Programming Project 1

This assignment is due by Tuesday September 29, 4:30pm - in hardcopy (in class) and electronically (via provide).

Overview: Unigram Model

Recall the unigram model from question 2 of assignment 1. A unigram model over a vocabulary of $K$ words is specified by a discrete distribution with parameter $\mu$. Under this model, the probability of the $k$-th word of the vocabulary appearing is given by $\mu_k$. In this experiment we will use the unigram model to evaluate different learning methods, and to perform model selection.

Part 1: Model Training & Prediction

In class we developed three methods for prediction: using the maximum likelihood estimate, using the MAP estimate, and using the predictive distribution. In this part, we evaluate the effectiveness of the three different methods. More specifically, we will use text training data to perform unigram model learning according to each method and then calculate the perplexity of the learned models on a held-out test data set. Perplexity is a standard effectiveness metric in probabilistic language modeling for scoring how well a model predicts a given collection of words (low perplexity values imply good performance). Perplexity is defined by

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

On the course webpage, you will find two files training_data.txt and test_data.txt each of size $N = 640,000$ words. We have pre-processed and “cleaned” the text so it does not require any further manipulation and you only have to read the space-separated strings from the corresponding ASCII text files.

For each size of training set in $[\frac{N}{128}, \frac{N}{64}, \frac{N}{16}, \frac{N}{4}, N]$, you should train a unigram model according to the three different methods discussed above (use the initial segment of the full training data). Normally, you would determine the size of the vocabulary from the training set; but, for Parts 1 & 2, you should fix the vocabulary to the unique strings present in the full training set (the size of the vocabulary is $K = 10,000$ words). Use a Dirichlet distribution with parameter $\alpha = \alpha' 1$ as a prior (this is a scalar $\alpha'$ multiplied by a vector of ones). Set $\alpha' = 2$ for this part. Please report
the perplexities on the test set under the trained models. Keep in mind that model training should always occur with training data only.

In your discussion of the results, please address the following:

- 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.

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

- For the full training set, how sensitive do you think the test set perplexity will be to small changes in $\alpha'$?

Part 2: Model Selection

In the previous part, we set the value of the hyperparameter, $\alpha'$, manually. In this part, you will use the evidence function (question 2 assignment 1) and training data to select a value of $\alpha'$. In general, direct maximization of the evidence function to determine the hyperparameter can be difficult. Here we only have one hyperparameter and can therefore use a “brute-force” grid search to select $\alpha'$.

In particular, compute the log evidence at $\alpha' = 1.0, 1.5, 2.0, \ldots, 9.5, 10.0$ for a training set of size $\frac{N}{128}$. Also, compute the perplexities on the test set (use the predictive distribution) at these same values. Plot the log evidence and test set perplexity as a function of $\alpha'$ and discuss what you see. In your discussion of the results, please address the following:

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

Part 3: Author Identification

Can the unigram model identify authors? In this part, we apply the model to this problem. On the course web page you will find 3 additional files for this assignment. Each of them is a cleaned text version of a classic novel (thanks to the Gutenberg project). Train the model on pg121.txt.clean (use the predictive distribution with $\alpha' = 2$, and vocabulary size $K = 18,261$) and evaluate the perplexity on each of the other two texts. Please report the resulting perplexities. One of the test files is by the same author as the training file but the other is not. Was the model successful in this classification task?

Useful Equations

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

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

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

• In above, \(m_j\) = No. of times \(k\)-th word of vocabulary appears in document, \(N\) = Total no. of words in document, and \(\alpha_0 = \sum_{k=1}^{K} \alpha_k\)

Submission

• You should submit the following items both electronically and in hardcopy:
  (1) All your code for the assignment. Remember to write clear code and document it as needed. Also, please supply a short README file describing how to run your code.
  (2) A short report on the experiments, their results, and your conclusions.

• For electronic submission, put all the files into a zip or tar archive, for example myfile.zip (you do not need to submit the data we give you). Then submit using provide comp136 pp1 myfile.zip.

• Your assignment will be graded based on the clarity and correctness of the code, and presentation and discussion of the results.