

CSC 4170

Web Intelligence and Social Computing

Project Specification for CSC4170 2009

Date: Tuesday, 6 October 2009

<http://wiki.cse.cuhk.edu.hk/irwin.king/teaching/csc4170/2009>

Target and Requirement

The target of this project is to improve your understanding of the concepts and rules in social network by implementing some of the computation introduced in lectures and to inspire your intelligence to find more new and interesting phenomenon in social networks around you. Thus we do not limit the content or the algorithms you need to do. To let you understand more about this project, we will give four examples below. You can just use one of these examples as your project or find something else that is more interesting. However, to ensure the quality and quantity of your work, if you want to do something beyond these examples, you should write a proposal (not more than two pages) first, send to us and get approved. Two students can form a group, but no more than two students. The final project scores for the two students in a group will be equal. After you finish your work, you need to hand out a written report and prepare a presentation in the class. The whole project will take 30% of your final score (20% for report and 10% for presentation). In the report, the documentation of your codes should be included, such as running environment and settings, input and output format, the whole structure of your codes, etc. In the presentation, besides introducing your work, a main part should also demo your codes, such as running your system, running your algorithms, etc. This part takes 20% from the presentation and 25% from your report.

Example 1: Implement Algorithms for Graphs

In the lectures, many algorithms for graphs are introduced. We summarize these algorithms as follows.

1. Given the size of V and E in a graph G , generate a random graph. The graph should be described in adjacency matrix.
2. Detect whether two graphs are isomorphic.
3. Calculate distance between two nodes in a graph.
4. Detect whether there is a cycle in a graph.
5. Calculate the eccentricity of a vertex in a graph.
6. Calculate the radius of a graph.
7. Calculate the diameter of a graph.
8. Calculate the girth of a graph.
9. Find peripheral vertices in a graph.
10. Detect whether an undirected graph is connected.

11. Detect whether a directed graph is strong connected or weak connected.
12. Find the cutpoint in a graph.
13. If there is no cutpoint in the graph, find the minimum weight cutset of the graph.
14. Find the bridge of a graph.
15. If there is no bridge in the graph, find the edge cutset with the smallest size.
16. Calculate graph density.
17. Calculate graph distance.
18. Detect whether a graph is a Bipartite Graph.
19. Find the largest clique/cliques in a graph.
20. Find the node/nodes with the highest degree centrality.

Implement 10 from these algorithms. The language can be chosen from Matlab, Java, C++, C#, and Python. This work is expected for single student. If a group wants to do this, the two persons can implement the algorithms in different languages.

Example 2: Improve One of the Graph Algorithms

Select one problem from the set above. This time you should survey existing algorithms for the problem. Find a classical method and a state-of-the-art method. After that, find at least one case that these two algorithms will not well perform. For example, there is a case that the calculation time is intolerable. Then try to develop your own algorithms to solve the problem. In this task, you should do comparisons in the cases among your approach and the two existing approaches, and demonstrate the effectiveness of your algorithm. This work is expected for single student. If a group wants to do this, the two persons can try different problems.

Example 3: Research from the Social Network around Us

There are many social networks around us. For example, all the professors, their Ph.D or Master students, their papers can be built as a social network. In this network, professors, students, and papers can be seen as nodes; and the supervising relationship between professors and students, the author relationship between professors/students and papers can be employed to build edges. In this task, you are requested to draw at least two graphs from the statistical information, such as power law distribution of in-degree or out-degree of different nodes, etc. You should also realize a mining function. For example, given two persons, student A and professor B, find a path between these two nodes. You can also use this function to verify the six-degree theory. This work is expected for two persons.

Example 4: Recommender System for Movies

In this work, you need to implement the algorithm for recommender systems. A movie dataset will be given to you. It includes 100000 rating information for 1682 movies from 943 users. The rating is scaled from 1 to 5. Higher value means better satisfaction. You can divide the dataset with 80% users as training and 20% as testing.

Input: 20% of a testing user's rating information.

Output: A list of movies that are unrated by the user.

Evaluation: In the top ten movies in your list, on average, how many will be hit by the user's rest 80% ratings?

Algorithm: Page Rank. You can build a graph in the following way. The transition probability can be built by the rating information of existing users. For example, if a user has rated both movie 2 and movie 4; another user has rated both movie 2 and movie 6. If no other users has rated movie 2, then the probability to transfer from movie 2 to movie 4 is $1/2$, so is movie 2 to movie 6. After all the probabilities are calculated, the stationary distribution can be calculated as the score for building your list. In this example, the 20% information of test user is not used, but you can think up how to use the 20% information to improve results for particular user.

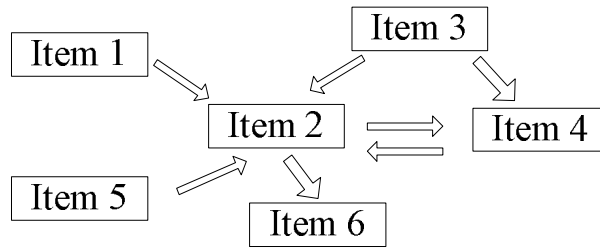


Figure 1: the page rank algorithm