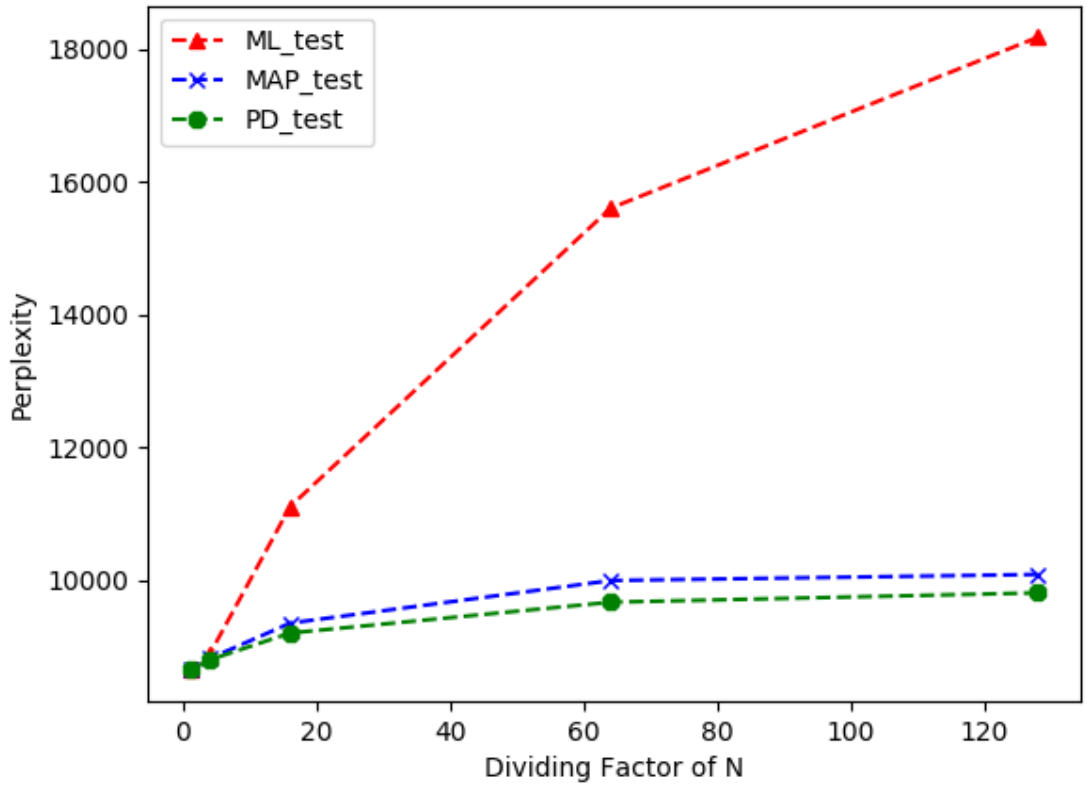
TASK 1:

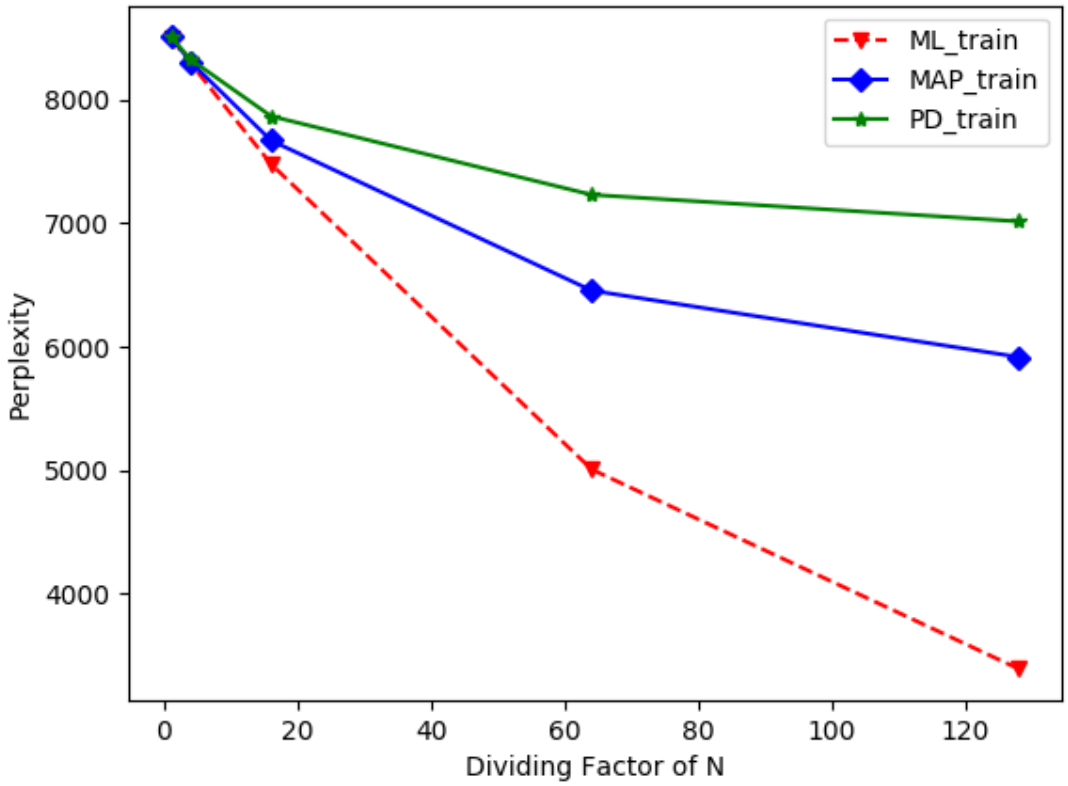
Let's look at the plot of Test Set Perplexities for all the three approaches. On the X-axis is the dividing factor for N = {1, 4, 16, 64, 128} and on the Y-axis is the related perplexity of the model trained on the training data divided by the dividing factor.



