Social Network Analysis Assignment
Time-Aware Network Centrality Measures & Link Prediction

Dr. Dionisios N. Sotiropoulos

June 8, 2022

This assignment focuses on the algorithmic manipulation of the **Stack Overflow Temporal Network**, whose description and related dataset may be found at [1]. Each edge of the underlying graph is associated with a timestamp, indicating the exact time instance where the edge was created. The complete set of directional edges for the aforementioned network, along with the associated timestamps, is row-wise stored in the file **sx-stackoverflow.txt** as consecutive triplets of the form **(source_id,target_id,timestamp)**. Thus, the dataset is actually a collection of timestamped edges of the following form

$$E = \{e_{ij}(t) = (v_i, v_j, t) \ : t_{min} \leq t \leq t_{max}\} \tag{1}$$

where t_{min} and t_{max} identify the oldest and latest time instances respectively.

Your analysis should be conducted on a sequence of N **non-overlapping** time periods $\{T_1, \dots T_N\}$, of equal duration δt, that span the entire time interval

$$T = [t_{min}, t_{max}] \tag{2}$$

The set of N time periods may be defined by considering a sequence of $N+1$ time-instances $\{t_0, \dots, t_N\}$ such that:

$$t_j = t_{min} + j * \delta t, \ 0 \leq j \leq N \tag{3}$$

where $\delta t = \frac{\Delta T}{N}$ and $\Delta T = t_{max} - t_{min}$. It is easy to deduce that the j-th time period may be defined as:

$$T_j = \begin{cases} \big[t_{j-1}, t_j\big), & 1 \leq j \leq N-1; \\ \big[t_{j-1}, t_j\big], & j = N. \end{cases} \tag{4}$$

For each time period T_j, with $1 \leq j \leq N$, we may consider the corresponding undirected subgraphs of the network, denoted as:

$$G[t_{j-1}, t_j] = (V[t_{j-1}, t_j], E[t_{j-1}, t_j]) \tag{5}$$

where

$$E[t_{j-1}, t_j] = \{e_{ij}(t) : t \in T_j\} \tag{6}$$

The set $V[t_{j-1}, t_j]$ of vertices for each period may be implicitly defined as the set of nodes that appear at the end points of edges pertaining to the set $E[t_{j-1}, t_j]$.

Part I Questions (Grade Percentage 35%):

[1] https://snap.stanford.edu/data/sx-stackoverflow.html

1. Partition the complete time period $T = [t_{min}, t_{max}]$ into a set of non-overlapping time periods $\{T_1, \ldots, T_N\}$ by computing the corresponding set of time instances $\{t_0, \ldots, t_N\}$ where $t_0 = t_{min}$ and $t_N = t_{max}$. Mind that N is a user defined parameter.

2. Choose an appropriate representation for each subgraph $G[t_{j-1}, t_j]$ of the network for each time period T_j where $1 \leq j \leq N$.

3. Provide a graph depicting the time evolution of the quantities $|V[t_{j-1}, t_j]|$ and $|E[t_{j-1}, t_j]|$ for each time period T_j where $1 \leq j \leq N$.

4. For each subgraph $G[t_{j-1}, t_j]$ compute and graphically represent the probability density functions (i.e. histograms of relative frequencies) for the following centrality measures:

   (a) Degree Centrality

   (b) Closeness Centrality

   (c) Betweenness Centrality

   (d) Eigenvector Centrality

   (e) Katz Centrality

Acquiring a more accurate description for the evolution of the network between successive time periods can be facilitated by considering the set of nodes that persist during the transition from T_j to T_{j+1}, formulated as:

$$V^*[t_{j-1}, t_{j+1}] = V[t_{j-1}, t_j] \cap V[t_j, t_{j+1}],\ 1 \leq j \leq N \tag{7}$$

In this setting, we are particularly interested in restricting the sets $E[t_{j-1}, t_j]$ and $E[t_j, t_{j+1}]$ within the common set of nodes $V^*[t_{j-1}, t_{j+1}]$ as:

$$E^*[t_{j-1}, t_j] = \{(u, v) \in E[t_{j-1}, t_j] : u \in V^*[t_{j-1}, t_{j+1}] \land v \in V^*[t_{j-1}, t_{j+1}]\} \tag{8}$$

$$E^*[t_j, t_{j+1}] = \{(u, v) \in E[t_j, t_{j+1}] : u \in V^*[t_{j-1}, t_{j+1}] \land v \in V^*[t_{j-1}, t_{j+1}]\} \tag{9}$$

Part II Questions (Grade Percentage 35%):

1. For each pair of successive network instances $(G[t_{j-1}, t_j], G[t_j, t_{j+1}])$, where $1 \leq j \leq N-1$, compute the following sets

   (a) $V^*[t_{j-1}, t_{j+1}]$

   (b) $E^*[t_{j-1}, t_j]$

   (c) $E^*[t_j, t_{j+1}]$

   and graphically represent their volumes $|V^*[t_{j-1}, t_{j+1}]|$, $|E^*[t_{j-1}, t_j]|$ and $|E^*[t_j, t_{j+1}]|$ as functions of the coupled time periods (T_j, T_{j+1}).

