Expert Systems With Applications 140 (2020) 112905

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Influence maximization across heterogeneous interconnected networks based on deep learning

Mohammad Mehdi Keikha[a,b,*], Maseud Rahgozar[a,*], Masoud Asadpour[a], Mohammad Faghih Abdollahi[c]

[a] *School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[b] *University of Sistan and Baluchestan, Zahedan, Iran*
[c] *Department of Computer engineering, Khatam University, Tehran, Iran*

ARTICLE INFO

Article history:
Received 17 September 2018
Revised 19 July 2019
Accepted 29 August 2019
Available online 31 August 2019

Keywords:
Influence maximization
Interconnected networks
Network embedding
Deep learning
Relevant users

ABSTRACT

With the fast development of online social networks, a large number of their members are involved in more than one social network. Finding most influential users is one of the interesting social network analysis tasks. The influence maximization (IM) problem aims to select a minimum set of users who maximize the influence spread on the underlying network. Most of the previous researches only focus on a single social networks, whereas in real world, users join to multiple social networks. Thus, influence can spread through common users on multiple networks. Besides, the existing works including simulation based, proxy based and sketch based approaches suffer from different issues including scalability, efficiency and feasibility due to the nature of these approaches for exploring networks and computation of their influence diffusion. Moreover, in the previous algorithms, several heuristics are employed to capture network topology for IM. But, these methods have information loss during network exploration because of their pruning strategies.

In this paper, a new research direction is presented for studying IM problem on interconnected networks. The proposed approach employs deep learning techniques to learn the feature vectors of network nodes while preserving both local and global structural information. To the best of our knowledge, network embedding has not yet been used to solve IM problem. Indeed, our algorithm leverages deep learning techniques for feature engineering to extract all the appropriate information related to IM problem for single and interconnected networks. Moreover, we prove that the proposed algorithm is monotone and submodular, thus, an optimal solution is guaranteed by the proposed approach. The experimental results on two interconnected networks including DBLP and Twitter-Foursquare illustrate the efficiency of the proposed algorithm in comparison to state of the art IM algorithms. We also conduct some experiments on NetHept dataset to evaluate the performance of the proposed approach on single networks.

1. Introduction

People usually prefer to gather information from their acquaintances in the form of “word of mouth” than the other medias such as TV (Bakshy, Rosenn, Marlow & Adamic, 2012). With the growth of popularity of Online Social Networks (OSNs), a piece of information could quickly spread among people. Thus, OSNs can be considered as a platform for viral marketing. Companies target a small number of users, aka seed set to recommend and advertise their new products to their friends in such a way, maximal number of people adopt the products. The idea of information spreading through “word of mouth” on OSNs was firstly proposed as influence maximization (IM) by Domingos and Richardson (2001). There are many applications for IM such as viral marketing (Domingos & Richardson, 2001), rumor control (Budak, Agrawal & El Abbadi, 2011) and recommendation (Ye, Liu & Lee, 2012).

In traditional IM problem, one network is given to be explored to find optimal seed set. While in real world, users usually join to multiple social network simultaneously and information can spread across multiple OSNs via the bridge users who are member of different networks, simultaneously (Gaeta, 2018; Nguyen, Zhang, Das, Thai & Dinh, 2013; Zhan, Zhang, Wang, Yu & Xie, 2015; Zhang, Nguyen, Zhang & Thai, 2016). Hence, users can be influenced by

[*] Corresponding authors.
E-mail addresses: mehdi.keikha@ut.ac.ir (M.M. Keikha), rahgozar@ut.ac.ir (M. Rahgozar), asadpour@ut.ac.ir (M. Asadpour), m.faghih@khatam.ac.ir (M.F. Abdollahi).

https://doi.org/10.1016/j.eswa.2019.112905
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

the users of the other OSNs (Zhan, Zhang, Yu, Emery & Xie, 2016). Thus, the effects of external influence from users of other networks is an important factor which should be considered in IM problem. Unfortunately, most of the previous researches on IM, propose a method on a single network and the impact of external influence and the bridge users across multiple OSNs are ignored.

Kempe, Kleinberg and Tardos (2003) was firstly formulated IM as an optimization problem and proved NP-hardness of IM under two diffusion models including independent cascade (IC) and linear threshold (LT) model. Then, they proposed a greedy algorithm with approximation factor of (1−1/e) to find optimal seed set. Even though the accuracy of greedy algorithm is better than classical degree based approaches, however it suffers from scalability issue for large scale networks. Many researches have been done to overcome the scalability issue of the greedy algorithm. Leskovec et al. (2007) extended the sub modularity property of the influence function to speed up IM. LDAG is proposed to solve IM under LT model (Chen, Wang & Wang, 2010). While it could compute influence spread on DAGs in polynomial time, but the process of generating DAGs is NP-hard. In the previous researches, different heuristics has been made to reduce network size; then a solution is proposed based on the heuristics. Though, these algorithms have been made significant improvements in comparison to the greedy algorithm under IC and LT models, the cost of influence spreading over large networks is still inefficient.

Recently, researchers have investigated interconnected networks via bridge nodes (Liu et al., 2012; Shen, Dinh, Zhang & Thai, 2012; Yagan, Qian, Zhang & Cochran, 2012). Nguyen et al. (2013) illustrated influence can propagate on inside and across social networks. They integrate all the networks into one scheme which preserves the features of the users on the source networks. Then, traditional IM solution are employed to assess the influence spread on multiple OSNs. Shen et al. (2012) also combine all the networks to measure the influence spread of network users. Zhan et al. (2016) first extract different information channels in the network and construct a multi relational network by using these channels. Next, the seed sets are chosen through some measures that computes the influence of each node on the multi relational network.

In recent years, deep learning techniques have been made great impacts on different applications such as speech recognition (Mohamed, Yu & Deng, 2010), image processing (Krizhevsky, Sutskever & Hinton, 2012), information retrieval (Hinton & Salakhutdinov, 2011) and social network analysis (Perozzi, Al-Rfou & Skiena, 2014). Network embedding by using deep learning as a representation learning method, encodes the local and global features of the network into feature vectors (Grover & Leskovec, 2016; Keikha, Rahgozar & Asadpour, 2018; Perozzi et al., 2014). Indeed, network embedding is a dimension reduction technique which can preserves all the structural features of the given network. The learned feature vectors can be applied on different applications such as clustering (Huang, Huang, Wang & Wang, 2014; Xie, Girshick & Farhadi, 2016) and link prediction (Grover & Leskovec, 2016).

In this paper, we propose a deep learning based algorithm named “DeepIM” for IM problem on interconnected networks by applying network embedding. Influence maximization across interconnected networks is highly challenging due to heterogeneous structural features, cross links and bridge nodes of the given networks. Furthermore, the complexity of IM problem on interconnected networks is more than the traditional IM because of increasing in problem size due to growth of network nodes. To the best of our knowledge, the proposed method is the first algorithm which has applied network embedding to solve IM problem.

We utilize CARE algorithm to extract global and local structural features of nodes on both networks (Keikha et al., 2018). We first generate a number of predefine customized paths for each node on both networks. These paths include both node’s neighbors as the local and community information of the node as the global structure. Then, the customized paths are used to learn the best structural feature vector of the network nodes by using Word2vec framework (Mikolov, Chen, Corrado & Dean, 2013a, 2013b). When the feature vectors are learned for each network node, we apply them to measure the extent of relevancy among users of interconnected networks. Next, the most influential nodes are chosen from the node who are related to more extent of the users inside and outside of the networks. Thus, we are able to find seed set by their feature vectors which are learned by network embedding. In contrast to the previous researches, in DeepIM algorithm, all the structural features of nodes are considered for influence spreading.

