ROSE: Role-based Signed Network Embedding

Amin Javari Tyler Derr Pouya Esmailian Jiliang Tang Kevin Chen-Chuan Chang

May 14, 2020

Overview

1. Motivation
2. Network Embedding Survey
   - Skip-Gram Based Embeddings
   - Signed Network Embeddings
3. ROSE
   - Network Transformation
   - Embedding
   - Results

Motivation

Motivation

- Graphs are universal data structures (maybe quite literally? [Wolfram, 2020])
- But
  1. Have a large computational complexity for storage as well as usage
  2. Low parallelizability
  3. Supervised Machine Learning tasks require handcrafted features for each graph
- We wish to learn dense, continuous and low dimensional representation for nodes [Cui et al., 2019]
- Use these Embeddings to solve many downstream tasks
  - Node importances
  - Community Detection
  - Link Prediction
  - Node classification

Network Embedding Survey

Survey of existing techiques

- A high level view of the existing Network Embedding approaches [Chami et al., 2020]
- Graph Encoder Decoder Model (GraphEDM) as a framework
- Graph $G = (V, E)$, weight matrix $W \in \mathbb{R}^{|V| \times |V|}$
- Optional *node features* of dimension d_0, $X \in \mathbb{R}^{|V| \times d_0}$
- Goal is to learn a vector representation, $Z \in \mathbb{R}^{|V| \times d}$, where $d \ll |V|$

GraphEDM Framework and Objective Functions

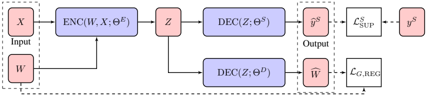


Figure 2: Illustration of the GraphEDM framework. Based on the supervision available, methods will use some or all of the branches. In particular, unsupervised methods do not leverage label decoding for training and only optimize the similarity decoder (lower branch). On the other hand, semi-supervised and supervised methods leverage the additional supervision to learn embeddings (upper branch).

1. Supervised Loss $\mathcal{L}^S_{SUP}$
   - Compares predicted labels $\widehat{y}^S$ to ground truth y^S
2. Graph Regularization loss $\mathcal{L}_{G,REG}$
   - Regularizes model parameters based on graph structure
   - $\mathcal{L}_{G,REG}(W, \widehat{W}; \theta) = d_1(s(W), \widehat{W})$
   - $s(W)$ is target similarity matrix and $d_1(\cdot,\cdot)$ distance function
3. Weight Regularization loss $\mathcal{L}_{REG}$
   - Used as prior, most common Gaussian prior (L2 regularization)
   - $\mathcal{L}_{\mathrm{REG}}(\Theta) = \sum_{\theta \in \Theta} \|\theta\|_2^2$

GraphEDM Total Loss

$$\mathcal{L} = \alpha \mathcal{L}^S_{\mathrm{SUP}}\left(y^S, \hat{y}^S; \Theta\right) + \beta \mathcal{L}_{G,\mathrm{REG}}(W, \widehat{W}; \Theta) + \gamma \mathcal{L}_{\mathrm{REG}}(\Theta)$$

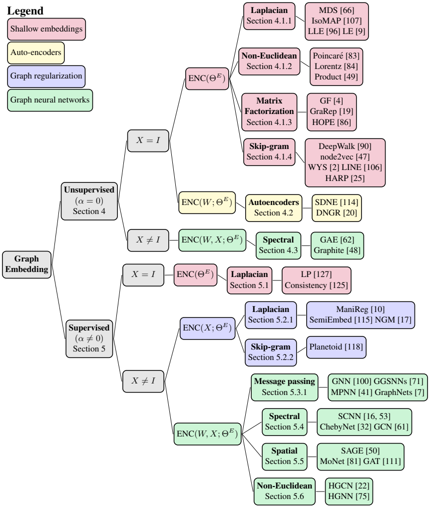


Figure: Taxonomy Proposed by [Chami et al., 2020]

