Name: Bosheng Li

PUID: 0028946785

2. Perceptron

1.

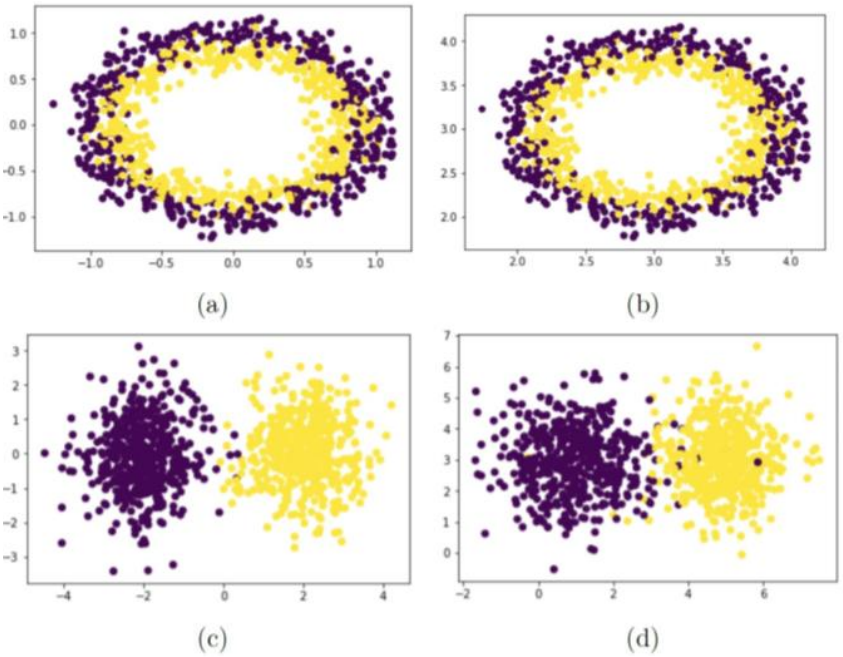
Here is the equation with the bias:

$$\boldsymbol{f(x)=\begin{cases}1, & \sum w_jx_j+b>0\\ 0, & \sum w_jx_j+b\leq 0\end{cases}}$$

The perceptron with bias term is more expressive as it allows the hyperplane to not intersect with the origin. A bias is required for those hyperplanes which are not intersect with the origin. Hence the equation with bias is more expressive.

2.

I pasted the image from the handout:



(a)

(b)

(c)

(d)

For (a) and (b): None of the choices would give a high classification accuracy because these two graphs cannot be linearly separated.

For (c): Since the hyperplane almost intersects with the origin, both choices, (i) and (ii) (with or without bias) would give a good classification accuracy.

For (d): Only (ii) would give a high perceptron accuracy since the hyperplane needs to pass through origin.

3.

The update rule for a vanilla perceptron is bias = bias + learning rate * gold label. For the perceptron algorithm is an error-driven algorithm, we only update the bias when an error is occurred.

3. Naïve Bayes

1.

$$\mathrm{P}(\boldsymbol{c}^{+}|\mathbf{d}) = \frac{\boldsymbol{P}(\boldsymbol{d}|\boldsymbol{c}^{+})\boldsymbol{P}(\boldsymbol{c}^{+})}{\boldsymbol{P}(\boldsymbol{d})}$$

2.

Since $\mathrm{P}(\mathbf{d}|\boldsymbol{c}^{+}) = \boldsymbol{P}(\boldsymbol{w}_{1} \cdot \boldsymbol{w}_{2} \cdot \boldsymbol{w}_{3} \cdot \boldsymbol{w}_{4} \cdot \ldots \boldsymbol{w}_{n}|\boldsymbol{c}^{+})$ as w stands of different words of the documents, we need V * l parameters.

3.

Assume occurrence of each word in the document is independent of the other words, $\mathrm{P}(\mathbf{d}|\boldsymbol{c}^{+}) = \boldsymbol{P}(\boldsymbol{w}_{1} \cdot \boldsymbol{w}_{2} \cdot \boldsymbol{w}_{3} \cdot \boldsymbol{w}_{4} \cdot \ldots \boldsymbol{w}_{n}|\boldsymbol{c}^{+}) = \boldsymbol{P}(\boldsymbol{w}_{1}|\boldsymbol{c}^{+}) \cdot \boldsymbol{P}(\boldsymbol{w}_{2}|\boldsymbol{c}^{+}) \cdot \boldsymbol{P}(\boldsymbol{w}_{3}|\boldsymbol{c}^{+}) \cdot \boldsymbol{P}(\boldsymbol{w}_{4}|\boldsymbol{c}^{+}) \ldots \boldsymbol{P}(\boldsymbol{w}_{n}|\boldsymbol{c}^{+})$**. In this case we need only V parameters.**

4.

$$\mathrm{P}(\boldsymbol{c}^{+}) = \frac{\boldsymbol{size}(\boldsymbol{c}^{+})}{\boldsymbol{size}(\boldsymbol{c}^{+}) + \boldsymbol{size}(\boldsymbol{c}^{-})}$$

$$\mathrm{P}(\boldsymbol{c}^{-}) = \frac{\boldsymbol{size}(\boldsymbol{c}^{-})}{\boldsymbol{size}(\boldsymbol{c}^{+}) + \boldsymbol{size}(\boldsymbol{c}^{-})}$$

[Next part on next page.]

