Confluence: Conformity Influence in Large Social Networks

Jie Tang
Computer Science
Tsinghua University, China
Beijing 100084, China
jietang@tsinghua.edu.cn

Sen Wu
Computer Science
Tsinghua University, China
Beijing 100084, China
ronaldosen@gmail.com

Jimeng Sun
IBM T. J. Watson Research Center
USA
jimeng@us.ibm.com

ABSTRACT

Conformity is a type of social influence involving a change in opinion or behavior in order to fit in with a group. Employing several social networks as the source for our experimental data, we study how the effect of conformity plays a role in changing users' online behavior. We formally define several major types of conformity in individual, peer, and group levels. We propose *Confluence* model to formalize the effects of social conformity into a probabilistic model. Confluence can distinguish and quantify the effects of the different types of conformities. To scale up to large social networks, we propose a distributed learning method that can construct the Confluence model efficiently with near-linear speedup.

Our experimental results on four different types of large social networks, i.e., Flickr, Gowalla, Weibo and Co-Author, verify the existence of the conformity phenomena. Leveraging the conformity information, Confluence can accurately predict actions of users. Our experiments show that Confluence significantly improves the prediction accuracy by up to 5-10% compared with several alternative methods.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Text Mining

General Terms

Algorithms, Experimentation

Keywords

Conformity; Social influence; Social network

1. INTRODUCTION

Conformity is the act of matching attitudes, beliefs, and behaviors to group norms [8]. The phenomenon of conformity could occur in small groups or the whole society, as a resultant of peer influence or group pressure. Conformity can have either good or bad effect depending on the situation. For example, it helps form and maintain the social norms, and helps prevent acts that are perceptually dangerous. Conformity can be influenced by various factors such as individual status, peer influence and group pressure. Therefore there is a clear need for quantitative methods for measuring conformity from different aspects, so as to understand the complex dynamics in social networks.

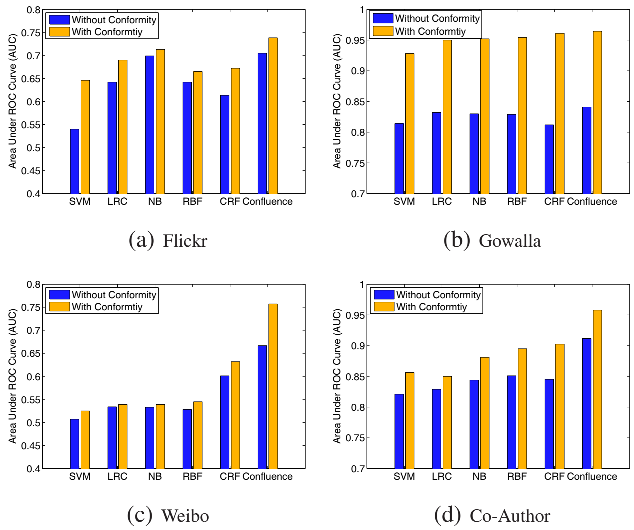
Conformity was first studied by psychologists through interviews with small groups of participants [17]. In economics, Bernheim [3] found that sometimes people are willing to conform simply because they recognize that departure from the social norm may impair their status. Bernheim further proposed a theory of social conformity and presented a model to describe the conformity process. However, due to the lack of real data, he only studied the model from the theoretical aspect. With the rapid proliferation of online social networks such as Facebook, Twitter, and Flickr, it becomes feasible and also very necessary to conduct an in-depth investigation of the conformity problem on real large social networks. In practice, the effect of conformity has been also observed in online social networks. For example, Bond et al. [4] reported results from a randomized controlled trial of political mobilization messages delivered to 61 million Facebook users. They found that when one is aware that their friends have made the political votes, their likelihood to vote will significantly increase. Bakshy et al. [2] also found that when their friends click an ad, they will be more likely to click the same ad.

From a broader viewpoint, conformity can be seen as a special type of social influence. There are a bulk of studies on social influence analysis. These studies can be roughly classified into three categories: influence testing [1, 9, 20], influence quantification [12, 13, 23, 32], and influence maximization models [6, 18]. However, most of the works focus on qualitative study of social influence. With an exception, Tang et al. [32] presented a Topical Affinity Propagation (TAP) approach to quantify the topic-level social influence in large networks. However, they do not distinguish the effect of peer influence and group conformity.

There are several challenges for the conformity influence analysis. First, how to formally define and differentiate different types of conformities? Unlike peer influence, which mainly considers how two connected friends influence each other, conformity occurs in different situations and exists with different forms. Second, how to construct a computational model to learn the different conformity factors? Third, how to validate the proposed model in real large networks.

