Influence Diffusion Model in Multiplex Networks

Senbo Chen[1, 3, *] and Wenan Tan[1, 2]

Abstract: The problem of influence maximizing in social networks refers to obtaining a set of nodes of a specified size under a specific propagation model so that the aggregation of the node-set in the network has the greatest influence. Up to now, most of the research has tended to focus on monolayer network rather than on multiplex networks. But in the real world, most individuals usually exist in multiplex networks. Multiplex networks are substantially different as compared with those of a monolayer network. In this paper, we integrate the multi-relationship of agents in multiplex networks by considering the existing and relevant correlations in each layer of relationships and study the problem of unbalanced distribution between various relationships. Meanwhile, we measure the distribution across the network by the similarity of the links in the different relationship layers and establish a unified propagation model. After that, place on the established multiplex network propagation model, we propose a basic greedy algorithm on it. To reduce complexity, we combine some of the characteristics of triggering model into our algorithm. Then we propose a novel MNStaticGreedy algorithm which is based on the efficiency and scalability of the StaticGreedy algorithm. Our experiments show that the novel model and algorithm are effective, efficient and adaptable.

Keywords: StaticGreedy, social networks, influence maximization, multiplex networks.

1 Introduction

With the exponential growth of information in social networks, the speed of information dissemination is accelerating, and the way people obtain information is becoming wider. The information-gathering structure based on social network communication enables people to disseminate and acquire information [Chen, Li, Zhang et al. (2018)]. In a sense, social network information dissemination is based on the integration of mass communication and interpersonal communication, where a new distributed network is constructed [Bond, Fariss, Jones et al. (2012); Gaye, Mendy, Ouya et al. (2015)]. The unprecedented improvement of the dissemination mechanism has further accelerated

[1] School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210000, China.

[2] School of Computer and Information, Shanghai Second Polytechnic University, Shanghai, China.

[3] Department of Infection, Immunity and Inflammation, University of Leicester, Leicester, LE1 7RH, UK.

[*] Corresponding Author: Senbo Chen. Email: sam_c@163.com.

Received: 19 January 2020; Accepted: 04 March 2020.

interpersonal communication and information dissemination [Chen, Hua, Yuan et al. (2018)]. In this kind of communication mechanism, the diffusion of information between users is often affected by user influence. In the research on the social network, how the information spreads through the interconnections of the people in the network has drawn more and more attention. The problem of influence maximizing in social networks refers to obtaining a set of nodes of a specified size under a specific propagation model so that the aggregation of the node-set in the network has the greatest influence. Essentially, the user's behavior and thoughts will be impacted by others. Meanwhile, the user's own behavior and thoughts will affect others. According to the characteristics of influence diffusion, when promoting new products, companies often apply advertising or "viral marketing" methods on social networks to expand their market share [Li, Luo, Huang et al. (2012); Mehmood, Barbieri, Bonchi et al. (2013); Wu, Liu, Yue et al. (2015)]. When formulating specific strategies, it is necessary to mine users who play a key role in information diffusion according to the influence of users to maximize the effectiveness of publicity. Besides, through the analysis of the influence of users and the use of influence to guide the public opinion, it can promote the relevant policies of the government and prevent the spread of harmful information. It can also effectively serve real-time monitoring, forecasting and early warning and emergency response of social security [Tsai, Yang and Chiang (2015)]. Therefore, research on maximizing the influence of social networks can make better use of the openness and freedom of online speech to serve social public safety.

Multiplex networks [Zhang, Nguyen, Zhang et al. (2016)] are networks in which the same of nodes are connected by distinct types of links organized in different interacting layers. This topic had received more and more attention in recent years. Some recent studies have shown that the topological properties of a multiplex network are substantially different as compared with those of a monolayer network and that the interaction of layers of a multiplex network can generate new interesting diffusion processes [Kuhnle, Alim, Li et al. (2018)].

In this study, we first put forward a linear threshold (LT) diffusion model in multiplex networks (named MNTM) for describing more than two types of influence spreads which can obtain better results. Specifically, we make the following contributions in this paper:

Our contributions can be summarized as follows:

1. We integrate the multi-relationship of agents by considering the existing and relevant correlations in each layer of relationships and establish a unified influence diffusion model.

