Embeddings for Signed Weighted, and Temporal Networks

Vasco Portilheiro (vascop@stanford.edu)

Stanford University

Motivations and Goal

Motivation:

- Little work has been done on embeddings for networks in which edge weights can be non-positive and non-integral.

Task:

- Train embeddings on signed weighted networks, and use edge-embeddings to predict edge weight, aiming to match or outperform the Fairness Goodness Algorithm (Kumar et al).

Approach

- Random walk embeddings using temporal random walks, and a change to the usual skip-gram objective to account for weights.

Data

	Bitcoin OTC	Bitcoin Alpha
Nodes	5881	3783
Edges	35592	24186
SCC	4709	3235
Isolates	0	0
Clustering	0.1775	0.1776

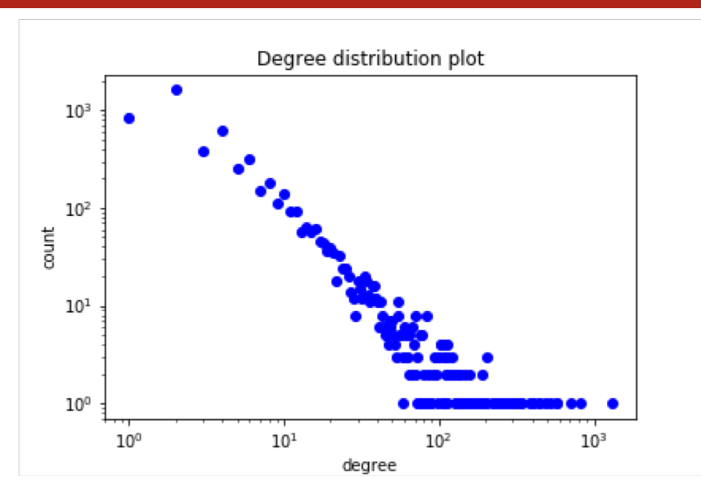

Figure 1: Bitcoin OTC degree distribution

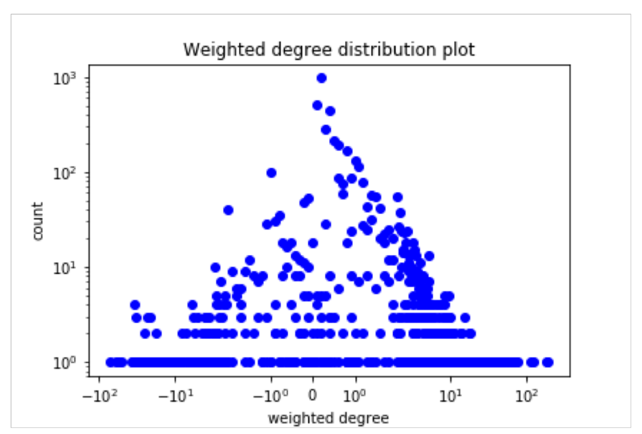

Figure 2: Bitcoin OTC weighted degree dist.

Datasets:

- Two networks used by Kumar et al, consisting of user peer-ratings on a scale from -10 to 10, which are normalized to -1 to 1.
- Networks are fairly small, but show mostly typical social network structure.

Baseline Models

Fairness Goodness:

- Recursively defines two values, the "fairness" and "goodness" of a node. Their product is the predicted weight of an edge.

$$g(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f(u) \times W(u,v)$$

$$f(u) = 1 - \frac{1}{|out(u)|} \sum_{v \in out(u)} \frac{|W(u,v) - g(v)|}{R}$$

DeepWalk/node2vec:

- Random walk embeddings, where walks form the "sentences" for the skip-gram objective.

$$P(\{v_1, \ldots, v_{i-1}, v_i, \ldots, v_\ell\} | v_i) = P(v_1|v_i) \cdots P(v_{i-1|v_i}) P(v_{i+1}|v_i) \cdots P(v_\ell|v_i)$$

$$P(w|v) = \frac{\exp(f(w) \cdot f(v))}{\sum_{w' \in V} \exp(f(w') \cdot f(v))}$$

SNE:

- Random walk embedding for binary signed-weight edges (+1 or -1).

$$\hat{f}(v) = \sum_{i=1}^{\ell} c_i \odot g(u_i)$$

Problem Definition: Edge Embeddings

… From Node Embeddings

- Given a weighted temporal graph $G = (V, E, \mathcal{W}, t)$, where the edge weights are $\mathcal{W}(e) \in [-1, 1]$ and edge times are $t_e \in [0, \infty)$, we use random walk embeddings to learn a node embedding,

$$f : V \to \mathbb{R}^d$$

which we use to create an edge embedding $g : E \to \mathbb{R}^d$, given by the component-wise product:

$$g((u,v)) = f(u) \odot f(v)$$

Temporal Random Walk Embedding

Temporal Random Walk

- A sequence of edges $e_1, \ldots, e_k$ is a temporal random walk if the edge times are such that $t_{e_1} \leq \ldots \leq t_{e_k}$.

