Social Influence Analysis in Large-scale Networks

Jie Tang  
Dept. of Computer Science  
Tsinghua University, China  
jietang@tsinghua.edu.cn

Jimeng Sun  
IBM TJ Watson Research Center, USA  
jimeng@us.ibm.com

Chi Wang and Zi Yang  
Dept. of Computer Science  
Tsinghua University, China  
yz@keg.cs.tsinghua.edu.cn

ABSTRACT

In large social networks, nodes (users, entities) are influenced by others for various reasons. For example, the colleagues have strong influence on one’s work, while the friends have strong influence on one’s daily life. How to differentiate the social influences from different angles(topics)? How to quantify the strength of those social influences? How to estimate the model on real large networks?

To address these fundamental questions, we propose Topical Affinity Propagation (TAP) to model the topic-level social influence on large networks. In particular, TAP can take results of any topic modeling and the existing network structure to perform topic-level influence propagation. With the help of the influence analysis, we present several important applications on real data sets such as 1) what are the representative nodes on a given topic? 2) how to identify the social influences of neighboring nodes on a particular node?

To scale to real large networks, TAP is designed with efficient distributed learning algorithms that is implemented and tested under the Map-Reduce framework. We further present the common characteristics of distributed learning algorithms for Map-Reduce. Finally, we demonstrate the effectiveness and efficiency of TAP on real large data sets.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Text Mining; H.2.8 [Database Management]: Database Applications

General Terms
Algorithms, Experimentation

Keywords
Social Influence Analysis, Topical Affinity Propagation, Large-scale Network, Social Networks

1. INTRODUCTION

With the emergence and rapid proliferation of social applications and media, such as instant messaging (e.g., IRC, AIM, MSN, Jabber, Skype), sharing sites (e.g., Flickr, Picassa, YouTube, Plaxo), blogs (e.g., Blogger, WordPress, LiveJournal), wikis (e.g., Wikipedia, PBWiki), microblogs (e.g., Twitter, Jaiku), social networks (e.g., MySpace, Facebook, Ning), collaboration networks (e.g., DBLP) 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. Therefore, there is a clear need for methods and techniques to analyze and quantify the social influences.

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 [1, 11, 19, 23]. 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[14, 16]. In this paper, we focus on measuring the strength of topic-level social influence quantitatively. With the proposed social influence analysis, many important questions can be answered such as 1) what are the representative nodes on a given topic? 2) how to identify topic-level experts and their social influence to a particular node? 3) how to quickly connect to a particular node through strong social ties?

Motivating Application

Several theories in sociology [14, 16] show that the effect of the social influence from different angles (topics) may be different. For example, in research community, such influences are well-known. Most researchers are influenced by others in terms of collaboration and citations. The most important information in the research community are 1) co-author networks, which capture the social dynamics of the community, 2) their publications, which imply the topic distribution (interests) of the authors. The key question is how to quantify the influence among researchers by leveraging these two pieces.

In Figure 1, the left figure illustrates the input: a co-author network of 7 researchers, and the topic distribution of each researcher. For example, George has the same probability (.5) on both topics, “data mining” and “database”; The right figure shows the output of our social influence analysis: two social influence graphs, one for each topic, where the arrows indicate the direction and strength. We see, Ada is the key person on “data mining”, while Eve is the key person on “database”. Thus, the goal is how to effectively and efficiently obtain the social influence graphs for real large networks.

Challenges and Contributions

The challenges of computing social influence graphs are the following:

- **Multi-aspect.** Social influences are associated with different topics. E.g., A can have high influence to B on a particular topic, but B may have a higher influence to A on another topic. It is important to be able to differentiate those influences from multiple aspects.
The goal of social influence analysis is to derive the topic-level social influence scores on each edge. For a social network $G = (V, E)$ and a topic model on the nodes $V$, we compute topic-level social influence graphs $G_z = (V_z, E_z)$ for all topics $1 \leq z \leq T$. The key features of TAP are the following:

- **Node-specific.** Social influences are not a global measure of importance of nodes, but an importance measure on links between nodes.
- **Scalability.** Real social networks are getting bigger with thousands or millions of nodes. It is important to develop the method that can scale well to real large data sets.

To address the above challenges, we propose Topical Affinity Propagation (TAP) to model the topic-level social influence on large networks. In particular, TAP takes 1) the results of any topic modeling such as a predefined topic ontology or topic clusters based on pLSI [15] and LDA [3] and 2) the existing network structure to perform topic-level influence propagation. More formally, given a social network $G = (V, E)$ and a topic model on the nodes $V$, we compute topic-level social influence graphs $G_z = (V_z, E_z)$ for all topics $1 \leq z \leq T$. The key features of TAP are the following:

- TAP provides topical influence graphs that quantitatively measure the influence on a fine-grain level;
- The influence graphs from TAP can be used to support other applications such as finding representative nodes or constructing the influential subgraphs;
- An efficient distributed learning algorithm is developed for TAP based on the Map-Reduce framework in order to scale to real large networks.

