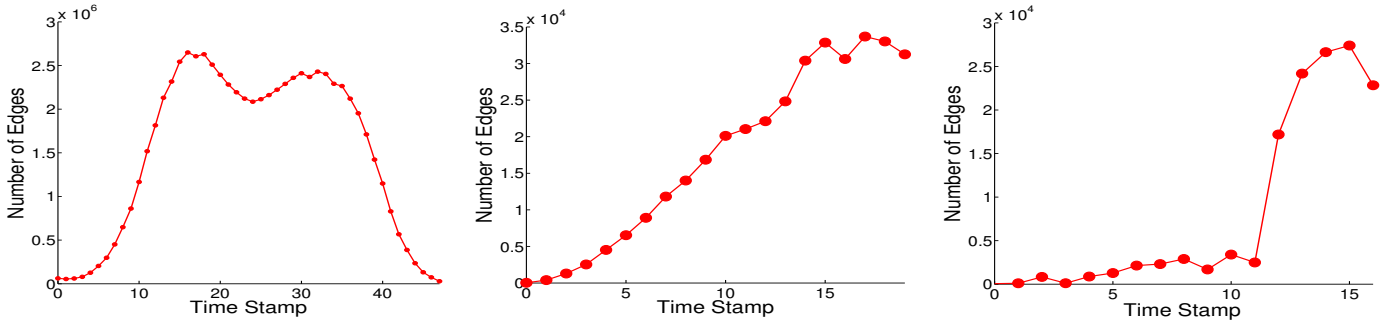


(a) Mobile (b) Hepph (c) Hepth

Fig. 2. Number of edges in snapshot graphs generated from three network datasets.

for each snapshot graph with parameters $\alpha = 0.7$ and $\theta = 1/320$ as reported in [17].

- **Degree**: As a baseline comparison, simply select the nodes with the highest degrees.
- **UBI**: Our UBI algorithm using SP1M [4] for influence estimation with $\gamma = 0.01$. The initial seed set S^0 is generated by **Greedy**. In UBI algorithm, we only calculate the upper bound of marginal gain when calculating the upper bound of node replacement gain.
- **UBI+**: Our UBI algorithm which calculates both the upper bound and the lower bound of the marginal gain when calculating the upper bound of node replacement gain.

We do not include other baseline methods for INT problem since it has already been shown that Greedy always has the best influence coverage while IRIE has slightly worse performance but runs significantly faster than other methods in time [17]. We use the average of 20000 rounds of Monte-Carlo simulations as estimation of the actual influence in order to evaluate the seed sets discovered by the algorithms. Moreover, all the experiments are carried out on a server with 32 cores (2.13G Hz) and 64G memory.

5.2 Experiment Results

5.2.1 Experiment Results of UBI

Influence coverage and running time on real dynamic networks. We first present our main result on comparing our UBI algorithm to other baseline methods on three real-world dynamic networks. For Mobile network, we set the window size to one hour while the time difference is set to two minutes. For both HepPh and HepTh network, we set the window size to three years and the time difference to one month. Moreover, we choose the seed size k as 30.

The results on influence coverage of the selected seed sets for each snapshot graph are shown in Figure 3 and Figure 4. As Greedy is too slow to finish within a reasonable time, we do not include Greedy on Mobile dataset.

We also calculate the average influence spread over all snapshot graphs for all three networks and present the results in Table 3 and Table 4 for better comparison.

For the above results, we can easily find that UBI algorithm results in better influence coverage compared with IRIE averaged over all datasets. As our method has a little loss of accuracy on influence to achieve fast tracking, UBI

TABLE 3
Average influence spread in UA Model

Dataset	Mobile	Hepph	Hepth
Greedy		71.49	65.49
IMM	95.42	71.24	65.43
IRIE	87.99	70.64	64.88
UBI	94.36	71.02	64.36

TABLE 4
Average influence spread in DWA model

Dataset	Mobile	Hepph	Hepth
Greedy		124.35	74.81
IMM	1053.78	124.33	74.48
IRIE	943.69	122.94	74.32
UBI	1033.74	123.79	74.35

