Assignment 1

CSCE509-Pattern Recognition

SM Zobaed-C00300901

March 13, 2019

1. Generated 10000 samples of two classes. 5000 class-0, 5000 class-1. Let this be defined as samples' set 1. The required code snippet is shown in Figure 1

```
#1. Generate Sample of Two Classes
x0data = np.transpose(np.array([[np.random.normal(1.0, 1.0) for i in range(5000)],
                                [np.random.normal(-1.0, 1.0) for i in range(5000)]]))
x1data = np.transpose(np.array([[np.random.normal(-1.0, np.sqrt(1.3)) for i in range(5000)],
                                [np.random.normal(1.0, np.sqrt(1.3)) for i in range(5000)]]))
t0vec = -np.ones(5000)
t1vec = np.ones(5000)
xdata = np.concatenate((x1data, x0data), axis=0)
tvec = np.concatenate((t1vec, t0vec))
shuffle_index = np.random.permutation(10000)
xdata, tvec = xdata[shuffle_index], tvec[shuffle_index]
```

Figure 1: Code snippet for generating the samples' set 1

2. Trained a Perceptron to classify samples' set 1. Line parameters and the accuracy are shown in Figure 2 from the output of the code. However, to calculate the accuracy, I generate a test set of samples. The test set contains 3000 rows and it is equally classified into the two classes. The code snippet that used to generate the test set is shown in Figure 3.

```
In [178]: runfile('/home/c00300901/Dropbox/Zobaed/assignment1.py', wdir='/home/c003009

For 1st samples' set:
Sample Shape:  (10000, 2)
Parameters:  Perceptron(alpha=0.0001, class_weight=None, eta0=1.0, fit_intercept=True,
      max_iter=None, n_iter=None, n_jobs=1, penalty=None, random_state=0,
      shuffle=True, tol=0.001, verbose=0, warm_start=False)
Coefficient: [[-0.7384946   2.42510148]]
Intercept: [-1.]
Iteration:  3
Synthetic Test Samples' set Shape: (3000, 2)
Accuracy:  0.865
```

Figure 2: Line parameters and accuracy for the samples' set 1

```
test_x0data = np.transpose(np.array([[np.random.normal(1.0, 1.0) for i in range(1500)],
                                     [np.random.normal(-1.0, 1.0) for i in range(1500)]]))
test_x1data = np.transpose(np.array([[np.random.normal(-1.0, np.sqrt(1.3)) for i in range(1500)],
                                     [np.random.normal(1.0, np.sqrt(1.3)) for i in range(1500)]]))
test_t0vec = -np.ones(1500)
test_t1vec = np.ones(1500)
test_xdata = np.concatenate((test_x1data, test_x0data), axis=0)
test_tvec = np.concatenate((test_t1vec, test_t0vec))
shuffle_index = np.random.permutation(3000)
test_xdata, test_tvec = test_xdata[shuffle_index], test_tvec[shuffle_index]
```

Figure 3: Code snippet for generating the test set

3. Generated 10000 samples of 9000 class-0 and 1000 class-1. Let this be defined as samples' set 2. The required code snippet is shown in Figure 4.

```
#Generating Sample
x0data = np.transpose(np.array([[np.random.normal(1.0, 1.0) for i in range(9000)],
                                [np.random.normal(-1.0, 1.0) for i in range(9000)]]))
x1data = np.transpose(np.array([[np.random.normal(-1.0, np.sqrt(1.3)) for i in range(1000)],
                                [np.random.normal(1.0, np.sqrt(1.3)) for i in range(1000)]]))
t0vec = -np.ones(9000)
t1vec = np.ones(1000)
xdata_new = np.concatenate((x1data, x0data), axis=0)
tvec_new = np.concatenate((t1vec, t0vec))
shuffle_index = np.random.permutation(10000)
xdata_new, tvec_new = xdata_new[shuffle_index], tvec_new[shuffle_index]
```

Figure 4: Code snippet for generating the samples' set 2

4. Trained a Perceptron to classify samples' set 2. Line parameters and the accuracy are shown in Figure 5 from the output of the code. However, to calculate the accuracy, the earlier test set is utilized.

```
For Second Samples' set:
Parameters:   Perceptron(alpha=0.0001, class_weight=None, eta0=1.0, fit_intercept=True,
      max_iter=None, n_iter=None, n_jobs=1, penalty=None, random_state=0,
      shuffle=True, tol=0.001, verbose=0, warm_start=False)
Coefficient: [[-0.80965716  0.92294689]]
Intercept: [-2.]
Iteration:  2
Synthetic Test Sample's set Shape: (3000, 2)
Accuracy:   0.722
```

Figure 5: Line parameters and accuracy for the samples' set 2

5. Considering Cross Validation=10, *Average accuracies* for samples' set 1 and 2 are provided in Figure 6.

```
Considering Cv:
Average Accuracy Considering CV=10 for 1st Samples' set:  0.8948
Average Accuracy Considering CV=10 for 2nd Samples' set:  0.9315
```

Figure 6: Average Accuracies considering CV=10

6. Considering Cross Validation=10, *Precision and Recall* score for samples' set 1 & 2 are provided in Figure 7.

```
Precision Score:
Precision Score Considering CV=10 for 1st Samples' set:  0.9043424825891029
Precision Score Considering CV=10 for 2nd Samples' set:  0.6354256233877902


Recall Score:
Precision Score Considering CV=10 for 1st Sample:  0.883
Precision Score Considering CV=10 for 2nd Sample:  0.1194
```

Figure 7: Precision and Recall score considering CV=10