Noname manuscript No.
(will be inserted by the editor)

DyHANE: Dynamic Heterogeneous Attributed Network Embedding

Liliana Martirano · Roberto Interdonato · Dino Ienco · Andrea Tagarelli

Received: date / Accepted: date

1 Introduction

As real world scenarios are inherently dynamic, graph continual learning is a machine learning paradigm gaining increasing popularity in recent years. Furthermore, most applications involve a multiplicity of entities and relationships with associated attributes, which should be captured and exploited effectively. This work considers the open problem of learning representations of changed nodes, i.e., new or updated nodes at current timestamp, without re-training the model from scratch or negatively affecting the learned representations of existing nodes at the previous timestamps.

As noted in [1], methods based on Graph Neural Networks (GNNs) are able to provide more refined graph representations, higher flexibility in leveraging attributes and generalization to unseen nodes, although they might suffer from more restrictive computational requirements with consequent impact on scalability w.r.t. shallow approaches. In dynamic scenarios, assuming a complete graph representation before the training process begins, is not applicable; pretrained models exploiting the inherent induction capability of GNNs fail in integrating new knowledge; also, retraining the model only on the changed nodes, although using the parameters learned at the previous timestamp to initialize, does not guarantee the preservation of previous knowledge. Several GNN-based works store all the past history of nodes and apply an additional recurrent architecture or an attention mechanism to update the node repre-

Liliana Martirano
Università della Calabria, Rende, Italy
E-mail: liliana.martirano@dimes.unical.it

Roberto Interdonato
CIRAD, UMR TETIS, Montpellier, France
E-mail: roberto.interdonato@cirad.fr

Dino Ienco
INRAE, UMR TETIS, Montpellier, France
E-mail: dino.ienco@inrae.fr

Andrea Tagarelli
Università della Calabria, Rende, Italy
E-mail: andrea.tagarelli@unical.it

sentations [2–5]. Literature lacks continual learning approaches for feature-rich heterogeneous networks capable of harnessing the expressive power of GNNs.

To address the above problem, we propose a novel framework, namely DyHANE (Dynamic Heterogeneous Attributed Network Embedding) framework, whose key idea is to learn at each new timestamp t an up-to-date node representation by incrementally updating model parameters, based on nodes of both the current and the previous timestamps, avoiding the need to store the history of nodes. To this aim, we propose a novel strategy to detect and integrate new knowledge, i.e., the changed nodes, and combine rehearsal and regularization approaches by employing experience nodes replay, i.e., nodes of previous timestamps stored in memory as experience and replayed at current timestamp, and model regularization for existing knowledge consolidation. DyHANE is designed for networks that may be dynamic, heterogeneous and attributed at the same time. We take into account node/edge addition and removal on feature-rich heterogeneous networks, i.e., networks showing multiple types of nodes and/or edges, and having external content associated to nodes.

2 Proposed framework

We define a *dynamic heterogeneous attributed graph* at a generic timestamp t as $G^t = \langle \mathcal{V}^t, \mathcal{E}^t, A, R, \phi, \varphi, \mathcal{X}^t \rangle$, where $\mathcal{V}^t$ and $\mathcal{E}^t$ are the sets of nodes and edges at timestamp t, A and R are the (fixed) sets of node and relation types, with $|A| + |R| > 2$, $\phi : \mathcal{V}^t \to A$ and $\varphi : \mathcal{E}^t \to R$ are the node- and edge-type mapping functions, and $\mathcal{X}^t$ is the matrix storing node attributes at time t.
We define the network evolution over time as a set of events at each timestamp t, corresponding to the set of changed edges ${\mathcal{E}_c}^t = \bigcup_{i,j} e_{ij}$, with $e_{ij} = (i, x_i, j, x_j, r, s)$. In our formulation, $s = 1$ (resp. $s = 0$) denotes the addition (resp. removal) of edge of type $\varphi(e) = r$ between v_i and v_j; x_i and x_j denote the attribute vectors of v_i and v_j. New nodes are initialized with their corresponding attribute vector, while nodes already existing in the network can be updated. The attribute vector is null if there are no changes. To handle attribute changes of isolated nodes, i.e. nodes not involved in any addition or removal, we define the corresponding event as a particular self loop.

Assuming a discrete set of timestamps $\{t_1, t_2, ..., t_T\}$, each of which corresponding to a set of events and consequent changes in graph data, our goal is to incrementally learn $\{\theta^{t_1}, \theta^{t_2}, ..., \theta^{t_T}\}$ where θ^t are the GNN parameters at timestamp t able to generate good representations ${z_i}^t \forall v_i \in G^t$.