2. Based on the established multiplex network propagation model, we proposed a novel MNStaticGreedy algorithm which is based on the efficiency and scalability of the StaticGreedy algorithm.

2 Preliminaries

2.1 Influence maximization in the single level relation network

There are two basic models for studying the maximization of influence propagation: The Independent Cascade Model (ICM) and the Linear Threshold Model (LTM) [Kempe, Kleinberg and Tardos (2003); Rahimkhani, Aleahmad, Rahgozar et al. (2015); Galhotra, Arora, Virinchi et al. (2015)]. The influence between the user and its neighbor nodes in the independent cascade model is independent of each other. It is not affected by the relationship between neighbor nodes in the influence propagation process. The difference from the independent cascade model is that the linear threshold model considers the cumulative effect of influence. On this basis, Kempe et al. [Kempe, Kleinberg and Tardos (2003)] extended the ICM and LTM models, abandoned the independence conditions, and proposed the General Cascade Model (GCM) and the general threshold model (GTM). These two extended propagation models can be converted to each other and are considered as two mutually equivalent models. Shultz et al. [Shultz and Rivest (2000)] proposed a Weighted Cascade Model (WCM) based on the factor of node degree based on the ICM. The edges associated with the generous nodes in the model are on a lower activation probability, so the WCM is a particular case of the ICM. Kempe et al. [Kempe, Kleinberg and Tardos (2005)] proposed a Decreasing Cascade Model (DCM) based on the idea that the probability that a node is activated may change as the neighboring node attempts to enable it. They believe that if multiple nodes have not successfully activated the target node for many times, the impact of the newly activated neighbor node on the target node will be weakened. Fazli et al. [Fazli, Ghodsi, Habibi et al. (2014)] believe that nodes can convert between active and inactive states, so a non-progressive model (NPM) is proposed. Li et al. [Li, Luo, Huang et al. (2012)] proposed a multi-layer network (MLN) based on the characteristics of Weibo data. Although that model combines the user's topology information and attribute information through linear combination, it does not consider the different influences of different relationship layers on users and the correlation between the relationship layers.

Kempe et al. [Kempe, Kleinberg and Tardos (2005)] took the lead in studying the problem of maximizing influence in detail and proved that it is an NP-hard problem. On this basis, an approximate greedy algorithm suitable for most influence propagation models is proposed [Mehmood, Barbieri, Bonchi et al. (2013)]. At each step, the algorithm puts the most influential node as a candidate node into the seed-set, and then iterates continuously until all seed nodes are selected. However, the optimal local strategy of this algorithm does not guarantee the global optimization of the final result, and the efficiency of the algorithm is relatively low. The time complexity is high, which is not suitable for actual large-scale networks. On this basis, Tsai et al. [Tsai, Yang and Chiang (2015)] improved the greedy algorithm and proposed the GNG algorithm (Genetic NewGreedy, GNG). Experiments show that the algorithm combines some characteristics of the genetic algorithm to improve the performance of the greedy algorithm by about 10%. To solve the dilemma of the accuracy and scalability of the influence maximization algorithm, Cheng et al. [Cheng, Shen, Huang et al. (2013)] proposed the heuristic SG algorithm (StaticGreedy, SG). The algorithm utilizes the submodular of the influence maximizing objective function to select the currently most influential node, thereby reducing the time taken for candidate node selection. Tang et al. [Tang, Xiao and Shi (2014)] propose a Reverse Reachable-SKETCH approach (TIM). After that, Tang et al. [Tang, Shi and Xiao (2015)] further propose IMM to improve over TIM by using a martingale analysis and a better bootstrap estimation technique. In recent years, several studies have a focus on the key influencer in the network for influence maximizing problem [Hua, Chen, Yuan et al. (2019)].

2.2 StaticGreedy in traditional network

StaticGreedy algorithm [Cheng, Shen, Huang et al. (2013)] produces several Monte Carlo snapshots at the very beginning, and uses the same set of snapshots in all iterations. Those snapshots are called "static". The algorithm ensures that the estimated impact range of each seed set is consistent in different iteration cycles. This guarantees submodular and monotonicity. Avoiding a vast number of Monte Carlo simulations needed in every iteration, this algorithm brings the possibility to reduce the computational expense without loss of accuracy significantly.

