Information Retrieval
INFO/CS 4300

- Instructor: Chris Buckley
  - Office hours Wednesdays 11am Gates 231
  - Office hours next week moved to Friday 11am
- Piazza will be the main communication tool
  - https://piazza.com/cornell/fall2014/info4300/home
  - Lecture notes will appear there.
  - TA office hours and locations appear there.

Course Admin

- Critique 1, Homework 1, Homework 2, Critique 2, Project 1, Homework 3 – Graded, grades available on-line through CMS, hard copy can be picked up in the homework return room in Gates 216, open Mon-Fri 12-4.
- Project 2, Critique 3 due today!
  - Late penalty of 5%/day, max 20%
- My office hours next week moved to Friday 11am.
- Check CMS and make sure we have graded all assignments that you have turned in!

Notes for Practice Exam

- These are my notes only
  - They are incomplete
    - I do not plan to expand on them in Piazza!
    - Ask your questions now!
  - They may not be completely correct in terms of arithmetic.

Question 1

- (1) Explain how vector space concepts can be used to calculate the similarity between two documents.
- Answer should/could address the following:
  - representation of document as a vector: elements indicate the frequency with which distinct words in the vocabulary occur in the document;
  - term weighting
  - similarity function (angle between vectors:cosine)

Question 2

(2) You have a collection of documents that contain the following index terms:

D_1: alpha bravo charlie delta echo foxtrot golf
D_2: golf golf golf delta alpha
D_3: bravo charlie bravo echo foxtrot bravo
D_4: foxtrot alpha alpha golf golf delta

(a) Show the term-document frequency matrix for the collection. Calculate the similarity between each pair of documents using just the term frequency weighting and an inner product similarity function.

(b) Show the term-document weight matrix for the collection, using weights that are proportional to the term frequency and inversely proportional to the document frequency (use raw-tf * 1/df)

(c) Calculate the inner product similarity between each pair of documents based on the weight matrix.

2a Frequency matrix

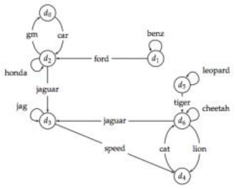
	D1	D2	D3	D4
ALPHA	1	1	0	2
BRAVO	1	0	3	0
CHARLIE	1	0	1	0
DELTA	1	1	0	1
ECHO	1	0	1	0
FOXTROT	1	0	1	1
GOLF	1	3	0	2

2b WEIGHT MATRIX

	D1	D2	D3	D4
ALPHA	1/3	1/3	0	2/3
BRAVO	1/2	0	3/2	0
CHARLIE	1/2	0	1/2	0
DELTA	1/3	1/3	0	1/3
ECHO	1/2	0	1/2	0
FOX	1/3	0	1/3	1/3
GOLF	1/3	3/3	0	2/3

Sim(D1 & D2) = 1/3*1/3 + 1/3*1/3 + 1/3*3/3
Sim (D1 & D3) = 1/2*3/2 + 1/2*1/2 + 1/2*1/2 + 1/3*1/3
Sim (D1 & d4)=1/3*2/3 + 1/3*1/3 + 1/3*1/3 + 1/3*2/3
Sim (D2 & D3)=0
Sim (D2 &D4)=1/3*2/3 + 1/3*1/3 + 3/3*2/3
Sim (D3 & D4)=1/3*1/3

PageRank

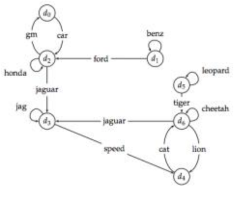


► Figure 21.4 A small web graph. Arcs are annotated with the word that occurs in the anchor text of the corresponding link.

Consider the web graph above (from the MRS book). With teleportation rate α=.14 (this is the same as the rate of clicking the "surprise me" button), the PageRank vector was determined to be:

(d_0, d_1, d_2, d_3, d_4, d_5, d_6) = (0.05, 0.04, 0.11, 0.25, 0.21, 0.04, 0.31),

giving page d_6 the highest rank, and page d_3 the second highest rank. Use intuition from the behavior of the random web-surfer to indicate whether the relative rank of d_3 and d_6 is (changed, unchanged) for each of the following actions applied to the above graph*:



► Figure 21.4 A small web graph. Arcs are annotated with the word that occurs in the anchor text of the corresponding link.

