Assignment 2

● Graded

Group

PRAKHAR SHRIVASTAV
ARSHIT
HARSH AGRAWAL
...and 3 more

View or edit group

Total Points

37 / 40 pts

Question 1

Calculations

10 / 10 pts

✓ **+ 10 pts** Description of how the model was constructed

− 2 pts Insufficient or missing details

+ 0 pts Completely wrong or else unanswered.

Question 2

Code

27 / 30 pts

+ 0 pts No rubric -- see comments for mark breakup

+ 27 pts GROUP NO: 5

Grading scheme for code:
Train time tt (in sec): tt < 0.01 (10 marks), 0.01 <= tt < 0.1 (9 marks), 0.1 <= tt < 1 (8 marks), tt >= 1 (7 marks)
Model size ms (in KB): ms < 100KB (5 marks), 100 <= ms < 500 KB (4 marks), ms >= 500 KB (3 marks)
Evaluation time et (in sec): et < 0.01 (15 marks), 0.01 <= et < 0.1 (14 marks), 0.1 <= et < 1.0 (13 marks), et >= 1.0 (12 marks)
Marks multiplier mm: (1 + prec) / 2
Final marks: ceil(mm * (tt marks + ms marks + et marks))

tt = 0.006 sec : 10 marks
ms = 120.889 KB : 4 marks
et = 0.196 sec : 13 marks
Precision: 0.993 i.e. Multiplier (mm): 0.996
TOTAL: 27 marks

No questions assigned to the following page.

Assignment 2 Solution

Name of Group: **Data Geeks**

Group Members:

Arshit	220209
Harsh Agrawal	220425
Kritnandan	220550
Kundan Kumar	220568
Prakhar	220786
Rudradeep Datta	220919

Indian Institute of Technology Kanpur

CS771 Introduction to Machine Learning

Instructor: Prof. Purushottam Kar

July 17, 2024

No questions assigned to the following page.

1 Algorithm:

We use the Bayesian approach to predict the word from the dictionary based on the highest posterior probability.

Example:

In this example, we take the dictionary words and bigram list and calculate the posterior probability, which has the highest posterior probability the predicting the words.

Given Data

- **Dictionary Words:** "apple", "chappal"
- **Bigram List:** ["ap", "pp", "le"]

Frequency Calculation for Training Data

For "apple":

- **Bigrams:** ["ap", "pp", "pl", "le"]
- **Bigram Frequencies:**
  - "ap": 1
  - "pp": 1
  - "pl": 1
  - "le": 1
- **Total Bigrams:** 4

For "chappal":

- **Bigrams:** ["ch", "ha", "ap", "pp", "pa", "al"]
- **Bigram Frequencies:**
  - "ch": 1
  - "ha": 1
  - "ap": 1
  - "pp": 1
  - "pa": 1
  - "al": 1
- **Total Bigrams:** 6

Calculate Bigram Probabilities

For "apple": P("ap" | $”apple”) = \frac{Count(”ap”)}{TotalBigrams} = \frac{1}{4} = 0.25$
$P(”pp” \mid ”apple”) = \frac{Count(”pp”)}{TotalBigrams} = \frac{1}{4} = 0.25$
$P(”le” \mid ”apple”) = \frac{Count(”le”)}{TotalBigrams} = \frac{1}{4} = 0.25$

For "chappal": P("ap" | $”chappal”) = \frac{Count(”ap”)}{TotalBigrams} = \frac{1}{6} \approx 0.167$
$P(”pp” \mid ”chappal”) = \frac{Count(”pp”)}{TotalBigrams} = \frac{1}{6} \approx 0.167$
$P(”le” \mid ”chappal”) = 0 \quad (since”le”isnotinthebigramsof”chappal”)$

No questions assigned to the following page.

Compute Posterior Probabilities for Each Word

For "apple": Posterior Probability = P("ap" | $”apple”) \times P(”pp” \mid ”apple”) \times P(”le” \mid ”apple”)$
$= 0.25 \times 0.25 \times 0.25$
$= 0.015625$

For "chappal": Posterior Probability = P("ap" | $”chappal”) \times P(”pp” \mid ”chappal”) \times P(”le” \mid$
$”chappal”)$
$= 0.167 \times 0.167 \times 0$
$= 0$

Conclusion

- **Probability for "apple":** 0.015625
- **Probability for "chappal":** 0

Bigram Probability Map Preprocessing

Preprocessing the 2D Map

To efficiently handle the prediction phase, we preprocess the 2D map from each word in the dictionary to all its bigrams, calculating the probability of each bigram being present in the word. This map is stored for quick lookup during prediction.

Bigram Probability Calculation

For each word in the dictionary, we extract the bigrams and compute their respective probabilities. These probabilities are derived from the frequency counts of the bigrams within each word, as calculated in the previous section. The steps are as follows:

1. **Extract Bigrams:** For each word, we extract all possible bigrams.
2. **Count Frequencies:** Count the occurrences of each bigram within the word.
3. **Compute Probabilities:** Calculate the probability of each bigram by dividing its frequency by the total number of bigrams in the word.

Storing the Probabilities

We store the computed bigram probabilities in a structured format for efficient retrieval during the prediction phase. The map is structured as follows:

- **Key:** Word from the dictionary.
- **Value:** Dictionary of bigram probabilities.

Example Map Structure:

```
{
    "apple": {"ap": 0.25, "pp": 0.25, "pl": 0.25, "le": 0.25},
    "chappal": {"ch": 0.167, "ha": 0.167, "ap": 0.167, "pp": 0.167, "pa": 0.167, "al": 0.167}
}
```

This preprocessing step ensures that we can quickly access the bigram probabilities for any word in the dictionary during the prediction phase, optimizing the performance of our Bayesian model.

Posterior Probability Calculation for Prediction

Prediction Phase

When presented with a list of bigrams, we iterate through the entire dictionary and calculate the posterior probability for each word. This is achieved using the precomputed bigram probabilities

No questions assigned to the following page.

from the previous section. The words are then ranked based on their posterior probabilities, and the words with the highest probabilities are returned as predictions.

Procedure

1. **Extract Bigrams:** Extract the bigrams from the given list.
2. **Initialize Posterior Probabilities:** Set the initial posterior probability for each word as its prior probability.
3. **Iterate Through Dictionary:** For each word in the dictionary:
   (a) Calculate the posterior probability by multiplying the prior probability with the probabilities of the given bigrams.
   (b) If a bigram is not present in the word, the posterior probability is set to zero.
4. **Rank Words:** Rank the words based on their posterior probabilities.
5. **Return Predictions:** Return the words with the highest posterior probabilities.

Conclusion

- **Probability for "apple":** $P("apple") \times 0.015625$
- **Probability for "chappal":** 0

Predicted Words: Based on the posterior probabilities, the word "apple" has a non-zero probability, while "chappal" has a zero probability. Therefore, the predicted word list will contain "apple".

2 Solution for Part 2:

2.1 Results Obtained on Training Data

- **Precision :** 97.5%
- **Training time:** 0.053 seconds
- **Testing time:** 6.63 seconds
- **Model size:** 643705