Qn I. [In the following, you must SHOW YOUR WORK to get partial credit] Assume that the total number of documents in a corpus is 1024 and that the following words occur in the following number of documents:

"Computer" occurs in 32 documents
"software" occurs in 8 documents
"intelligent" occurs in 16 documents
"robust" occurs in 1024 documents

1. [6pt] Calculate the TF-IDF weighted term vector for the following document D. Assume that the log in the idf weight is taken to the base 2. (Hint: all the numbers above are powers of 2).

"Computer intelligent software robust computer software"

\[
\begin{align*}
\omega (\text{Computer}) &= 2 \times \log\left(\frac{2^{10}}{2^{35}}\right) = 2 \times 5 = 10 \\
\omega (\text{Software}) &= 2 \times \log\left(\frac{2^{10}}{2^{3}}\right) = 2 \times 7 = 14 \\
\omega (\text{Intelligent}) &= 1 \times \log\left(\frac{2^{10}}{2^{4}}\right) = 2 \times 6 = 6 \\
\omega (\text{Robust}) &= 1 \times \log\left(\frac{2^{10}}{2^{10}}\right) = 1 \times 0 = 0
\end{align*}
\]

\[D = [10, 14, 6, 0]\]

The vector corresponding to Computer, Software, intelligent, Robust.
2. [4pt] Suppose I have a query Q which is specified as "Intelligent Software".

Assuming that query vector is computed just in terms of TF weights (no IDF weights), and similarity is measured by the cosine metric, what is the similarity between Q and D?

\[
Q = \begin{bmatrix}
0 & 1 & 1 & 0
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
10 & 14 & 6 & 0
\end{bmatrix}
\]

\[
\text{Sim} = \frac{0 \cdot 10 + 1 \cdot 14 + 1 \cdot 6 + 0 \cdot 0}{\sqrt{2} \sqrt{10^2 + 14^2 + 6^2}} = \frac{20}{\sqrt{2} \sqrt{332}}
\]

3. [3pt] Suppose the user is shown D in response to the query Q, and the user says that D is relevant to his query. If we now use relevance feedback to modify Q, what will the query vector become? Assume that alpha, beta and gamma are all 1.

\[
Q' = \alpha Q + \beta (\text{centroid of relevant docs}) - \gamma (\text{centroid of irrelevant docs})
\]

Since only one doc D, and no irrelevant docs:

\[
Q' = \begin{bmatrix}
0 & 1 & 1 & 0
\end{bmatrix} + \begin{bmatrix}
10 & 14 & 6 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
10 & 15 & 7 & 0
\end{bmatrix}
\]
Qn II. Suppose we have 3 web pages p1, p2 and p3, such that p1 has links to p2 and p3; p2 has link to p3 and p3 has link to p1 (see the picture)

(a) [6pt] Show one iteration of authorities and hubs algorithm. Assume you set all the authorities and hub values to 1 in the beginning. Show all the steps.

\[
A = \begin{bmatrix}
  p_1 & p_2 & p_3 \\
p_2 & 0 & 1 \\
p_3 & 1 & 0 \\
\end{bmatrix}
\]

\[
\Phi = \begin{bmatrix}
  1/3 & 1/3 & 1/3 \\
\end{bmatrix}
\]

\[
A u_0 = \begin{bmatrix}
  1/3 \\
end{bmatrix}
\]

\[
H_0 = \begin{bmatrix}
  1/3 \\
\end{bmatrix}
\]

\[
A u_1 = A' \times A u_0 = \begin{bmatrix}
  0 & 0 & 1 \\
  1/3 & 0 & 1/3 \\
  1/3 & 1/3 & 0 \\
\end{bmatrix} \begin{bmatrix}
  1/3 \\
  1/3 \\
  1/3 \\
\end{bmatrix} = \begin{bmatrix}
  1/3 \\
  1/3 \\
  1/3 \\
\end{bmatrix}
\]

normalized
\[
\hat{A} u_1 = \frac{1}{\sqrt{6}} \begin{bmatrix}
  1 \\
  1 \\
  1 \\
\end{bmatrix}
\]

