Time constraint influence maximization algorithm in the age of big data

Meng Han and Zhuojun Duan

The Department of Computer Science,
Georgia State University,
Atlanta, Georgia, 30303, USA
Email: mhan@cs.gsu.edu
Email: zduan2@student.gsu.edu

Chunyu Ai

Division of Mathematics and Computer Science,
University of South Carolina Upstate,
Spartanburg, South Carolina, 29303, USA
Email: aic@uscupstate.edu

Forrest Wong Lybarger, Yingshu Li* and
Anu G. Bourgeois

The Department of Computer Science,
Georgia State University,
Atlanta, Georgia, 30303, USA
Email: flybarger1@student.gsu.edu
Email: yili@gsu.edu
Email: abourgeois@cs.gsu.edu
*Corresponding author

Abstract: The new generation of social networks contains billions of nodes and edges. Managing and mining this data is a new academic and industrial challenge. Influence maximization is the problem of finding a set of nodes in a social network that result in the highest amount of influence diffusion. Many research works have been developed, which focus exclusively on the efficiency of algorithms, but overlook some features of social network data such as time sensitivity and the practicality in a large scale. Furthermore, the new era of 'big data' is changing dramatically right before our eyes – the increase of big data growth gives all researchers many challenges as well as opportunities. This paper proposes two new models TIC and TLT and considers the time constraint during the influence spreading process in practice. Empirical studies on different synthetic and real large scale social networks demonstrate that our models together with solutions on both Hadoop and Spark platforms are more practical as well as providing a regulatory mechanism for enhancing influence maximization. Not only that but also outperforming most existing alternative algorithms.

Keywords: influence maximization; cloud computing; data mining; data modelling.

Reference to this paper should be made as follows: Han, M., Duan, Z., Ai, C., Lybarger, F.W., Li, Y. and Bourgeois, A.G. (2017) 'Time constraint influence maximization algorithm in the age of big data', *Int. J. Computational Science and Engineering*, Vol. 15, Nos. 3/4, pp.165–175.

Biographical notes: Meng Han received his BS degree in Software Engineer and MS degree in Computer Science from Heilongjiang University, China, and his second MS in Computer Science from Georgia State University. He is currently a PhD candidate in Computer Science Department in Georgia State University, and his research interests include social influence analysis, social privacy, and data mining in big data.

Zhuojun Duan received her MS degree in Department of Computer Science at ShaanXi Normal University and BS degree from Xi`an University of Posts & Telecommunications. She is currently a PhD student in Department of Computer Science at Georgia State University. Her research areas focus on networking, game theory and big data.

Chunyu Ai received her BS and MS degrees in Computer Science from Heilongjiang University, China, and her PhD degree in Computer Science from Georgia State University. She is currently an Assistant Professor in the Division of Math and Computer Science at University of South Carolina Upstate. Her research interests include wireless sensor networks, data management, and social networks.

Forrest Wong Lybarger is a Computer Science major student at Georgia State University. He is currently in his second year and worked under Yingshu Li and Meng Han in the Computer Science Department focusing on studies surrounding the topic of social networks and influence maximization.

Yingshu Li received her BS degree in Computer Science from Beijing Institute of Technology, China, and the MS and PhD degrees in Computer Science from the University of Minnesota at Twin Cities. She is currently an Associate Professor in the Department of Computer Science at Georgia State University. Her research interests include wireless networking, wireless sensor networks, and data management in sensor networks. She is the recipient of the US National Science Foundation (NSF) CAREER Award. She is a member of the ACM and a senior member of the IEEE.

Anu G. Bourgeois is an Associate Professor in the Department of Computer Science at Georgia State University. She received her Master's and PhD in Electrical and Computer Engineering from Louisiana State University in 1997 and 2000, respectively. Her research interests include parallel and distributed computing, wireless networks, security and privacy, and fault tolerant computing. She is a senior member of the IEEE.

1 Introduction

As the internet develops, the increasing popularity of many online social network sites (Facebook, Google+ and Twitter, etc.) enables us to investigate large-scale social networks in a close view. However, we are facing challenges at all levels ranging from infrastructures to programming models for managing and mining large graphs.

Motivated by applications such as personalised recommendations, online advertising and microblog marketing, the study of influence diffusion and maximization attracts more and more attention (Han et al., 2016a, 2016b). Domingos and Richardson (2001) introduced the problem of identifying influential customs in the marketing campaign as a learning problem first. After that, Kempe et al. (2003) studied the influence maximization problem and proposed two primary information diffusion models, namely the independent cascade (*IC*) model and the linear threshold (*IT*) model.

In both of the models, the input is a network with nodes and weighted edges. Each node is either active or inactive. The possibility of one node becoming active increases monotonically with the number of its active neighbours. If one node becomes active, it will never be inactive again. This assumption is coming from real life observation. If we consider the influence diffusion process, let us look at the following example. One customer *Mary* just bought the latest iPhone 6S and posted one status on her Facebook page as 'iPhone 6S is great, get one, you will never regret!'. When her friend *Mike* on Facebook got this message and trusted her, then he could directly purchase one of his own. We naturally consider this process as *Mary* influencing *Mike*, as well as *Mike* is influenced (activated), his status will keep active (he has already purchased the device) and might continue to influence others through social media. Different users could have different levels of susceptibility; this characteristic was modelled as the probability of each edge between different users in the network in our model. In the *IC* model, the beginning moment is denoted as time t_0, nodes with active status perform as 'seeds' in the network. These nodes are considered contagious. A node u has one chance of influencing its inactive neighbour v with probability $p_{u,v}$, which represents the ability of the influence spreading from u to v. If u succeeds in this attempt, node v becomes active at time t_1, otherwise, u will not try to influence node v anymore. This process will continue until no new node becomes active in the network. There is also the same set of seed nodes in the *LT* model as there is in the *IC* model. Whether a node v will be influenced is determined by the sum of the weight of $\sum_1^{|N(v)|} p_{u_i,v}$, such that the sum of all the incoming weights to v is less than or equal to 1. $\{u_1, u_2, \dots, u_i\} \in N(v)$, where $\{u_1, u_2, \dots, u_i\}$ are v's neighbours, and $N(v)$ is v's neighbour set. In each time stamp, node v selects a random threshold θ_v uniformly from [0, 1]. If the sum of weight from all the active neighbours of an inactive node v is more than θ_v, v becomes active at the next time stamp, otherwise, keep inactive. This process also repeats as well as *IC* to the end until no new node becomes active anymore.