2. For each pair of nodes $(u, v) \in V^*[t_{j-1}, t_{j+1}]$ and for every set of common vertices $V^*[t_{j-1}, t_{j+1}]$, where $1 \leq j \leq N-1$, compute the following similarity matrices:

   (a) $\mathbf{S_{GD}} : S_{GD}(u, v) = -d_{geodesic}(u, v)$ **[Graph Distance]**

(b) $\mathbf{S_{CN}} : S_{CN}(u,v) = |\Gamma(u) \cap \Gamma(v)|$ [**Common Neighbors**][2]

(c) $\mathbf{S_{JC}} : S_{JC}(u,v) = \frac{|\Gamma(u)\cap\Gamma(v)|}{|\Gamma(u)\cup\Gamma(v)|}$ [**Jaccard's Coefficient**]

(d) $\mathbf{S_{A}} : S_{A}(u,v) = \sum\limits_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{log(|\Gamma(z)|)}$ [**Adamic / Adar**]

(e) $\mathbf{S_{PA}} : S_{PA}(u,v) = |\Gamma(u)| * |\Gamma(c)|$ [**Preferential Attachment**]

According to the previous discussion, it is easy to deduce that during the successive time periods T_j and T_{j+1}, the set of all possible edges between vertices in the common set of nodes $V^*[t_{j-1}, t_{j+1}]$ may be given as:

$$E^0[t_{j-1}, t_{j+1}] = V^*[t_{j-1}, t_{j+1}] \times V^*[t_{j-1}, t_{j+1}],\ 1 \leq j \leq N-1 \tag{10}$$

However, the subset of edges that are actually realized corresponds to the set

$$E^*[t_{j-1}, t_{j+1}] = E^*[t_{j-1}, t_j] \cup E^*[t_j, t_{j+1}] \tag{11}$$

In the context of the link prediction task, $E^*[t_{j-1}, t_j]$ will serve as the training set, whereas $E^*[t_j, t_{j+1}]$ will be used for testing. Each one of the previously defined similarity metrics

$$\mathbf{S_X} :\ X \in \{GD, CN, JC, A, PA\} \tag{12}$$

can be employed in order to implement a simple classification mechanism that provides an estimation for the actual set of edges $E^*[t_{j-1}, t_{j+1}]$ according to the following equation:

$$\hat{E}^*_X[t_{j-1}, t_{j+1}] = \{(u,v) \in E^0[t_{j-1}, t_{j+1}] : S_X(u,v) \in R_X\} \tag{13}$$

where R_X indicates a range of values for the similarity score S_X. The prediction accuracy of each classification rule defined by Eq. 13 can be assessed with respect to a ground truth set of edges E through the utilization of the quantity given below:

$$ACC(R_X, E) = \lambda * TPR(R_X, E) + (1-\lambda) * TNR(R_X, E) \tag{14}$$

where

$$\begin{aligned} TPR(R_X, E) &= \frac{|\hat{E}^*_X[t_{j-1}, t_{j+1}] \cap E|}{|E|} && (15) \\ TNR(R_X, E) &= 1 - \frac{|\hat{E}^*_X[t_{j-1}, t_{j+1}]| - |\hat{E}^*_X[t_{j-1}, t_{j+1}] \cap E|}{|E^0[t_{j-1}, t_{j+1}]| - |E|} && (16) \\ \lambda &= \frac{|E|}{|E^0[t_{j-1}, t_{j+1}]|} && (17) \end{aligned}$$

The simplest way to define R_X is as a continuous interval of the following form:

$$R_X = [S^L_X, S^U_X] \tag{18}$$

where S^L_X and S^U_X are the lower and upper bounds respectively. However, more accurate classification results can be obtained by considering a more composite

[2] For a graph $G = (V, E)$, function $\Gamma : V \rightarrow P(V)$, evaluated on a particular node $z \in V$ provides the subset $\Gamma(z) \subset V$ of nodes neighboring with z

form for the set R_X, composed by the union of n_X non-overlapping intervals, formulated as:

$$R_X = \bigcup_{k=1}^{k=n_X} [S_X^{L_k}, S_X^{U_k}] \tag{19}$$

Part III Questions (Grade Percentage 30%):

1. Describe and implement a training algorithm which determines the optimal range sets R_X^*, defined by Eq. 19, for each similarity measure. The goal of the training algorithm should be the maximization of accuracy given by Eq. 14 within the training set. Therefore, the training algorithm reduces to solving the following maximization problem:

$$R_X^* = \arg\max_{R_X} ACC(R_X, E^*[t_{j-1}, t_j]) \tag{20}$$

2. Having determined the optimal range sets R_X^* for each similarity measure, evaluate and rank the corresponding training accuracy measurements $ACC(R_X^*, E^*[t_{j-1}, t_j])$.

3. Evaluate and rank the testing accuracy measurements $ACC(R_X^*, E^*[t_j, t_{j+1}])$.

For this assignment you can work in groups of no more than 3 students. Your implementation can be in any programming language. The final deliverable should contain:

1. **Well documented code of your implementation.**

2. **A concise report explaining your assumptions and implementation decisions.**

3. **Example runs of your code providing the required graphical representations and classification measurements.**