The rest of the paper is organized as follows: Section 2 formally formulates the problem; Section 3 explains the proposed approach. Section 4 presents experimental results that validate the computational efficiency of our methodology. Finally, Section 5 discusses related work and Section 6 concludes.

2. **OVERVIEW**

In this section, we present the problem formulation and the intuition of our approach.

2.1 **Problem Formulation**

The goal of social influence analysis is to derive the topic-level social influences based on the input network and topic distribution on each node. First we introduce some terminology, and then define the social influence analysis problem.

**Topic distribution:** In social networks, a user usually has interests on multiple topics. Formally, each node $v \in V$ is associated with a vector $\theta_v \in \mathbb{R}^T$ of $T$-dimensional topic distribution ($\sum_z \theta_{vz} = 1$). Each element $\theta_{vz}$ is the probability (importance) of the node on topic $z$.

**Topic-based social influences:** Social influence from node $s$ to $t$ denoted as $\mu_{st}$ is a numerical weight associated with the edge $e_{st}$. In most cases, the social influence score is asymmetric, i.e., $\mu_{st} \neq \mu_{ts}$. Furthermore, the social influence from node $s$ to $t$ will vary on different topics.

Thus based on the above concepts, we can define the tasks of topic-based social influence analysis. Given a social network $G = (V, E)$ and a topic distribution for each node, the goal is to find the topic-level influence scores on each edge.

**Problem 1.** Given 1) a network $G = (V, E)$, where $V$ is the set of nodes (users, entities) and $E$ is the set of directed/undirected edges, 2) $T$-dimensional topic distribution $\theta_v \in \mathbb{R}^T$ for all node $v \in V$, how to find the topic-level influence network $G_z = (V_z, E_z)$ for all topics $1 \leq z \leq T$? Here $V_z$ is a subset of nodes that are related to topic $z$ and $E_z$ is the set of pair-wise weighted influence relations over $V_z$, each edge is the form of a triplet $(v_{s}, v_{t}, \mu_{st}^z)$ (or shortly $(e_{st}, \mu_{st}^z)$), where the edge is from node $v_s$ to node $v_t$ with the weight $\mu_{st}^z$.

Our formulation of topic-based social influence analysis is quite different from existing works on social network analysis. For social influence analysis, [2] and [21] propose methods to qualitatively measure the existence of influence. [6] studies the correlation between social similarity and influence. The existing methods mainly focus on qualitative identification of the existence of influence, but do not provide a quantitative measure of the influence strength.

2.2 **Our Approach**

The social influence analysis problem poses a unique set of challenges:

First, how to leverage both node-specific topic distribution and network structure to quantify social influence? In another word, a user’s influence on others not only depends on their own topic distribution, but also relies on what kinds of social relationships they have with others. The goal is to design a unified approach to utilize both the local attributes (topic distribution) and the global structure (network information) for social influence analysis.

Figure 1: Social influence analysis illustration using the co-author network.
Second, how to scale the proposed analysis to a real large social network? For example, the academic community of Computer Science has more than 1 million researchers and more than 10 million coauthor relations; Facebook has more than 50 million users and hundreds of millions of different social ties. How to efficiently identify the topic-based influential strength for each social tie is really a challenging problem.

Next we discuss the data input and the main intuition of the proposed method.

**Data Input:**
Two inputs are required to our social influence analysis: 1) networks and 2) topic distribution on all nodes.

The first input is the network backbone obtained by any social networks, such as online social networks like Facebook and MySpace.

The second input is the topic distribution for all nodes. In general, the topic information can be obtained in many different ways. For example, in a social network, one can use the predefined categories as the topic information, or use user-assigned tags as the topic information. In addition, we can use statistical topic modeling [3, 15, 18] to automatically extract topics from the social networking data. In this paper, we use the topic modeling approach to initialize the topic distribution of each node.

**Topical Affinity Propagation (TAP):**
Based on the input network and topic distribution on the nodes, we formalize the social influence problem in a topical factor graph model and propose a topical affinity propagation on the factor graph to automatically identify the topic-specific social influence.

Our main idea is to leverage an affinity propagation at the topic-level for social influence identification. The approach is based on the theory of factor graph [17], in which the observation data are cohesive on both local attributes and relationships. In our setting, the node corresponds to the observation data in the factor graph and the social relationship corresponds to edge between the observation data in the graph. Finally, we propose two different propagation rules: one based on message passing on graphical models, the other one is a parallel update rule that is suitable for Map-Reduce framework.