Kempe et al. (2003) first formulated influence maximization as a discrete optimisation problem. Considering a social network as a graph $G = (V, E, p)$, where V and E is the set of vertices and edges with size $|V|$ and $|E|$, and p: $E \rightarrow (0, 1]$ is the function assigning each edge $e \in E$ a probability $p(e)$. Choose an influence diffusion model (*IC* or *LT*) and an initial active seed set $S \subseteq V$, the expectation of the active node's number at the end of the process is the expected diffusion spread of S, denoted as $\delta_m(S)$. Then the *influence maximization* problem is defined as follows: A directed social graph $G = (V, E, p)$, find the best seed set S to maximise the $\delta_m(S)$.

However, in both the *IC* and *IT* models, the evolution and influence are diffused in unlimited time. The only termination condition is no new active node appears, but this assumption is not completely supported by the facts in real social networks. Information always depends on timeliness, such as the advertisement of some products is limited to just in a period of time, and the news is only meaningful during a particular period, influence diffusion in general is stopped before the original stop condition in *IC* and *LT*. Based on observations above, we employ a time constraint τ to strengthen the classical models, then propose 'time constraint IC model (*TIC*)' and 'time constraint LT model (*TLT*)', which restrict the output and let the algorithm aim at maximising the influence within the threshold τ time.

On the other hand, as shown in Chen et al. (2010b), the influence diffusion processing itself in *IC* and *LT* are '#P-Hard' problems, and both *IC* and *LT* in their original paper are proved to be 'NP-Hard' problems. These facts told us that developing an exact algorithm to solve this problem is impossible if $NP \neq P$. And the simulation processing of the model itself is also very time consuming. Even though several heuristic algorithms have been developed recently, it is hard to give a theoretical guarantee. As a result, the existing works cannot handle the real large-scale network as efficiently as we illustrated.

Additionally, 'big data' is hinting at a future in which we could compute in a relatively transparent environment (such as a cluster) but local-machine-computation is not the only new buzzword after Web 2.0. As probably the most notable big data frameworks *Hadoop* and *Spark* provide us a potential solutions for large-scale networks to solve the influence maximization problem (Fazio et al., 2013; Lam et al., 2013; Liao et al., 2013). Hadoop is an Apache project providing a distributed file system and a framework for the analysis and transformation of very large datasets using the MapReduce (Jeffrey and Ghemawat, 2008) paradigm. Hadoop is available via the Apache open source license, which provided us an opportunity to develop a big data environment based on Hadoop for our influence maximization problem. As well as *Hadoop*, *Spark* is another open source Apache project. Different from the traditional map reduce, *Spark* approaches data processing mainly in memory instead of a hard drive space.

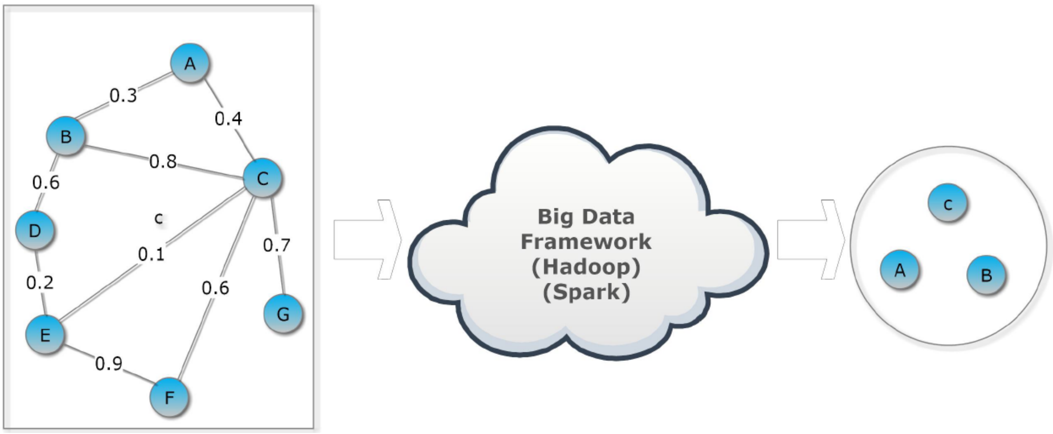
As shown in Figure 1, the input of our problem is a social network with a huge number of nodes and edges between nodes. Each edge has a probability (weight) representing the influence between the nodes pair, by processing the influence maximization on both *Hadoop* and *Spark* environments, the process outputs a seed set S with size k, which can maximise influence when followed by our influence model.

In this paper, we have the contribution below:

- First, we introduce two new influence maximization models with time-constraint properties. We give the formal definition of the new models and analyse their complexity.
- Then, we propose the theoretical proof result of the proper monotonicity and submodularity which give us the possibility of using an efficient greedy algorithm. We will also give the theoretical analysis of the approximation ratio of our algorithm.
- Thirdly, based on the new model and efficient algorithm we proposed, the Hadoop-based cloud computing environment is used to deploy our experimental dataset and algorithm. Considering the specific problem, we also suggest new strategies to optimise the Hadoop-based algorithm.
- Last but not the least, by using both large-scale simulation data and real social networks data, we implement the algorithm in a Hadoop-based cloud environment and evaluate the large-scale data by several efficient distributed strategies.

The rest of this paper is organised as follows. Section 2 reviews the related works. Section 3 presents the preliminaries and problem definition. Section 4 illustrates the algorithm and theoretical analysis. Evaluation results based on real and synthetic datasets are shown in Section 5. Section 6 concludes our paper.

Figure 1 Influence maximization processing in cloud environment (see online version for colours)



2 Related work

In 2003, Kempe et al. initially proposed several influence diffusion models and provided the greedy approximation algorithm. Many researchers studied efficient optimisations and scalable heuristics for the influence maximization problem in different perspectives (Chen et al., 2011; Kimura et al., 2010; Li et al., 2011). Under both the *IC* and *LT* influence diffusion models which were proposed in Kempe et al. (2003). Kempe et al. et al. (2003) showed that finding the maximization problem is NP-Hard which cannot be solved in polynomial time. They also showed that the function $\delta_m(S)$ is *monotone* and *submodular*. Monotonicity means that as the set of activated nodes grows, the chance of a node getting activated would not decrease (Han et al., 2010). Submodularity says that the probability of an active node to activate some other inactive node does not increase if more nodes have already attempted to activate this node. According to these two properties, the influence maximization problem can be approximated by a simple greedy Algorithm 1, and this algorithm has a theoretical guarantee of the approximate ratio $(1 - 1 / e)$. Based on this important result, Leskovec et al. (2007) proposed a 700 times faster greedy algorithm which is based on a 'lazy-forward' optimisation when selecting new seeds. Although Leskovec's algorithm is much better than Algorithm 1, their method also faces serious scalability problems. Chen et al. (2009) improved the efficiency of the greedy algorithm and proposed a new degree-based heuristic to solve this problem. Chen et al. (2010a) also proposed scalable heuristics to estimate the influence diffusion maximization problem under the *IC* model and the *LT* model (Chen et al., 2010b).

