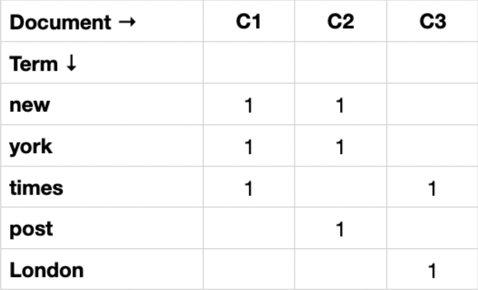
Terry Shipman
tshipm5

Homework 4
CSC 3730 Machine Learning

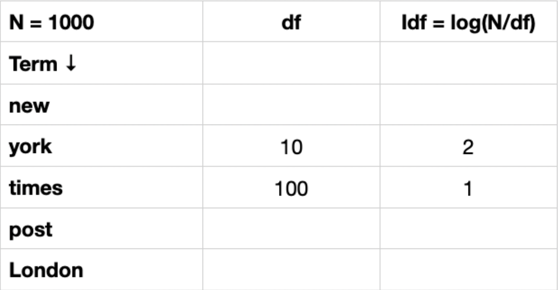
11-24-2020

Problem 1: [15 pt] Web Search and Text Mining

Consider the following newspaper collection C = C1, C2, C3 (given as one document per line):
C1: new york times
C2: new york post
C3: London Times



Document →	C1	C2	C3
Term ↓			
new	1	1	
york	1	1	
times	1		1
post		1	
London			1



N = 1000	df	Idf = log(N/df)
Term ↓		
new		
york	10	2
times	100	1
post		
London		

a) [5 pt] What is the table that describes the information of term frequencies for each of the three documents? A matching score for a query (q) and a document (d) can be computed as $\Sigma_{t \subset q \cap d} tf_{t,d}$ Given a query "york times", what are the matching scores for each of the three documents?

C1: $1 + 1 = 2,$ **C2:** $1 + 0 = 1,$ **C3:** $0 + 1 = 1$

b) [5pt] Suppose there are N=1000 documents in the whole newspaper collections. The word "times" occurs in 100 of those and "York" appears in 10 of those respectively. A matching score for a query (q) and a document (d) can be computed as $\Sigma_{t \subset q \cap d} tf_{t,d}$ Given a query "york times", what are the matching scores for each of the three documents?

C1: $(1 \times 2) + (1 \times 1) = 3,$ **C2:** $(1 \times 2) + (0 \times 1) = 2,$ **C1:** $(0 \times 2) + (1 \times 1) = 1$

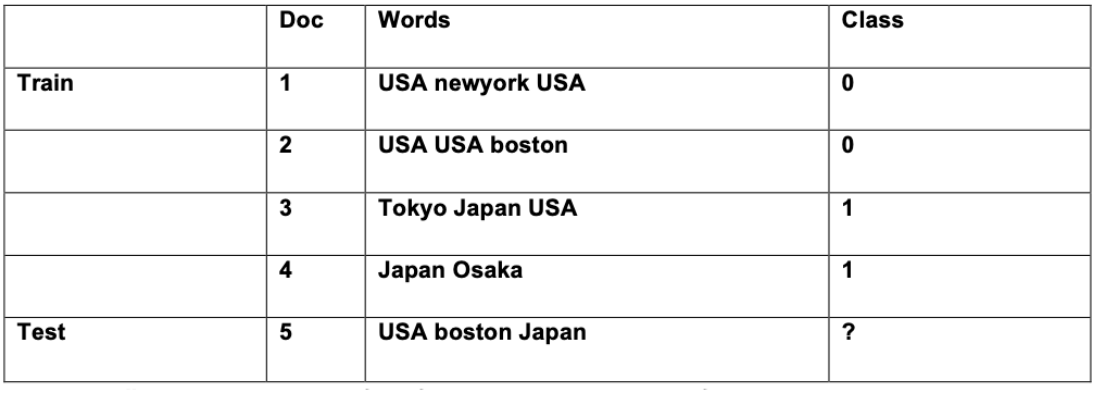
c) [5 pt] For a query, search engine G returns a ranked list C1>C2>C3. Search engine Y returns a ranked list C1>C3>C2. A human judge evaluates the results and labels C1 and C2 as relevant documents. What are the Precision @K (K=1, 2, 3) and mean average precision (MAP) scores for the two engines? Which engine is better?

