Social Influence Analysis in Large-scale Networks

Jie Tang
Dept. of Computer Science
Tsinghua University, China
jietang@tsinghau.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
sonicive@gmail.com
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.

KDD'09, June 28–July 1, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-495-9/09/06 ...$5.00.

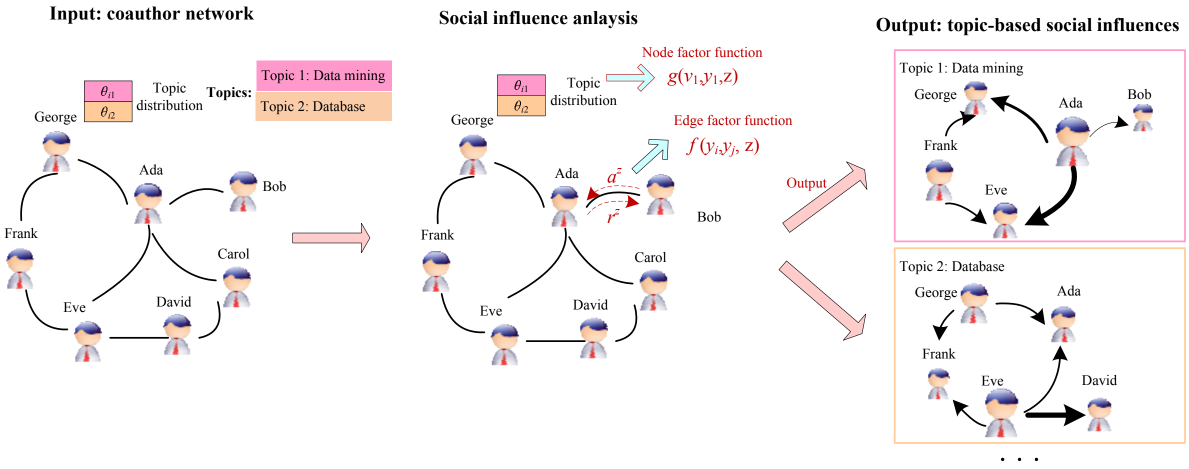


Figure 1: Social influence analysis illustration using the co-author network.

- *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 topic $1 \leq z \leq T$. 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 θ_{vz} is the probability(importance) of the node on topic z.

Topic-based social influences: Social influence from node s to t denoted as μ_{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, μ^z_{st}) (or shortly (e_{st}, μ^z_{st})), where the edge is from node v_s to node v_t with the weight μ^z_{st}.

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 influential 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.

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 millions 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 $\{\mathbf{y}_i\}_{i=1}^N$, which corresponds to the N nodes in the input network. Notations are summarized in table 1.

The hidden vector $\mathbf{y}_i \in \{1, \ldots, N\}^T$ models the topic-level influences from other nodes to node v_i. Each element y_i^z, 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 $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_4\}$. The edges between the hidden nodes indicate the four social relationships in the original network (aka the edges of the input network).

Table 1: Notations.

SYMBOL	DESCRIPTION
N	number of nodes in the social network
M	number of edges in the social network
T	number of topics
V	the set of nodes in the social network
E	the set of edges
v_i	a single node
y_i^z	node-v_i's representative on topic z
$\mathbf{y}_i$	the hidden vector of representatives for all topics on node v_i
θ_i^z	the probability for topic z to be generated by the node v_i
e_{st}	an edge connecting node v_s and node v_t
w_{st}^z	the similarity weight of the edge e_{st} w.r.t. topic z
μ_{st}^z	the social influence of node v_s on node v_t w.r.t. topic z

Feature Functions There are three kinds of feature functions:

