

THS-GWNN: A Deep Learning Framework for Temporal Network Link Prediction

Xian Mo ¹, Jun Pang ², Zhiming Liu (✉)¹

¹ College of Computer & Information Science, Southwest University, Chongqing 400715, China

² Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette L-4364, Luxembourg

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2020

1 Introduction and main contributions

Link prediction for temporal networks aims to evaluate the likelihood of the future linkage among nodes, which has significant applications in social networks, biological networks and traffic analysis [1], etc. Network embedding [4] is an important analytical tool for temporal network link prediction, which helps us better understand network evolution [2]. How to encode high-dimensional and non-Euclidean network information is a crucial problem for node embedding in temporal networks. One of the challenges is to reveal the spatial structure at each timestamp and the temporal property over time [3]. Some existing work [6] shows that extracting the spatial relation of each node can be used as a valid feature representation for each node. Moreover, the emergence of deep learning techniques [3, 6] brings new insights for learning temporal properties, but most models using deep learning still fail to achieve satisfying prediction accuracy.

In this paper, we propose a graph wavelet neural network (THS-GWNN) framework, based on timestamp hierarchical sampling, for link prediction in temporal networks.

Our major contributions in this work can be summarised as follows. (1) We propose a model THS-GWNN to perform link prediction in temporal networks. The model adopts a Graph Wavelet Neural Network (GWNN) [4] to deeply embed nodes, which can better capture the nonlinear features of temporal networks. (2) We propose a timestamp hierarchical sampling algorithm (THS) for both spatial and temporal feature extraction. More precisely, it samples neighbors for the current node v from the current snapshot's K -hop neighbours to the previous snapshots' K -hop neighbours, which can hierarchically extract both spatial and temporal features for the nodes. It also incorporates a decaying exponential to assign the more sampled nodes to fewer hops and closer snapshots, which can better preserve the evolving behavior of temporal networks. (3) Experiments on real-world network datasets (i.e., Hep-Ph, Digg, Facebook friendships and Facebook wall posts¹⁾) demonstrate that our THS-GWNN outperforms a few state-of-the-art baseline models.

2 Problem Definition

Definition 1 [Network] A network can be represented graphically: $G = \langle V, E \rangle$, where $V = \{v_1, \dots, v_n\}$ represents a set of nodes, and n is the number of nodes, and $E \subseteq V \times V$ repre-

Received month dd, yyyy; accepted month dd, yyyy

E-mail: zhimingliu88@swu.edu.cn

¹⁾<http://konect.uni-koblenz.de/>

sents a set of links (edges) among nodes.

Definition 2 [Temporal network] We follow the temporal network settings in [2] that the set of nodes is fixed, while the edges E_t can evolve over time. Hence, a temporal network is defined as $G_t = \langle V, E_t \rangle$, which represents a network $G = \langle V, E \rangle$ evolving over time and generates a sequence of snapshots $\{G_1, \dots, G_T\}$, where $t \in \{1, \dots, T\}$ represents the timestamps.

Definition 3 [Node K -hop neighbours] Let $G_t = \langle V, E_t \rangle$ be a temporal network. For a node v in timestamp t , its K -hop neighbours can be defined as the set $N^t(v, K) = \bigcup_{k=1}^{K=K} \{R^k(v, A_t^k)\}$, which contains some of the the neighbours of v with K hops in the timestamp t . The value k represents the hop, and $R^k(v, A_t^k)$ represents the set of randomly sampled neighbors of v in hop k in timestamp t and A_t^k represents the number of neighbors sampled for v in hop k in timestamp t and $|R^k(v, A_t^k)| \leq A_t^k$, where $|R^k(v, A_t^k)|$ represents the number of neighbors sampled for $R^k(v, A_t^k)$. We set $\widehat{A}_t^{k-1} = \gamma A_t^{k-1}$ and $\widehat{A}_{t-1}^1 = \gamma A_{t-1}^1$ ($t > 1$), where γ is a decaying exponential between 0 and 1 and \widehat{A} represents rounding up A .

