Support Vector Machines

INF552

Aviral Upadhyay

Vandit Maheshwari

Work Breakup

Aviral Upadhyay

Non Linear SVM and documentation

Vandit Maheswari

Linear SVM and Standard Library Function

INTRODUCTION

Objectives

Implement Support vector machines for 2D data in linsep.txt and nonlinsep.txt.

The result should report the support vectors and the equation of the separation line for linear separation SVM. It should also report the support vectors and the kernel used for non-linearly separable data.

Functionality

Linear Separable

1. Input
2. Solve for alpha using quadratic programmer
3. Get indices using alpha
4. Calculate weights
5. Calculate the bias value (intercept).
6. Get the equation of the line with y = mx+c where, m = -weights[0][0]/weights[0][1] and c = -intercept/ weights[0][1]

Non Linear Separable

1. Input
2. Solve for alpha using quadratic programmer
3. Get indices from alpha
4. Calculate support vectors using indices

Design

Data Structure

1. Input data is stored in Numpy arrays in both linear and non linear seperable data
2. **Alpha** is stored in Numpy array and so is the **indices** and weights.
3. Support vectors are stored in List in python

Procedures

1. Input
2. Calculate Alpha using the quadratic programming solver.
    - The **kernel** used for non-linear separable data is polynomial kernel

$$K(x,z) = (Z^T x + c)^n$$

3. Calculate indices using the value in alpha that are non zero.
4. Remove all zero values from alpha
5. Get the support vectors using the indices on X.

Comparison with Scikit Learn

- Machine Learning Library in Python
sklearn.svm.SVC and sklearn.svm.LinearSVC
LinearSVC is particularly meant for Linear kernels when dealing with linearly separable data, whereas SVC accepts a parameter called kernel where you can define the type of kernel transformation that you would like to apply for the non-linear separable data. Our algorithm is developed currently only to deal with hard classification problems. SVC and LinearSVC both consist of an argument 'C' that corresponds to the error term that needs to be included to allow certain misclassification of points. The different kinds of kernels available are 'linear', 'poly', 'rbf', 'sigmoid',
'precomputed'. If none is given, 'rbf' will be used. Also our implemented algorithm only works for binary classification whereas LinearSVC
also includes support for multi-class classification through one-vs-the-rest. SVC on the other hand works on one-against-one approach for multi- class classification. If there are n classes, then a combination of their classifiers are trained and then the 'density_function_shape' parameter is used in order to provide a consistent interface to all the classifiers.

Optimizations and Challenges

Optimization

- Avoiding dot operator:

  Calculating b we do not use any for loop and thus code is faster than using loop for every value

- Removing non-zero and getting there incides using flatnonzero of numpy is faster than iterating through a python list

Challenges

Understanding which kernel to choose. Figuring out computation using quadratic functions and using constraints while doing so. Understanding how to convert the data points using kernel function and computing the weights in the new dimension.