Sampling Temporal Walks

- To sample temporal random walks, we first sample a starting node u_1, and the sample uniformly from its neighbors to get the starting time $t_{(u_1,u_2)}$. We then continuously sample neighbors uniformly among those neighbors of the current walk endpoint such that $t_{(u_i,u_{i+1})} \geq t_{(u_{i-1},u_i)}$.

Accounting for Signed Weighted Edges

A different sense of similarity

- Noting that the skip-gram objective is to maximize the probability of nodes co-occurring on a random walk:

$$P(w|v) = \frac{\exp(f(w) \cdot f(v))}{\sum_{w' \in V} \exp(f(w') \cdot f(v))}$$

We can observe that the this objective is trying to maximize the cosine similarity between nodes' embeddings

$$\frac{f(w)}{||f(w)||} \cdot \frac{f(v)}{||f(v)||} = 1.$$

We replace this with the objective the cosine similarity capturing the "similarity" encoded in edge weights

$$\frac{f(w)}{||f(w)||} \cdot \frac{f(v)}{||f(v)||} = \mathcal{W}(w,v) \quad \text{or} \quad \frac{1}{\mathcal{W}(w,v)} \frac{f(w)}{||f(w)||} \cdot \frac{f(v)}{||f(v)||} = 1$$

- If the two nodes aren't neighbors in the walk, we use the principle "the enemy of my enemy is my friend":

$$c_{w,v} = \left(\text{sign}(\mathcal{W}(w,u_1)) \prod_{i=1}^{m} \text{sign}(\mathcal{W}(u_i,v)) \right) \frac{|\mathcal{W}(w,u_1)| + \sum_{i=1}^{m} |\mathcal{W}(u_i,v)|}{m+1}$$

For the partition function, we the the average edge weight $\epsilon \neq 0$

$$P(w|v) = \frac{\exp\left(\frac{f(w) \cdot f(v)}{c_{w,v}}\right)}{\sum_{w' \in V} \exp\left(\frac{f(w') \cdot f(v)}{\epsilon}\right)}$$

Results

	Bitcoin OTC			Bitcoin Alpha		
train-test	80-10	50-40	20-70	80-10	50-40	20-70
FG	**0.313**	**0.321**	**0.336**	**0.263**	**0.275**	0.291
DW	0.332	0.323	0.358	0.288	0.291	0.290
n2v	0.355	0.330	0.345	0.288	0.288	**0.288**
SNE	0.336	0.360	0.407	0.313	0.327	0.335
DW-T	0.321	0.335	0.342	0.273	0.285	0.292
n2v-T	0.320	0.331	**0.336**	0.272	0.284	0.289
SNE-T	0.324	0.338	0.350	0.280	0.287	0.290
DW-RW	0.360	0.365	0.368	0.289	0.296	0.296
DW-RW-T	0.369	0.336	0.370	0.277	0.285	0.290

Root Mean Square Error on Edge-Weight prediction: "-T" denotes temporal random walks, and "-RW' denotes relational weighting (the proposed objective modification)

Evaluation procedure

- For each model (even for Fairness Goodness) 10% of the data was held out to be used as a validation set. Then, starting with an initial test set of 10%, we increased the relative size of the test with to the training set to see how the models performed under data sparsity. For embedding models, edge embeddings were used as inputs to Support Vector Regression.

Analysis

- While embedding methods come close to achieving state of the art performance, only outperform under data-sparse conditions.
- Temporal random walks seem to improve embeddings, but only when enough data is available (since there are necessarily fewer valid temporal walks than regular walks).
- The backpropagation step with the modified objective function turned out to not be very numerically stable, and so the "RW" models were trained with very few both iterations and negative samples, and generally underperformed.

Conclusion

Improvements:

- Train embeddings on independent similarity notions: edge-existence and edge-weight.

References and Acknowledgements

A special thanks to Srijan Kumar for his guidance on this project.

[1] Srijan Kumar et al. "Edge weight prediction in weighted signed networks". In: Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE. 2016, pp. 221– 230.

[2] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: (). doi: 10.1145/2939672.2939754. url: http://dx.doi.org/10.1145/2939672. 2939754.

[3] Shuhan Yuan, Xintao Wu, and Yang Xiang. SNE: Signed Network Embedding. Tech. rep. url: https://arxiv.org/pdf/1703.04837.pdf.