Cpts_570: Machine Learning

Homework III

10/14/2019

Instructor: Janardhan Rao (Jana) Doppa

Student: Yu-Chieh Wang 11641327

1. Analytical Part

1. Jensen's inequality $f(E[X]) \leq E[f(X)]$

   a. Suppose $x = (x_1, x_2, \ldots x_d)$ and $z = (z_1, z_2, \ldots z_d)$. Prove that:

$$(\frac{1}{\sqrt{d}} \sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}} \sum_{i=1}^{d} z_i)^2 \leq \sum_{i=1}^{d} (x_i - z_i)^2$$

sol.

1. $f(x) = x^2$. We assume $x = x_i - z_i$, then we get $f(x) = (x_i - z_i)^2$

2. $E[x] = x_i * P(x_i)$. We assume $x = x_i - z_i$, so we get

$$E[x] = E[x_i - z_i] = (\frac{1}{d} * x_1 + \frac{1}{d} * x_2 + \ldots \frac{1}{d} * x_d) - (\frac{1}{d} * z_1 + \frac{1}{d} * z_2 + \ldots \frac{1}{d} * z_d)$$

$$= \frac{1}{d} \sum_{i=1}^{d} x_i - \frac{1}{d} \sum_{i=1}^{d} z_i = \frac{1}{d}(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i).$$

Next, we use Jensen's inequality $f(E[X]) \leq E[f(X)]$:

combine with the first and the second equations, we get:

$f(x) = x^2 \rightarrow f(E[x]) = (\frac{1}{d}(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i))^2$ and

$$E[x] = E[x_i - z_i] \rightarrow E[f(x)] = E[(x_i - z_i)^2] = \frac{1}{d} \sum_{i=1} d(x_i - z_i)^2.$$

Finally, we get $f(E[X]) \leq E[f(X)] = (\frac{1}{d}(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i))^2 \leq \frac{1}{d}\sum_{i=1} d(x_i - z_i)^2$

$$= (\frac{1}{\sqrt{d}}\sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d} z_i)^2 \leq \sum_{i=1}^{d} (x_i - z_i)^2.$$

b. Using this equation, we can give a boundary to limit the number of calculations, which results in lower costs.

2. According to the article[1], we know that Locality sensitive Hashing(LSH) is a method to get the value of the nearest point by calculating the Euclidean distance. In addition, it uses hash function to calculate the probability of each point. When the probability shows lower, the point we count is far from the original one. On the other hand, when the probability is higher, the point is close to the original point, and we can get the value of this point.

3. It is possible to convert the rule set R into an equivalent decision tree. First of all, we consolidate the scope of all the rules. Then, we arrange the numbers for all conditions into a list. Finally, we use the dichotomy, treating the middle value as a point and dividing the list into two lists. Repeat this step until all the lists are cleared.

4. According to the article[2], people traditionally think that Logistic Regression works better than Naive Bayes because of some reasons such as solving problems directly, addressing missing data easily, and helping validation by its linear parameters. However, after experiments we find that although the asymptotic error of discriminative model is lower, it converges very slowly. On the other hand, the generative classifier cannot do

better at the beginning, but it can reduces the error quickly. Therefor, when training data set size is smell, Naive Bayes can may have better result.

5. Logistic regression v.s. Naive Bayes
   a. Naive Bayes will produce better because the features of the training data are independent, the compute result will converge soon which makes the accuracy higher.
   b. Logistic regression will produce better because when the training data approaches infinity, the Logistic regression performs better.
   c. No, we cannot compute $P(X)$ from the learned parameters of a Naive Bayes classifier because the algorithm doesn't provide parameters.
   d. Yes, we can compute $P(X)$ from the learned parameters of a Logistic Regression classifier. Based on the algorithm of Logistic regression: $y = \dfrac{e^{f(x)}}{1 + e^{f(x)}}$, $f(x) = B_0 + X_1B_1 + X_2B_2 \ldots X_nB_n$ which $B_0, B_1, B_2 \ldots B_n$ are coefficients, and we can use them to compute $P(X)$.

6. According to the article[3], the author points that we should focus on raising the accuracy of testing data but training data. Sometime we want to optimizate our algorithm to make the accuracy of training data higher, but we also overfit the prediction model at the same time. To avoid overfitting problem, we should make our problems undercomputing.

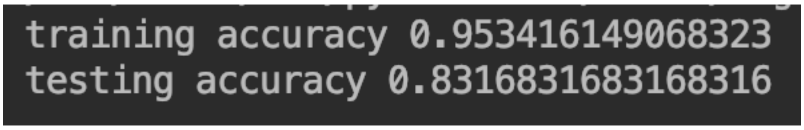
7. This article compare mentions that there are five methods people can use to compare any two algorithms such as McNemar's test, the difference of two proportions test,

resampled paired t test, k-fold cross- validated paired t test and 5xcv paired t test.