Extensive assessments of the proposed algorithm are performed on three datasets including DBLP networks (Tang et al., 2008), Twitter-Foursquare (Zhang, Kong & Yu, 2013) and NetHept (Kempe et al., 2003). Experimental evaluations indicate that bridge nodes of the input networks have a great impact on maximizing the influence inside and between the networks. The empirical analysis verifies the significant improvements of the proposed method in comparison to the previous researches on IM.

To summarize, we make the following contributions:

- We present a novel algorithm for IM across interconnected networks that learns the best feature vector of nodes on different types of networks such as weighted, directed and complex networks.
- To the best of our knowledge, the proposed method is the first algorithm that utilizes deep learning techniques to extract best structural features of the network nodes for IM.
- We show the impact of bridge node to diffuse the influence across OSNs.
- DeepIM finds most influential nodes inside and between networks for each node based on their local and global structural properties.
- Network changes can be considered simply by the proposed method. So, the influence of the new nodes is computed without repeating the process of influence spreading for all the nodes.
- We empirically evaluate the proposed method on two interconnected networks datasets and a single network. The experimental results indicate the scalability and efficiency DeepIM in contrast to the other IM approaches.

The rest of paper is organized as follows: Section 2 presents a formal definition of IM on interconnected networks. In Section 3, we summarize related works to IM methods and network embedding techniques. We explain the details of DeepIM algorithm in Section 4. Section 5 outlines the experimental results of the proposed method on different datasets. Finally, Section 6 presents conclusion and future works.

2. Influence maximization on interconnected networks

The goal of IM problem across interconnected networks is to select best seed set from both networks in which the maximum number of users have been influenced by the seed set in both networks. Suppose, two network graphs $\mathrm{G}_1 = (\mathrm{V}_1, \mathrm{E}_1)$ and $\mathrm{G}_2 = (\mathrm{V}_2, \mathrm{E}_2)$ are given, in which each edge $\mathrm{e}_i = (u_i, v_i, w_i) \in E_i$ represents a collaboration between u_i and v_i with weight w_i in network i. With a budget K, we are going to find a seed set S of size K from users of both networks based on information diffusion model M in such a way that the maximum number of users on the networks are influenced by S. The influence spread of S is shown by $\sigma(S)$. It is worth noting that the IM problem across interconnected network is NP-hard which have proved in Zhan et al. (2015).

In the IM problem, diffusion model M is a stochastic process which denotes how the users can be active or inactive during information diffusion. Independent cascade model (Goldenberg, Libai & Muller, 2001) and linear threshold model (Granovetter, 1978) are two popular diffusion models for IM on social networks. In the IC model, influence is propagated by active nodes based on edge activation probabilities. While in the LT model, each node has a threshold to be active by its neighbors. When the weighted sum of active neighbors of an inactive node exceeds its threshold, then the node become active. In both models, diffusion process continues until no new node can become activate.

Though the IM problem on interconnected networks is a NP-hard under different diffusion models (Zhan et al., 2015), but DeepIM influence function $\sigma(S)$ is monotone and submodular. Thus, our algorithm guarantees an approximation ratio of $(1 - {}^{1}/_{e})$. The proof of monotonicity and submodularity properties of the proposed influence function is explained in Section 4.3.

3. Related works

Kempe et al. (2003) proved IM is a NP-hard optimization problem and solve the problem in a greedy framework but their proposed method is inefficient over large networks. In the recent years, several IM approaches have been proposed on a single network by using different heuristics to improve the efficiency of the greedy framework. Li, Fan, Wang and Tan (2018) classified existing IM approaches based on their heuristics into three categories including simulation based, proxy based and sketch based methods.

Simulation based approaches utilize Monte-Carlo simulation to obtain $\sigma(S)$ for any candidate set. Though, these approaches are applied on different diffusion models such as IC and LT models, but their computational complexity is expensive due to NP-hardness of calculating influence spread of $\sigma(S)$. Leskovec et al. (2007) improved the greedy algorithm by exploiting sub-modularity of $\sigma(S)$ to reduce the running time of influence spread computation for each node by reducing the number of Monte Carlo simulations. Their evaluation showed that CELF performs 700 times faster than the greedy algorithm. CELF++ is an extension of CELF which reduce the number of influence computation in the first step of CELF by avoiding unnecessary MC simulations (Goyal, Lu & Lakshmanan, 2011a). Wang, Cong, Song and Xie (2010) reduce the running time of influence spread for each node by partitioning the given network and computation of influence on different partitions. Overall, the simulation based algorithms have information loss during optimal seed set selection because they leverage different pruning strategies to reduce the complexity overhead of influence spreading.

Proxy based models substitute heavy Monte Carlo simulations with some heuristics such as degree, page rank or other centrality measures. These approaches reduce the computational overhead of influence spread by considering the properties of the nodes in the graph and diffusion models. In DEGDIS algorithm (Chen, Wang & Yang, 2009), the nodes are selected based on their degrees but, the degree of selected node are discounted. Liu et al. (2014a) propose GPR for a set of nodes by considering the page rank of the nodes involved in the candidate seed set as a measure of influence spread. In their method, the mutual influence of selected nodes is ignored. Chen et al. calculates the maximum influence of each node in IC model by using maximum influence arborescence (MIP) data structure. Then, most influential nodes are chosen based on their MIPs in a greedy manner (Chen et al., 2010). Kimura and Saito (2006) consider shortest path of two nodes to measure the extent of influence for each user. In LDAG algorithm (Chen, Yuan & Zhang, 2010), a number of local directed acyclic graphs are generated for each node and the influence spread of each node is computed in LT model, subsequently. Goyal, Lu and Lakshmanan (2011b) propose to enumerate all simple paths of the nodes to their neighborhoods. While the proxy based methods achieve a good efficiency than simulation based methods but their accuracy may decrease because of using heuristics for influence spread over the network.

The third group of IM approaches are Sketch based algorithms which are theoretically efficient in comparison to the other two classes of IM solutions. In Sketch based algorithms, the properties of diffusion models are considered during computation of influence spread on different subgraphs (sketches) of the given network. So, they are not general than simulation based algorithms. Cohen, Delling, Pajor and Werneck (2014) proposed SKIM which uses reverse BFS on each sketch and boosts the influence spread on the seed set through bottom-K minHash. While SKIM shows significant performance than simulation based algorithms but its time complexity is an issue because of generating different sketches. Borgs, Brautbar, Chayes and Lucier (2014) measure the influence spread of seed set by selecting a number of random nodes and the nodes which are reachable by these nodes. Indeed, Random reachable set of each node is its sketch which is generated in lower running time which brings about low time complexity.