Search Engine G: Precision @K: Precision @1: $1/1$ Precision @2: $2/2$ Precision @3: $2/3$
MAP = $\frac{1}{2}(1/1 + 2/2) \ = 1$

Search Engine Y: Precision @K: Precision @1: $1/1$ Precision @2: $1/2$ Precision @3: $2/3$
MAP = $\frac{1}{2}(1/1 + 2/3) \ = 5/6$

Search Engine G would be the better option.

Problem 2: [10 pt] Naïve Bayes Document Classification



	Doc	Words	Class
Train	1	USA newyork USA	0
	2	USA USA boston	0
	3	Tokyo Japan USA	1
	4	Japan Osaka	1
Test	5	USA boston Japan	?

Use Naïve bayes classifier for document classification. A test sample may contain a word that is not present in the dictionary and the P(word|class) becomes 0. To mitigate this issue, one solution is to employ Laplace smoothing with smoothing parameter “α”. Set “α =1” (add-one smoothing). Compute the probability of test document (doc5) belonging to each of the two classes.

$$\hat{P}(c=0) = \tfrac{2}{4} \quad \hat{P}(c=1) = \tfrac{2}{4} \qquad \hat{P}(w|c) = \tfrac{count(w,c)+1}{count(c)+|V|} \qquad \alpha = 1$$

Class 0:

$$P(USA|c_0) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{4+1}{6+3} = 5/9 \qquad P(boston|c_0) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{1+1}{6+3} = 2/9$$

$$P(Japan|c_0) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{0+1}{6+3} = 1/9$$

Class 1:

$$P(USA|c_1) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{1+1}{5+4} = 2/9 \qquad P(boston|c_1) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{0+1}{5+4} = 1/9$$

$$P(Japan|c_1) = \frac{n_k+\alpha}{n+\alpha|V|} = \frac{2+1}{5+4} = 3/9$$

Class probability for document 5:

$$P(c_0|d_5) \propto \frac{2}{4} \times \frac{5}{9} \times \frac{2}{9} \times \frac{1}{9} \approx 0.0069 \qquad P(c_1|d_5) \propto \frac{2}{4} \times \frac{2}{9} \times \frac{1}{9} \times \frac{3}{9} \approx 0.0041$$

Based on the class probability, there is a better chance of document 5 belonging to class 0.

Problem 3: [10 pt] Page Rank

Given a directed graph G = (V;E) with V = 1; 2; 3; 4; 5, and E = (1→2); (1→3); (2→1); (2→3), (3→4); (3→5); (4→5); (5→4). Assume that with probability 0.7, a surfer follows a link at random. With probability 0.3, the surfer jumps to some random page.

a) [5pt] What is the page rank equation for G?

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n} = \sum_{1 \to 5} 0.7 \frac{r_i}{d_i} + 0.3\frac{1}{5} = \sum_{\mathbf{1 \to 5}} \mathbf{0.7 \frac{r_i}{d_i} + 0.06}$$

b) [5pt] Assume the initial page rank for each of the five nodes is 0.2, compute the page rank vector using power iteration algorithm with tolerance =0.0001. What is the page rank vector after one iteration? What is the page rank vector when the algorithm converges?

$OutDegree_1 = 2$	$OutDegree_4 = 1$	$r_j = 0.20$
$OutDegree_2 = 2$	$OutDegree_5 = 1$	$\sum_{1 \to 5} \frac{r_i}{d_i}$
$OutDegree_3 = 2$	$\varepsilon = 0.0001$	

	$node_1$	$node_2$	$node_3$	$node_4$	$node5$
I_0	0.2	0.2	0.2	0.2	0.2
I_1	0.13	0.13	0.20	0.27	0.27
I_2	0.1055	0.1055	0.1510	0.3190	0.3190
I_3	0.0969	0.0969	0.1339	0.3362	0.3362
I_4	0.0939	0.0939	0.1278	0.3422	0.3422
I_5	0.0929	0.0929	0.1257	0.3433	0.3433
I_6	0.0925	0.0925	0.1250	0.3450	0.3450
I_7	0.0924	0.0924	0.1248	0.3452	0.3452
I_8	0.0923	0.0923	0.1247	0.3453	0.3453
I_9	0.0923	0.0923	0.1246	0.3454	0.3454
I_{10}	0.0923	0.0923	0.1246	0.3454	0.3454

$$Page\ rank\ vector_{i=1} = \begin{pmatrix} 0.13 \\ 0.13 \\ 0.20 \\ 0.27 \\ 0.27 \end{pmatrix} \qquad Page\ rank\ vector_{Convergence} = \begin{pmatrix} 0.0923 \\ 0.0923 \\ 0.1246 \\ 0.3454 \\ 0.3454 \end{pmatrix}$$