Table 1: StaticGreedy algorithm

Algorithm1 StaticGreedy(G, k, R)
initialize S=ϕ
for i=1 to R do
generate snapshot $G_i^{'}$ by removing each edge (u, v) from G with probability $1 - p(u,v)$
end for
for i=1 to k do
set s_v=0 for all v $\in$ V \ S // s_v stores the influence spread after adding node v
for j=1 to R do
for all v $V\backslash S$ do
s_v+=|R($G_j^{'}$,S $\cup$ {v})| //R($G_j^{'}$, S $\cup$ {v}) is the influence spread of S$\cup${v} in snapshot $G_j^{'}$
end for
end for
S=S$\cup${ $argmax_{v\in V\backslash S}$ { s_v /R } }
end for
output S

3 Diffusion model in multiplex networks

3.1 Multiplex networks threshold model

In a multiplex network, when different relationships are merged, we need to establish a unified model for m different types of link relationships. The Multiplex Networks Threshold Model (MNTM) is exhibited in Fig. 1. As the separate relationship layer that has a distinct impact on the agent, some of the relationships may be stronger than others, and we intend to increase the weight of such relation links. Conversely, for the relationship layer that has week impact on the node, we want to reduce the weight of such links.

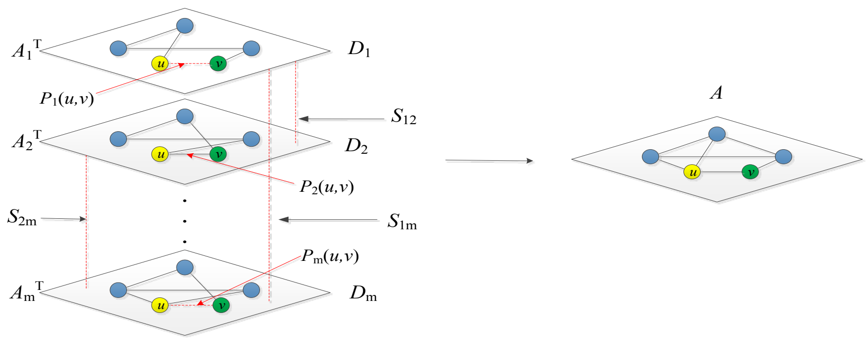

Figure 1: Multiplex networks threshold model

We need to calculate the correlation between agent u and the r-th relationship layer. Let $\Gamma_r(u)$denote the set of neighbor nodes of node u in the r-th relational layer, and $d_r^k(u)$ be the number of k level neighbor of node u in the r-type relational layer, where k can take 1 or 2. Let $c_r(u, v)$ denote the correlation of vertex u and v in the r-th relational layer. We can use the quantity of links or the remaining topological attributes of nodes to measure it. The $c(u, v)$ is the sum of the correlation values of the vertices u and v in all relational layers i.e.,

$$c(u,v)=\sum_{i=1}^{m}c_i(u,v) \tag{1}$$

Let $R_r(u)$ be Correlation between node u and r-type relationship layer, it can be formed like

$$R_r(u)=\frac{1}{d_r^k(u)}\sum_{v\in\Gamma_r(u)}\frac{c_r(u,v)}{c(u,v)} \tag{2}$$

Similarly, we can get the correlation between the vertex v and the r-th relationship layer $R_r(v)$. When we merging distinct relation links of edge (u, v), their weights on the r-th relationship layer can be calculated by the following formula:

$$\boldsymbol{P_r(u,v)}=\frac{\boldsymbol{R_r(u)+R_r(v)}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}\left(\boldsymbol{R_i(u)+R_i(v)}\right)} \tag{3}$$

It can be seen from the formula that when the sum of $R_r(u)+R_r(v)$ increasing, the weight of the node pair (u, v) in r-th relationship. Therefore, we can get the correlation between all pairs of nodes, the formula is as follow:

$$A(u,v)=\sum_{i=1}^{m}\left(P_i(u,v)\cdot c_i(u,v)\right) \tag{4}$$

On this basis, we must also study the problem of unbalanced distribution between various relationships. We intend to measure their distribution across the network by the similarity of the links in the different relationship layers. Consider two relation layers i-th layer and j-th layer, let E_i and E_j denote the path set with the same length l in the two layers. K is the maximum path length between any two points in each layer, we define the similarity of this relationship type as S_{ij}, and the formula can be as follow:

$$S_{ij}=\frac{\sum_{l=1}^{K}\eta^{K-l+1}\cdot|E_i^l\cap E_j^l|}{\sum_{l=1}^{K}|E_i^l\cap E_j^l|} \tag{5}$$

