Deep Inamdar
Riley Gusa

Network Science
Team 1 - Project Proposal

April 08, 2021

Probabilistic Synthesis of Temporal Graphs

Idea

To construct a probabilistic model that would simulate evolution of a temporal network.

Concept

We propose a design strategy to construct a temporal network model. The model will consume independent attributes and output a time dependent temporal network.
If we consider undirected graphs, we observe the following –

1. Only two elements to take care of – nodes and edges.
2. Both elements have a birth and a lifespan.
3. If our model can take the above inputs, we will be able to simulate a temporal network.

Applications

The synthesis of temporal graphs can be used to model networks such as Pandemic, Social Network, or Demographic. All these networks have one thing in common that they evolve dynamically with time. When it comes to probabilistic temporal modeling of a graph, the real-world application of the same will only be on the networks which randomly evolve with time. i.e., man-made networks e.g., Internet, Power Grid cannot be modeled or envisioned by the synthetic temporal model because they depend on strategic methodologies.

Proposal of the Naïve Model

We observe that if we could control the lifecycle of individual nodes and edges, we can trigger a temporal network.
The following model is currently envisioned –
Inputs -

1. Starting state of a graph. The future instances can be built upon an input graph.
2. Rate of node creation per unit time. $N_c = f\ (t, \ldots)$
3. Probability of deletion of node based on a decay function. $N_d = P\ (t, \ldots)$
4. Edge creation for all nodes. $E_c = P\ (t, \ldots)$
5. Edge deletion for all existing edges. $E_d = P\ (t, \ldots)$
6. Weighted graph – $E_w = P\ (t, \ldots)$
7. Boolean flag isDirected
8. End condition, e.g., Maximum number of nodes. This is put to stop the temporal evolution.

For directed graphs,
The direction of edge will depend on a weighted tradeoff between the in-degree and out-degree. It will be taken into consideration in the E_c function.
Key points –

- The … in the function resemble any other dependent parameters.
- All the above attributes depend on time.

Example
Node Rate = New Nodes/unit time
Node decay = 1/Lifetime
P (edge creation) = P (t, degree)
P (edge deletion) = P (t, degree)
The 'degree' above is the degree at the instance t.

Background study
The reading done while writing the reaction papers was exhaustive. The background study is pretty much what is mentioned in the reaction papers. On top of it, we found a very closely related paper on the project topic we propose. The link is shared in the citations section [5]. It relates to constructing a synthetic temporal model. The idea they used is to focus on preserving local temporal structures instead of satisfying global network properties. Their idea differs from ours in a way that we work with the flexibility to provide functions for input parameters. Rest, the paper written is very diplomatic and it is hard to understand the exact model they proposed. Interesting fact is that it is a eecs.wsu.edu paper published in 2018. Thus, we hope we can get some more information on the same if it is a paper from our Network Science lab.

Data set
Source - http://www.sociopatterns.org/datasets/
The above page has interesting temporal datasets. The uniqueness of the datasets on this page is that these are constructed empirically. Since, we must test the model, a dataset generated from an experimental setup would be an ideal one to compare with. The datasets are also light weight. Out of all the datasets mentioned on the page we focus on the - High school contact dynamic network dataset.

Project Plan
Step 1 –
Finalize on the above model. Brainstorm a bit on the above model to know if we can make it more robust. But for now, we think that the model is simple yet dynamic as the parameters are functions.
Step 2 –
Write Python code to simulate the working of the model. We can also make the code spit out the graphs in real time for visualization. To do a temporal analysis, we need the adjacency matrix of each instance of the graph. This can be done by either storing them and analyze at the end or, do the analysis in place for each instance generated.
Step 3 –
Do a temporal analysis on the model. Note down observations on the temporal behavior of our model.
The features, statistics, temporal features to be analyzed would pretty much be like the citation paper 1 and paper 2 of the reaction paper. E.g., Static network features evolution – Average degree, Max Path length, etc.
Time permitting, for special cases, some other features can be studied such as Average temporal graph, theoretical strategy to predict graphs in future time t., etc.

Step 4 –
Adjusting the inputs of the model, try simulating the model which can closely resemble the results of a real dataset. We choose a dataset from the Socio-patterns website for testing our model.
Step 5 –
Compare the graph generated by our model and the corresponding real-world network. Do a similarity analysis, and quote observations.
Step 6 –
Write a report by amalgamating all the findings, comparing model with real-world network, critiquing the model, listing applications, limitations, future scope, etc.

Citations

1. Temporal Analysis of the Wikigraphs
2. Time Varying Graphs and Social Network Analysis: Temporal Indicators and Metrics
3. Sampling of Large Graphs
4. The evolution of structural balance in time-varying signed networks
5. Closely relevant paper - https://eecs.wsu.edu/~holder/pubs/NS18-synthetictemp.pdf
6. Dataset source - http://www.sociopatterns.org/datasets/