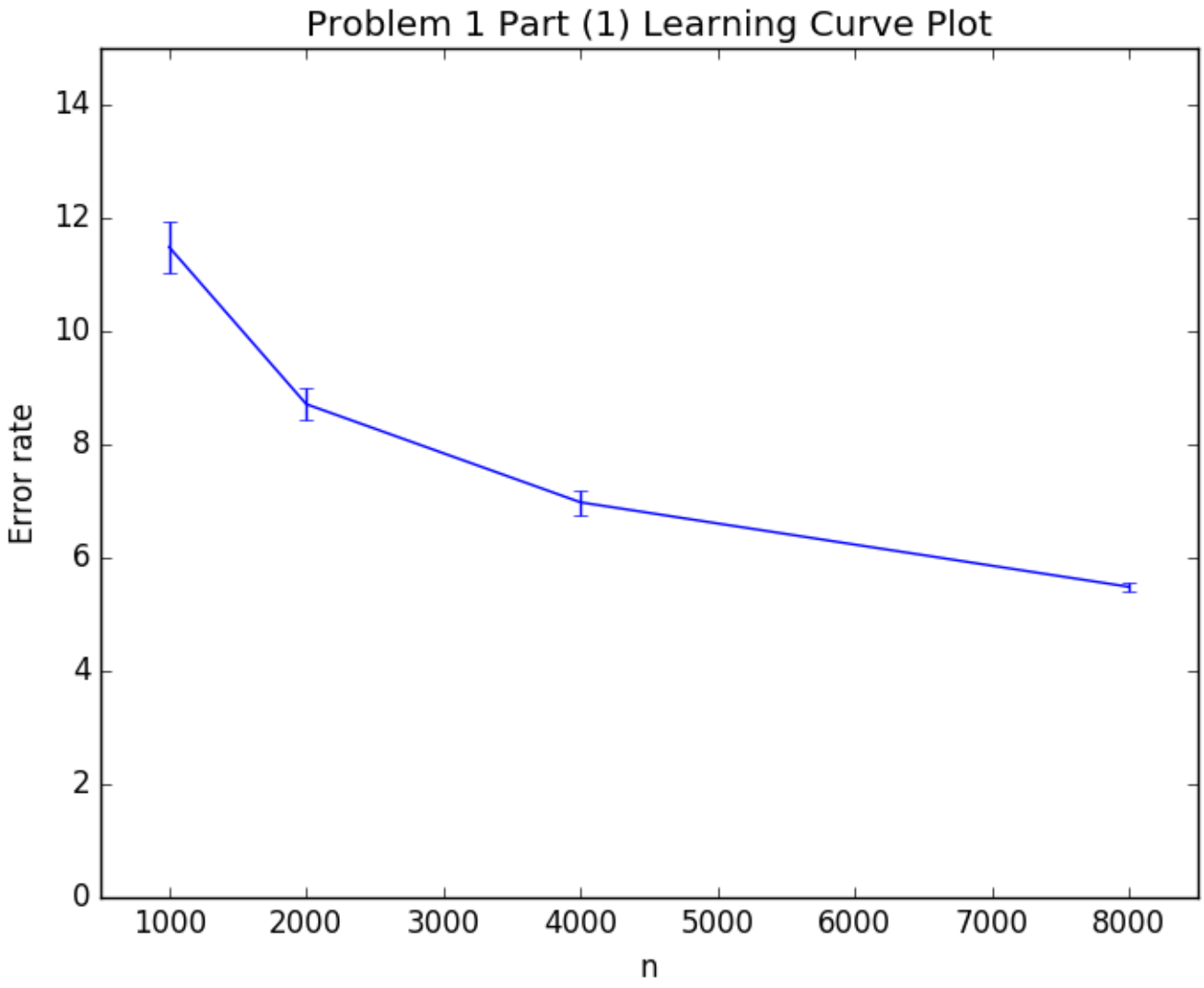
COMS W4771 : Machine Learning - Problem Set #1

Dhananjay Shrouty - ds3521@columbia.edu

September 21, 2016

Problem 1



Problem 2

1. In the prototype I implemented, I am first segregating the entire training data set into small data sets all of which represent a particular distinct label. I am now choosing equal number of data points from all the distinct label datasets (eg. for m=1000, take 100 training data points from all the distinct label training data) so that there is no bias in terms of all data being of the same label. (If for example, all training data that are sampled are of the same label, the prediction will always be a particular label and nothing else). So, as we will have equal data points from all the distinct labels, this training data will represent a good sample. Therefore, it must increase the efficiency of the classifier.