Besides the basic *IC* and *LT* models, the topic-aware influence propagation models can also be considered as an important complement which studies the social influence from a topic modelling perspective (Nicola et al., 2012; Jie et al., 2009). After many works appeared to solve the influence maximization problem, one very important previous problem resurfaced, which is how to evaluate the probability between each pair of nodes in the real network (Wu and Shen, 2015). Han et al. (2014) proposed one method to model the uncertainty of the network which considers the weight on each edge with all the 'possible world' and calculate the result by sampling techniques. Kimura et al. (2010) first proposed a learning-based method for extracting influential nodes on a social network. Goyal et al. (2010) also considered this problem and proposed their algorithm to learn probability in the social networks. Then, they presented a data-based approximate algorithm (Goyal et al., 2011) which kept the same approximate ratio guarantee (Long and Hu, 2014).

Li et al. (2015) addressed the problem of finding densely connected subgraphs that satisfy the query conditions considering the influence of a community in a network. However, their method is based on the concept of *k*-core, which is not an influence diffusion expectation model but a network structure model. This kind of model could not provide influence expectation measures which means it does not completely follow the information diffusion process (Saito et al., 2015; Teng et al., 2015; Li et al., 2016). This has been proven and verified in other experiments (Cha et al., 2009; Rossi et al., 2013) to some extent. On the other hand, blocking maximization (He et al., 2012; He et al., 2015) in social networks (Wang et al., 2015; Zhang et al., 2016) can also be considered in several different models. This perspective can support a lot of important applications such as protecting rumours, reducing terrorist information attacks et al., or the influence can also have a totally opposite effect on the social network. We still, however, have not seen any results concerning both the influence spread with a time constraint and an influence decaying process. Furthermore, only single machine implementation struggles to satisfy the request from the big data era. Different from previous related works, we proposed a time constraining model with influence decay to catch the main feature of influence in real life. And we also deploy our models and algorithms to the up-to-date platform for further work reference.

3 Data model and problem definition

In this section, the formal definition of the problem we solved and the corresponding analysis will be proposed. Consider a time threshold $\tau \in \mathbb{Z}_+$, defined as $\delta_\tau : 2^V \to \mathbb{R}_+$ to be the set function such that $\delta_\tau(S)$ with $S \subseteq V$ is the expected number of the activated nodes by the end of the time constraint τ under our model.

Time constraint influence maximization is the problem of finding the seed set S with at most $k = |S|$ seeds such that the expected number of activated nodes by time τ is maximised. Formally, $\delta(S^*) \geq \delta(S)$ for any set S of at most k nodes, find

$$S^* = \arg\max_{S \subseteq V, |S| \leq k} \delta_\tau(S) \tag{1}$$

More specifically, consider the evolution when influence is spreading in the *IC* and *LT* models:

Let $\lambda \geq 1$ be the decay factor in the influence evolution function. A large λ means a slow-decay effect. Then the decay evolution is the function $g(\lambda)$ equal to $\left(\frac{1}{2}\right)^{\frac{t-t'}{\lambda}}$. When the value of this function is below the minimum threshold $\dot{p}$, we stop to calculate the probability of that edge. In practice, this decay function could also be a linear, logarithm or even an exponential function which simulates the decay of relationships in the social network.

3.1 Time constraint IC model

Based on the classical *IC* model introduced in Section 1, each node v will be influenced by all $\{u_1, u_2, \ldots, u_i\}$ from v's neighbour set $N(v)$ according to the weight of $p_{u_i,v}$ on each neighbour u_i, such that the weight of any of the incoming weights from u_i to v is less or equal to 1. In each iteration, the node u_i selects a random threshold $\theta(u_i)$ uniformly from [0, 1]. If the weight $p_{u_i,v}$ between u_i to v is more than the threshold $\theta(u_i)$, then v becomes active at the next time stamp. Otherwise, u_i will not try to influence v anymore. The objective is to maximise the influence function $\delta(S^*)$ which is the expected number of influenced nodes at the end of the propagation. Different from the basic *IC* model, the probability $p_{u_i,v}$ on each edge in our model will decay per the decay function as the influence path from which the original seed set S continues, and the process will terminate by the threshold τ.

3.2 Time constraint LT model

Based on *LT* model, each node v will be influenced by all $\{u_1, u_2, \dots, u_i\}$ from v's neighbour set $N(v)$ according to the sum of the weight of $\sum_1^{|N(v)|} p_{u_i,v}$, such that the sum of all the incoming weights to v is less than or equal to 1. The node v chooses a random threshold θ_v uniformly from [0, 1] at each time stamp. If the sum of weight from all the active neighbours of an inactive node v is more than the threshold, then v becomes active at the next time stamp. Similarly as the *TIC* model, the process functions until one of the basic conditions or the threshold τ is satisfied.

4 Algorithm and theoretical result

According to the previous related work of Kempe et al. (2003), the following algorithm can solve the problem with an approximate ratio of $(1 - 1 / e)$.

Algorithm 1 Greedy algorithm (GA)

```
input: $T, k, \delta_m$
output: seed set $S$
$S \leftarrow \emptyset$;
while $|S| < k$ do
    $u \leftarrow \text{argmax}_{\omega \in V \setminus S}(\delta_m(S \cup \{\omega\}) - \delta_m(S)$;
    $S \leftarrow S \cup \{u\}$;
```

Different from *IC* and *LT*, we add the time constrains and the probability decay functions to make the problem more suitable for social networks.

Next, we will give the theoretical analysis of the problem property and the hardness of our problem. According to the approximate algorithm theory, if one algorithm problem can satisfy monotonicity and submodularity, then we can directly get an efficient approximate algorithm with the approximate ratio as $(1 - 1 / e)$. We first provide the proof of our problem of the property monotonicity and submodularity.

Theorem 1: The influence maximization model *TIC* a monotonic and submodular model.

Proof: For the monotonicity, since the influence function of *TIC* is an increasing function, the conclusion is obvious.

For the submodularity, let X denote one sample point in this space, $|\delta_X(A)|$ is the total number of nodes activated by the process when A is the initial set. $R(v, X)$ denotes the set of all nodes that can be reached from v on a path consisting entirely of live edges. $\delta_X(A)$ is equal to the union $\cup_{v \in A} R(v, X)$.

1 $\delta_X(S \cup \{v\}) - \delta_X(S)$ is the number of elements in $R(v, X)$ that are not already in the $\cup_{v \in S} R(v, X)$.

2 $\delta_X(T \cup \{v\}) - \delta_X(T)$ is the number of elements in $R(v, X)$ that are not already in the $\cup_{v \in S} R(v, X)$.

