Learning Node Embeddings for Influence Set Completion

Conference Paper · September 2018

DOI: 10.1109/ICDMW.2018.00149

3 authors, including:

Nikita Alexeevich Durasov
Moscow Institute of Physics and Technology
4 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Evgeny Burnaev
Skolkovo Institute of Science and Technology
221 PUBLICATIONS **1,532** CITATIONS
SEE PROFILE

Learning Node Embeddings for Influence Set Completion

Sergei Ivanov*†, Nikita Durasov‡, Evgeny Burnaev*
*Skoltech, Russia †Criteo Research ‡Moscow Institute of Physics and Technology, Russia
contact email: sergei.ivanov@skolkovotech.ru

***Abstract*—Influence Maximization problem has found numerous applications in the real world and attracted a lot of research in the recent years. However, all previous attempts to provide a solution were based solely on the graph topology. Instead, we show how to employ recent advancement in representation learning and use node embeddings for finding solution as a supervised task. In our experiments, we show that the ranked list of nodes obtained by classifier yields better influence completion set than other baselines.**

***Index Terms*—influence maximization, node embeddings.**

I. Introduction

Influence Maximization is the problem of finding influential nodes in the network according to the influence propagation model. It has found use cases in numerous domains such as control of contamination in water networks [1], viral marketing in social networks [2], and content recommendation for users [3]. For example, a marketing company that wants to acquire a small initial set of initial adopters to promote a product to their followers is a typical application of influence maximization. From the outset, this problem has gained a lot of attention and many challenges associated with the selection of the initial set, with the design of realistic propagation model, and with the computation of influence function have been addressed. These works have made a major leap towards understanding the influence in the networks from a purely topological perspective of the network and the influence model therein.

Somewhat parallel track of research has been concerned with vector representations of networks, also known as *embeddings*. There has been a substantial effort to design the node, edge, or (sub)graph vector representations as a native data format for classical machine learning algorithms such as SVM and neural networks. Link prediction [4], network visualization [5], taxonomy recovery [6], and protein classification [7] problems are some examples of applications, where graph embeddings have been used successfully. In these problems, the nodes, or other graph substructures, are embedded in a latent vector space so that machine learning algorithms could be applied to the vectors directly. The net effect of this approach is that after the representation embeddings have been obtained one can focus on the appropriate selection and design of machine learning algorithms that have been studied over the last decades.

In this work we make an attempt to apply representation learning algorithms to facilitate the seed set completion of influence maximization. In particular, we assume that we identified a small set of nodes that we can consider as influential. We then seek to extend this seed set by using pairs (embedding, label) from the seed set to train a binary classification algorithm. This problem, which we frame as influence completion, is motivated by the high cost of finding a large set of influential users due to substantial running time or significant use of memory resources of the traditional algorithms [8]. Instead we use node embeddings to find the extension of the seed set by using only a fraction of all the nodes in the graph.

II. Related work

As an optimization problem Influence Maximization was first formulated by Kempe et al. [2], where the greedy algorithm with approximation guarantees and influence models were proposed. That was followed by a large study of the algorithms [1], [9]–[11], which aim to decrease the running time of the time-consuming greedy algorithm while preserving the quality of the influence set. While the significant progress has been made towards the goal of fast and efficient algorithm, the question of selecting a large set of influencers in a large graph has not been precisely addressed and the algorithms remain to be resource demanding [12].

Representation learning on graphs has seen an increase of research advancements in recent years, where the embeddings are searched for the node, edge, and/or subgraph structures of the graphs for different applications. As such, node classification algorithms use node embeddings [4], [13], [14] based on walks and graph patterns, taxonomy relation algorithms exploit the hyperbolic space of node embeddings [6], [15], and visualization techniques aim to group similar nodes embeddings close to each other in the latent space [16]. Also representation learning algorithms have been applied to estimate the size of the propagation cascade [17], [18]; yet, to the best of our knowledge, we are the first to exploit node embeddings for searching influential users in a network.