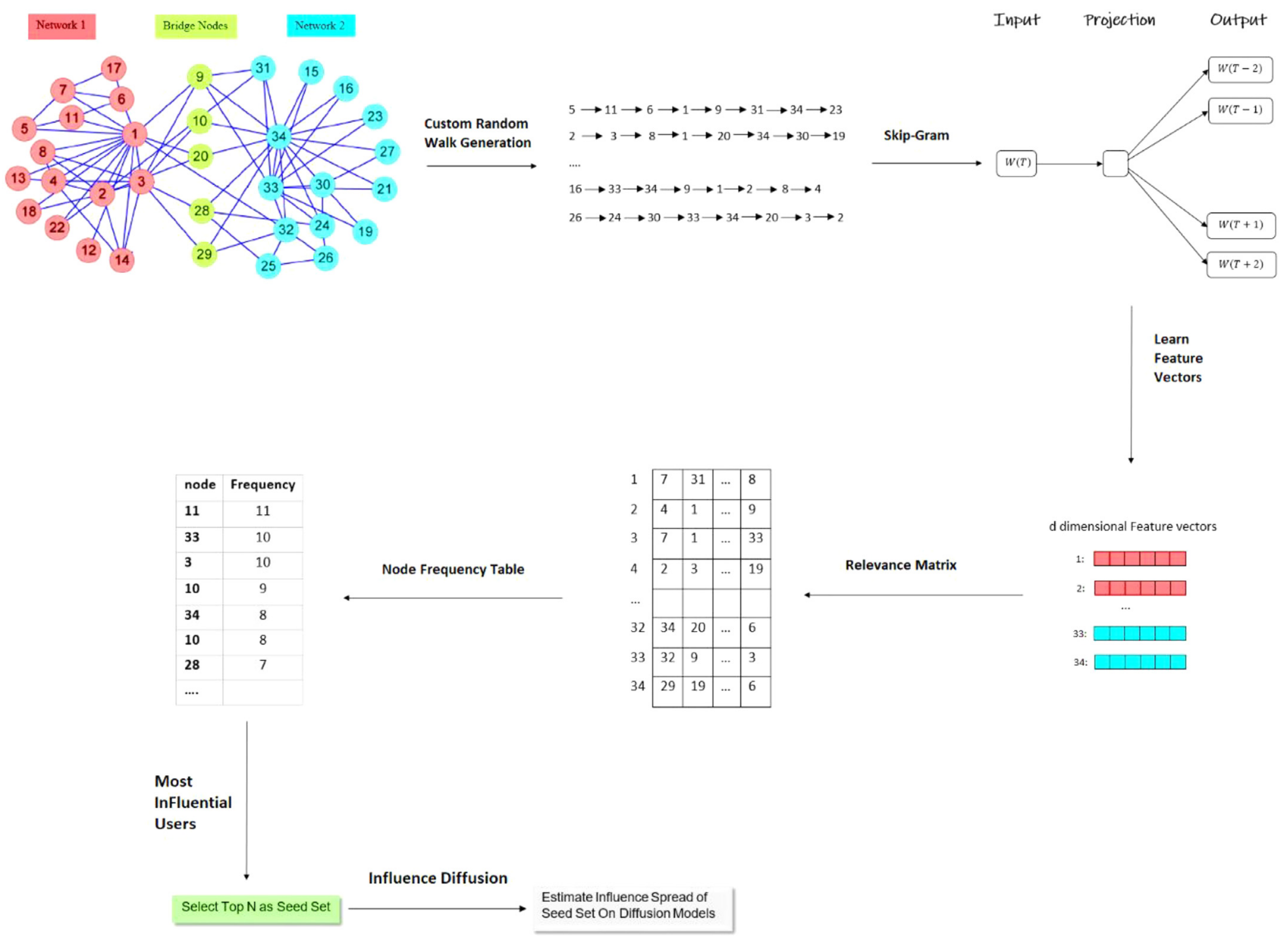
As can be seen, all the previous researches employ different network features in IM. But a large number of them are unscalable. In addition to, the employed features are not general enough to present best performance on different networks with different topology. In the recent years, network embedding is employed on different applications such as social network analysis (Keikha et al., 2018; Perozzi et al., 2014), clustering (Xie et al., 2016). Network embedding methods preserve the local and global features of the network nodes. Thus, it can be considered as a feature engineering approach in different applications. The learned feature vectors are used to measure how much the nodes are related to the other nodes based on their structural properties in the given network. Intuitively, the concept of relevancy of nodes can be considered as a proxy based approach in IM categories.

Deep learning techniques was first used by Perozzi et al. for network embedding in DeepWalk (Perozzi et al., 2014). They used DFS like search strategy to generate random walks. However, the global network structure is not preserved because the community information of nodes is not regarded during the path generation. Tang and Qu (2015) use the first and second order proximities to learn nodes' feature vector but they also preserved local information of the networks(Tang & Qu, 2015). In DeepWalk and LINE, the global structure of networks such as social theories properties are ignored and only local information of nodes are employed to learn best feature vector. In Node2Vec algorithm, random walks are generated by DFS and BFS like strategies (Grover & Leskovec, 2016). They also consider the first and second order proximities during search strategies. One of the great advantages of network embedding is that dynamic changes of the network can be considered without learning the feature vector of the existing nodes.

Overall, many researches have been conducted on IM, However, most of them are suffered from scalability and efficiency issues. In addition, OSNs evolve during the time while the dynamic property of networks is ignored in most of the existing IM approaches. Furthermore, most of the previous researches investigate the IM on a single social network while the external influence of the other networks is considerable in real world. In this paper, an influence maximization algorithm on multiple OSNs is presented to capture both internal and external influence during influence diffusion. We present the details of the proposed algorithm in Section 4.

4. Influence maximization on interconnected networks based on deep learning

In this section, we will describe the proposed algorithm for IM over interconnected networks. To find the best seed set on the net-



1	7	31	...	8
2	4	1	...	9
3	7	1	...	33
4	2	3	...	19
...				
32	34	20	...	6
33	32	9	...	3
34	29	19	...	6

node	Frequency
11	11
33	10
3	10
10	9
34	8
10	8
28	7
....	

Fig. 1. The overall framework of the DeepIM algorithm.

works, in the first step, a structural feature vector for each node is obtained. When the network model is learned by using deep learning techniques, we measure the extent of relatedness of any two users on the given networks. Afterwards, the most relevant users for each node are extracted from both networks and a vector of relevant users for each node is formed. Finally, we select the nodes that are present in the relevant vector of more nodes on both networks. Fig. 1 shows different steps of our algorithm.

As can be seen in Fig. 1, two interconnected networks with bridge nodes are given. To build the structural feature vector of nodes on the networks, CARE algorithm (Keikha et al., 2018) is employed. Keikha et al. extract global and local structural features of nodes by a predefined number of customized paths. The paths contain node's neighbors alongside the nodes which are the same community with the current node of the path. Next, Word2vec as a deep learning model is employed to extract the best structural feature vectors of users based on the generated paths (Mikolov et al., 2013a, 2013b). The network models are employed to identify the most relevant users for each network node based on their structural properties. As a result, we obtain a relevant vector for each user which contains the most relevant nodes of the network nodes based on their structural feature vectors. Ultimately, the most influential users across interconnected networks is chosen based on the number of their presence as relevant node in relevant vectors of all nodes. In Section 4.1, we explain details of CARE as a state of the art network embedding algorithm.

4.1. Network embedding as feature learning in IM

Different structural features including local and global information are widely used in IM algorithms yet. Most of the algorithms examine that how two users can connect to each other based on their structural information and network topology. In this paper, we use a network embedding method named "CARE" to preserve all the structural information of nodes (Keikha et al., 2018). The feature vectors which are learned by CARE contain local neighborhood and community information. To the best of our knowledge, our proposed algorithm is the first method that employs network embedding methods to extract local and global structural information of nodes in IM. In CARE, Word2vec model as a deep learning technique is used to learn the user's feature vectors.

In CARE, Different communities of nodes in each network are first extracted by using Louvain method (Blondel, Guillaume, Lambiotte & Lefebvre, 2008). The main assumption of Louvain method is that the nodes of the same community have more links with each other in comparison to the nodes of other communities. Next, the nodes' communities are employed during customized path generation. A custom path contains local neighborhood of nodes along with the nodes that are in the same community with the last node of the path.