\[
H u_1 = A \times \hat{A} u_1 = \begin{bmatrix}
  0 & 1 & 1 \\
  1/3 & 0 & 1/3 \\
  1/3 & 1/3 & 0 \\
\end{bmatrix} \begin{bmatrix}
  1/3 \\
  1/3 \\
  1/3 \\
\end{bmatrix} = \begin{bmatrix}
  3/\sqrt{6} \\
  2/\sqrt{6} \\
  3/\sqrt{6} \\
\end{bmatrix}
\]

normalized
\[
\hat{H} u_1 = \frac{1}{\sqrt{14/6}} \begin{bmatrix}
  3/\sqrt{6} \\
  2/\sqrt{6} \\
  3/\sqrt{6} \\
\end{bmatrix} = \begin{bmatrix}
  3/\sqrt{14} \\
  2/\sqrt{14} \\
  3/\sqrt{14} \\
\end{bmatrix}
\]
(b) [5pt] Show the augmented transition matrix, that will be used by the PageRank algorithm, assuming that with \( c \) probability a random surfer will follow the links on the current page, and that with \((1-c)\) probability she will transition to any of the (three) pages with uniform probability; where \( c \) is set to 0.8.

\[
\begin{pmatrix}
0 & 0 & 1 \\
0.2 & 0 & 0 \\
0.2 & 1 & 0 \\
\end{pmatrix} + 0.2 \times 
\begin{pmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\end{pmatrix}
\]

(c) [2pt] Suppose we set \( c \) to 0, then what will be the page ranks associated with the three pages?

\[
\begin{pmatrix}
0 & 0 & 0.8 \\
0.4 & 0 & 0 \\
0.4 & 0.8 & 0 \\
\end{pmatrix} + 
\begin{pmatrix}
0.066 & 0.066 & 0.066 \\
0.066 & 0.066 & 0.066 \\
0.066 & 0.066 & 0.066 \\
\end{pmatrix}
\]

A note on interpreting eigenvectors as probability distributions. You need to divide each element of the vector by the sum of all elements (so that all sum to 1). In contrast, for the \( H^* \) in part a, the page ranks will be 0.384 \( P_1 \), 0.220 \( P_2 \), and 0.396 \( P_3 \). So the PageRank, which is just the stationary probability distribution, is going to be 0.333 for all pages.
Qn III. Consider the following T-D matrix defining 6 documents defined in terms of 4 keywords.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush</td>
<td>5</td>
<td>15</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Kalahari</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Iraq</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Saddam</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

We decide to reduce the noise and dimensionality of this data through SVD analysis. The SVD of this T-D matrix, according to MATLAB is: \( U S V^T \) where \( U, S, V \) are given by:

\[
\begin{align*}
U &= \begin{pmatrix}
0.8817 & 0.1969 & -0.0444 & -0.4264 \\
0.2887 & 0.4928 & 0.1190 & 0.8122 \\
0.3033 & -0.6652 & -0.5674 & 0.3790 \\
0.2173 & -0.5253 & 0.8136 & 0.1222
\end{pmatrix} \\
S &= \begin{pmatrix}
23.33 & 0 & 0 & 0 & 0 & 0 \\
0 & 9.76 & 0 & 0 & 0 & 0 \\
0 & 0 & 5.03 & 0 & 0 & 0 \\
0 & 0 & 0 & 3.27 & 0 & 0
\end{pmatrix} \\
V^T &= \begin{pmatrix}
0.2638 & 0.6627 & 0.4237 & 0.4293 & 0.3549 & 0.0373 \\
0.2850 & 0.6018 & -0.6079 & -0.3061 & -0.2171 & -0.2151 \\
-0.0385 & 0.1948 & 0.1425 & 0.1162 & -0.7138 & 0.6460 \\
0.7038 & -0.1795 & 0.3700 & -0.5590 & 0.0308 & 0.1491 \\
0.5587 & -0.3294 & -0.1526 & 0.6077 & -0.3198 & -0.2965 \\
-0.2090 & 0.1411 & 0.5201 & -0.1635 & -0.4629 & -0.6519
\end{pmatrix}
\]

1. [3pt] Suppose we are willing to sacrifice up to a maximum of 10% of the total variance in the data, then what is the least number of dimensions we need to keep? Explain how you arrived at your answer.

   a) If we take one dimension the loss is \( \frac{23.33^2 - 23.33^2 + 9.76^2 + 5.03^2 + 3.27^2}{23.33^2 + 9.76^2 + 5.03^2 + 3.27^2} \approx 0.194 \text{ or } 19.4\% \).

   b) If we take two dimensions the loss is \( \frac{23.33^2 + 9.76^2 + 5.03^2 + 3.27^2}{23.33^2 + 9.76^2 + 5.03^2 + 3.27^2} \approx 0.05331 \text{ or } 5.3\% \). Since we want less than 10\% loss, we need 2 dims.

