CS47300 Fall 2018 Midterm 2 solutions, November 14, 2018
Prof. Chris Clifton

Turn Off Your Cell Phone. Use of any electronic device during the test is prohibited. As previously noted, you are allowed notes: Up to four sheets of 8.5x11 or A4 paper, single-sided (or two sheets double-sided).

Time will be tight. If you spend more than the recommended time on any question, go on to the next one. If you can’t answer it in the recommended time, you are either giving too much detail or the question is material you don’t know well. You can skip one or two parts and still demonstrate what I believe to be an A-level understanding of the material.

Note: It is okay to abbreviate in your answers, as long as the abbreviations are unambiguous and reasonably obvious.

In all cases, it is important that you give some idea of how you derived the answer, not simply give an answer. Setting up the derivation correctly, even if you don’t carry out the calculations to get the final answer, is good for nearly full credit.

The total points possible is 30. I would consider a 25 and above to be A-level performance, 19 and above to be B-level, 13 and above to be C-level.

1 Query Expansion (8 minutes, 4 points)

You are given a choice of two query expansions methods, both based on a thesaurus. One uses synonyms (words with the same meaning), and adds synonyms to the query. The other uses antonyms (words that have the opposite meaning), but adds the antonyms to the query using a negative weight.

A. Which method would you expect to improve precision? Explain your answer.

Negatively weighting antonyms will result in lowering the score for documents that talk about the opposite of what is in the query. As this would reduce the number of non-relevant documents (reducing the numerator and denominator by the same valuees), I would expect this to improve precision.

Scoring: 1 for antonym, 1 for good explanation

B. Which method would you expect to most improve recall? (Again, explain your answer.)

Adding synonyms will find additional documents that talk about similar things, so this should find additional relevant documents, improving recall.

Scoring: 1 for synonym, 1 for good explanation capturing that any increase in relevant documents return improves recall.

Minimum 1 point for showing some understanding that query expansion adds words to the query.

Note that there is no guarantee of improvement. For example, if I am searching for Thanksgiving dinner with the query “turkey”, adding a synonym like “fool” will hurt precision, and probably will not help recall.

2 Categorization (10 minutes, 6 points)

Given the following corpus containing documents divided into three categories:

<table>
<thead>
<tr>
<th>Hot Food</th>
<th>Cold Food</th>
<th>Dance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$: mashed potato</td>
<td>$D_4$: marinated beet salad</td>
<td>$D_7$: mashed potato</td>
</tr>
<tr>
<td>$D_2$: baked potato</td>
<td>$D_5$: blueberry chill</td>
<td>$D_8$: mambo shuffle</td>
</tr>
<tr>
<td>$D_3$: lyn’s chill chaser soup</td>
<td>$D_6$: cold cucumber soup</td>
<td>$D_9$: let’s chill</td>
</tr>
</tbody>
</table>

You are asked to classify the document $D_{new}$: chill beet soup

A. Using $k$-nearest neighbor, with $k = 1$, and cosine similarity as a distance metric, what category does $D_{new}$ go into? Weight the terms using the raw term count. Briefly explain how you come up with your
answer. *Hint: You only need to calculate cosine similarity two or three times, but you need to explain why.*

Using raw term count, and no IDF, we basically have documents with no matching terms (similarity 0), 1 matching term (D4, D5, D6, D9), and two matching terms (D3). This lets us know that it will be either D3, or (D4, D5, D6, D9). D4 and D6 have the same term overlap and would be the same, as do D5 and D9. Since the numerators will be the same, and the denominator is smaller for D5/D9 (fewer total words), we really only need to calculate:

\[
D5: \frac{1}{\sqrt{1^2+1^2+1^2}}
\]

\[
D3: \frac{1+1}{\sqrt{3\sqrt{4}}}
\]

Clearly \( \frac{1}{\sqrt{3}} > \frac{1}{\sqrt{3}\sqrt{2}} \), so \( D_{new} \) is close to \( D_3 \), and is classified with Hot Food.

*Scoring:* 1 for correct answer, 1 for nearest DOCUMENT for 1-nn, 1 for using term counts, 1 for showing understanding of cosine similarity.

B. Would your answer be different using Naïve Bayes? You don’t need to work this out completely, you can get credit by explaining why the answer would or would not be different.

Naïve Bayes uses all words in the category, and looks at the probabilities of each word independently. The word “chill” appear once in all categories, so won’t have much impact. The word “beet” appears in only one category, so will have higher impact. Combined with Cold Food being the only category with all words, this makes Cold Food the most likely result.

*Scoring:* 1 for more weight to terms only in one category or uses all terms in category 1 for “beet” carries more weight than “chill” or other means of deriving answer (or working out answer.) Simply noting that this was based on probabilities and showing some understanding of how that applied to this problem was worth one point.