$\eta\in(0,1)$ denotes the weight coefficient. If the common paths' number in i-th and j-th relationship are big, the more similar they are to the topological similarity. The higher weight should be given to it, i.e., the more similar the two relationship layer topologies are, the higher the value S_{ij}. After that, we define the degree of centrifugation between all links, i.e., the distribution of all links in i-th relationship layer. Let D_i formula can be as follow:

$$D_i=\frac{\lambda}{\sum_{j=1,j\neq i}^{m}S_{ij}} \tag{6}$$

λ is a constant. It can be seen from the formula that for relation layer links with a significant degree of centrifugation, to prevent their links from being masked due to uneven distribution, we should give higher weight to them. So we improve the topological correlation $W(u,v)$ between the pairs of nodes. The formula can be as follow:

$$W(u,v)=\sum_{i=1}^{m}\left(D_i\cdot P_i(u,v)\cdot c_i(u,v)\right) \tag{7}$$

In the influence propagation model, assume that there is an active state function f for each agent. If the total influence value $p(u,v)$ from neighbors is higher than the threshold value θ_v of the agent, the agent will be active otherwise will not. The $q_1(u)$ denotes the nodes set which include active neighbors. $p(u,v)$ formula is as follow:

$$p(u,v)=\sum_{v\in q_1^T(u)}W(u,v) \tag{8}$$

Therefore, assuming that the initial seed node-set is S, the propagation process of the MNTM model can be briefly described as follow:

I) Merge multiplex layer network into one network by the method we have proposed above.

II) Chose a threshold value for each agent in the network.

III) In t step, if agent $v\in V\backslash S$, $v\neq\phi$ and $p(u,v)>\theta_v$, agent v will be active otherwise will not.

IV) The propagation process will finish when no new agent is being active.

3.2 Influence maximization in multiplex networks

Based on the above, let σ_s denote the influence range of seed set S. Our influence maximization in multiplex networks problem can be descript as Input $G=\{G_1, G_2, \dots G_i\}$, $G_i=(V,E)$, G include i distinct type relationship networks, give a constant $k<|V|$, finding a seeds set S, when $|S|=k$, σ_s is maximization.

The MNTM model considers the impact of topological relationships between multiple relational networks on propagation, as well as the overlapping effects of nodes between multiplex networks and the strength of relationships among them. The propagation mechanism of MNTM is very similar to the traditional LT model. So the MNTM model is still an impact accumulation model. It has the same conditions as a typical LT model when it successfully activates a node during the propagation process. Therefore, it can be proved that under the MNTM model, the influence propagation function σ_s is also submodular and monotone. Therefore, the greedy algorithm can also be used to approximate the influence of the multiplex network under the MNTM model. Based on the above theory, we can give the basic greedy algorithm on the multiplex network as Algorithm 2.

Table 2: Greedy algorithm on MNTM

Algorithm2 BasicMNGreedy(G,k)
1: calculate $W(u,v)$ from G set
2: generate multiplex G' from G
3: $S=\phi$
4: **for** i=1~k,$\|S\|=i$ **do**
find $u= argmax_{v\in V\setminus S}\{\sigma(S\cup u)-\sigma(S)\}$
5: $S=S\cup u$
6: **return** S

From the Algorithm 2, we can see that the process of propagating the search seed still uses the traditional greedy algorithm. Based on the unique feature of the multiplex network, we merge multiple networks into an entire network according to the previously proposed multiple-layer relationship fusion method. It can easily be expanded.

3.3 MNStaticGreedy

Based on the proposed MNTM and basic greedy algorithm, we can find that the new propagation process needs to consider the mutual integration of multi-layer relationships and the shortcoming of the greedy algorithm is the problem of time complexity. Vast numbers of iterative iterations will be processed. These problems need an efficient and scalable algorithm. Most of the existing algorithms are specific to specific conditional constraints and frameworks, while the traditional StaticGreedy is for simple relationship cascaded propagation. It has good scalability and effect. It can combine with a lot of different algorithms. Meanwhile, Tang et al. [Tang, Xiao and Shi (2014)] proposed a method based on triggering model and reverse reachable (RR) set. Under the LT model (which is s special case of the triggering model), $T(v)$ denotes a sample from triggering distribution of v, it has p_i probability to be a unit set containing the i-th incoming neighbor of v, and has $1-\sum_{i=1}^{x} p_i$ probability to be an empty set. Based on the above problems and methods, we propose our MNStaticGreedy algorithm. The algorithm consists of three steps:

1. Building a multiplex network by merge different type networks. Generate R snapshot of multiplex networks using $W(u,v)$.