-d0(d2) = .02 + .86 * (.11/3)
-d1(d1) = .02 + .86 * (.04/2)
-d2(d0d1d2) = .02 + .86 * (.05/1+ .04/2 + .11/3)
-d3(d2d3d6) = .02 + .86 * (.11/3 + .25/2 + .31/3)
-d4(d3d6) = .02 + .86 * (.25/2 + .31/3)
-d5(d5) = .02 + .86 * (.04/2)
-d6(d4d5d6) = .02 + .86 * (.21/1 + .04/2 + .31/3)

(i) A link is added from d1 to d3: no
.d3(d1d2d3d6)
~d = (0.048 0.028 0.097 0.256 0.221 0.035 0.315)

(ii) The link from d6 to itself is removed: yes
.d6(d4d5)
~d = (0.052 0.035 0.112 0.276 0.244 0.035 0.245)

(iii) The link from d3 to d4 is removed: yes
.d4(d6)
~d = (0.052 0.035 0.112 0.601 0.052 0.035 0.112)

(iv) The link from d4 to d6 is removed: yes
.d6(d5d6)
~d = (0.108 0.073 0.232 0.240 0.174 0.073 0.102)

(v) A link is added from d4 to d3: yes
.d3(d2d3d4d6)
~d = (0.052 0.035 0.112 0.355 0.226 0.035 0.185)

Search Engine Architecture

Document filtering is an application that stores a large number of queries or user profiles and compares these profiles to every incoming document on a feed. Documents that are sufficiently similar to the profile are forwarded to that person via email or some other mechanism. Describe the architecture of a filtering engine and how it may differ from a search engine.

Answers should/could include:
Store queries, not documents
Inverted list of query terms
Don't rank queries, have threshold of similarity with each
Must keep track of df of each term explicitly (not available via inverted list)
Query at a time processing

Word Distributions

(1) Assuming a Zipfian distribution and a vocabulary size of 10, what is the relative frequency (with respect to the full vocabulary) of the most frequent word in the vocabulary?

(2) Recall Heaps' Law, $M = kT^b$ and assume parameters: k=50 and b = 0.5. Consider a blog corpus with 10,000 archived postings and 1,000 newly received postings. How many new words are likely to appear in the newly received part compared to the archive?

Word distributions

1. 1 / (1 + ½ + 1/3 + ¼ + 1/5 + 1/6 + 1/7 + 1/8 + 1/9 + 1/10) = 34%

2. Assume each posting is L words long

$$\frac{M'}{M} = \frac{50 \ * (11000 * L)^{.5}}{50 \ * (10000 * L)^{.5}} = \sqrt{\frac{11}{10}}$$

If L = 100, 244 words added

Metrics

(1) Define the terms recall and precision.

(2) Q is a query. D is a collection of 1,000,000 documents. When the query Q is run, a set of 200 documents is returned.

(a) How in a practical experiment would you calculate the precision?
(b) How in a practical experiment would you calculate the recall?

(3) Suppose that, by some means, it is known that 100 of the documents in D are relevant to Q. Of the 200 documents returned by the search, 50 are relevant.

(a) What is the precision?
(b) What is the recall?

(4) Explain what is meant by a "precision–recall graph", and how it is used to evaluate an information retrieval system.

(4) A "precision-recall graph", (aka recall-precision graph) with ranked retrieval, plots for each of the first k ranked items, the precision (fraction of the k returned items that are relevant to query) vs. the recall (number of the k returned items relevant divided by the total number of relevant items in the corpus). Usually the precision is interpolated so that its value is the maximum of future values of recall.