2. Programming Part

1. Naive Bayes

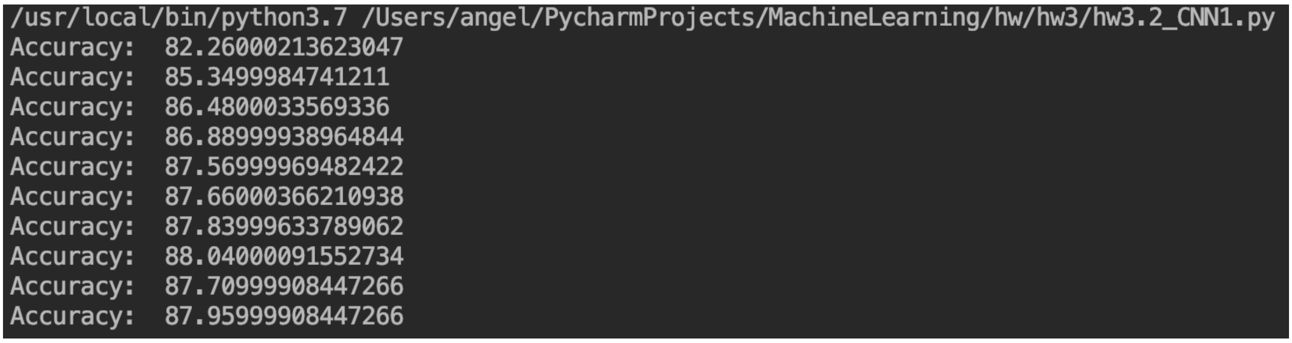
-Run the code and get the accuracies as follow:



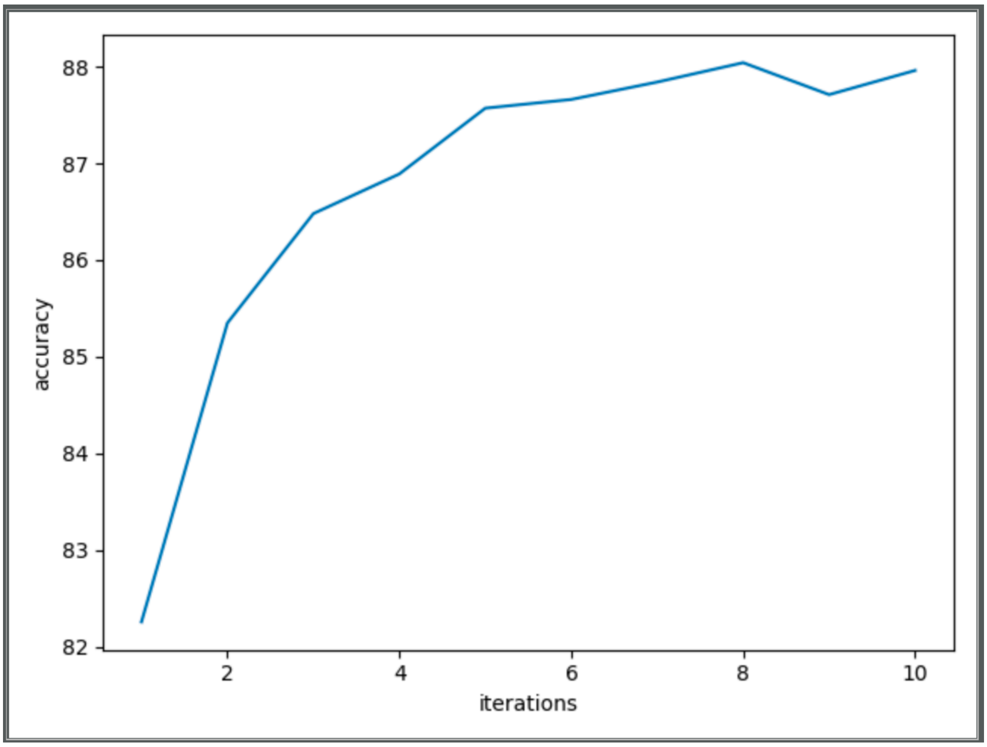
```
training accuracy 0.953416149068323
testing accuracy 0.8316831683168316
```

2. CNN

-Modify TA's code and get the testing accuracy as follow:



```
/usr/local/bin/python3.7 /Users/angel/PycharmProjects/MachineLearning/hw/hw3/hw3.2_CNN1.py
Accuracy:  82.26000213623047
Accuracy:  85.3499984741211
Accuracy:  86.4800033569336
Accuracy:  86.88999938964844
Accuracy:  87.56999969482422
Accuracy:  87.66000366210938
Accuracy:  87.83999633789062
Accuracy:  88.04000091552734
Accuracy:  87.70999908447266
Accuracy:  87.95999908447266
```



Reference

[1]Andoni, A. and Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1), p.117.

[2]Andrew Y. Ng, Michael I. Jordan: On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. NIPS 2001: 841-848

[3]Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM Computing Surveys*, 27(3), pp.326-327.

[4]Thomas G. Dietterich: Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. Neural Computation 10(7): 1895-1923 (1998)