$S \subseteq T \Rightarrow (1) > (2)$; $\delta(A) = \sum_{outcomesX} Prob[X] \delta_X(A)$. Since the basic *IC* follows the process as we mentioned, limited by the time threshold, the process will be terminated earlier, but it will output a similar result since it can be considered to be a specific case of the basic *IC* model, end.

Theorem 2: The influence maximization model *TLT* is monotone and submodular.

Proof: For the monotonicity, since the influence function of *TLT* is an increasing function, the conclusion is obvious.

For the submodularity, let v have an influence weight $b_{v,w} \geq 0$ and $\sum_w b_{v,w} \leq 1$. Suppose v picks at most one of its incoming edges at random, selecting the edge from w with $b_{v,w}$ and no edge with $1 \sum_w b_{v,w}$. Similarly, as in the Theorem 1, selected as live and unselected as blocked. We prove the following two distributions are the same:

1 The distribution over active sets obtained by running the *LT* process to completion starting from A.

2 The distribution over sets reachable from A via live-edge paths, under the random selection of live edges defined above.

For directed and acyclic graphs: If the *TLT* subset S_i of v_is neighbours is active, the probability is $\sum_{v \in S_i} b_{v_i,w}$. If graph G is not acyclic, A_t is the set of active nodes at the end of iteration t. If A_0 is the initially set, the probability node v becomes active in iterations of $t + 1$ equal to the chance that the influence weight in $A_t \setminus A_{t-1}$ has to push it over its threshold $\dfrac{\sum_{v \in A_t \setminus A_{t-1}} b_{v,u}}{1 - \sum_{u \in A_{t-1}} b_{v,u}}$.

If graph G is not acyclic: Start with set A, for each node v with at least one edge of the set A, determine whether v's live edge comes from A. If yes, v is reachable, if no then keep the source of v's live edge unknown. Expose a new set of reachable nodes A_1 in the first stage, then produce sets A_2, A_3, …. Similarly to the proof of *TIC*, we can easily get that the *TIT* is a specific case of the *LT* but with the time constraint and the decay process, end.

Algorithm 2 Time constraint greedy algorithm (TGA)

```
input: $\tau$, $k$, $g$, $\delta_m$
output: seed set $S$
$S \leftarrow \emptyset$;
while $|S| < k$ do
    $u \leftarrow \text{argmax}_{\omega \in V \setminus S}(\delta_m(S \cup \{\omega\}) - \delta_m(S))$;
    $S \leftarrow S \cup \{u\}$
    Recalculate the probability on of new edge by function $g$;
    if the time exceed the threshold $\tau$ then
        Break;
```

According to Theorem 2, we have shown a simple greedy algorithm can be used to efficiently solve the new problem of *TIC*, and *TLT*.

We can easily determine that our new algorithm has the same time complexity as *IC* and *LT*. Since we consider the feature of time constrains and probability decay, our models simulate the real phenomenon more accurately and the algorithms based on our models are also much more practical.

Besides, we also provide an improved version degree discount algorithm (Chen et al., 2009) to solve our influence maximization models. Different from traditional degree discounts, our algorithm as shown in Algorithm 3 considers the time constraint and the influence decay process, together with several optimisations in implementation.

Algorithm 3 Time constraint degree discount algorithm (TDDA)

```
input: $\tau$, $k$, $g$, $\delta_m$
output: seed set $S$
$S \leftarrow \emptyset$;
for each vertex $v$ do
    calculate v's expected degree $d_v$;
    $dd_v = d_v$;
    $t_v = 0$;
    Recalculate the probability on of new edge by function $g$;
    Update $S$;
    if the time exceed the threshold $\tau$ then
        Break;
```

5 Experiment evaluation

We used both real world networks and synthetic networks to demonstrate the effectiveness and the efficiency of our models and algorithms. We also evaluated our algorithms' quality and efficiency. All experiments were performed on a cluster server with Hadoop environment.

5.1 Environment setup

1 Single node setup

We implemented the algorithms in Python 2.7.2 based on the latest version of Snap.py[1], and all experiments are performed on a PC running Windows 10 with Intel(R) Core(TM) i3-2120 CPU 3.30 GHz and 12 GB memory.

2 Cluster setup

We set up both Hadoop and Spark environment on the Apache Hadoop NextGen MapReduce (YARN). We installed the latest version Hadoop 2.7.0 which released on 21 April 2015. We use 7 of the sever nodes to finish the Map-Reduce process. Apache Spark's GraphX version 1.5.2 in Python and Apache Hadoop's Giraph version 2.0 in Java are implemented to deploy in the cluster. (Detail of nodes in cluster could be found in Table 1).

5.2 Synthetic social networks

- *Small-world graphs*: the small-world network model is a classical model following the small-world features according to 'small-world' (Watts and Strogatz, 1998). This model is referred to as *Syn-SmallWord.*
- *Kronecker graphs*: this model proposed in Leskovec et al. (2005) generates a network in a natural way. The networks grow from five initial nodes and then Kronecker's idea is repeatedly applied to expand the network. This model is referred to as *Syn-Kronecker.*

Based on the initial networks generated from the above models, we dynamically change each network based on the idea proposed in Barabási and Albert (1999). Since we have multiple synthetic networks in the experiments, the average summary of ten networks' statistic features has been used instead. As shown in Table 2, we generate two scales [small (S) and large (L)] of networks for both models with a time stamp length of 50 and 100, respectively.

Table 1 Dataset 1 in our experiment

Node name	*CPU*	*Memory*	*Function*
gpu10	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Master and SecondaryNameNode
gpu05	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode
gpu06	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode
gpu07	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode
gpu08	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode
gpu09	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode
gpu11	Dual Intel Xeon E5-2650	64 GB DDR3 (1,866 MHz)	Worker and DataNode

Table 2 Details of synthetic data

Network model	*Nodes*	*Edges*
Syn-SmallWord(S)	1,000	7,356
Syn-SmallWord(L)	50,000	638,274
Syn-Kronecker(S)	1,000	19,215
Syn-Kronecker(L)	50,000	693,473