2. Then random select n nodes for each snapshot to generate their reverse reachable set. The reverse reachable set *RRv* of node v represents a set of all nodes on the multiplex network that can reach v. All *RRv* set ultimately constitutes the reverse reachable set *RRi* for each snapshot.

3. Select the *RR* set that contains the snapshot of the most nodes. The seed node is selected by the maximum coverage method. At the same time, the more nodes in the reverse reachable concentration, the more nodes that the node can activate.

The description of *MNStaticGreedy* is given in Algorithm 3.

Table 3: MNStaticGreedy Algorithm

Algorithm3 MNStaticGreedy(G,k,R)
calculate $W(u,v)$ from G set
generate multiplex G' from G
for i=1 to R
generate snapshot G_i' by removing each edge (u, v) from G' with probability $1-p(u,v)$
end for
$S=\Phi$, $RR=\Phi$
for i=1 to R do
for j=1 to n do
random sample a node v generates RR_v set
end for
end for
$RR=\max\{\lvert V_i\rvert, i\in [1,R]\}$
for 1 to k do
$S=S\cup u$ //u cover the most RR_v
$RR_v = RR_v \backslash RR_u$
end for
return S

It can be seen that the algorithm combines the characteristics of the *StaticGreedy* algorithm and the *RRset* method. The snapshot generation and the *RR* generation can be merged in actual operation. After finding seeds, the simulation propagation can directly use the generated snapshot instead of using a large number of Monte Carlo simulations.

4 Experiments

4.1 Experiment setting

We consider Triggering Model of information diffusion for our experiments. We use the intersection method as described in Section 3 for generating node distribution. All code is implemented in Python. Code is executed on a standalone machine with Intel Core i7 processor, 8 GB of RAM.

Two real-world networks are employed to demonstrate the performance of our algorithms by comparing with other existing algorithms. These two datasets are both undirected as shown in Tab. 4. ca-GrQc is a collaboration network. It is from the e-print arXiv (http://www.arXiv.org) and covers scientific collaborations between authors papers submitted to General Relativity and Quantum Cosmology category. ca-HepPh's data came from High Energy Physics-Phenomenology category. Since this is two overlapping categories, many authors publish articles in both categories. So we can consider them as related networks of two different types of relationships. We choose two algorithms as the benchmark, one is the classic LT algorithm LDAG, and the other is the ordinary StaticGreedy algorithm.

Table 4: Statistics of two real world networks

Datasets	#Nodes	#Edges	Directed?
ca-GrQc	5,242	14,496	undirected
ca-HepPh	12,008	118,521	undirected

4.2 Results

This paper compares MNStaticGreedy and LDAG, StaticGreedy, on two networks of ca-GrQc and ca-HepPh. To compare these three influence maximization algorithms, we selected five different numbers of seed sets, i.e., 10, 20, 30, 40, 50. The MNStaticGreedy algorithm and the StaticGreedy algorithm use static snapshots to simulate propagation. We set the number of snapshots to 100 and take the final average as the propagation range. For the LDAG algorithm, we use 1000 Monte Carlo algorithms to simulate the propagation of information in a multivariate network and use the average as its range of influence.