As we can make out from the plot, the perplexity for all the models increases as the size of training data decreases. This intuitively makes sense as there are less words in the training data now and thus the calculated probabilities of each word would be less than it would have been for a training dataset of larger size.

Moreover, we can see that the predictive distribution outperforms all the other models. This is because it gives us an estimate of the new data after considering all the possible uncertainty in the unknown parameter. It thus has a lower variability by underestimating the variability of all the new data points. Thus, as the size of our training dataset decreases, predictive distribution outperforms all the other models more and more.

Let's look at the plot of Test Set Perplexities for all the three approaches. On the X-axis is the dividing factor for N = {1, 4, 16, 64, 128} and on the Y-axis is the related perplexity of the model trained on the training data divided by the dividing factor



As we can make out from the graph the perplexities for all the three approaches decreases as the size of training data set decreases. This intuitively makes sense as a larger training dataset would have more words, thus a higher probability is associated with the words in the model. However, we can notice that MLE tends to perform the best as the size of data decreases. This is because the other approaches apply smoothing to all the words that don't occur in the training data set and as the size of the training data set decreases the smoothing being applied tends to overfit the model constructed using the training data. It associates a higher probability to a word occurring than would have been associated had the training dataset been bigger.

An obvious shortcoming of the Maximum Likelihood estimation is the case of missing words. If a word doesn't occur in the training set, the probability of that word occurring becomes zero instantly. Thus, when we now calculate the perplexity of test data, we get a value of infinity if that word appears even once in the test data. This scenario has been handled in my program by assigning a very small value, almost equivalent to zero, for the words that don't occur in the training data but are present in the test set. This is a naïve approach and is overcome by the other two models. This is done by applying smoothing to the probabilities of all words by assuming a prior for each word. This is done by using parameter alpha_k for each word (alpha_0 is the sum of all alpha_k). As we can make out in the formulae:

Prediction using MAP: P(next word = kth word of vocab) = (m_k + alpha_k -1) / (N + alpha_0 – K)

Prediction using PD: P(next word = kth word of vocab) = (m_k + alpha_k) / (N + alpha_0)

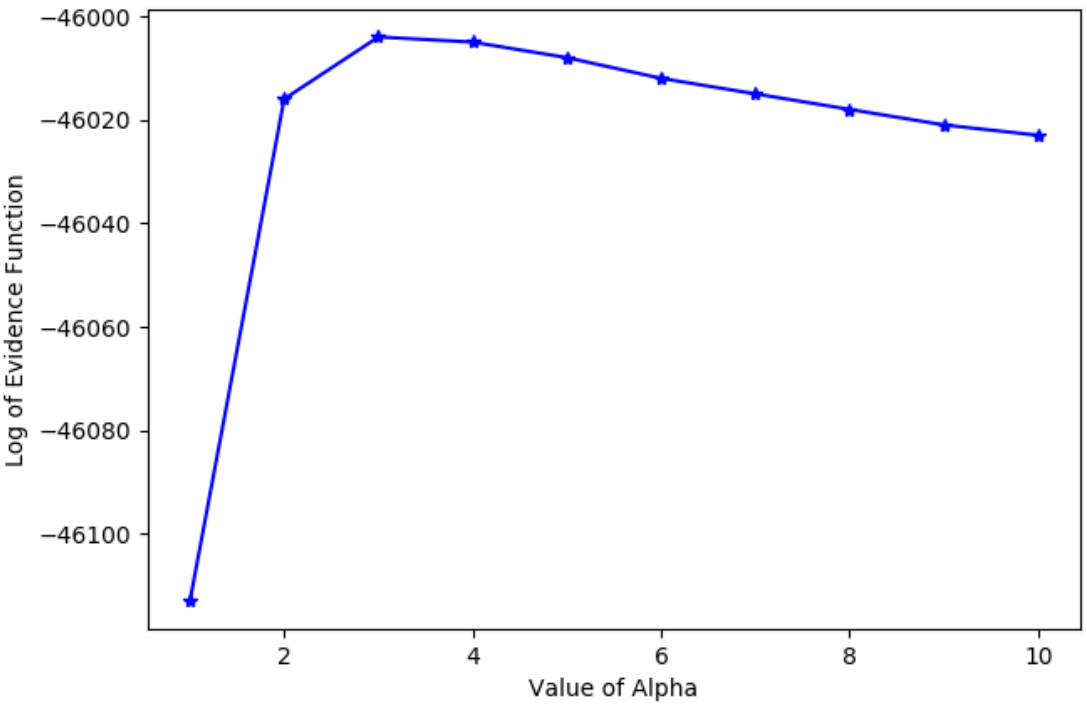
Thus, the probability of a word that doesn't occur in the training dataset but is present in the testing data is not assigned a probability of zero. Rather it gets assigned a probability depending on the alphas we assumed for each word. In this manner the other two models never overfit the data and consistently give better results than MLE.

