291-Quality-Aware Streaming Network Embedding with Memory Refreshing

Hsi-Wen Chen¹, Hong-Han Shuai², Sheng-De Wang¹, and De-Nian Yang³,⁴

¹ Department of Electrical Engineering, National Taiwan University
² Department of Electrical and Computer Engineering, National Chiao Tung University
³ Institute of Information Science, Academia Sinica
⁴ Research Center for Information Technology Innovation (CITI), Academia Sinica
Outline

- Introduction
- Related Works
- Problem Formulations
- Challenges
- Methods
- Theoretical Analysis
- Experiments
Introduction

- Learn a low-dimensional vector representation through graph topology
- Application [6,7,23]
  - Node Classification
  - Link Prediction
  - Network Visualization
- The real world is dynamic [8,9,11]
  - Average number of friends on Facebook increases from 155 in 2016 and to 338 in 2018
  - Correctly estimate the influence vertices from the updated edges
  - Efficiently update the corresponding network embedding
Related Works

- **Static Network Embedding**
  - Skip Gram [6,7,23]
  - AutoEncoder [10,22,27]
  - Graph Neural Network [13,18,16]
  - *Not design for the dynamic networks with multi-type change*

- **Dynamic Network Embedding**
  - Extend the static network embedding methods [8,11]
  - Take the sequence of networks over time to learn the temporal behaviors
  - *Without considering the efficiency (whole network vs. change part)*
Problem Formulations

**Definition 1. (Streaming Networks).** A dynamic network $G$ is a sequence of networks $G = \{G_1, \cdots, G_T\}$ over time, where $G_t = (V_t, E_t)$ is the network snapshot at timestamp $t$. $\Delta G_t = (\Delta V_t, \Delta E_t)$ represents the streaming network with the changed part $\Delta V_t$ and $\Delta E_t$ as the sets of vertices and edges inserted or deleted between $t$ and $t+1$.

**Definition 2. (Streaming Network Embeddings).** Let $z_{i,t}$ denote the streaming network embedding that preserves the structural property of $v_i \in G_t$ at timestamp $t$. The streaming network embeddings are derived by $\Phi^s = (\phi^s_1, \cdots, \phi^s_{t+1}, \cdots, \phi^s_T)$, where $\phi^s_{t+1}$ updates the node embedding $z_{i,t+1}$ at timestamp $t+1$ according to $z_t$ and $\Delta G_t$, i.e., $z_{i,t+1} = \phi^s_{t+1}(z_t, \Delta G_t)$, where $z_t = \{z_{i,t} | \forall v_i \in V_t\}$. 
Problem Formulations

**Definition 3. (Quality-aware Multi-type Streaming Network Embeddings).** Given a streaming network with $\Delta V_t$ and $\Delta E_t$ as the sets of the vertices and edges inserted or deleted between $t$ and $t + 1$, the goal is to find the streaming network embedding and derive the corresponding embedding quality to ensure that the nodes with similar structures share similar embeddings.
Challenges

- Multi-type change
- Evaluation of affected nodes
- Transduction
- Theoretical analysis
Graph Memory Refreshing

- Multi-type Embedding Updating
  - Enhance the skip gram algorithm under the multi-type change

- Hierarchical Addressing
  - Allocate the node to update efficiently

- Refresh and Percolate
  - Interpolate previous embedding with the newly arrived information
  - Propagate the refreshed embedding to its neighborhoods
Multi-type Embedding Updating

\[ \sum_{(v_i, v_j) \in E} \sigma(z_i^T z_j) + \sum_{v_i \in V} \mathbb{E}_{v_j \sim P_N(v_i)}[\sigma(-z_i^T z_j)] + \alpha \sum_{(v_i, v_j) \in D} \sigma(-z_i^T z_j) \]

- **positive sample**
- **randomly negative sample**
- **deletion sample**

- The deletion samples carry more information than unlinked pair samples[6]
  - We also prove that the \( \alpha \) should be greater than 1 in our theoretical analysis
Hierarchical Addressing

- Global addressing [18]
  - All nodes
  - Computationally intensive
- Local addressing [27]
  - Only neighboring nodes
  - The ripple-effect area usually has an arbitrary shape
- Hierarchical addressing
  - Transform a large classification into a series of binary classification ($O(|V|)$ -> $O(\log(|V|))$)
  - Adopt the beam search on the addressing tree
Tree Contraction-Graph Coarsening