After the generation of a specified number of directed paths for each node, we use Word2vec model to learn the structural feature vectors (Mikolov et al., 2013a, 2013b). In Word2vec algorithm, two nodes are similar when they are visited in many paths with each other. Apparently, two users who are observed in many custom paths, are related to each other. Thus, they can influence each other during information diffusion. The probability of the current nodes' neighbors in the generated path is maximized using Eq. (1):

$$\mathrm{P}(N(u)|f(u)) = \max_{f} \prod\nolimits_{\substack{j = i - w \\ j \neq i}}^{i+w} \mathrm{P}(v_j|f(u))$$

$$N(u) = \{v_{i-w}, \ldots, v_{i+w}\} \backslash u \tag{1}$$

where w is the length of a context window that consists of local neighborhood as well as communities of the nodes in a custom path. The best features of node u are kept in $f(u)$. The conditional probability of visiting node v_j given $f(u)$ is calculated using Eq. (2):

$$\mathrm{P}\left(v_j|f(u)\right) = 1 \Big/ \left(1 + e^{-f(u).f(v_j)}\right) \tag{2}$$

In Eq. (2), $f(u)$, $f(v_j)$ are the feature vectors of $u,\ v_j$, respectively.

Utilizing network embedding as an interesting properties of DeepIM leads to compute the influence spread of each node without dependency to number of edges of the input networks. In other words, against to the previous approaches which explore the networks based on their number of edges, we embed structural properties into custom paths. Thus, the nodes' degree is ignored for network exploration and a fixed number of paths are generated for each node. So, DeepIM spread the influence on networks faster than the previous algorithms and it can be used for IM on real large scale networks.

4.2. Selecting most influential users on interconnected networks

In the previous section, we explain how we model interconnected networks. Afterward the structural feature vectors are learned with deep learning, we form a relevant vector for each node of both networks. Obviously, the relevant vector contains the nodes which are presented either in the local neighborhood or in the same community with the source node. Because, during network embedding, we generate a number of custom paths for each node based on their local neighborhood and the communities that a node belongs to them. To identify the most relevant nodes of each node, the cosine similarity between the feature vectors of two users $u,\ v$ is calculated by the following formula:

$$cosine(f(u), f(v)) = \frac{f(u) \cdot f(v)}{|f(u)||f(v)|} = \frac{\sum_{i=1}^{d} f(u)_i f(v)_i}{\sqrt{\sum_{i=1}^{d} f(u)_i^2}\sqrt{\sum_{i=1}^{d} f(v)_i^2}} \tag{3}$$

In Eq. (3), $f(u)$, $f(v)$ stand for the structural feature vectors of nodes $u,\ v$, respectively. Consecutively, we select r users with high relevancy measure as the relevant vector of each node on both networks. Finally, the nodes that are present in the relevant vector of more nodes, are chosen as most influential nodes on both networks. In other words, we sort the nodes based on their number of occurrence in the relevant vectors of all nodes of interconnected network. Then, the k top frequently used nodes are considered as the seed set for influence spreading on interconnected networks. These nodes are able to transfer information inside and across interconnected networks through bridge nodes. Algorithm 1 describes the process of seed set selection on interconnected networks.

As can be seen in Algorithm 1, we select all the nodes on both networks one by one in line 2. Based on the selected node is either on the first network or the second one, the function $most\_similar(v, r)$ is used in lines 3–7. When the relevant vector of all the nodes are extracted, we enumerate the number of times, each node is visited in relevant vectors in lines 8–12. Then, the most visited nodes are chosen as seed set for IM in line 13. Finally, in line 14, the number of influenced users on both networks is calculated under different diffusion models.

Algorithm 1 Seed selection (M_1, M_2, n, m, r, k).

```
Input:
  Network models $M_1$, $M_2$
  Relevant vector size $r$
  Seed set size $k$
Output:
  Most influential nodes on interconnected nodes
1: initialize $relevant\_vector^{(n+m)\ *\ r}$, $occurance\_vector^{(n+m)\ *\ r}$
2: for each $v$ in network models $M_1$, $M_2$
3:    if $v$ in $M_1$:
4:       $relevant\_vector[v] = M_1 \cdot most\_similar(v, r)$
5:    else:
6:       $relevant\_vector[v] = M_2 \cdot most\_similar(v, r)$
7: end for
8: for $i = 1, \ldots, (n+m)$ do
9:    for $j = 1, \ldots, r$ do
10:       $Occurance\_vector\ [relevant\_vector[i][j]] ++$
11:    end for $j$
12: end for $i$
13: $Seed\_set = occurance\_vector[1 : k]$
14: compute the influence spread of $Seed\_set$ under diffusion models
```

4.3. Monotonicity and submodularity properties in DeepIM

As stated in Section 2, the influence maximization across interconnected networks is NP-hard because it can be considered as vertex cover problem which is NP-complete. Moreover, the proposed influence function $\sigma(S)$ in DeepIM is monotone and submodular. Because, for each node u which is selected as a seed node, we have $\sigma(S+u) \geq \sigma(S)$; Since the seeds are chosen based on their number of occurrences in relevant vectors. Furthermore, $\sigma(S)$ is submodular due to selecting seeds in the order of their presence on *relevant_vectors*. Actually, a submodular function is defined by the following formula:

$$\forall\ S \subseteq T\ \subseteq V,\ \forall\ u\ \in V \backslash T,\ \sigma\ (S+v)\ -\ \sigma\ (s) \geq\ \sigma\ (T+v) -\ \sigma\ (T) \tag{4}$$

While the size of seed set is increased, the margin of influence spread is decreased. Because, the first selected nodes for the seed set have higher occurrence frequency than the latter nodes in seed set. As can be seen, while IM on interconnected networks is NP-hard but the proposed method guarantees a solution with approximation ratio of $(1 - {}^{1}/_{e})$ (Kempe et al., 2003).

4.4. DeepIM complexity

To evaluate time complexity of DeepIM, we determine complexity for each of the mentioned steps, separately. Then we calculate the total complexity of the algorithm. Let n_1 and n_2 are the number of nodes in G_1 and G_2, respectively where $n_1 \geq\ n_2$. Let the total number of nodes in both networks is shown by n. In DeepIM, we first detect communities by Louvain method which its complexity is $O(n_1 log\, n_1 +\ n_2 log\, n_2)$ (Blondel et al., 2008). Then, we generate μ custom path for each node with length l which its complexity is $((\frac{\mu\ \iota}{p})\ n)$, where p is the number of processors in parallel setting. Time complexity of learning feature vector by SkipGram is $O(wd\log n)$ where w is the length of context window and d is the feature vector size. Selecting most relevant nodes is done in $O(n\, r)$ time. Finally, the nodes are sorted in $O\,(n\log n)$ to select most influential nodes.

Overall, the time complexity of DeepIM algorithm is $O\,(n_1 log\, n_1 + \ n_2 log\, n_2 + \ (\frac{\mu\ \iota}{p})\ n + wd\log n + n\ r)$. So, the final time complexity of DeepIM is $O\,(n\log n)$. Based on the latest studies on IM algorithms (Li et al., 2018), DeepIM has the best time complexity between them is $O\,(n^{\,2})$. So, to the best of our knowledge, DeepIM can be used in large real networks to find most influential nodes.

5. Experimental results

In this section, we have conducted several evaluations to measure the efficiency and performance of DeepIM algorithm on different datasets including interconnected networks and NetHept network. We compare our results with a number of baseline algorithms for IM, which are introduced in the following.