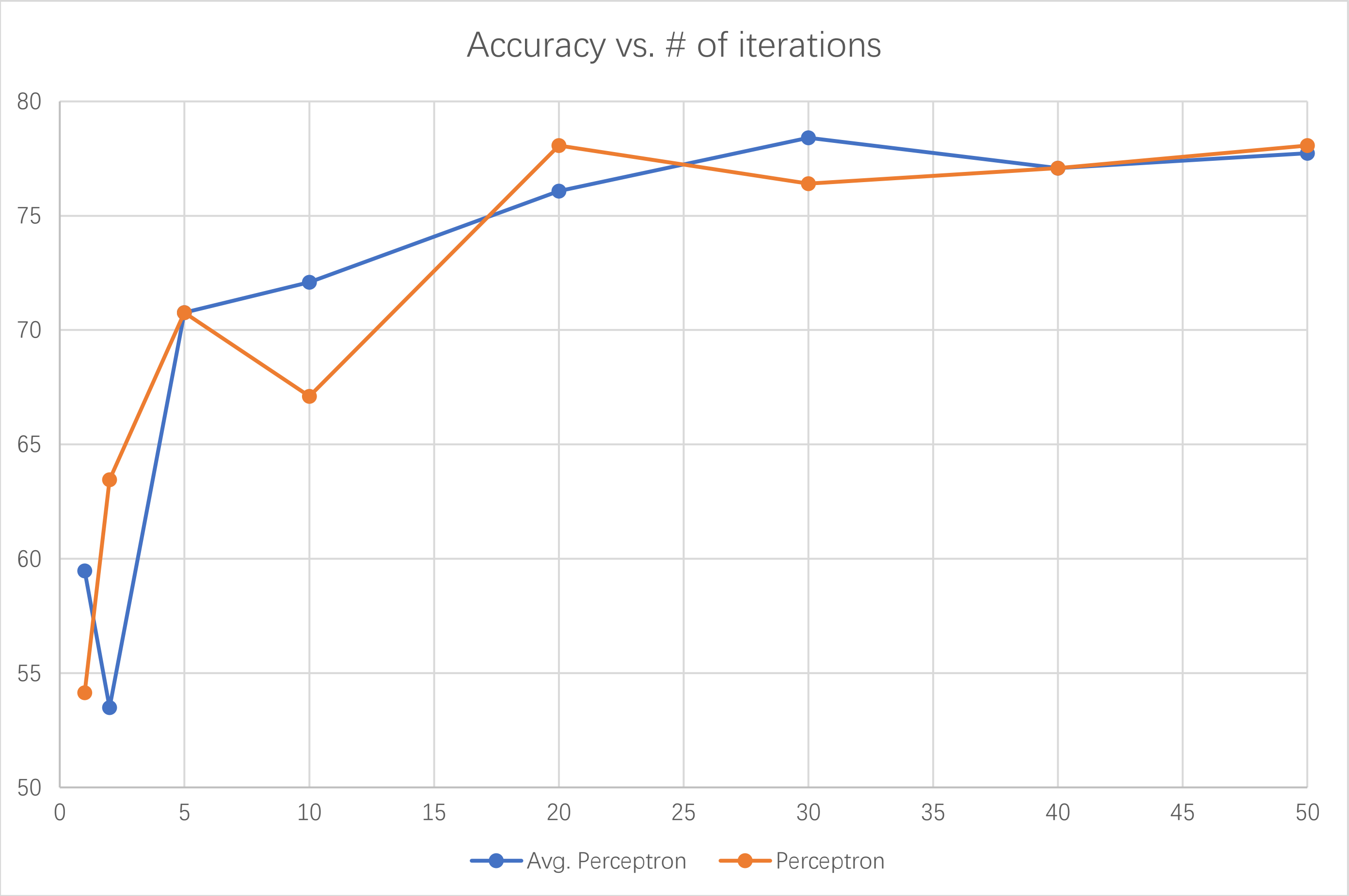
4. Analysis:

1.