- **Node feature function** $g(v_i, \mathbf{y}_i, z)$ is a feature function defined on node v_i specific to topic z.
- **Edge feature function** $f(\mathbf{y}_i, \mathbf{y}_j, z)$ is a feature function defined on the edge of the input network specific to topic z.
- **Global feature function** $h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z)$ 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:

$$g(v_i, \mathbf{y}_i, z) = \begin{cases} \frac{w^z_{iy^z_i}}{\sum_{j \in NB(i)}(w^z_{ij} + w^z_{ji})} & y^z_i \neq i \\ \frac{\sum_{j \in NB(i)} w^z_{ji}}{\sum_{j \in NB(i)}(w^z_{ij} + w^z_{ji})} & y^z_i = i \end{cases} \quad (1)$$

where $NB(i)$ represents the indices of the neighboring nodes of node v_i; $w_{ij}^z = \theta_j^z \alpha_{ij}$ reflects the topical similarity or interaction strength between v_i and v_j, with θ_j^z denoting the importance of node-j to topic z, and α_{ij} denoting the weight of the edge e_{ij}. α_{ij} can be defined by different ways. For example, in a coauthor network, α_{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_{y_i}, then v_{y_i} 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., $f(\mathbf{y}_i, \mathbf{y}_j, z) = 1$ 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.:

$$h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z) = \begin{cases} 0 & \text{if } y_k^z = k \text{ and } y_i^z \neq k \text{ for all } i \neq k \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

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., $y_k^z = k$. And it must be a representative of at least another node v_i, i.e., $\exists y_i^z = k, i \neq k$.

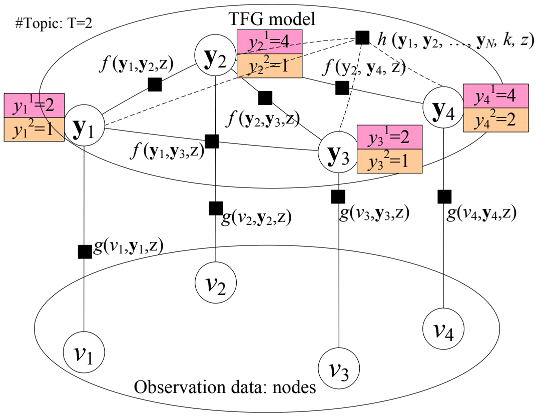


Figure 2: Graphical representation of the topical factor graph model. $\{v_1, \ldots, v_4\}$ are observable nodes in the social network; $\{\mathbf{y}_1, \ldots, \mathbf{y}_4\}$ are hidden vectors defined on all nodes, with each element representing which node has the highest probability to influence the corresponding node; $g(.)$ represents a feature function defined on a node, $f(.)$ represents a feature function defined on an edge; and $h(.)$ represents a global feature function defined for each node, i.e. $k \in \{1, \ldots, N\}$.

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:

$$P(\mathbf{v}, \mathbf{Y}) = \frac{1}{Z} \prod_{k=1}^{N} \prod_{z=1}^{T} h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z)$$
$$\prod_{i=1}^{N} \prod_{z=1}^{T} g(v_i, \mathbf{y}_i, z) \prod_{e_{kl} \in E} \prod_{z=1}^{T} f(\mathbf{y}_k, \mathbf{y}_l, z) \tag{3}$$

where $\mathbf{v} = [v_1, \ldots, v_N]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ 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 $P(\mathbf{v}, \mathbf{Y})$. 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 $\mathbf{y}$.

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 compute a message to be sent onto the one 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 $\mathbf{m}$ 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.

$$\begin{aligned}
m_{y \to f}(y, z) &= \prod_{f' \sim y \backslash f} m_{f' \to y}(y, z) \prod_{z' \neq z} \prod_{f' \sim y \backslash f} m_{f' \to y}(y, z')^{(\tau_{z'z})} \\
m_{f \to y}(y, z) &= \sum_{\sim\{y\}} \left(f(Y, z) \prod_{y' \sim f \backslash y} m_{y' \to f}(y', z) \right) \\
&\quad + \sum_{z' \neq z} \tau_{z'z} \sum_{\sim\{y\}} \left(f(Y, z') \prod_{y' \sim f \backslash y} m_{y' \to f}(y', z') \right)
\end{aligned} \tag{4}$$

where

- $f' \sim y \backslash f$ represents f' is a 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 $f(y_i, y_j)$ is defined on edge e_{ij}, then we have $Y = \{y_i, y_j\}$; $\sim \{y\}$ represents all variables in Y except y;
- the sum $\sum_{\sim\{y\}}$ actually corresponds to a marginal function for y on topic z;
- and coefficient τ 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, $\tau_{zz'} = 1$ when $z = z'$ and $\tau_{zz'} = 0$ 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 $\{r_{ij}^z\}_{z=1}^T$ and $\{a_{ij}^z\}_{z=1}^T$ for each edge e_{ij}. The new update rules for the variables are as follows: (Derivation is omitted for brevity.)