- Merged two leaf node to a super node
Simliarity Search

- Extract the top-k results at each iteration with the beam search
  - To prevent over-fitting in a local minimum
Refresh and Percolate

- Parameterize the update of the embedding
  - Refreshing gate $g_r$
  - Percolation gate $g_p$
- Act as deciders to learn the impact of the streaming network
  - Compute the correlation between current and incoming embedding

$$\rho_r = \frac{a_r}{a_r}^T [z_{q,t+1} \mid z_{i,t}]$$

$$g_r = \sigma(\rho_r)$$

$$z_{i,t+1} \leftarrow g_r z_{q,t+1} + (1 - g_r) z_{i,t}$$
Isomorphic Retaining Score

Definition 4. (Isomorphic Pair). Any two different nodes $v_i$ and $v_j$ form an isomorphic pair if the sets of their first-hop neighbors $N_1(.)$ are the same.

Lemma 2. The embeddings $z_i$ and $z_j$ are the same after GMR converges if and only if $(v_i, v_j)$ is an isomorphic pair.

Definition 5. (Isomorphic Retaining Score). The isomorphic retaining score, denoted as $S_t$, is the summation of the cosine similarity over every isomorphic pair in $G_t$, $S_t \in [-1, 1]$. Specifically,

$$S_t = \frac{1}{|\xi_t|} \sum_{(v_i, v_j) \in \xi_t} s_{ij, t},$$

(5.1)
Theoretical Result

**Theorem 1.** GMR outperforms other Skip-Gram-based models regarding the isomorphic retaining score under edge insertion after each update by gradient descent.
Time complexity

Time Complexity. The time complexity of GMR is $O(kd_{\text{max}}|\Delta V_t| + k|\Delta V_t| \log(|V_t|))$, while retraining the whole network requires $O(|V_t| \log(|V_t|))$ time at each timestamp. Since $k$ is a small constant, $d_{\text{max}} \ll |V_t|$, and $|\Delta V_t| \ll |V_t|$, GMR is faster than retraining.
Experiment Setup

- Hyperparameter
  - Deletion weight, $\alpha = 1$
  - Search Bean, $k = 3$
- Stochastic gradient descent (SGD) with Adagrad is adopted to optimize the loss function.

$$
\sum_{(v_i, v_j) \in E} \sigma(z_i^T z_j) + \sum_{v_i \in V} \mathbb{E}_{v_j \sim P_N(v_i)} [\sigma(-z_i^T z_j)] + \alpha \sum_{(v_i, v_j) \in D} \sigma(-z_i^T z_j)
$$
Link Prediction

- Facebook
  - 63,731 nodes
  - 1,269,502 edges
  - 736,675 timestamps
- Yahoo
  - 100,001 nodes
  - 3,179,718 edges
  - 1,498,868 timestamps
- Epinions
  - 131,828 nodes
  - 841,372 edges
  - 939 timestamps

|                | Facebook |       |       | Yahoo |       |       | Epinions |       |       |
|----------------|----------|-------|-------|-------|-------|-------|----------|-------|-------|
|                |  AUC     |  S    |  sec  |  AUC  |  S    |  sec  |  AUC     |  S    |  sec  |
| GMR            | 0.7943   | 0.94  | 3325  | 0.7674| 0.93  | 3456  | 0.9294   | 0.92  | 3507  |
| FULL           | 0.8004   | 0.95  | 66412 | 0.7641| 0.95  | 72197 | 0.9512   | 0.96  | 61133 |
| CHANGE         | 0.6926   | 0.79  | 2488  | 0.6326| 0.82  | 2721  | 0.8233   | 0.84  | 2429  |
| GraphSAGE      | 0.6569   | 0.77  | 4094  | 0.6441| 0.79  | 5117  | 0.8158   | 0.85  | 4588  |
| SDNE           | 0.6712   | 0.81  | 7078  | 0.6585| 0.83  | 7622  | 0.8456   | 0.88  | 6799  |
| CTDNE          | 0.7091   | 0.85  | 4322  | 0.6799| 0.84  | 5136  | 0.8398   | 0.90  | 5097  |
| DNE            | 0.7294   | 0.87  | 2699  | 0.6892| 0.86  | 2843  | 0.8648   | 0.92  | 2613  |
| SLA            | 0.7148   | 0.86  | 2398  | 0.6910| 0.86  | 2438  | 0.8598   | 0.91  | 2569  |
| DHPE           | 0.7350   | 0.88  | 3571  | 0.7102| 0.88  | 3543  | 0.8458   | 0.90  | 3913  |
Node Classification