Clustering

Consider the following five "documents":
doc1 = (cornell cayuga bigred bears)
doc2 = (cornell cornell nyc bigred bigred)
doc3 = (cornell bells ring everyday)
doc4 = (cornell cornell cold ring ring)
doc5 = (bigred bigred bells ring everyday very cool)

(1) Using the cosine similarity measure (vector dot product with document vectors normalized to unit length) --- just use the raw term frequencies for the document vector, i.e., no log tf or idf weights) --- determine the 5**x**5 similarity matrix for the above five documents.
[Note: the sum of the squares of the vector components should sum to 1 for unit length, so, e.g., the first document vector would be something like (1, 1, 1, 1, 0, . . .)/2]

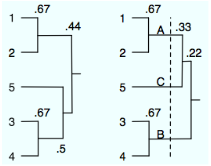
vocabulary = (cornell cayuga bigred bears nyc bells ring everyday cold very cool)
doc1=(1 1 1 1 0 0 0 0 0 0 0)/2
doc2=(2 0 2 0 1 0 0 0 0 0 0)/3
doc3=(1 0 0 0 0 1 1 1 0 0 0)/2
doc4=(2 0 0 0 0 0 2 0 1 0 0)/3
doc5=(0 0 2 0 0 1 1 1 0 1 1)/3
similarity matrix

	1	2	3	4	5
1	1.00	0.67	0.25	0.33	0.33
2		1.00	0.33	0.44	0.44
3			1.00	0.67	0.50
4				1.00	0.22

(a) Cluster these elements using the **single linkage** method. Express your results as a dendrogram, being careful to indicate the value of the similarity at each join.

(b) Cluster these elements using the **complete linkage** method. Express your results as a dendrogram, being careful to indicate the value of the similarity at each join.

(2.2a) Single linkage similarities:
(1,2)=.67; (3,4)=.67; (5,(3,4))=(3,5)=.5; ((3,4,5),(1,2))=(2,4)=(2,5)=.44
(2.2b) Complete linkage similarities:
(1,2)=.67; (3,4)=.67; (5,(1,2))=(1,5)=.33; ((3,4),(1,2,5))=(4,5)=.22



(3) Now consider the following test "document":

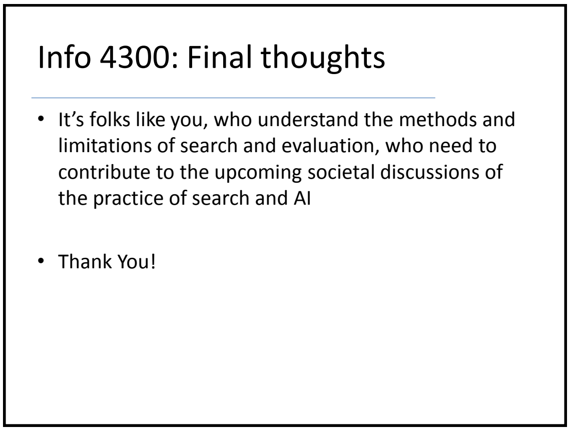
test1 = (cornell cornell cornell bears bears nyc cayuga cool)

Determine to which of the above three clusters A,B,C it belongs, using
(a) kNN (k-nearest neighbors) for k=1
(b) kNN (k-nearest neighbors) for k=3
(c) the centroid method (using cosine similarity(?) to calculate centroids of clusters: remember to calculate the centroids using the document vectors before normalizing to unit length)

(2.2c) test=(3 0 1 2 1 0 0 0 0 1 0)/4
(2.3a) cosine similarities to doc1–5 are (.75 .583 .375 .5 .083) so 1NN is A
(2.3b) 3NN gives (AAB) so cluster A
(2.3c) cosine similarities to centroid clusters A,B,C are (.71 .49 .08), so A

Info 4300: Final Thoughts

- Information Retrieval is a very broad area
  - Primary focus within it is changing from "document retrieval" to "information retrieval"
- The core of IR is evaluation
  - The activities of IR are "AI complete"
    - If we knew how to do IR from scratch, we would have AI
    - We don't know or understand enough, therefore rely on evaluation



Info 4300: Final thoughts

- It’s folks like you, who understand the methods and limitations of search and evaluation, who need to contribute to the upcoming societal discussions of the practice of search and AI
- Thank You!