To implement the Support Vector Machine and find the equation of the fattest margin line/curve that separates the data points.

PROJECT DESCRIPTION

- Programming language used: Python
- Data Structure: Lists, Arrays, CVXOPT matrix
- File Name: Linear_SVM.py, NL_SVM.py
- Inputs: linsep.txt, nonlinsep.txt
- Output: weight and bias point values for the equation $W^TX + b = y$

IMPLEMENTATION

- <u>Modules created:</u>

  - Linear_SVM.py – for the linearly separable data points
    - Read_File(): to save the input file into array of points for the given dimensions in samples, and the classification labels in labels.

    - svm(X, Y) : For X data points and Y labels, this function computes the alpha by solving the Quadratic Programming equation :

      $$\min L(\alpha) = \frac{1}{2} * \alpha^T * Q * \alpha + (-1)^T * \alpha$$

      where Q = $\begin{bmatrix} y_i y_j X_i^T X_j & . & . \\ . & . & . \\ . & . & . \end{bmatrix}_{(N \times N)}$

      by using the CVXOPT Quadratic problem solver.

    - supportVectors(a,s) – For all gives sample inputs s and their α, the main task of this module is find all the support vectors. A support vector is a point for which the value of $\alpha >= 0$. We have received three such values where the values of α are highly positive. For the remaining data points, the values of α are too small to be considered.

    - calculateW(a,L,S) – Computes the weight vector using the SVM optimality condition : $w = \Sigma\, \alpha_n * y_n * X_n$

    - calculateb(sv,l,w) – Using one support vector sv from the support vector list, its corresponding classification label and weight vector computes the value of the bias point using the equation : $b = \frac{1}{y_n} - W^T * X$

  - NL_SVM.py – for the non-linearly separable data points

- Read_File(): to save the input file into array of points for the given dimensions in samples, and the classification labels in labels.

- kernel(X): For all the given input data points, converts them to a higher dimension Z where the points are linearly separable. The kernel function used here is: $Z = \Phi(X) = (1, X_1^2, X_2^2, \sqrt{2}\, X_1, \sqrt{2}\, X_2, \sqrt{2}\, X_1 X_2)$ which is a 6 dimensional representation of X.

- svm(X, Y) : For X(=Z) higher dimension data points and Y labels, this function computes the alpha by solving the Quadratic Programming equation :

  $$\min L(\alpha) = \frac{1}{2} * \alpha^T * Q * \alpha + (-1)^T * \alpha$$

  where Q = $\begin{bmatrix} y_i y_j Z_i^T Z_j & . & . \\ . & . & . \\ . & . & . \end{bmatrix}_{(N \times N)}$

  by using the CVXOPT Quadratic problem solver.

- supportVectors(a,s) – For all gives sample inputs s and their α, the main task of this module is find all the support vectors. A support vector is a point for which the value of α >=0. We have received three such values where the values of α are highly positive. For the remaining data points, the values of α are too small to be considered.

- calculateW(a,L,S) – Computes the weight vector using the SVM optimality condition : $w = \Sigma\, \alpha_n * y_n * X_n$

- calculateb(sv,l,w) – Using one support vector sv from the support vector list, its corresponding classification label and weight vector computes the value of the bias point using the equation : $b = \frac{1}{y_n} - W^T * X$

* <u>Termination Condition</u>:

  When the weight and bias vectors are found after finding the optimal solution from the Quadratic Solver, the program terminates.

* <u>Result Interpretation</u>:

  Linear_SVM : The result represents the weight and the bias value that satisfies the equation $W^T X + b = y$. Additionally, we have printed the support vectors and their corresponding alpha values. We have tested the output on all the input data points and found 100% correct classification.

```
Console  Terminal
&lt;terminated&gt; C:\Users\dell\workspace\HW6_SVM\src\Linear_SVM.py
Optimal solution found.
weight [ 7.2500563  -3.86188924]
b [-0.10698734]
support vector 1 : [ 0.24979414  0.18230306]  alpha 1 : 33.7387519247
support vector 2 : [ 0.3917889   0.96675591]  alpha 2 : 1.29468505521
support vector 3 : [ 0.02066458  0.27003158]  alpha 3 : 32.4440671964
```

NL_SVM : The result displays all the higher dimensional vector points Z, the support vectors found over the data set and the corresponding alpha values. The primary output here is the weight and the bias value that satisfies the equation $W^T X + b = y$.

We have tested our results on the other points in the Z dimension, and found 100% correct classification based on our output values.

```
Console  Terminal
&lt;terminated&gt; C:\Users\dell\workspace\HW6_SVM\src\NL_SVM.py
 alpha 6 : 0.00638511049352
-------------------------------------------------------------------

weight vector in Z:
[  1.92860944e-18   1.60702133e-01   1.58698035e-01  -9.47654970e-03
  -3.85759176e-02  -1.10221159e-03]
bias point =  [-16.66005058]
```