FID3018 - Paper Review

Modern Techniques for Dynamic Graph Representation Learning

Lodovico Giaretta
lodovico@kth.se

March 2021

1 Introduction

The field of Graph Representation Learning (GRL) has now been very active for several years, thanks to the increase in the amount of graph-structured data that needs to be collected and analyzed in many different domains. However, most of the techniques developed are based on static graphs that encode which relationships exist (or have been seen until now), but not how they evolved over time. However, in many domains, the overall dynamics that drive the creation and destruction of relationships can change with time, and thus the future behaviour of the graph can only be accurately predicted by taking into account the temporal dimension. For this reason, recent works have focused on how to introduce and exploit the temporal dimension in GRL, giving birth do the sub-field of Dynamic GRL.

This review will focus on three of the most recent papers in this sub-field:

- [1] Goyal et al., *dyngraph2vec: Capturing Network Dynamics using Dynamic Graph Representation Learning*, Knowledge-Based Systems 2020
- [2] Sankar et al., *DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks*, WSDM 2020
- [3] Fathy et al., *TemporalGAT: Attention-Based Dynamic Graph Representation Learning*, PAKDD 2020

All three these approaches present some similarities.

- They are based on *graph snapshots*; that is, the temporal evolution of the graph is presented to the model by feeding it with several different snapshots of the entire graph, each taken at a different point in time. This can be seen as a "discretized" version of the alternative approach, not considered in this review, which consists in feeding the model with a continuous stream of changes, such as edge insertions and edge deletions.
- They employ well-known deep learning-based techniques from the static GRL field, such as Graph Autoencoders and Graph Neural Networks (GNNs). This is in contrast with some other works in both the static and dynamic GRL areas, which compute shallow embeddings that can be reconducted to the basic problem of matrix factorization.

2 dyngraph2vec

2.1 Approach

Similarly to their previous approach DynGEM [4], Goyal et al. use an encoder-decoder architecture to predict the adjacency matrix at the next snapshot. However, instead of simply encoding the information from the previous snapshot, as in DynGEM, the dyngraph2vec encoder takes as input the adjacency information from the last L snapshots. The paper introduces three separate variants of this approach.

The first, simplest variant, called dyngraph2vecAE, consists of a traditional encoder-decoder architecture with dense layers. This approach, however, presents issues in terms of the number of parameters requires, with the first layer scaling as $O(NLD)$, where N is the number of nodes in graph (and thus in the adjacency row of the target node in each snapshot), L is the lookback size (the number of past snapshots to pass as input) and D is the output size of the first input layer (i.e., the number of neurons, each of which is connected to every input). This does not take into account the fact that each block of N inputs represents one adjacency row, and thus should be treated similarly. Furthermore, the temporal direction is not explicitly modeled, but left to guess by the training.

To overcome these limitations, the authors introduce dyngraph2vecRNN, which replaces the dense layers with LSTMs. This reduces the number of parameters of the first layer to $O(ND + D^2)$, making it independent of L. However, this architecture requires the LSTMs to directly input and output sparse adjacency vectors, a task for which they might not be suited. Therefore, a third model, called dyngraph2vecAERNN is introduced, which uses dense layers to convert sparse adjacency vectors to low-dimensional dense representations and vice-versa, while keeping LSTM layers in the deepest part of the encoder.

The authors show that the different variants of dyngraph2vec match or outperform each selected baseline in link prediction tasks, with dyngraph2vecAERNN outpeforming the other variants.

2.2 Strong Points

The authors perform a thorough investigation of their architecture, identifying limitations and presenting iterative upgrades, achieving better and better quality. Furthermore, an extensive study of hyperparameter sensitivity shows that the better results achieved by their latest architecture are consistent across datasets and hyperparameter settings.

2.3 Weak Points

The approaches introduced in the paper present several limitations. First, the use of adjacency vectors from a single node as input limits the expressive power of the system. The encoder-decoder does not have access to the wider structure of the graph and is not resilient to graph permutations. Furthermore, while possible (as discussed in [4]), increasing the size of the model to accommodate node additions is not straightforward.

A second limitation caused by the direct use of the adjacency matrix is low scalability. The adjacency vectors are directly passed to the model, which therefore scales in number of weights with the size of the network. The sparsity of the adjacency information is not exploited. This can be seen in the experiments, which require subgraph sampling and

cannot scale to long time windows, despite the small size of the datasets employed.

Finally, while this paper was first made available as a preprint in 2018, it was only accepted for publication in 2020. In those two years, the field of Dynamic GRL has evolved substantially, rendering most of the comparisons in this paper outdated. While the selected baselines share most of the limitations of dyngraph2vec, more recent approaches can easily overcome them and produce better results.

3 DySAT

3.1 Approach

As the title of the paper suggests, DySAT makes extensive use of attention mechanisms to automatically capture the relevant information from the stream of snapshots. The first step consists in applying a Graph Convolutional Network with self-attention independently on each snapshot in the window, therefore extracting structural information about each node at each timestamp. The formulation of this step with additive self-attention is similar to the Graph Attention Network proposed in [5].