Skip-Gram Based Embeddings

Shallow Encodings

- $Z = ENC(\Theta^E) = \Theta^E \in \mathbb{R}^{|V| \times d}$
- Each node $v \in V$ has a unique encoding vector $\mathbf{z}_v$
- Just a simple embedding lookup
- Similarity between nodes u and v in embedding space is dot product $\mathbf{z}_u^\top \mathbf{z}_v$
- What the similarity will be in the original network?
    - Adjacency
    - Shared Neighbours
    - Structural Roles

Random Walk Embeddings

- Use a random walk strategy R to estimate probability of visiting v when starting from u, $P_R(v \mid u)$
- Train embeddings such that for every pair $\mathbf{z}_u^\top \mathbf{z}_v \propto P_R(v \mid u)$
- Why Random Walks?
  - Incorporate local and global network information
  - Highly efficient to generate random Walks
  - Unsupervised approach
- E.g, DeepWalk [Perozzi et al., 2014] and node2vec [Grover and Leskovec, 2016]

Learn Random Walk Embeddings

- Use *skip-grams* theory from language model literature [Mikolov et al., 2013]
- Neighbourhood $N_R(u)$ for a node u from the random walk strategy R
- Minimize the following loss function $\mathcal{L}$ using SGD

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} - \log\left(P(v \mid \mathbf{z}_u)\right),$$

- Define the probability as a softmax

$$P(v \mid \mathbf{z}_u) = \frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \mathbf{z}_u^\top \mathbf{z}_n}$$

- Drawback is that $\mathcal{L}$ has a quadratic complexity $\mathcal{O}(|V|^2)$

Fix Softmax

Again language model literature has two proposed solutions [Mikolov et al., 2013]

1. Hierarchical Softmax
   - Reduced complexity of computing $P(v \mid \mathbf{z}_k)$ from $\mathcal{O}(|V|)$ to $\mathcal{O}(\log |V|)$
   - Can be further optimized using Huffman encoding etc.
   - E.g. DeepWalk [Perozzi et al., 2014]
2. Negative Sampling
   - Approximate using negative samples $n_i \sim P_V$ and sigmoid function $\sigma(x) = 1/(1+e^{-x})$

$$\log(P(v \mid \mathbf{z}_u)) \approx \log\left(\sigma\left(\mathbf{z}_u^\top \mathbf{z}_v\right)\right) - \sum_{i=1}^{k} \log\left(\sigma\left(\mathbf{z}_u^\top \mathbf{z}_{n_i}\right)\right)$$

   - Sample k negative nodes proportional to the degree
   - Empirically better than hierarchical softmax
   - E.g., node2vec [Grover and Leskovec, 2016]

How to Walk?

1. Unbiased Random Walks [Perozzi et al., 2014]
2. Biased Random Walks [Grover and Leskovec, 2016]
   - Allows to tune between *local* (BFS) and *global* (DFS) walks
   - Add return probability p and walk-away probability q to random walks
   - Low p similar to BFS abd low q similar to DFS

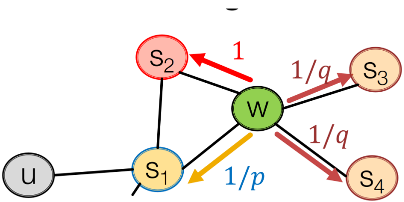


Figure: From lecture slides [Leskovec, 2018]

node2vec Algorithm

1. Choose probabilities p and q and compute random walk probabilities
2. Generate r random walks of length l from each starting node
3. Learn embeddings by optimizing loss with SGD or directly use `gensim` word2vec trainer

Signed Network Embeddings

Additional Hurdles in Signed Grpahs

- Conventional unsigned embedding techniques misunderstand signed edges
- Traditional negative sampling does not work as expected
- Incorporate structural theories of *Balance* and *Status*?
- Considering cycles and long path features
- Learn asymmetric embeddings for directed graphs

Existing Approaches

1. **SiNE** (Signed Network Embeddings) [Wang et al., 2017]

2. **SIDE** (SIgned Directed network Embedding) [Kim et al., 2018]

3. **SIGNet** (SIGned Network embeddings) [Islam et al., 2018]

4. **BESIDE** (Bridge Enhanced Signed Directed Network Embedding) [Chen et al., 2018]

