Influential Nodes Selection: A Data Reconstruction Perspective

Zhefeng Wang, Hao Wang, Qi Liu, Enhong Chen*

School of Computer Science and Technology, University of Science and Technology of China
E-mail: {zhefwang, xdwangh}@mail.ustc.edu.cn, {qiliuql, cheneh}@ustc.edu.cn

ABSTRACT

Influence maximization is the problem of finding a set of seed nodes in social network for maximizing the spread of influence. Traditionally, researchers view influence propagation as a stochastic process and formulate the influence maximization problem as a discrete optimization problem. Thus, most previous works focus on finding efficient and effective heuristic algorithms within the greedy framework. In this paper, we view the influence maximization problem from the perspective of data reconstruction and propose a novel framework named *Data Reconstruction for Influence Maximization*(DRIM). In our framework, we first construct an influence matrix, each row of which is the influence of a node to other nodes. Then, we select k most informative rows to reconstruct the matrix and the corresponding nodes are the seed nodes which could maximize the influence spread. Finally, we evaluate our framework on two real-world data sets, and the results show that DRIM is at least as effective as the traditional greedy algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

Keywords

Social Networks, Influence Maximization, Data Reconstruction

1. INTRODUCTION

In recent years, social network sites such as Facebook[1] and Twitter[2] have become very popular and many people communicate with each other on them. Therefore, social network is now an important platform for viral marketing [4]. Unlike traditional marketing strategies, viral marketing only targets at a small set of influential individuals (nodes) in the network. Thus, the key problem is how to figure out these influential nodes. To this end, Kempe et al. [9] formulate it as a discrete optimization problem known as influence maximization problem. In their paper, two classical propagation models were discussed: Linear Threshold (LT) model [7] and Independent Cascade(IC) model [5]. Furthermore, they proved that influence maximization is an NP-hard problem in both models and proposed a greedy solution. Due to the submodular property of this problem, the greedy algorithm can approximate the optimal solution within a constant factor. Meanwhile, Monte Carlo simulation is used to estimate the influence spread (i.e., the expected number of nodes that will be influenced) of the node set in the greedy algorithm, as influence propagation is treated as a stochastic process in both LT and IC models.

One of the biggest challenges of the greedy algorithm is that we need to run Monte Carlo simulation sufficiently many times (e.g., 10,000) for accurately estimating the influence spread, i.e., very time-consuming. Thus, the following researchers mainly focus on reducing the calculation. Leskovec et al. [11] exploited the submodular property and proposed CELF algorithm which is 700 times faster than the greedy algorithm without effectiveness loss. Chen et al. [3] proposed degree discount heuristic algorithm which is much faster than the greedy algorithm but not as effective as it. Along this line, several other heuristic algorithms have been proposed [2, 4, 6, 15]. These algorithms exploit the property of the social network or approximate the influence propagation models for computing the influence spread of a set of nodes faster [1]. In this way, they speed up the algorithm but are usually less effective.

However, we can explore other aspects of the influence maximization problem besides searching for more effective heuristics. That is, we may think about this problem from a different perspective. Actually, influence maximization finds some influential nodes whose influence can cover the whole network, which is similar to selecting some informative rows to reconstruct a matrix. Inspired by [16, 8], in this paper we treat influence maximization as a data reconstruction problem. Specifically, we propose a novel framework called *Data Reconstruction for Influence Maximization*(DRIM) which finds the influential nodes by minimizing the reconstruction error. In our framework, we first construct an influence matrix, each row of which is the influence of a node to other nodes. Then, we try to select k most informative rows to reconstruct the influence matrix and the corresponding nodes are the seed nodes which could maximize the influence

*Corresponding author

[1] http://www.facebook.com

[2] http://www.twitter.com

SIGIR '14, July 06 - 11 2014, Gold Coast, QLD, Australia
Copyright 2014 ACM 978-1-4503-2257-7/14/07.

spread. We verify our framework on two real-world data sets, and the experimental results show that DRIM is at least as effective as the greedy algorithm. Our contributions can be summarized as follows:

- We view the influence maximization problem from the perspective of data reconstruction and propose a novel framework named DRIM to solve it.
- We evaluate the proposed framework on two real-world data sets. Experimental results show that DRIM is at least as effective as the traditional greedy algorithm.

2. FINDING INFLUENTIAL NODES

In this section, we propose a new framework to solve the influence maximization problem from the perspective of data reconstruction. We first review the definition of influence maximization problem, and then describe the proposed framework in detail.