5.1. Baseline algorithms

To evaluate the performance of the proposed algorithm, we compare it with the following algorithms that have the best results in IM on a single network.

Greedy (Kempe et al., 2003)**:** Kempe et al. propose a simple greedy method to select seed nodes based on their marginal influence. In greedy method, the influence of each selected node is obtained by several Monte Carlo simulations. So, the greedy approach suffers from scalability issue for large real networks.

Celf++ (Goyal et al., 2011a)**:** Leskovec et al. (2007) propose Celf algorithm to increase the speed of greedy method by utilizing the submodularity property of the influence functions. The basic assumption of Celf algorithm is that most nodes in a social network have very small influences. So, they can be easily pruned at subsequent iterations. Celf++ is an extension of Celf which reduce the number of influence estimation in the next iterations of Celf.

SimPath (Goyal et al., 2011b)**:** Goyal et al. suggest SIMPATH approach to spread the influence of a set of nodes by enumerating all simple paths starting from every node in the set under LT model. SIMPATH restricts the enumeration to a small neighborhood by pruning the length of paths.

LDAG (Chen et al., 2010)**:** In LDAG algorithm, the influence of each node is predicted in a directed acyclic graph called "DAG". Each DAG is constructed by using shortest paths from the starting node in DAG. LDAG algorithm is proposed for LT model.

Table 1
Statistics of interconnected networks.

Dataset	|V|	|E|	Bridge users
DBLP (Data Mining)	9120	12,090	3406
DBLP (Machine Learning)	13,450	18,136	3406
Twitter	5223	164,920	1681
Foursquare	5392	31,312	1681

Parameter settings: To compare our results with the above algorithms, we have used the same parameter settings that are reported in the original algorithms.

5.2. Dataset description

To evaluate IM algorithms, we have used different datasets including DBLP, Twitter-Foursquare and NetHept. To build DBLP networks, we have extracted two citation networks from data mining and machine learning research domains from Aminer dataset (Tang et al., 2008). Each network contains citation relationships between authors of a research area. Edge $(u,\ v)$ illustrates author u cites a paper of author v. There are number of researchers who do research on interdisciplinary research domains. In the extracted dataset, there are 3406 bridge users which are able to apply a new idea from other research domain. In addition to, we evaluate our algorithm on Twitter-Foursquare (Zhang et al., 2013) dataset which is used by previous researches in IM on interconnected networks (Zhan et al., 2015; Zhang et al., 2016). Table 1 illustrates the statistics of each network along with the bridge nodes.

In the following, details of evaluations are presented to measure the performance of DeepIM algorithm against to the other IM approaches.

5.3. Evaluation of different IM algorithms on interconnected networks

In this section, we compare the network coverage of seed sets which are obtained by Greedy, CELF++, SimPath, LDAG and DeepIM. While SimPath and LDAG are suggested for LT model, we compare DeepIM with them on LT model. Furthermore, we compare DeepIM with greedy and CELF++ on IC model. To do fair evaluation, seed sets are first extracted from the networks by each of the baseline algorithms. Then, Ndlib framework (Rossetti et al.,

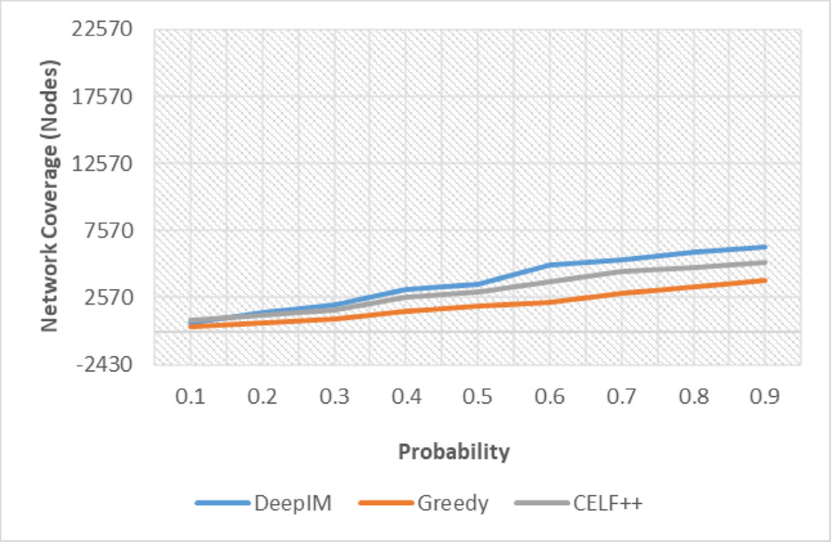


Fig. 2. Comparison of influence spread on IC model in DBLP networks.

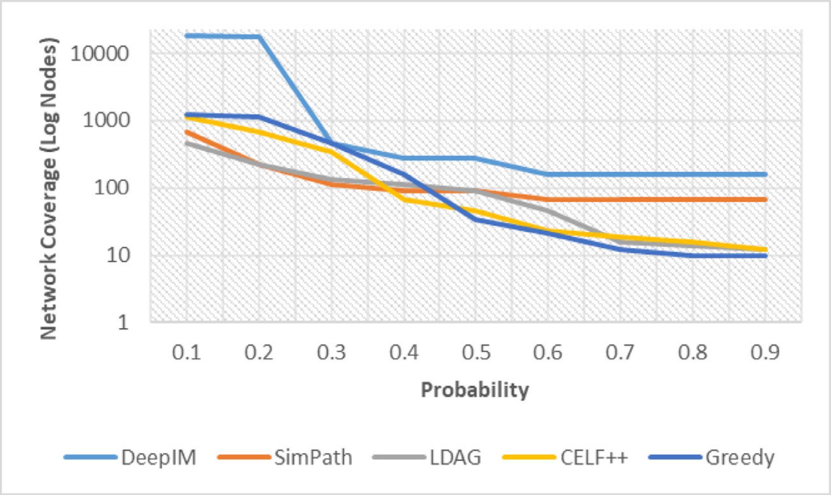


Fig. 3. Comparison of influence spread on LT model in DBLP networks.

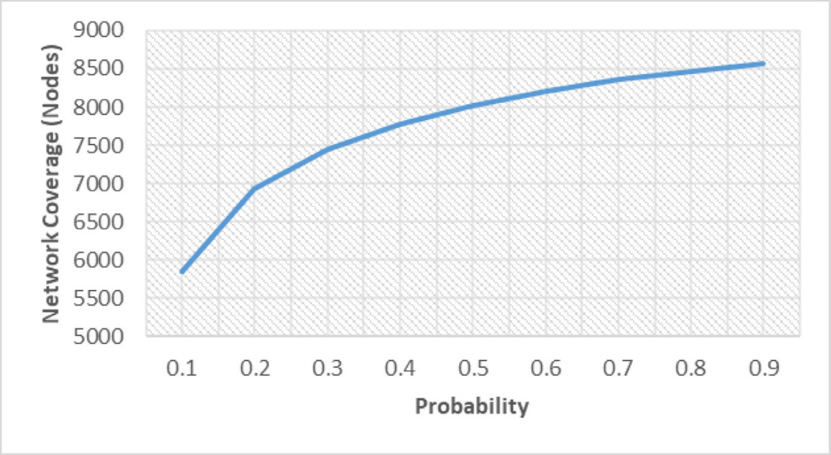


Fig. 4. Influence spread of DeepIM on IC Model in Twitter-Foursquare dataset.

2017, 2018) are employed to estimate the influence of seed sets for all the baseline algorithms on different diffusion models. In Ndlib framework, there is a probability for each diffusion model. In LT model, the probability denotes the threshold value of each network node while in the IC model, it demonstrates the activation probability of each edge during information diffusion. Fig. 2 illustrate the coverage of different algorithms with IC model on DBLP networks.

