CS 498: Computational Advertising **Fall 2018**

Homework 1

Handed Out: Sep 12, 2018 *Due: Sep 26, 2018 11:59 pm*

1 General Instructions

- This assignment is due at 11:59 PM on the due date. We will be using Compass (http://compass2g.illinois.edu) for collecting the non-programming part of this assignment. Contact TAs if you face technical difficulties in submitting the assignment. We will NOT accept any late submission, no exceptions.
- The non-programming part of homework MUST be submitted in pdf format. Handwritten answers are not acceptable. Name your pdf file as YourNetid-HW1.pdf
- For Questions 1 and 2, you need to explain the logic of your answer/result for every subpart. A result/answer without any explanation will not receive any points.
- The programming part of this assignment will be hosted on hackerrank (https://www.hackerrank.com/) as a programming contest. To participate in this contest, please create a hackerrank account with your illinois.edu email id. The contest framework will allow you to verify the correctness of your submission based on a set of sample test cases. We will use additional test cases to grade your submission (hidden test cases will be significantly larger than the samples). Please check the course piazza page for the contest URL. You need to register in the contest to solve the programming question. Remember to use the hackerrank account associated with your Illinois email id.
- It is OK to discuss the problems with the TAs and your classmates, however, it is NOT OK to work together or share code. Plagiarism is an academic violation to copy, to include text from other sources, including online sources, without proper citation. To get a better idea of what constitutes plagiarism, consult the CS Honor code (http://cs.illinois.edu/academics/honor-code) on academic integrity violations, including examples, and recommended penalties. There is a zero tolerance policy on academic integrity violations. Any student found to be violating this code will be subject to disciplinary action.
- Please use Piazza if you have questions about the homework. Also feel free to send TAs emails and come to office hours.
- Questions 1 and 2 are based on the following reference[1] chapter.

[1] https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book-ch14.pdf

2 Question 1 (5 points)

Scroll to Section 14.7, Exercises in the reference chapter linked on the first page. In this question, you are required to solve Problem (2). In parts (a) and (b), you must show the precise steps to arrive at your answers. They carry 2 points each. Part (c) is worth a single point. You need to both, provide the right answer and an appropriate explanation for why you think you obtained this result to receive this point.

Note: Wordy explanations will be penalized, be as concise as you can.

3 Question 2 (10 points)

In this question, you are required to solve Problem (3), Section 14.7 in the reference chapter. Part (a) carries 2 points, while Parts (b) and (c) are worth 4 points each, split across the computation of authority scores and your explanations for the results.

4 Programming Question 3 (10 points)

This question aims to provide you a better understanding of the HITS algorithm in action. We will host the question via Hackerrank, the contest link will be posted on piazza soon.

- Please read the problem description carefully.
- The input will always be valid. We are mainly testing your understanding of the HITS algorithm, not your coding skills.
- Please pay special attention to the output format. We will be using the hackerrank based autograder and it is extremely important that your generated output satisfies the requirement.
- We don't have specific constrains for this programming question. The only constrains are the standard environment constraints in hackerrank: https://www.hackerrank.com/environment.
- The grading will be based on the test cases your code passes. You are provided with two sample test cases to debug your code. For the final grading, we will use real-world datasets. Make sure your code can scale to large graphs.

5 Question 4 (2 points)

Assume you know the K most prolific movie reviewers (highest hub-scores) in the above programming question, and the set of movies each of them have reviewed. A new user decides to review movies in a manner to maximize his potential hub-score, however he has a limited budget (he can review a small fixed number of movies). What is a good strategy for him to adopt given the above information?
(Limit your answer to a single line, wordy answers will be penalized)
Does the same strategy work for both large and small graphs? *Hint: The effects of adding a new user are much smaller in a large graph*