5.3 *Real social networks data experiment*

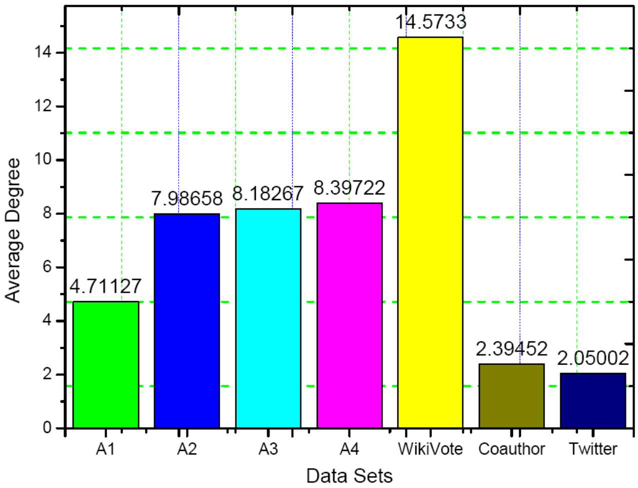
Different kinds of real datasets are used in our experiment, the first group of datasets shown in Table 3 come from Stanford Large Network Dataset Collection (SNAP)[2] which is an open network dataset collection built by Stanford university for researchers. The network statistics were evaluated by a number of nodes(edges), a number of nodes (edges) in largest weakly connected component (WCC) and strongly connected component (SCC), and diameter (longest shortest path). Table 3 is based on the *customers who bought this item also bought* feature of the Amazon website. Four networks come from March to May in 2003. Each network contains a directed edge from *i* to *j* if a product *i* is frequently co-purchased with product *j* (Leskovec et al., 2007). The co-purchased network is an undirected graph and we generate the two directions on each edge.

Besides the Amazon product co-purchasing networks in Table 3, we also evaluate our algorithm in the real datasets below:

- *WikiVote* is a dataset obtained from Leskovec et al. (2010) which collected the vote history data of Wikipedia[3]. The network includes 7,115 vertices and 103,689 edges which contains the voting data of Wikipedia from the inception till January 2008. If user *i* voted for user *j* in the administrator election, there will be a directed edge from *i* to *j*.
- *Coauthor* is a dataset obtained from Tang et al. (2008), which collected the authors' network by ArnetMiner[4]. We used the subset which includes 53,442 vertices and 127,968 edges. Since the co-author relationship is symmetrical, when the author *i* has a relationship with author *j*, there will be one direct edge from *i* to *j* and another edge from *i* to *j*.
- *Twitter* is a dataset obtained from Hopcroft et al. (2011) and Lou et al. (2013) which collected the information from Twitter[5]. We use the subset that includes 92,180 vertices and 188,971 edges which represents the follower relationship.

Same as the experiment in Chi et al. (2012), we set the positive probability as 1 / deg(*v*) for an edge (*u*, *v*), where deg(*v*) is the degree of *v*. We let the negative probability on each edge be 10, 30, 50 and 80% of the positive probability.

Figure 2 Average degree of each real social network (see online version for colours)



In Figure 2, we give the average degree of each real network; the WikiVote has the highest average degree, which means the WikiVote potentially is the densest network among others.

5.4 *Simulation data experiment*

We present the experimental result of the synthetic data on a single node first. By default, we test the two greedy algorithms GA and TGA corresponding to model *IC* and *TIC*.

As show in Figure 3, we test the change by the size of network under Kronecker model (the small world model presents a similar result), when the network size increase, the different between our *TIC* model and *IC* model became larger. This trend is resulting from *TIC* model consider the time constraint and influence decay during the propagation. Also, when the network size was increased, the propagation scope increased, but limited by the time and influence damping, the number of influenced nodes was also decreased.

Table 3 Amazon dataset details

Data	*Nodes*	*Edges*	*Nodes in LWCC*	*Edges in LWCC*	*Nodes in LSCC*	*Edges in LSCC*
Amazon0302(A1)	262,111	1,234,877	262,111 (1.000)	1,234,877 (1.000)	241,761 (0.922)	1,131,217 (0.916)
Amazon0312(A2)	400,727	3,200,440	400,727 (1.000)	3,200,440 (1.000)	380,167 (0.949)	3,069,889 (0.959)
Amazon0505(A3)	410,236	3,356,824	410,236 (1.000)	3,356,824 (1.000)	390,304 (0.951)	3,255,816 (0.970)
Amazon0601(A4)	403,394	3,387,388	403,364 (1.000)	3,387,224 (1.000)	395,234 (0.980)	3,301,092 (0.975)

Figure 3 Comparison between *IC* and *TIC* with network size increase (see online version for colours)

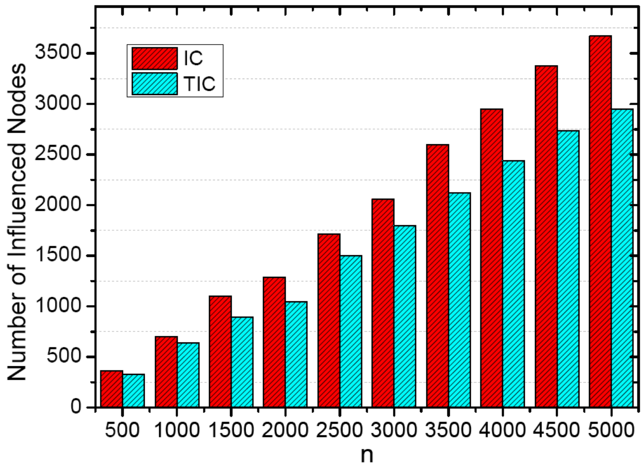


Figure 4 Comparison between *IC* and *TIC* in Amazon co-purchase data (see online version for colours)

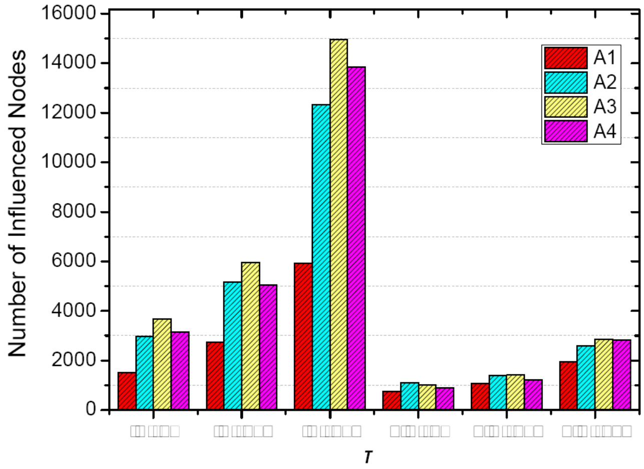


Figure 5 Comparison between *IC* and *TIC* in other real social networks (see online version for colours)