III. Influence Completion Problem

In Influence Maximization problem one seeks a small set of nodes that would maximize the influence function $\sigma_\mu(S)$ for a given probabilistic graph $G = (V, E, P)$, where P defines the probabilities for every edge, given a set of nodes $S \subset V$ and a diffusion model μ. Diffusion models define the way the information propagates from the initial set S to other nodes in the graphs. Example of a diffusion model is Independent

Cascade model, where each node in S has a single and independent attempt to append its neighbor to a set of activated nodes. Due to the space limit, we refer an interested reader to exact definition of Independent Cascade model and influence function in [2]. We are interested however in extending a small seed set of influential nodes by an additional set of marginally influential nodes. We define the problem as follows:

Problem 1 (Influence Completion problem): Let G be a probabilistic graph and S is the set of the most influential nodes of size k. We are asked to find a set T of size $k+l$ and $S \subset T$ so that the influence function $\sigma_\mu(T)$ is maximized.

The problem is motivated when we are already given a seed set of initial adopters and we seek to find additional extension of this set to maximize the value. At the same time, graph embeddings are useful data representation, which can be used for multiple applications such as graph classification [7] and clustering [19]. Therefore we aim to use node embeddings for learning influence of each node by observing relationship between the node representations and the set of first influential users. The intuition behind it is that if node embeddings define local and global properties of a graph then given a small set of influential nodes we are able to capture "influential" topology of the network by learning a model on influential node embeddings.

Influence Completion problem is NP-hard problem as we can remove the set S and the corresponding edges from G, in which case the problem is reduced to Influence Maximization, which is known to be NP-hard. However, one can first obtain some ground-truth seed set, either as an output of Influence Maximization algorithm or by domain specific knowledge, and then use the embeddings of the nodes to acquire new "similarly influential" nodes, which we show next.

IV. Algorithms

Our approach relies on already identified initial set of nodes, which we use to build the dataset for our supervised classification model. The algorithm **InfEmb** takes as input a graph G, a seed set S, and an integer l and outputs an extended set $T \supset S$ that attempts to maximize the influence function $\sigma(T)$.

Algorithm 1: **InfEmb** algorithm

Input: graph G, seed set S, and integer l.
Output: set T, s.t. $|T| = |S| + l$

1: Compute a feature map $\phi : v \mapsto R^d$ for all nodes in G.
2: Train a classification model $f : \mathbb{R}^d \mapsto [0, 1]$ of influence score for each node.
3: $T = S$
4: Append top-l nodes to T based on influence score that are not yet in S.
5: **return** T

The algorithm **InfEmb** has essentially three steps. In the first step (Line 1), the algorithm computes embeddings for each node in the graph, including those in S. In the experimental section we deal with several node embeddings algorithm that encode graph structure, as well as, direct computation of the statistics related to the node neighborhoods and seed set. In the second step (Line 2), the algorithm trains a classification algorithm f that outputs influence score for each node in the graph. For this, we use the nodes in S as the positive labels. We sample the negative labels from the remaining nodes reversely proportional to their degrees. That is the nodes that have small degree will most likely appear as negatives, which is motivated by the fact that degree of a node serves as a good proxy for the influence score [2]. The influence score is then the likelihood the algorithm f assigns to each node to be a positive label. In experiments we compare different options for a feature map ϕ and a classification model f. In the third step (Line 4), the algorithm includes top-l nodes that are not yet in the given set S. We evaluate the quality of the node embeddings in the following section.

TABLE I: Networks used in experiments. The columns are: name of the network, number of nodes and edges, average clustering coefficient, diameter.

Dataset	Directed	Nodes	Edges	Clustering Coefficient	Diameter
GRQC	No	5242	14496	0.52	17
Wiki	Yes	7115	103689	0.14	7
FB	No	4039	88234	0.60	8

V. Experiments

For evaluation of our approach we obtain solutions to Influence Completion problem on several datasets and compare it against several baselines using several evaluation metrics.

Datasets. We use three publicly available datasets, GRQC[1], Wiki[2], and Facebook[3]. GRQC dataset is a collaboration network in General Relativity and Quantum Cosmology area authors are nodes and edges correspond to their co-authorship of the paper. Wiki network is a graph where nodes are the users in Wikipedia and edges are among the users who give the votes to become administrators. Facebook is a social network with users as nodes and edges denote their friendships in the network. Each edge in undirected network is replaced by two directed edges. The datasets vary in their structural properties as shown in the Table I.