2.

```
Values = [0,1,2,3,4,5,6,7,8,9]
indices_label_0 = np.where(training_labels == 0)[0]                    ▷ Step 1
indices_label_1 = np.where(training_labels == 1)[0]
...
indices_label_9 = np.where(training_labels == 9)[0]
m ⇐ [1000, 2000, 4000, 8000]
iterations ⇐ 10
for i in m do
    x=i/len(Values)
    for j in range(0,iterations) do
        data_index_0 = random.sample(indices_label_0, x)               ▷ Step 2
        data_index_1 = random.sample(indices_label_1, x)
        ...
        data_index_9 = random.sample(indices_label_9, x)
        data_indices = np.append(data_indices,[data_index_0, data_index_1,...,
data_index_9])                                                         ▷ Step 3
        training_data = ocr['data'][data_indices]
        training_labels = ocr['labels'][data_indices]
        error_count = NN_classifier(training_data,training_labels,test_data,test_labels,i,j)
▷ Step 4

    end for
end for
Evaluate Mean Error Rate
Evaluate Standard Deviation of Errors
```

(Step 1) Segregate the data points according to the distinct labels

(Step 2) Take (m/len(Values)) random samples from each distinct label training set

(Step 3) Construct the training data from the chosen indices from all the distinct-label training data sets

(Step 4) Feed this training data along with test data, test labels and training labels in the NN classifier to get the error count and use the error counts in each iteration for m = [1000,2000,4000,8000] to generate the mean error array and standard-deviation-of-error array

3. Test Error Rate table for m=[1000, 2000, 4000, 8000]

m	Error Rate%	Standard Deviation
1000	11.337	0.373
2000	8.567	0.231
4000	6.851	0.204
8000	5.449	0.186

Problem 3

(a) Let E be the event that the first and the second ball have different colors. Because we are picking two balls at random from the urn of 100 balls with replacement, the probability that the first balls is red and the second is not red (denoted as $R^{'}$) is

$$P(R, R^{'}) = \frac{n_r}{100} \times \frac{(100 - n_r)}{100}$$

Similarly,

$$P(G, G^{'}) = \frac{n_g}{100} \times \frac{(100 - n_g)}{100}$$
$$P(Y, Y^{'}) = \frac{n_y}{100} \times \frac{(100 - n_y)}{100}$$
$$P(O, O^{'}) = \frac{n_o}{100} \times \frac{(100 - n_o)}{100}$$
$$P(B, B^{'}) = \frac{n_b}{100} \times \frac{(100 - n_b)}{100}$$

To get the total probability, we add all the probabilities so,

$$P(E) = \frac{100(n_r + n_g + n_y + n_o + n_b) - \sum_{i \in (r,o,g,y,b)} n_i^2}{10000} = 1 - \frac{\sum_{i \in (r,o,g,y,b)} n_i^2}{10000}$$

(b) To maximize $P(E)$, we have to minimize the term $\sum_{i \in (r,o,g,y,b)} n_i^2$, where

$$\sum_{i \in (r,o,g,y,b)} n_i = 100$$

The only solution to minimize this function is by equally dividing the sum by the number of terms, i.e., 5. Therefore, the number of balls of each color comes out to be $100/5 = 20$ and the maximum probability comes out as

$$P(E) = 1 - \frac{\sum_{i \in (r,o,g,y,b)} 20^2}{10000} = \frac{4}{5}$$