CS769 Advanced NLP Homework 3

Due 2/24/2010 before class

1 Language Modeling with the CMU-Cambridge Toolkit

Download the CMU-Cambridge LM toolkit from `http://www.speech.cs.cmu.edu/SLM/toolkit.html`. Follow the documentation to `make install` it (check endian please). This should produce a set of executables in 'bin/'.

Download the training corpus from `http://pages.cs.wisc.edu/~jerryzhu/cs769/dataset/polarity-dataset-v2.0/training.text`. These are 1000 movie review articles. You will notice they have one sentence per line, and there is a sentence-beginning token `<s>` in front of each sentence. Do not further process the corpus. We will train language models on this corpus. Please follow the ML toolkit documentations in the following steps.

> **Question 1.** [**5**] Using `text2wfreq` and `wfreq2vocab`, create a vocabulary for words that *appear more than 4 times* (i.e., a count of at least 5). How many word types are there in your vocabulary?

With the above vocabulary, use `text2idngram` to collect unigram (specify the flag `-n 1`) counts from training.text. Create a `context cue` file `movie.ccs` with a single line `<s>` in it—We tell the program that this is the special sentence-beginning symbol. Use `idngram2lm` to create a *unigram* LM (use `-binary`, `-context` and `-n 1` flags). Save this unigram LM for later use.

> **Question 2.** [**5**] Using `evallm`, and the interactive command `perplexity`, compute the perplexity of the unigram LM on *test.text* (download from the same address above). What is the perplexity on test.text? What is the perplexity on the training corpus itself (training.text)?

> **Question 3.** [**5**] Repeat from `text2idngram`, but this time collect and build a *bigram* LM. What is the perplexity of the bigram LM on test.text and training.text?

> **Question 4.** [**5**] Collect and build a *trigram* LM. What is the perplexity of the trigram LM on test.text and training.text?

> **Question 5.** [**5**] Discuss the difference between test and training perplexity, as you move to more complicated LMs. Why training corpus perplexity is not a reliable measure of LM quality?

Now make a copy of your vocabulary file. Edit the copy:

- The first 4 lines starting with `##` are comments. Remove them so that the file has one word type per line.
- Remove `<s>` from the copy.

Run `evallm` again with the *unigram LM*. Run `perplexity` on the copy, this time with a `-probs vocab.probs` flag. The file *vocab.probs* contains the unigram probabilities of each word type, in the order specified in the copy.

> **Question 6.** [**5**] Find the unigram probability of the following words in vocab.probs:
>
> - the
> - movie
> - mulan
> - album

2 Add-1 Smoothing as MAP Estimate [15]

Prove that add-ϵ smoothing is the MAP estimate, with a Dirichlet prior with hyperparameters ϵ+1. Hint: formulate the problem as constrained optimization, and apply Lagrange multiplier.

3 Language Identification with Naive Bayes

Implement a letter-based Naive Bayes classifier that classifies a document as English, Spanish, or Japanese - all written with the 26 lower case letters and space.

Download the dataset from `http://pages.cs.wisc.edu/~jerryzhu/cs769/dataset/languageID.tgz`. This dataset consists of 60 documents in English, Spanish and Japanese. The correct class label is the first letter of the filename.

We will be using a character-based Nave Bayes model. You need to view each document as a stream of characters, including space. We have made sure that there are only 27 different types of characters (a to z, and space).

You must compute and store the prior probabilities, P(English), P(Spanish) and P(Japanese), as well as the conditional probabilities, P(c|English), P(c|Spanish), and P(c|Japanese), from the training set (see below). Use add-1 smoothing for all of these. Store all probabilities as logs to avoid underflow. This also means you need to do arithmetic in log-space.

- **Question 1.** [**10**] Use files 0.txt to 9.txt in each language as the training data, build a Naive Bayes classifier for the three languages. Print P(English), P(Spanish) and P(Japanese), as well as the conditional probabilities P(c|English), P(c|Spanish), and P(c|Japanese) for all 27 characters.

- **Question 2.** **[5]** Classify e10.txt. List P(English|e10.txt), P(Spanish|e10.txt), and P(Japanese|e10.txt).
- **Question 3.** **[10]** Evaluate the performance of your classifier on the test set (files 10.txt to 19.txt in three languages). Present the performance using a confusion matrix. A confusion matrix summarizes the types of errors your classifier makes, as shown in the table below. The columns are the true language a document is in, and the rows are the classified outcome of that document. The cells are the number of test documents in that situation. For example, the cell with row = English and column = Spanish contains the number of test documents that are really Spanish, but misclassified as English by your classifier.

	English	Spanish	Japanese
English			
Spanish			
Japanese			

- **Question 4.** **[15]** Repeat Questions 1,2,3, but this time train your Naive Bayes classifer using the following data, which simulates the case that sometimes your training data can be of low quality:
  - English training files: e0.txt, e1.txt, s2.txt, j3.txt
  - Spanish training files: s0.txt, s1.txt, j2.txt, e3.txt
  - Japanese training files: j0.txt, j1.txt, e2.txt, s3.txt

4 Asymptotic Behavior of LMs

Question 1. [5] Consider a training corpus TRAIN and a test corpus TEST, both artificially generated from the same bigram language model LMg. You train separately a unigram language model LM1, and a bigram language model LM2, using TRAIN. Both LM1 and LM2 are maximum likelihood estimates (i.e., not smoothed). As the size of TRAIN and TEST approaches infinity, is LM1 or LM2 better on TEST? Briefly justify your answer using words.

Question 2. [5] Same as above, except that the underlying LMg is a unigram instead of a bigram.

Question 3. [5] Same as above where LMg is a unigram, except that the size of TRAIN is small (size of TEST still approaches infinity).