CS 598cxz (/course/598f16) Fall 2016

Assignment #2: Basic Concepts in Information Theory

⚠ **Notice:** This assignment is due **Saturday, September 10th at 11:59pm**.

Please submit your solutions via Compass (https://compass2g.illinois.edu). You should submit your assignment as a **PDF**, and your accompanying code as either a **.zip** or **.tar.gz** containing your files.

So, since my NetID is `geigle1`, I would submit the following files:

- `geigle1-assignment2.pdf`
- `geigle1-assignment2.tar.gz` or `geigle1-assignment2.zip`

1. Entropy [30 pts]

Consider the random experiment of picking a word from an English text article. Let W be a random variable denoting the word that we might obtain from the article. Thus W can have any value from the set of words in our vocabularly $V = \{w_1, \ldots, w_N\}$, where w_i is a unique word in the vocabulary, and we have a probability distribution over all the words, which we can denote as $\{p(W = w_i)\}$, where $p(W = w_i)$ is the probability that we would obtain word w_i. Now we can compute the entropy of such a variable, i.e., $H(W)$.

a. **[10 pts]** Suppose we have in total N unique words in our vocabulary. What is the theoretical minimum value of $H(W)$? What is the theoretical maximum value of $H(W)$?

b. **[10 pts]** Suppose we have only 6 words in the vocabulary $\{w_1, w_2, w_3, w_4, w_5, w_6\}$. Give two sample articles using this small vocabulary set for which $H(W)$ reaches the minimum value and maximum value, respectively.

c. **[10 pts]** Suppose we have two articles A_1 and A_2 for which $H(W) = 0$. Suppose we concatenate A_1 and A_2 to form a longer article A_3. What is the maximum value can $H(W)$ be for article A_3? Give an example of A_1 and an example of A_2 for which A_3 would have the maximum $H(W)$.

2. Conditional Entropy and Mutual Information [20 pts]

a. **[10 pts]** What is the value of the conditional entropy $H(X \mid X)$?

b. **[10 pts]** What is the value of mutual information $I(X; Y)$ if X and Y are independent? Why?

3. Mutual Information of Words (Programming Exercise) [50 pts]

Warning: This part of the assignment requires programming, so please make sure you start early on it!

You should, in addition to your **PDF** submission, submit the code you write. You should have a README in the root directory of your archive file that explains how to run your code for **both** problems.

Mutual information can be used to measure the correlation of two words. Suppose we have a collection of N documents. For a word A in the collection, we use $p(X_A)$, where $X_A \in \{0, 1\}$, to represent the probability that A occurs ($X_A = 1$) in one document or not ($X_A = 0$). If word A appears in N_A documents, then $p(X_A = 1) = \frac{N_A}{N}$ and $p(X_A = 0) = \frac{N - N_A}{N}$. Similarly, we can define the probability $p(X_B)$ for another word B. We also define the joint probability of word A and B as follows:

- $p(X_A = 1, X_B = 1)$: the probability of word A and word B co-occurring in one document. If there are N_{AB} documents containing both word A and B in the collection, then $p(X_A = 1, X_B = 1) = \frac{N_{AB}}{N}$
- $p(X_A = 1, X_B = 0)$: the probability that word A occurs in one document but B does not occur in that document. It can be calculated as $p(X_A = 1, X_B = 0) = \frac{N_A - N_{AB}}{N}$.

a. **[10 pts]** Given the values of N_A, N_B, N_{AB} for two words A and B in a collection of N documents, can you write down the formulas for the rest two joint probabilities of A and B, i.e. $p(X_A = 0, X_B = 1)$ and $p(X_A = 0, X_B = 0)$?

b. **[20 pts]** Next, we will use the CACM test collection to do some real computation. You can download the data here (http://times.cs.uiuc.edu/course/598f16/cacm.trec.filtered), in which highly frequent words and very low frequent words have been removed (the vocabulary size of the original data (http://times.cs.uiuc.edu/course/598f16/cacm.trec) is very large, so we won't use it for this assignment). The CACM collection is a collection of titles and abstracts from the journal CACM. There are about 3,000 documents in the collection. The data set has been processed into lines. Each line contains one document, and the terms of each document are separated by blank space.

Using any programming language you like, for each pair of words in the collection, calculate the number of documents that contain both of the two words. Then, rank all the word pairs by their co-occurrence document counts (N_{AB}). Print the largest 10 counts (one count number per line).

(Hint: you may consider using a hash table to store the document counts for each word pair)

c. **[20 pts]** Now, calculate the mutual information $I(A; B)$ of all the possible word pairs in the collection. Rank the word pairs by their mutual information and print the results out.

Info: In practice, we need to do some smoothing in our formulas in order to avoid the $\log 0$ problem. For joint probability estimation, we'll assume that each of the four cases (corresponding to four different combinations of values of X_A and X_B) gets 0.25 "pseudo-counts". Thus, in total, we

introduced $0.25 \cdot 4 = 1$ "pseudo-count". We can then compute marginal probabilities based on the joint probability, i.e. $p(X_A = 1) = p(X_A = 1, X_B = 0) + p(X_A = 1, X_B = 1)$. For example, $p(X_A = 1, X_B = 1) = \frac{N_{AB}+0.25}{N+1}$ and $p(X_A = 1) = \frac{N_A+0.5}{N+1}$.

Please use these smoothing formulas in your code.

i. How are the top 10 pairs with the highest mutual information different from the top 10 pairs based on co-occurrence counts (from part b)?

ii. Write down the top 5 words which have the highest mutual information with the word "programming" in the collection. Do you think your results are reasonable?