Propagation models. Similar to previous research we use Independent Cascade model to determine influence propagation in the network. In this model, the influence propagates from one node to another node with a probability of the edge. We assigned probabilities according to a weighted cascade model [2], i.e. an edge (u, v) has a probability $1/d_v$, where d_v is degree of a node v.

Algorithms. We test our algorithm **InfEmb** with different choices of functions ϕ and f. We select the feature map ϕ from the embedding methods node2vec [4], DeepWalk [13], and AWE [7]. We denote the embeddings as N2V, DW, and

[1]http://snap.stanford.edu/data/ca-GrQc.html
[2]http://snap.stanford.edu/data/wiki-Vote.html
[3]https://snap.stanford.edu/data/ego-Facebook.html

Fig. 1: Accuracy and Relative Change in influence spread for **GRQC** dataset.

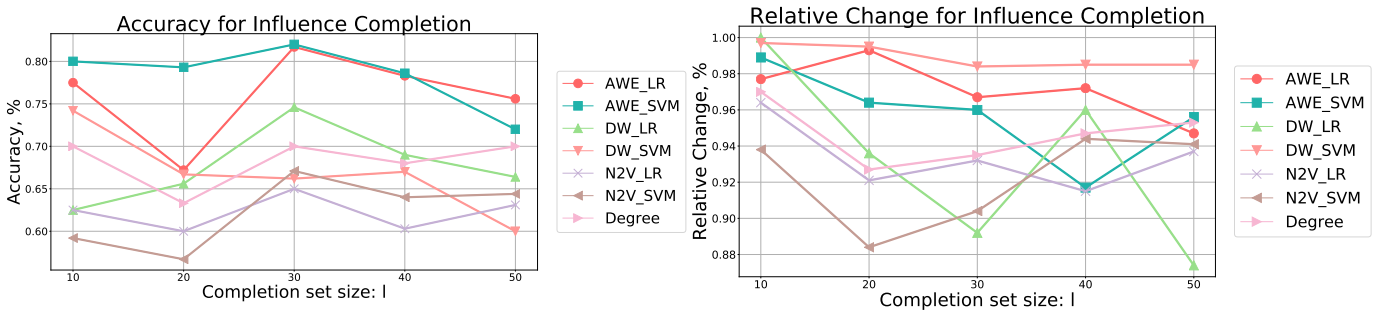


Fig. 2: Accuracy and Relative Change in influence spread for **Wiki** dataset.