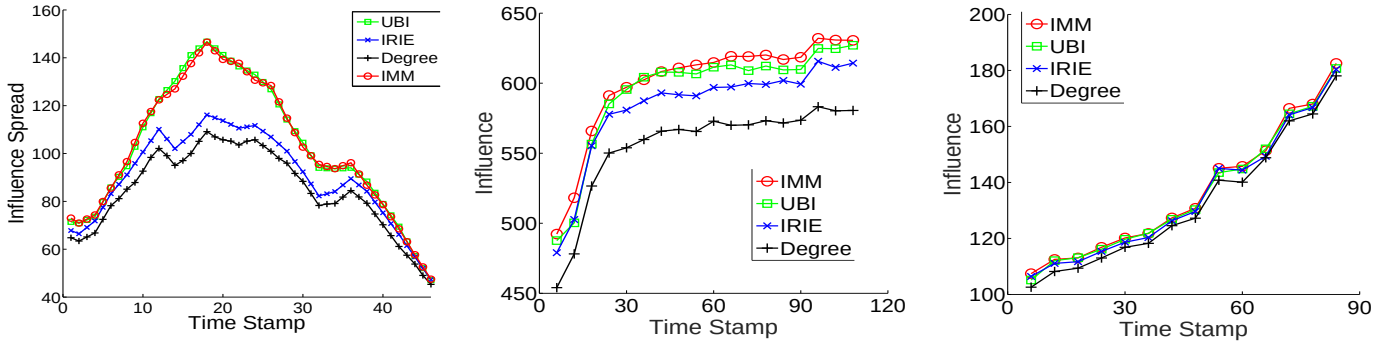
achieves slightly lower influence compared to IMM and Greedy. Moreover, the running time taking average over different snapshot graphs for all three networks results of the above experiments are shown in Table 5 and Table 6.

Reader may ask that if the influential users remain unchanged in most of real datasets, we do not have to track them with an online algorithm. To answer this question, we calculate the average influential users coverage of the result of UBI, which means the total number of users chosen to be the influential user of all time. Results are 151, 119, 143 for Mobile, Hepph and Hepth dataset. Because at every timestamp, the seed set only contains 30 nodes, the result reflects the fact that influential users vary frequently under the scenario of dynamic network.

TABLE 5
Statistics of Running Time for UA Model

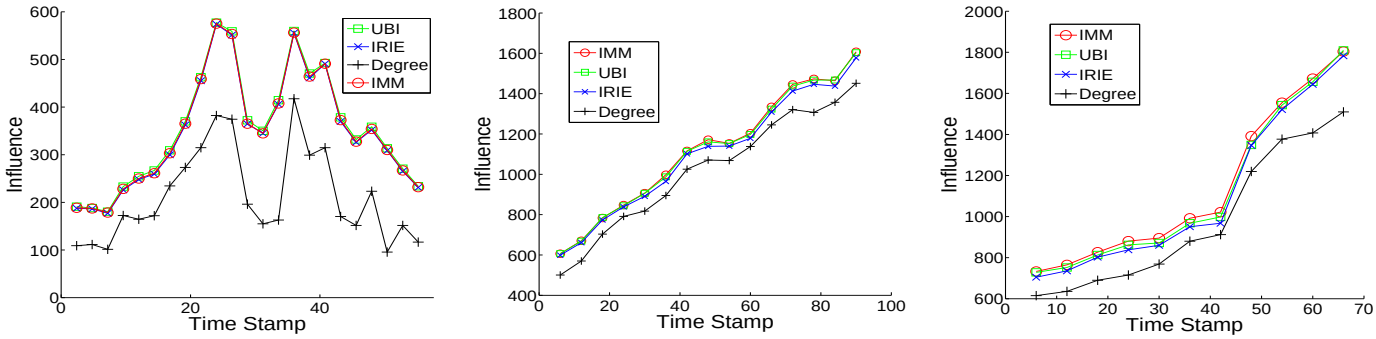
Dataset Running Time	Mobile	Hepph	Hepth
Greedy		12m	8m
IMM	1.8s	38ms	32ms
IRIE	19.1s	50ms	37ms
UBI	1.1s	22ms	17ms

We can easily find that Greedy is extremely slow that it even fails to finish on the largest Mobile network. Though performing well in influence coverage, IRIE performs well in running time on Hepth and Hepph but bad on Mobile with million nodes and edges. IMM performs better than IRIE on Mobile. However, our method, UBI, achieves



(a) Mobile (b) HepPh (c) HepTh

Fig. 3. Influence Tracking Results under UA model with $k = 30$.