With the change in value of alpha, the value of perplexity changes only for the models MAP and PD obviously. As we decrease the value of alpha slightly, we observe a lot of changes in the perplexities calculated on test data. The perplexity keeps increasing and hits infinity for alpha = 1. This is because for alpha=1, the models transform into MLE model with P(next word = kth word of vocab) = m_k/N.

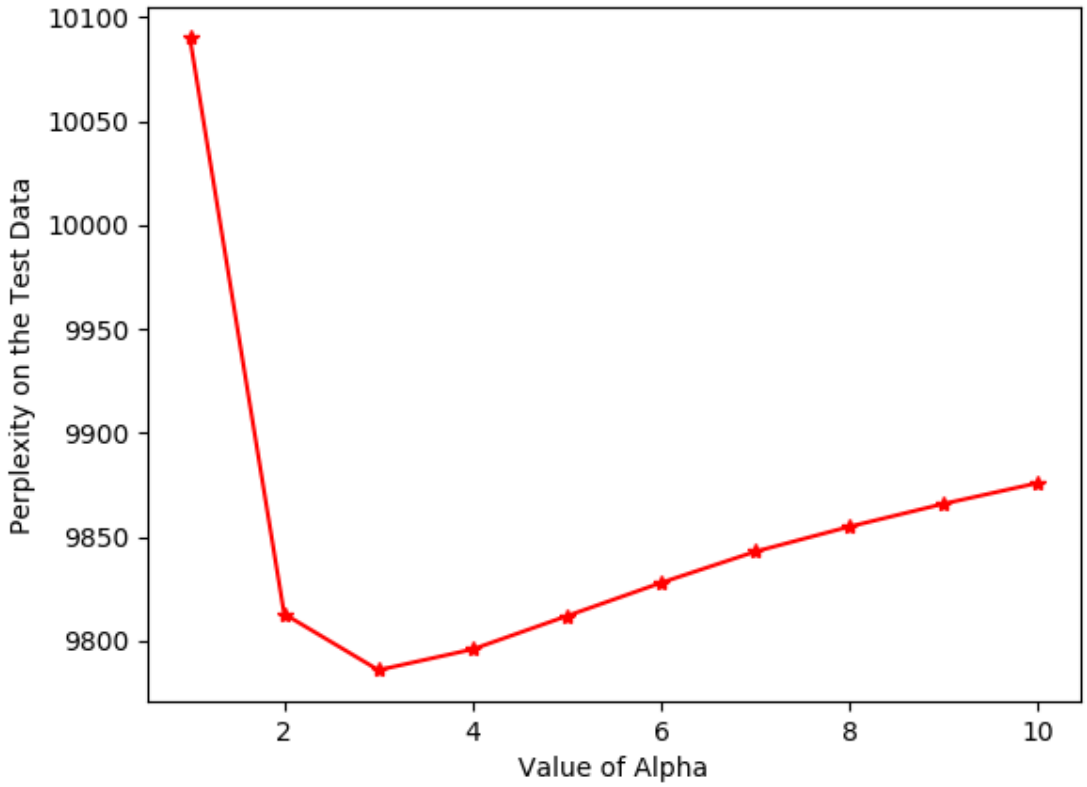
And the perplexity would become higher as we go nearer MLE. However, when we increase the value of alpha, there is a slight change. We notice a slight decrease in the perplexities values that reach their lowest and then again gradually keep increasing. This is shown in the plot in question 2 but only for PD.

TASK 2:

The plot of Log of Evidence Function versus the Alpha values is shown below:



The plot of Perplexity Values on Test Data versus the Alpha values is shown below:



As we can see from the graphs, the maximum value of log evidence function is at alpha = 3. Moreover, the least value for the perplexity also is at alpha = 3. Thus we can infer that maximizing the Log of the evidence function is a good way of selecting the right parameters for our model and thus model selection for this dataset.

TASK 3:

As mentioned, PD with alpha=2 has been used to detect the authors. The model is trained on pg121 and tested for pg141, pg1400

Perplexity of PD on pg141.txt.clean = 4780

Perplexity of PD on pg1400.txt.clean = 6388

As we can notice from the perplexity's values, perplexity for pg141 is lesser than the perplexity for pg1400. Thus, we can say that pg141 is more likely to be written by author of pg121. Thus the unigram model is successful in classification task for authorship.