To address the above challenges, we formally define the problem of conformity influence analysis in social networks and categorize conformity into individual conformity, peer conformity, and group conformity. We propose an Confluence method to formalize the effects of social conformity into a probabilistic factor graph model.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.



(a) Flickr (b) Gowalla

(c) Weibo (d) Co-Author

Figure 1: Action prediction accuracy (AUC) of different methods by considering the effect of conformity in four networks.

The Confluence method can distinguish and quantify the effect of the different types of conformities. We test the proposed method on four social networks, i.e., Flickr, Gowalla, Weibo[1], and Co-Author. Our experimental analysis verifies the existence of conformity in all the four social networks. We further apply the Confluence method to model and predict users' online behavior. To scale up to large social networks, we propose a distributed learning version of Confluence that can leverage parallel computing to significantly reduce the computational time to nearly linear speedup ($\sim 9\times$ with 16 computer cores). Figure 1 shows the prediction accuracy in terms of Area Under Curve (AUC) on the four networks. It can be clearly seen that by incorporating the conformity factors, the prediction performance can be significantly improved up to 5-10% compared to various baseline methods.

Organization Section 2 formulates the problem; Section 3 explains the proposed model and describes the algorithm for learning the model; Section 4 presents the experimental results; finally, Section 5 discusses related work and Section 6 concludes the work.

2. PROBLEM DEFINITION

Let $G = (V, E)$ denote the social network, where V is a set of $|V| = N$ users and $E \subset V \times V$ is a set of relationships between users. A user's behavior is time dependent. Specifically, we use the triple (a, v_i, t) to represent user v_i performed action $a \in \mathcal{A}$ at time t, where $\mathcal{A}$ is the set of possible actions. For example, on Flickr, the action can be defined as adding comment to a specific picture. We further assume that users in a network form m groups and use an $N \times m$ matrix $\mathbf{C}$ to represent users' group memberships. Specifically, each binary-valued element $c_{ik} \in \mathbf{C}$ represents whether user v_i belongs to the k^{th} group. We use C_k to represent the k^{th} group. The group can be formed through user participation. For example, users on Flickr can build and join different groups. If a network does not have the explicit group information, we use a community detection algorithm (e.g., [27]) to automatically detect groups from the network structure. In addition, each user v_i is associated with a set of d numeric attributes $\mathbf{x}_i$. The attributes can be defined based on users' profile (e.g., interest or posted tweets). Given this, we can define the input of our problem as follows. (Table 1 summarizes the notations used throughout this paper.)

[1]Weibo.com, the most popular microblogging service in China with more than 400 million users.

Table 1: Notations.

SYMBOL	DESCRIPTION
V	a set of $\|V\| = N$ users
E	a set of relationships between users
$\mathbf{C}$	a $N \times m$ matrix to represent users' group memberships
$\mathbf{X}$	a $N \times d$ matrix to represent users' attributes
A	the action history of all users in the network
$c_{ij} \in \mathbf{C}$	a binary-valued variable to represent whether user v_i belongs to group C_k
$x_{ij} \in \mathbf{X}$	the j^{th} attribute of user v_i
$(a, v_i, t) \in A$	a triple to represent user v_i performed the action a at time t

Input: The input of our problem consists of two components, i.e., an attribute augmented network $G = (V, E, \mathbf{C}, \mathbf{X})$ and action history $A = \{(a, v_i, t)\}_{a,i,t}$, where $\mathbf{X}$ denotes an $N \times d$ attribute matrix with an element x_{ij} indicating the j^{th} attribute of user v_i.

Our goal is to study how a user's behavior conforms to her peer friends and the communities (groups) that she belongs to. In this work, we define three levels of conformities respectively from the aspect of individual, peer relationship, and group. The individual conformity represents how easily user v's behavior conforms to her friends. Formally, we have

Definition 1. **Individual conformity:** The individual conformity is defined as the ratio between the number of actions for which we have evidence that the user v conforms to one of her friends v', over the total number of actions performed by user v. More precisely we define:

$$icf(v) = \frac{|(a, v, t) \in A_v | \exists (a, v', t') : e_{vv'} \in E \wedge \epsilon \geq t - t' \geq 0|}{|A_v|}$$

where $A_v \subset A$ denotes the action history of user v and ϵ is a threshold of difference between the time when the two users v and v' performed the same action a, and $|\cdot|$ denotes the cardinality over a set.

We also define peer conformity to represent how likely the user v's behavior is influenced by one particular friend v'.