### 3. TOPICAL AFFINITY PROPAGATION

The goal of topic-based social influence analysis is to capture the following information: nodes’ topic distributions, similarity between nodes, and network structure. In addition, the approach has to be able to scale up to a large scale network. Following this thread, we first propose a Topical Factor Graph (TFG) model to incorporate all the information into a unified probabilistic model. Second, we propose Topical Affinity Propagation (TAP) for model learning. Third, we discuss how to do distributed learning in the Map-Reduce framework. Finally, we illustrate several applications based on the results of social influence analysis.

#### 3.1 Topical Factor Graph (TFG) Model

Now we formally define the proposed TFG model.

**Variables** The TFG model has the following components: a set of observed variables \( \{v_i\}_{i=1}^{N} \) and a set of hidden vectors \( \{y_i\}_{i=1}^{N} \), which corresponds to the \( N \) nodes in the input network. Notations are summarized in table 1.

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>number of nodes in the social network</td>
</tr>
<tr>
<td>( M )</td>
<td>number of edges in the social network</td>
</tr>
<tr>
<td>( T )</td>
<td>number of topics</td>
</tr>
<tr>
<td>( V )</td>
<td>the set of nodes in the social network</td>
</tr>
<tr>
<td>( E )</td>
<td>the set of edges</td>
</tr>
<tr>
<td>( v_i )</td>
<td>a single node</td>
</tr>
<tr>
<td>( y_{iz} )</td>
<td>node-( v_i )’s representative on topic ( z )</td>
</tr>
<tr>
<td>( y_i )</td>
<td>the hidden vector of representatives for all topics on node ( v_i )</td>
</tr>
<tr>
<td>( \theta_i^z )</td>
<td>the probability for topic ( z ) to be generated by the node ( v_i )</td>
</tr>
<tr>
<td>( e_{ij} )</td>
<td>an edge connecting node ( v_i ) and node ( v_j )</td>
</tr>
<tr>
<td>( w_{ij} )</td>
<td>the similarity weight of the edge ( e_{ij} ) w.r.t. topic ( z )</td>
</tr>
<tr>
<td>( \mu_{iz} )</td>
<td>the social influence of node ( v_i ) on node ( v_j ) w.r.t. topic ( z )</td>
</tr>
</tbody>
</table>

The hidden vector \( y_i \in \{1, \ldots, N\}^{T} \) models the topic-level influences from other nodes to node \( v_i \). Each element \( y_{iz} \), taking the value from the set \( \{1, \ldots, N\} \), represents the node that has the highest probability to influence node \( v_i \) on topic \( z \).

For example, Figure 2 shows a simple example of an TFG. The observed data consists of four nodes \( \{v_1, \ldots, v_4\} \), which have corresponding hidden vectors \( Y = \{y_1, \ldots, y_4\} \). The edges between the hidden nodes indicate the four social relationships in the original network (aka the edges of the input network).

**Feature Functions** There are three kinds of feature functions:

- **Node feature function** \( g(v_i, y_i, z) \) is a feature function defined on node \( v_i \) specific to topic \( z \).
- **Edge feature function** \( f(y_i, y_j, z) \) is a feature function defined on the edge of the input network specific to topic \( z \).
- **Global feature function** \( h(y_1, \ldots, y_N, k, z) \) is a feature function defined on all nodes of the input network w.r.t. topic \( z \).

Basically, node feature function \( g \) describes local information on nodes, edge feature function \( f \) describes dependencies between nodes via the edge on the graph model, and global feature function captures constraints defined on the network.

In this work, we define the node feature function \( g \) as:

\[
\begin{align*}
g(v_i, y_i, z) = & \begin{cases} 
\frac{w_{ij}^z y_i^z}{\sum_{j \in NB(i)(z)} w_{ij}^z} & y_i^z \neq i \\
\frac{w_{ij}^z y_i^z}{\sum_{j \in NB(i)(z)} w_{ij}^z} & y_i^z = i 
\end{cases}
\end{align*}
\]

where \( NB(i) \) represents the indices of the neighboring nodes of node \( v_i \); \( v_i^z = \theta_i^z \alpha_{ij} \) reflects the topical similarity or interaction strength between \( v_i \) and \( v_j \), with \( \theta_i^z \) denoting the importance of node \( j \) to topic \( z \), and \( \alpha_{ij} \) denoting the weight of the edge \( e_{ij} \). \( \alpha_{ij} \) can be defined by different ways. For example, in a coauthor network, \( \alpha_{ij} \) can be defined as the number of papers coauthored by \( v_i \) and \( v_j \). The above definition of the node feature function has the following intuition: if node \( v_i \) has a high similarity/weight with node \( v_j \), then \( v_{ij} \) may have a high influence on node \( v_i \); or if node \( v_i \) is trusted by other users, i.e. other users take him as an high influential node on them, then it must also “trust” himself highly (taking himself as a most influential user on him).