2. [4pt] Suppose we decided to just keep top two most important dimensions after the LSI analysis. Draw a bounding box around the parts of U,S,V matrices above that will be retained after this decision. [You answer this question by directly marking the matrices above]

   [Matrix shown with 2-D reconstruction data loss as 0.05331]

   Compare this to original T-D Matrix

\[
\begin{pmatrix}
5.97 & 14.79 & 7.54 & 8.2 & 6.9 & 0.3 \\
3.14 & 7.36 & -0.07 & 1.4 & 1.3 & -0.8 \\
0.01 & 0.77 & 6.94 & 5.0 & 6.3 & 9.1 \\
-0.12 & 0.27 & 5.26 & 3.7 & 2.9 & 1.29
\end{pmatrix}
\]
(3) [6pt] Suppose the two most important dimensions after LSI are called \( f_1 \) and \( f_2 \) respectively. Plot the six documents as points in the factor space (use the plot below). (It is okay if you put the points in the rough place they will come; no need to spoil your eyesight counting all the small grid lines). Comment on the way the documents appear in the plot—is their placement related in any rational way to their similarity you would intuitively attach to them?

\[ x \cdot d = (0.263, 0.785) \]
\[ \text{scaled by (23.33, 24.76)} \]
giving 6.15, 2.78

(4) [5pt] What is the vector space similarity between D5 and D6 before and after the LSI transformation (assume, in the latter case, that we are using the top two dimensions). Is the change intuitively justified?

Before

\[
\begin{bmatrix}
7 & 1 & 60 \\
0 & 0 & 4
\end{bmatrix}
\]

Normalization factor

= 0

(Since d5, d6 have no common terms, their similarity is zero)

After

\[
\begin{bmatrix}
1 \cdot 6.15 & 2.78 \\
15.4 & 5.87 \\
6.88 & 5.93 \\
10 & 2.9 \\
8.3 & 2.1 \\
0.8 & 2.1
\end{bmatrix}
\]

Pretty high similarity. LSI allows us to see that the documents are similar even though they don't share words.
(3) [6pt] Suppose the two most important dimensions after LSI are called \( f_1 \) and \( f_2 \) respectively. Plot the six documents as points in the factor space (use the plot below). (It is okay if you put the points in the rough place they will come; no need to spoil your eyesight counting all the small grid lines). Comment on the way the documents appear in the plot—is their placement related in any rational way to their similarity you would intuitively attach to them?

This is what the plot looks like if you plotted \( f_d \) without the scaling with \( f_f \). The other plot essentially elongates x-axis 2 times more than y-axis (233 vs. 976).

had the ratio between \( \lambda_1 \) \& \( \lambda_2 \) been much higher, things would start looking way different in the two plots.

(4) [5] What is the vector space similarity between D5 and D6 before and after the LSI transformation (assume, in the latter case, that we are using the top two dimensions). Is the change intuitively justified?

\[
\begin{align*}
\text{before} & \quad [7 \ 160] \cdot [0 \ 0 \ 0 \ 4] = 0 \\
& \quad \frac{10511061}{10511061} \\
& \quad \text{no common words between D5 \& D6 so no similarity}
\end{align*}
\]

\[
\begin{align*}
\text{after} & \quad \text{if we use } f_d \text{ without } f_f \text{ scaling} \\
& \quad (0.35 \ -0.21) \cdot (0.03 \ -0.21) \\
& \quad \frac{\sqrt{0.35^2 + 0.21^2}}{\sqrt{0.03^2 + 0.21^2}} \\
& \quad = 0.627
\end{align*}
\]
Qn IV Suppose you have a set of documents that basically contain only one key word, repeated multiple times. Suppose the dissimilarity between the documents is judged in terms of the difference in frequency of occurrence of that single keyword. The documents are given by:

D1: 5
D2: 7
D3: 9
D4: 14
D5: 15

(a)[5pt] Suppose we want to use K-Means algorithm to cluster this data into 2 clusters. Show how the clustering progresses, if you start K-means off with D3 and D4 as the seeds. What is the cumulative intra-cluster dissimilarity measure for the final clustering?

