B555 - Machine Learning - Fall 2024

Programming Assignment 1 Report

Mithileshan Muralidharan
Fall 24 MSCS Student

Table Of Contents

Task 1: Model Training, Prediction, & Evaluation

Objective:

In this task we use Maximum Likelihood Estimation, Maximum A Posteriori with a Dirichlet prior and Predictive Distribution to estimate the unigram model and the goal is to evaluate the effectiveness of each method by calculating perplexity on a test set.

Approach:

We are using training data and a test data file, each containing 640,000 words and constructing a vocabulary from both training and test sets, containing 9,999 distinct words. We calculate perplexity for different training set sizes N/128, N/64, N/16, N/4, N and compare them.

Steps:

1. Vocabulary Construction:
    - create a vocabulary of 9,999 words by combine the distinct words of training and test datasets.
2. Perplexity Calculation:
    - Perplexity is calculated to measure how well a predicts a sample of data and it is calculated using the formula:

$$PP = p(w_1, w_2, \cdots, w_N|\text{model})^{-\frac{1}{N}} \overset{\text{unigram}}{=} \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\ln p(w_i)\right)$$

3. Maximum Likelihood Estimation (MLE):
    - Calculating word probabilities based on their frequency in the training set using the formula

$$\text{Prediction using ML estimate: } p(\text{next word} = k\text{-th word of vocabulary}) = \frac{m_k}{N}$$

    - Where, mk is count of each word

      N is the total number of words in the training set.

4. MAP Estimation:
    - Calculating MAP Estimation using Dirichlet prior with alpha=2 using the formula:

$$\text{Prediction using MAP estimate: } p(\text{next word} = k\text{-th word of vocabulary}) = \frac{m_k+\alpha_k-1}{N+\alpha_0-K}$$

    - Where, mk is count of each word

      K is the Vocabulary size

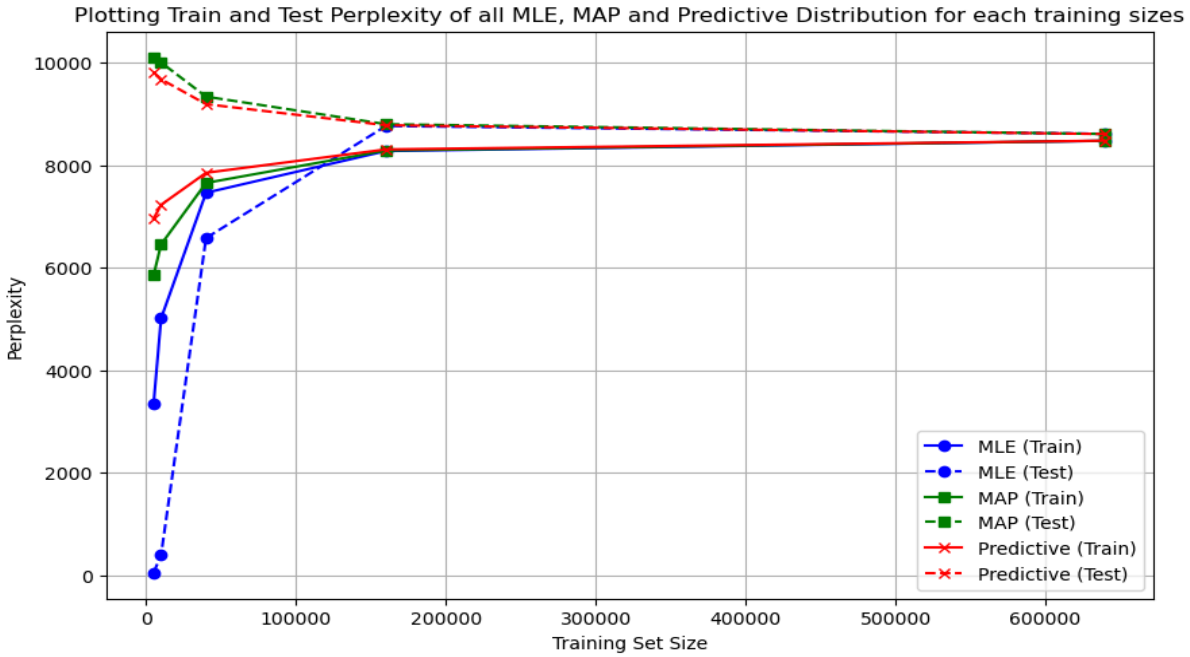
      N is the total number of words in the training set.

5. Predictive Distribution:
    - Calculating the Predictive Distribution using the formula:

$$\text{Prediction using predictive distribution: } p(\text{next word} = k\text{-th word of vocabulary}) = \frac{m_k+\alpha_k}{N+\alpha_0}$$

6. Calculate Perplexities:
    - Calculate both test and train Perplexities of all three methods for all training sizes and plot them.

Observations:



The plot compares the Train and Test Perplexities of three different models Maximum Likelihood Estimate (MLE), Maximum A Posteriori (MAP), and Predictive Distribution for different training set size.

Conclusions:

1.What happens to the test set perplexities of the different methods with respect to each other as the training set size increases? Please explain why this occurs.

Ans: As the training set size increases, the test set perplexities for all methods converge to similar values. Initially, there is a difference between MLE and the other methods for smaller training sizes, but as the training set grows, the difference reduces.
For small training sizes, the MLE performs well by providing less test perplexity values and when the training size expands it is not able to generalise and overfits.
On the other hand MAP and Predictive Distribution perform poorly with small datasets, because they use a prior distribution to adjust word probabilities. This leads to underfitting when there's not much data. But as the training set grows, both MAP and PD improve and become better at generalising.

