Continuous-Time Dynamic Network Embeddings

1 CONTINUOUS-TIME DYNAMIC NETWORK EMBEDDINGS - NGUYEN ET AL.

1.1 Goals

- Describe a general framework for incorporating temporal information into network embedding methods
- Methods for learning time-respecting embeddings from continuous-time dynamic networks
- TLDR: Describes a temporal walk strategy

1.2 Preliminaries

- ...

1.3 Challenges

- **General & Unifying Framework**: general framework for incorporating temporal dependencies in node embedding and deep graph models that leverage random walks
- **Continuous-Time Dynamic Networks**: timedependent network representation for continuous-time dynamic networks
- **Effectiveness**: Must outperform baselines

1.4 Previous Work / Citations

- Static Snapshot Graphs:
  - each static snapshot graph represents all edges that occur between a user-specified discrete-time interval (e.g., day or week)
  - Refs: [57, 59, 63, 64]
  - Drawbacks:
    - Noisy approximation on continuous time
    - Selecting appropriate granularity
- **This Work:**
  - Random Walks
  - Supports **graph streams** (edges come and go live)
  - Any work using random walks can benefit from the proposed methods

1.5 Definitions

- **Continuous-Time Dynamic Network**: $G = (V, E_T, \mathcal{T})$
  - E_T edges at continuous times (actually events)
- **Temporal Walk**: temporal walk represents a temporally valid sequence of edges traversed in increasing order of edge times
- **Temporal Neighborhood**: $\Gamma_t(v) = \{(w, t') \mid e = (v, w, t') \in E_T \wedge \mathcal{T}(e) > t\}$
  - Neighbors of a node v at time t

Author's address:

  - Nodes may appear multiple times (multiple edge events)
- –

1.6 Outline / Structure

- Random Walks
  - Changes walk space $\mathbb{S}$ to $\mathbb{S}_T$
- **Goal**: $f : V \rightarrow \mathbb{R}^D$: mapping nodes in G to D-dimensional **time-dependent feature representation**
  - For ml tasks such as link prediction
- **Temporal Walks**:
  - Require starting time (Randomly samples or from a randomly samples edge)
  - Edges from further time may be less predictive (so bias wisely)
  - Has min length ω
- **Biasing the walks**
  - Unbiased: $Pr(e) = 1/N$
  - Biased: Used a distribution based on time
  - Favor newer edges: $\Pr(e) = \frac{\exp[\mathcal{T}(e) - t_{\min}]}{\sum_{e' \in E_T} \exp[\mathcal{T}(e') - t_{\min}]}$
    * Exp dist with $t_m in$ as starting time
- **Biasing Neighbor selection**: Uniform or Biased (bias for time difference for example)
  - Walk bias van be reused based on $\tau(v)$
- Temporal Context Windows:
  - Window count: $\beta = \sum_{i=1}^{k} \left| \mathcal{S}_{t_i} \right| - \omega + 1$
    * Number of walks that can be derived from the window with size ω
- node2vec approach for

1.7 Evaluation

- Try different versions of network (2 main components to swap)
- Baselines: node2vec [26], DeepWalk [52], and LINE [65]. ,
- Datasets from NetworkRepository [58].
-

1.8 Code

- https://github.com/LogicJake/CTDNE
- https://stellargraph.readthedocs.io/en/stable/demos/link-prediction/ctdne-link-prediction.html?highligl

1.9 Resources

- …