(a) Mobile (b) HepPh (c) HepTh

Fig. 4. Influence Tracking Results under DWA model with $k = 30$.

TABLE 6
Statistics of Running Time for DWA Model

Dataset Running Time	Mobile	Hepph	Hepth
Greedy		53m	42m
IMM	2.1s	50ms	46ms
IRIE	30.3s	80ms	60ms
UBI	1.2s	21ms	16ms

consistent lowest running time on all the three networks with comparable influence coverage compared with IRIE, IMM and Greedy algorithm. UBI is about 30 times faster than IRIE and 2 times faster than IMM. Notice that UBI achieves insignificant improvement compared with IRIE and IMM under the last two dataset. This is because they are in relatively small size. At the same time, as the size of networks grows, UBI can scale to networks like Mobile with million nodes and edges as shown in the following experiment.

Memory usage. We measure the memory usage of each algorithm to measure the space complexity and the result is shown in Table 7 and Table 8. As the dataset is also stored in memory, we measure the memory usage when only loading the dataset into the memory, which is marked as "None" in the table. As it can be seen from the result, our UBI algorithm uses a little more memory than Greedy and IRIE and less memory than IMM. UBI uses some additional space to calculate the upper bound and the lower bound to reach a much better influence coverage, but UBI uses only linear additional space so the space complexity is acceptable.

TABLE 7
Statistics of Memory Usage for UA Model

Dataset Memory Usage	Mobile	Hepph	Hepth
None	2376.2MB	27.4MB	24.1MB
Greedy		27.4MB	24.1MB
IMM	2564.7MB	28.9MB	26.0MB
IRIE	2496.5MB	27.8MB	24.5MB
UBI	2536.7MB	29.5MB	26.3MB

TABLE 8
Statistics of Memory Usage for DWA Model

Dataset Memory Usage	Mobile	Hepph	Hepth
None	2375.1MB	27.3MB	24.0MB
Greedy		27.3MB	24.0MB
IMM	2522.7MB	28.5MB	25.2MB
IRIE	2493.7MB	27.9MB	24.6MB
UBI	2537.9MB	29.2MB	26.4MB

Varying K. As the third experiment, we test the algorithm with a large $K = 50$. We have the same setting except K as the first experiment. The results on influence coverage of the selected seed sets for each snapshot graph are shown in Figure 5. As Greedy is too slow to finish within a reasonable time, we do not include Greedy on this experiment. Note that the results under the UA model are similar, which are not included in this paper due to the limited space.

From Figure 5, we have the similar conclusion that UBI has very close influence coverage compared to IMM, which is already proved in [20] that has consistently close influence coverage as Greedy when K is varying. Our method performs consistently while K is different.

Scalability. As the fourth experiment, we test the scalability of our method on networks with different size. We construct a family of snapshot graphs from the Mobile dataset by varying the time window size ω from $2, 4, ..., 512$ minutes with a fixed time difference $\Delta t = 2$ minutes. The average numbers of edges in these graphs vary from 15K to 4M. We run algorithms to track $k = 30$ influential nodes under both the UA and the DWA Model for propagation probability. The running time under DWA model is shown in Figure 6, with normal scale in Figure 6(a) and log-log scale of the same figure in Figure 6 (b). The results under UA model are similar and omitted. We don't plot the running time of Greedy for the measurement of Greedy on Mobile network is unaffordable.

As it is shown in Figure 6, our UBI algorithm is one magnitude faster than the IRIE and achieves about 2x speed up compared to the IMM algorithm. It clearly demonstrates the scalability of our algorithm for INT problem under large-scale dynamic networks.

Random seed vs. S^{t-1} As the fifth experiment, we test the influence of using random seed or S^{t-1} when updating the influence vector. We run the two algorithms to track $k = 30$ influential nodes under both the UA and the DWA Model for propagation probability. The average influence spread under DWA and UA model is shown in Figure 8, while the running time is shown in Figure 9(The figure is in log scale). From the result, it can be clearly seen that using random seed and S^{t-1} to can reach the same influence spread with enough interchange times, but using random seed is about 10 times slower than just using S^{t-1}.