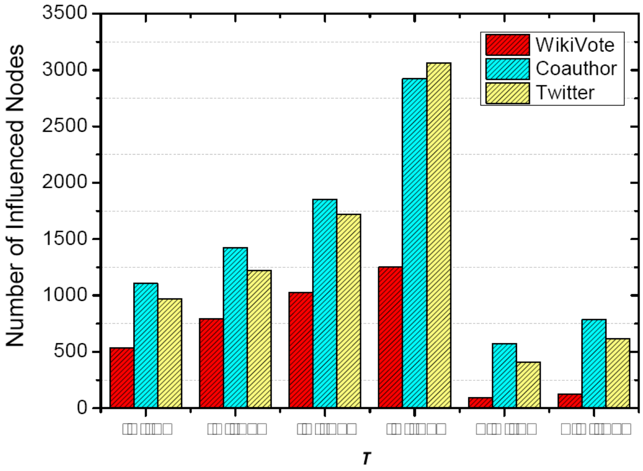


Figure 6 Influenced nodes comparison on small size synthetic data (see online version for colours)

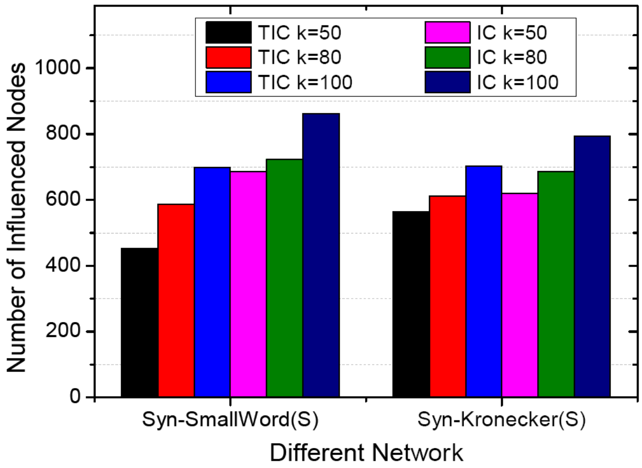


Figure 7 Influenced nodes comparison on large size synthetic data (see online version for colours)

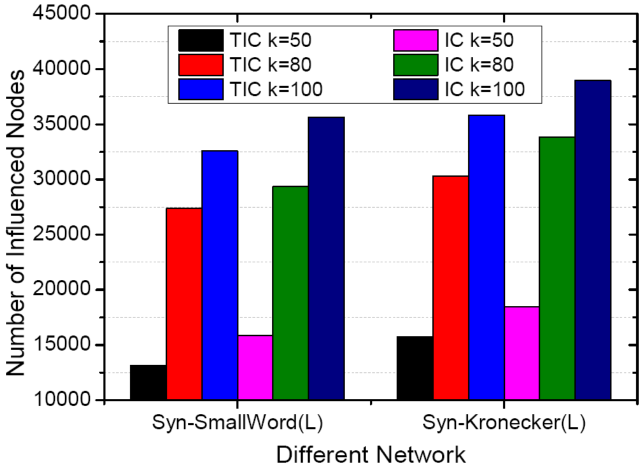


Figure 8 Running time comparison between *IC* and *TIC* (see online version for colours)

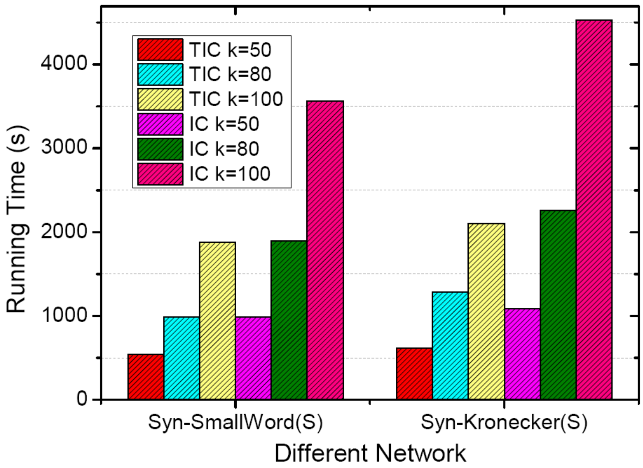


Figure 9 Running time comparison between *IC* and *TIC* on large size synthetic data (see online version for colours)

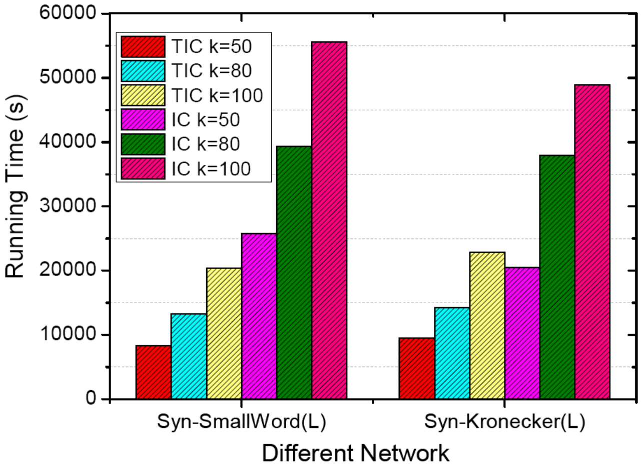


Figure 10 Running time comparison between single machine and cluster under small-world (see online version for colours)

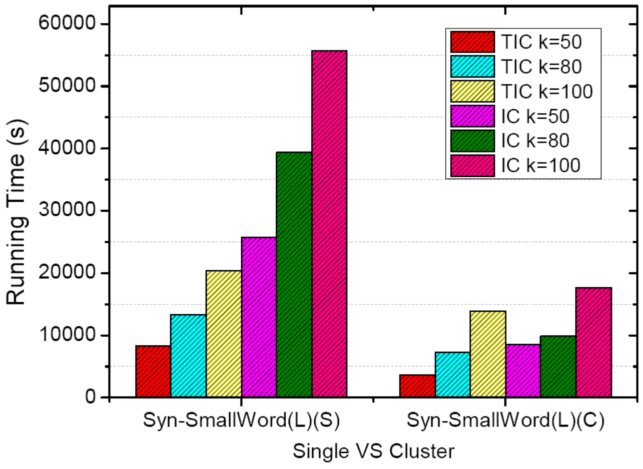
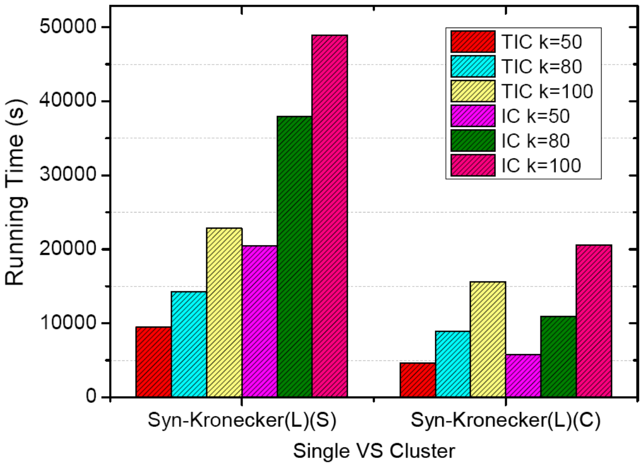


Figure 11 Running time comparison between single machine and cluster under Kronecker (see online version for colours)



As shown in Figure 6 and Figure 7, we can get that the algorithm of *IC* can influence more nodes than our new model *TIC* because our new model considers the time constraint and the probability decay process. Even though, since the *IC* and *LT* are the ideal assumed models, our models do describe the influence process in a more practical way than the classical ones.

