Scalable Influence Analysis in Mobile Social Networks

1, First Author
Ke Xu, 2 Xinfang Zhang, 3 Jianxin Zhu

1 School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
College of Computer Science, South-Central University for Nationalities, Wuhan, China
cole.xu@gmail.com

2, Corresponding Author
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

3 School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

Abstract

Influence is a complex and subtle force that governs the dynamics of social networks as well as the behaviors of involved users. In large social networks, nodes are influenced by others for various reasons. Understanding influence can benefit various applications such as viral marketing, recommendation, and information retrieval. However, most existing works on social influence analysis have focused on verifying the existence of social influence. In this paper, an algorithm is proposed which utilizes the heterogeneous link information and the textual content associated with each node in the network to mine micro and macro influence. Based on the direct and indirect influence, the Map-Reduce framework for social influence analysis algorithm is proposed to derive the indirect influence between nodes. We further study how the discovered topic-level influence can help the prediction of user behaviors. Empirical studies on a large real-world mobile social network show that our algorithm has a good scalability performance.

Keywords: Mobile Network, Influence, Map-reduce

1. Introduction

With the emergence and rapid proliferation of social applications and media, such as instant messaging (e.g., MSN, Skype), blogs (e.g., Blogger, WordPress), wikis (e.g., Wikipedia, PBWiki), microblogs (e.g., Twitter, Renren), social networks (e.g., MySpace, Facebook) to mention a few, there is little doubt that social influence is becoming a prevalent, complex and subtle force that governs the dynamics of all social networks. As data grows, data mining and machine learning applications also start to embrace the Map-Reduce paradigm, e.g., news personalization with Map-Reduce EM algorithm [1, 20, 21], Map-Reduce of several machine learning algorithms on multi-core architecture [2]. Recently, Papadimitriou and Sun [3] illustrates a mining framework on Map-Reduce along with a case-study using co-clustering [22, 23].

Social network analysis often focus on macro-level models such as degree distributions, diameter, clustering coefficient, communities, small world effect, preferential attachment, etc.; work in this area includes [4]. Recently, social influence study has started to attract more attention due to many important applications. However, most of the works on this area present qualitative findings about social influences [5].

Much effort has been made for social network analysis and a large number of works has been done [23, 24]. Quite a few metrics have been defined to characterize a social network, such as betweenness, closeness, centrality, centralization, etc. A common application of the social network analysis is Web community discovery. For example, Crandall et al. studied the correlation between social similarity and influence [6]. As for social influence analysis, [7, 8] propose methods to qualitatively measure the existence of influence. Other similar work can be referred to [9]. To the best of our knowledge, no previous work has been conducted for quantitatively measuring scalable social influence on mobile networks. In this paper, we focus on measuring the strength of social influence quantitatively among mobile phone users.
2. Related works

Recently some attempts have been made to analyze the dynamics in the social networks. For example, Scripps et al. [10] investigated how different pre-processing decisions and different network forces such as selection and influence affect the modeling of dynamic networks. Other similar work can be referred to [9].

Social network and source analysis has attracted many researchers’ interests recently [11, 12]. For example, Sun et al. [11] studied the clustering problem on heterogeneous networks. Ye et al. [12] fused heterogeneous data sources to study the Alzheimer’s disease. Many works have tried to combine the sufficient information on heterogeneous networks, e.g., the text and links, to detect communities, to analyze the evolution of networks and to model relational learning [13, 14]. Besides, some researchers studied the problem of information diffusion over networks [15].

Many efforts have been made for estimating link influence between individual pages. For example, Dietz et al. [16] proposed a citation influence topic model to model the influential strength between papers. Nallapati et al. [17] proposed two topic models to jointly model text and citation relationships. Considerable work has been conducted to validate the existence of influence and study its effort from the global view of the whole network, e.g., influence maximization on a person network [18, 5, 19]; influence diffusion over networks [20]; influence and correlation on social activities [7]; correlation between influence and similarity [6]. Several efforts have been made to identify the existence of social influence in online social networks. Anagnostopoulos et al. [7] gave a theoretical justification to identify influence as a source of social correlation when the time series of user actions are available. Singla and Richardson [8] studied the correlation between personal behaviors and their interests. Crandall et al. [6] further investigated the correlation between social similarity and influence.