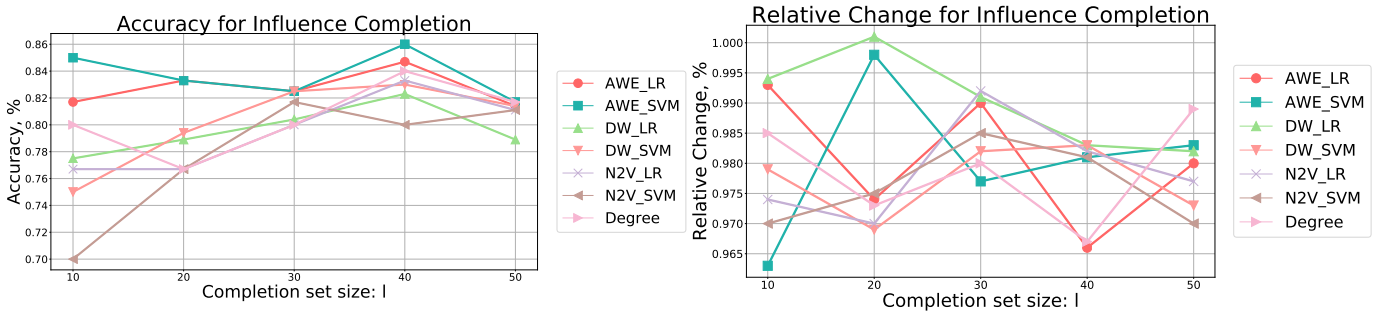
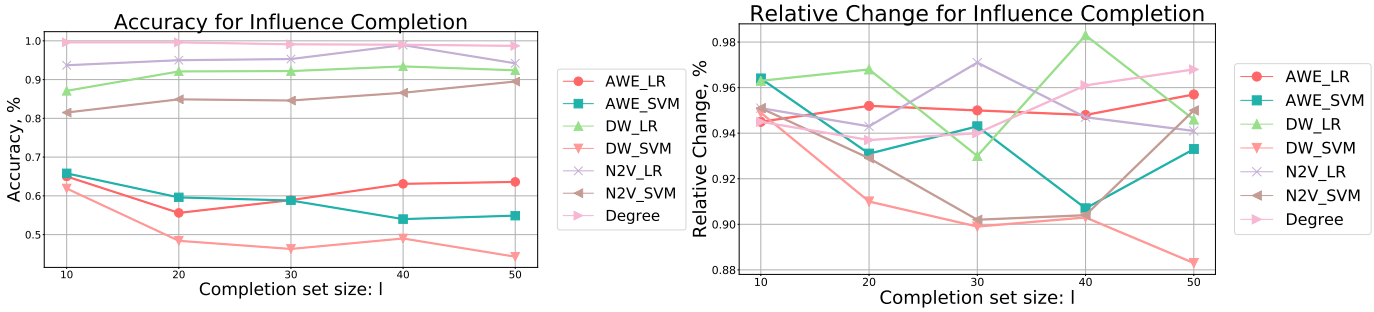


Fig. 3: Accuracy and Relative Change in influence spread for **FB** dataset.



AWE respectively. The first two are popular algorithms for node embeddings that create vector representations from the random walks in the graph. AWE is a version of anonymous walk embeddings [7] adapted for node embeddings. It has been shown [20] that the distribution of anonymous walks for a single node is sufficient to reconstruct a neighborhood around this node, i.e. AWE node embeddings encode the local graph structure around the node. Hence, if the propagation of influence happens within a neighborhood of influential nodes, this method could distinguish the influence of nodes by looking at the vector representations of the neighborhoods. For the AWE we compute a distribution of anonymous walks for each node starting at that node with a fixed length $L = 4$. In addition to the vectors obtained by the feature maps ϕ, we append information about the number of influential nodes in the neighborhood and its degree to the node embeddings, which should eliminate the addition of two influential nodes to the seed set that are located in one-hop distance.

We also experiment with the classification function f and select it as SVM or Logistic Regression (LR). We use the prediction values from classification model f to rank all the nodes in the graph and we select top-l nodes from this list that are not yet in the set S. We compare our model with a baseline algorithm, Degree, where the top-nodes are selected from the ranked list according to their degrees.

Evaluation metrics. We measure performance of the proposed algorithms by two metrics related to influence maximization. To measure *accuracy* of the approach, we train the model on k first positive and k negative nodes, and compute the overlap between the top $[k+1,\ k+2,\ \ldots,\ k+l]$ nodes

returned by the model and the one that we consider as ground-truth. We set $k = 10$ and vary the completion set size l in the range [10, ..., 50]. To get ground-truth nodes one can either use a standard traditional influence maximization algorithm or rely upon a domain knowledge. We use D-SSA algorithm [11] that proved to be among state-of-the-art approaches for influence maximization. Intuitively, accuracy of the model show the discrepancy between the predictions of the classifier and the ground-truth values that are hard to obtain. Additionally, in some scenarios we want to measure the *relative change* (RC) of influence spread by the model's seed set compared to the given oracle seed set. While the model may miss some of the top-influential nodes, the returned by model nodes can still be highly influential and therefore RC measures the discrepancy of influence spread. More formally, we use first k ground-truth nodes and top-l nodes returned by the model to compute influence spread s_{cl}. We then compute influence spread s_{gr} of ground-truth $k + l$ nodes and define relative change of the model as $\frac{s_{gr}}{s_{cl}}$.

