Type Sequence Preserving Heterogeneous Information Network Embedding*

Yuxin Chen,[1,2] **Tengjiao Wang,**[1,2,3] **Wei Chen,**[1,2] **Qiang Li,**[4] **Zhen Qiu**[4]
[1]Key Lab of High Confidence Software Technologies(MOE), School of EECS, Peking University, Beijing, China, 100871
[2]Center for Computational Social Science, Peking University, Beijing, China, 100871
[3]Center for Data Science, Peking University, Beijing, China, 100871
[4]State Grid Information and Telecommunication Group, Beijing, China, 102209
{chen.yuxin, tjwang, pekingchenwei}@pku.edu.cn, {liqiang, qiuzhen}@sgitg.sgcc.com.cn

Abstract

Lacking in sequence preserving mechanism, existing heterogeneous information network (HIN) embedding discards the essential type sequence information during embedding. We propose a Type Sequence Preserving HIN Embedding model (SeqHINE) which expands the HIN embedding to sequence level. SeqHINE incorporates the type sequence information via *type-aware GRU* and preserves representative sequence information by *decay function*. Abundant experiments show that SeqHINE can outperform state-of-the-art even with 50% less labeled data.

Introduction

Network embedding maps nodes into low-dimensional vectors which can be used as the feature input for various downstream analyses. Existing approaches mainly focus on homogeneous networks. Compared with homogeneous networks, heterogeneous information networks (HINs), consisting of multi-typed entities and relations, have been demonstrated as a more efficient way to model real-world data (Shi et al. 2017). Since the various type information carries rich semantics, HIN embedding is worthwhile and challenging.

Abundant intrinsic semantic information was lost in the existing HIN embedding process, especially *type sequences*. Type sequence, a series of entity types and relation types, expresses rich semantic relationships between nodes. For example, the type sequence "$Actor \xrightarrow{actIn} Movie \xrightarrow{belongTo} Genre$" indicates the genre of film the actor starred in. Based on this type sequence, existing methods (Dong, Chawla, and Swami 2017; Fu, Lee, and Lei 2017) generate a node set as the context: $Leonardo\ DiCaprio \xrightarrow[Extracting]{Context} \{Titanic,\ TheGreatGatsby,\ Inception,\ Sciencefiction,\ Romantic\}$. The sequence information is lost in the context extracting phase. The better context is a sequence set: $Leonardo\ DiCaprio \xrightarrow[Extracting]{Context} \{Leonardo \xrightarrow{actIn} Titanic \xrightarrow{belongTo} Romantic,\ Leonardo \xrightarrow{actIn} TheGreatGatsby \xrightarrow{belongTo} Romantic,\ Leonardo \xrightarrow{actIn} Inception \xrightarrow{belongTo} Sciencefiction\}$. Due to lack of the sequence maintaining mechanism in the existing methods, the type sequence information is not fully exploited. We believe this partly account for the application accuracy bottleneck of the existing HIN embedding. It is thus of great need to study type sequence preserving HIN embedding.

The challenges are two folds: 1) *how to preserve*: Existing embedding methods cannot maintain complex type sequence information, so an effective model is required. 2) *which to preserve*: The theoretically optimal way is to retain all the type sequence information in the HIN. As the length of the sequence increases, the cost grows exponentially. Thus which to preserve is worth exploring.

To preserve type sequences, our model (**SeqHINE**) is designed as a novel encoder-decoder framework with *type-aware GRU* to incorporate the sequence and type information. For the second challenge, representativeness of the type sequence is highly related with the distance to the current node. Intuitively, the closer the better. We devise a *nearest first policy* and further design the *decay function* for various hop-count distances to implement it.

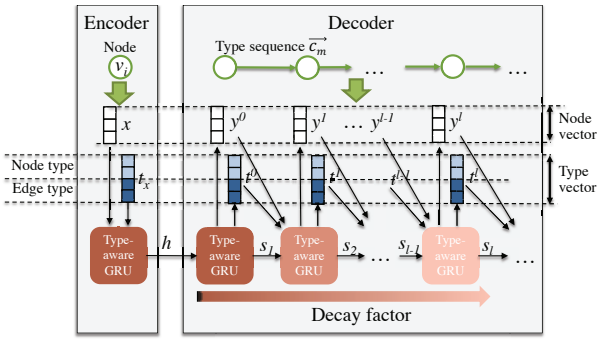

Figure 1: SeqHINE model.

SeqHINE Model

SeqHINE is an encoder-decoder model with *type-aware GRU* and *decay function* to embed HINs into low-dimensional vectors. SeqHINE, as shown in Figure 1, has an encoder to map the node v_i into an embedding vector and

*This work was supported by Natural Science Foundation of China (No.61572043) and State Grid Technical Project (No. 52110418002W). Tengjiao Wang is the corresponding author.

a decoder to reconstruct the surrounding type sequence $\vec{c}_m$. *Type-aware GRU* learns a hidden state h representing the embedded node v_i by incorporating its type information t_x in the encoder and predicts the current component c_m^l of the type sequence by leveraging the previous type information t^{l-1} in the decoder. *Decay function* introduces a decay factor to combine *type-aware GRU* of each time step to adopt the *nearest first policy*. The objective of our model is to find node representations that are useful to predict surrounding type sequences.

