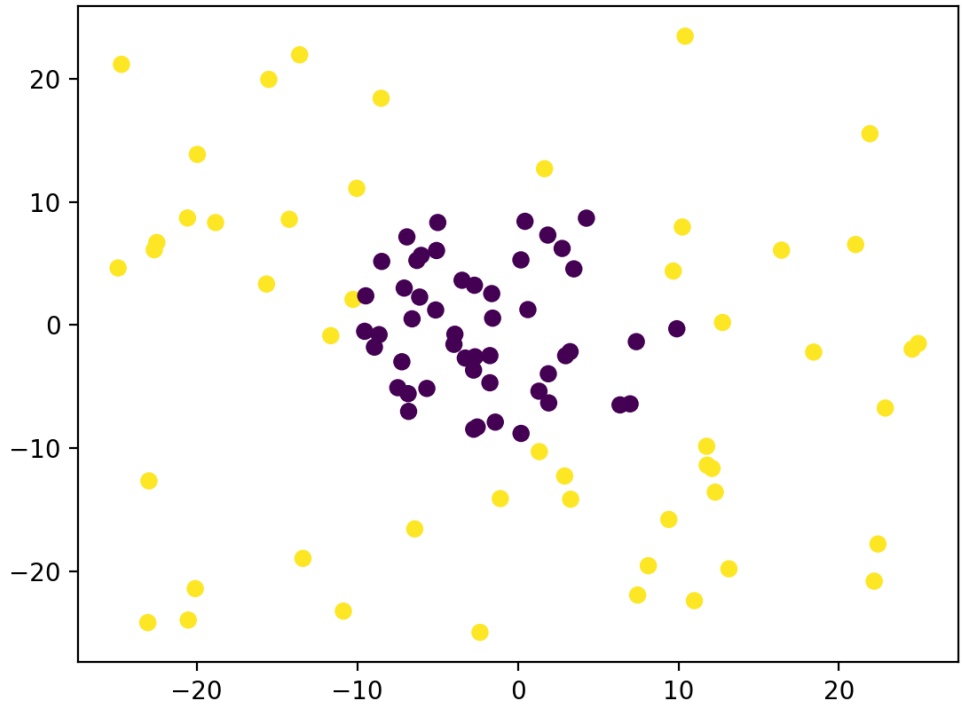

Figure 2: Non-Linear data points before transformation

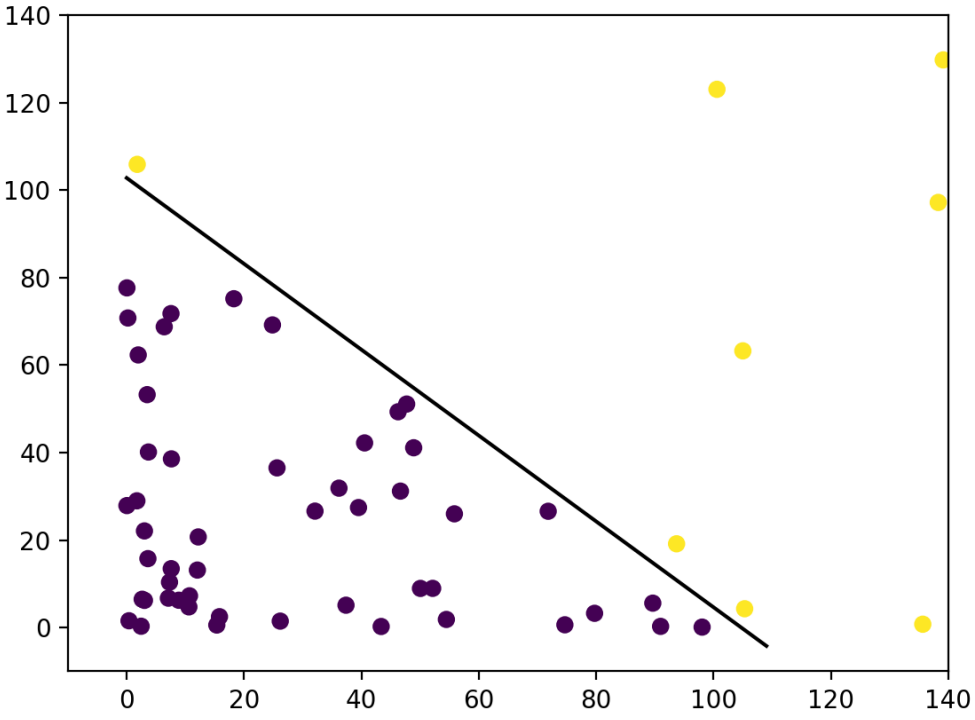

Figure 3: Non-Linear data points after transformation along with separator. Please note that all points are not depicted to increase clarity of separation

```
(1.3393312959221899, -10.290988220210201), (-6.9064756211679104, 7.1483384933465297)]
```

2 Software Familiarization

2.1 Linear SVM Sci-Kit Learn

Linear SVM is implemented as follows.

```
clf = svm.SVC(kernel='linear')
clf.fit(coordinates, labels)
weights = clf.coef_[0]
bias = clf.intercept_[0]
```

2.2 Non-Linear SVM SciKit Learn

For non linear data, Scikit Learn can be implemented as follows. Which ever kernel which gives a better result, can be retained.

```
for kernel in ('poly', 'rbf'):
    clf = svm.SVC(kernel=kernel)
    clf.fit(X, Y)
```

2.3 Improvements for Implemented Algorithm

- Further accuracy can be explored with different varieties of Kernel.
- The program is restricted to two dimensions. If a new data set requires more than two dimensions, the program will have to be tweaked.

3 Applications

3.1 Text and Hypertext Categorization

In standard Inductive and Transductive settings, application of SVM help in reducing the need for labeled training instances. This significantly improves the accuracy for Text and Hypertext Categorization.

3.2 Classification of images

SVMs are significantly used in Classification of images. SVMs are also used for image segmentation. As SVMs result in high accuracy for lower number of relevance feedback, it is chosen over traditional query techniques.

3.3 Biological Sciences

Experiments on protein classification using SVMs have resulted in a 90% accuracy, paving way for SVMs being a popular algorithm. Popular usages include studies on Diabetes and Cancer.

4 Responsibility division between Teammates

We, Karthik and Praneet have brain-stormed and implemented the assignment with equal contribution for every line of code.

5 References

https://en.wikipedia.org/wiki/Support_vector_machine#Applications
http://cvxopt.org/applications/svm/
http://goelhardik.github.io/2016/11/28/svm-cvxopt/
http://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html