$$r_{ij}^z = b_{ij}^z - \max_{k \in NB(j)} \{b_{ik}^z + a_{ik}^z\} \tag{5}$$

$$a_{jj}^z = \max_{k \in NB(j)} \min \{r_{kj}^z, 0\} \tag{6}$$

$$\begin{aligned}
a_{ij}^z = {} & \min(\max \{r_{jj}^z, 0\}, -\min \{r_{jj}^z, 0\} \\
& - \max_{k \in NB(j) \backslash \{i\}} \min \{r_{kj}^z, 0\}), i \in NB(j)
\end{aligned} \tag{7}$$

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

$$b_{ij}^z = \log \frac{g(v_i, \mathbf{y}_i, z)|_{y_i^z=j}}{\sum_{k\in NB(i)\cup\{i\}} g(v_i, \mathbf{y}_i, z)|_{y_i^z=k}} \tag{8}$$

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_j thinks he/she influences on node v_i with respect to topic z, while message r_{ij}^z reflects, from the perspective of node v_i, how likely node v_i 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:

$$\mu_{st}^z = \frac{1}{1+e^{-(r_{ts}^z + a_{ts}^z)}} \tag{9}$$

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

$$y_t^z = arg \max_{s\in NB(t)\cup\{t\}} \mu_{st}^z \tag{10}$$

```
Input: $G = (V, E)$ and topic distributions $\{\theta_v\}_{v\in V}$
Output: topic-level social influence graphs $\{G_z = (V_z, E_z)\}_{z=1}^T$
Calculate the node feature function $g(v_i, \mathbf{y}_i, z)$;
Calculate $b_{ij}^z$ according to Eq. 8;
Initialize all $\{r_{ij}^z\} \leftarrow 0$;
repeat
    foreach edge-topic pair $(e_{ij}, z)$ do
        Update $r_{ij}^z$ according to Eq. 5;
    end
    foreach node-topic pair $(v_j, z)$ do
        Update $a_{jj}^z$ according to Eq. 6;
    end
    foreach edge-topic pair $(e_{ij}, z)$ do
        Update $a_{ij}^z$ according to Eq. 7;
    end
until convergence;
foreach node $v_t$ do
    foreach neighboring node $s \in NB(t) \cup \{t\}$ do
        Compute $\mu_{st}^z$ according to Eq. 9;
    end
end
Generate $G_z = (V_z, E_z)$ for every topic $z$ according to $\{\mu_{st}^z\}$;
```

Algorithm 1: The new TAP learning algorithm.

Finally, according to the obtained influence scores $\{\mu_{st}^z\}$ and the topic distribution $\{\theta_v\}$, we can easily generate the topic-level social influence graphs. Specifically, for each topic z, we first filter out irrelevant nodes, i.e., nodes that have a lower probability than a predefined threshold. An alternative way is to keep only a fixed number (e.g., 1,000) of nodes for each topic-based social influence graph. (This filtering process can be also taken as a preprocessing step of our approach, which is the way we conducted our experiments.) Then, for a pair of nodes (v_s, v_t) that has an edge in the original network G, we create two directed edges between the two nodes and respectively assign the social influence scores μ_{st}^z and μ_{ts}^z. Finally, we obtain a directed social influence graph G_z for the topic z.

The new algorithm reduces the complexity of each iteration from $O(N^4 \times T)$ in the sum-product algorithm to $O(M \times T)$. More importantly, the new update rules can be easily parallelized.

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}. Thus, the map function is defined as for every key/value pair e_{ij}/a_{ij}, it issues an intermediate key/value pair $e_{i*}/(b_{ij} + a_{ij})$; and for key/value pair e_{ij}/r_{ij}, it issues an intermediate key/value pair e_{*j}/r_{ij}.

