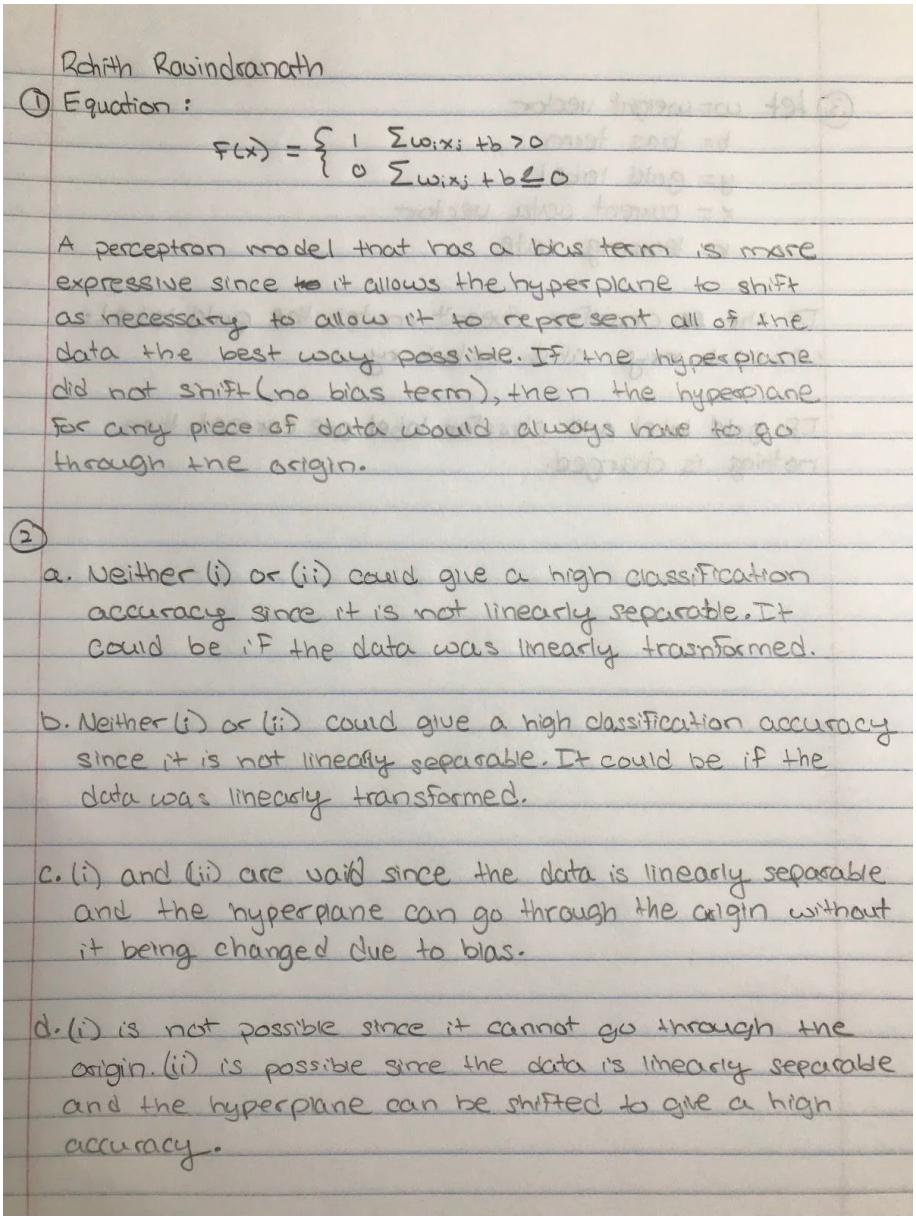
Rohith Ravindranath
PUID: 0028822977
Dan Goldwasser
CS 37300
1st April 2019

CS 373 HW 3

Perceptron



Rohith Ravindranath

① Equation :

$$F(x) = \begin{cases} 1 & \sum w_i x_i + b > 0 \\ 0 & \sum w_i x_i + b \leq 0 \end{cases}$$

A perceptron model that has a bias term is more expressive since it allows the hyperplane to shift as necessary to allow it to represent all of the data the best way possible. If the hyperplane did not shift (no bias term), then the hyperplane for any piece of data would always have to go through the origin.

②

a. Neither (i) or (ii) could give a high classification accuracy since it is not linearly separable. It could be if the data was linearly transformed.

b. Neither (i) or (ii) could give a high classification accuracy since it is not linearly separable. It could be if the data was linearly transformed.

c. (i) and (ii) are valid since the data is linearly separable and the hyperplane can go through the origin without it being changed due to bias.

d. (i) is not possible since it cannot go through the origin. (ii) is possible since the data is linearly separable and the hyperplane can be shifted to give a high accuracy.

③ Let w = weight vector
b = bias term
y = gold label
x = current data vector
r = learning rate

If the classifier doesn't match the gold label:
$w = w + rxy$ and $b = b + ry$.

If gold label and classifier label are correct, then nothing is changed.

Naive Bayes

Rohith Ravindranath

① $P(c^+|d) = \frac{P(d|c^+)P(c^+)}{P(d)} = \frac{P(w_1, w_2, \dots w_n|c^+)}{P(w_1, w_2 \dots w_n)} \propto$

$P(w_1|c^+)\, P(w_2|c^+) \cdot \dots \cdot P(w_n|c^+)$

② IF we cannot make an independence assumption, the $P(d|c^+)$ is correctly estimated by learning all dependent sets of word given c^+. Since there are V subsets of words for each word in a document of length l, we must learn V^l parameters.

