Boston University
CS 562- Advanced Database Systems

Take Home Final Exam – Spring 2020
Return by: 5:00PM EST, Thursday, May 7, 2020 on Gradescope

Instructions

- This exam is OPEN books and notes.
- You have **48 hours** to complete it. There are 4 Problems.
- Make sure that you answer all the questions.
- Use the Gradescope and return your answer by 5:00PM EST, Thursday, May 7, 2020.

Good Luck!

Problem 1

1) Dynamic Time Warping (DTW) is a technique to find an optimal alignment between two time series and is also used to define the similarity (or distance) between time series. Subsequence matching is used to find a subsequence that best matches a query sequence: Given a large time series $X=[x_1, x_2, \ldots, x_n]$ and a query $Q[q_1, q_2, \ldots, q_m]$, we need to find a subsequence $X[x_i,\ldots,x_j]$ from X that has the smallest DTW distance to Q. Provide an algorithm to find the best subsequence matching efficiently. What is the cost of your algorithm?

2) Using the algorithm that you discussed in the previous question, compute the subsequence similarity between the time series X=[1,3,3,2,1,3,2,4] and Q=[3,1,1] and report the best subsequence from X that has the smallest DTW distance to Q. (If there are more than one, you just need to report one).

3) Give an approach to index a large set of strings for similarity queries assuming the similarity function is the Edit distance. Namely, how we can organize the set S of strings so when we have a new query string Q, we can find the string in S that is the most similar to Q under the Edit distance.

Problem 2

1) FastMap is a dimensionality reduction technique that maps points from a high dimensional Euclidian space into a lower dimensional space. What is the problem of the FastMap when the space is not Euclidean (for example when the space is a non-vector space)? Explain what can be the issue in this case. Also, propose an approach to solve the problem.

2) Consider the following distance matrix (pairwise distances) between 5 objects. Apply FastMap on these objects to get the projection to 1-d space (one dimension). Explain every step.

	O1	O2	O3	O4	O5
O1	0				
O2	2	0			
O3	1	1	0		
O4	3	3	3	0	
O5	4	3	3	1	0

3) Explain which property of the *metric distance* we use in the search algorithm of the **M-tree** and how we use it.

Problem 3

1) Explain how SVD is used to index documents and retrieve the most similar documents to user queries. What is the difference (if there is any) between LSI and SVD ?

2) In typical large document databases, the number of unique (different) words (terms) that appear in all the documents, which is called dictionary, can be very large. What is the challenge of using SVD in that case?
Can you propose a technique that combines another dimensionality reduction technique with SVD in order to address this problem? What are the advantages and disadvantages of your approach? Explain.

For a specific example, assume that you have a database of 10 million documents and a dictionary of 50000 words. How you will apply this technique to this example.

3) Assume a dataset of 4 documents and 6 terms with the following term-doc matrix.

D =

```
1     2     0     0
1     1     0     1
3     4     0     0
1     1     1     0
0     1     2     3
0     0     2     3
```

with SVD decomposition, D = U *S* V_T:

U =

-0.32	0.19	0.06	0.54	0.08	0.74
-0.26	-0.02	0.53	-0.50	0.61	0.13
-0.73	0.45	0.05	-0.11	-0.36	-0.32
-0.24	0.03	-0.83	-0.24	0.40	0.09
-0.38	-0.58	0.03	0.50	0.28	-0.42
-0.26	-0.64	-0.03	-0.36	-0.48	0.37

S =

6.07	0	0	0
0	4.96	0	0
0	0	1.0	0
0	0	0	0.67
0	0	0	0
0	0	0	0

V =

-0.50	0.31	-0.08	-0.80
-0.74	0.32	0.07	0.58
-0.25	-0.48	-0.83	0.05
-0.36	-0.74	0.54	-0.11

and 1/6.07 = 0.16, 1 / 4.96 = 0.20

i) Given the above SVD decomposition, compute the mapping of the 5 documents to a 2-dimensional LSI space.

ii) Given the following query: Q = $[1\ 1\ 0\ 0\ 0\ 0]_T$, find the best 2 matches between the query and the documents using the LSI approach.

Problem 4

1) What are the differences and what the similarities between Pig and Hive.

2) Provide a Map Reduce algorithm to compute the product of a matrix M [m x n] and a matrix N [n x s]. You need to provide the map and reduce functions.

   The product of two matrices P = M * N is defined as:

$$P[i,j] = \sum_{k=1}^{n} M[i,k] * N[k,j]$$

   You can make your own assumptions about the storage and structure of the matrices M and N (for example, if they are dense or sparse, etc).

   Is your algorithm a minimal Map Reduce algorithm? If yes, show why it is. If not, discuss if it is possible to get minimal Map Reduce algorithm for this problem.

3) Spark uses *lazy evaluation* to compute transformations to RDD's. That is, it does not apply transformations to RDDs until it is required to do so. What is the benefit for Spark to do that? Explain.