Definition 2. **Peer conformity:** The peer conformity is defined as the ratio between the number of actions for which we have evidence that the user v conforms to her friend v', over the total number of actions performed by the friend v', that is:

$$pcf(v, v') = \frac{|(a, v', t') \in A_{v'} | \exists (a, v, t) : e_{vv'} \in E \wedge \epsilon \geq t - t' \geq 0|}{|A_{v'}|}$$

where $A_{v'} \subset A$ denotes the action history of user v'.

We further define group conformity to represent the conformity of user v's behavior to groups that the user belongs to. In a group, there might be a large number of actions performed by its users. However many actions may be performed by only one single user. To begin with, we first define the τ-group action as the action that was performed by more than a percentage τ of all users in the community (group) C_k. Given this, we define the group conformity as follows:

Definition 3. **Group conformity:** The group conformity is then defined as the ratio between the number of actions for which we have evidence that the user v conforms to the group, over the total number of τ-group actions performed by users in the group C_k,

$$gcf^\tau(v, C_{vk}) = \frac{|(a, v', t') \in A^\tau_{C_k} | \exists(a, v, t) : \mathbb{I}[c_{ik}] \wedge \epsilon \geq t - t' \geq 0|}{|A^\tau_{C_k}|}$$

where $A^\tau_{C_k} \subset A$ denotes actions performed by more than a percentage τ of all users in the group C_k; $\mathbb{I}[c_{ik}]$ is an indicator function, which returns true if the value of c_{ik} is 1 and false otherwise.

Please note that a user may be involved into more than one groups, thus has different conformity degrees in the different groups. The above definitions quantify the conformity from different levels (individual, peer relationship, and group). Further, given the action history $A = \{(a, v_i, t)\}_{a,i,t}$, we use variable $y_i = a$ to indicate whether user v_i performs the an action a and use the collection of variables $Y^t = \{y_i\}_{1,\cdots,N}$ to represent the action labels of all users at time t. Next, we define the problem of conformity influence analysis.

Problem: Given 1) an attribute augmented network $G = (V, E, \mathbf{C}, \mathbf{X})$ and 2) action history $A = \{(a, v_i, t)\}_{a,i,t}$, how to quantify the importance of the different types of conformities for each user? This is formalized as finding a model parameters $\theta^\star$ of different conformities to maximize the following conditional probability, i.e., $\theta^\star = \arg\max_\theta P_\theta(Y^t|G, A)$. The second problem is how to incorporate the defined conformities and the learned model parameters into a unified model to predict users' future action in the social network, i.e., $Y^\star = \arg\max_{Y^{t+1}} P_{\theta^\star}(Y^{t+1}|G, A)$.

3. CONFLUENCE MODEL FRAMEWORK

Our goal is to design a unified model to capture users' action dynamics and model conformities from different levels. We propose Confluence, a conformity-aware factor graph model. To handle real large networks, we develop a distributed model learning algorithm.

3.1 Conformity-aware Factor Graph Model

In the Confluence model, we attempt to maximize the conditional probability of user actions given their corresponding attributes and the input network, i.e., $P_\theta(Y^t|G, A)$. More precisely, for each action $(a, v_i, t) \in A$, we construct a training instance. Then learning the model becomes how to find a configuration of parameters θ to maximize the joint conditional probability for all users' actions. When applying the learned model parameters to predict users' future actions, it tries to find a setting of action labels Y^{t+1} at time $t + 1$ to maximize the conditional probability $P_\theta(Y^{t+1}|G, A)$ based on the learned parameters.

Directly maximizing the conditional probability $P(Y|G, A)$ is often intractable. Factor graph provides a method to factorize the "global" probability as a product of "local" factor functions, each of which depends on a subset of the variables in the graph [19]. In the proposed Confluence model, we try to capture two kinds of information, i.e., the attributes associated with each user and three types of conformities we defined in § 2. Specifically, we use three factor functions to represent the individual conformity, peer conformity, and group conformity, respectively.

- **Individual conformity factor:** $g(y_i, icf(v_i))$ represents the correlation between user v_i's action and his individual conformity.
- **Peer conformity factor:** $g(y_i, y'_j, pcf(v_i, v_j))$ represents the correlation between user v_i's action and his peer conformity to v_j.