As for the edge feature function, we define a binary feature function, i.e., \( f(y_i, y_j, z) = 1 \) if and only if there is an edge \( e_{ij} \) between node \( v_i \) and node \( v_j \), otherwise 0. We also define a global edge feature function \( h \) on all nodes, i.e.:

\[
h(y_1, \ldots, y_N, k, z) = \begin{cases} 
0 & \text{if } y_i^k = k \text{ and } y_i^z \neq k \text{ for all } i \neq k \\
1 & \text{otherwise.}
\end{cases}
\]

Intuitively, \( h(\cdot) \) constrains the model to bias towards the “true” representative nodes. More specially, a representative node on topic \( z \) must be the representative of itself on topic \( z \), i.e., \( y_i^z = k \).

And it must be a representative of at least another node \( v_i \), i.e., \( \exists y_i^z = k, i \neq k \).
Joint Distribution Next, a factor graph model is constructed based on this formulation. Typically, we hope that a model can best fit (reconstruct) the observation data, which is usually represented by maximizing the likelihood of the observation. Thus we can define the objective likelihood function as:

$$P(v, Y) = \frac{1}{Z} \prod_{k=1}^{N} \prod_{i=1}^{T} h(y_{ik}, \ldots, y_{Nk}; k, z) \prod_{i=1}^{N} \prod_{z=1}^{T} g(v_{i1}, \ldots, v_{N}, z) \prod_{e \in E} f(y_{e1}, y_{e2}; z)$$

(3)

where \(v = [v_1, \ldots, v_N]\) and \(Y = [y_1, \ldots, y_N]\) corresponds to all observed and hidden variables, respectively; \(g\) and \(f\) are the node and edge feature functions; \(h\) is the global feature function; \(Z\) is a normalizing factor.

The factor graph in Figure 2 describes this factorization. Each black box corresponds to a term in the factorization, and it is connected to the variables on which the term depends.

Based on this formulation, the task of social influence is cast as identifying which node has the highest probability to influence another node on a specific topic along with the edge. That is, to maximize the likelihood function \(P(v, Y)\). One parameter configuration is shown in Figure 2. On topic 1, both node \(v_1\) and node \(v_3\) are strongly influenced by node \(v_2\), while node \(v_2\) is mainly influenced by node \(v_4\). On topic 2, the situation is different. Almost all nodes are influenced by node \(v_1\), where node \(v_4\) is indirectly influenced by node \(v_1\) via the node \(v_2\).

3.2 Basic TAP Learning

Baseline: Sum-Product To train the TFG model, we can take Eq. 3 as the objective function to find the parameter configuration that maximizes the objective function. While it is intractable to find the exact solution to Eq. 3, approximate inference algorithms such as sum-product algorithm[17], can be used to infer the variables \(y\).

In sum-product algorithm, messages are passed between nodes and functions. Message passing is initiated at the leaves. Each node \(v_i\) remains idle until messages have arrived on all but one of the edges incident on the node \(v_i\). Once these messages have arrived, node \(v_i\) is able to communicate a message to be sent onto the remaining edge to its neighbor. After sending out a message, node \(v_i\) returns to the idle state, waiting for a “return message” to arrive from the edge. Once this message has arrived, the node is able to compute and send messages to each of neighborhood nodes. This process runs iteratively until convergence.

However, traditional sum-product algorithm cannot be directly applied for multiple topics. We first consider a basic extension of the sum-product algorithm: topical sum-product. The algorithm iteratively updates a vector of messages \(m\) between variable nodes and factor (i.e. feature function) nodes. Hence, two update rules can be defined respectively for a topic-specific message sent from variable node to factor node and for a topic-specific message sent from factor node to variable node.

$$m_{y \rightarrow f}(y, z) = \prod_{y' \sim y} m_{y' \rightarrow y}(y, z) \prod_{y' \sim y} m_{f \rightarrow y}(y, z')(\tau_{y' y})$$

$$m_{f \rightarrow y}(y, z) = \sum_{y' \sim y} f(y, z) \prod_{y' \sim y} m_{y' \rightarrow y}(y', z) + \sum_{y' \sim y} \tau_{y' y} \sum_{y' \sim y} f(y, z') \prod_{y' \sim y} m_{y' \rightarrow y}(y', z')$$

(4)

where

- \(y' \sim y\) represents \(f\)'s neighbor node of variable \(y\) on the factor graph except factor \(f\);
- \(Y\) is a subset of hidden variables that feature function \(f\) is defined on; for example, a feature \(f(y_1, y_3)\) is defined on edge \(e_{13}\), then we have \(Y = \{y_1, y_3\}\); \(\sim \{y\}\) represents all variables in \(Y\) except \(y\);
- the sum \(\sum_{y' \sim y}\) actually corresponds to a marginal function for \(y\) on topic \(z\);
- and coefficient \(\tau\) represents the correlation between topics, which can be defined in many different ways. In this work we, for simplicity, assume that topics are independent. That is, \(\tau_{y' y} = 1\) when \(z = z'\) and \(\tau_{y' y} = 0\) when \(z \neq z'\). In the following, we will propose two new learning algorithms, which are also based this independent assumption.