In the reduce stage, each process node collects all values associated with an intermediate key e_{i*} to generate new r_{i*} according to Eq. (5), and all intermediate values associated with the same key e_{*j} to generate new a_{*j} 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 illustrate 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

$$r[v] = \beta \frac{1}{|V|} + (1-\beta) \sum_{v':v'\to v} r[v']p(v|v') \tag{11}$$

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

$$p(v|v') = \frac{\sum_z \mu_{v'v}^z}{\sum_{v_j:v'\to v_j} \sum_z \mu_{v'v_j}^z}$$

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:

$$r[v, z] = \beta \frac{1}{|V|} p(z_k|v) + (1 - \beta) \sum_{v':v' \to v} r[v', z] p(v|v', z) \quad (12)$$

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

$$p(v|v', z) = \frac{\mu^z_{v'v}}{\sum_{v_j:v' \to v_j} \mu^z_{v'v_j}}$$

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/soinf/.

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 $\{p(v_i|z)\}_{z=1}^T$ of likelihood for each node. Then we calculate the topic distribution $\{p(z|v_i)\}_{z=1}^T$ 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 train 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 $p(v|z)$.

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.

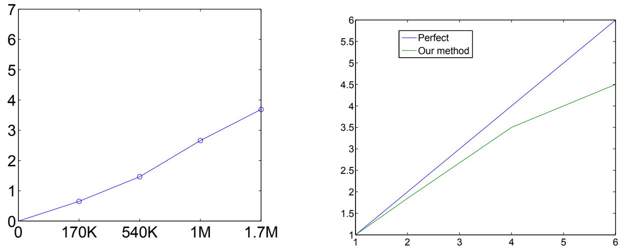
[1] http://arnetminer.org

[2] `http://en.wikipedia.org/wiki/Category:English-language_films`

[3] `http://hadoop.apache.org/`

Table 2: Scalability performance of different methods on real data sets. >10hr means that the algorithm did not terminate when the algorithm runs more than 10 hours.

Methods	Citation	Coauthor	Film
Sum-Product	N/A	>10hr	1.8 hr
Basic TAP Learning	>10hr	369s	**57s**
Distributed TAP Learning	**39.33m**	**104s**	148s



(a) Dataset size vs. speedup (b) #Computer nodes vs. speedup

Table 3: Speedup results.

Using our large citation data set, we also perform speedup experiments on a Hadoop platform using 1, 2, 4, 6 computer nodes (since we did not have access to a large number of computer nodes). The speedup, shown in 3 (b), show reasonable parallel efficiency, with a $> 4\times$ speedup using 6 computer nodes.

4.3 Qualitative Case Study

Now we demonstrate the effectiveness of TAP of representative nodes identification on the Coauthor and Citation data sets.

Table 4 shows representative nodes (authors and papers) found by our algorithm on different topics from the coauthor data set and the citation data set. The representative score of each node is the probability of the node influencing the other nodes on this topic. The probability is calculated by $\frac{\sum_{j\in NB(i)\cup\{i\}}\mu_{ij}}{\sum_{i=1}^{N}\sum_{j\in NB(i)\cup\{j\}}\mu_{ij}}$. We can see some interesting results. For example, some papers (e.g., "FaCT and iFaCT") that do have have a high citation number might be selected as the representative nodes. This is because our algorithm can identify the influences between papers, thus can differentiate the citations of the theoretical background of a paper and an odd citation in the reference.

Table 5 shows four representative authors and researchers who are mostly influenced by them. Table 6 shows two representative papers and papers that are mostly influence by the two papers. Some other method e.g., the similarity-based baseline method using cosine metric, can be also used to estimate the influence according to the similarity score. Such a method was previously used for analyzing the social influence in online communities [6]. Comparing with the similarity-based baseline method, our method has several distinct advantages: First, such a method can only measure the similarity between nodes, but cannot tell which node has a stronger influence on the other one. Second, the method cannot tell which nodes have the highest influences in the network, which our approach naturally has the capacity to do this. This provides many immediate applications, for example, expert finding.

