MTP Presentation

High Performance Graph Analytics

Aditya Navphule, M20CS001

Supervisor: Dr. Dip Sankar Banerjee

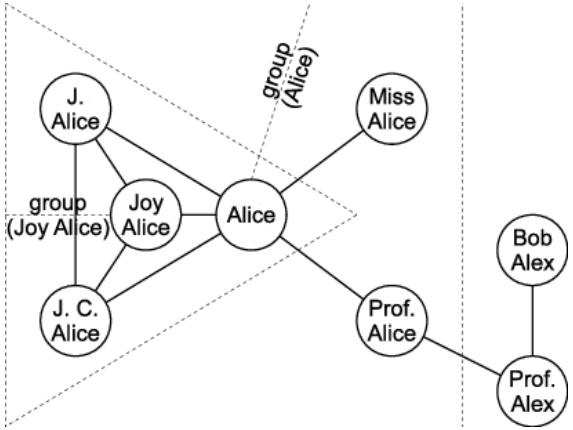
IIT, Jodhpur

November 30, 2021

Contents

Introduction

Nowadays, graphs in the real world become much larger and more dynamic. The existing solutions for similarity search are expensive in terms of time and space cost.



So, we need a faster method for computation over graph, some methods which are employed to reduce computation time are - Approximations, and by using GPU's to do the parallel calculations.

SimRank Similarity Measure

SimRank[1] says that, "two objects are similar if they are related to similar objects." It calculates the similarity between any two objects which can be represented in a graph, and can be employed in various applications combined with other features.

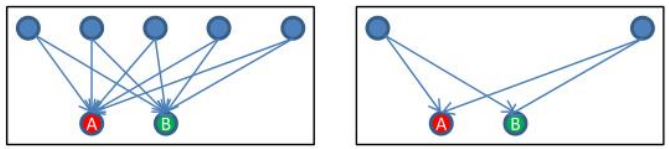
Simrank equation (2)

- Similarity btw a and b:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

- s(a,b) is normalized into (0,1)

- Proof: By induction
  - C<1
  - $s(I_i(a),I_j(b))<1$



Related Work

There are many implementations already done for SimRank, some use extra features to try and control the number of iterations of SimRank, some use approximations[2] to keep down the computation time.

One such method to keep down the computation load on CPU is to use GPU for the independent computations present in the SimRank. Exploiting the inherent parallelism and high memory bandwidth of Graphical Processors for calculating SimRank.

There is not an state of the art GPU based algorithm present for SimRank as of now, it is still in experiments using various metrics and methods.

Current Progress

In this MTP Project, I have been working on creating a GPU based algorithm for SimRank. To summarize what have I done :

- Creating a CPU bound SimRank algorithm and optimizing it for reducing time complexity.
- Experiment with Norms of a Matrix, to find if the algorithm converges prematurely, and timing it.
- Creating a GPU bound algorithm using CUDA parallel programming platform, to use the GPU's inherent parallelism for independent computations.
- Currently working on the CUDA based SimRank algorithm, and further optimising it.
- Future work contains: further modifying the algorithm for Dynamic Graphs, i.e. Continuously adding nodes and links to the graph.

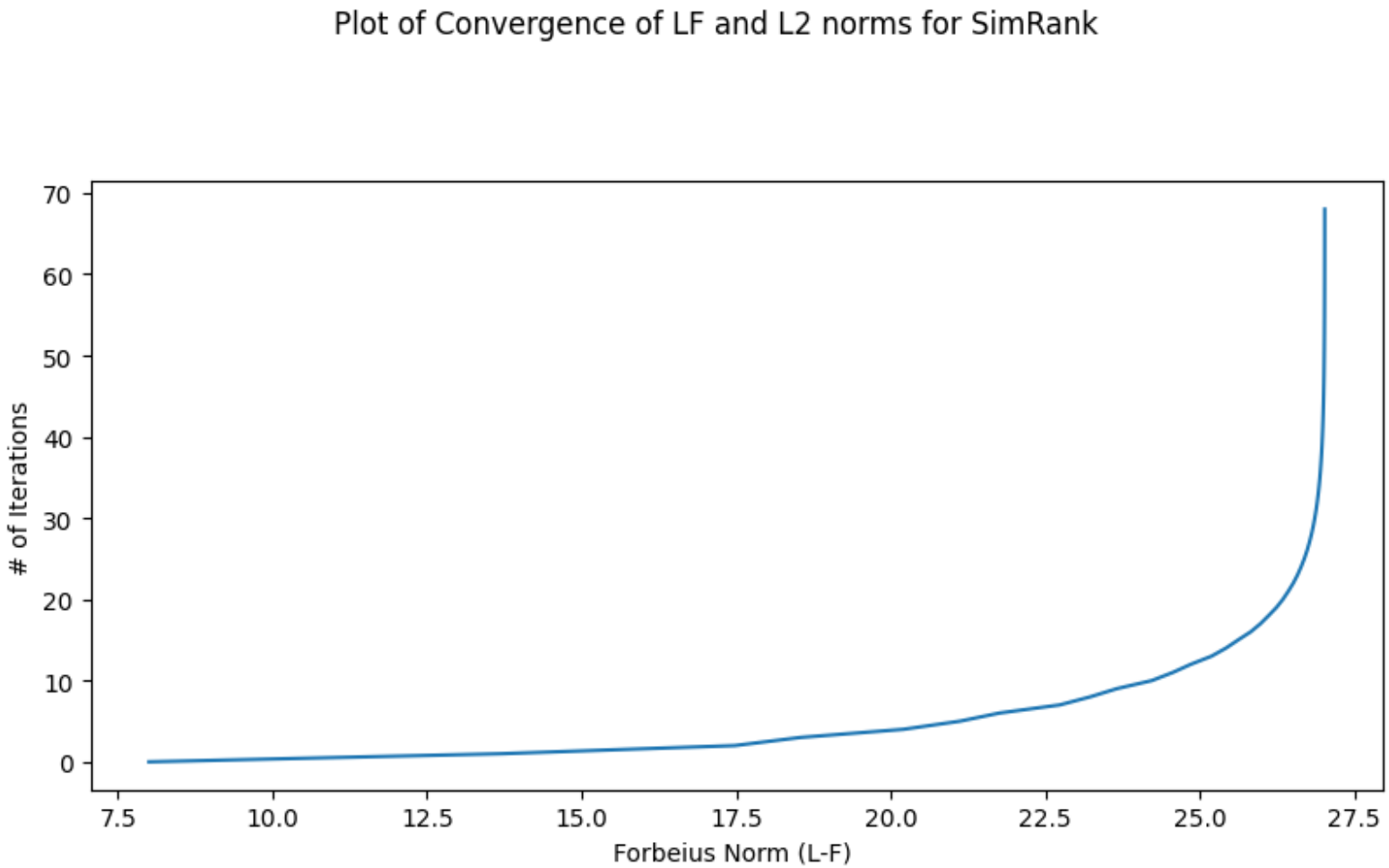
Current Work

For the current implementation of SimRank over GPU for a graph of 8 edges and 12 nodes,
Below computation times are computed in my system.

- CPU Time : 0.0184sec.
- GPU Time : 0.0123sec.
- Speed Up : 1.5

Current Work

Below is a graph showing the convergence pattern of the Forbeius Norm of the SimRank matrix for the GPU implementation.



References

1. SimRank: A Measure of Structural-Context Similarity, Glen Jeh(glenj@db.stanford.edu), Jennifer Widom(widom@db.stanford.edu)
2. An Efficient Similarity Search Framework for SimRank over Large Dynamic Graphs, Yingxia Shao et al.