İpek Öztaş
22003250
CS464-Sec2

CS464 Introduction to Machine Learning

Fall 2023

Homework 1

İpek Öztaş

22003250

Section-2

Contents

İpek Öztaş
22003250
CS464-Sec2

Probability Review

1.1

Let's denote the colors blue as B, yellow as Y, red as R and the boxes B1 and B2 respectively. We can create a table with the probabilities of selecting a coin from the given box as follows:

	Blue (Fair)	Yellow (25% Heads)	Red (10% Heads)
Box1	P(B|B1) = 2/3	P(Y|B1) = 1/3	P(R|B1) = 0
Box2	P(B|B2) = 1/2	P(Y|B2) = 0	P(R|B2) = 1/2

Let's also denote getting heads as H and tails as T. Then we can form the following table for getting two heads in a row for different coin types:

	Blue (Fair)	Yellow (25% Heads)	Red (10% Heads)
P(HH|x)	P(HH|B) = 0.5*0.5 = 0.25	P(HH|Y) = 0.25*0.25 = 0.0625	P(HH|R) = 0.1*0.1 = 0.01

Calculate the probability of getting two heads in a row:

- Probability of getting heads with a blue coin (fair) = 0.5 * 0.5 = 0.25
- Probability of getting heads with the yellow coin = 0.25 * 0.25 = 0.0625
- Probability of getting heads with the red coin = 0.1 * 0.1 = 0.01

Combine the probabilities from the blue and yellow coins in Box 1:

- P(HH|B1) = P(B|B1) * P(HH|B) + P(Y|B1) * P(HH|Y)
- P(HH|B1) = 2/3 * 0.25 + 1/3 * 0.0625

Combine the probabilities from the blue and red coins in Box 2:

- P(HH|B2) = P(B|B2) * P(HH|B) + P(R|B1) * P(HH|R)
- P(HH|B2) = 1/2 * 0.25 + 1/2 * 0.01

Now, calculate the overall probability of getting two heads regardless of which box is chosen:

- P(HH) = P(HH|B1) * P(B1) + P(HH|B2) * P(B2) (Marginalization)

It is clear that we have the same possibility to choose box 1 or box 2. Hence, both P(B1) and P(B2) are equal to 0.5.

P(HH) = 2/3 * 0.25 * ½ + 1/3 * 0.0625 * 0.5 + 1/2 * 0.25 * 1/2 + 1/2 * 0.01 * 0.5 = **0.15875**

İpek Öztaş
22003250
CS464-Sec2

1.2

Here we need to find the probability of selecting Blue coin (fair) if we have a prior knowledge on getting two heads in a row.

In order to find that, we use the Bayes rule,

$$P(B|HH) = \frac{P(HH|B) * P(B)}{P(HH)}$$

P(HH|B) = 0.5 * 0.5 = 0.25

P(B) = P(B|B1) * P(B1) + P(B|B2) * P(B2) = 2/3 * 0.5 + 0.5 * 0.5 = 7/12

P(HH) = 2/3 * 0.25 * ½ + 1/3 * 0.0625 * 0.5 + 1/2 * 0.25 * 1/2 + 1/2 * 0.01 * 0.5 = 0.15875

Hence,

$$= \frac{0.25 * 7/12}{0.15875} \approx \mathbf{0.91864}$$

1.3

The calculations are the same with the previous part basically. Here we need to find the probability of selecting red coin if we have a prior knowledge on getting two heads in a row.