New Learning Algorithm However, the sum-product algorithm requires that each node need wait for all-(but-one) message to arrive, thus the algorithm can only run in a sequential mode. This results in a high complexity of \(O(N^4 \times T)\) in each iteration. To deal with this problem, 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 \(\{r_{ij}\}_{j=1}^{T}\) and \(\{a_{ij}\}_{j=1}^{T}\) for each edge \(e_{ij}\). The new update rules for the variables are as follows: (Derivation is omitted for brevity.)

$$r_{ij} = \frac{b_{ij}}{a_{ij}} - \operatorname{max}_{k \in N B(j)} \{b_{ik} + a_{ik}\}$$

(5)

$$a_{ij} = \operatorname{max}_{k \in N B(i)} \min \{r_{kj}, 0\}$$

(6)

$$a_{ij} = \min(\max \{r_{ij}, 0\}, -\min \{r_{ij}, 0\})$$

(7)
where $NB(j)$ denotes the neighboring nodes of node $j$, $r_{ij}^z$ is the influence message sent from node $i$ to node $j$ and $a_{ij}^z$ is the influence message sent from node $j$ to node $i$, initiated by 0, and $b_{ij}^z$ is the logarithm of the normalized feature function

$$b_{ij}^z = \log \frac{g(v_i, y_j, z)|y_j = j}{\sum_{k \in NB(i) \cup \{j\}} g(v_i, y_k, z)|y_k = k}$$

(8)

The introduced variables $r$ and $a$ have the following nice explanation. Message $a_{ij}^z$ reflects, from the perspective of node $v_j$, how likely node $v_i$ thinks he/she influences on node $v_j$ with respect to topic $z$, while message $r_{ij}^z$ reflects, from the perspective of node $v_i$, how likely node $v_j$ agrees that node $v_j$ influence on him/her with respect to topic $z$. Finally, we can define the social influence score based on the two variables $r$ and $a$ using a sigmoid function:

$$\mu_{st}^z = \frac{1}{1 + e^{-(r_{st}^z + a_{st}^z)}}$$

(9)

The score $\mu_{st}^z$ actually reflects the maximum of $P(v, y, z)$ for $y_t = s$, thus the maximization of $P(v, y, z)$ can be obtained by

$$y_t^z = \arg \max_{s \in NB(t) \cup \{t\}} \mu_{st}^z$$

(10)

### 3.3 Distributed TAP Learning

As a social network may contain millions of users and hundreds of millions of social ties between users, it is impractical to learn a TFG from such a huge data using a single machine. To address this challenge, we deploy the learning task on a distributed system under the map-reduce programming model [9].

Map-Reduce is a programming model for distributed processing of large data sets. In the map stage, each machine (called a process node) receives a subset of data as input and produces a set of intermediate key/value pairs. In the reduce stage, each process node merges all intermediate values associated with the same intermediate key and outputs the final computation results. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

In our affinity propagation process, 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 $\bar{G}$. 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.

In the map stage, each process node scans the closed graph $\bar{G}$ of the assigned subgraph $G$. Note that every edge $e_{ij}$ has two values $a_{ij}^z$ and $r_{ij}^z$. Thus, the map function is defined as for every key/value pair $e_{ij}/a_{ij}^z$, it issues an intermediate key/value pair $e_{ij}/\{a_{ij}^z + r_{ij}^z\}$; and for key/value pair $e_{ij}/r_{ij}^z$, it issues an intermediate key/value pair $e_{ij}/r_{ij}^z$.

In the reduce stage, each process node collects all values associated with an intermediate key $e_{ij}$ to generate new $a_{ij}^z$ according to Eq. (5), and all intermediate values associated with the same key $e_{ij}$ to generate new $a_{ij}$ according to Eqs. (6) and (7). Thus, the one time map-reduce process corresponds to one iteration in our affinity propagation algorithm.

### 3.4 Model Application

The social influence graphs by TAP can help with many applications. Here we present one application on expert identification, i.e., to identify representative nodes from social networks on a specific topic.

Here we present 3 methods for expert identification: 1) PageRank+LanguageModeling (PR), 2) PageRank with global Influence (PRI), and 3) PageRank with topic-based influence (TPRI).