As shown in Fig. 2, the influence spread of all algorithms is risen with the growth of probability. In the lower probabilities, there is less opportunities to diffuse influence because each edge has small chances to active destination node, whereas in higher probabilities, a considerable number of users can be affected by the seed set which are chosen by DeepIM because it leverages different channels of influence propagation during the feature selection, while the other methods simply regarded to local neighborhood. It should be noted SimPath and LDAG are designed originally for LT model, so we don't compare DeepIM with them on IC model. Fig. 3 depicts the performance of different approaches based on LT model on DBLP interconnected networks.

In Fig. 3, the number of influenced users are dropped with the growth of threshold probability for each nodes because each user requires more active neighbors to be influenced. But in the lower threshold values, DeepIM achieve better performance because its selected seed set have relations with a large number of users on both networks based on our feature engineering strategies during network embedding. As it is apparent in Figs. 2 and 3, DeepIM has higher network coverage in comparison to the baseline algorithms. Besides, we carry out the same evaluations for DeepIM algorithm on Twitter-Foursquare dataset which its results is demonstrated in Figs. 4 and 5. It's worth noting that other algorithms cannot obtain seed set for this dataset in 10 days while DeepIM select seed set in 12 h. So, we just report the performance of DeepIM.

As it is shown in Fig. 4, DeepIM algorithm can approximately spread the influence to the whole users of both Twitter and Foursquare networks. While in DBLP dataset, all the algorithms infect a few number of nodes during influence propagation. Its main reason is that the density of Twitter-Foursquare dataset is much higher than DBLP networks. Fig. 5 shows the network coverage of DeepIM based on LT model on Twitter-Foursquare dataset.

In Fig. 5, Similar to DBLP dataset, the network coverage of DeepIM drastically diminish when the threshold values of each node are increased but the total number of influenced users are higher than DBLP interconnected networks. Overall, the network coverage of different algorithms are dependent to the underlying networks but in IC model, the more extent of network nodes can be influence than LT model.

5.4. Analysis of DeepIM

To understand the benefit of taking consideration of DeepIM algorithm, we are going to investigate the role of different pa-

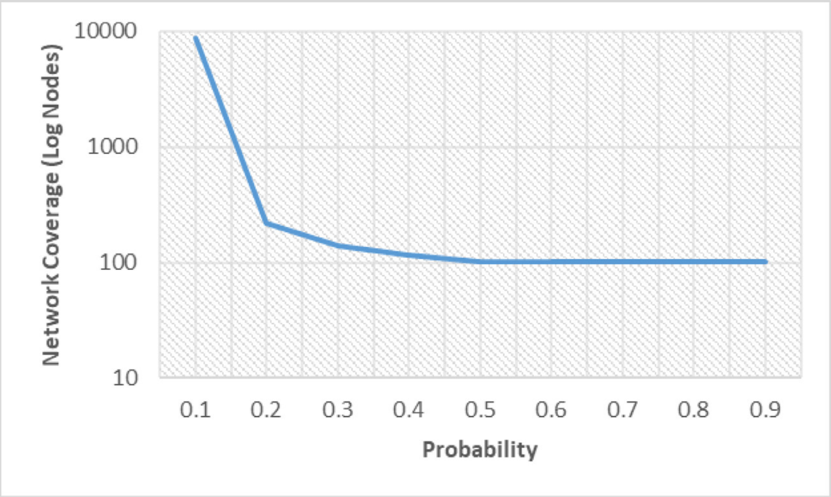


Fig. 5. Influence spread of DeepIM on LT model in Twitter-Foursquare dataset.

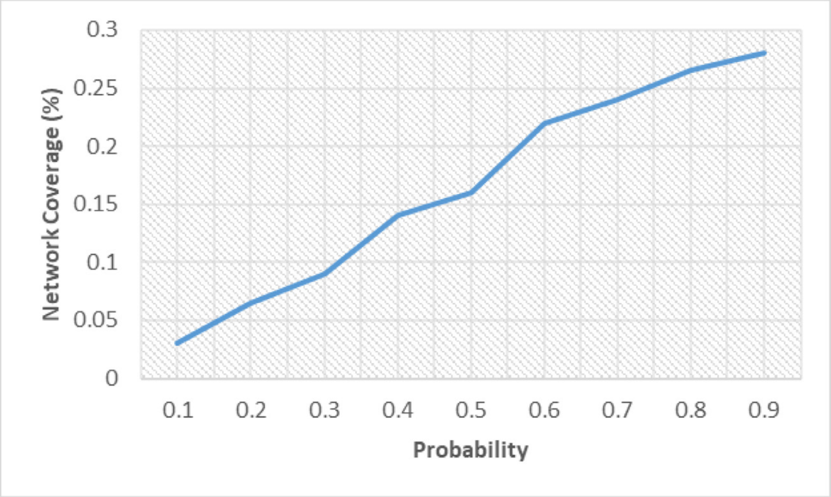


Fig. 6. DeepIM coverage with different diffusion thresholds on IC.

rameters on it. Fig. 6 shows the effect of threshold probability on DeepIM algorithm.

In this experiment, a seed set of size 50 is first extracted by DeepIM. Then, the influence spread of the seed set is measured on IC model. As it's apparent in Fig. 6, in the lower edge activation probability, a few percent of nodes on both networks can be infected by the seed set. When the threshold reaches to 0.5, about 16% of network nodes can be influenced by the seed set. While the other algorithms can achieve to the same network coverage with a seed set of size more than 50 which is demonstrated in Fig. 2.

A detail investigation of seed set of DeepIM indicates near to 20% of seed set users are chosen from bridge users across interconnected networks. This valuable finding verifies the role of bridge users on transferring external influence to the other nodes on both networks. We have also examine the impact of seed size on DeepIM algorithm in Fig. 7.

In Fig. 7, while the network coverage growth monotonic with the seed size, but the marginal influence decreases in the upper seed size which is verified the submodularity property of DeepIM influence function. In addition to, the maximal margin is happened when the number of seed users is in the range of 35 to 50. But when the size of seed set is 35, a more number of iterations is required to spread the influence through the networks. Fig. 8 shows the number of iterations for each seed size in DeepIM algorithm.

As can be seen in Fig. 8, an optimal seed size for DeepIM with consideration of spreading time is 50 which it can maximize the influence on both interconnected networks. It should be mentioned, while other baseline algorithms need to run about 10,000 Monte Carlo simulation to find optimal seed set on both networks, DeepIM passes through the networks 3 times including network modeling, building relevant vector and selecting most relevant nodes. In other words, the previous approaches use all the networks' edges, while DeepIM generate a fixed number of custom path to capture the node's structural features. Thus, DeepIM is very faster than the previous approaches on running time and time complexity.

