SI 650 / EECS 549: Homework 1 – Probabilities, Text, and Ranking

Due: Wednesday, September 23, 4:30pm (see syllabus for late policy)

1 Probabilistic Reasoning and Bayes Rule (40 points)

Alice lost her phone a week ago. When she finally got a new phone with a replaced SIM card, she found she got a thousand new messages, many of which are just spam. She wanted to filter out the spam. To her sadness, she lost the contacts as well and could not know which messages are from her friends.

Luckily, Alice attended SI650 and decided to filter the messages in a Bayesian way. She went through 12 messages and noted down 4 observations for each message in Table 1.

1. S: whether the message is Spam (1 for yes, 0 for no);
2. H: whether the sender's number is local (1 for yes, 0 for no);
3. U: whether the message has an URL link (1 for yes, 0 for no);
4. E: whether the message has an emoji (1 for yes, 0 otherwise).

Alice wants to build a filter with these observations. Now in terms of probabilistic reasoning, we can formulate the question as evaluating the conditional probability $P(S|H, U, L)$, we say that the message is a spam if $P(S = 1|H, U, E) > P(S0 = 0|H, U, E)$. We make a further conditional independence assumption that $P(H, U, E|S) = P(H|S)P(U|S)P(E|S)$. In other words, we assume that if the status whether a message is a spam is known (i.e., value of S is known), the values of H, U, and E would be independent to each other.

a) (6 points) Fill in the Table 2 with conditional probabilities using only the information present in the 12 samples.

b) (8 points) With the independence assumption, use the Bayes formula and the calculated conditional probabilities to compute the probabilities that message M with $H = 0$, $U = 1$, $E = 0$ is a spam. That is, compute $P(S = 1|H = 0, U = 1, E = 0)$ and $P(S = 0|H = 0, U = 1, E = 0)$. Would you conclude that message M is a spam? Show your computation.

c) (6 points) Now, compute $P(S = 1|H = 0, U = 1, E = 0)$ and $P(S = 0|H = 0, U = 1, E = 0)$ directly from the 12 examples in Table 1, just like what you did in problem A. Do you get the same value as in problem B? Why?

S	H	U	E
0	1	1	1
0	0	1	0
1	0	1	1
0	1	1	0
1	1	0	1
0	0	0	0
0	1	1	0
0	0	1	1
0	0	0	0
1	0	1	1
1	0	1	0
1	1	0	0

Table 1: Sample observations of the messages

d) (5 points) Now, ignore Table 1, and consider any possibilities you can fill in Table 2. Are there any constraints on these values that we must respect when assigning these values? In other words, can we fill in Table 2 with 8 arbitrary values between 0 and 1? If not, are there any constraints on some values that we must follow? Describe your answer.

e) (5 points) Can you change your conclusion of problem B (i.e., whether message M is a spam) by only changing the value H (i.e., if the message comes from a local number) in one example of Table 1? Describe your answer.

f) (5 points) Note that the conditional independence assumption $P(H, U, E|S) = P(H|S)P(U|Y)P(E|S)$ helps simplify the computation of $P(H, U, E|S)$. In particular, with this assumption, we can compute $P(H, U, L|Y)$ based on $P(H|Y)$, $P(U|Y)$, and $P(L|Y)$. If we were to specify the values for $P(H, U, E|S)$ directly, what is the minimum number of probability values that we would have to specify in order to fully characterize the conditional probability distribution $P(H, U, E|S)$? Why? Note that all the probability values of a distribution must sum to 1.

g) (5 points) Explain why the independence assumption $P(H, U, E|S) = P(H|S)P(U|S)P(E|S)$ does not necessarily hold in reality.

S	$P(H = 1\|S)$	$P(U = 1\|S)$	$P(E = 1\|S)$	prior $P(S)$
1	0.4	?	?	?
0	?	?	?	0.583

Table 2: Conditional Probabilities and Prior

2 Text Data Analyses [40 points]

In this exercise, we are going to get our hands dirty and play with some data in the wild. Download two collections from Canvas, reddit-questions.10k.txt and wiki-bios.10k.txt. The first collection are 10,000 questions randomly sampled from r/AskReddit. The second collection is 10,000 biographies of people taken from Wikipedia. You can also find a stopword list in stoplist.txt.

