Node Embeddings in Dynamic Graphs

Ferenc Béres[1], Róbert Pálovics[2], Domokos Kelen[1], Dávid Szabó[1], and András Benczúr[1]

[1] Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2] Department of Computer Science, Stanford University

1 Introduction

Based on the success of **neural network embeddings** for natural language processing introduced by the Word2Vec algorithm in [6], several network embedding methods have been proposed recently [8, 10, 5, 9] that are highly successful in multi-label classification and link prediction in a variety of real-world networks from diverse domains.

We consider **edge streams** such as Twitter mentions or retweets where edges arrive continuously over time and have no duration [1]. In edge streams with fast dynamics, it is challenging to maintain an embedding for tracking and measuring node properties and similarities as the edges arrive. Our main result is an **online updateable node embedding** that respects the order of edge creation.

In a supervised experiment, we show that our online updateable embeddings capture node similarities better than static embeddings. We selected a Twitter data set for tennis tournaments where ground truth labels can be defined based on player participation. We are not aware of any other graph data sets with high temporal granularity, **dynamically changing ground truth** labeling available.

Our algorithm performs **online machine learning** [2] by continuously updating a model as we read the edge stream. Its key ingredients are online gradient descent optimization [4], fingerprinting for fast access to neighborhood [3], and temporal walks [1].

2 Online Node2Vec

Graph embeddings [8, 10, 5, 9] build node sequences and consider them as sentences for Word2Vec embedding [6]. The simplest variants generate **random walks** [8]. In [10], node pairs with **similar neighborhoods** are also considered. Node2Vec [5] considers random walks and neighborhood similarity by second order Markov chains that tend to stay closer to the neighborhoods of the last nodes.

We define online updateable embedding methods first by turning the neural network optimization procedure online. The skip-gram with negative-sampling (SGNS) method of [7] uses gradient descent, which we can turn online updateable as in [4]. For generating random walks online, we use the temporal walk method of [1], and for neighborhood similarity, we use fingerprinting [3]. We describe these algorithms next.

In our **Temporal Walk** algorithm, we maintain time respecting walks as new edges uv arrive with timestamp t. First, we update the weight of all walks ending at u by multiplying with the exponential time decay function $\exp(-c(t-t(u)))$ where $t(u)$ is the last visit time of node u. We proceed similarly for v; for computational efficiency,

Algorithm 1 Update procedures for neighborhood similarity

```
procedure UpdateNeighborhood(uv)
    Update the fingerprints of v by the new edge uv
    for all out_neighbors x ≠ v of u do
        Update the fingerprints of x by the new edge uv
        for all i in 1...k with f_i(v) = u do
            if f_i(x) = u then
                call Word2Vec with v and x
```

we can delete walks whose weight becomes very small. For each walk from s to u, we obtain a new walk from s to v by adding the new edge uv. Hence, we increase the weight of walks from s to v by the weight of walks from s to u. After computing all weights that involve uv, we generate s,v pairs for Word2Vec with probability proportional to the weight of walks from s to v.

In our **Temporal Neighborhood** algorithm, we generate node pairs whose neighborhood is similar and give as input to Word2Vec. For each node v, we maintain k MinHash fingerprints [3] that we update dynamically as new edges enter the network. Given a new edge uv, in Algorithm 1 we select a node x as input pair for v to Word2Vec if its temporal MinHash fingerprint is equal to u. We can also generate fingerprints for higher order similarity in the sense of SimRank and its variants [3].

We can take computational advantage of the fingerprints by a heuristic procedure. Note that updating all fingerprints takes the same order of running time as computing the overlap of the neighborhood of u, v and all their neighbors x, which is computationally infeasible. Instead of a costly exact update, we update the MinHash fingerprints of the neighbors x of u as follows. First, we maintain $t(x)$, the last update time of node x, and keep each fingerprint independently with probability $\exp(-c(\text{now}-t(x)))$; otherwise we discard the fingerprint. While heuristic update correctly handles the probability for discarding an old fingerprint, unfortunately the probability of selecting the new vertex as fingerprint is much higher than in the exact naive algorithm. Yet, as we see in our experiments, our heuristic procedure works well to generate input pairs for Word2Vec.

3 Experiments

In our experiments, we use the Roland-Garros 2017 (RG17) and US Open (OU17) Twitter data sets [1]. By representing mentions as directed edges between Twitter accounts, we obtain a dynamic network where the instance of the directed edge uv appears whenever user u mentions another user v.

For evaluation, we define **node labels that dynamically change in time** based on the tennis championship schedule. On a given day, a Twitter account is relevant if it belongs to a tennis player who participated in a game on that day. For each relevant node, we generate a list of similar nodes by computing the dot product of the 128-dimensional embeddings that we evaluate by NDCG. The higher the NDCG, the closer are players who play the same day in the embedding. In particular, players who are active on the same day are more similar to each other than to general tennis related accounts and other players who do not play on the given day.

Table 1. Average NDCG in time over the RG17 and UO17 mention graphs.

	RG17	**UO17**
Indegree	0.229	0.196
Node2Vec	0.191	0.207
Temporal Walk	0.273	0.215
Temporal Neighborhood	**0.278**	**0.277**

Table 2. Similarity list of Rafael Nadal for both methods based on embeddings generated at 12:00 on May 31 (RG17). Relevant players are highlighted in yellow and media accounts in gray.

	Node2Vec	**Temporal Walk**	**Temporal Neighborhood**
1	rolandgarros	rolandgarros	rolandgarros
2	aquiactualidad	tsonga7	Gael_Monfils
3	KikiMladenovic	DjokerNole	DjokerNole
4	TennisChannel	KikiMladenovic	KikiMladenovic
5	emirates	delpotrojuan	stanwawrinka
6	tsonga7	andy_murray	tsonga7
7	Gael_Monfils	Gael_Monfils	FerVerdasco
8	Maly_Tweet	TennisChannel	Simona_Halep
9	N1infoSA	stanwawrinka	andy_murray
10	Simona_Halep	GarbiMuguruza	WTA

In Table 1, we show NDCG for both datasets, averaged for all active players every six hours. As a baseline, we calculated Indegree and Node2Vec [5] on the graph snapshot of the last 12 hours. Both Temporal Walk and Neighborhood outperform the baselines. In Table 2, we show the ten most similar accounts to Rafael Nadal. Our online updateable models perform better at showing daily players rather than popular media accounts.

Our data and codes are available at https://github.com/ferencberes/online-node2vec.

References

1. Béres, F., Pálovics, R., Oláh, A., Benczúr, A.A.: Temporal walk based centrality metric for graph streams. Applied Network Science 3(1), 32 (2018)
2. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive online analysis. Journal of Machine Learning Research **11**(May), 1601–1604 (2010)
3. Fogaras, D., Rácz, B.: Scaling link-based similarity search. WWW. pp. 641–650. (2005)
4. Frigó, E., Pálovics, R., Kelen, D., Benczúr, A.A., Kocsis, L.: Online ranking prediction in non-stationary environments. In: Proc. TRRS Workshop in conjunction with RecSys (2017)
5. Grover, A., Leskovec, J.: Node2Vec: Scalable feature learning for networks. In: Proc. KDD. pp. 855–864. (2016)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
8. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proc. KDD. pp. 701–710. ACM
9. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J.: Network embedding as matrix factorization: Unifying Deepwalk, Line, PTE, and Node2Vec. In: Proc. WSDM. pp. 459–467. (2018)
10. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proc. WWW. pp. 1067–1077. (2015)