Assignment 2: Implementing different algorithms for constructing Linear Discriminant Functions

Satarupa Guha 201307566

Part 1

Task

To implement the following algorithms for constructing binary classifier using the MS2CD dataset. We are also required to compare the accuracies we get using this technique with those we get by using 1 Nearest Neighbor Classifier.

i. Single Sample Perceptron without margin
ii. Single Sample Perceptron with margin
iii. Batch Perceptron without margin
iv. Batch Perceptron with margin
v. Batch Relaxation

Linear Discriminant Function

A discriminant function that is a linear combination of the components of x can be written as

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

where w is the weight vector and w0 the bias or threshold weight. Linear discriminant functions are going to be studied for the two-category case, multi-category case, and general case. For the general case there will be c such discriminant functions, one for each of c categories.

Single Sample Perceptron

```
begin initialize a,k<-0
    do k<-(k+1)mod n
        if $y^k$ is misclassified by a then a<-a+ $y^k$
    until all patterns properly classified
return a
end
```

Batch Perceptron

```
begin initialize a,k<-0
    do k<-(k+1)
        if $y^k$ is misclassified by a then a <- a + eta(k) * ∑k $y^k$
    until all patterns properly classified
return a
```

```
end
```

Batch Relaxation

```
begin initialize a,k<-0
    do k<-(k+1)
        if yk is misclassified by a then  factor=(margin-a'yk-1)/|y|^2
            a= a+eta(k) ∑factor *y^k
    until all patterns properly classified
return a
end
```

Methology

TRAINING PHASE

Step 1

Preprocessing- The each data point consists of 784 coordinates. We augment each data point vector with '1'. So after this, each data point is basically a vector having 785 dimensions.

Step 2

Now, for each of the 10 digits, we then negate the coordinate values of those points whose label is equal to that digit.

Step 3

Then for each of the 10 digits, we compute the 'a' vector that is the coordinate vector of the decision boundary separating this digit from the rest of the digits. The 10 'a' vectors are stored in a 'v' vector that we return from this function.

TESTING PHASE

It takes as input the 'a' vector that we obtain from the training phase, along with the testing data that we take as input in the main program. It checks a particular test sample lies on which side of the decision boundary as specifies by the ‘a’ vector. Hence we compute the accuracy.

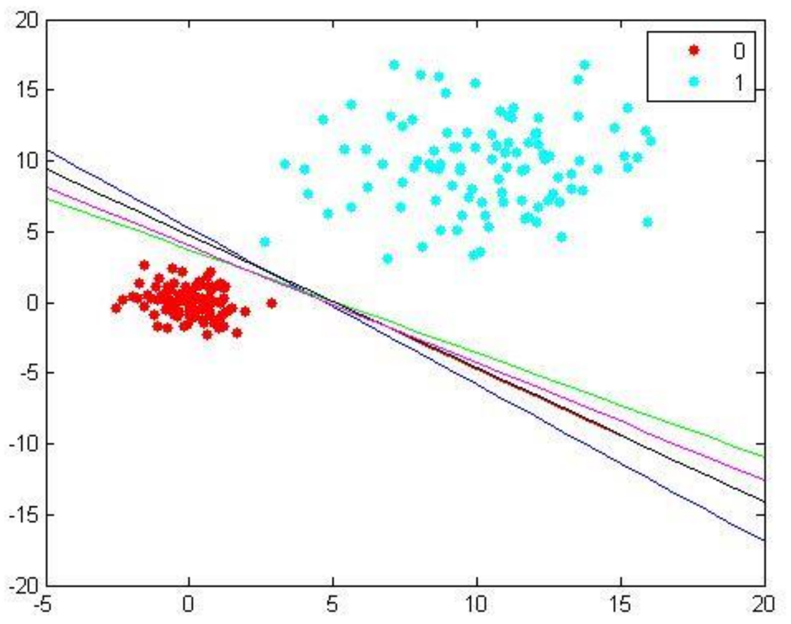
Results

Table showing Train and Test Accuracies for both LDF and I-NN for the 5 algorithms

Algorithm	LDF Train Accuracy	LDF Test Accuracy
Single Sample Perceptron	100	100
Single Sample Perceptron with margin	100	100
Batch Perceptron	100	100
Batch Perceptron with margin	100	100
1 Nearest Neighbor	100	100

For all the above cases, I have taken margin as 5, eta as 0.1.

Graph showing the decision boundaries corresponding to each of the 5 algorithms



The line in Red color denotes simple sample perceptron with margin(20), that in Blue denotes simple sample perceptron without margin, that in Green denotes batch perceptron without margin, Black denotes batch perceptron with margin(20) while line in magenta denotes the decision boundary forbatch relaxation. In this figure, the red and black lines has coincided.

Part 2

Task

To extend the algorithms for 2-class classifications to multiclass classification. The data set we use for training and testing is a subset of the MNIST dataset. We are also required to compare the accuracies we get using this technique with those we get by using 1 Nearest Neighbor Classifier.

Methology

The preprocessing step is same as the binary classification case. I have used 1 vs rest technique for the multiclass classification. Since there are 10 digits, I had to use 10 binary classifiers to distinguish each digit from the rest. In the testing phase, we come across situations where more than one class claims a particular test sample. In this case, we assign the test sample to the class whose decision boundary is at a maximum distance from the test sample as this signifies maximum confidence. There are also situations where none of the classes claim a particular test sample. In this case, we assign the label for that sample as undefined.

Results

Table showing Train and Test Accuracies for both LDF and INN for the 5 algorithms

Algorithm	LDF Train Accuracy	LDF Test Accuracy
Single Sample Perceptron	**100**	**81.2623**
Single Sample Perceptron with margin(=5)	**100**	**81.85**
Batch Perceptron	**100**	**75.8383**
Batch Perceptron with margin	**100**	**78.3037**
1 Nearest Neighbor	**100**	**86.9822**