A handy toolkit is the NLTK package (`http://www.nltk.org/`), which is often an easy-to-use library. You may also choose other NLP toolkits like SpaCy[1] or Stanza[2] that often offer more state-of-the-art algorithms or faster processing (or even deep learning!).

1. (10 points) Tokenize the text (e.g. use the `nltk.word_tokenize()` function in the NLTK package) and compute the frequency of words. Then, plot the frequency distribution of words in each collection after the removal of the stopwords: x-axis - word frequency (number of times a word appears in the collection); y-axis - proportion of words with this frequency. Plot the distributions on a log-log scale. Does each plot look like a power-law distribution? Are the two distributions similar or different?

2. (15 points) Now compare the two collections more rigorously. Report the following properties of each collection. Can you explain these differences based on the nature of the two collections? (20 points) (You can use the `nltk.pos_tag()` function of the NLTK package or the `nlp` function from SpaCy for part of speech tagging.)

   a) frequency of stopwords (percentage of the word occurrences that are stopwords.);

   b) percentage of capital letters;

   c) average number of characters per word;

   d) percentage of nouns, adjectives, verbs, adverbs, and pronouns;

   e) the top 10 nouns, top 10 verbs, and top 10 adjectives.

3. (10 points) We would like to summarize each document with a few words. However, picking the most frequently used words in each document would be a bad idea, since they are more likely to appear in other document as well. Instead, we pick the words with the highest TF-IDF weights in each document.

   In this problem, term frequency (TF) and inverse document frequency (IDF) are defined as:

   $$TF(t,d) = log(c(t,d)+1) IDF(t) = 1 + log(N/k).$$

   $c(t,d)$ is the frequency count of term t in doc d, N is the total number of documents in the collection, and k is the document frequency of term t in the collection.

   For each of the first 10 documents in the Wikipedia biographies collection, print out the 5 words that have the highest TF-IDF weights.

[1] `https://spacy.io`

[2] `https://stanfordnlp.github.io/stanza/`

4. (5 points) As discussed in the class, TF-IDF is a common way to weight the terms in each document. It can also be easily calculated from the inverted index, since TF can be obtained from the postings and IDF can be summarized as a dictionary. Could you think of another weighting that cannot be calculated directly from inverted index? What is the advantage of such a weighting?

- **Hint 1:** You can find a tutorial of nltk.word tokenize() and nltk.pos tag() in the NLTK book chapter 3 and 5: `http://www.nltk.org/book/ch03.htmlandhttp://www.nltk.org/book/ch05.html`. There is also a simple tutorial of NLTK at `http://www.slideshare.net/japerk/nltk-in-20-minutes`. Just use the simple, default settings.)
- **Hint 2:** You may find a lot of decision to make: Should I lower-case the words? Should I use stemmer or a lemmatizer? What to do with the punctuation? How should I handle html, markdown, or emoji? There is not always right and wrong, different answers are accepted. But you should write down clearly how you process the data in each parts and explain your decisions.

3 Document Ranking and Evaluation (20 points)

Suppose we have a query with a total of 10 relevant documents in a collection of 100 documents. A system has retrieved 20 documents whose relevance status is [++, -, ++, +, -, +, -, -, ++, +, -, ++, +, -, -, +, ++, -, ++, -] in the order of ranking. A + or ++ indicates that the corresponding document is relevant, while a - indicates that the corresponding document is non-relevant.

- (10 points) Compute the precision, recall, F1 score, and the mean average precision (MAP).
- (10 points) Consider ++ as the corresponding document being highly relevant ($r_i = 2$), while + indicates somewhat relevant ($r_i = 1$), - being non-relevant ($r_i = 0$). For the two rest relevant documents, treat them as somewhat relevant ($r_i = 1$) Calculate the Cumulative Gain (CG) at rank 10, Discounted Cumulative Gain (DCG) at rank 10, and Normalized Cumulative Gain (NDCG), at rank 10. Use $\log_2$ for the discounting function.

  Note You may find the definition of DCG in Wikipedia is different from the definition in our lecture. Please use the one in our lecture to calculate DCG and NDCG. (i.e. $DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i}$

Academic Honesty Policy

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the University's policies on

Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and professional integrity will be reported to Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed.