# CSIT 5930 Search Engines and Applications

# Final Examination, Fall 2020

# Dec 12, 2020

Time Allowed: 2.5 hrs

|  |
| --- |
| ***Required:*** *Type in the above statement in the space below:*  [ ] I acknowledge that the exam is from 2pm to 4:30pm (2.5 hrs)  *Full Name:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Student ID:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Date:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Submission: Submission your answer paper to Canvas.**

Please read the following clearly:

* You can type in your answer in this Word file and submit a PDF version of it on Canvas; a PDF version of this file is available for your reference.
* This exam can be completed without programming. If you choose to write programs to verify your computational results, the time is at your own expense, and there is NO NEED to submit your program.
* If the questions are unclear or need clarifications, email to [dlee@cse.ust.hk](mailto:dlee@cse.ust.hk) or use private message in Zoom's chatroom; continue with other questions until you get a reply.
* Check your emails frequently; important clarification will be emailed to everybody.
* Zoom meeting has been set up; you can ask private questions in the chatroom:

https://hkust.zoom.us/j/98243201848?pwd=Y2lHd0tZTS9INU9GYnFrQ3ozRVVqQT09

Meeting ID: 982 4320 1848 Passcode: 9m040t

* If you feel the question is not clear and information is insufficient for you to answer the questions, you can write down your assumptions for consideration in grading.
* You need to submit your answer file to Canvas before the deadline. To avoid overloading Canvas, I will keep the system open for 5 min after the deadline. However, you cannot modify the file after the deadline, and we will check the last modified time of your submission.

1. [3] Which of the following statements are true about **relevance feedback**?

|  |  |  |
| --- | --- | --- |
| T | **F** | a) All keywords extracted from relevant documents relevant (through implicit or explicit feedback) should be included into the reformulated query |
| T | **F** | b) User clicks on the search results are called explicit feedback |
| **T** | F | c) In the relevance feedback formulae covered in the lecture, user feedbacks from several rounds of query reformulations are accumulative in the feedback process |
| **T** | F | d) Relevance feedback may find relevant documents that do NOT contain any of the keywords in the original query |
|  |  |  |
|  |  | (i) (b), (c) and (d) only  (ii) (a), (b) and (d) only  (iii) (a), (b) and (c) only  (iv) (c) and (d) only |

1. [3] Which of the following statements are true about **PageRank** and **HITS**?

|  |  |  |
| --- | --- | --- |
| **T** | F | a) Given a link from page X to page Y, the anchor text of the link in X is considered as part of the content of Page Y |
| **T** | F | b) Google weights keywords enclosed in some HTML tags higher |
| **T** | F | c) Clever, the search architecture proposed based on HITS, computes hub and authority weights during query processing time |
| **T** | F | d) Hub and authority weights are more sensitive to changes on the web graph than PageRank |
|  |  |  |
|  |  | (i) All of the above  (ii) (b), (c) and (d) only  (iii) (b) and (d) only  (iv) (b) and (c) only |

1. [3]Which of the following assumption(s) is/are required in the five page preference heuristics covered in the lectures:

|  |  |  |
| --- | --- | --- |
| **T** | F | a) The user must read and evaluate the relevance of the list of results from top to bottom |
| **T** | F | b) The user must click on pages relevant to his/her query starting from the top of the result page |
| T | **F** | c) The user must read and evaluate the relevance of all of the results shown on the entire result page |
| T | **F** | d) The user must indicate a relevance rating for pages relevant to his/her query |
|  |  |  |
|  |  | (i) (b) and (d) only  (ii) (b) and (c) only  (iii) (a) and (b) only  (iv) (a) (b) and (c) only |

1. [3] Curiosity based recommendation systems consider the following factors in choosing items for recommendations

**a)** Whether the user is already familiar to the item

**b)** Whether the item is freshly accessed by the user

**c)** Whether the item is frequently accessed by the user

d) Whether the item is relevant to user’s expressed interest in his/her profile

(i) All of the above

(ii) (a) and (b) only

(iii) (b) and (c) only

(iv) (a), (b) and (c) only

1. **[**3**]** Which of the following statements are true about the factorization approach for recommendation system as studied by Koren.

a) A rating matrix can be factorized in the same way as term-document matrix to obtain latent features  
b) SVD is rarely used to obtain latent features of rating matrices because it is very expensive to run  
**c)** User bias is used to account for the toughness of a user in his/her ratings  
**d)** Item bias is used to account for items which on average are more popular than the average items.

e) Training data is not required but desirable to obtain good recommendation performance  
  
(i) (a) and (b) only

(ii) (b), (c) and (d) only

(iii) (c) and (d) only

(iv) (c), (d) and (e) only

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **[10]** The following tables show the ranked search results of two queries. 0 means non-relevant, 1 somewhat relevant, 2, relevant, 3, very relevant. Considering 0 and 1 as non-relevant, and 2 and 3 relevant. Assume that the two tables contain all of the documents retrieved. For each of the two queries:

(i) Compute precision@5

(ii) Compute AP and their MAP

(iii) NDCG.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Query 1** | |  | **Query 2** | |
| **Rank** | **Score** |  | **Rank** | **Score** |
| 1 | 1 |  | 1 | 3 |
| 2 | 0 |  | 2 | 0 |
| 3 | 2 |  | 3 | 1 |
| 4 | 3 |  | 4 | 2 |
| 5 | 0 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Query 1 | Query 2 |
| (i) | Precision@5 | 2/5 | 2/4 |
| (ii) | AP | (1/3 + 2/4)/2 = 0.42 / 5/12 | (1/1 + 2/4)/2 = 0.75 |
|  | MAP | (0.42 + 0.75)/2 = 0.585 / 7/12 | |
| (iii) | DCG | 1 + 2/log2(4) + 3/log2(5) = 3.29 | 3 + 1/log2(4) + 2/log2(5) = 4.36 |
|  | Ideal Ranking | 3, 2, 1, 0, 0 | 3, 2, 1, 0 |
|  | IDCG | 3 + 2/log2(3) + 1/log2(4) = 4.76 | 3 + 2/log2(3) + 1/log2(4) = 4.76 |
|  | NDCG | 0.691 | 0.915 |

1. **[5]**

**(i) [2]** What is the main conceptual difference between HyPursuit/Wise and PageRank/HITS in how they utilize links in ranking pages.

The main difference is that HyPursuit/Wise considers links as an indicator for the similarity between the source and target pages, whereas PageRank/HITS uses links to infer page quality or authority.

**(ii) [3]** Consider a bipartite graph consisting of 5 web pages, where 2 pages link to the same set of pages as illustrated here. Based on the connections, can you infer, conceptually, the similarity of the two source pages and the similarity of the three destination pages? Justify your answer.

A page links to another page because the target page must be fulfilling additional information needed by the source page. Thus, two pages linking to the same page means that the two source pages need the same information from the target page. Conversely, we can vaguely infer that the two source pages must be talking about the same subjects. Now, one co-cited page may not be a strong signal, but three pages co-cited by the two source pages provide a strong signal that the two pages are about the same topic (i.e., have high similarity).

1. **[10**] Given the web graph below,

C

A

B

D

**(i) [3]** write down the PR formula for each page,

**(ii) [4]** compute the PageRank values of the pages using damping factor d=0.8. Use L1 normalization in each iteration,

**(iii) [3]** How many rank sinks exist in the graph and how is (are) the rank sink(s) affect convergence.

PR(A) = 0.2 + 0.8 (PR(B)/2) = 0.2 + 0.4\*PR(B)

PR(B) = 0.2 + 0.8 \*PR(A)

PR(C) = 0.2 + 0.8 (PR(B)/2 + PR(D)) = 0.2 + 0.4\*PR(B) + 0.8\*PR(D)

PR(D) = 0.2 + 0.8 PR(C)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteration** | **0** | **1** | **Normalized** | **2** | **Normalized** |
| PageRank(A) | 1/4 | 0.2 + 0.4\* 0.25  = 0.3 | 0.19 | 0.30 | 0.19 |
| PageRank(B) | 1/4 | 0.4 | 0.25 | 0.35 | 0.22 |
| PageRank(C) | 1/4 | 0.5 | 0.31 | 0.5 | 0.31 |
| PageRank(D) | 1/4 | 0.4 | 0.25 | 0.45 | 0.28 |

The following discussion is based on pure reasoning based on the PR formula and the structure of the graph. Thus, the discussion cannot be as accurate as if the PR values were computed in large iterations to observe the convergence. Further, my discussion is more lengthy than necessary.

(i) See above.

(ii) See table above.

(iii) About number of rank sinks: There is only one rank sink C-D, i.e., surfer would be trapped in C-D (but will not be trapped in A-B).

About convergence:

Important observation: A-B has a PR leakage; every loop B passes only half of its PR to A (leaking one half to C). Thus, eventually, the PR of A and B will converge to (1-d). This impacts the C-D loop.

C-D is a classical rank sink. However, in each loop, it takes in a PR value from the A-B loop. The final result depends on d.

If d=1, PR of the A-B loop will converge to zero, and C-D is expected to converge to some constant. [The guess of the converged value is not required in this question.]

If d<1, we can see that C always receive a positive PR from B due to (1-d)>0 and PR inherited from D, d\*PR(D), whereas D only receives d\*PR(C). These two values will not be the same and will swap between them in each loop. Thus, we cannot call this convergence.

Here is my R program for reference only:

|  |
| --- |
| Iteration 1 before:[1] 0.3 0.4 0.5 0.4  Iteration 1 after :[1] 0.1875 0.2500 0.3125 0.2500  Iteration 2 before:[1] 0.30 0.35 0.50 0.45  Iteration 2 after :[1] 0.18750 0.21875 0.31250 0.28125  Iteration 3 before:[1] 0.2875 0.3500 0.5125 0.4500  Iteration 3 after :[1] 0.179688 0.218750 0.320312 0.281250  Iteration 4 before:[1] 0.28750 0.34375 0.51250 0.45625  Iteration 4 after :[1] 0.179688 0.214844 0.320312 0.285156  Iteration 5 before:[1] 0.285937 0.343750 0.514062 0.456250  Iteration 5 after :[1] 0.178711 0.214844 0.321289 0.285156  Iteration 6 before:[1] 0.285937 0.342969 0.514062 0.457031  Iteration 6 after :[1] 0.178711 0.214355 0.321289 0.285645  Iteration 7 before:[1] 0.285742 0.342969 0.514258 0.457031  Iteration 7 after :[1] 0.178589 0.214355 0.321411 0.285645  Iteration 8 before:[1] 0.285742 0.342871 0.514258 0.457129  Iteration 8 after :[1] 0.178589 0.214294 0.321411 0.285706  Iteration 9 before:[1] 0.285718 0.342871 0.514282 0.457129  Iteration 9 after :[1] 0.178574 0.214294 0.321426 0.285706  Iteration 10 before:[1] 0.285718 0.342859 0.514282 0.457141  Iteration 10 after :[1] 0.178574 0.214287 0.321426 0.285713 |
| options(digits=6)  pagerank <- function(PR) {  PR.x <- vector()  PR.x[1] <- (1-d) + d\*(PR[2]/2)  PR.x[2] <- (1-d) + d\*PR[1]  PR.x[3] <- (1-d) + d\*(PR[2]/2 + PR[4])  PR.x[4] <- (1-d) + d\*(PR[3])    cat(sprintf("Iteration %2d before:", i))  print(PR.x)  cat(sprintf("Iteration %2d after :", i))  PR.x <- PR.x/sum(PR.x)  }  PR <- c(0.25, 0.25, 0.25, 0.25)  d <- 0.8  for (i in 1:10) {  PR <- print(pagerank(PR))  } |

1. [5] The web consists of a set of linked web pages; search engine directs users to a small subset of the webpages (typically 10 pages for each query). Discuss how the existence of Google, which implements PageRank and indexes all pages on the web, impacts the random surfer model. Write no more than 10 lines.

There is no standard argument for this question.

Take the random surfer model, there are two factors affecting a surfer's visit to a page, teleporting and link traversal. Teleporting is assumed to give equal probability of visiting any page, but with search engine, teleporting probability to a page is significantly skewed by the search engine. On the other hand, link following refers to the probability of visiting a child after the surfer has landed on a page. Link following is not affected by search engine since the surfer has already left the search engine. Given the tremendous traffic volume created by Google, I believe the skewed teleporting probability alone has changed the random surfing pattern significantly.

[My optional comments: If the search engine is based on PageRank, then there is positive feedback effect in that pages having high PageRank will be ranked high and visited by surfers. These pages will attract more links and hence higher PageRank. If the search engine is purely content based (e.g., vector space model), search engine will skew the traffic but have no permanently change to the surfing behavior since contents will not change because of search engine traffic.]

1. **[15]** Given the two-dimensional document space below showing the set of documents returned from a query. In answering this question, a rough visual estimation of distances is enough (of course, you can use a ruler if you want).

d1

d3

d5

d4

d6

d2

Term 1

Term 2

**(i) [7]** Suppose d2, d4 and d6 are relevant documents and d1, d3 and d5 are non-relevant documents of the query. We want to learn an SVM based on the given sets of relevant and non-relevant documents. On the diagram, (i) draw the classification function (also called weight function in the lecture); (ii) label clearly the support vectors. **Argue** why your choice of the classification function is the best you can find.

**d1, d2, d3, d4, d5, d6**

d1

d3

d5

d4

d6

d2

Term 1

Term 2

Support vectors

We need to draw a line to separate the red and black nodes. We can try two possible choices, one passing through (d2,d4), and the other (d1,d5). It is obvious to my eye that (d1,d5), the pair of blue dotted lines has wider margin than the pair of green dotted lines. I try all possibilities since the number of points is small, but I cannot find any better ones.

While students may find other possibilities, but the justification must make sense. It is obvious that the classification must be correct, i.e., it should not classify a non-relevant document as relevant and a relevant document as non-relevant.

**(ii) [3]** Based on intuition and visual inspection of the figure, which term is more important in determining the relevance of the documents?

Term 2 is more important than Term 1; a horizontal line (i.e., independent of Term 1) can separate the relevant and non-relevant documents clearly (i.e., below d2 and d4 and above d5), but no vertical line can do that. Also, we can obtain the weight vector but it requires assuming the coordinator values and do some calculation.

**(iii) [5]** Suppose you have leant a classification function for a user. Describe how you can use the function to personalize the search result for a user. Use no more than 5 lines.

The classification function (or weight vector) indicates the relative importance of the two terms for identifying relevant and non-relevant documents. Thus, following the vector space model, one can do an inner product between the weight vector (or its normal vector) and the query vector. This will add score (similarity) to documents which has high weight in Term 1.

1. **[10]** Given term-document matrix (each row is a word, and each column is a document). Its SVD: is given below.

|  |  |  |  |
| --- | --- | --- | --- |
| C | U | S | VT |
| 1.0 0.0 1.0 1.0  0.0 1.0 0.0 0.0  1.0 1.0 0.0 0.0  1.0 0.0 0.0 1.0 | -0.7 -0.4 0.6 0.1  -0.1 0.6 0.4 -0.7  -0.4 0.7 -0.2 0.6  -0.6 -0.2 -0.7 -0.4 | 2.3 0.0 0.0 0.0  0.0 1.5 0.0 0.0  0.0 0.0 0.7 0.0  0.0 0.0 0.0 0.4 | -0.7 -0.2 -0.3 -0.6  0.1 0.9 -0.2 -0.4  -0.4 0.2 0.9 -0.1  0.6 -0.3 0.2 -0.7 |

**(i) [2]** Obtain the rank-2 approximation of , and , i.e., , and . Show one decimal digit only.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| -0.7 -0.4  -0.1 0.6  -0.4 0.7  -0.6 -0.2 | 2.3 0.0  0.0 1.5 | -0.7 -0.2 -0.3 -0.6  0.1 0.9 -0.2 -0.4 |

**(ii) [8]** For the query , in the two-dimensional reduced space, obtain the cosine similarity of to the documents, and thus their rankings.

Q\_2 = Q \* U\_2 \* inv(S\_2)

Q\_2\_norm = Norm(Q\_2)

V\_2\_norm = Norm(V\_2)

Similarity = V\_2\_norm \* transpose( Q\_2\_norm)

Ps: all “\*” refer to matrix multiplication.

Check the detailed code in the attachment

[[ 0.72637256]

[-0.38683878]

[ 0.99961173]

[ 0.99961173]]

1. **[15]** In the following rating matrix, ratings are given from 0 to 1 and blanks mean missing/unknown ratings.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i1 | i2 | i3 | i4 | i5 | i6 | i7 | i8 |
| u1 |  |  | 0.2 |  | 0.3 |  |  | 0.4 |
| u2 | 0.4 |  | 0.3 |  |  | 0.2 | 0.4 |  |
| u3 |  | 0.4 | 0.9 |  | 0.4 |  | **???** | 0.5 |
| u4 |  |  | 0.7 |  | 0.6 |  | 0.7 | 0.5 |

We are going to use item-based collaborative filtering to predict the missing rating labelled with "???" in the matrix.

**(i) [2]** In this example, what is the maximum number of items that we can use in predicting the missing rating?

In item-based CF, we must find items rated by u3. Since u3 has rated 4 items, the maximum is 4.

**(ii) [8]** Using the top-2 most similarity items, predict the missing rating "???", considering item-item similarity and item bias (i.e., items could have strong appeal or repulse to all users).

We need to find the most similar two items. Since no similarity measure is specified, we use inner product.

sim(i2, i7) = 0

sim(i3, i7) = 0.12 + 0.49 = 0.61

sim(i5, i7) = 0.42

sim(i2, i8) = 0.35

Thus, the top-2 items are i3 and i5. To cater for item bias, we use the delta formula.

**(iii) [5]** Is it possible to predict missing ratings for i4 using either user-based or item-based CF? Justify your answer.

For IBCF, it is no possible because the i4 vector is undefined, so we cannot find the "neighbors" of i4. Note we cannot treat i4 has a zero vector.

For UBCF, it is not possible since we need to find users who have rated i4 and no one has rated i4.

However, given that we cannot find any good item or user neighbors, we can still resort to averaging the user's ratings on the other items (in this case, i2, i3, i5, i8) based on IBCF, but for UBCF, I cannot see any reasonable alternatives.

I can accept both answers if students can give reasonable explanation.

1. **[5]** Does collaborative filtering support accuracy-based or curiosity-based recommendation, both or neither? Justify your answer using user-based or item-based CF or both. This is an open-end discussion; use no more than 10 lines.

There is no standard answer.

Accuracy-based RS tries to predict user's future interests that are similar to user's past interests. Thus, CF is accuracy-based. Item-based CF tries to be smart by exploiting ratings given to similar items while user-based CF tries to exploit similar users.

Curiosity-based RS in fact tries to recommend items that deviate from user's past interests. Thus, on the surface, CF is not curiosity based. However, one can argue that CF, user-based CF can recommend items that deviate from a user's past interest since the recommend items come from items consumed by other users. One can argue that no two persons are exactly identical in their interests. This is a weak argument because we can argue that UBCF tries to identify similar users so that the recommended item can be as close to the target user's past interest as possible. Anyway, this answer may deserve some credit.

1. **[5]** SVD and Latent Factor Model (LFM, ref: Koren’s paper) aim to factorize a matrix into two smaller matrices, the product of which approximates the original matrix. **(i)** Is there any common/shared objectives between SVD and LFM? please explain; **(ii)** Explain why it is not appropriate to apply SVD to factorize a user-item rating matrix. [max: 2 sentences of each part]

(i) Matrix factorization's common objective is to reduce an object to a small set of representative features (latent features/factors/indexes) with reduced redundancy and noise. [For ref only: SVD is not based on machine learning while LFM is based on machine learning.]

(ii) SVD must operate on concrete values but in rating prediction we have many missing values. If we assume missing values are zero (i.e., lowest rating) or takes on the average rating, either way, it is not true in reality.

1. **[5]** Given the following text string, a text window containing two words in front of and two words behind a target word. Identify the positive samples you can use for the target word "bond" and suggest a few negative samples you can use to train a word2vec model? Do not apply stop word removal or stemming.

|  |
| --- |
| I like to watch action movies such as james bond series and I will watch the new james bond movie when it is released |

Possible samples are the words within window centered at bond.

(bond, as)

(bond, james)

(bond, series)

(bond, and)

(bond, new)

(bond, james)

(bond, movie)

(bond, when)

Negative samples are randomly generated. A few is enough for this question.

(bond, I)

(bond, watch)

(bond, released)

… …