GPU Project

TRÉMAS Loumi

loumi-05@hotmail.fr

Master of Science in Informatics in Grenoble

10 février 2022

Table des matières

1 Introduction

This project is about optimizing a previous project I had to do for my M1 internship for the Turing Center for Living Systems as a intern of the team MMG under the supervisions of Anthony Baptista and Anaïs Baudot .

The project was about prioritizing nodes in Multi-layer networks using the similarity of Katz as the measurement. This project was built under the ambition of predict potential new interactions between some Multi-layer networks and more precisely for medical network .

This method is working pretty well , the only issue we had was the time of computation as the similarity measure :
$S = \sum_{k=1}^{N} \alpha^k (C)_k$ with $\alpha \in [0, 1]$ and $(C)_k$ a matrix .
Rely a lot on matrix multiplication and so instantly figured that it could be optimize by using GPU-computing as it is known for calculating matrix multiplication more efficiently.

This entire work was programmed using the language Python , so naturally we tried a GPU computing implementation using the Cuda library called cupyx .

At the end of my internship , our measurements were working properly on a small network representing interaction between different airport and with a small depth in the path . Unfortunately it was not working as a fast and as good as expected for the huge genetic network representing interaction between 3 multi-layers networks composed of (genes , drugs , diseases).

So my goal for this project is to optimize the computation using GPU-computation .

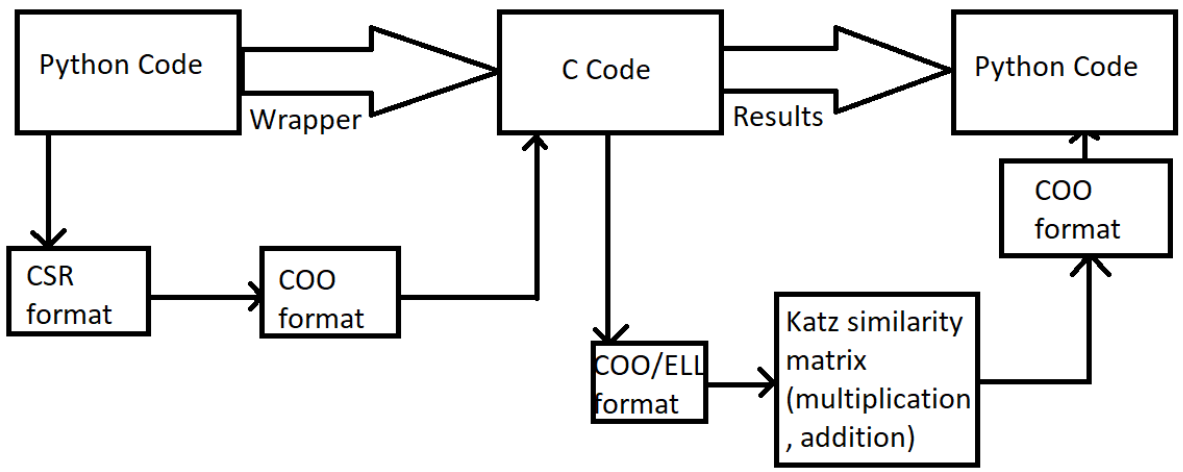
2 Theoretical organisation

As the global project was built using python and by time necessity i will not build everything from scratch . My plan is to use the python programming , to read and build the data with the library NetworkX giving me a supra-adjacency matrix which is composed of multiplexes and bipartite matrices in format of CSR .

As recommended by my teacher I will use an hybrid format between ELL/COO , to compute all calculations on the matrices .

For doing so I will first need to convert all the data generated in CSR format to COO format and then create a wrapper in C which will take the COO matrices as input ; will convert them into the Hybrid format and then the whole computation from matrix multiplication to matrix addition getting a final matrix as results of the computation . This final matrix would be converted back to the COO format and then transferred back to the python in order to be used or printed using python toolbox .
The pipeline can be resumed as the following .

FIGURE 1 – Katz Similarity Pipeline



This pipeline was the first version of my project ,after some questioning and interrogation

3 Implementation

4 Difficulties

5