2.What is the obvious shortcoming of the maximum likelihood estimate for a unigram model? How do the other two approaches address this issue?

Ans: The main shortcoming of MLE for a unigram model is that it assigns zero probability to words that don't appear in the training set because it only looks at the words it has seen.
However MAP and PD introduces a prior distribution over the word probabilities, which helps to avoid assigning zero probabilities to unseen words. This makes them less likely to overfit.

3.For the full training set, how sensitive do you think the test set perplexity will be to small changes in α' ? why?

Ans: For the full training set, the test set perplexity won’t be very sensitive to small changes in α', this is because as the training set gets bigger the word frequencies from the data play a bigger role in determining the perplexity, so small changes in α' have less effect on the model or its perplexity.

Task 2: Model Selection

Objective:

In this task we are asked to compute the log evidence and test set perplexity (use the predictive distribution), for a training set of size N/128 for each α ′(considering α ′ = 1.0, 2.0, . . . , 10.0).

Approach:

Calculate the log evidence function and test perplexity(using PD) for each α′ for training set size of N/128 and plot them.

Steps:

1. Log evidence function:

   Calculate the log evidence function for a training set of size N/128 for each α ′ using the formula:

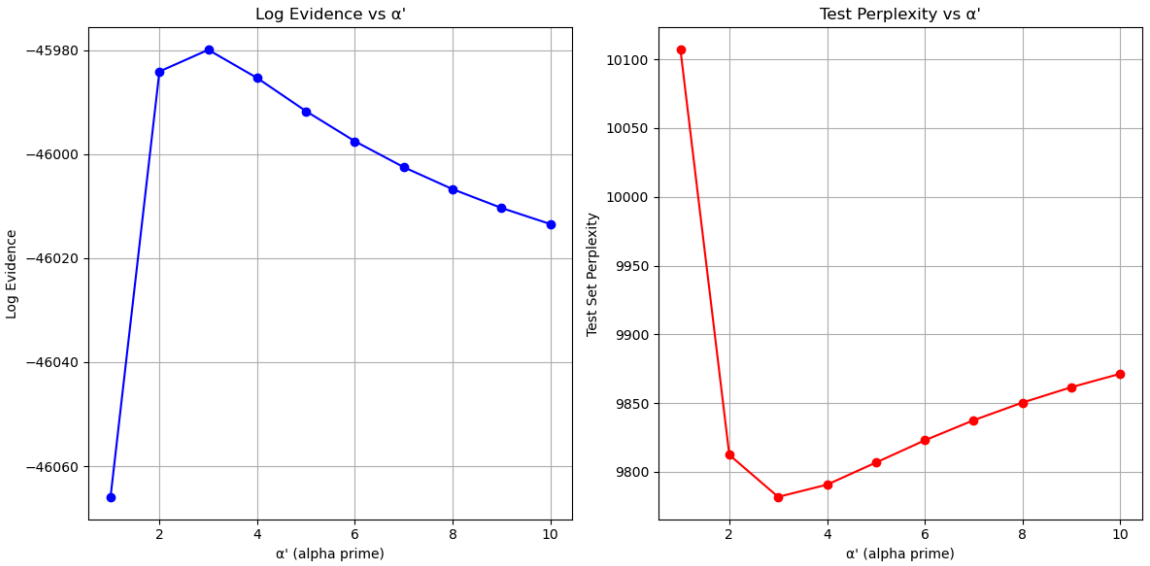
   $$\text{Evidence: } \Pr(\text{Data}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)\prod_{k=1}^{K}\Gamma(\alpha_k+m_k)}{\Gamma(\alpha_0+N)\prod_{k=1}^{K}\Gamma(\alpha_k)}$$

2. Perplexity:

   Calculate the test set perplexity using PD for a training set of size N/128 for each α ′.

3. Plot the log evidence and test set perplexity as a function of α ′.

Observation:



The plots show a clear relationship between α′ and both log evidence and test perplexity. At α′=1 the model has higher perplexity , indicating that the model is struggling to generalize. As α′ increases to 2, the perplexity drops significantly, which implies that the model fits the data well at this point. As α′ increases beyond 2, the perplexity rises again, suggesting that the model is again struggling to generalize, making it less effective at predicting unseen data.

Conclusion:

1.Is maximizing the evidence function a good method for model selection on this dataset?

Ans: The model provides the lowest perplexity for alpha value 3, meaning the model's performance improves for alpha value 3. However, if we keep increasing alpha value, the perplexity starts to rise again. This shows that picking the right value for alpha is important for getting the best performance.

Task 3: Author Identification

Objective:

In this task we are asked to determine whether a unigram model, trained on the text of one author, can differentiate between texts from the same author and from a different author based on word usage patterns.

Approach:

Train the model on pg345.txt.clean and find the perplexity of the other two files using this model.

Step:

1. Upload the files
2. Train the model on pg345.txt.clean using PD
3. Calculate the perplexity of the other 2 files using this model
4. Compare the perplexities

Observation:

```
Perplexity on pg1188.txt.clean (same author): 5864.256928400647
Perplexity on pg84.txt.clean (different author): 8270.556453793237

The perplexity of the file written by the same author is lesser than
the other.
```

Reference

1. Matplotlib 3.9.2 documentation
   “reference:https://matplotlib.org/stable/index.html”
2. NumPy Documentation “reference :https://numpy.org/devdocs/user/”
3. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer (India) Private Limited, 2013.