**Support Vector Machines** (using Sequential Minimal Optimization)**:**

Support vector machines is highly efficient technique for the image classification problems. Though the algorithm takes lot of time to learn, the prediction time is very less. Accuracy is very high and it depends on the hyperparameters. Hard Margin SVM is used for linearly separable data points and it does not allow points to fall between the Margin, and Hence in case of overlapping points or misclassified points we use Soft Margin SVM where we introduce a slack variable(epsilon). We start with finding the distance of point from the boundary and then to minimize the distance. The slack variables are added to as the misclassified distance.

When we try to solve this problem by using langrage’s equation, we reach to a quadratic equation which is called as Dual problem.

Graphical user interface, text, Word

Description automatically generated

Where alpha is lagrangian solver, C is regularization parameter. The higher C goes, higher is the computational power and SVM tend towards Hard Margin.

Where we can replace the inner product of Xi and Xj with the Kernel function K(Xi,Xj) to get the exact answer if we try to project it onto higher dimensional space**.**

Below are the KKT conditions which are used to check the convergence:

Text, letter

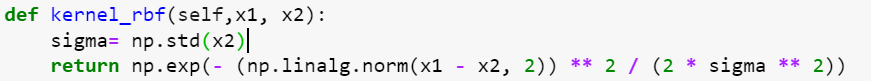
Description automatically generated

**Approach:**

We have used SMO to optimize the above equation using John Platt’s paper.

We are solving for the alpha values first, keeping one of them fixed and iterating over the second. Then we calculate w and b values. We are using KKT conditions and constraints mentioned in the John Platt’s paper for converging conditions. In our project we have used 2 convergence conditions explained in the below steps:

1. In the project we first iterated through C values 1,10,100 and 1000 and checked the high F1 score range and found that the C values in the range of 100 are getting better F1 score for epsilon value 0.00001
2. We are using Radial Basis Function as our kernel using below function, we also have trained model using quadratic function but we RBF performs better.



1. We used Max iterations=1000 to train our model
2. For Every C We are choosing the best F1 score values of the Validation dataset, training dataset and plotting it against the iterations to see if the model is learning correctly and there is no overfitting. And marking the safe max iteration which could also be used as the convergence condition
3. As another convergence condition, We used the norm function to get the difference between old alphas and new alphas and compared it with the epsilon which is our misclassification allowance, and if it is less than the allowance the model will converge
4. We followed the same procedure for PCA reduced dimensional data and dimension reduced data using Autoencoder

**Results:**

**For PCA reduced Dimension Dataset**

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

**For Dimension reduced by Autoencoder file:**

Chart, line chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | PCA Reduced Dimension Dataset | Dimension reduced by Autoencoder Dataset |
| Best F1 Score | 0.8459 | 0.8205 |
| Number of optimal Iterations | ~326 | ~340 |
| C(Regularization parameter) | 107 | 109 |

Chart, bar chart

Description automatically generated