In order to find that, we use the Bayes rule,

$$P(R|HH) = \frac{P(HH|R) * P(R)}{P(HH)}$$

P(HH|R) = 0.1*0.1 = 0.01

P(R) = P(R|B1) * P(B1) + P(R|B2) * P(B2) = 0 * ½ + ½ * ½ = 0.25

P(HH) = 2/3 * 0.25 * ½ + 1/3 * 0.0625 * 0.5 + 1/2 * 0.25 * 1/2 + 1/2 * 0.01 * 0.5 = 0.15875

Hence,

$$= \frac{0.01 * 1/4}{0.15875} \approx \mathbf{0.01575}$$

MLE and MAP

2.1

We should maximize the likelihood function. The likelihood function for a normal distribution is given by the product of the probability density functions (PDFs) for each data point. The log-likelihood function is often used for convenience, as it simplifies the product into a sum. Hence, we need to take the logarithm likelihood function for the ML estimator and find its argmax.

İpek Öztaş
22003250
CS464-Sec2

$$P(D|\,\mu,\sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)\prod_{i=1}^{N} e^{\frac{-(x_i-\mu)}{2\sigma^2}}$$

Log-likelihood of data:

$$\ln P(D|\,\mu,\sigma) = \ln\left(\left(\frac{1}{\sigma\sqrt{2\pi}}\right)\prod_{i=1}^{N} e^{\frac{-(x_i-\mu)}{2\sigma^2}}\right)$$

$$\ln P(D|\,\mu,\sigma) = -N\ln\left(\sigma\sqrt{2\pi}\right) - \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial}{\partial\mu}\ln P(D|\,\mu,\sigma) = \frac{\partial}{\partial\mu}\left[-N\ln\left(\sigma\sqrt{2\pi}\right) - \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

$$\frac{\partial}{\partial\mu}\ln P(D|\,\mu,\sigma) = \frac{\partial}{\partial\mu}\left[-N\ln\left(\sigma\sqrt{2\pi}\right)\right] - \left[\sum_{i=1}^{N}\frac{dy}{d\mu}\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

$$= \sum_{i=1}^{N}\frac{(x_i-\mu)}{\sigma^2} = 0$$

$$= \sum_{i=1}^{N}(x_i - N\mu) = 0$$

$$\hat{\mu}MLE = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

This equals to the sample mean.

2.2

For the MAP estimate, we use Bayes rule in order to find the probability of P $(\mu|\,x_1,\ldots,x_n)$, then apply the algorithm function and take its derivative to find the MAP estimate for μ.

$$\frac{\partial}{\partial\mu}\ln\frac{P(\mu|x_1,\ldots,x_n)*\mathrm{P}(\mu)}{P(x_1,\ldots,x_n)} = \frac{\partial}{\partial\mu}\left[\ln\left(P(\mu|x_1,\ldots,x_n)\right) + \ln\mathrm{P}(\mu) - ln\,P(x_1,\ldots,x_n)\right]$$

We know the first end the last term from the previous question, so we plug them in and find the maximum argument.

$$= \sum_{i=1}^{N}\frac{(x_i-\mu)}{\sigma^2} + \frac{\partial}{\partial\mu}\ln P(\mu) = \sum_{i=1}^{N}\frac{(x_i-\mu)}{\sigma^2} + \frac{\partial}{\partial\mu}\ln\lambda e^{-\lambda x} = 0$$

İpek Öztaş
22003250
CS464-Sec2

$$= \sum_{i=1}^{N} \frac{(x_i - \mu)}{\sigma^2} + \frac{\partial}{\partial \mu} \ln \mu - \frac{\partial}{\partial \mu} \lambda \mu = \sum_{i=1}^{N} \frac{(x_i - \mu)}{\sigma^2} - \lambda = 0$$

$$\hat{\mu} MAP = \frac{1}{N} \sum_{i=1}^{N} x_i - \frac{\lambda \sigma^2}{N} = 0$$

2.3

Inserting the given $\mu = 1$ and $\sigma = 1$ to calculate the probability that the new data point is equal to 1:

$$P(x_{n+1} = 1) = \prod_{i=n+1}^{n+1} \left(\frac{1}{\sigma\sqrt{2\pi}}\right) e^{\frac{-(x_i - \mu)^2}{2\sigma^2}}$$