Let $G = (V, E, T)$ represent a social network, where V is the node set and E is the edge set. If there is an edge from node i to node j in E, $t_{i,j}$ in influence propagation matrix $T = [t_{i,j}]_{n*n}$ is the influence transmission probability from i to j. According to the definition of Kempe et al. [9], influence maximization problem is a discrete optimization problem: given a social network, and a number k, find k nodes, called the seed set, such that by activating them, the expected number of final activated nodes is maximized. Different from traditional methods, we find the influential seed nodes from the perspective of data reconstruction. The proposed framework contains two steps:

1. Construct an influence matrix $X \in R^{N \times N}$, where $\mathbf{x_i} \in R^{1 \times N}$ indicates the influence of node i.
2. Select k most informative rows from the matrix X to reconstruct it, and we can find the influential seed nodes simultaneously.

2.1 Constructing Influence Matrix

There are a few ways to construct the influence matrix. However, in IC and LT model, we have to use Monte Carlo simulation to estimate the influence spread which is very time-consuming. Thus we turn to the linear model proposed by Xiang et al. [14]. In their model, we can get a closed-form solution of the influence of a single node or a set of nodes. Specifically, the influence from i to j, $f_{i \to j}$, is defined as :

$$f_{i \to j} = \alpha_i, \quad \alpha_i > 0, \quad for\ j = i$$

$$f_{i \to j} = \frac{1}{1 + \lambda_j} \sum_{k \in N_j} t_{kj} f_{i \to k}, \quad for\ j \neq i$$

where $N_j = \{j_1, j_2, ... j_m\}$ is j's neighborhood node set, α_j and λ_j are two parameters of the model[3].

According to the solution given by Xiang et al. [14], we can obtain the influence vector $\mathbf{f_i} = [f_{i \to 1}, f_{i \to 2}, ... f_{i \to n}]'$ of node i by solving the linear system $P^{-1} P_{\cdot i} = \mathbf{e_i}$, where $P = (I + \lambda I - T')^{-1}$. And $\mathbf{f_i}$ is exactly the i-th row of the influence matrix X.

2.2 Finding Influential Nodes

Given influence matrix, we can find influential nodes with data reconstruction method. Since each row of the influence matrix is the influence of a node to other nodes, finding informative rows to reconstruct the influence matrix essentially means finding influential nodes whose influence can cover the whole network. Specifically, we try to select k most informative rows to reconstruct the influence matrix and the corresponding nodes are the seed nodes. Inspired by [16, 8], we formulate this problem as follows:

$$\min_{A, \boldsymbol{\beta}} \quad J(A, \boldsymbol{\beta}) = \sum_{i=1}^{N} \left\{ \|\mathbf{x_i} - X^T \mathbf{a_i}\|^2 + \sum_{j=1}^{N} \frac{a_{i,j}^2}{\beta_j} \right\} + \gamma |\boldsymbol{\beta}|_1$$

$$s.t. \quad a_{i,j} \geq 0 \quad \beta_j \geq 0 \quad and \quad \mathbf{a_i} \in R^n \quad i = 1...n, \quad j = 1...n \tag{1}$$

where $A^T = [\mathbf{a_1}, ... \mathbf{a_n}]$. $\boldsymbol{\beta} = [\beta_1, ..., \beta_n]^T$ is an auxiliary variable to control nodes selection. We impose l_1 norm [13] on $\boldsymbol{\beta}$ to induce sparsity. If $\beta_j = 0$, the j-th column must be 0 which means the j-th node is not selected.

We'd like to give more implication about the nonnegative constraints [10] used in Eq. (1). Since one's influence will not fade with the existence of others, which shares a similar assumption with IC model, the nonnegative constraints used here can make the result more interpretable. That is, it allows only additive, not subtractive, combination of the nodes' influence vectors.

By fixing $\mathbf{a_i}$'s and setting the derivative of J with respect to $\boldsymbol{\beta}$ to be zero, we can obtain the closed-form solution of $\boldsymbol{\beta}$:

$$\beta_j = \sqrt{\frac{\sum_{i=1}^{N} a_{i,j}^2}{\gamma}} \tag{2}$$

By fixing $\boldsymbol{\beta}$, the remaining problem can be solved by projected gradient descent. The gradient is :

$$\frac{\partial J(\mathbf{a_i})}{\partial \mathbf{a_i}} = -2X(\mathbf{x_i} - X^T \mathbf{a_i}) + 2diag(\beta)^{-1} \mathbf{a_i} \tag{3}$$

Thus the update formula is:

$$\mathbf{a_i} = max(0, \mathbf{a_i} - C \frac{\partial J(\mathbf{a_i})}{\partial \mathbf{a_i}}) \tag{4}$$

Algorithm 1 gives the details of the procedure.