Existing Approaches

1. **SiNE** (Signed Network Embeddings) [Wang et al., 2017]
   - Multilayer Neuaral Network
   - Optimizes objective function that satisfies balance theory
   - Only for undirected graphs
2. **SIDE** (SIgned Directed network Embedding) [Kim et al., 2018]
3. **SIGNet** (SIGned Network embeddings) [Islam et al., 2018]
4. **BESIDE** (Bridge Enhanced Signed Directed Network Embedding) [Chen et al., 2018]

Existing Approaches

1. **SiNE** (Signed Network Embeddings) [Wang et al., 2017]
   - Multilayer Neuaral Network
   - Optimizes objective function that satisfies balance theory
   - Only for undirected graphs
2. **SIDE** (SIgned Directed network Embedding) [Kim et al., 2018]
   - Random Walk based approach
   - Aggregates sign of co-occurring nodes in a walk using balance theory
   - Accordingly changes likelihood for loss term
   - Works for directed and undirected networks
3. **SIGNet** (SIGned Network embeddings) [Islam et al., 2018]
4. **BESIDE** (Bridge Enhanced Signed Directed Network Embedding) [Chen et al., 2018]

Existing Approaches

1. **SiNE** (Signed Network Embeddings) [Wang et al., 2017]
   - Multilayer Neuaral Network
   - Optimizes objective function that satisfies balance theory
   - Only for undirected graphs
2. **SIDE** (SIgned Directed network Embedding) [Kim et al., 2018]
   - Random Walk based approach
   - Aggregates sign of co-occurring nodes in a walk using balance theory
   - Accordingly changes likelihood for loss term
   - Works for directed and undirected networks
3. **SIGNet** (SIGned Network embeddings) [Islam et al., 2018]
   - Also random walk based
   - Uses targeted negative sampling along with balance theory
4. **BESIDE** (Bridge Enhanced Signed Directed Network Embedding) [Chen et al., 2018]

Existing Approaches

1. **SiNE** (Signed Network Embeddings) [Wang et al., 2017]
   - Multilayer Neuaral Network
   - Optimizes objective function that satisfies balance theory
   - Only for undirected graphs
2. **SIDE** (SIgned Directed network Embedding) [Kim et al., 2018]
   - Random Walk based approach
   - Aggregates sign of co-occurring nodes in a walk using balance theory
   - Accordingly changes likelihood for loss term
   - Works for directed and undirected networks
3. **SIGNet** (SIGned Network embeddings) [Islam et al., 2018]
   - Also random walk based
   - Uses targeted negative sampling along with balance theory
4. **BESIDE** (Bridge Enhanced Signed Directed Network Embedding) [Chen et al., 2018]
   - Utilizes bridge edges to encode status theory based features
   - Uses triads to encode balance theory based features
   - Creates a combined loss function and optimizes using mini-batch SGD

ROSE

ROSE: Role-based Signed Network Embedding

What they aim to Fix

1. A general framework not based on social theories
    - Social theories of balance and status are restrictive
    - They might be inaccurate in certain networks
    - May not consider longer cycle features when creating embeddings
2. Utilize missing links as information
    - Most models utilize negative and positive links
    - Cannot predict the absence of links between nodes

ROSE: Role-based Signed Network Embedding

What they propose

- Network transformation based embedding
- Convert signed network to a unsigned bipartite network
- Model each original node using several "role" nodes
- Embed the transformed network using traditional methods
- Aggregate the "role" node embeddings to get original node embeddings

Network Transformation

Network Transformation

1. Transform to Bipartite Network

2. Transform to Unsigned Bipartite Network

3. Augment Network

Network Transformation

1. Transform to Bipartite Network
   - Each node has a "*user*" role for out edges and "*item*" role for in edges
   - Split each node u into u_{out} and u_{in} and map edges
   - Still has signed edges
2. Transform to Unsigned Bipartite Network