Insert the given values:

$$x_{n+1} = \left(\frac{1}{1\sqrt{2\pi}}\right) e^{\frac{-(1-1)^2}{2}}$$

$$\frac{1}{1\sqrt{2\pi}} = 0.39894$$

Same procedure for $x_{n+1} = 2$

$$P(x_{n+1} = 2) = \prod_{i=n+1}^{n+1} \left(\frac{1}{\sigma\sqrt{2\pi}}\right) e^{\frac{-(x_i - \mu)^2}{2\sigma^2}}$$

Insert the given values:

$$x_{n+1} = \left(\frac{1}{1\sqrt{2\pi}}\right) e^{\frac{-(2-1)^2}{2}}$$

$$\frac{1}{1\sqrt{2\pi}} e^{-0.5} = 0.24197$$

İpek Öztaş
22003250
CS464-Sec2

BBC News Classification

3.1 Bag-of-Words Representation and Multinomial Naive Bayes Model

3.1.1

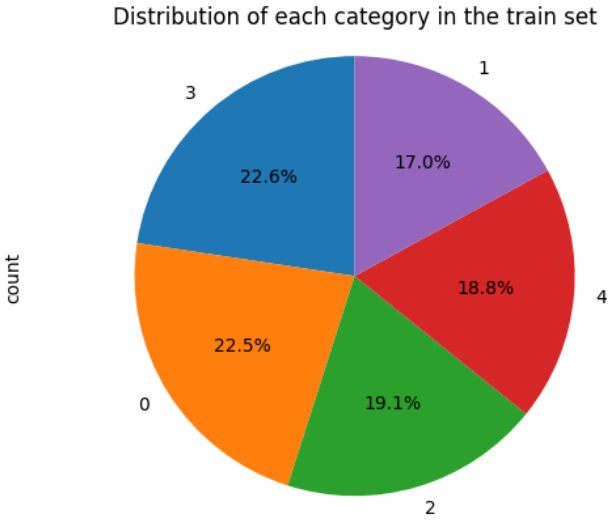

Figure 1 Percentages of each category in the train.csv

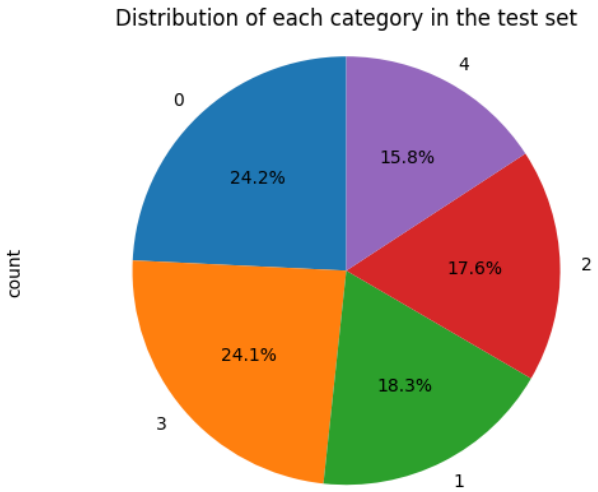

Figure 2 Percentages of each category in the test.csv

İpek Öztaş
22003250
CS464-Sec2

3.1.2

Priors:

$$\mathrm{P}(\mathrm{Y} = y_k) = \frac{N_k}{N}$$

According to the equation, the prior probability is equal to the number of documents in each category divided by all the given documents. This should be applied to the train set.

P(Y = Business) = 0.225

P(Y = Entertainment) = 0.17

P(Y = Politics) = 0.191

P(Y = Sport) = 0.226

P(Y = Tech) = 0.188

3.1.3

The dataset is not perfectly balanced but all the categories are close to 20% which can be considered as being reasonably balanced. None of the classes dominate the dataset significantly. The class 'Sport' has the highest prior probability (0.226), indicating that there are slightly more documents labeled as 'Sport' compared to other classes.