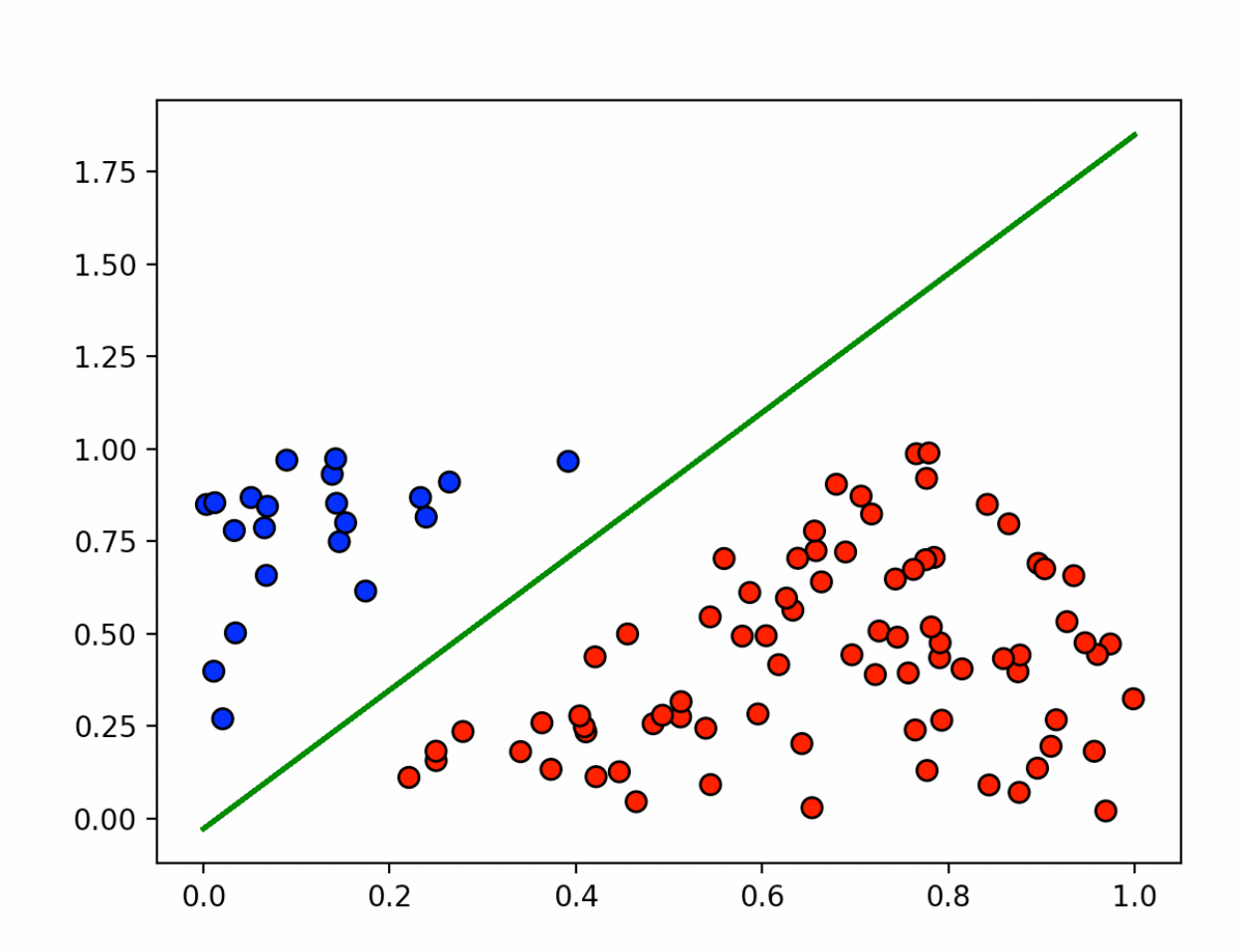
Usage Instruction

Use Scikit-learn, Python3 the code also requires cvxopt for solving quadratic equations

Results

```
(venv) aviral:HW-6 aviral$ python3 nonlinsep.py
     pcost       dcost       gap    pres   dres
 0: -4.0666e+01 -1.0206e+02  5e+02  2e+01  3e+00
 1: -1.5924e+02 -2.2789e+02  3e+02  1e+01  1e+00
 2: -2.9280e+02 -3.6244e+02  3e+02  1e+01  1e+00
 3: -5.7710e+02 -6.0303e+02  4e+02  9e+00  1e+00
 4: -1.2873e+03 -1.2409e+03  5e+02  9e+00  1e+00
 5: -1.2647e+03 -1.0924e+03  7e+02  8e+00  9e-01
 6: -6.9076e+02 -4.0802e+02  1e+03  5e+00  6e-01
 7: -1.8688e+02 -2.9779e+01  4e+02  1e+00  2e-01
 8: -3.4731e+00 -5.2038e-02  1e+01  3e-02  4e-03
 9: -3.5053e-02 -3.8447e-02  1e-01  3e-04  3e-05
10: -2.1413e-02 -2.7448e-02  6e-03  3e-18  2e-13
11: -2.6166e-02 -2.6328e-02  2e-04  6e-18  2e-13
12: -2.6293e-02 -2.6295e-02  2e-06  5e-18  2e-13
13: -2.6295e-02 -2.6295e-02  2e-08  2e-18  1e-13
Optimal solution found.
[[0.0125012 ]
 [0.0149904 ]
 [0.00664213]
 [0.00466215]
 [0.00740837]
 [0.00638511]]
The Suppport Vectors are:
[[ -8.47422847   5.15621613]
 [-10.260969     2.07391791]
 [  1.3393313  -10.29098822]
 [  9.67917724   4.3759541 ]
 [ -6.80002274  -7.02384335]
 [  9.90143538  -0.31483149]]
(venv) aviral:HW-6 aviral$
```

