CS47300 Fall 2020 Assignment 3 Solutions

1 LSI

1. Given a term-document matrix and after applying SVD on it, explain in your own words the semantics of the U, S and V matrices.
   U, term-concept. S, concept space. V, document-concept.

2. Given the following U, S, and V matrices, and the query term frequenc, show how to compute similarity between the query and the documents. What is the ranking of documents?

```
       high      0 1                D1  D2  D3
   U:  term      1 0    S:  2 0  VT:  1   1   0
       frequenc  2 1        0 1       0   1   1
```

   Bonus: Could this actually be a real set of matrices under LSI? Briefly explain why or why not. Hint: there is a way you can answer this mathematically.
   $cos(q, d_1) = \frac{q \cdot d_1}{|q||d_1|} = 0.83250$
   $cos(q, d_2) = \frac{q \cdot d_2}{|q||d_2|} = 0.98058$
   $cos(q, d_3) = \frac{q \cdot d_3}{|q||d_3|} = 0.5547$
   Rank: $D_2 > D_1 > D_3$
   Bonus: No. $U * S * V^T$ **should be orthogonal.**

3. Explain why zeroing low singular values works when we are reducing the number of dimensions? (This should be a mathematical explanation.)
   Given $A = U * S * V^T$**, dimensionality reduction is done by neglecting small singular values in the diagonal matrix** S**. Then you will obtain a matrix** $\widetilde{A} = U_r * S_r * V_r^T$**. In particular, you don't drop any rows or columns. It has exactly the same dimensions as before, but has a reduced rank. Which you can keep the most important information.**

4. Give some pros and cons (at least one of each) of using Latent Semantic Indexing for web search, other than computational cost/complexity/space. (In other words, assume that we have a cost- and space-efficient way of doing SVD and the appropriate matrix operations at web scale.) (Explaining each in one or two sentences are enough.)

   (a) **pros: can reduce high dimentionality to low and keep the most important information.**

   (b) **cons: perform not well when the same word has different meanings.**

   Comment: I'm open to any reasonable answers.

2 Binary Independence Model

Given the following corpus:

D1 I love cats cats cats

D2 I love cats and I love dogs

D3 I love dogs

D4 I don't like cats

D5 I love puppies

D6 I love kittens

And the following queries:

Q1 cats

Q2 dogs

1. First, start with standard assumptions: a large corpus (N=1000), few relevant documents (S=3), and each term occurs in only the documents above. Because the numbers are small relative to N, you can assume that N-n or N-S is approximately N. Also assume pi=0.5; i.e., if a term is relevant, it has a 0.5 probability of occuring in a relevant document. Calculate the Retrieval Status Value for each query and the documents above, and rank the documents.

   RSV(Q1,D1) $= \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \approx \sum_{x_i=q_i=1} \log \frac{N}{n_i} = \log \frac{1000}{3}$
   D2 and D4 are the same since cats appears in them). D3, D5 and D6 are 0, as the sum includes no items (nothing for which $x_i = q_i = 1$).

   This gives a ranking of D1=D2=D4 > D3=D5=D6.

   The only difference is in $n_i = 2$ (and which queries have the term), so:
   RSV(Q2,D2) $= \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \approx \sum_{x_i=q_i=1} \log \frac{N}{n_i} = \log \frac{1000}{2}$
   D3 is the same. D1, D4, D5 and D6 are 0, as the sum includes no items (nothing for which $x_i = q_i = 1$).

   This gives a ranking of D2=D4 > D1=D3=D5=D6.

2. Assume you were told that the relevant documents for Q1 were D1, D2, and D6; and the relevant documents for Q2 were D2, D3, and D5. Use these relevance scores to estimate values for p_{cat}, r_{cat}, p_{dog}, r_{dog}. Again, calculate the RSV between each query and the six documents and rank the results for each query.

   We now have $p_{cat} = p_d oc = 2/3$, $r_{cat} = (3-2)/1000$, and $r_{dog} = 0/1000$.

   RSV(Q1,D1) $= \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)} = \log \frac{(2/3)(1-.001)}{.001(1-(2/3))} \approx \log \frac{2-.002}{.001} \approx \log 2000$
   Again, D2 and D4 are the same, and the others are 0. While the RSV has changed, the ranking is unchanged.

   For Q2, we have a problem, as the denominator in the RSV is 0 since $r_{dog} = 0$. We can see that it is the same (undefined) for D2 and D3, and 0 for the rest. To make it interesting, we can either apply some sort of smoothing, or we can look what happens in the limit as $r_i \rightarrow 0$. In this case the RSV goes to infinity as r_i goes to 0, which suggests that if a word occurs in no irrelevant documents, then its presence in a document means it has a high probability of being relevant (as we'd expect.)

3. Using some known example queries and relevant documents to calculate the term relevance probabilities, as with the second part of this question, can cause a few interesting things to happen. Pick one difference between the results for Part 1 and Part 2, and briefly describe what has changed and why.