The second step consists in applying self-attention to the sequence of structural embeddings of a single node over time, in order to obtain a final embedding that represents the evolution of that node over the time window. A scaled dot-product attention mechanism is used, inspired by the Tranformer architecture proposed in [6].

The entire model is trained end-to-end using a "Graph Context Prediction" task. Inspired by the unsupervised learning objectives of well-known NLP and GRL techniques such as DeepWalk [7], it is an extension of link prediction which focuses not only on the immediate neigbours of a node, but on its more wider neighbourhood area, by predicing nodes that will co-appear with it in a random walk window.

Extensive comparisons demonstrate the superiority of DySAT to both static and dynamic baselines, including dyngraph2vecAERNN, both in prediction quality and computational efficiency.

3.2 Strong Points

Compared to previous approaches such as dyngraph2vec, the use of more advanced and robust attention-based mechanisms allow DySAT to exploit the sparsity of the graph adjacency, encode the wider neighbourhood structure of each node in the embeddings and only require a constant model size, independent from the number of nodes in the graph or the number of snapshots in a window. Thanks to this DySAT is both more expressive and more scalable than previous techniques.

The paper presents an extensive evaluation section, with multiple research questions clearly highlighted and answered exhaustively. An ablation study, a hyperparameter sensitivity study and a scalability study further solidify the claims in the paper.

3.3 Weak Points

One aspect that was not covered by the ablation study, and whose impact therefore remains unclear, is the graph context prediction task used instead of link prediction to optimize the model. Considering the results obtained by the TemporalGAT paper discussed below, it appears that simple link prediction might be sufficient to achieve good results in downstream dynamic link prediction tasks.

4 TemporalGAT

4.1 Approach

Fathy et al. introduce an approach that is extremely similar to DySAT. TemporalGAT also employs (as the name suggests) graph self-attention based on [5] to extract structural embeddings at each timestamp, and it also feeds the embeddings of the same node at different timestamps to a second model that computes one final embedding.

However, this second part of the network is what distinguishes TemporalGAT. Fathy et al. choose to use a dilated Temporal Convolutional Network (TCN) [8] to perform the temporal aggregation task. Generally speaking, TCNs represent an interesting alternative to traditional recurrent models (e.g. RNNs, LSTMs) for dealing with sequential data. Compared to most recurrent models, dilated TCNs can more easily model long-range relationships, thanks to their peculiar backpropagation structure.

Another difference of TemporalGAT compared to DySAT is the use of a more traditional link prediction task, rather than the graph context prediction introduced by the previous approach.

Experimental results show that TemporalGAT is marginally better than DySAT on most datasets.

4.2 Strong Points

This paper demonstrates that complex graph context prediction optimization and high-capacity temporal attention models may not be necessary to achieve good results in dynamic link prediction tasks. Simple link prediction optimization and dilated TCNs (which are simple, low-capacity models) may provide equivalent or better results.

One advantage of this type of simplification, which is unfortunately not evaluated in the paper, is the lower model size, faster training and faster inference that these smaller, simpler models can provide.

4.3 Weak Points

As hinted above, the paper lacks an analysis of the size and speed benefits of replacing temporal attention with a TCN and removing the graph context prediction optimization task. Given that the link prediction results of TemporalGAT are only marginally better than DySAT, the performance improvement of TemporalGAT would have made the paper stronger and the model itself more appealing for future research.

Another issue with the paper is the quality of its presentation and of the language used. This issue makes it harder for the reader to go through the paper and properly understand how the authors employed the dilated TCN within their model.

5 Conclusion

Dynamic Graph Representation Learning is quickly gaining traction, as many real-world graphs are indeed dynamic and the papers reviewed in this report show how traditional static GRL techniques fail to achieve good results. The research community is aware of this, with more and more papers on dynamic GRL being published in recent years.

One characteristic of the specific set of papers reviewed in this report is the reuse of existing, well-established techniques that are known to work well to solve specific sub-problems: Graph Autoencoders and Graph Attention Networks to extract structural representations from each snapshot, recurrent models, Transformers and TCNs to model temporal evolution across multiple snapshots. Thus, without "reinventing the wheel", strong dynamic approaches can be easily built and tested.

Another positive characteristic of these work is that they can be easily compared. Due in part to the novelty of the specific sub-field and to the scarcity of high-quality, publicly-accessible temporal interaction networks, each new technique is evaluated against all previous relevant approaches, using few well-established datasets and tasks. This is in stark comparison with the larger GRL field, where the abundance of methods, datasets and tasks renders exhaustive comparisons prohibitive.

References

[1] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187:104816, 2020.

[2] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 519–527, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Ahmed Fathy and Kan Li. Temporalgat: Attention-based dynamic graph representation learning. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 413–423, Cham, 2020. Springer International Publishing.

[4] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. *CoRR*, abs/1805.11273, 2018.

[5] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[7] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[8] Shaojie Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018.