Results. Results for GRQC dataset are presented in Figure 1. On the left, the accuracy, i.e. the overlap between the ground-truth seed set and the set returned by the classifier, is presented. AWE is consistently on the top with 10% and 2% uplift in accuracy compared to degree approach for l=10 and l=50 respectively. The value of accuracy decreases from 80% to 0.68% for AWE-SVM model. DW and N2V embeddings, in general perform comparably with the degree approach. We also see that the classification models perform similarly for all algorithms in terms of accuracy, while SVM is generally more robust across the whole range of values l.

The Relative Change (RC), i.e. the influence spread of the ground-truth set divided by the influence spread of the influence set of the algorithm, is presented on the right. RC for AWE and DW is at the top, with 0.97 and 0.99 change in influence spread respectively for l=10. RC for DW and N2V models is above 80% and in general comparable with the baseline degree algorithm.

Results for Wiki and FB datasets are presented in Figures 2 and 3 and share the main insights as for the GRQC dataset. In particular, for Wiki dataset the accuracy and relative decrease for AWE is consistently at the top, being always higher 80% and 0.9 for two metrics respectively. N2V and D2W performs similarly to the degree baseline with 78%-80% accuracy and 0.95 ± 0.16 relative change. For FB dataset, the accuracy for degree baseline is quite high, which implies there is a strong correlation between the degree and the influence of the node. Yet, in terms of relative change the list of nodes of InfEmb algorithm is still highly influential, especially when it comes to predict the first nodes with $l = 10$. We also obtained the results for the random baseline, when the nodes for completion set are taken uniformly at random across all nodes in the graph. We didn't include it in the figures for visibility reasons, however, the accuracy and relative decrease are significantly smaller compared to other algorithms, being lower than 1% and 50% for two metrics respectively.

VI. Conclusion

In this work, we proposed an approach for exploiting embedding representation of the graph structures to facilitate finding influential nodes given a small ground-truth seed set. In the experiments we found that this approach can yield better completion set than other baselines in terms of accuracy and relative change. Moreover, one could use the top ranked nodes as the candidate set for the greedy algorithm for speed up of the algorithm, while not sacrificing much on the influence spread, if the number of candidates is set properly. One extension of Influence Completion problem that is worth studying in the future work is the prediction of the influence spread by the embedding of a node or a small set of nodes.

Acknowledgment

The research was supported by the Russian Foundation for Basic Research grant 16-29-09649 ofi m.

References

[1] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks."
[2] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," ser. SIGKDD 2003.
[3] S. Ivanov, K. Theocharidis, M. Terrovitis, and P. Karras, "Content recommendation for viral social influence," ser. SIGIR '17.
[4] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, 2016, pp. 855–864.
[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," ser. ICLR '17.
[6] M. Nickel and D. Kiela, "Learning continuous hierarchies in the lorentz model of hyperbolic geometry," in *ICML '18*.
[7] S. Ivanov and E. Burnaev, "Anonymous walk embeddings," ser. ICML '18.
[8] A. Arora, S. Galhotra, and S. Ranu, "Debunking the myths of influence maximization: An in-depth benchmarking study," ser. SIGMOD '17.
[9] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," ser. WWW '11.
[10] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," ser. SIGMOD '15.
[11] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," ser. SIGMOD '16.
[12] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan, "Revisiting the stop-and-stare algorithms for influence maximization," *Proc. VLDB Endow., 2017*.
[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," ser. KDD '14.
[14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW '15*, 2015.
[15] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," ser. NIPS, 2017.
[16] M. D. Ben Chamberlain and J. Clough, "Neural embeddings of graphs in hyperbolic space," ser. Workshop on Mining and Learning with Graphs (MLG), 2017.
[17] C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: An end-to-end predictor of information cascades," ser. WWW '17.
[18] S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari, "Learning social network embeddings for predicting information diffusion," ser. WSDM '14.
[19] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, and Y. Liu, "graph2vec: Learning distributed representations of graphs."
[20] S. Micali and Z. Allen Zhu, "Reconstructing markov processes from independent and anonymous experiments," *Discrete Applied Mathematics*, 2016.