Imbalanced training set can affect the model in several ways. If one class dominates the training set, the model might become biased toward that class. The model may focus more on learning patterns from the majority class. Also, the model may not generalize well to minority classes because it has seen less examples of those classes during training. Accuracy might not be a suitable metric for the model's performance in an imbalanced dataset. A model that often predicts the majority class might achieve high accuracy but perform poorly on minority classes. Resampling and using various evaluation metrics can be considered.

3.1.4

The word ‘alien’ occurs 3 times and ‘thunder’ occurs 0 times in the training documents with the label ”Tech” (there are 313 Tech documents). Their log ratios are -4.64759 for the word *alien* and –inf for the word *thunder* respectively. The second log ratio is negative infinity because the nominator is equal to zero.

$$\ln(\mathrm{P}(\mathrm{alien} \mid \mathrm{Y} = \mathrm{Tech})) = -4.64759$$

$$\ln(\mathrm{P}(\mathrm{thunder} \mid \mathrm{Y} = \mathrm{Tech})) = -inf$$

İpek Öztaş
22003250
CS464-Sec2

3.2 Multinomial Naïve Bayes Model

There are 375, 284, 319, 377 and 313 news for each category respectively. We have 5 classes and 9635 different features. First, we calculate the prior probabilities for each class by dividing the number of news in each class to the total number of news. We take the logarithm function for the calculations. Then, we calculate the estimator for the probability that a particular word in class yk will be the j-th word of the vocabulary by dividing the number of occurrences of the word j in news with class yk in the training set including the multiple occurrences of the word by the total number of words in the class yk.

The testing accuracy is 0.242 which can be considered bad because the majority of the predictions is wrong. This is due to the lack of smoothing which prevents zero probabilities.

Predicted

Table 1 Confusion Matrix for Multinomial Naïve Bayes Classifier

Actual					
	135	0	0	0	0
	102	0	0	0	0
	98	0	0	0	0
	134	0	0	0	0
	88	0	0	0	0

3.3 Multinomial Naïve Bayes Model with Dirichlet prior

In this part, I extend my classifier so that it can compute an estimate of θ parameters using a fair Dirichlet prior. The accuracy is 0.977. In the first model, with no smoothing, any unseen word in a particular class would result in a probability of zero, leading to inaccurate predictions. The Dirichlet prior in the second model adds a small count to every possible word-class combination, ensuring that no probability becomes zero. Hence, even if a word wasn't observed in a specific class during training, it still gets a non-zero probability during testing.

Thus, the Dirichlet prior smoothens out the probabilities and prevents overfitting to the training data, leading to more accurate predictions. It can be observed from the confusion matrix below as well.

İpek Öztaş
22003250
CS464-Sec2

Predicted

Table 2 Confusion Matrix for Multinomial Naïve Bayes Classifier with ML estimator

Actual					
	131	0	2	0	2
	0	97	0	0	5
	1	0	96	0	1
	0	0	1	133	0
	1	0	0	0	87

3.4 Bag-of-Words Representation and Bernoulli Naive Bayes Model

In Multinomial NB, the representation is based on the frequency of words, while in Bernoulli NB, it's binary, focusing on presence or absence. In the Bernoulli model, each feature (word) is treated as a binary variable, indicating whether it's present (1) or absent (0) in the document. The accuracy is 0.966 which is a high accuracy, and the confusion matrix indicates that the model is performing well across different classes. Again, we employed Dirichlet prior with $\alpha = 1$ to handle zero occurrences during parameter estimation. Comparing these results with our previous Multinomial Naive Bayes (MNB) model, we observe that both models achieved high accuracies—0.977 for MNB and 0.966 for BNB. The slight difference in accuracy may be attributed to the distinct nature of the two models.

Predicted

Table 3 Confusion Matrix for Bernolli Naïve Bayes Classifier

Actual					
	132	0	2	0	1
	3	96	1	0	2
	4	0	94	0	0
	0	0	0	134	0
	4	2	0	0	82