**CPT_S 570 HW4**

**Yang Zhang**

**11529139**

**1.** The paper states two points: (1) The discriminative learning has lower asymptotic error than generative learning. (2) The generative learning may approach its asymptotic error faster than discriminative learning.

**2. (a)** According to the paper from question 1, it really depends on what rule to measure the performance. Logistic Regress has lower asymptotic error, but Naïve Bayes approach its asymptotic error faster.

**(b)** Logistic Regress is better, because LR doesn't require the data to be independent labelling, so its result won't be affected by independence. However, the Naïve Bayes do require independence, so the result from Naïve Bayes is unreliable is the data does not satisfy the assumption.

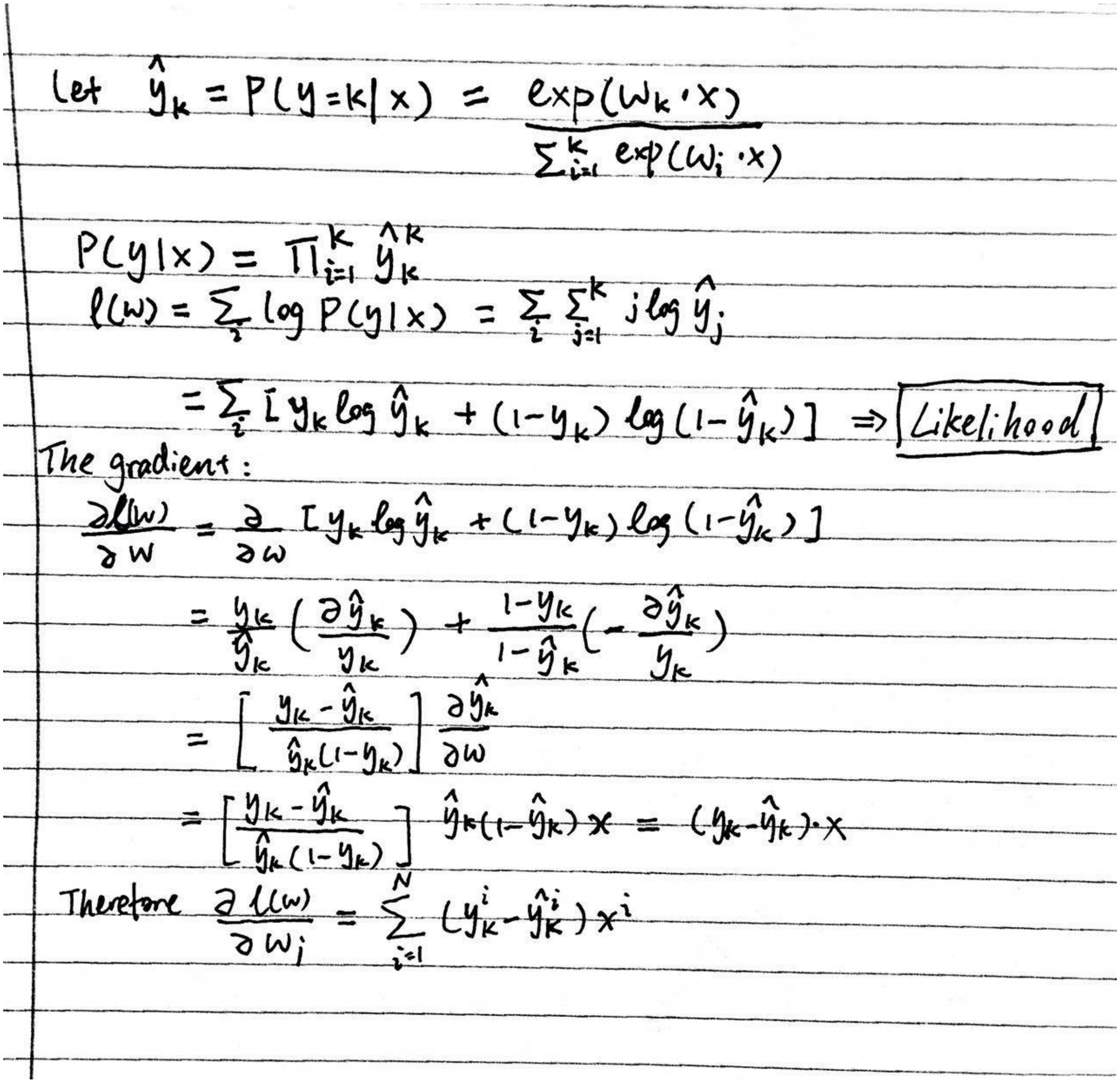
**3.(a) Can we compute P(X) from the learned parameters of a Naive Bayes classifier? Please explain your reasoning.**

Yes, because Naïve Bayes involve computing P(x|y) and P(y). we can compute P(x) from the sum of $P(x|y_i)P(y_i)$

**b. Can we compute P(X) from the learned parameters of a Logistic Regression classifier? Please explain your reasoning.**

No, because Logistic Regression learns P(y|x) directly.