**Baseline: PR** One 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.

**Proposed 1: PRI** In PRI, we replace the transition probability in PageRank with the influence score. Thus we have

$$r[v] = \beta \frac{1}{|V|} + (1 - \beta) \sum_{v' : v' \prec v} r[v']p(v|v')$$

(11)

In traditional PageRank algorithm, $p(v|v')$ is simply the value of one divides the number of outlinks of node $v'$. Here, we consider the influence score. Specifically we define

$$p(v|v') = \frac{\sum_{z} \mu_{v'v}^z}{\sum_{v, v' \rightarrow v} \sum_{z} \mu_{v'v}^z}$$
Proposed 2: TPRI In the second extension, we introduce, for each node \( v \), a vector of ranking scores \( r[v, z] \), each of which is specific to topic \( z \). Random walk is performed along with the coauthor relationship between authors within the same topic. Thus the topic-based ranking score is defined as:

\[
r[v, z] = \beta \frac{1}{|V|} p(z|v) + (1 - \beta) \sum_{v' \in V} r[v', z] p(v|v', z)
\]

where \( p(z|v) \) is the probability of topic \( z \) generated by node \( v \) and it is obtained from the topic model; \( p(v|v', z) \) represents the probability of node \( v' \) influencing node \( v \) on topic \( z \); we define it as:

\[
p(v|v', z) = \frac{\mu_{v,v'}^{z}}{\sum_{v_j \in V^*} \mu_{v,v_j}^{z}} \mu_{v,v'}^{z}
\]

4. EXPERIMENTAL RESULTS

In this section, we present various experiments to evaluate the efficiency and effectiveness of the proposed approach. All data sets, codes, and tools to visualize the generated influence graphs are publicly available at http://arnetminer.org/lab-datasets/soinfl/.

4.1 Experimental Setup

4.1.1 Data Sets

We perform our experiments on three real-world data sets: two homogeneous networks and one heterogeneous network. The homogeneous networks are academic coauthor network (shortly Coauthor) and paper citation network (shortly Citation). Both are extracted from academic search system Arnetminer\(^1\). The coauthor data set consists of 640,134 authors and 1,554,643 coauthor relations, while the citation data set contains 2,329,760 papers and 12,710,347 citations between these papers. Topic distributions of authors and papers are discovered using a statistical topic modeling approach, Author-Conference-Topic (ACT) model [25]. The ACT approach automatically extracts 200 topics and assigns an author-specific topic distribution to each author and a paper-specific topic distribution to each paper.

The other heterogeneous network is a film-director-actor-writer network (shortly Film), which is crawled from Wikipedia under the category of “English-language films”\(^2\). In total, there are 18,518 films, 7,211 directors, 10,128 actors, and 9,784 writers. There are 142,426 relationships between the heterogeneous nodes in the dataset. The relationship types include: film-director, film-actor, film-writer, and other relationships between actors, directors, and writers. The first three types of relationships are extracted from the “infobox” on the films’ Wiki pages. All the other types of people relationships are created as follows: if one people (including actors, directors, and writers) appears on another people’s page, then a directed relationship is created between them. Topic distributions of the heterogeneous network is initialized using the category information defined on the Wikipedia page. More specifically, we take 10 categories with the highest occurring times as the topics. The 10 categories are: “American film actors”, “American television actors”, “Black and white films”, “Drama films”, “Comedy films”, “British films”, “American film directors”, “Independent films”, “American screenwriters”, and “American stage actors”. As for the topic distribution of each node in the Film network, we first calculate how likely a node \( v_i \) belong to a category (topic) \( z \), i.e., \( p(v_i|z) \), according to \( \frac{1}{V_z} \), where \( |V_z| \) is the number of nodes in the category (topic) \( z \). Thus, for each node, we will obtain a set \( \{p(v_i|z)\}_{z=1}^{Z} \) of likelihood for each node. Then we calculate the topic distribution \( \{p(z|v_i)\}_{z=1}^{Z} \) according to the Bayesian rule \( p(z|v_i) \propto p(z)p(v_i|z) \), where \( p(z) \) is the probability of the category (topic).

4.1.2 Evaluation Measures

For quantitatively evaluate our method, we consider three performance metrics:

- **CPU time.** It is the execution elapsed time of the computation. This determines how efficient our method is.
- **Case study.** We use several case studies to demonstrate how effective our method can identify the topic-based social influence graphs.
- **Application improvement.** We apply the identified topic-based social influence to help expert finding, an important application in social network. This will demonstrate how the quantitative measurement of the social influence can benefit the other social networking application.

The basic learning algorithm is implemented using MATLAB 2007b and all experiments with it are performed on a Server running Windows 2003 with two Dual-Core Intel Xeon processors (3.0 GHz) and 8GB memory. The distributed learning algorithm is implemented under the Map-Reduce programming model using the Hadoop platform\(^3\). We perform the distributed TAP on 6 computer nodes (24 CPU cores) with AMD processors (2.3GHz) and 48GB memory in total. We set the maximum number of iterations as 100 and the threshold for the change of \( r \) and \( a \) to 1e-3. The algorithm can quickly converge after 7-10 iterations in most of the times. In all experiments, for generating each of the topic-based social influence graphs, we only keep 1,000 nodes that have the highest probabilities \( p(v_i|z) \).

