COS 435, Spring 2014 - Problem Set 4

Due at 1:30PM, Wednesday, April 2, 2014.

Collaboration and Reference Policy

You may discuss the general methods of solving the problems with other students in the class. However, each student must work out the details and write up his or her own solution to each problem independently.

Some problems have been used in previous offerings of COS 435. You are NOT allowed to use any solutions posted for previous offerings of COS 435 or any solutions produced by anyone else for the assigned problems. You may use other reference materials; you must give citations to all reference materials that you use.

Lateness Policy

A late penalty will be applied, unless there are extraordinary circumstances and/or prior arrangements:

- Penalized 10% of the earned score if submitted by 11:59 pm Wed. (4/2/14).
- Penalized 25% of the earned score if submitted by 4:30pm Friday (4/4/14).
- Penalized 50% if submitted later than 4:30 pm Friday (4/4/14).

Problem 1 (part of a 2011 exam 2 problem)

Recall that skip pointers can be used to speed up query evaluation by allowing the algorithm that executes the intersection of postings lists for the different terms of the query to skip sections of a postings list when then next document on one of the other postings list has a much higher docID. This question asks you to estimate the savings in space if the skip pointer representation is combined with the compressed representation of docIDs using gaps.

Assume that a list containing L postings uses floor(sqrt(L))-1 skip pointers that are evenly spaced, starting at the first posting, so that each skip bridges about sqrt(L) postings and all skips are of the same size. (Having skips of the same size allows the intersection algorithm to know whether it is at a skip pointer posting by counting.)

Assume that the docID of the destination of the skip pointer is stored at the source of the skip pointer. This is what is done in the textbook. (If the docID of the destination were not provided at the source, then every skip pointer would need to be followed to find the

destination docID before deciding whether to use the skip pointer; this saves space but wastes time.)

For a collection of one hundred billion documents, and postings that are pairs (DocID, term frequency), let the representation of one posting in a postings list, using ***no*** compression, be one of the following two forms:

form of posting when there is skip pointer:

docID		doc ID of destination of skip pointer		skip pointer		term frequency
5 bytes		5 bytes	3 bytes	2 bytes

form when there is no skip pointer:

doc ID		term frequency
5 bytes		2bytes

Part A Suppose we compress each postings list by representing each gap between successive docIDs and each gap between docIDs on either end of a skip pointer, both using ***variable byte encoding***. Estimate the space in bytes required for a postings list with this compression. Your estimate should be in terms of L. Your answer should be an estimate of the space used, but it will be graded on the ***quality and correctness of the estimate***, i.e. expect deductions for very coarse estimates.

Part B For a list of one million postings, how much compression is being achieved with the representation of Part A in comparison to the representation without compression presented at the beginning of this problem?

Problem 2 (2013 exam 2 problem)

The example of recommendation by content filtering given in class (slides 6 and 7 of "recommenders and search" under 3/26/14) uses a very simple method: Books are characterized by a weight for each of a set of characteristics, giving a vector characterizing the book. For each book characteristic, the weights of all books purchased by a user are averaged to produce a vector characterization for the user. A new book is scored by taking the dot product of the new book's vector of characteristics and the vector characterizing the user. If a user gives an explicit preference weight for a characteristic, this weight replaces the average for that characteristic in the vector characterization for the user.

Part A: Develop a recommendation method based on content filtering that combines for ***each*** characteristic both a user's expressed preferences and a user's ratings of books read. Now, instead of simply knowing whether the user purchased a book or not, all purchased books have a 1 (worst) to 5 (best) integer rating from the user (with a default of 3 in case a users does not bother to rate a book). Books are still characterized by weights for different book characteristics, each weight a real number ranging between 0 and 1, inclusive, that indicates the degree to which a book has the characteristic. Users may still

give integer-valued preference values for characteristics, ranging between -2 (strongly dislike) and 2 (strongly like). A user may give preference values for some but not all characteristics. Give precise descriptions of your algorithm for scoring new books for the user and of the criteria for recommending a new book. ***Your algorithm will be judged on the potential for effectiveness and efficiency.***

Part B: Apply your method to the example below. An "X" indicates a characteristic for which the user did not give a preference value:

	1st person	Romance	Mystery	Sci-fi	User rating
Book 1	0	1	1	0	5
Book 2	0	1	0	0	4
New book A	1	.5	0	0	
New book B	0	1	0	.2	
User pref.	0	1	X	-2	

Part C: What features of your method lead you to believe it will be effective?