Algorithm 1: Influential Nodes Selection

```
Input: seeds number $k$ and the influence matrix $X$
Output: $k$ influential nodes
initialize $A$ with random numbers
set $\beta_j = 0, \forall j$
while not converge do
    update $\boldsymbol{\beta}$ by Eq. (2)
    while not converge do
        update $A$ by Eq. (4)
    end
end
return the subscripts of $k$ largest values in $\beta$
```

3. EXPERIMENTS

We provide validation on two real-world data sets. One of them is the Wikipedia who-votes-on-whom network (Wiki-Vote), and the other one is the collaboration network from DBLP. Specifically, we demonstrate that our framework is at least as effective as the greedy algorithm.

[3] Please see [14] for more details

Table 1: Statistics of Data Sets

Name	Nodes	Edges
Wiki-Vote	7,115	103,689
DBLP-IR	8,958	27,732
DBLP-ML	8,896	26,629
DBLP-DM	10,347	33,466

3.1 Experimental Setup

Data Sets. The first data set is downloaded from SNAP[4]. The second data set is downloaded from DBLP[5]. As for the second data set from DBLP, we focus on three research domains (i.e., three subnetworks), which are "Information Retrieval"(IR), "Data Mining"(DM) and "Machine Learning"(ML). We select the papers that are published before January 2013 from several top-ranked journals and conferences for each domain. The authors of these papers are used as nodes in the collaboration network. When two authors have one co-authored paper, an edge will be added between the corresponding nodes.

The propagation probability of an edge (i, j) is set to $\frac{weight(i,j)}{indegree(j)}$, as widely used in the literature [9, 3, 6, 15]. In order to make the linear system in Section 2.1 converge, we slightly change the probability by multiplying it with a small real number. For the propagation probability in real world is quite small, we set it as 0.1 in our experiment.

In total, we get four social networks. More detailed information is shown in Table 1.

Parameter Settings. There are two parameters: α and λ in linear model. When we compute the influence matrix, we set the same α value for all nodes. Here, we set the α to 1, which means that each node has full confidence of itself and the self-influence is 1. As for λ, we choose the same value as [14], which is 0.176. In the data reconstruction step, there is a parameter γ that controls the sparsity of the matrix. We have tried a few values in our experiment and discovered that the algorithm gets its best performance when γ equals to 0.4.

Baseline Algorithms. We compare the proposed framework, referred as **DRIM**, with three baseline algorithms on the IC model. The baseline algorithms are as follows:

- **Greedy**: The greedy algorithm is proposed by Kempe et al. [9]. For each candidate seed set S, we run 10,000 times simulations to obtain the influence spread of S.
- **Degree Discount**: Degree Discount algorithm is a heuristic algorithm proposed by Chen et al. [3]. We set parameter p of this algorithm to 0.01, the same value used in [3].
- **PageRank**: PageRank [12] is often used in network structure and social influence analysis, e.g., Chen et al. [2] also used it as a baseline algorithm in their paper. We run PageRank (with damping factor $d = 0.85$) in the network, and use the top k nodes as the influential seeds.

Given the output of each algorithm, we use it as the initial seeds to compute their influence spread on the IC model. In the computation process, we run 10,000 Monte Carlo simulations to obtain an estimation of the influence spread.

3.2 Experimental Results

Effectiveness validation. We run tests on four social networks to obtain influence spread results. The seed set size k ranges from 1 to 100, and the parameter γ is 0.4. Figure 1 shows the influence spread results on the four social networks. The influence spread results in Figure 1 show that **DRIM** and **Greedy** obviously outperform the **Degree Discount** and **PageRank**. **DRIM** and **Greedy** have a similar effectiveness when the seed size k is small. However, when k increases, **DRIM** has a bit better performance than **Greedy**. We can discover such a tendency in all four social networks. Actually, Table 2 compares the influence spread results of **DRIM** and **Greedy** when k is greater than 80. In Table 2, we use "+" and "-" to denote the wins and losses of **DRIM** compared with **Greedy** respectively. When we pay attention to the nodes these algorithms find, we discover another interesting phenomenon: **DRIM** and **Greedy** find the same top 10 influential nodes on Wiki-Vote data set. As for other social networks, most of the influential nodes that **DRIM** and **Greedy** find are the same, when k increases from 10 to 100. Finally, Table 3 shows the high number of the same influential nodes found by **DRIM** and **Greedy**. This phenomenon demonstrates the effectiveness of **DRIM** from another angle.