3. Augment Network

Network Transformation

1. Transform to Bipartite Network
   - Each node has a "*user*" role for out edges and "*item*" role for in edges
   - Split each node u into u_{out} and u_{in} and map edges
   - Still has signed edges
2. Transform to Unsigned Bipartite Network
   - Split each role u_{in} further into u_{in}^{+} and u_{in}^{-}
   - Remap signed edges as unsigned edges to positive and negative role nodes respectively
   - Can now use traditional similarity measures
3. Augment Network

Network Transformation

1. Transform to Bipartite Network
   - Each node has a "*user*" role for out edges and "*item*" role for in edges
   - Split each node u into u_{out} and u_{in} and map edges
   - Still has signed edges
2. Transform to Unsigned Bipartite Network
   - Split each role u_{in} further into u_{in}^{+} and u_{in}^{-}
   - Remap signed edges as unsigned edges to positive and negative role nodes respectively
   - Can now use traditional similarity measures
3. Augment Network
   - Negative links are under-represented in signed networks
   - Roles of in^{-} has very low degree
   - Assume u_{out} is connected to v_{in}^{+} and w_{in}^{-}
   - This implies v_{in}^{+} and w_{in}^{-} are adjacent **and also** v_{in}^{-} and w_{in}^{+} are related
   - Add dummy u_{out}^{dum} and connect opposite role nodes

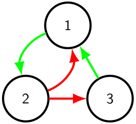
Network Transformation

1. Transform to Bipartite Network
    - Each node has a "*user*" role for out edges and "*item*" role for in edges
    - Split each node u into u_{out} and u_{in} and map edges
    - Still has signed edges
2. Transform to Unsigned Bipartite Network
    - Split each role u_{in} further into u_{in}^{+} and u_{in}^{-}
    - Remap signed edges as unsigned edges to positive and negative role nodes respectively
    - Can now use traditional similarity measures
3. Augment Network
    - Negative links are under-represented in signed networks
    - Roles of in^{-} has very low degree
    - Assume u_{out} is connected to v_{in}^{+} and w_{in}^{-}
    - This implies v_{in}^{+} and w_{in}^{-} are adjacent **and also** v_{in}^{-} and w_{in}^{+} are related
    - Add dummy u_{out}^{dum} and connect opposite role nodes