The performance is worse. This is because the bias allows the model to be more expressive as it allows the hyperplane to not intersect with the origin.

2.

The plot is showed below:



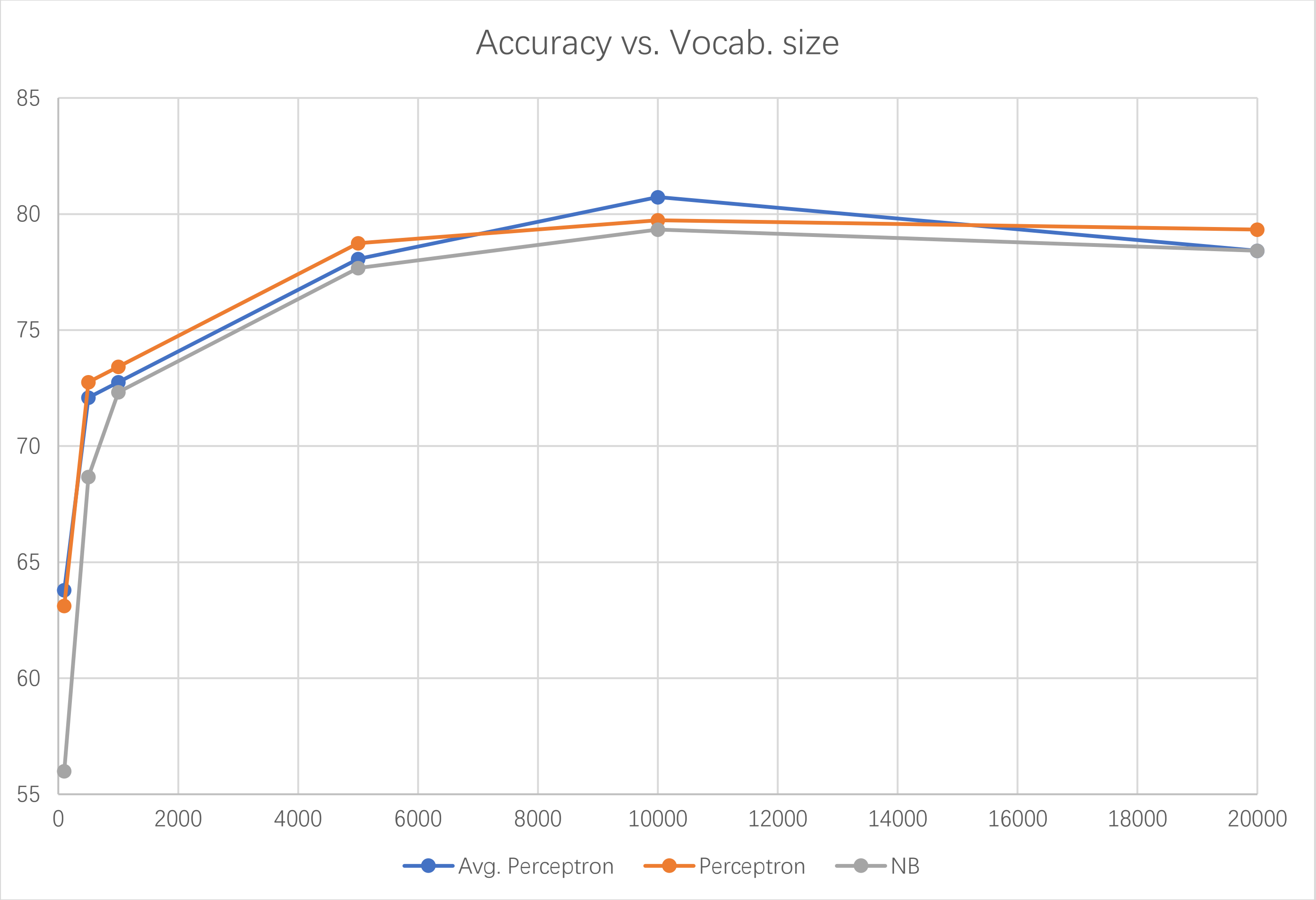
In our example, the accuracy does not reach 100% but get very close to it after 30 numbers of iteration, so I say our example converged after 30 times of iteration.

Does Perceptron converge? I should say if the dataset is linearly separable, the perceptron will converge since the perceptron is a linear classifier. However, for the accuracy, it may never reach 100% no matter how many iterations because of the noises in the dataset.

I think the question is more likely to ask about the number of iterations to achieve a converge, in that case, the perceptron algorithm converges after making $O(R^2/\gamma^2)$ updates.

3.

The plot is showed below:



The performance increases rapidly when the vocab size is low, especially for NB algorithm, but slows down as the vocab size pass 1000. We have the best accuracy at around 10000 words.