The core of DyHANE consists of three main steps reiterated at each new timestamp t (Algorithm 1): (i) identification of the minimum set of nodes affected by changes, hereinafter referred to as influenced nodes $\mathcal{I}^t$, (ii) update of the GNN parameters θ^t using both the influenced nodes $\mathcal{I}^t$ and the replay nodes stored in the memory buffer, hereinafter referred to as memory buffer or experience buffer $\mathcal{B}^{t-1}$, (iii) update of the memory buffer $\mathcal{B}^t$ from the influenced node set $\mathcal{I}^t$. We point out that the proposed algorithm is relatively flexible, as it is not constrained to a single GNN architecture or to a unique im-

Algorithm 1 DyHANE - updating

Require: Set of events, i.e., changed edges ${\mathcal{E}_c}^t$ at current timestamp t, experience node buffer $\mathcal{B}^{t-1}$ stored at previous timestamp $t-1$, GNN parametrized by θ^{t-1} learned at previous timestamp $t-1$, number of epochs num_epochs.
Ensure: GNN parametrized by θ^t learned at current timestamp t, updated experience buffer $\mathcal{B}^t$.

Obtain influenced node set $\mathcal{I}^t$ from ${\mathcal{E}_c}^t$
Load the experience buffer $\mathcal{B}^{t-1}$
Compute parameter importance by estimating F Fisher Information matrix
$i = 0$
while i $<$ num_epochs **do**
    Calculate loss function $\mathcal{L}_{tot} = \gamma\mathcal{L}_{new} + (1-\gamma)\mathcal{L}_{ex}$
    Update parameters using SGD
    $i = i + 1$
end while
Update the experience buffer $\mathcal{B}^t$ using the influenced node set $\mathcal{I}^t$
return θ^t

plementation of individual steps. While proposing tailored strategies, we also suggest possible alternatives to be investigated according to specific needs. In the following, we will describe in detail the steps of the proposed approach.

Identification of influenced nodes. To identify the minimum set of influenced nodes $\mathcal{I}^t$, we classify the new events in the graph (i.e., added or removed edges) into *strong* and *weak* events, according to their impact on the network topology. Inspired by a recent work on graph representation learning for dynamic homogeneous networks [6] and motivated by the well-established superiority of using meta-path based models to capture heterogeneous information even in large networks [7,8], we mark an event as strong or weak according to the impact of the corresponding edge on the meta-path based adjacency matrix of target type, i.e., the node type targeted for a task at hand. A meta-path is a path of the form $a_1 \xrightarrow{r_1} a_2 \xrightarrow{r_2} ... \xrightarrow{r_x} a_{x+1}$ describing a composite relation between two nodes of the same type, i.e., $\phi(a_1) = \phi(a_{x+1})$. We thus define for the selected target node type a unique (homogeneous) meta-path based graph obtained as union of all meta-path instances with terminal nodes of that type, by removing the intermediate nodes and establishing a link between the terminal nodes weighted on the number of meta-path instances connecting them. The new event is said *strong* if it adds a new entry to the corresponding meta-path based adjacency matrix, and the set of influenced nodes will include, in addition to the terminal nodes of the edge, also their one-hop neighborhood and meta-path based neighborhood; otherwise, it is said *weak* event. Notice that other strategies relying on the graph structure, such as triadic closure or open processes proposed by [9] for homogeneous networks and extended by [10] to multiple node/edge types, can be employed for differentiating between strong and weak events.

Update of the memory buffer. To update the memory buffer $\mathcal{B}^t$, we borrow from [11] the idea of selecting the most representative nodes for each class among the influenced node set, which has been proven to improve the effectiveness and stability of data replaying, i.e., the experience of previous timestamps used at the current one. Since the influenced node set is expected to be unbalanced, the idea is to store in the buffer the same amount of information for each node class, ensuring stable distribution of memory categories. For nodes of target type, both strategies of leveraging node attributes (mean of features and coverage maximization) and of estimating node influence (influence maximization) proved to be effective in the homogeneous case [11]. Dealing with heterogeneous networks, we further propose an extension to handle nodes of types different from the target one. Specifically, we select for each class the nodes that appear most frequently as intermediate nodes in meta-paths instances with terminal nodes of that class. We also foresee a further balancing to ensure the presence of all types of nodes in memory, giving more importance to the nodes of target type, and representing the other types proportionally to their presence in the influenced node set. We plan to test also the opposite strategy of selecting nodes with attributes different from their neighborhood and storing in memory the nodes located at the class boundary, assuming they contribute more to the gradient [12]. We also do not exclude the investigation of further strategies independent of the final classification task.