In this paper, we propose a distributed algorithm has been implemented under the Map-reduce programming model, for quantitatively analyzing the micro and macro social influences in campus mobile network. Compared with the existing work, our algorithm can have a good scalability performance.

3. Theoretical modeling

Community structure is a basic property of a CMSN and communities represent real circles of social groups in which members are more likely to have common interest with each other [4]. Suppose we are given a social network together with the estimates of reciprocal influence between individuals in the network, and suppose that we want to push new application software in the china mobile market. The idea behind viral marketing is that by targeting the most influential users in the network we can activate a chain-reaction of influence driven by word-of-mouth.

3.1 Mining Influence Assumptions

Influence is interacted with many potential factors, e.g., similarity, correlation and etc. [1, 4]. Commonsense knowledge is needed to quantitatively model the influence strength. Here we have two general assumptions.

Assumption 1. Users with similar interests have a stronger influence on each other.

This assumption actually corresponds to the influence and selection theory [1]. In real networks, the similarity can be calculated based on the textual content associated with each user. Thus, influence can be represented as to which extent the textual content is "copied" from the influencing nodes.

Assumption 2. Users whose actions frequently correlate have a stronger influence on each other.

In heterogeneous networks, the link weight is usually used to indicate the correlation strength between nodes, which can be calculated by the co-occurrence frequency of nodes.

3.2 Campus Mobile Social Network

We extract a Campus Mobile Social Network from the call log and model it as a network: a phone user corresponds to a node; a directed edge from node \( u \) to node \( v \) is established, if there exists communication from \( u \) to \( v \). We denoted a network \( G = (V, E ; \Omega) \). \( V \) is a set of nodes, which are
Scalable Influence Analysis in Mobile Social Networks
Ke Xu, Xinfang Zhang, Jianxin Zhu

The links can be directed or undirected. Between users, authoring relationships between users and documents and links between documents. The links can be directed or undirected.

Definition 1. [Micro-influence] Given two user nodes \( u, v \) in a heterogeneous network, we denote \( \phi_v(u) \in R \) as the influential strength of user \( u \) on user \( V \). Furthermore, if \( e_{uv} = 1 \), we call \( \phi_v(u) \) the direct micro-influence of user \( u \) on \( v \); if \( e_{uv} = 0 \), we call \( \phi_v(u) \) the indirect micro-influence of user \( u \) on \( v \).

Therefore, the micro-influence propagation is defined as follows:

\[
\phi_v(u) = \text{Arg maximum} \ (s \in N(v) : \phi_v(s) \cdot \phi_v(u))
\]

where \( N(v) \) is the set of neighbors of node \( v \). In particular, if we use multiplication as the concatenation function and addition as the aggregation function, then the atomic influence propagation can be instantiated as:

\[
\phi_v(u) = \sum_{s \in N(v)} \phi_v(s) \cdot \phi_v(u)
\]

Direct influence indicates the influence between two nodes which are connected while indirect influence indicates the influence of two nodes which are not connected. Please note that the influence is asymmetric, i.e., \( \phi_v(u) \neq \phi_u(v) \). Based on the influence between node pairs, we can further define the concept of global influence.

Definition 2. [Macro-influence] Given a heterogeneous network, \( \psi(v) \) is defined as the global macro-influence of \( v \), which represents the global macro-influential strength of \( v \) over the whole network.

\[
\psi(v) = \sum u \phi_v(u)
\]

Intuitively, the global influence of one node on the network \( \psi(u) \) should be related to its influence on all the other nodes. If one node strongly influences many other nodes, its global influence might be also strong. The influence scores \( \phi_v(u) \) include both direct and indirect influences.

