Assignment: “Top-k Most Probable Triangles in Uncertain Graphs”
M.Sc. Data and Web Science 2020-2021 - Mining of Massive Datasets

Introduction

In this project, you will work with graph data. Given a potentially large probabilistic network and an integer k, you must design and implement a solution to discover the top-k most probable triangles. A probabilistic network is a network where edges are annotated with existential probabilities. This means that an edge *e* is present with a probability *p*(*e*). Therefore, if a triangle is formed among the edges *a*, *b* and *c*, then the existential probability of the triangle is *p*(*a*) * *p*(*b*) * *p*(*c*). An example is given in Figure 1.

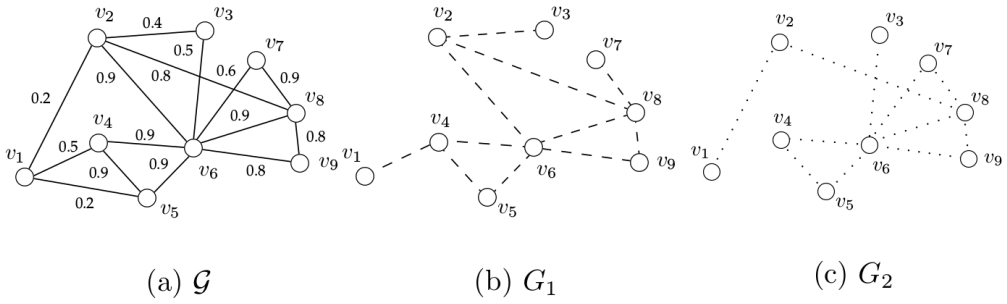

Figure 1: A probabilistic graph $\mathcal{G}$ and two possible instances G_1 and G_2. The numbers near the edges in (a) denote existential probabilities.

Datasets

You are free to use any probabilistic graph available. A small dataset will be given. However, since not many real-life probabilistic networks are publicly available, you may work use synthetic datasets as follows: you may take any undirected network and simply assign probabilities to the edges by using a probability distribution such as uniform, normal, power-law, etc. Two excellent repositories for graph data are:

http://snap.stanford.edu/data/index.html
http://networkrepository.com/

Requirements

Given a probabilistic graph you should implement a scalable and efficient algorithm to detect the *k* most probable triangles. Your algorithm must be implemented in Scala or Python and you should use Apache Spark. Note that the parameter *k* is user-defined and must be given as an input to the algorithm. Please note that in case you want to use a graph library, in Scala you may use GraphX or GraphFrames. However, if you want to use Python, then only the GraphFrames library is supported. More information about the GraphFrames library may be found in the following link:

http://graphframes.github.io/graphframes/docs/_site/index.html

The graph will be given in edge-list format. Each line contains three numbers. The first two correspond to the edge and the third one corresponds to the probability of the edge. Please note, that it is not mandatory to work with a graph library. In case you have other ideas to work with such as using a sparse adjacency matrix or adjacency lists you are free to use them. However, please take care of the scalability issue, since your solution must be able to scale for large networks.

Deliverables

You should deliver the source code of your solution and a report describing what you did, in detail. Also, you need to prepare slides for the presentation session that will take place at the end of the semester.