   Several interesting things. One would be a thorough discussion of the meaning of $r_i = 0$, as above. Another is to note that the relative weighting of different words changes, or RSVs increase as we are more certain of the answer. The key is to give some idea of what this change means in terms of the problem we are trying to solve (ad-hoc retrieval), which could be it doesn't change ranking (at least with one-word queries), or that it gives a stronger differentiation between how important different words are (think what would happen with the query "cat dog".)

3 OKAPI (BM25)

The BM25 model builds on the Binary Independence Model, but incorporates term frequency:

$tf_t d$ Frequency of the term t in the document d. (f_i in the Croft et al. book Section 7.2.2.)

$tf_t q$ Frequency of the term t in the query. qf_i in the book.

The other change is to normalize by document length, so that documents with many terms or more frequent terms don't "win" simply because they are longer. There are also three parameters k1, k3 (k2 in the book), and b that govern how much effect document term frequency, query term frequency, and length normalization have.

1. The Okapi / BM25 model is based on the Binary Independence Model. What values of the k/b parameters would cause BM25 to most closely approximate the BIM model, and how would it differ?

   First, BIM:

$$c_i = \log \frac{\frac{s}{(S-s)}}{\frac{(n-s)}{N-n-S+s}}$$

   For Okapi:

$$RSV_d = \sum_{t \in q} [[\frac{\frac{|VR_t|+\frac{1}{2}}{|VNR_t|+\frac{1}{2}}}{\frac{df_t-|VR_t|+\frac{1}{2}}{N-df_t-|VR|+|VR_t|+\frac{1}{2}}}] \times \frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}})+tf_{td})} \times \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}]$$

   First terms look very similar! So, other terms in Okapi multiplication should equal 1:

$$\frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}})+tf_{td})} \times \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}] = 1$$

   Therefore:

$$\frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}})+tf_{td})} = 1$$

$$\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}} = 1$$

   For first term:

$$\frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}+tf_{td}})} = 1 \rightarrow (k_1+1)tf_{td} = k_1((1-b)+b(\frac{L_d}{L_{ave}})+tf_{td})$$

$$\rightarrow k_1 * tf_{td} + tf_{td} = k_1 - k_1 * b + k_1 * b(\frac{L_d}{L_{ave}}) + k_1 * tf_{td}$$

Set b = 0:

$$\rightarrow k_1 * tf_{td} + tf_{td} = k_1 + k_1 * tf_{td}$$

$$\rightarrow k_1 * tf_{td} - k_1 * tf_{td} + tf_{td} = k_1$$

$$\rightarrow tf_{td} = k_1$$

For term 2:

$$\frac{(k_3+1)tf_{tq}}{k_3 + tf_{tq}} = 1$$

Set $k_3t = 0$:

$$\frac{(0+1)tf_{tq}}{0 + tf_{tq}} = \frac{tf_{tq}}{tf_{tq}} = 1$$

So for THIS EQUATION (different between slides and book!), $k_1 = tf_td$, $k_3 = 0$, $b = 0$.

Eqn 2 for Okapi:

$$RSV_d = \sum_{t \in q}[[\frac{\frac{|VR_t|+\frac{1}{2}}{|VNR_t|+\frac{1}{2}}}{\frac{df_t-|VR_t|+\frac{1}{2}}{N-df_t-|VR|+|VR_t|+\frac{1}{2}}}] \times \frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}}))+tf_{td}} \times \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}]$$

For first term:

$$\frac{(k_1+1)tf_{td}}{k_1((1-b)+b(\frac{L_d}{L_{ave}}))+tf_{td}} = 1$$

Set $k_1 = 0$, $b = 0$:

$$\frac{(0+1)tf_{td}}{0((1-0)+0(\frac{L_d}{L_{ave}}))+tf_{td}} = \frac{tf_{td}}{tf_{td}} = 1$$

For second term: exact same work as for first equation, nothing changed.

So for ALTERNATE EQUATION (this is the one from the book!), $k_1 = 0$, $k_3 = 0$, $b = 0$. Both answers should be correct. This degree of mathematical work would have been idea, but full credit was given for less. And finally, the difference even if you set k/b properly is the BM25 model is a smoothed version of the BIM, indicated by the $+\frac{1}{2}$ on each numerator and denominator in the first term of BM25 Okapi.

2. We've already shown that the BIM is very close to Inverse Document Frequency. With the k and b values set to 1, BM25 also incorporates term frequency. How does this differ from tf*idf with cosine similarity?