5.5. Evaluation of different IM algorithms on single network

As stated in the previous sections, DeepIM can also be used for influence maximization on single networks in a similar way to the previous researches (Goyal et al., 2011a, 2011b; Kempe et al., 2003; Leskovec et al., 2007; Liu et al., 2014b). In this section, we have evaluated the performance of DeepIM in comparison to the previous algorithms on NetHept dataset. NetHept network contains 15,200 nodes and 61,300 edges which illustrate the collaboration of researchers in High Energy physics which are extracted from arX-

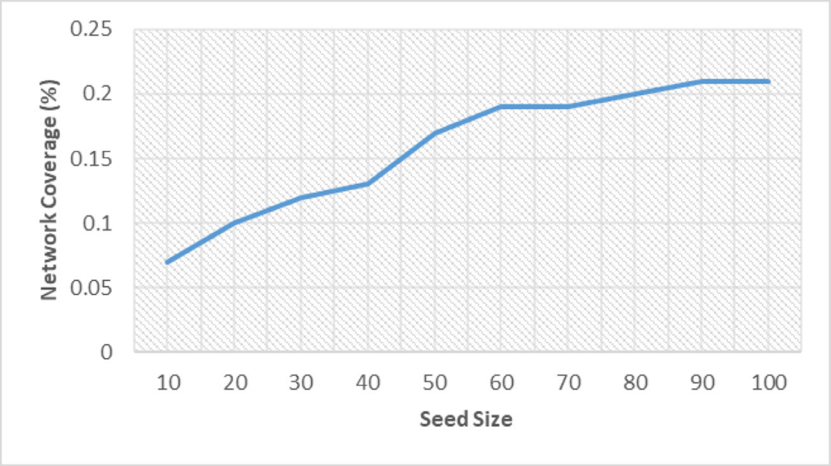


Fig. 7. The effect of seed set size on influence spread by DeepIM.

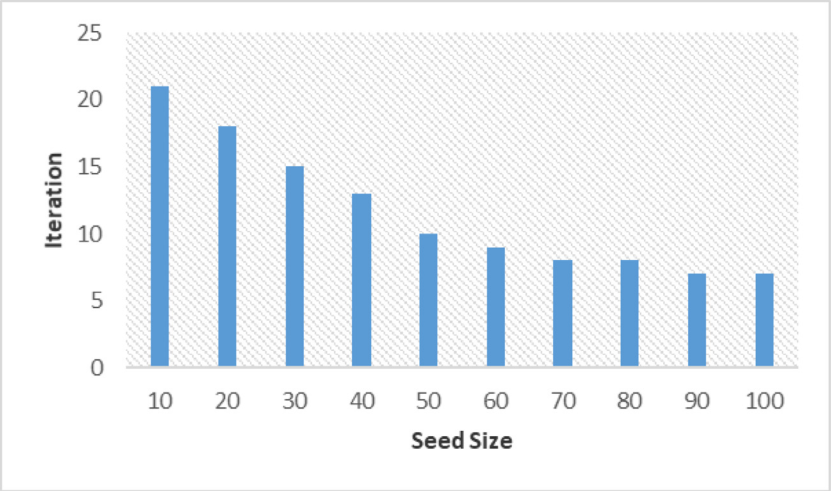


Fig. 8. Number of iteration for different seed size on DeepIM.

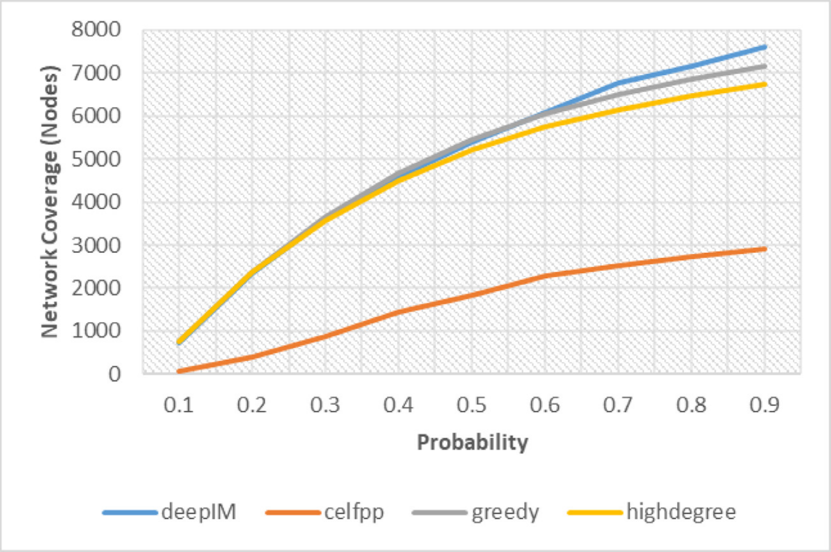


Fig. 9. Influence spread of IM algorithms on IC model in NetHept network.

ive dataset. In this dataset, nodes show the authors of papers and links demonstrate the collaboration of two authors in a paper.

In the experiments, we first select 50 most influential nodes from the network by different baseline algorithms. Afterwards, the influence spread of each seed set on IC model is measured with Ndlib framework (Rossetti et al., 2017, 2018). Fig. 9 shows the network coverage of different algorithms in IC model for NetHept network.

As it's clear in Fig. 9, though CELF++ select most influential nodes faster than the other algorithms, it has lower network coverage in comparison to the other approaches. Because, a number of valuable candidate nodes are pruned in the first stages of the algorithm. Thus, CELF++ falls into a local optimum and it won't be able to select most influential nodes for IM. In Fig. 9, in the smaller probabilities, the network coverage of the most approaches is the same, but when the activation probability of edges is increased, a more number of users are infected in the network. Because in higher probabilities, there are more activated paths for influence propagation. So, DeepIM can transfer the influence through these new discovered paths. While in the lower probabilities, these paths are not active yet, because their edges has lower probabilities.

6. Conclusion

Recently, many people tend to join multiple online social networks which are considered as bridge users. In this paper, a novel influence maximization algorithm on interconnected networks is presented which leverages deep learning techniques. In the proposed algorithm which is named DeepIM, all the structural properties of networks are employed to maximize the influence. To learn the feature vector of nodes, we generate a number of custom paths for each user which contain neighbors of the current nodes on the path as local structural information and community members of the current node on the custom path as global structural information. Then, Word2vec model is employed to learn the best structural features based on the paths. Next, the nodes are sorted based on their relevancy to the other network nodes on interconnected networks. Finally, most influential nodes are chosen from the sorted relevant list of nodes. We proof the submodularity and monotonicity of the influence function. So, DeepIM guarantees an optimal solution with the ratio of $(1 - {}^{1}/_{e})$ approximation.

Experimental results verify the performance of DeepIM on interconnected networks in comparison to the other algorithms. We illustrate that bridge users have a major role to maximize the influence on interconnected networks and about 20% of bridge users are chosen as most influential nodes on interconnected networks. In addition to, DeepIM is faster than the previous researches because it uses a fixed number of edges while the previous approach should examine all the edges several times to find the best seed set. We can influence more than users on both networks in comparison to the state of the art algorithms with a fixed seed set size which it implies the effectiveness of the proposed method. We are going to use some heuristics to find most relevant nodes in the future. Besides, we are going to use context information such as profile and text during feature engineering.

Declaration of Competing Interest

None.

Credit authorship contribution statement

Mohammad Mehdi Keikha: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Maseud Rahgozar:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration. **Masoud Asadpour:** Conceptualization, Formal analysis, Writing - original draft, Supervision. **Mohammad Faghih Abdollahi:** Software, Validation, Investigation, Resources, Data curation, Writing - review & editing.

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript, but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

Bakshy, E., Rosenn, I., Marlow, C., & Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web - WWW '12* (p. 519). New York, New York: ACM Press. https://doi.org/10.1145/2187836.2187907.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics, 2008*(10). https://doi.org/10.1088/1742-5468/2008/10/P10008.

Borgs, C., Brautbar, M., Chayes, J., & Lucier, B. (2014). Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth Annual ACM-SIAM symposium on discrete algorithms* (pp. 946–957). Philadelphia, PA: Society for Industrial and Applied Mathematics. Retrieved from. http://dl.acm.org/citation.cfm?id=2634074.2634144 .