As is shown in Figure 7, using random seed can reach the same influence spread as using the greedy algorithm when the interchange time is 30 or above, while using S^{t-1} only needs about 5-7 times. It clearly demonstrates the efficiency gap between using random seed and using S^{t-1}.

Similarity v.s. Updating time. The efficiency of our UBI algorithm comes from the fact that we utilize the similarity between two consecutive snapshot graphs. To quantitatively characterize the speedup, we conduct an experiment to explore how the similarity of the consecutive graphs correlates with the updating time for UBI. We use the Jaccard similarity to measure the similarity between two consecutive snapshot graphs G^t and G^{t+1}. Formally, we have:

$$\text{Jaccard}(G^t, G^{t+1}) = \frac{|E^t \cap E^{t+1}|}{|E^t + E^{t+1}|}$$

By varying the time difference Δt from $1, 2, 4, ..., 64$ minutes with a fixed one hour window, we construct a series of snapshot graphs with different Jaccard similarity from the Mobile dataset.

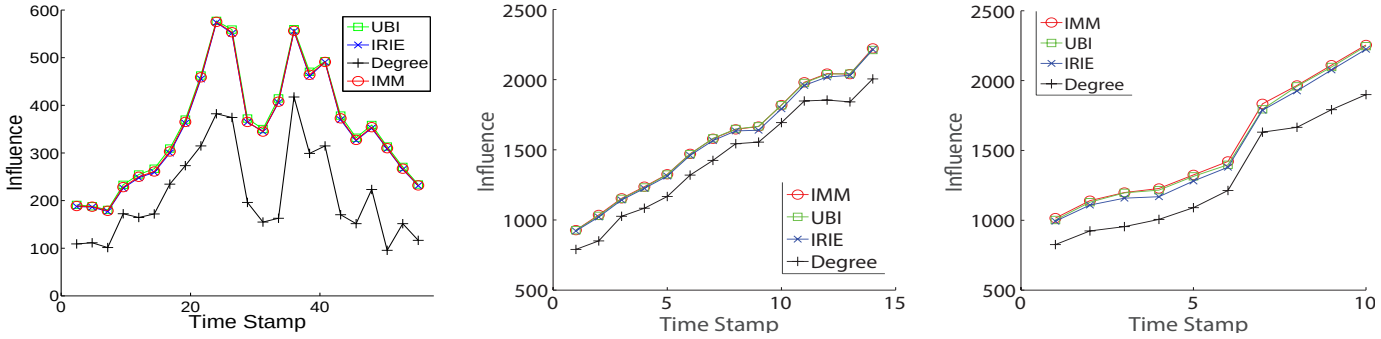
Figure 10 shows how the average updating time of our UBI algorithm is related to the average Jaccard similarity. In line with out intuition, the more similar two consecutive snapshots graphs are, the less time it takes by UBI algorithm to update the seed set. Moreover, even under extremely low Jaccard similarity, where the current snapshot differs greatly from the previous one, our UBI algorithm can still achieve low updating time by utilizing the upper bound on the node replacement gain.

Upper bounds comparison. As we discussed in section 3, our upper bound termed as active nodes' path excluded upper bound (AB), is theoretically tighter than the upper bound proposed in [13], which we call it the naive upper bound (NB). In order to validate our theory, we run empirical experiments to compare our bound AB with the naive upper bound. We first extract a series of snapshot graphs from Mobile datasets by setting both time window and time difference to one hour. We run equivalent number of iterations in computing both AB and NB on the same node set with size $k = 30$ where propagation probabilities are set according to DWA model. The seed set is selected by Greedy algorithm that maximizes the influence under each snapshot. As is shown in Figure 9, our bound is consistently tighter than the naive bound proposed in [13] as suggested by our theory. It should be noticed that the poor performance of NB under DWA model is due to the fact that sometimes NB fails to converge in Mobile network.

