CS 276B Problem Set #1 Solutions

#1 Search Engines

Pick a search need for each of the following types defined in class:

1. Informational
2. Transactional
3. Navigational.

Try each of these needs on 3 distinct public web search engines.

From your observations on the top 5 hits returned by each engine, write down the factor(s) you think weigh the heaviest in the ranking algorithm for that engine. You may find it useful to view the HTML source of the retrieved pages. What we're looking for: good analysis based on your observations, rather than deciphering completely what the engine is doing under the hood.

Answers vary…

#2 Collaborative Filtering

Consider the following ratings judgments for 5 users and 6 products. We are trying to recommend a movie to UA from the set of movies he hasn't rated (D, E or F).

	User				
Item	U1	U2	U3	U4	UA
A	8		10		10
B		2		3	3
C	2	8		4	1
D	8			2	
E		7	3		
F	5		8	1	

a. Using the version of the GroupLens collaborative filtering scheme shown on slide 32, work out an ordered list of recommendations for User A (hint: use a spreadsheet!). To do this, you need to make a couple of things concrete in the algorithm shown there:
i) Exactly what value do you give to z_{iq} in each case?
ii) What value do you give to σ_a?

Show your calculations, and the final ordered list of recommendations.

The value of z_{iq} is defined by the algorithm.

The value of σ_a has no effect on the ordering of recommendations (as long as it's positive), so you just want to pick a value that will result in reasonable predicted ratings

(0 <= p_{aq} <= 10). One way to do this is to calculate p_{aq} for items that UA has already rated and then solve for σ_a. If you average the two reasonable values, it yields σ_a = .04.

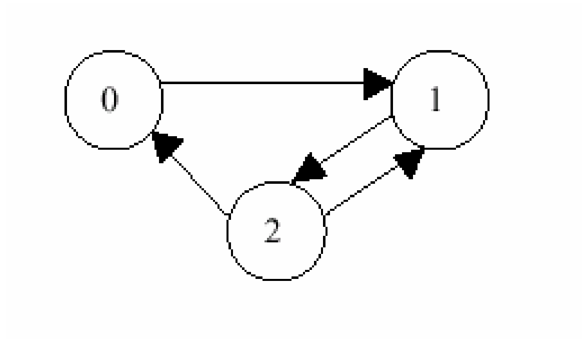
Running the algorithm, the final ordering from best to worst is D, F, E.

b. Suppose that you remove the normalization of z_{iq} via each user's mean vote. Does this change the ordered list of recommendations? (If so, does it seem better or worse or unclear?)

Now the ordering is F, D, E. It's unclear if this is better or worse – A is pretty similar to all three users who rated D and/or F. Their mean rating for F is only 4.7, but the mean rating for D is only 5.

#3 Markov Chains

(a) Represent the following simplified graph of the web as a Markov chain by providing the corresponding transition probability matrix. Assume teleportation (jumping to any page in the graph with uniform probability) to a random page (including the start page) occurs with **50%** probability.



$$0.5 \text{ x} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.5 & 0.5 & 0 \end{bmatrix} + 0.5 \text{ x} \begin{bmatrix} 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 \end{bmatrix} = \begin{bmatrix} 0.167 & 0.667 & 0.167 \\ 0.167 & 0.167 & 0.667 \\ 0.417 & 0.417 & 0.167 \end{bmatrix}$$

(b) Using the initial probability vector [0 1 0], carry forward the Markov chain 1 time step. (That is, give the probability vector for time t = 1.)

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \text{ x} \begin{bmatrix} 0.167 & 0.667 & 0.167 \\ 0.167 & 0.167 & 0.667 \\ 0.417 & 0.417 & 0.167 \end{bmatrix} = \begin{bmatrix} 0.167 & 0.167 & 0.667 \end{bmatrix}$$

#4 LR Wrappers

Note that the problem set had an unfortunate typo – part of the line for Jennifer Widom was missing. We won't count this question.

Here is a small (valid!) HTML document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
  <title>People</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  </head>

<body>
<h1>People</h1>

<table>
<tr><th><b>Name</b></th><th>Office></th><th>Phone</th><th>Mail</th></tr>
<tr><td>Chris Manning</td><td>418</td><td>3-7683</td><td>9040</td></tr>
<tr><td>Teg Grenager</td><td>454</td><td>1-2345</td><td>9040</td></tr>
<tr><td>Hector Garcia-Molina</td><td>276</td><td>3-9745</td><td>9025</td></tr>
<tr><td>Jennifer Widom</td><td>422</td><td>5-4321</td><td>9040</td></tr>
</table>
</body>
</html>
```

a. Suppose one wanted to write individual LR (Kushmerick Left Right) context wrappers for each of the fields Name, Office, Phone, Mail. Which ones can you do it for, and which ones can you not? If you can do it, give a LR wrapper that would work, and if you cannot, briefly explain why.

Name: L: <tr><td> R: </td> works
Office: can't distinguish office and phone
Phone: can't distinguish office and phone
Mail: L: <td> R: </td></tr> works

b. Suppose you enhanced the LR wrapper framework so that they also did a regular expression match on the field content. Now, which fields can you do it for, and which ones can you not? If you can do it, give a LR wrapper that would work, and if you cannot, briefly explain why.

Name: as above (filler can be .)*
Office: L: <td> F: [0-9]+ R: </td><td> (note that you need the trailing <td> to differentiate from Mail)

Phone: L: <td>   F:[0-9]-[0-9]+   R: </td>
Mail: as above

c. Alternatively (i.e., not using a regular expression match on field content, suppose that one had a notion of relation, and assumed that fields were ordered in the wrapper for a relation. Under the assumption of a known ordering, which fields can you do it for, and which ones can you not? If you can do it, give a LR wrapper that would work, and if you cannot, briefly explain why.

Assume that relation field order is Name, Office, Phone Mail. Then, each wrapper could just be L: <td> R:</td>, though this is maximally dangerous (any inconsistency, and you could get arbitrarily out of sync). Continuing to check for a match on <tr> and </tr> for Name and Mail would be safer.