The global influence strength has a close relationship with the direct/indirect influence. For example, if a user has a strong influence on other users, it is probably that he is very influential globally. We propose an affinity propagation algorithm, which converts the message passing rules into equivalent update rules passing message directly between nodes rather than on the factor graph. The algorithm is summarized in Algorithm 1. In the algorithm, we first use logarithm to transform sum-product into max-sum, and introduce two sets of variables \( \{r_{uv}\}_{v=1}^{T} \) and \( \{a_{uv}\}_{v=1}^{T} \) for each edge \( e_{uv} \). The new update rules for the variables are as follows:

\[
r_{uv} = b_{uv} - \max k \in N(v) \{ b_{uv} + a_{uk} \}
\]

\[
a_{uv} = \min k \in N(v) \{ r_{uv} , 0 \}
\]

where \( N(v) \) denotes the neighboring nodes of node \( v \), \( r_{uv} \) is the influence message sent from node \( u \) to node \( v \), and \( a_{uv} \) is the influence message sent from node \( v \) to node \( u \), initiated by 0, and \( b_{uv} \) is the logarithm of the normalized feature function.
3.1 Distributed Social Influence Analyze Algorithm

Map-Reduce is a parallel programming paradigm, originally introduced by Google [1, 8], whose central focus is to simplify the processing of large datasets on inexpensive cluster computers. The map function takes as input a set of key-value pairs, designated as \(k_1 \) and \(v_1\), provided directly from the user-defined input files. Within the map function, the user specifies what to do with these keys and values. The map function outputs another set of keys and values, designated as \(k_2\) and \(v_2\). The reduce function sorts the key value pairs by \(k_2\). All of the associated values \(v_2\) are reduced and emitted as value \(v_3\). The map and reduce functions are as follows:

\[
\text{Map} \ (k_1, v_1) \rightarrow [ \ (k_2, v_2) ] \quad (7)
\]

\[
\text{Reduce} \ (k_2, [v_2]) \rightarrow [ \ v_3 ] \quad (8)
\]

At the Map-Reduce run-time level, the map operations are distributed by the master-server to the chunk-servers. The scheduler makes an effort to schedule computation on the same node where the data is stored. Meanwhile, other chunk-servers assigned to the reduce phase begin to take the \((k_2, v_2)\) value pairs and sort them by \(k_2\). These sorted arrays of \(v_2\) values are passed to the reduce functions on these same assigned nodes. These outputs are finally saved on the GFS. It is quite common for an application to string together many simpler Map-Reduce operations.

We first partition the large social network graph into subgraphs and distribute each subgraph to a process node. In each subgraph, there are two kinds of nodes: internal nodes and marginal nodes. Internal nodes are those all of whose neighbors are inside the very subgraph; marginal nodes have neighbors in other subgraphs. For every subgraph \(G\), all internal nodes and edges between them construct the closed graph \(G\). The marginal nodes can be viewed as "the supporting information" for updating the rules. For easy explanation, we consider the distributed learning algorithm on a single topic and thus the map stage and the reduce stage can be defined as follows.

(1) Mapper-stage: each process node scans the closed graph \(G\) of the assigned subgraph \(G\). Note that every edge \(e_{uv}\) has two values \(d_{uv}\) and \(r_{uv}\). Thus, the map function is defined as for every key/value pair \(e_{uv} / a_{uv}\), it issues an intermediate key/value pair \(e_{uv} / (b_{uv} + a_{uv})\); and for key/value pair \(e_{uv} / r_{uv}\), it issues an intermediate key/value pair \(e_{uv} / r_{uv}\).

(2) Reducer-stage: each process node collects all values associated with an intermediate key \(e_{uv}\) to generate new \(r_{uv}\) according to Eq. (4), and all intermediate values associated with the same key \(e_{uv}\) to generate new \(a_{uv}\) according to Eqs. (5) and (6).