4.2 Scalability Performance

We evaluate the efficiency of our approach on the three data sets. We also compare our approach with the sum-product algorithm.

Table 2 lists the CPU time required on the three data sets with the following observations:

**Sum-Product vs TAP** The new TAP approach is much faster than the traditional sum-product algorithm, which even cannot complete on the citation data set.

**Basic vs Distributed TAP** The distributed TAP can typically achieve a significant reduction of the CPU time on the large-scale network. For example, on the citation data set, we obtain a speedup 15X. While on a moderate scaled network (the coauthor data set), the speedup of the distributed TAP is limited, only 3.6. On a relative smaller network (the Film data set), the distributed learning underperforms the basic TAP learning algorithm, which is due to the communication overhead of the Map-Reduce framework.

**Distributed Scalability** We further conduct a scalability experiment with our distributed TAP. We evaluate the speedup of the distributed learning algorithm on the 6 computer nodes using the citation data set with different sizes. It can be seen from Figure 3 (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 TAP like many distributed learning algorithms is good on large-scale data sets.

\(^1\)http://arnetminer.org
\(^3\)http://hadoop.apache.org/
Much effort has been made for social network analysis and a large number of work has been done. For example, methods are proposed for identifying cohesive subgraphs within a network where cohesive subgraphs are defined as “subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties” [26]. 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, Flake et al. [12] propose a method based on maximum flow/minimum cut to identify Web communities. As for social influence analysis, [2, 21] propose methods to qualitatively measure the existence of influence. [6] studies the correlation between social similarity and influence. Other similar
In this paper, we propose a Topical Factor Graph (TFG) model, for quantitatively analyzing the topic-based social influences. We propose a Topical Affinity Propagation (TAP) approach to describe the problem using a graphical probabilistic model. To deal with the efficient problem, we present a new algorithm for training the TFG model. A distributed learning algorithm has been implemented under the Map-reduce programming model. Experimental results on three different types of 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 apply the proposed approach to expert finding. Experiments show that the discovered topic-based influences by the proposed approach can improve the performance of expert finding.

The general problem of network influence analysis represents an new and interesting research direction in social network mining. There are many potential future directions of this work. One interesting issue is to extend the TFG model so that it can learn topic distributions and social influences together. Another issue is to design the TAP approach for (semi-)supervised learning. Users may provide feedbacks to the analysis system. How to make use of the feedbacks to the analysis system. How to make use of the

6. CONCLUSION AND FUTURE WORK

In this paper, we study a novel problem of topic-based social influence analysis. We propose a Topical Affinity Propagation (TAP) approach to describe the problem using a graphical probabilistic model. To deal with the efficient problem, we present a new algorithm for training the TFG model. A distributed learning algorithm has been implemented under the Map-reduce programming model. Experimental results on three different types of 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 apply the proposed approach to expert finding. Experiments show that the discovered topic-based influences by the proposed approach can improve the performance of expert finding.

The general problem of network influence analysis represents an new and interesting research direction in social network mining. There are many potential future directions of this work. One interesting issue is to extend the TFG model so that it can learn topic distributions and social influences together. Another issue is to design the TAP approach for (semi-)supervised learning. Users may provide feedbacks to the analysis system. How to make use of the

5.2 Large-scale Mining

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 [8]. Map-Reduce of several machine learning algorithms on multicore architecture [5].

Recently Papadimitriou and Sun [20] illustrates a mining framework on Map-Reduce along with a case-study using co-clustering.

work can be referred to [10]. To the best of our knowledge, no previous work has been conducted for quantitatively measuring the topic-level social influence on large-scale networks.

For the networking data, graphical probabilistic models are often employed to describe the dependencies between observation data. Markov random field [22], factor graph [17], Restricted Boltzmann Machine(RBM) [27], and many others are widely used graphical models. One relevant work is [13], which proposes an affinity propagation algorithm for clustering by passing messages between data points. The algorithm tries to identify exemplars among data points and forms clusters of data points around these exemplars. In this paper, we propose a Topical Factor Graph (TFG) model, for quantitatively analyzing the topic-based social influences. Compared with the existing work, the TFG can incorporate the correlation between topics. We propose a very efficient algorithm for learning the TFG model. In particular, a distributed learning algorithm has been implemented under the Map-reduce programming model.

