

Project 3 – Influence Maximization Problem(IMP)

1. Overview

In this project, you have two computational tasks for Influence Maximization Problems (IMPs) in social networks. The first is to implement an estimation algorithm for the influence spread and the second is to design and implement a search algorithm for IMPs. An introduction to IMP (including problem formulations and influence spread definition) can be found in the package we provide (IMP.pdf). The IMP is NP-hard and the influence spread computation is #P-hard under the definitions shown in the introduction. Overall, your estimation algorithm needs to give a good estimation of the influence spread and your search algorithm needs to find a high-quality solution to IMPs as fast as possible within a limited time. The scores you get in this project will be given according to the performance of your algorithms in our test.

2. General rules

After your submission, we will test your influence spread estimator and IMP solver on different IMP problem instances. To make this process as smooth as possible, the package you submit must satisfy the following requirements.

a) Algorithm description

- i. The description of your algorithms should be submitted in pdf format, in which you should describe the core idea of your design and give the pseudo code.

b) Programming aspects

- i. In order to get rid of the operating system related issues and the execution efficiency issues of different programming languages, your algorithm must be implemented using Python 2.7 (Python 3.0+ not included) and the only allowed library is numpy.
- ii. The executable estimator must be named by ISE.py and the executable solver must be named by IMP.py

c) Task 1: influence spread computation

- i. Input: the format of the estimator call should be as follows:
python ISE.py -i <social network> -s <seed set> -m <diffusion

model> -b <termination type> -t <time budget> -r <random seed>

<social network> is the absolute path of the social network file

<seed set> is the absolute path of the seed set file

<diffusion model> can only be IC or LT

<termination type> specifies the termination manner and the value can only be 0 or 1. If it is set to 0, the termination condition is as the same defined in your algorithm. Otherwise, the maximal time budget specifies the termination condition of your algorithm.

<time budget> is a positive number which indicates how many seconds (in Wall clock time, range: [60s, 1200s]) your algorithm can spend on this instance. If the <termination type> is 0, it still needs to accept **-t <time budget>**, but can just ignore it while estimating.

<random seed> specifies the random seed used in this run. In case that your solver is stochastic, the random seed controls all the stochastic behaviors of your solver, such that the same random seeds will make your solver produce the same results. If your solver is deterministic, it still needs to accept **-r <random seed>**, but can just ignore them while estimating.

- ii. Output: the value of the estimated influence spread.

d) Task 2: influence maximization

- i. Input: the format of the solver call should be as follows:

python IMP -i <social network> -k <predefined size of the seed set> -m <diffusion model> -b <termination type> -t <time budget> -r <random seed>

<social network> is the absolute path of the social network file

<predefined size of the seed set> is a positive integer

<diffusion model> can only be IC or LT

<termination type> specifies the termination options and the value can only be 0 or 1. If it is set to 0, the termination condition is as the same defined in your algorithm. Otherwise, the maximal time budget specifies the termination condition of your algorithm.

<time budget> is a positive number which indicates how many seconds

(in Wall clock time, range: [60s, 1200s]) your algorithm can spend on this instance. If the <termination type> is 0, it still needs to accept **-t <time budget>**, but can just ignore it while solving IMPs.

<random seed> specifies the random seed used in this run. In case that your solver is stochastic, the random seed controls all the stochastic behaviors of your solver, such that the same random seeds will make your solver produce the same results. If your solver is deterministic, it still needs to accept **-r <random seed>**, but can just ignore them while solving IMP.

- ii. Output: the seed set found by your algorithm.

The format of the seed set output should be as follows: each line contains a node index. An example is also included in the package.

e) Supplementary instruction

- i. The social networks are given in a uniform format and an example is provided in the package. The first line contains the number of nodes n and number of edges m, each of the next m lines describes an edge following the format: <src> <dest> <weight>. Node index starts from 1.
- ii. The seed set are given in a uniform format and an example is provided in the package. Each line contains a node index. Your algorithm should give the output in this format as well.
- iii. Use random seeds to control all the stochastic behaviors.

3. Evaluation

We will evaluate the performance of your estimator and your solver, respectively. For each task, the test is divided into three parts.

a) Usability test (40%)

We will use some IMP problem instances and seed sets to check whether your solver or estimator are usable.

- i. Here the usability means that your estimator or solver can satisfy the input and the output requirements shown in Section 2.
- ii. Once your estimator and solver have passed a test, you will get all the corresponding scores.

b) Efficacy test (60%)

We will focus on how good your estimators and solvers are. In this test, we will select different IMP problem instances and set a common cut-off time for all participants.

- i. Influence spread computation (10%): We will select different seed sets with different sizes and test the estimation error of your estimator on the instances with these seed set.
- ii. Influence maximization (30%): we will specify different sizes of the seed set and test your solver on the instances. The solution quality of all participant solvers are compared in terms of the influence spread and consumed time.
- iii. The scores you get in this test depend on the rankings your estimator or solver obtain.

c) Scalability test (Bonus:0~10%)

We will focus on how well your estimators and solvers can scale up to large-scale IMP problems. In this test, we will select some social networks with different scales, either in the number of nodes or the number of edges, and set a common cut-off time for all participants and compare their solution quality.

- i. Similar to the efficacy test, we will test your estimator (10%) and solver (30%) on the instances and compare their solution quality.
- ii. The scores you get in this test depend on the rankings your estimator or solver obtain.

4. Test Environment

- 1) Operation System: Ubuntu
- 2) Server CPU : 2.5GHz, 14-core
- 3) Python version: 2.7.12