Assignment

Homework 02

Andrew Park

Applied Machine Learning Homework Assignment

CORNELL TECH | HOME OF THE JACOBS TECHNION-CORNELL INSTITUTE

September 23, 2023

1. General Summary

Homework 2 proved to be considerably more challenging compared to our previous assignments, yet it offered a valuable opportunity to put into practice the various mathematical theories we had learned in class when working with real-world models. I successfully completed the assignment by reviewing what I learned from the class, and seeking assistance from my peers on topics I found challenging, particularly when it came to comprehending and applying the n-gram model, which required a significant amount of time and effort.

During my studies, I also gained a profound understanding of how even minor variations in limit values can substantially impact the success of predictions. This became especially apparent when performing predictions after applying techniques like l1 regression and l2 regression for data normalization. I observed that these techniques led to notable increases in numerical values, reinforcing the crucial role of data preprocessing.

Furthermore, while tackling the first problem of our programming exercises, I learned the inherent difficulty in completely eliminating random noise from a model. Additionally, I came to appreciate the vital importance of data set independence for achieving high-performance models. This realization underscored the significance of meticulous data preprocessing and the necessity of addressing noise when working with real-world data. At this point, I noticed that my predictions which used l1 regularization were lower than expected, possibly due to the inherent randomness in the selected data type and the generation process. I found this aspect disappointing, and I plan to explore potential solutions to address this issue in the future.

Regarding the additional bonus questions, it was fascinating to plot graphs and visually observe how the graphs converged towards the given lines. This hands-on experience added an extra layer of interest to the task.

For the written exercise, it was interesting to review over the Naive Bayes equation by my self, and it was beneficial to revisit and reinforce the mathematical foundations of this theory.

WRITTEN EXERCISES

Problem 1
Naive Bayes with Binary Features

Solution.

a. In the context of the question, the Naive Bayes assumption asserts that the features used to predict the target class (in this case, whether a student is a Master's or PhD student) are conditionally independent or unrelated to any of the other features in the model. It also assumes that all features contribute equally to the outcome. Thus, when we know whether a student is a PhD or Master's student, this assumption implies that the probability of the student biking is independent of whether the student likes to ski, and vice versa. In simpler terms, if we know a student's academic status (PhD or Master's), according to this assumption, knowing whether they bike or ski doesn't provide any additional information about their skiing or biking habits.

b. In this question, we are interested in finding the probability $P(\text{Master's}|\text{notlike Bike\&notlike Ski})$.
We can use Bayes' theorem to calculate this probability:

$$P(\text{Master's}|\text{notlike Bike\&notlike Ski}) = \frac{P(\text{notlike Bike\&notlike Ski}|\text{Master's}) \cdot P(\text{Master's})}{P(\text{notlike Bike\&notlike Ski})}$$

First, let's consider the numerator, $P(\text{notlike Bike\&notlike Ski}|\text{Master's})$:

$$P(\text{notlike Bike\&notlike Ski}|\text{Master's}) = (1 - 0.25) \cdot (1 - 0.25) = 0.5625$$

Now, for the denominator, $P(\text{notlike Bike\&notlike Ski})$:

$$P(\text{notlike Bike\&notlike Ski}) = P(\text{Master's}) \cdot P(\text{notlike Bike\&notlike Ski}|\text{Master's})$$

$$+P(\text{PhD}) \cdot P(\text{notlike Bike\&notlike Ski}|\text{PhD})$$

We are provided with $P(\text{Master's}) = 0.4$ and we've already found $P(\text{notlike Bike\&notlike Ski}|\text{Master's})$ to be 0.5625. As for $P(\text{PhD})$ and $P(\text{notlike Bike\&notlike Ski}|\text{PhD})$, we know $P(\text{PhD}) = 0.6$, and since every PhD student who skis also bikes, $P(\text{notlike Bike\&notlike Ski}|\text{PhD}) = \frac{1}{3} \cdot 0.5$.
Now, let's calculate $P(\text{notlike Bike\&notlike Ski})$:

$$P(\text{notlike Bike\&notlike Ski}) = 0.4 \cdot 0.5625 + 0.6 \cdot \left(\frac{1}{3} \cdot 0.5\right) = 0.225 + 0.1 = 0.325$$

Substituting these values back into the equation:

$$\frac{0.5625 \cdot 0.4}{0.325} = 0.6923$$

Therefore, the probability of a student who neither bikes nor skis being a Master's student is approximately 69.23%.

c. Given the altered information, it's evident that skiing and biking are not entirely independent activities for PhD students. This implies that the Naive Bayes assumption, which assumes feature independence, no longer holds in this context. Consequently, when recalculating part b with these updated considerations, the probability denominator P(notlike Bike&notlike Ski|Master's) $\cdot$ P(Master's)/P(notlike Bike&notlike Ski) should yield a different result.