Table 4: Representative nodes discovered by our algorithm on the Coauthor data set and the Citation data set.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Topic</th>
<th>Representative Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Mining</td>
<td>Heikki Mannila, Philip S. Yu, Dimitrios Gunopulos, Jiawei Han, Christos Faloutsos, Bing Liu, Vipin Kumar, Tom M. Mitchell, Wei Wang, Qiang Yang, Xindong Wu, Jeffrey Xu Yu, Osmar R. Zaiane</td>
<td></td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Pat Langley, Alex Waibel, Trevor Darrell, C. Lee Giles, Terrence J. Sejnowski, Samy Bengio, Daphne Koller, Luc De Raedt, Vasant Honavar, Floriana Esposito, Bernhard Scholkopf</td>
<td></td>
</tr>
<tr>
<td>Database System</td>
<td>Gerard Venkata, John Mylopoulos, Michael Stonebraker, Barbara Fennici, Philip S. Yu, Shaiard Mehriotis, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han</td>
<td></td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>Gerard Salton, W. Bruce Croft, Ricardo A. Baeza-Yates, James Allan, Yi Zhang, Mouna Lalmas, Zheng Chen, Ophir Frieder, Alan F. Smeaton, Rong Jin</td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td>Yan Wang, Liang-ye Zhang, Schahram Dastidar, Jian Yang, Fabio Casati, Wei Xu, Zharaka Maumair, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah</td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>Wolfgang Nejdl, Daniel Schwabe, Stefan Stab, Mark A. Masek, Andrew Tomkins, Juhana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta</td>
<td></td>
</tr>
<tr>
<td>Bayesian Network</td>
<td>Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Dutien Dubois, David Heckerman, Philippe Smets</td>
<td></td>
</tr>
<tr>
<td>Citation</td>
<td>Data Mining</td>
<td>Fast Algorithms for Mining Association Rules in Large Databases, Using Segmented Right-Deep Trees for the Execution of Pipelined Hash Joins, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Discovery of Multiple-Level Association Rules from Large Databases, Interleaving a Jons Sequence with Semijoins in Distributed Query Processing</td>
</tr>
<tr>
<td>Semantic Web</td>
<td>FaCT and FaCT++: The GRAIL concept modelling language for medical terminology, Semantic Integration of Semistructured and Structured Data Sources, Description of the RACER System and its Applications, DL-Lite: Practical Reasoning for Rich Dls</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Example of influence analysis from the coauthor data set. There are two representative authors and example list of researchers who are mostly influenced by them on topic “data mining”, and their corresponding influenced order on topic “database” and “machine learning”.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Dataset</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Mining</td>
<td>Heikki Mannila, Philip S. Yu, Dimitrios Gunopulos, Jiawei Han, Christos Faloutsos, Bing Liu, Vipin Kumar, Tom M. Mitchell, Wei Wang, Qiang Yang, Xindong Wu, Jeffrey Xu Yu, Osmar R. Zaiane</td>
<td></td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Pat Langley, Alex Waibel, Trevor Darrell, C. Lee Giles, Terrence J. Sejnowski, Samy Bengio, Daphne Koller, Luc De Raedt, Vasant Honavar, Floriana Esposito, Bernhard Scholkopf</td>
<td></td>
</tr>
<tr>
<td>Database System</td>
<td>Gerard Venkata, John Mylopoulos, Michael Stonebraker, Barbara Fennici, Philip S. Yu, Shaiard Mehriotis, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han</td>
<td></td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>Gerard Salton, W. Bruce Croft, Ricardo A. Baeza-Yates, James Allan, Yi Zhang, Mouna Lalmas, Zheng Chen, Ophir Frieder, Alan F. Smeaton, Rong Jin</td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td>Yan Wang, Liang-ye Zhang, Schahram Dastidar, Jian Yang, Fabio Casati, Wei Xu, Zharaka Maumair, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah</td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>Wolfgang Nejdl, Daniel Schwabe, Stefan Stab, Mark A. Masek, Andrew Tomkins, Juhana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta</td>
<td></td>
</tr>
<tr>
<td>Bayesian Network</td>
<td>Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Dutien Dubois, David Heckerman, Philippe Smets</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Citation</th>
<th>Dataset</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Mining</td>
<td>Fast Algorithms for Mining Association Rules in Large Databases, Using Segmented Right-Deep Trees for the Execution of Pipelined Hash Joins, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Discovery of Multiple-Level Association Rules from Large Databases, Interleaving a Jons Sequence with Semijoins in Distributed Query Processing</td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>FaCT and FaCT++: The GRAIL concept modelling language for medical terminology, Semantic Integration of Semistructured and Structured Data Sources, Description of the RACER System and its Applications, DL-Lite: Practical Reasoning for Rich Dls</td>
<td></td>
</tr>
</tbody>
</table>
useful supervised information to improve the analysis quality is an interesting problem. Another potential issue is to apply the proposed approach to other applications (e.g., community discovery) to further validate its effectiveness.

7. ACKNOWLEDGMENTS

The work is supported by the Natural Science Foundation of China (No. 60703059), Chinese National Key Foundation Research (No. 2007CB310803), National High-tech R&D Program (No. 2009AA01Z138), and Chinese Young Faculty Research Fund (No. 20070003093).

8. REFERENCES


Table 6: Example of influence analysis results on topic “data mining” from the citation data set. There are two representative papers and example paper lists that are mostly influenced by them.