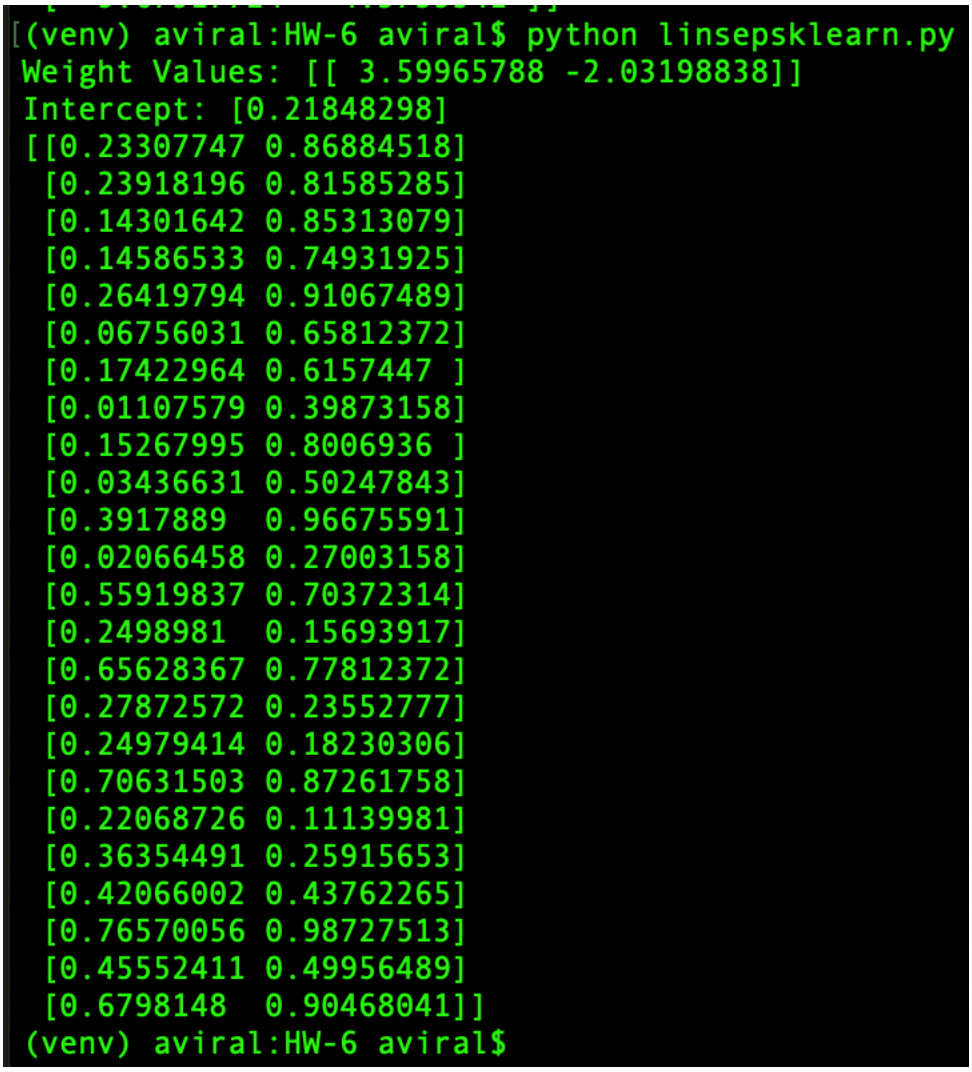
Non Linear Separable data



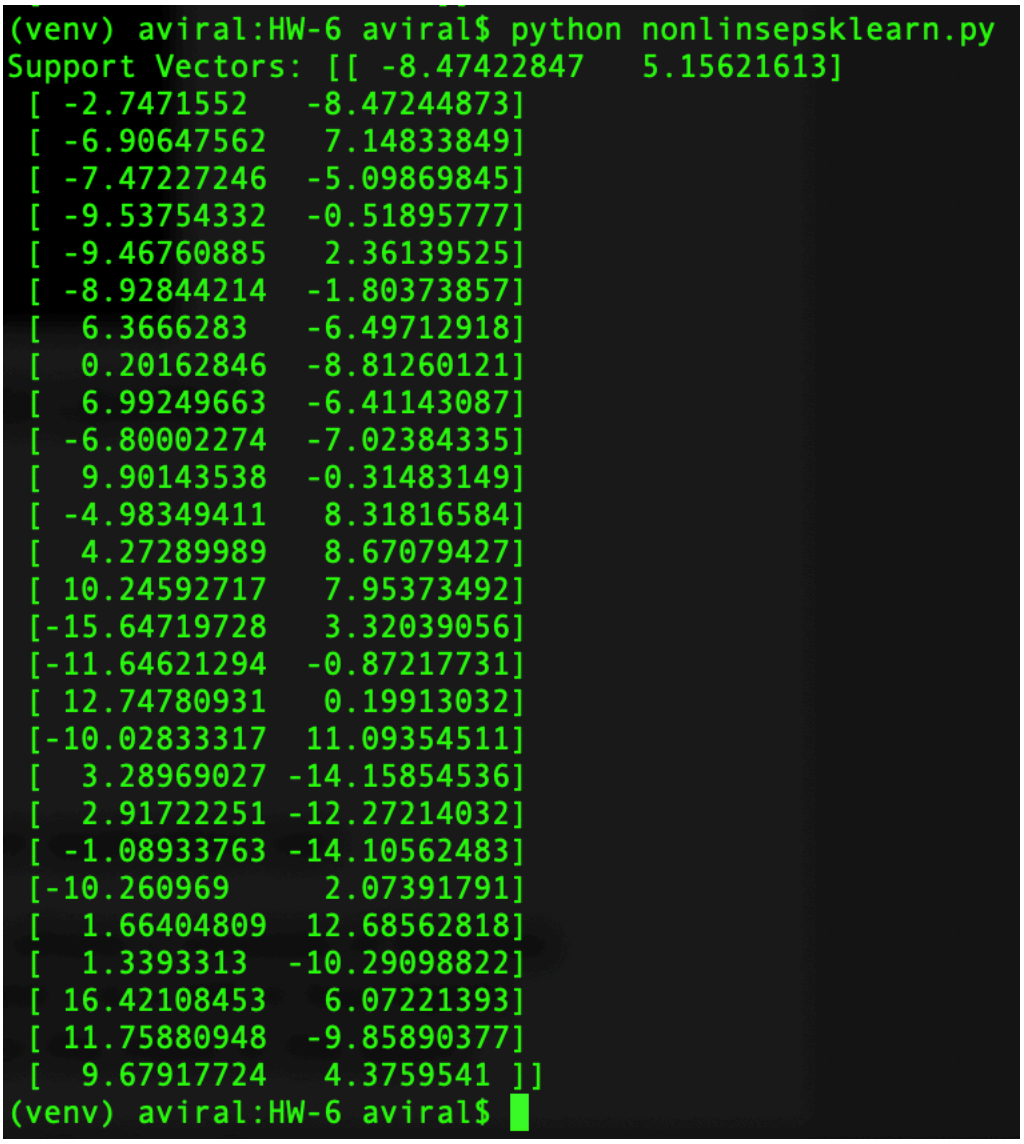
Linear Separable Data Graph

```
(venv) aviral:HW-6 aviral$ python3 linsep.py
     pcost       dcost       gap    pres   dres
 0: -2.0636e+01 -4.3905e+01  3e+02  2e+01  2e+00
 1: -2.2372e+01 -3.7202e+01  9e+01  5e+00  5e-01
 2: -2.3112e+01 -3.8857e+01  5e+01  2e+00  2e-01
 3: -2.8318e+01 -3.3963e+01  1e+01  4e-01  4e-02
 4: -3.2264e+01 -3.3927e+01  2e+00  1e-02  1e-03
 5: -3.3568e+01 -3.3764e+01  2e-01  1e-03  1e-04
 6: -3.3737e+01 -3.3739e+01  2e-03  1e-05  1e-06
 7: -3.3739e+01 -3.3739e+01  2e-05  1e-07  1e-08
 8: -3.3739e+01 -3.3739e+01  2e-07  1e-09  1e-10
Optimal solution found.
Weight Values : [[ 7.25005369 -3.86189521]]
Intercept Value:  -0.10698559836302746
The equation of the line is :
[7.250053688191241] x + [-3.861895209680986] y + -0.10698559836302746
```

Linear Separable Result and equation of line



Linear Separable Result and equation of line (Sci-Kit LEARN)



Non-linear Separable Result and equation of line (Sci-Kit LEARN)

Applications

SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances.

- In Quantitative finance, for example predictive tasks, where x consists of features derived from a historical stock indicator time series and y is a sell or buy signal.
- Text based SVM, filter e-mail to be forwarded to either the inbox or the spam folder
- Used in Bioinformatics and research (Eg: Protein classification, Cancer classification)
- Another outstanding application of SVMs is the detection of human faces in gray-levelimages. The problem is to determine in an image the location of human faces and, if there are any, return an encoding of their position.
- Hand-written characters can be recognized using SVM.
- Many problems of image processing and image classification are accomplished by SVM.