5.2.2 *Experiment Results of UBI+*

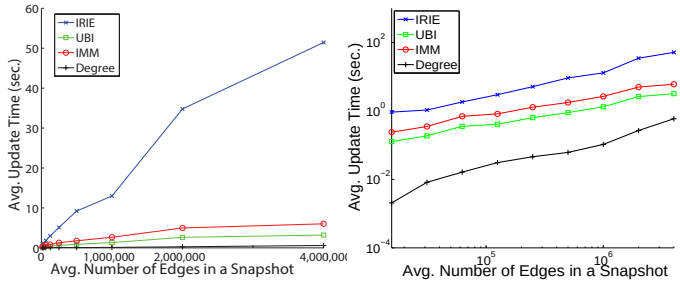
Influence coverage on dynamic networks. We present our result on comparing our improved UBI algorithm, UBI+ to UBI on three real-world dynamic networks. For Mobile network, we set the window size to one hour while the time difference is set to two minutes. For both HepPh and HepTh network, we set the window size to three years and the time difference to one month. Moreover, we choose the seed size k as 30. We calculate the average influence spread over all snapshot graphs for all three networks and present the results in Table 9 and Table 10. For the above results, we can easily find that our UBI+ algorithm achieves a better influence spread than UBI. Notice that UBI+ merely reaches about 2% and 1% better on the Hepph and Hepth dataset, this is because that UBI already performances very close to the influence spread upper bound(which is also the Greedy algorithm's result), so UBI+ only reaches an influence much closer to the theoretically influence bound. However, UBI+ get a 10% improvement in Mobile dataset and this shows that our new algorithm significantly improves the result in large datasets. Similar to the experiment results of UBI, the average influential users coverage of UBI+ is are 154, 119, 143 for Mobile, Hepph and Hepth dataset.

Running time on dynamic networks. As it can be seen from Table 11 and Table 12, though being a little slower because that an additional bound need to be computed, UBI+ performs as well as UBI in running time. Notice that UBI+



(a) Mobile (b) HepPh (c) HepTh

Fig. 5. Influence Tracking Results under DWA model with $k = 50$.



(a) Normal Scale (b) log-log Scale

Fig. 6. Scalability results on Mobile network with different number of edges in each snapshot graph.

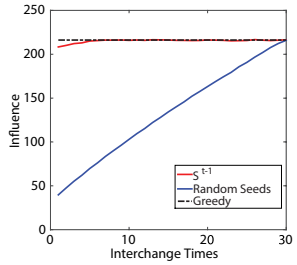
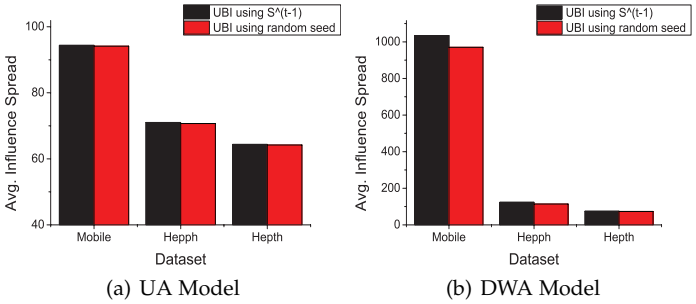
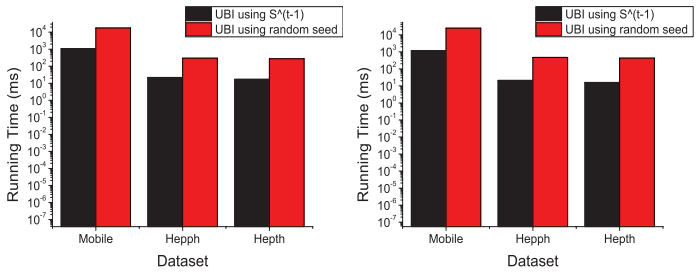


Fig. 7. An example illustrating UBI algorithm interchange from random seed set and S^{t-1}