Problem 4: [15 pt] Graph Visualization

Analyze the TMDb data in the files below, as an undirected graph (network).

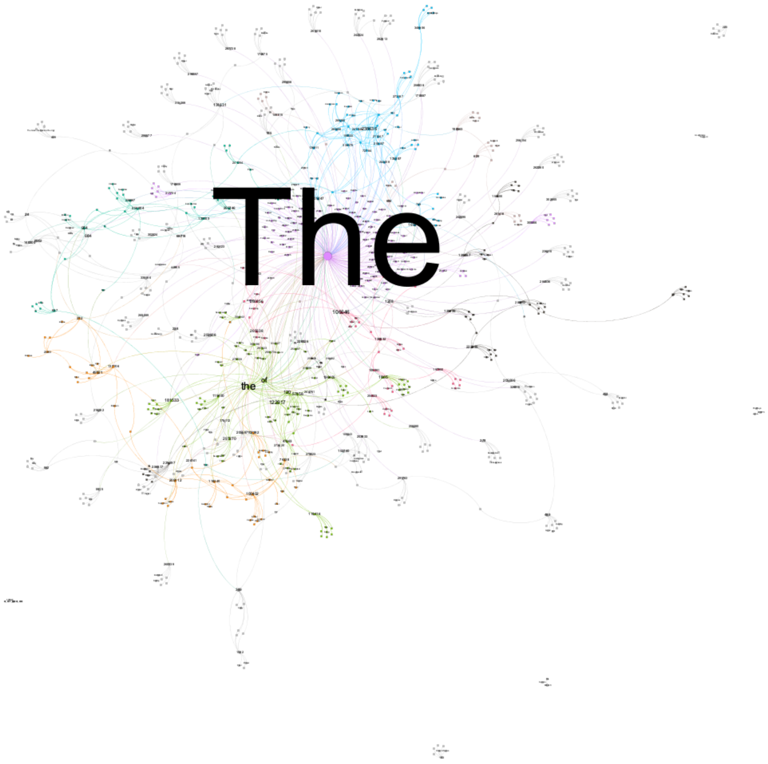
Each movie in movie_ID_name_gephi.csv is represented as a node in the graph and each movie pair in movie_ID_sim_movie_ID_gephi.csv represents an undirected edge. You will visualize the graph of similar movies using Gephi (free at http://gephi.org). Note: Make sure your system fulfils all requirements for running Gephi.

Import all the edges in movie_ID_sim_movie_ID_gephi.csvby checking the "create missing nodes" option, since many of the IDs in movie_ID_sim_movie_ID_gephi.csv may not be present in movie_ID_name_gephi.csv. Note that in some cases, the data would need to be imported using the data lab rather than File→Open. Gephi quick start guide: https://gephi.org/users/quick-start/

a. [5 pt] Visualize the graph. Submit a snapshot of a visually meaningful view of this graph and a description of your choice of visualization Here are some general guidelines for a visually meaningful graph:

- Keep the edge crossing to minimum.
- Keep the edge length roughly the same.
- Try and avoid very long edges.
- Keep the edges as straight lines without any bends.
- Keep the graph compact and symmetric if possible.
- All the nodes and edges should be properly visible.
- Make it easy to visualize which nodes any given node is connected to and also count its degree.

Experiment with Gephi's features, such as graph layouts, changing node size and color, edge thickness, etc. The objective of this task is to familiarize yourself with Gephi and hence is a fairly open ended task.