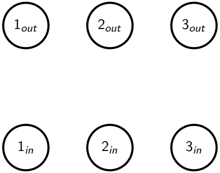
Let's see an example

Network Transformation Example

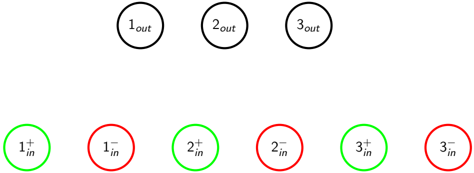
Signed Network



Bipartite Signed Network

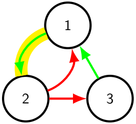


Unsigned Bipartite Role Network

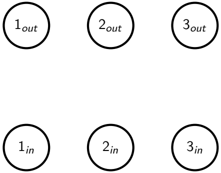


Network Transformation Example

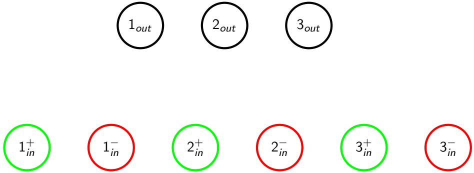
Signed Network



Bipartite Signed Network

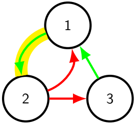


Unsigned Bipartite Role Network

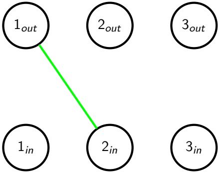


Network Transformation Example

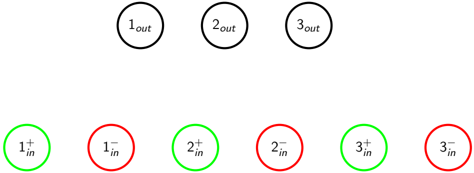
Signed Network



Bipartite Signed Network

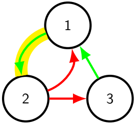


Unsigned Bipartite Role Network

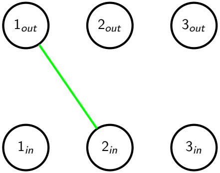


Network Transformation Example

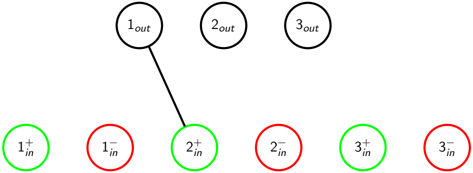
Signed Network



Bipartite Signed Network

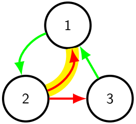


Unsigned Bipartite Role Network

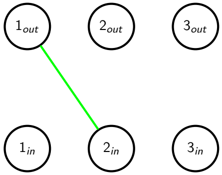


Network Transformation Example

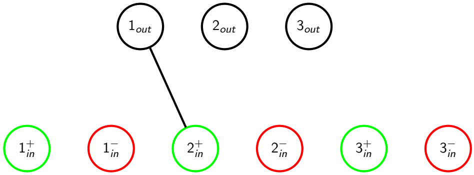
Signed Network



Bipartite Signed Network

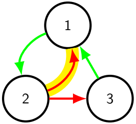


Unsigned Bipartite Role Network

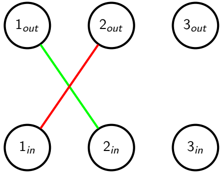


Network Transformation Example

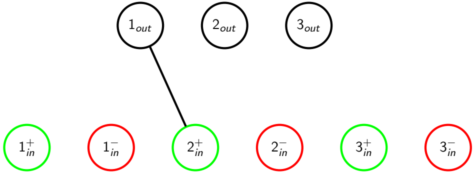
Signed Network



Bipartite Signed Network

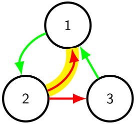


Unsigned Bipartite Role Network

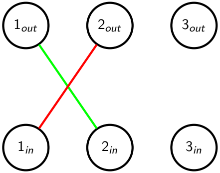


Network Transformation Example

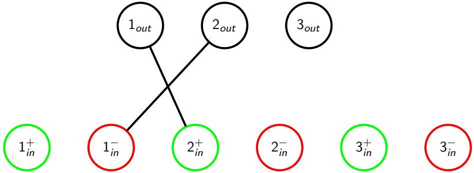
Signed Network



Bipartite Signed Network

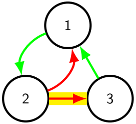


Unsigned Bipartite Role Network

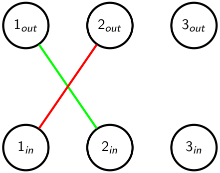


Network Transformation Example

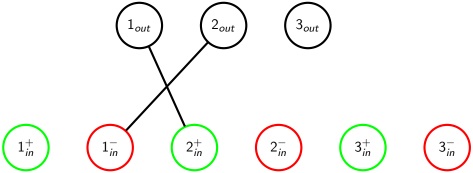
Signed Network



Bipartite Signed Network

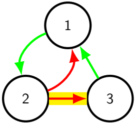


Unsigned Bipartite Role Network

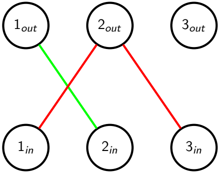