### 3 Clustering (12 minutes, 6 points)

Given the following documents:

- **D1** Information Retrieval Machine Learning
- **D2** Electron Microscope Materials Engineering
- **D3** Machine Learning Prediction
- **D4** Ad-hoc Information Retrieval
- **D5** Materials Fabrication
- **D6** Microscope Materials Design

**A. What do you think is a good clustering for these documents?**

- Cluster 1:
  - D1 D3 D4
- Cluster 2:
  - D2 D5 D6

*Scoring:* 1 for all documents clustered, 1 for correct clustering
B. Given D1 and D2 as seed documents, compute the 2-means clustering for these documents. Use raw term weights and cosine similarity. Hint: You only need to know which cluster center a document is more similar to. With these documents, you can easily prove which is closer without actually calculating the full cosine similarity, and prove how the clustering terminates without actually calculating the new cluster means. Although calculating the cosine similarities you need won’t take that long.

Clearly the cosine similarity is 0 for D1 to D5 and D6, or D2 to D3 and D4 (no overlapping terms), and is non-zero (at least one overlapping term) for D1 to D3/D4 and D2 to D5/D6. Computing the new means will still leave 0 in the terms “Electron Microscope Materials Engineering Fabrication Design” for everything in the D1/D3/D4 cluster, and vice-versa, so the same argument says that the next iteration gets the same clustering. Since the clustering didn’t change, we terminate with the same clusters as given in part 1.

Scoring: 1 for zero cosine similarity calculation or proof, 1 for non-zero cosine similarity calculation or proof, 1 for termination calculation or proof, 1 for correct final clustering, 1 for showing some understanding of k-means. Max 4 points.

4 PageRank (8 minutes, 7 points)

Given the following graph:

A. Give the Transition probability matrix \((B)\) for this graph. You can assume the “teleport rate” \(\alpha = 0.1\).

\[
\begin{bmatrix}
0.0 & 0.9 & 0.1 \\
0.1 & 0.0 & 0.9 \\
0.9 & 0.1 & 0.0 \\
\end{bmatrix}
\]

Scoring: 1 for 3x3 matrix, 1 for 1 large value per non-diagonal row/column, 1 for equal large values, 1 for rows/columns sum to 1 each value. If you only has 1 point based on the above, you could get a 2nd for having 0’s on the diagonal.

B. Calculate PageRank for the nodes in the graph. Hint: If you think of what PageRank means, you can answer this without doing the actual calculation – but explain how you came up with your answer.

If we think of this as a random walk, we can see that it is symmetrical, with no “orphan” sources or sinks, so a random surfer will not get stuck anywhere and will reach everywhere. Because of the symmetry, the amount of time spent in each pages is equal, so the answer is \(PR^T = [1/3 \ 1/3 \ 1/3]\).

Scoring: 1 for decent start on calculation or decent explanation, 1 for equal weights or decent start on calculation, 1 for 1/3

5 Collaborative Filtering (12 minutes, 7 points)

A. Given the following table of TA course preferences:

<table>
<thead>
<tr>
<th></th>
<th>CS348</th>
<th>CS373</th>
<th>CS448</th>
<th>CS471</th>
<th>CS473</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akshat</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Rajkumar</td>
<td>4</td>
<td></td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Rohan</td>
<td>4</td>
<td>2</td>
<td>?</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Using Vector Space Similarity, is the rating for the \( ? \) (predicting Rohan’s preference for CS471) greater than or less than 3? For full credit, set up the relevant formulas and explain why they lead to your answer (you may carry out the calculations, but you don’t need to.)

The key is the formula 
\[
R_{Rohan}(CS471) = \bar{R}_{Rohan} + \frac{\sum_u w_{u, Rohan}(R_u(CS471) - \bar{R}_u)}{\sum_u |w_{u, Rohan}|}.
\]

Since Akshat didn’t rank CS471, he isn’t considered. Rajkumar’s rating for CS471 was below his average, so unless the cosine similarity was negative, the only possibility is that the adjustment from Rohan’s average is negative, giving a result less than his average of 3.

Scoring: 1 for incorporating user similarity, 1 for Rajkumar’s rating, 1 for complete formula, 1 for “<”.

B. One criticism of collaborative filtering is that it leads to “filter bubbles” and cultural isolation: you only see what people who are like you see. Explain how over the long term, memory-based collaborative filtering would be expected to increase this effect. For full credit, show mathematically (in terms of a memory-based approach formula) why this happens. Hint: Assume that you are more likely to see (and rate) items that have been recommended to you by the collaborative filter.

Memory-based collaborative filtering will give more weight to things highly rated by similar individuals, and you will only be similar to those who have rated things you have.

\[
w_{u, u'} = \frac{\sum_u R_u(o) R_u(o)}{||R_u'||||R_u||}
\]

So if nobody in your group rates item \( X \), you won’t see (or rate) item \( X \) - you won’t have the opportunity to get closer to those who see other things, but you can get increased similarity to those in your group. By:

\[
R_{u'}(CS471) = \bar{R}_{u'} + \frac{\sum_u w_{u, u'}(R_u(CS471) - \bar{R}_u)}{\sum_u |w_{u, u'}|}
\]

your recommendations will be heavily weighted by what those in your group like. Since this holds for all in the group, the chance of seeing new things continues to decrease.

Scoring: 1 for becoming more similar to users who see things you see, 1 for this increases their weight in prediction, 1 for formula.