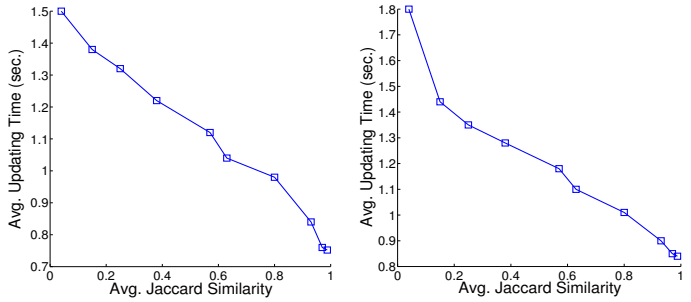
(a) UA Model (b) DWA Model

Fig. 8. Average influence spread of S^{t-1} vs. random seed



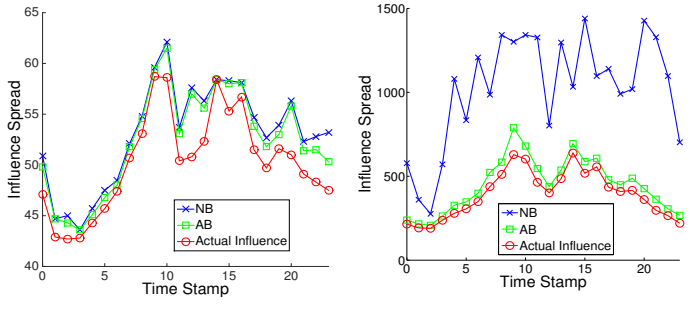
(a) UA Model (b) DWA Model

Fig. 9. Statistics of Running time of S^{t-1} vs. random seed



(a) UA Model (b) DWA Model

Fig. 10. Jaccard similarity vs. Updating time



(a) UA Model (b) DWA Model

Fig. 11. Actual influence spread compared with upper bounds

TABLE 9
Average influence spread in UA Model

Dataset	Mobile	Hepph	Hepth
IMM	95.42	71.24	65.43
UBI	94.36	71.02	64.36
UBI+	95.01	71.15	65.04

TABLE 10
Average influence spread in DWA model

Dataset	Mobile	Hepph	Hepth
IMM	1053.78	124.33	74.48
UBI	1033.74	123.79	74.35
UBI+	1045.15	124.01	74.42

achieves significant improvement in influence coverage in large datasets as the previous experiment shows, so a slight increase in the consumption of time is acceptable. So, in conclusion, UBI+ performs better than UBI in solving INT problem in large datasets. In small datasets, because that UBI already performs well so that UBI+'s improvement is not obvious.

TABLE 11
Statistics of Running Time for UA Model

Dataset Running Time	Mobile	Hepph	Hepth
IMM	1.8s	38ms	32ms
UBI	1.1s	22ms	17ms
UBI+	1.4s	25ms	21ms

TABLE 12
Statistics of Running Time for DWA Model

Dataset Running Time	Mobile	Hepph	Hepth
IMM	2.1s	50ms	46ms
UBI	1.1s	22ms	17ms
UBI+	1.5s	24ms	20ms

5.2.3 Experiment Results of UBI and UBI+ on viral marketing

Benchmark for viral marketing We use the benchmark proposed by Amit Goyal, etc. in [24] to measure our methods' performance. We generate a dataset by applying their benchmark algorithm to the Flixster dataset. The working principle of the benchmark is that the propagation probabilities between users in a social network can be learned from users' actions, such like making comments on movies, traveling to scenic spots, etc.. The Flixster dataset contains links between users and informations about which movie they've made comments on. The links in Flixster are undirected, but the influence probabilities learned is applicable for directed connections. As the result, the learned Flixter dataset with is a directed network, which contains 786.9k nodes and 4.7M edges.

Influence coverage We present our result on comparing our algorithms, UBI+ and UBI on the benchmark for viral marketing. We generate snapshot graphs from the flicker dataset generated by the benchmark mentioned in the previous section.

From Table 13, it can be seen that UBI and UBI+, similar to the results on HepPh, HepTh and mobile, achieves close influence spread to Greedy and IMM. This also supports our previous experiment results that UBI and UBI+ performs well in real dynamic networks.

TABLE 13
Average influence spread on benchmark for viral marketing(Flixster dataset)

Algorithm	Influence
Greedy	534.32
IMM	532.60
IRIE	524.14
UBI	529.16
UBI+	531.87

Running time. As it is shown in Table 14, though being a little slower because that an additional bound need to be computed, UBI+ performs as well as UBI in running time. Experiments result indicates that UBI and UBI+ runs much faster and proves our algorithm's ability to track influential users in a real, dynamic network. These experiments prove that our proposal works better on viral marketing.