Network Transformation Example

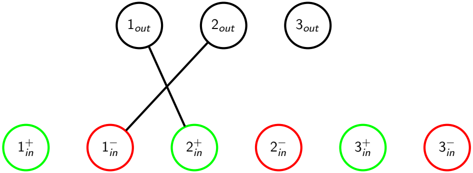
Signed Network



Bipartite Signed Network

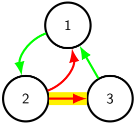


Unsigned Bipartite Role Network

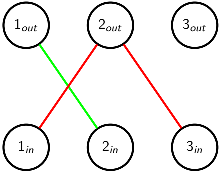


Network Transformation Example

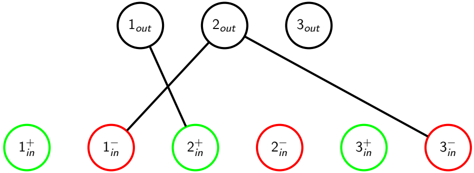
Signed Network



Bipartite Signed Network

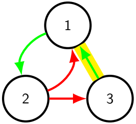


Unsigned Bipartite Role Network

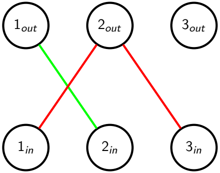


Network Transformation Example

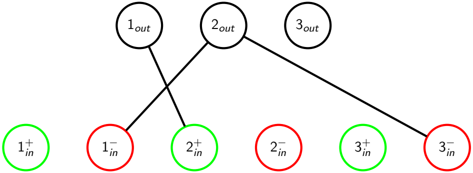
Signed Network



Bipartite Signed Network

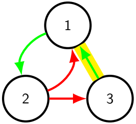


Unsigned Bipartite Role Network

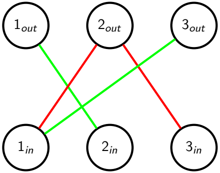


Network Transformation Example

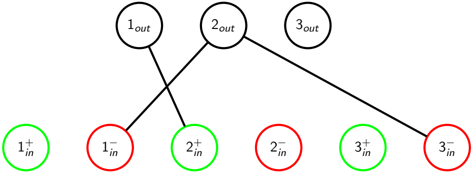
Signed Network



Bipartite Signed Network

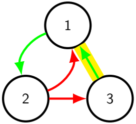


Unsigned Bipartite Role Network

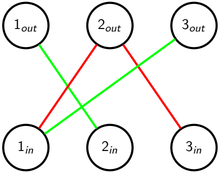


Network Transformation Example

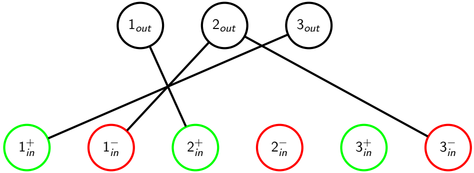
Signed Network



Bipartite Signed Network

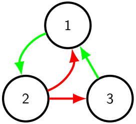


Unsigned Bipartite Role Network

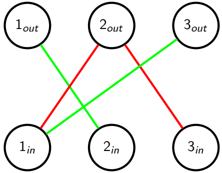


Network Transformation Example

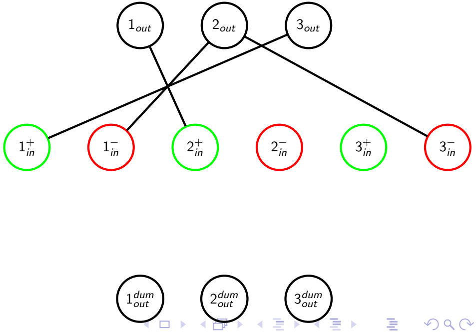
Signed Network



Bipartite Signed Network

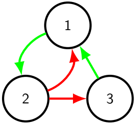


Unsigned Bipartite Role Network

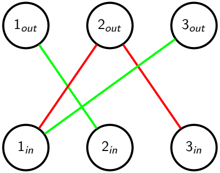


Network Transformation Example

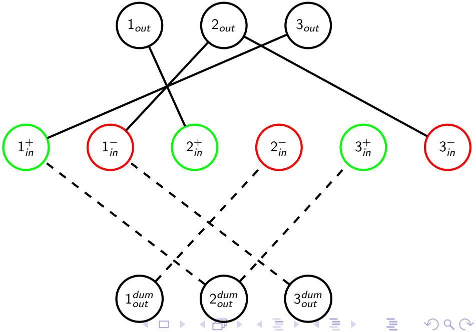
Signed Network



Bipartite Signed Network



Unsigned Bipartite Role Network



Network Transformation Summary

- Graph $G = (V, E)$ transformed to bipartite unsigned $G_u = (V_u, E_u)$
- $|V_u| = 4|V|$ and $|E_u| = 2|E|$
- Each $u \in V$ becomes $u_{out}, u_{in}^{+}, u_{in}^{-}, u_{out}^{dum} \in V_u$
- Each $(u, v) \in E$ with label l becomes (u_{out}, v_{in}^{l}) and $(u_{out}^{dum}, v_{in}^{l'})$
- Transformation is **lossless**

