# COMP 5790/ 6790/ 6796

# Special Topics: Information Retrieval

**Instructor: Shubhra ("Santu") Karmaker**

# Assignment #2: Basic Concepts of Information Theory. [100 points]

⚠ **Notice:** This assignment is due **Wednesday, August 8, 2021 at 11:59pm**.

Please submit your solutions via Canvas (https://auburn.instructure.com/). You should submit your assignment as a **typeset PDF**. Please do not include scanned or photographed equations as they are difficult for us to grade.

## 1. 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?

## 2. Mutual Information of Words [55 pts]

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. **[10 pts]** Next, we will use the following tables to do some real computation of Mutual Information. The tables contain the document counts for different words. There are a total of $N = 26,394$ documents in the collection.

Table 1 contains the document counts for words ‘computer’ and ‘program’, derived from the document collection (Hint: If $A = computer$ and $B = program$, then $N_{AB} = 349$. This means there are 349 documents that contain ‘computer’ AND ‘program’):

| | $X_{computer} = 1$ | $X_{computer} = 0$ |
|---|---|---|
| $X_{program} = 1$ | 349 | 2,021 |
| $X_{program} = 0$ | 1,041 | 22,983 |

Table 2 contains the document counts for words ‘computer’ and ‘baseball’, derived from the same document collection:

| | $X_{computer} = 1$ | $X_{computer} = 0$ |
|---|---|---|
| $X_{baseball} = 1$ | 23 | 2,121 |
| $X_{baseball} = 0$ | 1,367 | 22,883 |

Calculate $I(X_{computer}\ ;\ X_{program})$ and $I(X_{computer}\ ;\ X_{baseball})$ using the document counts from Table 1 and 2.

c. **[5 pts]** Compare the results of $I(X_{computer}\ ;\ X_{program})$ and $I(X_{computer}\ ;\ X_{baseball})$. Do the results conform with your intuition? Explain your intuition.

d. **[10 points]** Next, we will use the CACM test collection to do some real computation. You can download the data from here, in which highly frequent words and very low frequent words have been removed (the vocabulary size of the original data 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.

Use any programming language you like (You may find it relatively easy if you use perl or python), 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 cooccurrence document counts. Print the largest 10 counts (one count number per line) (Hint: you may consider using hash to store the document counts for each word pair)

e. **[20 points]** Calculate the mutual information of all the possible word pairs in the collection. Rank the word pairs by their mutual information and print the results out. How are the top 10 pairs with the highest mutual information different from the top 10 pairs that you've got from problem B (i.e., the 10 pairs with the highest counts of co-occurrences)? Please write down the top 5 words which have the highest mutual information with word "programming" in the

collection. Do you think your results reasonable? (Hint: In practice, we need to do some smoothing in our formulas in order to avoid the log0 problem. For joint probability estimation, we assume that each of the four cases (corresponding to four different combinations of values of $X_a$ and $X_b$) gets 0.25 pseudo count, thus in total we introduced 0.25*4=1 pseudo count. We can then compute marginal probability 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, $p(X_A=1, X_B=1) = (N_{AB} + 0.25) / (1 + N)$ and $p(X_A=1) = (N_A + 0.5)/(1+N)$. Please use these smoothing formulas in your code)

## 3. Kullback-Leibler Divergence (KL Divergence) [25 pts]

a. **[7 pts]** Please answer the following questions: 1) What is the range of KL Divergence? 2) Under which circumstances is KL Divergence equal to 0?

b. **[10 pts]** From the course we know that KL Divergence is not symmetric. Show that this is true by creating two probability distributions p and q, where $D(p||q) \neq D(q||p)$.

c. **[8 pts]** When calculating $D(p||q)$, what issues do you run into when an event has 0 probability in distribution q? How can you deal with 0 probabilities in this case?