TABLE 14
Statistics of Running Time on benchmark for viral marketing(Flixter dataset)

Algorithm	Running Time
Greedy	3h14m
IMM	1.23s
IRIE	13.15s
UBI	0.69s
UBI+	0.94s

6 CONCLUSIONS AND FUTURE WORK

In this paper, we explore a novel problem, namely Influential Node Tracking problem, as an extension of Influence Maximization problem to dynamic networks, which aims at tracking a set of influential nodes dynamically such that the influence spread is maximized at any moment. We propose an efficient algorithm UBI to solve the INT problem based idea of the Interchange Greedy method. We utilize the upper bound on node replacement gain to accelerate the process. Moreover, an efficient method for updating the upper bound is proposed to handle the evolution of the network structure. Extensive experiments on three real social networks show that our method outperforms state-of-the-art baselines in terms of both influence coverage and running time. Then we propose UBI+ algorithm that improves the computation of the upper bound and achieves better influence spread.

As a direct future work, we would like to generalize our UBI algorithm to track influential nodes under the other widely adopted diffusion model, Linear Threshold model under dynamic networks. Moreover, it will be interesting if we can combine our work with [21]. That is to track a series of influential nodes where the diffusion process is also carried out under a dynamic network instead of the static snapshot graph.

ACKNOWLEDGMENTS

This work was supported in part by the National High Technology Research and Development Program of China (2014AA015103), by Beijing Natural Science Foundation (4152023) and by the National Science and Technology Support Plan (2014BAG01B02)

REFERENCES

[1] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social network," in *KDD*, 2009, pp. 199–208.
[2] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD*, 2001, pp. 57–66.
[3] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of inffluence through a social network," in *KDD*, 2003, pp. 137–146.
[4] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *PKDD*, 2006, pp. 259–271.
[5] W.Yu, G.Cong, G.Song, and K.Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *KDD*, 2010, pp. 1039–1048.
[6] W.Chen, C.Wang, and Y.Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD*, 2010, pp. 1029–1038.
[7] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *AAAI*, 2012.
[8] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 88–97.
[9] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Advances in neural information processing systems*, 2013, pp. 3147–3155.
[10] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*, 2005, pp. 177–187.
[11] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *TKDD*, vol. 1, 2007.
[12] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *KDD*, 2008, pp. 462–470.
[13] C. Zhou, P. Zhang, J. Guo, X. Zhu, and L. Guo, "Ublf: An upper bound based approach to discover influential nodes in social networks," in *ICDM*, 2013.
[14] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *KDD*, 2002, pp. 61–70.
[15] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks." in *KDD*, 2007, pp. 420–429.
[16] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social network," in *AAAI*, 2011.
[17] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *ICDM*, 2012, pp. 918–923.
[18] M. G. Rodriguez and B. Schölkopf, "Influence maximization in continuous time diffusion networks," *arXiv preprint arXiv:1205.1682*, 2012.
[19] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 75–86.
[20] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: a martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1539–1554.
[21] C. C. Aggarwal, S. Lin, and S. Y. Philip, "On influential node discovery in dynamic social networks." in *SDM*, 2012, pp. 636–647.
[22] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun, "Influence maximization in dynamic social networks," in *ICDM*, 2013, pp. 1313–1318.
[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
[24] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "A data-based approach to social influence maximization," *Proceedings of the VLDB Endowment*, vol. 5, no. 1, pp. 73–84, 2011.

Guojie Song received the BS and MS degrees from Zhengzhou University, Zhengzhou, China, in 1998 and 2001, respectively, and the PhD degree from Peking University, Beijing, China, in 2004. From 2004 to 2005, he was a research fellow with the Singapore Management University, Singapore. He is currently an associate professor with the School of Electronic Engineering and Computing Science and the vice director in the Research Center of Intelligent Information Processing, Peking University. His research interests include various techniques of data mining, machine learning and their applications in intelligent transportation systems, and social networks.

Yuanhao Li , undergraduate student in the Computer Science Department of Peking University. He is under advising of professor Prof. Guojie Song. His research interests lies in techniques of data mining, machine learning and their applications in social networks, and artificial intelligence.

Xiaodong Chen , Master student in the Computer Science Department of Peking University. He is in Key Laboratory of Machine Perception (Ministry of Education) of Peking University. His research interests include techniques of data mining, machine learning and their applications in intelligent transportation systems, and social networks.

Xinran He , PhD student in the Computer Science Department of University of Southern California. His research interest lies in social network analysis and social media analysis. He is interested in both solving real-world problem in social network with machine learning and data mining techniques and provide theoretic analysis of behaviors on social network with tools such as game theory. Particularly, he is interested in diffusion phenomena on social network, including influence maximization, network inference and competitive diffusion.