

Assignment 3

Teaching assistant: Wilson Tam (yct@cs.cmu.edu)

Due on: Nov 5th before class**Problem 1: LM estimation**

In this problem, we estimate a statistical bigram language model $\Lambda = \{P(v|u)\}$ using Maximum Likelihood Estimation (MLE) given a training corpus $W = \{w_1, w_2 \dots w_N\}$ with vocabulary size V . With the 1st-order Markov assumption, we know that $P(w_i | w_{i-1} w_{i-2} \dots w_1) = P(w_i | w_{i-1})$. Assume there are no OOVs (out of vocabulary), i.e. all the words in the test data are covered by the training corpus W .

1. Derive the log likelihood of the training data given the model Λ , i.e. $L(W; \Lambda) = \log P(w_1, w_2 \dots w_N; \Lambda)$.
2. By differentiation, show that the MLE of $\hat{\Lambda}$ is given by:

$$\hat{P}(v|u) = \frac{C(u, v)}{C(u)} \quad (1)$$

where $C(u, v)$ denotes the word bigram count of (u, v) and $C(u)$ denotes the word unigram count of u . [Hint: You need to integrate the constraints $\sum_v P(v|u) = 1 \forall u$ into $L(W)$.]

3. Show that a MLE bigram LM always performs better than a MLE unigram LM on the same training corpus W , i.e. prove that $\log P_{bg}(W) \geq \log P_{ug}(W)$ where bg and ug denotes the bigram and unigram LM respectively.
4. What is the major problem in Eqn 1?
5. One approach to fix the problem in Eqn 1 is interpolate a bigram LM with a unigram LM, i.e. $P_I(v|u) = \lambda \cdot P(v|u) + (1 - \lambda) \cdot P(v)$. The interpolated model has 3 components $\Lambda_I = \{P(v|u), P(v), \lambda\}$. The ML estimation is used to estimate $P(v)$ in a similar fashion using the same training corpus W . Draw the interpolated model as a one-state HMM specifying the transition probability and emission probability.
6. Using the above results, write down the ML estimate of the interpolation weight λ .
7. Why did the interpolation scheme not work in this case? [Hint: you may want to show $\hat{\lambda} = 1$ at the fixed point (i.e. at the training convergence).] Suggest a simple method to avoid it.

Problem 2: LM smoothing

1. The LM estimation using the Laplace (add-one) smoothing can be expressed as follows:

$$\hat{P}(v) = \frac{C(v) + 1}{\sum_v (C(v) + 1)} \quad (2)$$

where each unigram count is at least one. Show that this estimate can be written as a linear LM interpolation as follows:

$$\hat{P}(v) = \lambda \frac{C(v)}{\sum_v C(v)} + (1 - \lambda) \frac{1}{V} \quad (3)$$

where V is the size of the vocabulary.

2. Argue that Laplace smoothing is a bad smoothing approach. [Hint: Consider the case when V is very large.]
3. Show that the linear interpolation of a bigram LM and a unigram LM can be implemented as a backoff bigram LM. A backoff bigram LM is given as follows:

$$P_{bo}(v|u) = \begin{cases} f(v|u) & \text{if (u,v) exists} \\ bo(u) \cdot P(v) & \text{otherwise} \end{cases} \quad (4)$$

where $f(v|u)$ is a discounted distribution with $\sum_v f(v|u) < 1$.

4. Class-based LM is a common technique for LM smoothing. It is often represented as follows:

$$P(w_i|w_{i-1}) \approx P(w_i|c_i) \cdot P(c_i|c_{i-1}) \quad (5)$$

where each word maps into a single class. This assumption may not be reasonable when you want to allow a word to map into multiple classes with certain probability. Modify Eqn 5 to accommodate this change.

Problem 3: Computer Exercise

1. What does perplexity mean? Provide both a technical and an intuitive definition. How does reducing perplexity help with speech recognition? Can you think of a situation in which the task with lower perplexity is more difficult than the task with higher perplexity?
2. Build a statistical 3-gram LM using Good-Turing smoothing scheme with a training set. Find the perplexity of the LM on the heldout test sets.
You can download the SRILM toolkit at <http://www.speech.sri.com/projects/srilm/download.html>, and the data sets (swb_data.tgz) from the course webpage at <http://www.is.cs.cmu.edu/11-751/wiki>.
Training data: swb.train.100KW.text
Heldout test data: swb.heldout1, swb.heldout2, swb.heldout3
Report your trigram perplexity on these three test sets.
3. Look at the most frequent n-grams. Can you improve perplexity by manually collapsing some of the frequent n-grams into single words? (To do this, you need to modify the training and test sets using a script.)
Report top-20 most frequent n-grams (n=1,2,3).
Report your strategy and test result, including improvement, if any. If there is no improvement, explain.
4. Does a reverse (right-to-left) 3-gram LM perform better/equal/worse than a normal (left-to-right) 3-gram LM? Present the test perplexity results to support your intuition. (You need to reverse the training and test data using a script)
5. Devise and try at least one strategy to reduce test set perplexity. (You will win extra credit if you try more than one strategies).