Since the result of *IC* and *LT* present a very similar result, limited by space, we mainly just show *IC*'s result.

The first advantage of our model is that it captures the real life attributes of influence and models them. The second advantage is our model could end the influence process limited by time. The time constraint and influence decay could end the iteration in advance, which is also speeding up the algorithm running. As shown in Figure 8 and Figure 9, the other model, *TIC*, could apparently save more time to find out the most influential nodes in the two scales networks.

All experiments above show the features and advantages of our model. But if one considers the running time for either networks or either model, although our model *TIC* could outperform the classical *IC* with the greedy algorithm, the running time is hard to be satisfied and if the size of data increases further, it is very hard to finish in a reasonable time.

Next, we implement model *IC* and *TIC* in the cluster (Hadoop) under both small-world and Kronecker with greedy algorithms. As shown in Figure 10 and Figure 11, apparently that cluster could achieve much better performance. Later in this section, we will show more results and comparisons between Hadoop and Spark.

Figure 12 is the comparison of number of influenced nodes between the two algorithms in single machine and cluster under Kronecker. Actually, the only difference between one algorithm on both single machine and cluster is the performance. Implementation of the same algorithm with different platforms does not affect the quality of the result. Thus, as shown in Figure 12, there is no difference between single machine and cluster in terms of number of influence nodes.

Figure 12 Number of influenced nodes comparison between single machine and cluster (see online version for colours)

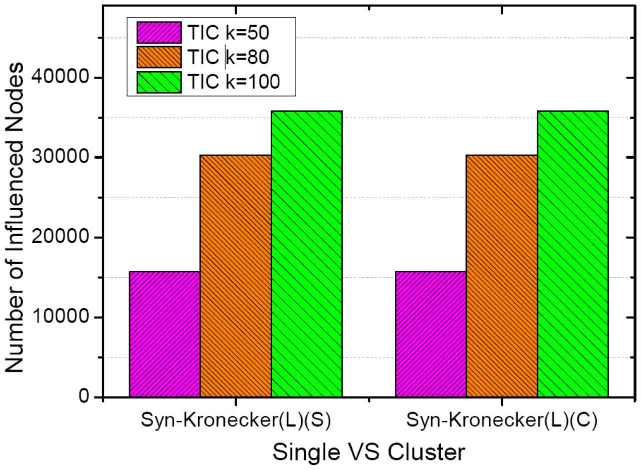
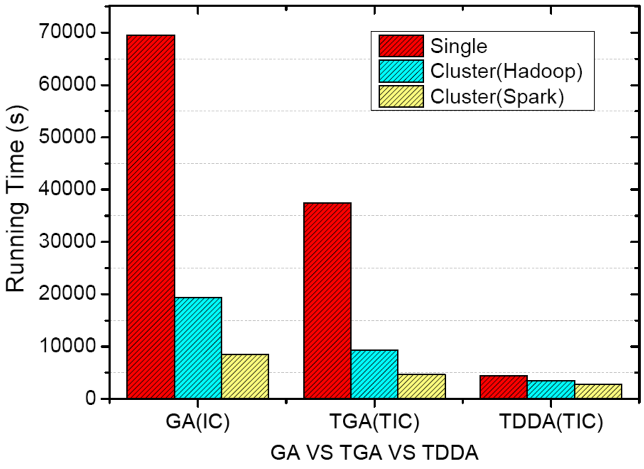


Figure 13 Comparison among of algorithm GA(*IC*), TGA(*TIC*), and TDDA(*TIC*) (see online version for colours)



5.5 Real world data experiment

Figure 4 and Figure 5 show the two real social networks match the analysis of our theory. But different networks have different distributions and topologies, in Figure 5, different networks show different changing trends.

To compare the performance of different big data frameworks, we implement algorithm GA, TGA, and TDDA to our largest real world network Amazon(A3) with 410,236 nodes and 3356,824 edges on both Hadoop and Spark, we set the size of seed set *k* as 50. As shown in Figure 13, the performance on Spark is much better than Hadoop. This result is based on the different mechanism of Hadoop and Spark. Spark basically is a memory-based framework for massive computation. Even though, performance of Hadoop is still much better than single machine. The reason our algorithm TDDA in Figure 13 performs better than the other two greedy algorithms is because TDDA is a heuristic algorithm with quite lower computation complexity.

Overall, the experiment shows that our model and algorithm match the theoretical result we proposed and we have tested the performance of different big data platforms on both synthetic data and real world data.

6 Conclusions

This paper presents two new models *TIC* and *TLT* which extend the practicality of the classical *IC* and *LT* models for influence maximization. The theoretical analysis shows that the two new models we propose both follow monotonicity and submodularity which can help us to design simple greedy algorithms with a guaranteed approximate ratio of (1 – 1 / *e*). Both the simulation and real social network data implementations on a Hadoop-based environment show that our new algorithm can solve the problem efficiently and effectively.

References

Barabási, A-L. and Albert, R. (1999) 'Emergence of scaling in random networks', *Science*, Vol. 286, No. 5439, pp.509–512.

Cha, M., Mislove, A. and Gummadi, K.P. (2009) 'A measurement-driven analysis of information propagation in the flickr social network', *Proceedings of the 18th International Conference on World Wide Web*, ACM.

Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincon, D. et al. (2011) 'Influence maximization in social networks when negative opinions may emerge and propagate', *Proceedings of the 2011 SIAM International Conference on Data Mining (SDM '2011)*, Mesa, Arizona, USA, 28 April 2011, pp.379–390.

Chen, W., Wang, C. and Wang, Y. (2010a) 'Scalable influence maximization for prevalent viral marketing in large-scale social networks', *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.1029–1038.

Chen, W., Yuan, Y. and Zhang, L. (2010b) 'Scalable influence maximization in social networks under the linear threshold model', *Proceedings of the 2010 IEEE International Conference on Data Mining*, IEEE, pp.88–97.

Chen, W., Wang, Y. and Yang, S. (2009) 'Efficient influence maximization in social networks', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, pp.199–208.

Chi, W., Chen, W. and Wang, Y. (2012) 'Scalable influence maximization for independent cascade model in large-scale social networks', *Data Mining and Knowledge Discovery*, Vol. 25, No. 3, pp.545–576.