Definition 4 Let $G_t = \langle V, E_t \rangle$ be a temporal network. For a node v , its all K -hop neighbours from time t_{Sta} up to time t_{End} (i.e., $t_{Sta} \leq t_{End}$) are defined as $\Gamma(v, K, t_{Sta}, t_{End}) = \bigcup_{t=t_{Sta}}^{t_{End}} \{N^t(v, K)\}$, where t represents a timestamp, K represents the number of the hops, and $N^t(v, K)$ is a set of K -hop neighbours of v in a network snapshot G_t where $t \in \{t_{Sta}, \dots, t_{End}\}$.

Link prediction for temporal networks. For a temporal network G_t that generates a sequence of snapshots $\{G_1, \dots, G_T\}$, we respectively use to adjacency matrix $\{A_1, \dots, A_T\}$ to describe its static topological structure. For the A_t (t represents a timestamp), it is a 2-D array that stores the vertex relationships. The element in the A_t can be represented as a_{ij} , where i and j represent rows and columns of the 2-D array, respectively. If $a_{ij} = 0$, there is no edge between vertex i and j , otherwise, there is an edge. Temporal link prediction aims at predicting the adjacency matrix A_{T+1} at timestamp $T + 1$ according to the previously adjacency matrix $\{A_1, \dots, A_T\}$.

propose a timestamp hierarchical sampling (THS) algorithm to extract spatial-temporal features. The extracted spatial-temporal features are then fed into the GWNN for network embedding. Finally, LSTMs [5] are adapted to predict new links. Because of the page limit, all technical details are given in the supplementary file.

3.1 Spatial-temporal feature extraction

Since the current snapshot topological structure of temporal networks is derived from the previous snapshot topology, it is necessary to combine the previous snapshot to extract the spatial-temporal features for the current snapshot. Inspired by this idea, we propose the THS algorithm to sample $\Gamma(v, K, t_{Sta}, t_{End})$ for each node v of each snapshot (see its details in Definition 3 and Definition 4).

The THS algorithm has four input parameters: A_t^1 : the number of sampled neighbors of v in 1-hop at current timestamp t , K : the number of hops defining the neighbours of distance from one to at most K to a given node v , L : a window size defining how many previous network snapshots are taken into account when sampling v 's neighbours and γ : the decaying exponential defining the fewer hops and the closer snapshots of the current snapshot to have more sampled nodes. We define $X[i]$ to represent the sampled neighbours for all nodes in the network at timestamp i , where $X[i]$ with $i \in \{1, \dots, T\}$. The THS algorithm extracts spatial-temporal features from previous L snapshots. If the number of the previous snapshots is greater than L , sampling neighbour nodes is performed between $i - L + 1$ and i snapshots. Otherwise, it is only sampled from the very first snapshot 1 to the current snapshot i . In this way THS can extract both spatial and temporal features for each node and better simulate the evolutionary behavior of the temporal network.

3.2 Embedding layer

In the model, we adopt GWNN [4] to map nonlinearly a node v to its D -dimensional representation $x_v \in R^D$. The GWNN was originally applied to static networks, and we design a m -layer GWNN for each snapshot for unsupervised node learning for temporal networks embedding. The formulation of our model for each snapshot is

$$X_{[:,k]}^{m+1} = ReLU(\psi_s \sum_{i=1}^q F_{i,j}^m \psi^{-1} X_{[:,i]}^m) \quad k = 1, \dots, c \quad (1)$$

