CMPE 343: Introduction to Probability and Statistics for Computer Engineers

Ali Gür, Ahmet Baha Bayrakçıoğlu, Veysel Çokbakar

December 15, 2024

Task 1

First, we took the sentences in the sentences.txt file and divided them into words. Then, we used a dictionary structure to keep the frequency of the words following each word. Then we filled in the dictionary by going over the bigrams. We normalized the frequencies to calculate the probability distributions of subsequent words. When creating new sentences, we started with the "start" token, used the probability distributions we obtained, and performed "random sampling" from these distributions to predict the next word until we reached the "end" token.

1. Generate five sentences using the conditional probabilities that you have obtained (Each sentence needs to start with the word "`<|start|>`"). Are these sentences meaningful? Are they already in the `sentences.txt` or completely new?

   The sentences are not always meaningful since there are a lot of word combinations and the model chooses next word probabilistically i.e. randomly. There could be certain generated sentences which can be in the input file.

   - `<|start|> we visit a local museum <|end|> -> Not New`
   - `<|start|> we have a scenic train <|end|> -> New`
   - `<|start|> they sit by the park <|end|> -> New`
   - `<|start|> you go to make homemade pizza <|end|> -> New`
   - `<|start|> they listen to play in the history of music <|end|> -> New`

2. Suppose that we want to calculate the probability, $P(w_1, w_2, \ldots, w_k)$, of a sentence $w_1, w_2, \ldots, w_k$ with k words. Given our assumption that each word depends only on the previous word, what is the formula for $P(w_1, w_2, \ldots, w_k)$?

   $P(w1, w2, w3, ..., wk) = P(w1)*P(w2|w1)*P(w3|w2)*P(w4|w3)*......*P(wk|wk-1)$

3. Generate a sentence, $w_1, w_2, \ldots, w_k$ using your language model and calculate $P(w_1, w_2, \ldots, w_k)$. Then, think of a sentence $v_1, v_2, \ldots, v_k$ on your own and calculate $P(v_1, v_2, \ldots, v_k)$. Which one is larger, $P(w_1, w_2, \ldots, w_k)$ or $P(v_1, v_2, \ldots, v_k)$? Comment on the results (You can calculate $\log P(w_1, w_2, \ldots, w_k)$ instead of $P(w_1,$

$w_2, \ldots, w_k$) to avoid numerical underflow.)

Generally the sentences in the original file are more likely to be created. Sentences we created are mostly have less chance to be created, but sometimes it is possible that they can be highly possible.

- `<|start|> they go to knit <|end|> -> Generated -> LogProb:  6.90e-06`
- `<|start|> i learn to play the trumpet <|end|> -> In file -> LogProb: 5.08e-05`

Part 2

Detection Time

1. Which probability distribution would be appropriate to model the arrival time of a sandworm? What are the mean and the standard deviation of this distribution? Evaluate them for R_1.

   Exponential distribution with $\beta = 25$:

   - $\mu = 25$
   - $\sigma = 25$
   - $\sigma^2 = 625$

2. Conduct a hypothesis test to analyze whether the rhythmic movement R_2 results in an average detection time higher than $\mu_1 = 25$ or not. Use significance level $\alpha = 0.05$. Write out all the steps, including the hypothesis, test statistic, computations, and decision.

   We have one sample and we know the standard deviation of the distribution. Therefore, we will calculate Z-value. First we will determine the null hypothesis and alternative hypothesis, then we will calculate the sample mean. We will calculate Z-statistic using sample mean and hypothesized mu value, population standard deviation and sample size. Then we will find the critical-z value for the desired alpha value. Then we will compare z-statistic and z-critical and decide. When we apply these steps using the data we have, We fail to reject null hypothesis

Predicting Detections

To apply the naive bayes algorithm, we need to calculate the values of $P(\text{Detect} \mid a, d)$ and $P(\text{No Detect} \mid a, d)$. Whichever is greater will be our guess. We make this possibility simpler by using Bayes' theorem to calculate these values. To calculate these, we obtain the probabilities we need to find with the help of pandas. For example, we find $P(\text{Detect})$ by dividing the number of detected samples by the total number of samples. Likewise, we find $P(\text{No Detect})$ by dividing the number of no detect samples by the total number of samples. Then we divide the data into detect and no detect. Using this data and making maximum likelihood estimation using standard deviation and mean values, we find the values of $P(a \mid \text{Detect})$,$P(d \mid \text{Detect})$,$P(a \mid \text{No Detect})$, $P(d \mid \text{No Detect})$. We also find $P(a)$ and $P(d)$ using initial data. By substituting these into the formula, we calculate the values of $P(\text{Detect} \mid a, d)$ and $P(\text{No Detect} \mid a, d)$ and make our predictions.