Embedding

Embedding the Network

- After network transformation we have traditional network
- Can employ any classic embedding scheme (see Slide 8)
- Paper uses well known node2vec
- Now each role node has an embedding, $\mathbf{z}_{u_{out}}, \mathbf{z}_{u_{in}^{+}}, \mathbf{z}_{u_{in}^{-}}, \mathbf{z}_{u_{out}^{dum}}$
- Paper proposed two methods to get the embedding of u of the original network

Aggregation Strategies

1. Fixed Aggregation
   - Simplest approach to concatenate all representations
   - $\mathbf{z}_u = [\mathbf{z}_{u_{out}}, \mathbf{z}_{u^+_{in}}, \mathbf{z}_{u^+_{in}}]$
   - Dummy nodes are not used, $\mathbf{z}_{u^{dum}_{out}}$ is inverse of $\mathbf{z}_{u_{out}}$
2. Target Aware Aggregation
   - Useful for tasks such as link prediction
   - Analogous to item-based collaborative filtering: predict rating of users towards a particular item
   - Use a weighted combination based on similarity to target item
   - Embed the "out" node u_{out} wrt target node v, $\mathbf{z}^v_{u_{out}}$
   - Combine target aware embedding along with personal embedding to get u wrt v

$$\mathbf{z}^v_u = [\mathbf{z}^v_{u_{out}}, \mathbf{z}_u].$$

Attention Embedding

- Compute $\mathbf{z}^{v}_{u_{out}}$ as a weighted sum of the neighbours u_{out}
- Each neighbour $w^{l}_{in} \in N(u_{out})$, where l is either $+$ or $-$

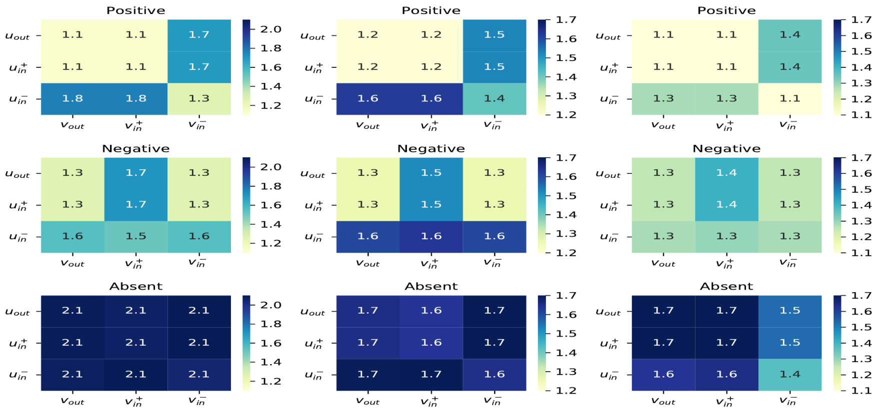
$$\mathbf{z}^{v}_{u_{out}} = \sum_{w^{l}_{in} \in N(u_{out})} attn(w^{l}_{in}, v)\mathbf{z}_{w^{l}_{in}},$$

- Weight $attn(w^{l}_{in}, v)$ is how relevant that neighbour is towards the target node v
- Intuition is that "in" nodes are more related if closer in network
- Label is not considered, so need $\mathbf{z}_{w_{in}}$ for every node
- Ignore signs of original network and convert to bipartite and get node2vec embeddings

$$attn(w^{l}_{in}, v) = \sigma(\mathbf{z}_{w_{in}}, \mathbf{z}_{v_{in}}) = \frac{1}{1 + \exp(-\mathbf{z}^{\top}_{w_{in}}\mathbf{z}_{v_{in}})}.$$

Results

Results



(a) Epinions (b) Slashdot (c) Wikipedia

Figure: The average pairwise distance of the encoding vectors of the role-nodes of a node pair (u, v) for different interaction types between them: positive link, negative link, and absence of a link.