First, lets see what the constants do to the BM25 Okapi equation:

$$RSV_d = \sum_{t \in q} [[\frac{\frac{|VR_t|+\frac{1}{2}}{|VNR_t|+\frac{1}{2}}}{\frac{df_t-|VR_t|+\frac{1}{2}}{N-df_t-|VR|+|VR_t|+\frac{1}{2}}}] \times \frac{(1+1)tf_{td}}{1((1-1)+1(\frac{L_d}{L_{ave}})+tf_{td})} \times \frac{(1+1)tf_{tq}}{1+tf_{tq}}]$$

$$= \sum_{t \in q} [[\frac{\frac{|VR_t|+\frac{1}{2}}{|VNR_t|+\frac{1}{2}}}{\frac{df_t-|VR_t|+\frac{1}{2}}{N-df_t-|VR|+|VR_t|+\frac{1}{2}}}] \times \frac{2 * tf_{td}}{\frac{L_d}{L_{ave}}+tf_{td}} \times \frac{2 * tf_{tq}}{1+tf_{tq}}]$$

First thing of note, L_d and L_{ave} are both present in the BM25 Okapi equation. This indicates that the length of the document as compared with the average length of documents in the corpus is taken into account, whereas with tfidf these values aren't taken into account. One way to think about this is as vectors. TFIDF being simple cosine similarity means that documents are on some level assumed to be all the same length; the only thing that changes between documents are angles. One large magnitude vector (lots of occurrences of words) could have a very similar angle with a low magnitude vector. A document with 1000 occurrences of word c_1 and 1 occurrence of word c_2 could end up possessing a surprisingly high similarity score, relative to other documents in the corpus, to a document that has 1 occurrence of word c_1 and word c_2 each, which could intuitively not make sense in a particular situation. I would assert this is the main, mostly quantifiable difference. One could also argue there is a difference in how they deal with the term frequencies, but that's more complicated though still credit-worthy if the reasoning was sound.

For the above questions, you should justify your answers mathematically. For example, for the second question you could show how a change in term frequency between two or three documents leads to different rankings in each model.

4 Web search

1. In the class, we discussed three bad assumptions for the relevance of classic IR: context ignored, individuals ignored, and corpus predetermined. For each of these, give one idea on how can we incorporate it into the relevance calculation. (2-3 sentences each.)

   - ***Context* includes user search history, location, the timeline of search (day, night), and many others. The ranking model can leverage this information during the ranking documents. First, rank the documents based on the query terms and then re-rank them by refining with the contextual information.**
   - ***Individual*: The search system can create a user profile based on the information, including search history, age, occupation, and email text/message. It can use this profile for re-ranking or refining of retrieved documents.**

- ***Predetermined corpus*****: The search system can frequently update indexing. The ranking model can weight newer documents higher than old documents for a time-sensitive query.**

2. The goal of a search system is to fulfill the information needs of users that they typically specified with a query. However, the query text can be ambiguous and thus might have more than one meaning. For example, consider the query java. It might mean java programming language or an island in Indonesia. What would be a good strategy to retrieve documents for a search engine in such a scenario? Try searching with java in Google, and Bing and see how they perform! Do you see a winner? (hope they do not change their retrieval model frequently!)

   - **The search system can diversify returned results based on the query's possible meaning in case of ambiguity. Contextual information can help to prioritize relevant documents with a particular interpretation of the query.**

3. How does the size of the index impact search performance? Does a larger index size guarantee a better performance? Why or why not? (Remember, search performance is asking how relevant the returned documents are, not how fast it runs.)

   - **The search engine needs to have enough and diverse document collection to serve diverse users' information needs. However, the larger corpus also poses challenges to the ranking models, as they now need to deal with more documents. After all, the search system can show a fixed number of documents at the top (e.g., 10) despite having a larger index size.**
   - **The search performance (i.e., relevance) depends on many factors; thus, increasing the index size does not guarantee better performance. Some good index choices can be to avoid duplication, diversify corpus, and cover as much topic as possible based on the expected users' type.**

5 Web Crawling: Freshness

Assume we have two servers we want to crawl, A and B. We also have two class of documents: F changes frequently (30

To manage Freshness, we have a separate queue for F (frequent change) and S (infrequent change) documents. The crawler alternates between queues, taking one from F, then one from S, etc. (If a queue is empty, the next document is taken from the other queue.) Queues are FIFO.

Determine what happens in such a system, particularly:

1. Do the F documents get crawled more often than S documents?

   F may be crawled more often than S. Although F and S are crawled in the same rate, F updates more frequently than S. More documents from F can be filled into the queue. So F documents may get crawled more often than S.

2. Does one of the queues fill up? **No. Because the documents are put into the system for crawling at the same rate that they are crawled.**

3. Assume the mean change frequency of crawled web pages of A is once per 2 days, pages are crawled once every 4 days, and the mean change frequency of crawled web pages of B is once per 10 days, pages are crawled once every 10 days. What is the average age of the crawler A and B?

   Server A: $Age = \int_0^4 \frac{1}{2}e^{-\frac{1}{2}x}(4-x)dx \approx 2.271$ **days**

   Server B: $Age = \int_{10}^0 \frac{1}{10}e^{-\frac{1}{10}x}(10-x)dx \approx 3.679$ **days**