- **BlogCatalog**
  - 10,132 nodes
  - 333,983 edges
  - 39 classes
- **Wiki**
  - 2,405 nodes
  - 17,981 edges
  - 19 classes
- **DBLP**
  - 101,253 nodes
  - 223,810 edges
  - 4 classes
  - 48 timestamps

* denotes the simulation data[27]
Summary

- Outperform all six baselines in both link prediction and node classification.
  - Accuracy (20%) and Efficiency (20 times)
- Isomorphic retaining score $S$ is highly related to both task-oriented scores.
  - Correlation coefficient is 0.92
1. Cao, S.; Lu, W.; and Xu, Q. Grarep: Learning graph representations with global structural information. In CIKM, 2015
2. Y. Zhao and P. S Yu. On Graph Stream Clustering with Side Information. In SDM, 2013
3. Chen, C. and Tong. H. Fast Eigen-Functions Tracking on Dynamic Graphs. In SDM, 2015
4. Li, J.; Dani, H.; Hu, X.; Tang, J. Chang, Y.; and Liu. H. Attributed Network Embedding for Learning in a Dynamic Environment. In CIKM, 2017
5. Li, J.; Kewei, C.; Wu, L.; and Liu. H. Streaming link prediction on dynamic attributed networks. In WSDM, 2017
6. Perozzi, B.; Al-Rfou, R.; and Skiena, S. Deepwalk: Online learning of social representations. In SIGKDD, 2014
7. Grover, A.; and Leskovec, J. node2vec: Scalable feature learning for networks. In SIGKDD, 2016
8. Giang H. N.; J. B. Lee; Ryan A. R; Nesreen K. A.; Eun-yee. K.; and Sungchul K. Continuous-Time Dynamic Network Embeddings. In WWW, 2017
9. Zhou, L; Yang, Y; Ren, X; Wu, F and Zhuang, Y. Dynamic Network Embedding by Modelling Triadic Closure Process, In AAAI, 2018

10. Wang, D.; Cui, P.; and Zhu, W. Structural deep net-work embedding. In SIGKDD, 2016

11. Zhou, L; Yang, Y; Ren, X; Wu, F and Zhuang, Y. Dy-namic Network Embedding by Modelling Triadic Closure Process, In AAAI, 2018

12. Jian, L.; Li, J.; and Liu, H. 2018. Toward Online Node Classification on Streaming Networks. In DMKD, 2018.

13. Kipf T. N.; and Welling, M. Semi-supervised classification with graph convolutional networks. In ICLR, 2017

14. Velikovi, P.; Cucurull, G.; Casanova, A.; Romero, A.; Li, P.; and Bengio, Y. Graph Attention Networks. In ICLR, 2017

15. Manessi, F. Rozza, A. Manzo M. Dynamic Graph Convolutional Networks. In arXiv, 2017

16. Weston, J; Chopra, S.; and Bordes, A. Memory networks. In ICLR, 2015

17. Hamilton, W. and Ying, Z. and Leskovec, J. Inductive representation learning on large graphs. In NIPS, 2017

18. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. In arXiv, 2014
19. Jolliffe, I. Principal component analysis. In Springer, 2011
20. Hyvärinen, A.; Karhunen, J.; and Oja, E.. Independent component analysis. In 2004.
21. Roweis, S. T. and Saul, L. K.. Nonlinear dimensionality reduction by locally linear embedding. In Science, 2000.
22. Cao, S.; Lu, W; and Xu, Q. Deep neural networks for learning graph representations., In AAAI, 2016
23. Tang, J.; Qu, M.; Wang, M.; Zhang, M., Yan; J., and Mei, Q. Line: Large-scale information network embedding. In WWW, 2015
24. Tenenbaum, J B., Silva, V. D., and Langford, J. C.. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000. In Science, 2009.
25. De Lathauwer, L.; De Moor, B.; and Vandewalle, J.. A multilinear singular value decomposition. In JMAA, 2005
26. Benesty, J.. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. In JASA, 2000
27. Du, L., Wang, Y., Song, G., Lu, Z., Wang, J.: Dynamic network embedding: An extended approach for skip-gram based network embedding. In: IJCAI, 2018