Type-aware GRU We propose *type-aware GRU* as a new type of hidden unit. *Type-aware GRU* incorporates the type and sequence information into the hidden state to represent the type sequence processed so far. As shown in Figure 2, the current hidden state h_l is derived from the node vector x^l of the type sequence component c_m^l and the previous time step hidden state h_{l-1}, as well as the type vector t^l which consists of the node type and edge type. The type information also affects the update gate and reset gate. We introduce matrices $\vec{T}_u, \vec{T}_z$ and $\vec{T}$ to bias the update gate u, reset gate z and hidden state computation by the type vector t^l.

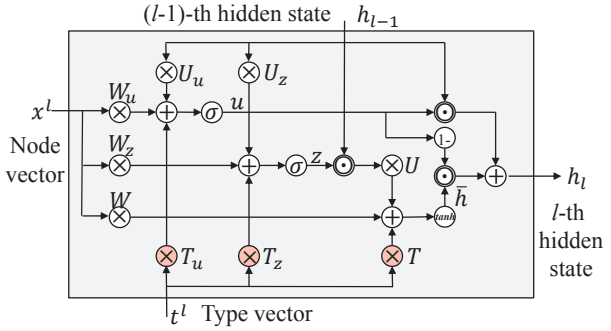

Figure 2: Type-aware GRU.

Decay Function We design a *nearest first policy* that works as follows: the closer to the current node it is, the higher priority it has. To adopt this policy, *decay function* introduces a decay factor w.r.t. the hop-count distance to reduce the weights of the latter part of the sequence. Let α be a decay factor controlling the rate of decay for different distances at each time step. By combining the conditional probability of each time step with the decay function, the conditional probability of the sequence $\vec{c}_m$ is defined as:

$$p(\vec{c}_m|v_i) = \prod_{l=1}^{L} p(c_m^l|c_m^{l-1}, ..., c_m^1, \vec{h})^{e^{-\alpha(l-1)}} \quad (1)$$

. When $l = 1$, $e^{-\alpha(l-1)} = 1$, which do not affect the results. As the hop-count increases, its impact on the conditional probability of c_m^l drops exponentially.

Experiments

We validate the effectiveness of SeqHINE over state-of-the-art: two representative network embedding methods, LINE (Tang et al. 2015) and Node2vec (Grover and Leskovec 2016), and two latest HIN embedding approaches, Metapath2vec (Dong, Chawla, and Swami 2017) and HIN2Vec (Fu, Lee, and Lei 2017), on tasks in two representative real-world HIN datasets (DBLP and PubMed, shown in Table 1).

Table 1: Statistics of two datasets.

Datasets	#(Author)	#(Paper)	#(Conference)	#(Term)	$\|V\|$	$\|E\|$
DBLP	767	5,158	798	8,439	15,162	120,573
PubMed	689	754	198	5,972	7,613	28,968

Results Table 2 shows the multi-label classification results on the DBLP and PubMed datasets. The training ratio varies from 2% to 80%, which is similar to baselines. Results show that: 1) SeqHINE performs better than all the baselines. With only 40% of the labeled nodes, the Micro-F1 performance outperforms all the baselines when they are given 80% of the nodes. In other words, SeqHINE can outperform the baselines with 50% less labeled data. 2) Given 80% of the labeled nodes, the Micro-F1 and Macro-F1 of SeqHINE show a 9%-17% increase. It means that our performance significantly increases for largely labeled data. It is not surprising because various type sequences encode sequence semantic insights which are very helpful for HIN embedding especially in the presence of abundant labels.

Table 2: Results of multi-label classification

Metric	Algorithm	PubMed				DBLP			
		2%	4%	6%	8%	20%	40%	60%	80%
Micro-F1	LINE	19.91	25.59	29.76	38.59	50.55	58.59	59.64	61.38
	Node2vec	30.89	40.30	43.04	47.00	45.20	48.48	51.45	50.86
	Metapath2Vec	36.72	40.91	39.93	43.90	55.28	57.51	57.89	61.39
	HIN2Vec	29.76	36.92	44.73	49.47	52.51	53.87	54.42	53.82
	SeqHINE	**38.34**	**42.73**	**46.48**	**50.11**	**58.41**	**66.85**	**70.40**	**74.52**
Macro-F1	LINE	8.93	16.99	18.35	22.85	27.91	36.23	37.40	37.69
	Node2vec	11.07	18.48	23.03	26.40	30.42	36.42	38.90	36.13
	Metapath2Vec	16.31	20.11	25.57	24.75	22.62	24.75	24.98	28.38
	HIN2Vec	11.93	15.38	20.90	22.42	23.85	28.87	27.41	25.34
	SeqHINE	**16.55**	**21.79**	**26.18**	**26.91**	**30.58**	**37.22**	**39.49**	**41.11**

Conclusion

Our model expands the HIN embedding to sequence level by capturing type sequence information. Diverse experiments demonstrate the effectiveness of the type sequence preserving methodology, especially for largely labeled data.

References

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 135–144. ACM.

Fu, T.-y.; Lee, W.-C.; and Lei, Z. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, 1797–1806. ACM.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*, 855–864. ACM.

Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; and Philip, S. Y. 2017. A survey of heterogeneous information network analysis. *TKDE* 29(1):17–37.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*, 1067–1077.