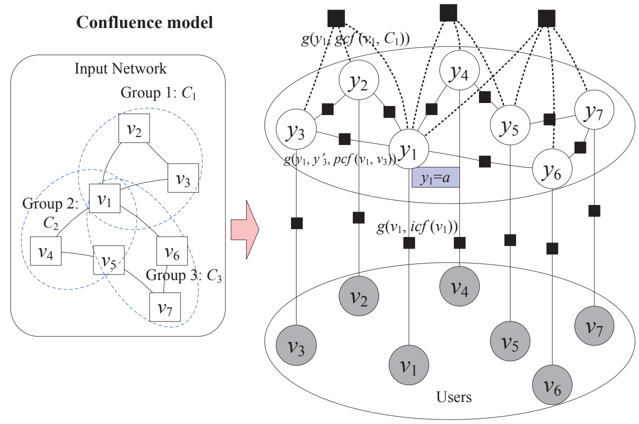


Figure 2: Graphical representation of the Confluence model. y_i indicates the corresponding action of user v_i; $g(y_i, icf(v_i))$, $g(y_i, y'_j, pcf(v_i, v_j))$, and $g(y_i, gcf^\tau(v_i, C_k))$ respectively represent the individual conformity factor, peer conformity factor, and group conformity factor.

- **Group conformity factor:** $g(y_i, gcf^\tau(v_i, C_k))$ represents the correlation between user v_i's action and his conformity to the group C_k.

In the above definitions, without ambiguity, we omit the time stamp ^{t} (e.g., simplifying y_i^t to y_i) and use the superscript $'$ (e.g., simplifying $y_i^{t'}$ to y'_i) to indicate a variable at time stamp t'.

The different factors quantify how different levels of conformities finally determine user v's actions. According to the defined correlations, we can construct the graphical structure in the Confluence model. An example is illustrated in Figure 2. In the input network, user v_1 belongs to three groups C_1, C_2, and C_3, thus in the constructed factor graph, its corresponding latent variable y_1 is connected to three group conformity factors (e.g., $g(y_1, gcf(v_1, C_1))$). User v_1 has four friends in the input network, thus four peer conformity factors in the factor graph model. We also define an individual conformity factor for each user. Besides the conformity factors, we use factor function $f(y_i, x_{ij})$ to capture the correlation between the user's attribute x_{ij} and user's action. By integrating all the factor functions together, and according to the Hammersley-Clifford theorem [15] we can obtain the following log-likelihood objective function,

$$\begin{aligned}
\mathcal{O}(\theta) &= \log P_\theta(Y|G, A) \\
&= \sum_{i=1}^N \left[\sum_{j=1}^d \alpha_j f(y_i, x_{ij}) + \beta_i g(y_i, icf(v_i)) \right] \\
&\quad + \sum_{e_{ij} \in E} \mathbb{I}[y'] \gamma_{ij} g(y_i, y'_j, pcf(v_i, v_j)) \\
&\quad + \sum_{i=1}^N \sum_{k=1}^m \mathbb{I}[c_{ik}] \mu_{ik} g(y_i, gcf(v_i, C_k)) - \log Z
\end{aligned} \tag{1}$$

where $\mathbb{I}[y'_j]$ is an indicator function to indicate whether user v_j performed the same action immediately before user v_i, more precisely, $[\exists(a, v_j, t') \wedge \epsilon \geq t - t' \geq 0]$ and $\mathbb{I}[c_{ik}]$ indicates whether user v_i belongs to the group C_k; α, β, γ, and μ are respectively weights of the different factor functions; $\theta = (\{\alpha\}, \{\beta\}, \{\gamma\}, \{\mu\})$ is a parameter configuration estimated from the training data; and Z is a normalization factor to ensure that the distribution is normalized so that the sum of the probabilities equals to 1.

In practice, choosing a good threshold ϵ for defining the conformity factors is challenging. Instead, we use a decay factor $\lambda \geq 1$ in each conformity function. A large λ means a slow-decay effect. Accordingly, the peer conformity factor is defined as:

$$g(y_i, y_j', pcf(v_i, v_j)) = (\frac{1}{2})^{\frac{t-t'}{\lambda}} pcf(v_i, v_j) \tag{2}$$

where t' corresponds to the latest past time when v_j performed the same action as v_i in the training data set. The decay factor decays the peer conformity exponentially over time, with the half-life, λ, serving as a tunable parameter. The basic idea is that friend's recent actions have higher influence on one's action. The group conformity factor is defined as:

$$g(y_i, gcf^{\tau}(v_i, C_k)) = (\frac{1}{2})^{\frac{t-t'}{\lambda}} gcf^{\tau}(v_i, C_k) \tag{3}$$

where t' represents the latest past time when some user in group C_k performed the same action as user v_i in the training data set; and τ is empirically set as 0.25. Please note that a user may belong to multiple groups and accordingly we could incorporate multiple different group community factors into the objective function. Similarly, the individual conformity factor is defined as:

$$g(y_i, icf(v_i)) = \frac{\sum_{k=1}^{|A_{v_i}|} (\frac{1}{2})^{\frac{t-t'}{\lambda}} \mathbb{I}[y_j' \wedge e_{ij} \in E]}{|A_v|} \tag{4}$$

where $\mathbb{I}[y_j' \wedge e_{ij} \in E]$ represents whether some friend v_j of v_i performed the same action $a_k \in A_{v_i}$ immediately before user v_i.