Domingos, P. and Richardson, M. (2001) 'Mining the network value of customers', *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, pp.57–66.

Fazio, M., Puliafito, A. and Distefano, S. (2013) 'Managing volunteer resources in the cloud', *Int. J. of Computational Science and Engineering*, Vol. 8, No. 3, pp.227–239.

Goyal, A., Bonchi, F. and Lakshmanan, L.V. (2011) 'A data-based approach to social influence maximization', *Proceedings of the VLDB Endowment*, Vol. 5, pp.73–84.

Goyal, A., Bonchi, F. and Lakshmanan, L.V.S. (2010) 'Learning influence probabilities in social networks', *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp.241–250.

Han, M., Yan, M. et al. (2014) 'Neighborhood-based uncertainty generation in social networks', *Journal of Combinatorial Optimization*, Vol. 28, No. 3, pp.561–576.

Han, M., Yan, M., Cai, Z. and Li, Y. (2016a) 'An exploration of broader influence maximization in timeliness networks with opportunistic selection', *Journal of Network and Computer Applications*, March 2016, Vol. 63, pp.39–49.

Han, M., Yan, M., Cai, Z., Li, Y., Cai, X. and Yu, J. (2016b) 'Influence maximization by probing partial communities in dynamic online social networks', *Transactions on Emerging Telecommunications Technologies*, DOI: 10.1002/ett.3054, ISSN: 2161-3915 [online] http://onlinelibrary.wiley.com/doi/10.1002/ett.3054/citedby.

Han, M., Zhang, W. and Li, J. (2010) 'RAKING: an efficient K-maximal frequent pattern mining algorithm on uncertain graph database', *Jisuanji Xuebao (Chinese Journal of Computers)*, Vol. 33, No. 8, pp.1387–1395.

He, X., Song, G., Chen, W. and Jiang, Q. (2012) *Influence Blocking Maximization in Social Networks under the Competitive Linear Threshold Model*, Technical Report.

He, Z., Cai, Z. and Wang, X. (2015) 'Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks', *The 35th IEEE International Conference on Distributed Computing Systems*, Columbus, Ohio, USA.

Hopcroft, J., Lou, T. and Tang, J. (2011) 'Who will follow you back? Reciprocal relationship prediction', in *CIKM11*, pp.1137–1146.

Jeffrey, D. and Ghemawat, S. (2008) 'MapReduce: simplified data processing on large clusters', *Communications of the ACM*, Vol. 51, No. 1, pp.107–113.

Jie, T., Sun, J., Wang, C. and Yang, Z. (2009) 'Social influence analysis in large-scale networks', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.807–816.

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

Kimura, M., Saito, K., Nakano, R. and Motoda, H. (2010) 'Extracting influential nodes on a social network for information diffusion', *Data Mining and Knowledge Discovery*, Vol. 20, No. 1, pp.70–97.

Lam, Y., Tsoi, K. and Luk, W. (2013) 'Parallel neighbourhood search on many-core platforms', *Int. J. of Computational Science and Engineering*, Vol. 8, No. 3, pp.281–293.

Leskovec, J. et al. (2005) 'Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication', in *Knowledge Discovery in Databases: PKDD 2005*, Springer, pp.133–145.

Leskovec, J., Adamic, L. and Adamic, B. (2007) 'The dynamics of viral marketing', *ACM Transactions on the Web (ACM TWEB)*, Vol. 1, No. 1, p.5.

Leskovec, J., Huttenlocher, D. and Kleinberg, J. (2010) 'Predicting positive and negative links in online social networks', *WWW 2010*.

Li, H., Xu, X., Lai, L. and Shen, Y. (2016) 'Online commercial intention detection framework based on web pages', *Int. J. of Computational Science and Engineering*, Vol. 12, Nos. 2/3, pp.176–185.

Li, R. et al. (2015) 'Influential community search in large networks', *Proceedings of the VLDB Endowment*, Vol. 8, No. 5, pp.509–520.

Li, Y., Chen, W., Wang, Y. and Zhang, Z. (2011) *Influence Diffusion Dynamics and Influence Maximization in Social Networks with Friend and Foe Relationships*, arXiv preprint, arXiv: 1111.4729.

Liao, C., Shih, J. and Chang, R. (2013) 'Simplifying MapReduce data processing', *Int. J. of Computational Science and Engineering*, Vol. 8, No. 3, pp.219–226.

Long, Y. and Hu, X. (2014) 'Dynamic data driven simulation with soft data', in *DEVS '14*, San Diego, CA, USA, pp.16:1–16:8.

Lou, T., Tang, J., Hopcroft, J., Fang, Z. and Ding, X. (2013) 'Learning to predict reciprocity and triadic closure in social networks', *TKDD*.

Nicola, B., Bonchi, F. and Manco, G. (2012) 'Topic-aware social influence propagation models', in *2012 IEEE 12th International Conference on Data Mining (ICDM)*, IEEE, pp.81–90.

Rossi, R.A. et al. (2013) 'Modeling dynamic behavior in large evolving graphs', in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ACM, Rome, Italy.

Saito, R., Kuboyama, T. and Yasuda, H. (2015) 'User behaviour modelling by abstracting low-level window transition logs', *Int. J. of Computational Science and Engineering*, Vol. 11, No. 3, pp.249–258.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L. and Su, Z. (2008) 'Arnetminer: extraction and mining of academic social networks', in *KDD08*, pp.990–998.

Teng, F., Yang, H., Li, T., Magoules, F. and Fan, X. (2015) 'MUS: a novel deadline-constrained scheduling algorithm for Hadoop', *Int. J. of Computational Science and Engineering*, Vol. 11, No. 4, pp.360–367.

Wang, Y., Yin, G., Cai, Z., Dong, Y. and Dong, H. (2015) 'A trust-based probabilistic recommendation model for social networks', *Journal of Network and Computer Applications*, Vol. 55, pp.59–67.

Watts, D.J. and Strogatz, S.H. (1998) 'Collective dynamics of 'small-world' networks', *Nature*, Vol. 393, No. 6684, pp.440–442.

Wu, B. and Shen, H. (2015) 'A time-efficient connected densest subgraph discovery algorithm for big data', *Proc. of the 10th IEEE International Conference on Networking, Architecture, and Storage (NAS)*, Boston, Massachusetts, 6–7 August.

Zhang, L., Cai, Z. and Wang, X. (2016) 'FakeMask: a novel privacy preserving approach for smartphones', *IEEE Transactions on Network and Service Management*, Vol. 13, No. 2, pp.335–348.

Notes

[1] http://snap.stanford.edu/snappy/index.html

[2] http://snap.stanford.edu/

[3] https://www.wikipedia.org/

[4] https://aminer.org/

[5] https://twitter.com/