1. Write out the formulas for $P(\text{Detect} \mid a, d)$ and $P(\text{No Detect} \mid a, d)$.

   Using Bayes Theorem and Statistical Independence

   $P(\text{Detect} \mid a, d) = \frac{P(a,d|\text{Detect})P(\text{Detect})}{P(a,d)} = \frac{P(a|\text{Detect})P(d|\text{Detect})P(\text{Detect})}{P(a)\ P(d)}$

   $P(\text{No Detect} \mid a, d) = \frac{P(a,d|\text{No Detect})P(\text{No Detect})}{P(a,d)} = \frac{P(a|\text{No Detect})P(d|\text{No Detect})P(\text{No Detect})}{P(a)\ P(d)}$

2. Make predictions for the 100 observations that you have obtained. How many of the 100 outcomes have you detected correctly?

   When we predicted the train dataset using the naive bayes algorithm, we got 90% accuracy. Therefore, we guessed 90 out of 100 correctly.

3. Coincidentally, another scientist has conducted the same experiment as you and sent their results to you so that you can test your prediction mechanism. Keeping your estimates of distribution parameters the same, make predictions for the new set of results that are available in detection_data_extra.csv. How many of the outcomes did you correctly predict? Did the number of correctly predicted outcomes change when compared to your correctly predicted outcomes for your original experiment results?

   When we predicted the test dataset with the model we trained, we got 81% accuracy. So we guessed 81 out of 100 correctly. It is normal that we get less accuracy from the train dataset because the model gives better results on the data it is trained on.

Part 3

1. Write the formula for $E[g(\mathbf{x})]$ but do not evaluate it.

   $$E[g(x)] = \int_{-\infty}^{\infty} g(x) f(x; \mu, \Sigma) \quad dx$$

   $$g(x) = 0.1x_1^2 + 12.5x_2^2 - 7.5x_3^2$$

   $f(x; \mu, \Sigma)$ is the probability density function of the multivariate normal distribution.

2. Estimate $E[g(\mathbf{x})]$ using Monte Carlo sampling with the following number of samples: $n_1 = 50$, $n_2 = 100$, $n_3 = 1000$, $n_4 = 10000$. For each experiment, calculate the 95% confidence interval.

We created samples of different sizes using numpy's relevant function (multivariate_normal) and the covariance matrix given to us. We obtained the function values we wanted to make a Monte Carlo approach by using samples. To create the Confidence Interval, we calculated the mean of these values and calculated the standard error using the standard deviation of these values and the sample size. We found the t-values required to obtain the desired confidence level with the help of t-Table and calculated the ranges.

Sample Size: 50, Mean: 45.349, 95% CI: (41.041, 49.657)
Sample Size: 100, Mean: 43.898, 95% CI: (41.077, 46.719)
Sample Size: 1000, Mean: 44.619, 95% CI: (43.536, 45.703)
Sample Size: 10000, Mean: 43.792, 95% CI: (43.468, 44.115)

3. Estimate $E[g(\mathbf{x})]$ using Monte Carlo sampling with $n_0 = 10000$ samples and $n_1 = 50$ samples and denote your estimates as g_0 and g_1, respectively. Given that you are confident with your estimate g_0, test your estimate g_1 on whether $g_0 = g_1$ or not. Use significance level $\alpha = 0.05$. Write out all the steps, including the hypothesis, test statistic, computations, and decision.

First, we create null and alternative hypotheses. Then, to test our hypothesis, we used numpy to obtain samples of size 1000 and 50. We have two samples from the same distribution, so the standard deviation of the populations of the two samples is equal. However, we do not know the standard deviation of this distribution. Therefore, we need to apply pooled t-test. For this, we calculate the variances of two samples. We calculate pooled standard deviation using this. Then we calculate the means of the samples and calculate t-statistic using them. We then use t-table to find the critical-t value at the desired cardinality. Then we compare t-statistic and t-critical and decide. When we apply this procedure, we fail to reject null hypothesis.

`--> Link to colab notebook.`

WARNING1! You should upload the related files to the colab session. Otherwise you will get file not found error.
WARNING2! The colab notebook will want permission to be able to view the codes. You can request permission.