**4. (a)**



let $\hat{y}_k = P(y=k|x) = \frac{\exp(w_k \cdot x)}{\sum_{i=1}^{k} \exp(w_i \cdot x)}$

$$P(y|x) = \Pi_{i=1}^{k} \hat{y}_k^{k}$$

$$l(w) = \sum_i \log P(y|x) = \sum_i \sum_{j=1}^{k} j \log \hat{y}_j$$

$$= \sum_i [y_k \log \hat{y}_k + (1-y_k)\log(1-\hat{y}_k)] \Rightarrow \boxed{\text{Likelihood}}$$

The gradient:

$$\frac{\partial l(w)}{\partial w} = \frac{\partial}{\partial w}[y_k \log \hat{y}_k + (1-y_k)\log(1-\hat{y}_k)]$$

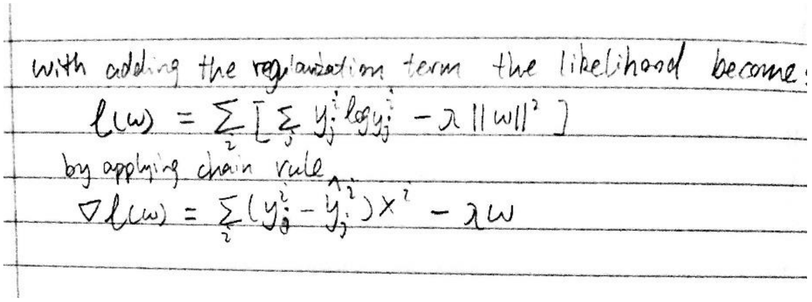
$$= \frac{y_k}{\hat{y}_k}\left(\frac{\partial \hat{y}_k}{y_k}\right) + \frac{1-y_k}{1-\hat{y}_k}\left(-\frac{\partial \hat{y}_k}{y_k}\right)$$

$$= \left[\frac{y_k - \hat{y}_k}{\hat{y}_k(1-y_k)}\right]\frac{\partial \hat{y}_k}{\partial w}$$

$$= \left[\frac{y_k - \hat{y}_k}{\hat{y}_k(1-y_k)}\right]\hat{y}_k(1-\hat{y}_k)x = (y_k - \hat{y}_k)\cdot x$$

Therefore $\frac{\partial l(w)}{\partial w_j} = \sum_{i=1}^{N} (y_k^i - \hat{y}_k^i) x^i$

(b)



with adding the regularization term the likelihood become:

$$l(w) = \sum_i [\sum_j y_j^i \log y_j^i - \lambda ||w||^2]$$

by applying chain rule,

$$\nabla l(w) = \sum_i (y_j^i - \hat{y}_j^i) x^i - \lambda w$$

**5.** This paper compared 5 tests for the propose of picking the algorithm with better performance. As a result, the test for the difference of two proportions and paired-differences t test (based on random train/test split) should never be used, because those 2 tests are shown Type I error in certain cases. While the 10 fold cross validation test somewhat elevates the Type I error. In contrast, McNemar's tests are shown to have low Type I error. The last test 5x2cv also provides the acceptable Type I error. As the matter of detecting differences between algorithms, the cross-validation test is the most powerful test.

**6.** This paper talked about the overfitting issue of machine learning. In most of the cases, the general objective function of a ML algorithm is to minimize the in sample error. However, by achieving this objective, the noise data has possibly been fitted as well, this is not the general solution we want. Under computing is one of the way to avoid overfitting.

**7.** This paper provided detailed introduction of popular ensemble methods (Adaboost, bootstrap .etc) and states that ensemble methods often perform better than any single classifier. The paper also explained why in 3 aspects: (1) statistical (2) computational (3) representational.

**8.** Result:

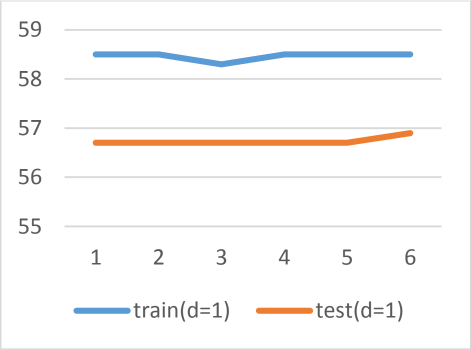
```
Training Accuracy: 0.953416149068323
Testing Accuracy: 0.7920792079207921
```

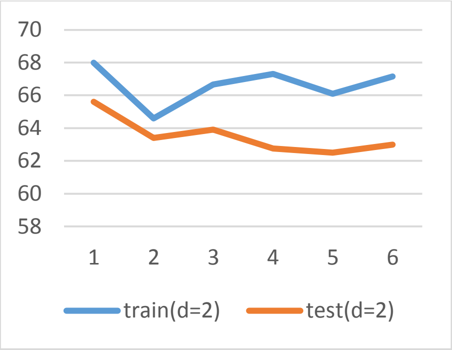
The training accuracy is higher than testing accuracy

