NYU Tandon - CS6923 Fall 2016
Machine Learning Professor Sandoval

Homework 3

Academic Honesty

In this homework you will not be allowed to collaborate. You may discuss with your classmates but all work submitted must be your own. Cheating and plagiarism **will not be tolerated**. If you have any questions about a specific case, *please ask me.*

NYU Poly's Policy on Academic Misconduct:
http://engineering.nyu.edu/academics/code-of-conduct/academic-misconduct

Due Date: **Dec 2 at noon.**

To Submit: Please submit using NYU classes. No late submissions accepted.

Lab Section:

Part 1 – Using Python for NLP - 60 points

Clone the following repo:
Github repo: https://github.com/GusSand/NyuPoly-CS6923-Fall16

In the homework3 folder you will a few files that you will use for this homework. One of them is data_file.txt. This file contains a list of all 77 talks at pyData Seattle 2015. Note that this data has all already been converted to UTF-8 for you. If you are interested in the code, it is the same directory, we are using a library called unidecode. This removes all the "funky" characters from your data.

We want to find an answer to the following question:
What is the trend in python and data in Seattle?

Questions:

Section A: Term Frequency with unigrams

Familiarize yourself with the python code inside the tools.py file. Note that it's well commented and at a high level basically reads the code from the data_file and parses it to remove weird characters.

Questions:

1. Without modifying the code, which is the most common word?
2. Notice that a lot of the more common words don't give much info since they are what we can stopwords. **Stop words** are natural language **words** which have very little **meaning**, such as "and", "the", "a", "an", and similar **words**. Now, look at the file called stopwords and modify tools.py so that it excludes stopwords in the counting. What are the 3 most common words now?
3. Given that this is a list of talks about data and python, do you think that there are some words that don't add much value? Name 3 of them and add them to the stopwords list
4. Now run your code again with the new list of stopwords. What's the most common word now?

Section B: Using TF-IDF
TF*IDF

Just counting unigrams, doesn't seem to help us much. It seems that we need more help. There's a technique called TF-IDF. This technique takes into account whether the word is rare or not, and gives higher values to rare words. It is composed of the Term Frequency (TF) and Inverse Document Frequence. Each of these is defined as follows:

TF = Term Frequency = Number of times the word appears in the document
IDF = Inverse Document Frequency = $\log (N/DF_t)$

TF*IDF then is defined as follows

$TF_t * IDF_t = TF_t * \log (N/DF_t)$

Where N is the number of documents. In our case 79.

Here is also a worked out example from Wikipedia on how to compute TF-IDF on a corpus:

Example of tf–idf [edit]

Suppose that we have term count tables of a corpus consisting of only two documents, as listed on the right.

Document 1	
Term	Term Count
this	1
is	1
a	2
sample	1

Document 2	
Term	Term Count
this	1
is	1
another	2
example	3

The calculation of tf–idf for the term "this" is performed as follows:

In its raw frequency form, tf is just the frequency of the "this" for each document. In each document, the word "this" appears once; but as the document 2 has more words, its relative frequency is smaller.

$$\mathrm{tf}(\text{"this"}, d_1) = \frac{1}{5} = 0.2$$
$$\mathrm{tf}(\text{"this"}, d_2) = \frac{1}{7} \approx 0.14$$

An idf is constant per corpus, and **accounts** for the ratio of documents that include the word "this". In this case, we have a corpus of two documents and all them include the word "this".

$$\mathrm{idf}(\text{"this"}, D) = \log\left(\frac{2}{2}\right) = 0$$

So tf–idf is zero for the word "this", which implies that the word is not very informative as it appears in all documents.

$$\mathrm{tfidf}(\text{"this"}, d_1) = 0.2 \times 0 = 0$$
$$\mathrm{tfidf}(\text{"this"}, d_2) = 0.14 \times 0 = 0$$

A slightly more interesting example arises from the word "example", which occurs three times but only in the second document:

$$\mathrm{tf}(\text{"example"}, d_1) = \frac{0}{5} = 0$$
$$\mathrm{tf}(\text{"example"}, d_2) = \frac{3}{7} \approx 0.429$$
$$\mathrm{idf}(\text{"example"}, D) = \log\left(\frac{2}{1}\right) = 0.301$$

Finally,

$$\mathrm{tfidf}(\text{"example"}, d_1) = \mathrm{tf}(\text{"example"}, d_1) \times \mathrm{idf}(\text{"example"}, D) = 0 \times 0.301 = 0$$
$$\mathrm{tfidf}(\text{"example"}, d_2) = \mathrm{tf}(\text{"example"}, d_2) \times \mathrm{idf}(\text{"example"}, D) = 0.429 \times 0.301 \approx 0.13$$

(using the base 10 logarithm).

5. Compute TF-IDF for the corpus. Note that you will need to compute a separate score for TF per document. Again in our file we have 79 documents.
6. Rank each unique word by their TF-IDF score. Which one has the highest score? Use only the highest for each word. So if a word appears twice only use the highest score.
7. Is the space ‘ ‘ still there? As one of the words. If so how can you remove it?
8. Paste the output for both Section A and Section B

Part 2 – Pen and Pencil (40 points)

2. Consider an information retrieval task, where the problem is to predict whether a given reader will be interested in book *x*. The book should be labeled + if the reader will be interested in it, and otherwise.

The following table shows a small training set for this problem, consisting of 7 books. It also shows the predictions made by a hypothesis *h* on these books. The

column labeled "r" gives the label of the book in the training set, and the column labeled *y* gives the prediction of *h*.

Book	r	y
War and Peace	+	-
Pattern Recognition	+	+
How to Win Friends and Influence People	-	-
The Philosophical Breakfast Club	+	-
Harry Potter	-	+
Godel Escher Bach	+	+
Photoshop for Dummies	-	-

Calculate the recall and precision of *h*.

(If we imagine using *h* to decide which books to give to the reader, then Recall is calculated by considering the set of all books that are interesting to the reader, and determining what percentage of those books are given to the reader. Precision is calculated by considering all books that are given to the reader, and determining what percentage of those books are interesting to the reader.)

3. Let *h* be a hypothesis for a binary classification problem which assigns a probability or score to each example *x* in a test set X. We can obtain different points on the ROC curve for *h* by changing the threshold θ in the rule "if $h(x) > \theta$, then classify *x* as +". The number of points on the ROC curve is |X| + 1.

In fact, the points on the ROC curve do not depend on the exact values assigned by *h*. To obtain the ROC curve, it is enough to give the ranking of the elements x^t of the test set induced by the values of *h* on those elements. In other words, it is enough to list the elements x^t in the test set in descending order of the value of $h(x^t)$.

We hope that *h* will assign higher values to the positive examples than to the negative examples.
Let X be a test set containing 6 examples. Three of the examples are labeled +, and three are labeled -. Suppose we list the examples x^t in X in descending order of the value of $h(x)$.

(a) Suppose *h* ranks all 3 positive examples in X higher than all 3 negative examples in X (i.e., the highest values of *h* are on the positive examples). Graph the resulting ROC curve. What is the AUC (the area under the curve in the interval [0,1])?

(b) Suppose h ranks all 3 negative examples in X higher than all 3 positive examples in X. Graph the resulting ROC curve. What is the AUC?

(c) Suppose that the ranking assigned by h corresponds to the following list of labels, if we list the examples in descending order of rank: (+,+,−,+,−,−). So, the two highest values of h are given to two of the positive examples, the next highest to a negative example, etc. Graph the resulting ROC curve. What is the AUC?