Algorithm 1: Marco-Influence computation

\begin{algorithm}
\caption{Marco-Influence computation}
\label{algo:marco}
\begin{algorithmic}
\Function{MarcoInfluence}{network $G$, Initial local influence $\psi^0$; and the number of iteration $K$}
\State Initialize: $\psi^f = \psi^0$
\For{$k = 1$ \textbf{to} $K$}
\For{each $\psi^k_v(u) \neq 0$}
\For{each $w$ : $\psi^k_v(u) = 0$}
\State $\psi^k_v(w) = \psi^k_v(u) \cdot \psi^{k-1}_v(w)$;
\State $\psi^f_v(w) = \psi^f_v(w) + \delta^{k-1} \cdot \psi^{k-1}_v(w)$;
\EndFor
\EndFor
\EndFor
\EndFunction
\end{algorithmic}
\end{algorithm}
4. Experiment

In this section, we present several of experiments to evaluate the efficiency and effectiveness of the proposed approach. The data sets we used and experimental setup beforehand are described at first, and then the results with different parameters are shown.

4.1. Data sets

We have a three-month CDR (call detailed record) data of a campus CMCC-V-Net from China Mobile, the largest mobile communication service provider in China. A virtual campus mobile network of China Mobile Communications Corporation (CMCC-V-Net) in which all users are from universities or colleges. The phone number belonged to which students are registered when they join in the CMCC-V-Net.

We extract a mobile social network from the CDR data for 3 months using the method presented in Section 3, and obtain a network with 263689 nodes from more than 30 universities. Figure 1 shows the degree distribution of the campus mobile social network. We can see that it follows the power-law distribution, i.e. it is a scale-free network as many other social networks.

![Figure 1. Degree Distribution of Campus Mobile Social Network](image_url)

<table>
<thead>
<tr>
<th>Date</th>
<th>Call number</th>
<th>Duration (s)</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-10-1 00:11:19</td>
<td>610 xxx</td>
<td>308</td>
<td>Received</td>
</tr>
<tr>
<td>2011-10-1 22:11:21</td>
<td>621 xxx</td>
<td>201</td>
<td>Dialed</td>
</tr>
<tr>
<td>2011-10-5 10:15:45</td>
<td>611xxx</td>
<td>112</td>
<td>Received</td>
</tr>
<tr>
<td>2011-10-6 20:19:28</td>
<td>810xxx</td>
<td>637</td>
<td>Received</td>
</tr>
<tr>
<td>2011-10-7 07:23:19</td>
<td>860xxx</td>
<td>46</td>
<td>Received</td>
</tr>
<tr>
<td>2011-10-7 15:29:27</td>
<td>660xxx</td>
<td>126</td>
<td>Dialed</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2011-10-11 01:19:18</td>
<td>661xxx</td>
<td>319</td>
<td>Received</td>
</tr>
<tr>
<td>2011-10-12 10:11:19</td>
<td>661xxx</td>
<td>62</td>
<td>Received</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1. Calling detail records
4.2. Experimental results

Our algorithm automatically divides the job and distributes them to each node. So we can dynamically add the quantity of nodes, which will enhance the performance of the algorithm. We run the program on a blade-cluster with 32 nodes. Each node has 4 processors which are equipped with Intel (R) Xeon (TM) CPU 2.80 GHZ and 4 GB memory. In the experiment, we run the algorithm to do the same task on cluster of varying nodes.

Here we present 2 methods for expert identification: (1) PageRank with Language Modeling; (2) Our algorithm - PageRank with global Macro-influence.

PageRank with Language Modeling as the baseline method is to combine the language model and PageRank [24]. Language model is to estimate the relevance of a candidate with the query and PageRank is to estimate the authority of the candidate. There are different combination methods. The simplest combination method is to multiply or sum the PageRank ranking score and the language model relevance score.

We evaluate the speedup of the distributed learning algorithm on the 32 computer nodes using the citation data set with different sizes. It can be seen from Figure 2 (a) that when the size of the data set increase to nearly one million edges, the distributed learning starts to show a good parallel efficiency (speedup>3). This confirms that distributed social influence analyze algorithm like many distributed learning algorithms is good on large-scale data sets.

Using our large citation data set, we also perform speedup experiments on a Hadoop platform using 2, 4, 8, 16, 24, 32 computer nodes (since we did not have access to a large number of computer nodes). The speedup, shown in 2 (b), show reasonable parallel efficiency, with a > 4x speedup using 32 computer nodes.