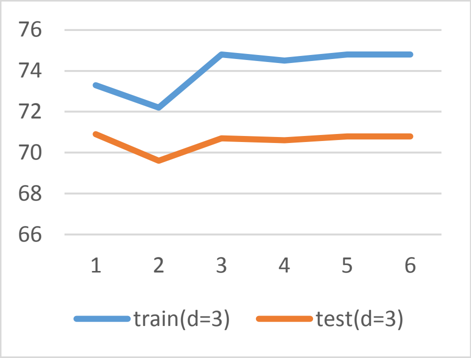
**9**

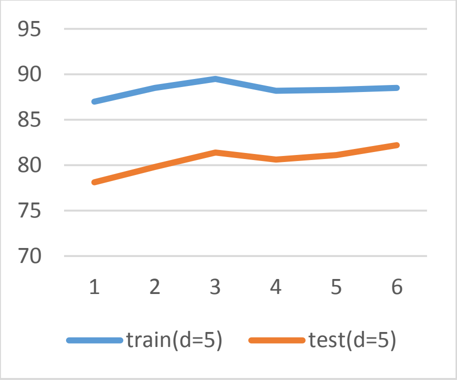
**Plots for bagging (average result from two datasets):**

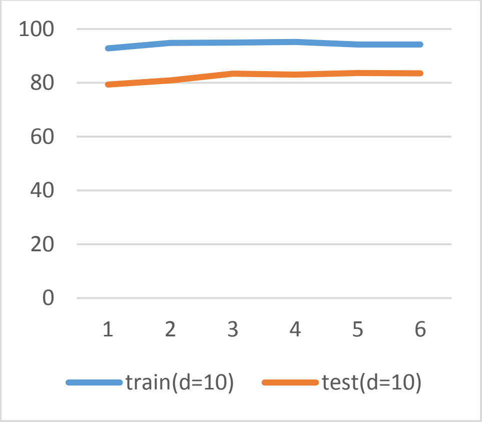
**X-axis mapping {1->10, 2->20, 3->40, 4->60, 5->80, 6->100}**







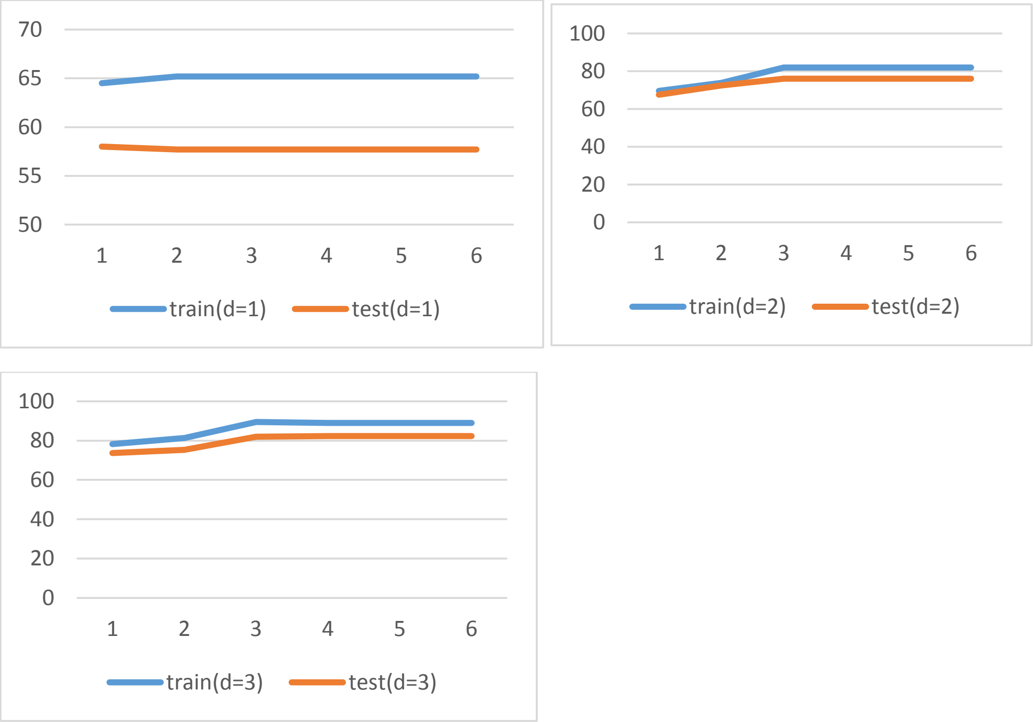




The accuracy depends on tree depth a lot more than the bag size and for smaller tree depth comes with lower accuracy but higher convergence speed. In contrast, bigger tree depth comes with higher accuracy but slower convergence speed.

**Plots for bagging (average result from two datasets):**

**X-axis mapping {1->10, 2->20, 3->40, 4->60, 5->80, 6->100}**



The accuracy depends on tree depth a lot more than the number of iterations and for smaller tree depth comes with lower accuracy but higher convergence speed. In contrast, bigger tree depth comes with higher accuracy but slower convergence speed.