4.4 Quantitative Case Study

Now we conduct quantitatively evaluation of the effectiveness of the topic-based social influence analysis through case study. Recall the goal of expert finding is to identify persons with some expertise or experience on a specific topic (query) q. We define the baseline method as the combination [24] of the language model $P(q|v)$ and PageRank $r[v]$.

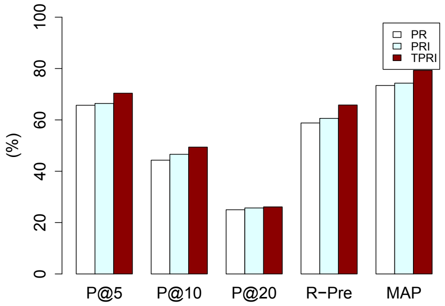


Table 7: Performance of expert finding with different approaches.

We use an academic data set used in [24] [25] for the experiments. Specifically, the data set contains $14,134$ authors, $10,716$ papers, and $1,434$ conferences. Four-grade scores (3, 2, 1, and 0) are manually labeled to represent definite expertise, expertise, marginal expertise, and no expertise. Using this data, we create a coauthor network. The topic model for each author is still obtained using the statistical topic modeling approach [25]. With the topic models, we apply the proposed TAP approach to the coauthor network to identify the topic-based influences.

With the learned topic-based influence scores, we define two extensions to the PageRank method: PageRank with Influence (PRI) and PageRank with topic-based influence (TPRI). Details of the extension is described in Section 3.4. For expert finding, we can further combine the extended PageRank model with the relevance model, for example the language model by $P(q|v)r[v]$ or a topic-based relevance model by $\sum_z p(q|z)p(z|v)r[v,z]$, where $r[v]$ and $r[v,z]$ are obtained respectively from PRI and TPRI; $p(q|z)$, $p(z|v)$ can be obtained from the statistical topic model [24].

We evaluate the performance of different methods in terms of Precision@5 (P@5), P@10, P@20, R-precision (R-Pre), and mean average precision (MAP) [4, 7]. Figure 7 shows the result of expert finding with different approaches. We see that the topic-based social influences discovered by the TAP approach can indeed improve the accuracy of expert finding, which confirms the effectiveness of the proposed approach for topic-based social influence analysis.

5. RELATED WORK

5.1 Social Network and Influence

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/minmum 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

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

Dataset	Topic	Representative Nodes
Author	Data Mining	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
	Machine Learning	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
	Database System	Gerhard Weikum, John Mylopoulos, Michael Stonebraker, Barbara Pernici, Philip S. Yu, Sharad Mehrotra, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han
	Information Retrieval	Gerard Salton, W. Bruce Croft, Ricardo A. Baeza-Yates, James Allan, Yi Zhang, Mounia Lalmas, Zheng Chen, Ophir Frieder, Alan F. Smeaton, Rong Jin
	Web Services	Yan Wang, Liang-jie Zhang, Schahram Dustdar, Jian Yang, Fabio Casati, Wei Xu, Zakaria Maamar, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah
	Semantic Web	Wolfgang Nejdl, Daniel Schwabe, Steffen Staab, Mark A. Musen, Andrew Tomkins, Juliana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta
	Bayesian Network	Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Didier Dubois, David Heckerman, Philippe Smets
Citation	Data Mining	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 Join Sequence with Semijoins in Distributed Query Processing
	Machine Learning	Object Recognition with Gradient-Based Learning, Correctness of Local Probability Propagation in Graphical Models with Loops, A Learning Theorem for Networks at Detailed Stochastic Equilibrium, The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length, A Unifying Review of Linear Gaussian Models
	Database System	Mediators in the Architecture of Future Information Systems, Database Techniques for the World-Wide Web: A Survey, The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles, Fast Algorithms for Mining Association Rules in Large Databases
	Web Services	The Web Service Modeling Framework WSMF, Interval Timed Coloured Petri Nets and their Analysis, The design and implementation of real-time schedulers in RED-linux, The Self-Serv Environment for Web Services Composition
	Web Mining	Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Fast Algorithms for Mining Association Rules in Large Databases, The OO-Binary Relationship Model: A Truly Object Oriented Conceptual Model, Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterizations, Improving Fault Tolerance and Supporting Partial Writes in Structured Coterie Protocols for Replicated Objects
	Semantic Web	FaCT and iFaCT, 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

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".