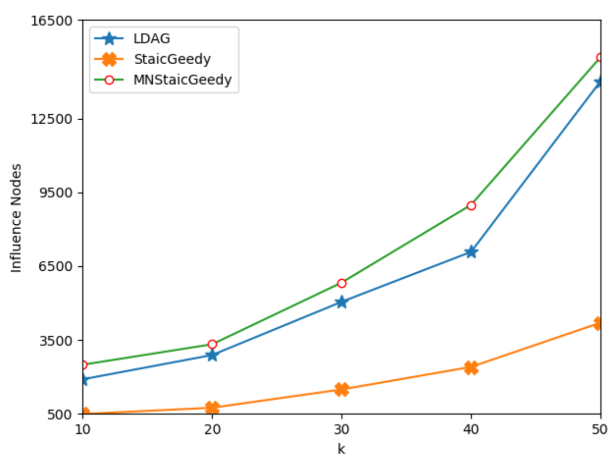

Figure 2: Total influence with different seed sizes

As can be seen from Fig. 2, due to the classic greedy algorithm used by the static greedy algorithm to perform seed iteration, its accuracy is far less than the other two algorithms. MNStaticGreedy algorithm and LDAG algorithm have better performance with the increase of seed nodes. Both two algorithms are based on topological structure. Therefore they can get better accuracy than ordinary greedy algorithms. It is easy to find out from the characteristics of the dataset that our dataset has a large number of edges, which brings a lot of topology information to each node in the networks. Our MNStaticGreedy algorithm uses the calculation of the influence value after the fusion of the two networks, and it produces better seed selection results.

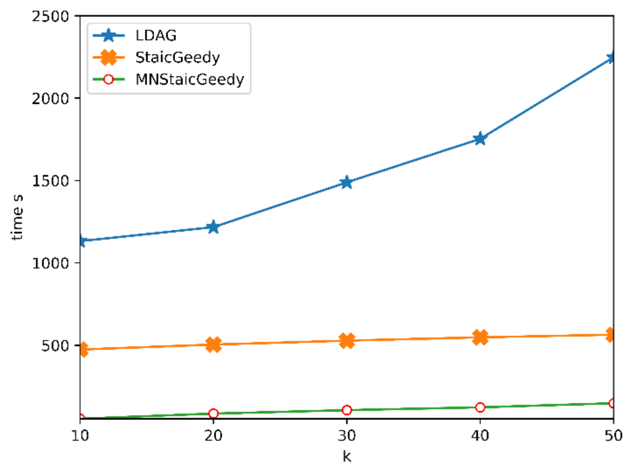

Figure 3: Run time with different seed sizes

As Fig. 3. shows that MNStaticGreedy and StaticGreedy algorithms have lower time complexity. The main reason is in that it is using static snapshots instead of a large number of Monte Carlo simulations. At the same time, the MNStaticGreedy algorithm combines the generation process of the *RR* set with the snapshot generation process and achieves a good time benefit. We can see that the LDAG algorithm has a higher complexity for a large number of linked graphs.

It can be seen from Fig. 4, as the threshold increases, the influence propagation range of the three algorithms show a downward trend with different seeds number. This is mainly because the threshold θ represents the ease with which the nodes in the MNLM model are affected, and the higher the threshold, the more significant the impact required to activate an inactive node. Besides, under the same threshold, as the number of nodes in the initial seed node-set increases, the range of influence also increases. Under the same threshold, when the initial seed node-set is the same, the propagation range of the proposed algorithm is significantly more extensive than the other two algorithms.

At the same time, we can see from Fig. 4. that the number of nodes activated when the threshold changes from 0.05 to 0.1 has a relatively significant change. This also shows that due to the large number of links in the network, the weight values are scattered to a large number of edges. The weight of a large number of edges is relatively small. With the uncertainty of the activated nodes around the random simulation, and the number of edges after the snapshot is reduced, the probability of node activation is small when the threshold is increased. However, from the range, after the threshold is greater than 0.1, including 0.1, 0.15, 0.2, these regions change gently. This shows that at the threshold of 0.1, this node filters out a large number of small-weight-intensive nodes in the graph, leaving a sparse node with a small number of neighbors. These nodes are stable with high thresholds.