Update of GNN parameters. Regardless of the specific strategies employed, we incrementally train the new model parameters on both nodes updated at the current timestamp and nodes updated at previous timestamp, i.e., the training set at timestamp t is $\mathcal{I}^t \bigcup \mathcal{B}^{t-1}$. We want to weigh the two contributions similarly to [11] and extend the weight factor to the heterogeneous case taking into account all the node types $a \in A$: $\gamma^a = \frac{|\mathcal{B}^a|}{|\mathcal{I}^a|+|\mathcal{B}^a|}$. Note that the number of nodes in $\mathcal{I}$, varying at each timestamp, is usually significantly larger than the size of the experience buffer, which is instead of fixed size. To solve the overfitting problem caused by the small number of replayed nodes, we introduce a weighted regularization method for model parameters based on Fisher Information matrix, that can be approximated based only on nodes stored in memory [12]. We aim to guarantee that the distance between the current and the historical model parameters will not deviate further, giving different importance to different parameters to keep small the changes of GNN parameters that are important to the past network while the others can be updated more drastically. For this purpose, the Elastic Weight Consolidation (EWC) regularization-based method [13] has shown to be effective.

Summary We have proposed DyHANE, a continual learning framework able to generate up-to-date node representations for dynamic, heterogeneous and attributed network. The proposed framework incrementally updates GNN parameters integrating new knowledge while consolidating the existing one.

We selected some real-world scenarios for the experimental evaluation of the proposed framework, including citation networks, and we are currently developing DyHANE by instantiating the various strategies specifically for a node classification task on a closed-world setting, i.e., with fixed and known number of node classes, such as authors' research fields.

References

1. S. Khoshraftar, A. An, CoRR **abs/2204.01855** (2022). DOI 10.48550/arXiv.2204.01855. URL https://doi.org/10.48550/arXiv.2204.01855
2. A. Sankar, Y. Wu, L. Gou, W. Zhang, H. Yang, in WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020, ed. by J. Caverlee, X.B. Hu, M. Lalmas, W. Wang (ACM, 2020), pp. 519–527. DOI 10.1145/3336191.3371845. URL https://doi.org/10.1145/3336191.3371845
3. Y. Ma, Z. Guo, Z. Ren, J. Tang, D. Yin, in Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, ed. by J.X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, Y. Liu (ACM, 2020), pp. 719–728. DOI 10.1145/3397271.3401092. URL https://doi.org/10.1145/3397271.3401092
4. L. Yang, Z. Xiao, W. Jiang, Y. Wei, Y. Hu, H. Wang, in Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II, Lecture Notes in Computer Science, vol. 12036, ed. by J.M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M.J. Silva, F. Martins (Springer, 2020), Lecture Notes in Computer Science, vol. 12036, pp. 425–432. DOI 10.1007/978-3-030-45442-5_53. URL https://doi.org/10.1007/978-3-030-45442-5_53
5. H. Xue, L. Yang, W. Jiang, Y. Wei, Y. Hu, Y. Lin, CoRR **abs/2004.01024** (2020). URL https://arxiv.org/abs/2004.01024
6. R. Trivedi, M. Farajtabar, P. Biswal, H. Zha, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019 (OpenReview.net, 2019). URL https://openreview.net/forum?id=HyePrhR5KX
7. Y. Dong, N.V. Chawla, A. Swami, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017 (ACM, 2017), pp. 135–144. DOI 10.1145/3097983.3098036. URL https://doi.org/10.1145/3097983.3098036
8. J. Shang, M. Qu, J. Liu, L.M. Kaplan, J. Han, J. Peng, CoRR **abs/1610.09769** (2016). URL http://arxiv.org/abs/1610.09769
9. L. Zhou, Y. Yang, X. Ren, F. Wu, Y. Zhuang, in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, ed. by S.A. McIlraith, K.Q. Weinberger (AAAI Press, 2018), pp. 571–578. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16572
10. R. Bian, Y.S. Koh, G. Dobbie, A. Divoli, in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, ed. by B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, F. Scholer (ACM, 2019), pp. 861–864. DOI 10.1145/3331184.3331273. URL https://doi.org/10.1145/3331184.3331273
11. F. Zhou, C. Cao, in Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021 (AAAI Press, 2021), pp. 4714–4722. URL https://ojs.aaai.org/index.php/AAAI/article/view/16602
12. J. Wang, G. Song, Y. Wu, L. Wang, CoRR **abs/2009.10951** (2020). URL https://arxiv.org/abs/2009.10951
13. J. Kirkpatrick, R. Pascanu, N.C. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, CoRR **abs/1612.00796** (2016). URL http://arxiv.org/abs/1612.00796