Project Proposal: inf-max

Yiming Fang
Email: yf2484@columbia.edu

November 29, 2022

1 Introduction

Influence Maximization (IM) is an exciting and well-researched topic, and it has practical applications to commercial marketing and social network management. In any given social network, it often is the case that some nodes are more influential than others. Sometimes, identifying a set of influential nodes can greatly help decision makers make marketing decisions. However, since the problem is proven to be NP-Hard, and most existing algorithms use greedy heuristics that run sequentially, resulting in high computational consumption. This project aims to implement massively parallel IM algorithms and compare the performance with traditional approaches.

2 Problem Formulation

The task of the IM problem identifying a "most influential set" of size k, such that the following condition holds

$$\max_{|S|=k} f(S)$$

where S is the most influential set, and $f(S)$ is the total expected cascade size of S.

There are two classic models, the Linear Threshold Model and the Independent Cascade Model. In the former, each node has a random threshold θ, and becomes influenced whenever the summed weights of its neighbors surpasses θ. In the latter, each edge $e_{(u,v)}$ has a random probability $p_{u,v}$, such that u has a one chance of influencing v with $p_{u,v}$ probability.

Furthermore, an extension of the IM problem is the Multiple Influence Maximization (MIM) proposed by [1], where multiple information can propapate independent in the same social network. This allows for a greater extent of parallelizability.

3 Implementation Plan

The baseline implementation of IM and MIM can be a simple Monte Carlo style search algorithm that runs sequentially. There are also libraries and packages that implement similar algorithms in

other languages such as Python, which can be used as a benchmark for comparison.

The main task is to implement parallel versions of IM and MIM algorithms. Some greedy algorithms for MIM are proposed by [1], where the authors give C-style pseudocode. In the MIM case, each different information can be searched using a different subroutine, and the main routine is responsible for dispatching and terminating workers by maintaining states. Parallelism is less obvious in the regular IM case, and I may need to do more research on this.

There are many choices in terms of dataset, and I am currently think of using [2].

References

[1] Wu, G., Gao, X., Yan, G. and Chen, G., 2021. Parallel greedy algorithm to multiple influence maximization in social network. ACM Transactions on Knowledge Discovery from Data (TKDD), 15(3), pp.1-21.

[2] https://github.com/mathbeveridge/gameofthrones