Case Study. We show a case study by illustrating the names of influential seed nodes (k=10) in DM domain in Table 4. From Table 4, we can see that the influential authors found by **DRIM** and **Greedy** are quite similar. Because of the lack of ground truth, we refer to top authors list in DM domain provided by Microsoft Academic Search[6]. This list ranks the authors in DM domain according to their field rating. The first four authors found by **DRIM** is exactly the top-4 authors in the list of Microsoft Academic Search. However, the fifth and seventh authors found by **DRIM** are ranked 147 and 466 in the list of Microsoft Academic Search. The remaining four authors are ranked top-100 in the list of Microsoft Academic Search. The influential authors found by **Greedy** have a similar phenomenon. The results show that neither **DRIM** nor **Greedy** simply ranks the influence of a single author. In fact, both algorithms select influential nodes that could maximize the influence spread.

Table 3: Number of the same nodes found by DRIM and Greedy.

k data sets	10	20	50	100
Wiki-Vote	10	15	41	84
DBLP-IR	8	18	35	72
DBLP-ML	9	17	38	72
DBLP-DM	8	17	33	72

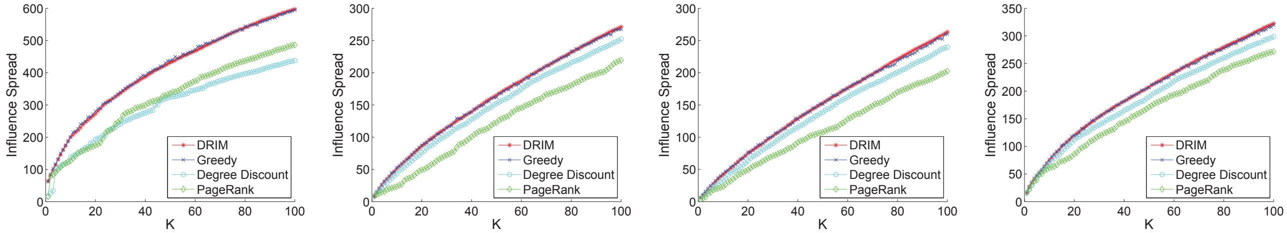
4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel framework to solve influence maximization problem from the perspective of data reconstruction. The proposed framework first constructs the influence matrix, and then finds the influential seed nodes with data reconstruction method. The experimental results show that the proposed framework is at least as effective as the traditional greedy algorithm and outperforms other heuristic algorithms.

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

[5] http://dblp.uni-trier.de/xml/

[6] http://academic.research.microsoft.com



(a) Wiki-Vote (b) DBLP-IR (c) DBLP-ML (d) DBLP-DM

Figure 1: Influence spread on four social networks.

Table 2: Comparison of influence spread of DRIM and Greedy.

k data sets	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	98	99	100
Wiki-Vote	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DBLP-IR	+	-	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DBLP-ML	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DBLP-DM	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Table 4: Names of influential seed nodes (k=10) in DM domain found by DRIM and Greedy.

DRIM	Greedy
J. Han	J. Han
P.S. Yu	P.S. Yu
R.Agrawal	R.Agrawal
C. Faloutsos	W. Fan
W. Fan	Y. Tao
Q. Yang	C.S. Jensen
B.W. Wah	C. Faloutsos
E. Bertino	E. Bertino
J. Pei	Q. Yang
S. Shekhar	J. Pei

There are several future directions related to this work. First, we will parallelize Algorithm 1 to speed up the proposed framework. Second, we try to preprocess the network with clustering method to reduce the size of the influence matrix. Third, we believe that the idea used to solve the influence maximization problem can be applied to solve other similar discrete optimization problems. Thus we'll extend our framework to solve other similar problems.

Acknowledgment. This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China(Grant No. 61325010), the Science and Technology Development of Anhui Province, China(Grant No.s 13Z02008-5 and 1301022064), the International Science & Technology Cooperation Plan of Anhui Province(Grant No. 1303063008), the Anhui Provincial Natural Science Foundation(Grant No. 1408085QF110), and the Fundamental Research Funds for the Central Universities of China(Grant No. WK0110000042).

5. REFERENCES

[1] W. Chen, L. V. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*. Morgan and Claypool, 2013.

[2] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD*, pages 1029–1038. ACM, 2010.

[3] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *SIGKDD*, pages 199–208. ACM, 2009.

[4] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97. IEEE, 2010.

[5] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[6] A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, pages 211–220. IEEE, 2011.

[7] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420, 1978.

[8] Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, D. Cai, and X. He. Document summarization based on data reconstruction. In *AAAI*, 2012.

[9] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.

[10] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD*, pages 420–429. ACM, 2007.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[13] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[14] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. Pagerank with priors: An influence propagation perspective. In *IJCAI*, 2013.

[15] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad. On approximation of real-world influence spread. In *Machine Learning and Knowledge Discovery in Databases*, pages 548–564. Springer, 2012.

[16] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *ICML*, pages 1081–1088. ACM, 2006.