Furthermore, we compare the influence prediction performance before and after influence propagation on our labeled data set. Table 2 shows the values when damping factor $\delta$ and iteration number $K$ changes, which proves that the influence prediction performance is enhanced based on indirect influence obtained by influence propagation. Moreover, the influence prediction performance is robust to the parameters $K$ and $\delta$. In particular, when $K$ changes, the performance change little, which is consistent to the observation in Fig. 1. It means that influence do propagate over the network, but the effort of propagation is reduced a lot when the degree increases.

<table>
<thead>
<tr>
<th>Table 2. Influence prediction performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
</tr>
<tr>
<td>0.7132</td>
</tr>
<tr>
<td>$K = 1$</td>
</tr>
<tr>
<td>$K = 5$</td>
</tr>
<tr>
<td>$K = 8$</td>
</tr>
<tr>
<td>$K = 10$</td>
</tr>
</tbody>
</table>
We apply our model on CCMN social networks and discover the concrete influence strength between users. Then we use the learned influence for user behavior prediction as described in Section 3 to demonstrate the efforts of obtained influence on social network applications. In this experiment, we empirically set the number of topics to be 30, \( K = 8 \) and \( \delta = 0.8 \).

We estimate each sample's probability by using the three prediction methods in Section 3. Table 3 shows the average and variance values of the predicted probabilities on all the samples, where \( d \) denotes the degree of indirect influence. The results demonstrate that using influence can improve the predicted probabilities a lot.

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>Direct influence</th>
<th>Indirect influence</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td></td>
<td>( d = 2 )</td>
<td>( d = 2 )</td>
</tr>
<tr>
<td>Average</td>
<td>0.316</td>
<td>0.462</td>
<td>0.456</td>
</tr>
<tr>
<td>Variance</td>
<td>0.192</td>
<td>0.225</td>
<td>0.233</td>
</tr>
</tbody>
</table>

Then given a threshold, we calculate the prediction precision, which means how many testing samples' probabilities are larger than the threshold. Fig. 3 show four curves of prediction precision changing with the threshold on CCMN network CRDs (Calling retail records) data sets respectively. The results demonstrate that influence-based behavior prediction approach outperforms the baseline. Thus it proves that the influence obtained from our model benefits the user behavior prediction greatly. In particular, it shows that on CCMN social network the indirect influence get lower performance than direct influence.

5. Conclusion

Science is rapidly becoming data management problem. Scaling existing data analysis techniques is very important to expedite the knowledge discovery process. In this paper, we study a novel problem of mining influence nodes on a CCMN. Our approach to solve this problem primarily consists of two steps, i.e., a probabilistic model to mine direct influence between nodes and a distributed influence propagation method to mine indirect and global influence.

A distributed algorithm has been implemented under the Map-reduce MapReduce style data analysis platform. Experimental results on CRDs data sets demonstrate that the proposed approach can effectively discover the topic-based social influences. The distributed learning algorithm also has a good scalability performance. We have done extensive experiments on different types of heterogeneous social networks, show some interesting cases and demonstrate that using influence can benefit the prediction performance greatly.

The general problem of influence analysis on informative networks represents a new and interesting research direction in social network mining. There are many potential future directions of this work.
How to make use of the useful supervised information to improve the analysis quality is an interesting problem. Another potential issue is to scale up the approach to large data set. This work opens to several interesting directions for future work. Notably, it is relevant to take spatial information of mobile customers into consideration, and construct locations based social networks to find influential nodes; it is also interesting to study the evolution of influential nodes over time.

6. Acknowledgment

This research was supported by Supported by “the Fundamental Research Funds for the Central Universities”, South-Central University for Nationalities (CZY12008), National Science and Technology Major Project of the Ministry of Science and Technology of China (2011ZX03002-001-01) and National Natural Science Foundation of China (60975021, 60902053).

7. References

[11] Yizhou Sun, Yintao Yu, Jiawei Han, “Ranking-based clustering of heterogeneous information networks with star network schema”, In Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’09), pp. 797-806, 2009.