Interpretations of Role Nodes

- Label of link from u to v is l if $\mathbf{z}_{u_{out}}$ is closer to $\mathbf{z}_{v_{in}^{l}}$ than $\mathbf{z}_{v_{in}^{l'}}$
- If $\mathbf{z}_{u_{out}}$ is further apart from both $\mathbf{z}_{v_{in}^{+}}$ and $\mathbf{z}_{v_{in}^{-}}$, then the link is absent
- See $d_{avg}(u_{out}, v_{in}^{+})$ compared to $d_{avg}(u_{out}, v_{in}^{-})$
- $d_{avg}(u_{out}, v_{in}^{+}) + d_{avg}(u_{out}, v_{in}^{-})$ is smaller when there is a link compared to when there is no link.
- There are **four** *implicit patterns* apart from this
- They analyse situations for a pair of nodes u and v

Intersting Patterns

1. *Pattern 1*: Incoming Edges
   - "If the sign of the link from u to v is positive, similar nodes rate them similarly and if it is negative, similar nodes rate them with different signs"
   - If edge $u \xrightarrow{+} v$, incoming edges are similar
   - $d_{avg}(u^+_{in}, v^+_{in})$ and $d_{avg}(u^-_{in}, v^-_{in})$ are smaller
   - If edge $u \xrightarrow{-} v$, incomes edges are of different signs
   - $d_{avg}(u^+_{in}, v^+_{in})$ and $d_{avg}(u^-_{in}, v^-_{in})$ are larger
   - Pattern aligns with balance theory
   - Triads in balance theory are special case of this pattern

results

Intersting Patterns

1. *Pattern 2*: Outgoing Edges
   - "u and v rate similar nodes more similarly when there is a positive link between them than when there is a negative a link connecting them."
   - When edge $u \xrightarrow{+} v$, $d_{avg}(u_{out}, v_{out})$ is small and vice-versa
   - Balance theory is a special case of this pattern
2. *Pattern 3*: Revered Direction Edges
   - Sign of link in the opposite direction are correlated
   - When $u \xrightarrow{+} v$, then $d_{avg}(v_{out}, u^{+}_{in}) - d_{avg}(v_{out}, u^{-}_{in})$ is smaller than when $u \xrightarrow{-} v$
   - Therefore, opposite edge is positive if forward edge is positive
   - Contradictory to status theory
3. *Pattern 4*: Absent Edges
   - If there is no link all role nodes are far away from each other
   - These role nodes are not tightly connected and belong to different clusters

results

Questions or Comments

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. (2020).
Machine learning on graphs: A model and comprehensive taxonomy.

Chen, Y., Qian, T., Liu, H., and Sun, K. (2018).
"bridge": Enhanced signed directed network embedding.
In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 773–782, New York, NY, USA. Association for Computing Machinery.

Cui, P., Wang, X., Pei, J., and Zhu, W. (2019).
A survey on network embedding.
IEEE Transactions on Knowledge and Data Engineering, 31(5):833–852.

Grover, A. and Leskovec, J. (2016).
node2vec.
Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Islam, M. R., Aditya Prakash, B., and Ramakrishnan, N. (2018).

Signet: Scalable embeddings for signed networks.
Lecture Notes in Computer Science, page 157–169.

Kim, J., Park, H., Lee, J.-E., and Kang, U. (2018).
Side: Representation learning in signed directed networks.
In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 509–518, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Leskovec, J. (2018).
Analysis of network course.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
In *Advances in neural information processing systems*, pages 3111–3119.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014).
Deepwalk.

Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14.

Wang, S., Tang, J., Aggarwal, C., Chang, Y., and Liu, H. (2017).
Signed network embedding in social media.
In Chawla, N. and Wang, W., editors, *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017*, Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017, pages 327–335. Society for Industrial and Applied Mathematics Publications.
17th SIAM International Conference on Data Mining, SDM 2017 ; Conference date: 27-04-2017 Through 29-04-2017.

Wolfram, S. (2020).
A class of models with the potential to represent fundamental physics.