③ When dealing with a unigram assumption, we assume each word is conditionally independent when given c^+, with this assumption, we have V choices for every word in a document of length l, so $V \cdot l$ parameters must be learned

④ Say we have a document d that is made up of l rows that can be classified as '+' or '-'. The number of rows w/ a classification of '+' is numPosRows, and the number of negative rows is numNegRows

$P(c^+) = \frac{\text{numPosRows}}{l} \quad P(c^-) = \frac{\text{numNegRows}}{l}$

Analysis

1.

Performance with bias term

```
~/Desktop/cs373-hw3/src python3 main.py
Perceptron Results:
Accuracy: 81.40, Precision: 84.32, Recall: 76.35, F1: 80.14

Averaged Perceptron Results:
Accuracy: 81.06, Precision: 81.37, Recall: 79.72, F1: 80.54
```

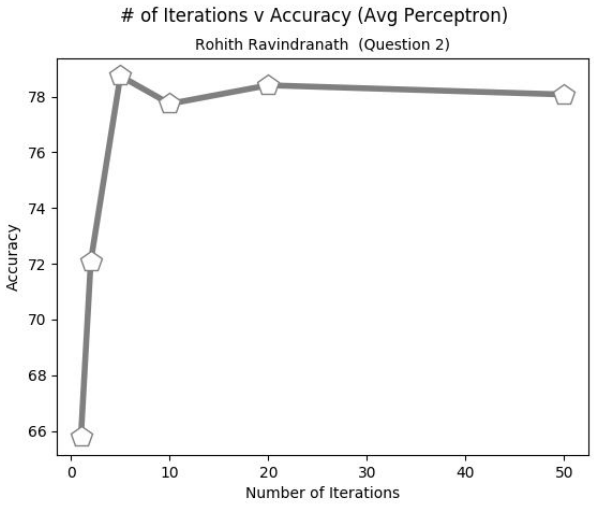
Performance without bias term

```
~/Desktop/cs373-hw3/src python3 main.py
Perceptron Results:
Accuracy: 79.73, Precision: 77.70, Recall: 82.43, F1: 79.99

Averaged Perceptron Results:
Accuracy: 78.07, Precision: 78.87, Recall: 75.67, F1: 77.24
```

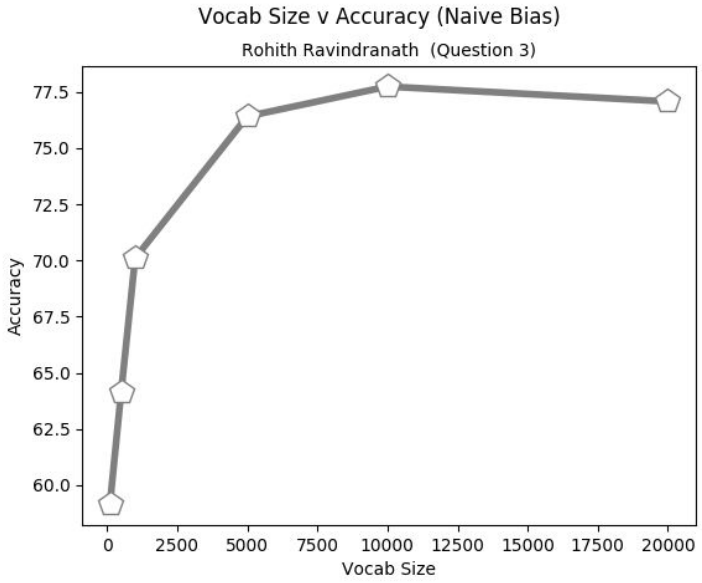
From these two screenshots we notice that the difference between the two models (with bias and without bias) is pretty negligible to say that their difference is statistically significance.

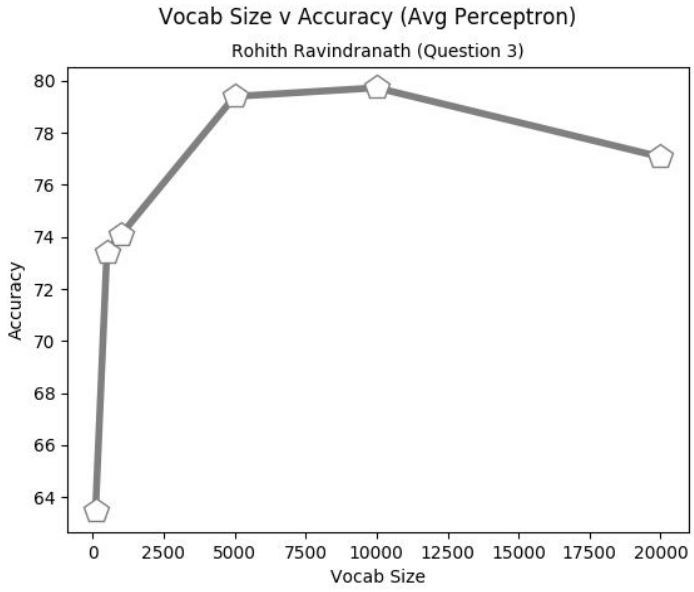
2.



Yes, my Perceptron model converges after 9 iterations during training.

3.





It seems when we change the vocabulary size, the graph resembles that of a logarithmic functions. We can see that the slope starts to stop increasing around the 7500 vocab size range. The constant slope in Naive Bayes and slow dip in Avg Perceptron after vocab size 10000 could be due to over-fitting the date.