COS 435, Spring 2017 - Problem Set 5

Due 11:59 pm Wednesday April 12, 2017 by DropBox submission

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. For each problem, list the students with whom you discussed general methods of solving the problem.

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 10am Thursday (4/13/17).
- Penalized 25% of the earned score if submitted by 4:30 pm Friday (4/14/17).
- Penalized 50% if submitted later than 4:30 pm Friday (4/14/17).

Submission

Submit your solutions as a PDF file using the Computer Science Department DropBox submission system for COS435 at
https://dropbox.cs.princeton.edu/COS435_S2017/HW5 Name your file HW5.pdf. If you have not used this facility before, consult the instructions at
https://csguide.cs.princeton.edu/academic/csdropbox - student
Note that you are automatically enrolled in CS DropBox using the registrar's COS435 enrollment list.

Problem 1 (similar to an old exam problem)
On the next page is the 5x7 term-document matrix C for a set of documents under the "set of terms" model (contains/doesn't contain) and the matrices U, Σ, and V^T that make up the singular value decomposition of C.

Part a. Give the matrices of the rank-three approximation of C. That is, give U'_3, Σ'_3, and V'^T_3.

Part b. What is the 3-dimensional representation of Doc 5 for the rank-three approximation?

Part c. What is the 3-dimensional representation of term "cat" for the rank-three approximation?

Part d. In the 3-dimensional representation, what is the similarity of "cat" and "cow"? of "cat" and "dog"? What are the dot product similarities of the original representations of these terms as given in matrix C?

Note: on slide 4 of the slides on Latent Semantic Indexing, I state U is an M×r matrix, V is an N×r matrix and Σ is an r×r diagonal matrix. In this problem statement, U is given as an M×M matrix, V as an N×N matrix and Σ as an M×N diagonal matrix. The larger dimension matrices are actually what is produced in computing SVD. They are what is shown in the on the next page. In the lecture notes, I truncate the matrices in the same way we do to get M_k, N_k, and Σ_k (slide 7) because I know only the first r singular values are non-zero (as indicated in the diagram on slide 4). For the matrix C below, r=5.

For Problem 1:

C =

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7
cat	1	0	1	1	0	0	1
cow	0	1	0	0	1	1	0
dog	1	0	1	1	0	1	1
pig	0	1	1	0	1	1	1
rabbit	1	1	1	0	0	1	0

U =

-0.40972	0.54966	-0.19439	-0.19354	0.67436
-0.27231	-0.59693	-0.05582	0.58538	0.47301
-0.53695	0.39476	-0.03386	0.55283	-0.49909
-0.51397	-0.40542	-0.51446	-0.48414	-0.26907
-0.45332	-0.14614	0.83263	-0.28260	0.00260

Σ=

3.73682	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	2.20586	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	1.19304	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.70548	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.49931	0.00000	0.00000

V^T=

-0.37465	-0.33173	-0.51219	-0.25333	-0.21041	-0.47542	-0.39088
0.36189	-0.52065	0.17810	0.42814	-0.45440	-0.34169	0.24435
0.50659	0.21990	0.07536	-0.19132	-0.47801	0.19151	-0.62254
0.10871	-0.25707	-0.57755	0.50928	0.14350	0.52655	-0.17698
0.35622	0.41365	-0.18266	0.35102	0.40844	-0.58592	-0.18787
0.50226	-0.49771	-0.00455	-0.50226	0.49771	0.00000	0.00455
0.28473	0.29260	-0.57733	-0.28473	-0.29260	0.00000	0.57733

computed with *www.dotnumerics.com/MatrixCalculator*
verified using *http://comnuan.com/cmnn01004/*

Problem 2:

Slide #26 of Part 2 of the slides for clustering, posted under April 3, presents an iterative improvement algorithm for divisive partitioning. This problem addresses recalculating the total relative cut cost (slides #20 and #21) incrementally for use with that algorithm.

Let U denote the set of objects to be clustered. Assume that for any objects v and w, sim(v,w)=sim(w,v) (we have been assuming this in class). Also assume that for any object v, sim(v,v)=0. Let C_p be an arbitrary cluster containing object x, C_q be an arbitrary cluster that does not contain x. (The set notation $C_p - \{x\}$ denotes C_p with x removed, and $C_q \cup \{x\}$ denotes C_q with x added.)

The following relationship holds for incremental changes to the intracost of a cluster when removing or adding an object x.

$$\text{intracost}(C_p) - \text{intracost}(C_p - \{x\}) = \sum_{v_i \text{ in } C_p - \{x\}} \text{sim}(v_i, x)$$

$$= \sum_{v_i \text{ in } C_p} \text{sim}(v_i, x) \qquad \text{since sim}(x,x) = 0$$

From this relationship we derive the incremental cost changes for intracost:

$$\text{intracost}(C_p - \{x\}) = \text{intracost}(C_p) - \sum_{v_i \text{ in } C_p} \text{sim}(v_i, x)$$

$$\text{intracost}(C_q \cup \{x\}) = \text{intracost}(C_q) + \sum_{v_i \text{ in } C_q} \text{sim}(v_i, x)$$

Your task is to derive incremental cost equations for cutcost. The ultimate goal is to minimize the computation time used by the iterative improvement algorithm.

Part a:

i. Give an equation for **cutcost(C_p)-cutcost($C_p - \{x\}$)** when x is an object in C_p. Your equation should be in terms of similarities between x and other objects.

Using your equation, derive equations for

ii. cutcost($C_p - \{x\}$) as an incremental change to cutcost(C_p);

iii. cutcost($C_q \cup \{x\}$) as an incremental change to cutcost(C_q).

Part b: Given the equations for the incremental changes in intracost and cutcost, what is the computational time complexity of the step:

move v_j to that cluster, if any, such that move gives maximum decrease in cost

of the iterative improvement algorithm on slide #25? Specify the data structures you are using and how they are used to achieve the time complexity.