3 THS-GWNN

In this section, we introduce our model, THS-GWNN, for temporal network link prediction. In our model, we first

where $X_{[:,i]}^m$ are the i -th column of X^m (with the dimension $q \times 1$, m represents layers, and X^m is the input of the GWNN) $ReLU$ is a non-linear activation function, ψ_s is wavelet bases, ψ^{-1} is the graph wavelet transform matrix at scale s which projects signal in vertex domain into spectral domain, $F_{i,j}^n$ is a diagonal filter matrix learned in spectral domain in layer n [4], c is the embed dimension of each node, X^{m+1} of dimensions $n \times c$ is the embedding matrix of networks. We define the following loss function for each snapshot embedding to train the model:

$$Loss = \frac{1}{n} \sum_{i=1}^n (X_{[:,i]}^{m+1} - Average(Adj(X_{[:,i]}^{m+1})))^2 \quad (2)$$

where n is the number of nodes, and $X_{[:,i]}^{m+1}$ represents the vector representation of the node i in layer $m + 1$, and $Adj(X_{[:,i]}^{m+1})$ obtains neighborhood node representation of i . *Average* means an average processing operation.

3.3 LSTM layer

In this paper, we adopt LSTM [5] to predict new links. We adopt the $\{Z_1, \dots, Z_T\}$ to train the model, where each Z_t with $t \in \{1, \dots, T\}$ represents the output of the GWNN at the t time snapshot. After training, we first shift the window one step towards the future to obtain the vector representation of each node for the last snapshot. Then we train a downstream support vector machine classifier to evaluate link prediction.

Table 1 Prediction results for the four datasets (AUC value).

Model	Hep-Ph	Digg	Facebook wall posts	Facebook friendships
STEP	0.61	0.74	0.76	0.57
NetWalk	0.69	0.71	0.74	0.70
T-GCN	0.70	0.75	0.72	0.71
THS-GWNN	0.74	0.82	0.78	0.73

To validate the performance of our THS-GWNN model, we compare it with SETP [2], T-GCN [3], and NetWalk [6] on four real-world networks. The detailed experimental results are shown in Table 1. Our method, THS-GWNN, achieves the best performance. The average AUC value of THS-GWNN is 9.75% higher than SETP, 4.25% higher than T-GCN, and 5.75% higher than NetWalk. We also conducted an ablation study in the first two datasets: the average AUC value is 2.5% higher than that replaces the GWNN [4] unit

with the GCN [3] unit and 4.5% higher than that replaces the THS algorithm with the PinSage sampling strategy [7].

4 Conclusions

In this paper, we have propose an effective framework, THS-GWNN, for link prediction in temporal networks. In particular, we propose the THS algorithm to extract both spatial and temporal features to model network evolution. We then adopt GWNN to embed the spatial-temporal features into vectors to better capture nonlinear features. Finally, we use LSTM to capture the time dependence among network snapshots for link prediction. Compared with other related algorithms, THS-GWNN achieves good performance on real-world network datasets.

Acknowledgements This work has been supported by Chongqing Graduate Student Research and Innovation Project (Grants No. CYB19096), the China Scholarship Council (Student No. 202006990041), the Fundamental Research Funds for the Central Universities (XDJK2020D021), the Capacity Development Grant of Southwest University (Grants No. SWU116007), and the National Natural Science Foundation of China (Grant No. 61672435, 61732019, 61811530327)

References

1. B. Li, D. Pi. Network representation learning: A systematic literature review. *Neural Computing and Applications*, 2020:7209.
2. H. Chen, J. Li. Exploiting structural and temporal evolution in dynamic link prediction. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018: 427-436.
3. L. Zhao, Y. Song, et al. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2019: 1-11.
4. B. Xu, H. Shen, et al. Graph wavelet neural network. In: *Proceedings of the 7th International Conference on Learning Representations*, 2019: 1-13.
5. S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, 1987, 9(8): 1735-1780.
6. W. Yu, W. Cheng, et al. NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In: *Proceedings of the 24th ACM Conference on Knowledge Discovery and Data Mining*. 2018:2672-2681.
7. R. Ying, R. He, et al. Graph convolutional neural networks for web-scale recommender Systems. In: *Proceedings of the 24th ACM Conference on Knowledge Discovery and Data Mining*, 2018: 974-983.