Budak, C., Agrawal, D., & El Abbadi, A. (2011). Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web - WWW '11* (p. 665). New York, New York: ACM Press. https://doi.org/10.1145/1963405.1963499.

Chen, W., Wang, C., & Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '10* (p. 1029). New York, New York: ACM Press. https://doi.org/10.1145/1835804.1835934.

Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '09* (p. 199). New York, New York: ACM Press. https://doi.org/10.1145/1557019.1557047.

Chen, W., Yuan, Y., & Zhang, L. (2010). Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining* (pp. 88–97). IEEE. https://doi.org/10.1109/ICDM.2010.118.

Cohen, E., Delling, D., Pajor, T., & Werneck, R. F. (2014). Sketch-based influence maximization and computation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management - CIKM '14* (pp. 629–638). New York, New York: ACM Press. https://doi.org/10.1145/2661829.2662077.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01* (pp. 57–66). New York, New York: ACM Press. https://doi.org/10.1145/502512.502525.

Gaeta, R. (2018). A model of information diffusion in interconnected online social networks. *ACM Transactions on the Web, 12*(2), 1–21. https://doi.org/10.1145/3160000.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters, 12*(3), 211–223. https://doi.org/10.1023/A:1011122126881.

Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011a). CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web - WWW '11* (p. 47). New York, New York: ACM Press. https://doi.org/10.1145/1963192.1963217.

Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011b). SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In *2011 IEEE 11th international conference on data mining* (pp. 211–220). IEEE. https://doi.org/10.1109/ICDM.2011.132.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology, 83*(6), 1420–1443. https://doi.org/10.1086/226707.

Grover, A., & Leskovec, J. (2016). Node2Vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864). New York, NY: ACM. https://doi.org/10.1145/2939672.2939754.

Hinton, G., & Salakhutdinov, R. (2011). Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science, 3*(1), 74–91. https://doi.org/10.1111/j.1756-8765.2010.01109.x.

Huang, P., Huang, Y., Wang, W., & Wang, L. (2014). Deep embedding network for clustering. In *2014 22nd international conference on pattern recognition* (pp. 1532–1537). IEEE. https://doi.org/10.1109/ICPR.2014.272.

Keikha, M. M., Rahgozar, M., & Asadpour, M. (2018). Community aware random walk for network embedding. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2018.02.028.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining - KDD '03* https://doi.org/10.1145/956755.956769.

Kimura, M., & Saito, K. (2006). Tractable models for information diffusion in social networks. *Knowledge Discovery in Databases*. https://doi.org/10.1007/11871637_27.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1–9. https://doi.org/10.1016/j.protcy.2014.09.007.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '07* (p. 420). New York, New York: ACM Press. https://doi.org/10.1145/1281192.1281239.

Li, Y., Fan, J., Wang, Y., & Tan, K.-. L. (2018). Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*. 1–1 https://doi.org/10.1109/TKDE.2018.2807843 .

Liu, Q., Xiang, B., Chen, E., Xiong, H., Tang, F., & Xu Yu, J. (2014a). Influence maximization over large-scale social networks : A bounded linear approach. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* https://doi.org/10.1145/2661829.2662009.

Liu, Q., Xiang, B., Chen, E., Xiong, H., Tang, F., & Yu, J. X. (2014b). Influence maximization over large-scale social networks. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management - CIKM '14* (pp. 171–180). New York, New York: ACM Press. https://doi.org/10.1145/2661829.2662009.

Liu, X., He, Q., Tian, Y., Lee, W.-. C., McPherson, J., & Han, J. (2012). Event-based social networks: Linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '12* (p. 1032). New York, New York: ACM Press. https://doi.org/10.1145/2339530.2339693.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Distributed representations of words and hrases and their compositionality. In *NIPS* (pp. 1–9).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. Retrieved from http://arxiv.org/abs/1301.3781

Mohamed, A., Yu, D., & Deng, L. (2010). *Investigation of full-sequence training of deep belief networks for speech recognition* (pp. 2846–2849). Interspeech. (September)Retrieved from. http://www.isca-speech.org/archive/interspeech_2010/i10_2846.html%5Cnhttp://131.107.65.14/pubs/135406/MMI-DBN-interspeech2010.pdf .

Nguyen, D. T., Zhang, H., Das, S., Thai, M. T., & Dinh, T. N. (2013). Least cost influence in multiplex social networks: Model representation and analysis. In *2013 IEEE 13th international conference on data mining* (pp. 567–576). IEEE. https://doi.org/10.1109/ICDM.2013.24.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations bryan. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '14* (pp. 701–710). https://doi.org/10.1145/2623330.2623732.

Rossetti, G., Milli, L., Rinzivillo, S., Sirbu, A., Pedreschi, D., & Giannotti, F. (2017). NDlib: Studying network diffusion dynamics. In *2017 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 155–164). IEEE. https://doi.org/10.1109/DSAA.2017.6.

Rossetti, G., Milli, L., Rinzivillo, S., Sîrbu, A., Pedreschi, D., & Giannotti, F. (2018). NDlib: A python library to model and analyze diffusion processes over complex networks. *International Journal of Data Science and Analytics*, *5*(1), 61–79. https://doi.org/10.1007/s41060-017-0086-6.

Shen, Y., Dinh, T. N., Zhang, H., & Thai, M. T. (2012). Interest-matching information propagation in multiple online social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12* (p. 1824). https://doi.org/10.1145/2396761.2398525.

Tang, J., & Qu, M. (2015). LINE : Large-scale information network embedding categories and subject descriptors. *ACM World Wide Web*, 1067–1077. https://doi.org/10.1145/2736277.2741093.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). ArnetMiner: Extraction and mining of academic social networks. In *Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining - KDD 08* (p. 990). New York, New York: ACM Press. https://doi.org/10.1145/1401890.1402008.

Wang, Y., Cong, G., Song, G., & Xie, K. (2010). Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '10* (p. 1039). New York, New York: ACM Press. https://doi.org/10.1145/1835804.1835935.

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd international conference on international conference on machine learning - volume 48*. JMLR.org Retrieved from https://dl.acm.org/citation.cfm?id=3045442 .

Yagan, O., Qian, D., Zhang, J., & Cochran, D. (2012). Information diffusion in overlaying social-physical networks. In *2012 46th annual conference on information sciences and systems (CISS)* (pp. 1–6). IEEE. https://doi.org/10.1109/CISS.2012.6310749.

Ye, M., Liu, X., & Lee, W.-. C. (2012). Exploring social influence for recommendation: A generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval* https://doi.org/10.1145/2348283.2348373.

Zhan, Q., Zhang, J., Wang, S., Yu, P. S., & Xie, J. (2015). Influence maximization across partially aligned heterogenous social networks. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* https://doi.org/10.1007/978-3-319-18038-0_5.

Zhan, Q., Zhang, J., Yu, P. S., Emery, S., & Xie, J. (2016). Discover tipping users for cross network influencing. In *Proceedings - 2016 IEEE 17th international conference on information reuse and integration, IRI 2016* https://doi.org/10.1109/IRI.2016.17.

Zhang, H., Nguyen, D. T., Zhang, H., & Thai, M. T. (2016). Least cost influence maximization across multiple social networks. *IEEE/ACM Transactions on Networking*, *24*(2), 929–939. https://doi.org/10.1109/TNET.2015.2394793.

Zhang, J., Kong, X., & Yu, P.S. (.2013). Predicting social links for new users across aligned heterogeneous social networks. Retrieved from http://arxiv.org/abs/1310.3492