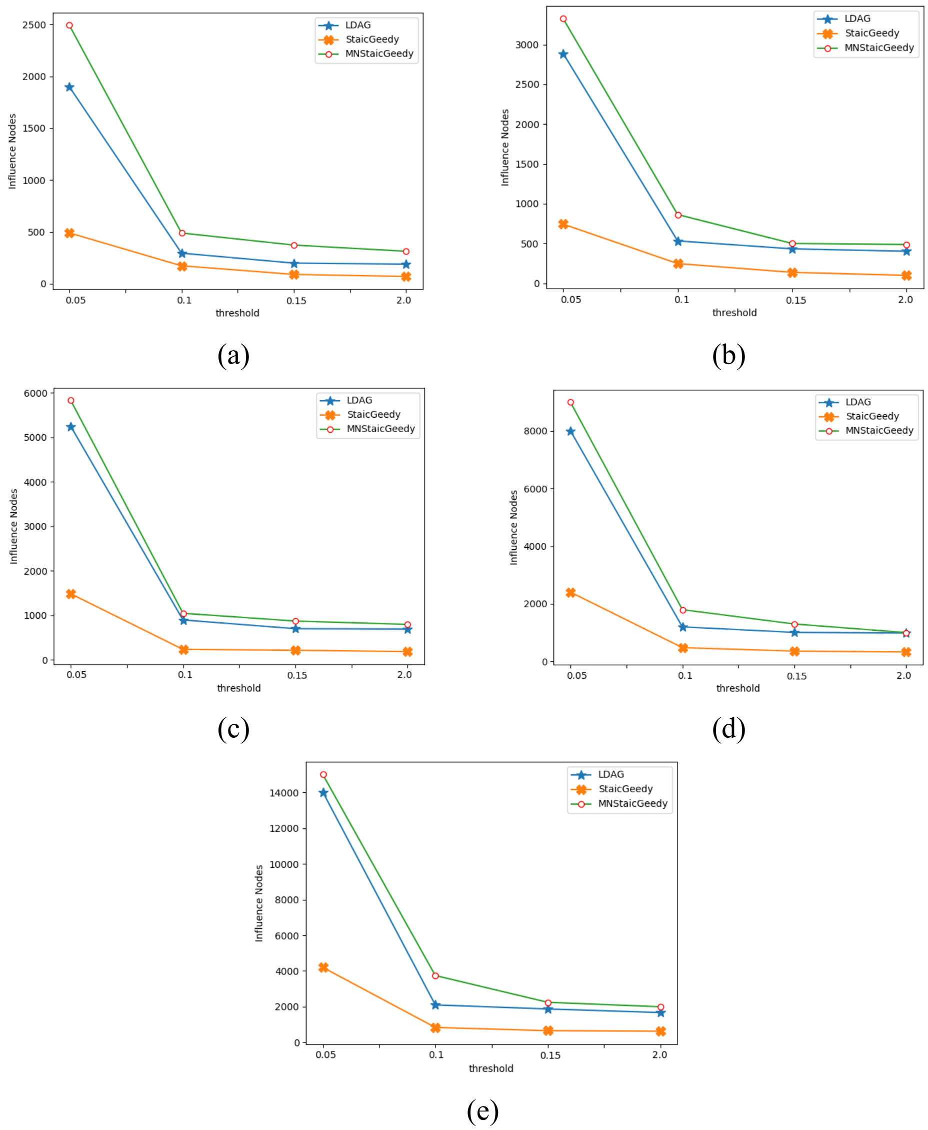

Figure 4: (a) Seed node number k=10; (b) Seed node number k=20; (c) Seed node number k=30; (d) Seed node number k=40; (e) Seed node number k=50;

From Fig. 5, we can see that the run time for the three algorithms decreases as the threshold increases because the more significant the threshold, the smaller the candidate seed node-set. The algorithm proposed in this paper runs faster than the other two algorithms. And the change of the threshold of MNStaticGreedy algorithm has little effect, which proves that our algorithm has better adaptability.

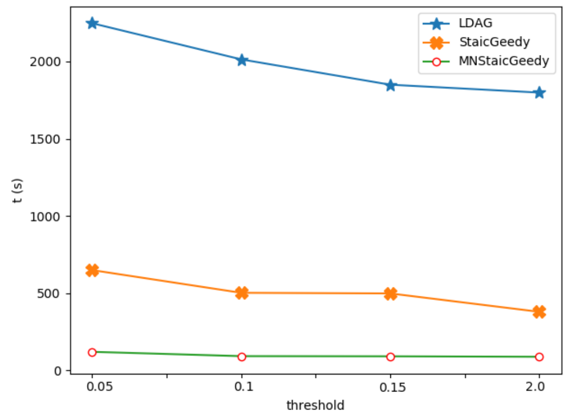

Figure 5: Run time with different threshold

5 Conclusions

In this paper, we explore an Influence Maximization Problem in Multiplex Networks. By considering the relationship between the independent agents in the multiplex network, it is valid and feasible to establish a multiplex network influence diffusion model through the correlation and importance of agents in the relationship layer. At the same time, a novel greedy algorithm MNStaticGreedy is proposed, which is based on a multivariate relationship. Our algorithm has distinct advantages compared with the existing advanced algorithms. In future, we will try to design the dynamic multiplex network influence propagation model based on the influence between nodes.