Iteration 1

D1 5
D2 7
D3 9

C1

C2

D4 14
D5 15

Center of C1: \( \frac{9 + 7 + 5}{3} = \frac{21}{3} = 7 \)

Center of C2: 14.5

Re-cluster with 7 and 14.5 as the cluster centers

No change. So stop

\( \text{Dissimilarity} = (5-7)^2 + (7-7)^2 + (9-7)^2 + (14-14.5)^2 + (15-14.5)^2 \)

(b)[2pt] Suppose we are allowed vary K, the number of clusters that K-means looks for. What is the lowest intra-cluster dissimilarity measure that can be achieved this way? When will it happen?

As we discussed in class, if we care only about intra-cluster dissimilarity and are allowed to look for any number of clusters, then the best is each element in its own cluster.

So 5 clusters. Dissimilarity = 0!
Qn V. Short answer questions. Except for the first question, all other questions carry 3 points.

1. [5pt] Suppose the number of keywords (size of vocabulary) is V, the average length of a document (in terms of words) is N, the number of documents in the corpus is M, the average length of a query is Q, and the average number of documents in which a query word appears is B. What is the time complexity, in vector-space retrieval, of: (a) Naïve query processing (without inverted index) and (b) query processing with inverted index. Why is (b) better?

Naïve
for each keyword, you look at each doc (and the query)

\[ V \times M \]

#Keywords #docs.

Inverted index
for each query word, you just look at docs that contain it

\[ Q \times B \]

#words in query #docs in which a query word appears

Since \( Q \ll V \) and \( B \ll M \) in general:

\[ V \times B \ll V \times M \]

2. [3] In the class, I mentioned that one way of making A/H computation more stable is to define the page importance in terms of subspaces rather than eigen vectors of the adjacency matrix. Explain how/why this is supposed to help. (Short answer in terms of examples is enough)

The idea is that subspaces spanned by the eigen vectors may remain stable w.r.t. changes to the adjacency matrix even though the eigen vectors may change quite drastically.

In the figure on the right, both \( v_1, v_2 \) and \( v_1', v_2' \) span the same subspace (the plane of this paper).
3. In the class, someone asked why Google doesn't remove all sink nodes (i.e., nodes that do not have any outlinks) from the page graph altogether before computing the page rank. What useful capability of Google will be lost if this were to be done?

If you do that, you may be removing important pages (see p in the picture on left) that have probably not yet been crawled.

(Google can return a page that it never crawled—see the white house URL in the Google Paper)

4. We talked about "stemming" as a technique that many text retrieval systems use. Comment on how stemming affects the precision and recall (i.e., improve/worsen).

Stemming increases recall but can reduce precision (some of the returned pages may be less relevant).

Eq. Query = "hostage killed" stemmed

query: "hostage kill"

But I am looking for

Google basically does not use stemming

5. The HITS analysis assumes that all outlinks on a page are relevant to the given query. In many cases, however, even pages that are among the top K in terms of their similarity to a query Q, may have links to pages that have nothing to do with that particular query. Give a technique that can offset this problem
6. [3] Give one good reason why we shouldn't replace cosine-metric with the inverse of euclidean distance (between query and document vectors) for deciding query-document similarity.

Euclidean distance is sensitive to magnitudes of the vectors. In the example on right, Euclidean distance will say $d_1$ is closer to $d_1$ rather than $d_2$. If we normalize the vectors beforehand, then this argument won't hold. But still inverted Euclidean distance is not well related to relative proportion of drift words.

7. [3] List four (4) magic parameters that Google uses (a magic parameter is a number that needs to be set by Brin & Page—or their underlings).

1) The weight for combining Page Rank and Similarity with Query & Rank + β similarity

2) The weight for words appearing in different parts of the page (anchor, header, ...)

3) The weight for deciding when the surfer follows the link on page and when surfer does something random: $P^{New} = C*P + (1-C)R$

4) # occurrences of a word beyond which the tf weight won't change