Topic: Data Mining		Topic: Database		Topic: Machine Learning	
Jiawei Han	Heikki Mannila	Jiawei Han	Heikki Mannila	Jiawei Han	Heikki Mannila
David Clutter	Arianna Gallo	David Clutter	Vladimir Estivill-Castro	Hasan M. Jamil	Vladimir Estivill-Castro
Hasan M. Jamil	Marcel Holsheimer	Shiwei Tang	Marcel Holsheimer	K. P. Unnikrishnan	Marcel Holsheimer
K. P. Unnikrishnan	Robert Gwadera	Hasan M. Jamil	Robert Gwadera	Shiwei Tang	Mika Klemettinen
Ramasamy Uthurusamy	Vladimir Estivill-Castro	Ramasamy Uthurusamy	Mika Klemettinen	Ramasamy Uthurusamy	Robert Gwadera
Shiwei Tang	Mika Klemettinen	K. P. Unnikrishnan	Arianna Gallo	David Clutter	Arianna Gallo

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.

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.

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

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.

Fast Algorithms for Mining Association Rules in Large Databases	Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data
Mining Large Itemsets for Association Rules	Mining Web Site?s Clusters from Link Topology and Site Hierarchy
A New Framework For Itemset Generation	Predictive Algorithms for Browser Support of Habitual User Activities on the Web
Efficient Mining of Partial Periodic Patterns in Time Series Database	A Fine Grained Heuristic to Capture Web Navigation Patterns
A New Method for Similarity Indexing of Market Basket Data	A Road Map to More Effective Web Personalization: Integrating Domain Knowledge with Web Usage Mining
A General Incremental Technique for Maintaining Discovered Association Rules	

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

[1] R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 2002.

[2] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’08)*, pages 7–15, 2008.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR’04*, pages 25–32, 2004.

[5] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun. Map-Reduce for machine learning on multicore. In *Proceedings of the 18th Neural Information Processing Systems (NIPS’06)*, 2006.

[6] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’08)*, pages 160–168, 2008.

[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec-2005 enterprise track. In *TREC 2005 Conference Notebook*, pages 199–205, 2005.

[8] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *Proceeding of the 16th international conference on World Wide Web (WWW’07)*, 2007.

[9] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation (OSDI’04)*, pages 10–10, 2004.

[10] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *WWW’07*, pages 461–470, 2007.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, New York, NY, USA, 1999. ACM.

[12] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’00)*, pages 150–160, 2000.

[13] B. J. Frey and D. Dueck. Mixture modeling by affinity propagation. In *Proceedings of the 18th Neural Information Processing Systems (NIPS’06)*, pages 379–386, 2006.

[14] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[15] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR’99)*, pages 50–57, 1999.

[16] D. Krackhardt. *The Strength of Strong ties: the importance of philos in networks and organization in Book of Nitin Nohria and Robert G. Eccles (Ed.), Networks and Organizations*. Harvard Business School Press, Boston, MA, 1992.

[17] F. R. Kschischang, S. Member, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[18] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th International World Wide Web Conference (WWW’08)*, pages 101–110, 2008.

[19] M. E. J. Newman. The structure and function of complex networks. *SIAM Reviews*, 45, 2003.

[20] S. Papadimitriou and J. Sun. Disco: Distributed co-clustering with map-reduce. In *Proceedings of IEEE International Conference on Data Mining (ICDM’08)*, 2008.

[21] P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *Proceeding of the 17th international conference on World Wide Web (WWW’08)*, pages 655–664, 2008.

[22] P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. pages 194–281, 1986.

[23] S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2003.

[24] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *Proceedings of IEEE International Conference on Data Mining (ICDM’08)*, pages 1055–1060, 2008.

[25] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD’08)*, pages 990–998, 2008.

[26] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press, 1994.

[27] M. Welling and G. E. Hinton. A new learning algorithm for mean field boltzmann machines. In *Proceedings of International Conference on Artificial Neural Network (ICANN’01)*, pages 351–357, 2001.