Funding Statement: This work is supported in part by the National Natural Science Foundation of China under Grant No. 61672022.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bond, R. M.; Fariss, C. J.; Jones, J. J.; Kramer, A. D.; Marlow, C. et al. (2012): A 61-million-person experiment in social influence and political mobilization. *Nature*, vol. 489, no. 7415, pp. 295-298.

Chen, B. L.; Li, F. F.; Zhang, Y. J.; Ma, J. L. (2018): Information filtering in evolving online networks. *Physics Letters A*, vol. 382, no. 5, pp. 265-271.

Chen, B; Hua, Y; Yuan, Y; Jin, Y. (2018): Link prediction on directed networks based on AUC optimization. *IEEE Access*, vol. 6, pp. 28122-28136.

Cheng, S.; Shen, H.; Huang, J.; Zhang, G.; Cheng, X. (2013): Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 509-518.

Fazli, M.; Ghodsi, M.; Habibi, J.; Jalaly, P.; Mirrokni, V. et al. (2014): On non-progressive spread of influence through social networks. *Theoretical Computer Science*, vol. 550, pp. 36-50.

Galhotra, S.; Arora, A.; Virinchi, S.; Roy, S. (2015): Asim: a scalable algorithm for influence maximization under the independent cascade model. *Proceedings of the 24th International Conference on World Wide Web*, pp. 35-36.

Gaye, I.; Mendy, G.; Ouya, S.; Seck, D. (2015): Spanning graph for maximizing the influence spread in Social Networks. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1389-1394.

Hua, Y; Chen, B; Yuan, Y; Zhu, G.; Ma, J. (2019): Influence maximization algorithm based on the mixed importance of nodes. *Computers, Materials & Continua*, vol. 59, no. 2, pp. 517-531.

Kempe, D.; Kleinberg, J.; Tardos, É. (2003): Maximizing the spread of influence through a social network. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137-146.

Kempe, D.; Kleinberg, J.; Tardos, É. (2005): Influential nodes in a diffusion model for social networks. *International Colloquium on Automata, Languages, and Programming*, pp. 1127-1138.

Kuhnle, A.; Alim, M. A.; Li, X.; Zhang, H.; Thai, M. T. (2018): Multiplex influence maximization in online social networks with heterogeneous diffusion models. *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 418-429.

Li, C.; Luo, J.; Huang, J. Z.; Fan, J. (2012): Multi-layer network for influence propagation over microblog. *Pacific-Asia Workshop on Intelligence and Security Informatics*, pp. 60-72.

Mehmood, Y.; Barbieri, N.; Bonchi, F.; Kkonen, A. (2013): CSI: community-level social influence analysis. *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 48-63.

Rahimkhani, K.; Aleahmad, A.; Rahgozar, M.; Moeini, A. (2015): A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications*, vol. 42, no. 3, pp. 1353-1361.

Shultz, T. R.; Rivest, F. (2000): Using knowledge to speed learning: a comparison of knowledge-based cascade-correlation and multi-task learning. *Seventeenth International Conference on Machine Learning*, pp. 871-878.

Tang, Y.; Shi, Y.; Xiao, X. (2015): Influence maximization in near-linear time: a martingale approach. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1539-1554.

Tang, Y.; Xiao, X.; Shi, Y. (2014): Influence maximization: near-optimal time complexity meets practical efficiency. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 75-86.

Tsai, C. W.; Yang, Y. C.; Chiang, M. C. (2015): A genetic NewGreedy algorithm for influence maximization in social network. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2549-2554.

Wu, H.; Liu, W.; Yue, K.; Huang, W.; Yang, K. (2015): Maximizing the spread of competitive influence in a social network oriented to viral marketing. *International Conference on Web-Age Information Management*, pp. 516-519.

Zhang, H.; Nguyen, D. T.; Zhang, H.; Thai, M. T. (2015): Least cost influence maximization across multiple social networks, *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 929-939.