To determine P(notlike Bike&notlike Ski|PhD), we must account for the fact that some PhD students who ski also bike. Therefore, P(notlike Bike&notlike Ski|PhD) can be calculated as:

$$P(\text{notlike Bike\&notlike Ski|PhD}) = 1 - P(\text{Bike\&notlike Ski|PhD}) - P(\text{Bike\&Ski|PhD})$$

$$= 1 - \frac{1}{6} - \frac{1}{2} = \frac{1}{3}$$

Now, let's reevaluate the denominator:

$$0.4 \cdot 0.5625 + 0.6 \cdot \left(\frac{1}{3}\right) = 0.225 + 0.2 = 0.425$$

Substituting these values back into the equation:

$$\frac{0.5625 \cdot 0.4}{0.425} = 0.5294$$

Consequently, the probability of a student who neither engages in biking nor skiing being a Master's student is approximately 52.94%. This revised probability is lower than the previous estimate due to the updated consideration of the non-independence of skiing and biking among PhD students.

Problem 2
Categorical Naive Bayes

Solution.

- Given $P_\theta(y = k) = \phi_k$, we can express the likelihood function as $L(\theta) = \prod_{i=1}^{n} P_\theta(x^{(i)}, y^{(i)})$.

$$L(\theta) = \prod_{i=1}^{n} \log P_\theta(x^{(i)}, y^{(i)})$$

where $\log P_\theta(y^{(i)}) = \log \phi_k$.

Setting the derivative to zero, we get:

$$\frac{\partial}{\partial \phi_k} \sum_{y^{(i)}=k} \log \phi_k = 0$$

Solving for ϕ_k:

$$\frac{n_k}{\phi_k} = 0, \quad \phi_k = \frac{n_k}{n}$$

Hence, we find that the maximum likelihood estimate for ϕ_k is $\phi_k^* = \frac{n_k}{n}$.

- Given $P_\theta(x_j = l|y = k) = \Psi_{jkl}$, the likelihood function is expressed as $L(\theta) = \prod_{i=1}^{n} P_\theta(x^{(i)}, y^{(i)})$.

$$L(\theta) = \sum_{i=1}^{n} \log P_\theta(x^{(i)}, y^{(i)}), \text{ where } \log P_\theta(x_j^{(i)} = l, y^{(i)} = k) = \log \Psi_{jkl}$$

Setting the derivative to zero, we obtain:

$$\frac{\partial}{\partial \Psi_{jkl}} \sum_{x_j^{(i)}=l, y^{(i)}=k} \log \Psi_{jkl} = 0$$

Solving for Ψ_{jkl}:

$$\frac{n_{jkl}}{\Psi_{jkl}} = 0$$

$$\Psi_{jkl} = \frac{n_{jkl}}{n_k}$$

Therefore, we have demonstrated that the maximum likelihood estimate for $\Psi_{jkl}^* = \frac{n_{jkl}}{n_k}$.

Problem 3
Bonus Point

Solution.

In relation to the graph provided, it's observable that both Alpha and Beta tend to approach the red line (Figure 2), which represents their initial values, as the sample sizes expand. This observation underscores the idea that the coefficients of the linear regression gradually approach and , which are intrinsic to the data generation process.

```python
import matplotlib.pyplot as plt
# extra credit
alpha_list = []
beta_list = []

sample_sizes = np.arange(10, 10000, 10)

for n in sample_sizes:
    X, Y = function(n)
    model = LinearRegression().fit(X, Y)

    alpha_list.append(model.intercept_[0])
    beta_list.append(model.coef_[0][0])

plt.figure(figsize=(12,6))

plt.subplot(1,2,1)
plt.plot(sample_sizes, alpha_list, label='Estimated Alpha')
plt.axhline(y=alpha, color='r', linestyle='--', label=f'True Alpha = {alpha}')
plt.xlabel('Sample Size')
plt.ylabel('Alpha')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(sample_sizes, beta_list, label='Estimated Beta')
plt.axhline(y=beta, color='r', linestyle='--', label=f'True Beta = {beta}')
plt.xlabel('Sample Size')
plt.ylabel('Beta')
plt.legend()

plt.tight_layout()
plt.show()
```

Figure 1: Codes to observe how Alpha and Beta change with increasing sample size

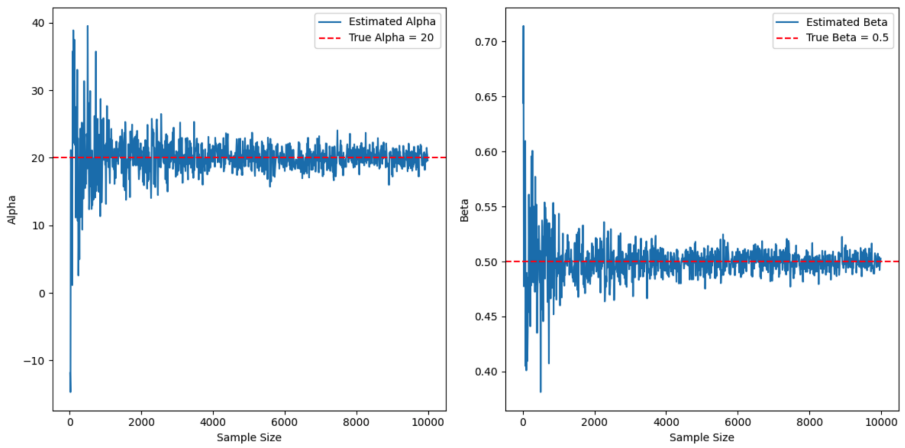

Figure 2: Both Alpha and Beta are converging to the red line as the sample size increases.