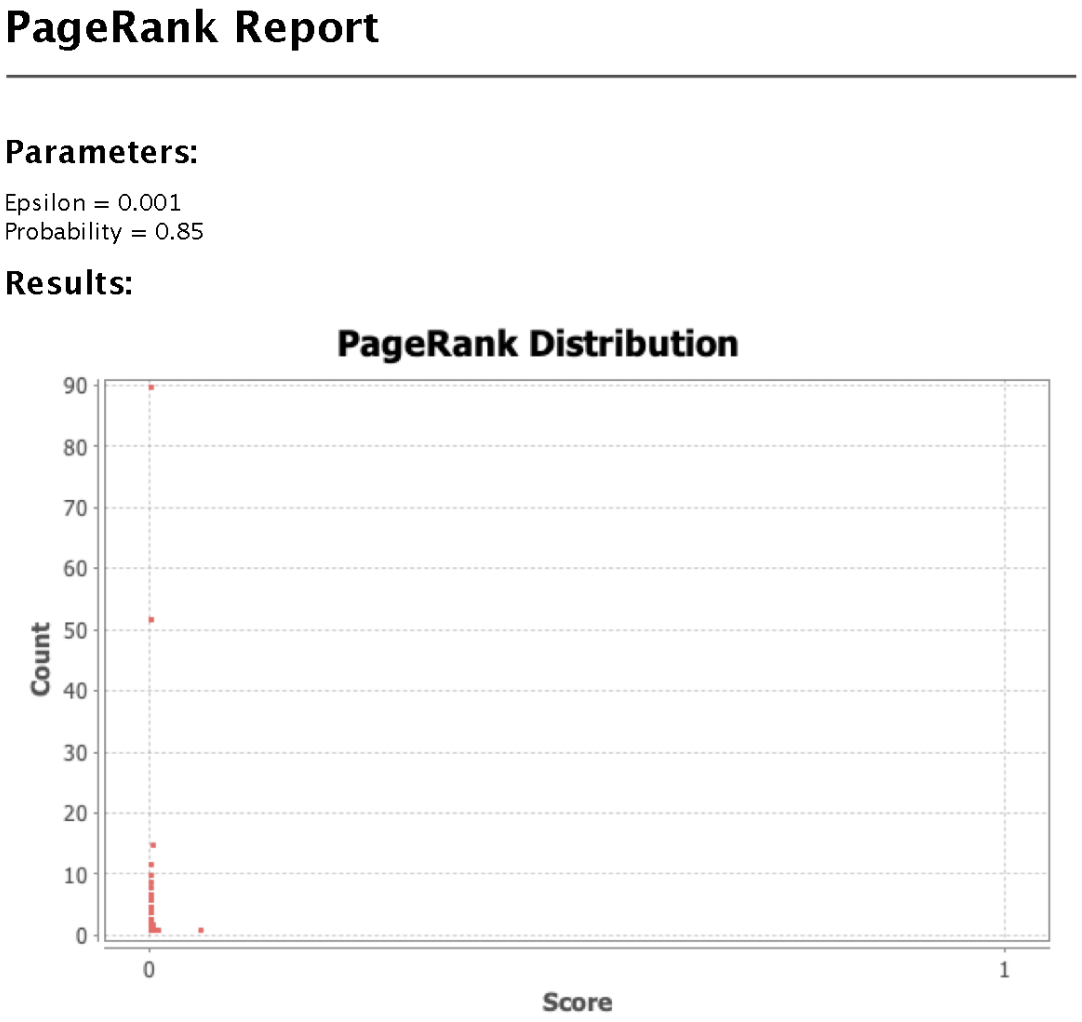


b. [5 pt] Using Gephi's built-in functions, compute and report the following metrics for your graph:

- Average node degree = 2.623
- Diameter of the graph = 8
- Average path length = 4.087

Briefly explain the intuitive meaning of each metric in your own words. (These metrics will be discussed in the lectures on graphs.)

The average node degree can let the user know how many connections each node as to other nodes. The graph diameter allows for the user to know the actual size of the graph. The average path length allows the user to know how close adjacent nodes are to a particular node.
c. [5 pt] Run Gephi's built-in PageRank algorithm on your graph.



PageRank Report

Parameters:

Epsilon = 0.001
Probability = 0.85

Results:

- Submit **an image** showing the distribution of the PageRank score.

(The "distributionïs generated by Gephi's built-in PageRank algorithm where the horizontal axis is the PageRank score and the vertical axis is the number of nodes with a specific PageRank score)

- List **the top 5 movies**(movie title and id) with the highest PageRank score.