Implementation note. In our experiments, we empirically set the parameters within the factor functions as follows: $\lambda = 2$, $\epsilon = 1$, and $\tau = 0.25$. We did try different parameter values. For example, for λ, we fixed all the other parameters and varied λ from 1 to 10 with an interval 1. We varied ϵ between $[1, 5]$. For τ, we varied between $[0.0, 1.0]$ with an interval 0.05. We tested the model accuracy (in terms of F1-score) for each parameter value and found that the performance was relatively stable across different settings. As for the community detection, we tried different algorithms such as local spectral partitioning (LSP), METIS, and Newman. We found that the prediction performance is also not very sensitive to the choice of the community detection algorithm. Finally, we use the Newman algorithm (e.g., [27]) due to its wide adoption.

3.2 Feature Definition

Besides the conformity features, we also define other features for modeling users' actions in the networks. The first type of features are based on users' attributes including the number of friends and users' interests. More specifically, to make the defined features as general as possible, we simply consider four attribute features, i.e., the number of friends, the number of new friends in the recent three time stamps, the number of total groups that the user joined, and the number of groups the user joined in recent three time stamps. The proposed model is also general and can incorporate other social theory-based features. Thus, the second type of features are defined based on some other social theories. Specifically,

Opinion leader: the feature is defined to represent whether the user is an opinion leader or not. We use the InfluenceRank [29], a PageRank-like algorithm, to rank and select opinion leaders in a network.

Input: network G, action history A, and learning rate η;
Output: learned parameters $\theta = (\{\alpha\}, \{\beta\}, \{\gamma\}, \{\mu\})$;

Initialize $\alpha, \beta, \gamma, \mu$;
Construct the graphical structure G in the Confluence model;
Partition the graph G into M subgraphs $[G_1, \cdots, G_M]$;
repeat
  %Distribute the parameter to calculate local belief ;
  Master broadcasts θ to all Slaves;
  for $l = 1$ *to* M **do**
    Each Slave calculates local belief for each marginal probability according to Eqs. 6 and 7 on subgraph G_l;
    Slave send back the obtained local belief;
  end
  %Calculate the marginals and update all parameters ;
  Master calculates the marginal according to Eq. 8;
  Master calculates the gradient for each parameter (e.g., by Eq. 5);
  Master updates all parameters, e.g. for α_j,

$$\alpha_j^{new} = \alpha_j^{old} + \eta \frac{\mathcal{O}(\theta)}{\alpha_j}$$

until *convergence*;

Algorithm 1: Distributed model learning.

Structure hole: the feature is defined as whether the user is a structural hole spanner[2]. We use the algorithm proposed in [24] to detect whether a user spans structural holes.

The above two features are defined as node-specific features and are recorded in the $f(y_i, x_{ij})$, except that here x_{ij} is replaced by the status of the user. In addition, we also define another two correlation features.

Social ties: a (binary) feature is defined to represent whether a tie between two users is a strong tie or weak tie [14]. Moreover, we quantify the tie strength according to the communication frequency (e.g., message sent) between users. Then, a new (real-valued) features is defined to represent the number of common neighbors between two users.

Social balance: social balance theory [11] suggests that people in a social network tend to form balanced (triad) structures (like "my friend's friend is also my friend"). For a undirected network, there are four types of (un)balanced triads. A binary feature is defined for each of the triad structure. The similar feature definition was also used in [31].

These two correlation features are incorporated into our model in a similar way as that of the peer conformity factor $g(y_i, y_j', pcf(v_i, v_j))$.

3.3 Distributed Model Learning

Learning the Confluence model is to find a configuration for the free parameters $\theta = (\{\alpha\}, \{\beta\}, \{\gamma\}, \{\mu\})$ that maximizes the log-likelihood objective function $\mathcal{O}(\theta)$. As real social networks may contain thousands or millions of nodes, we have developed a distributed learning algorithm to scale up our model to handle large networks. The distributed learning algorithm was developed based on MPI (Message Passing Interface). In general, the model learning algorithm can be viewed as two steps: 1) compute the gradient for each parameter; 2) optimize all parameters with the gradient descents. The most expensive part is the first step of calculating the gradient. Thus we develop a distributed algorithm to speed up the first step and perform the second step on a single (master) machine.

[2]Roughly speaking, a person is said to span a structural hole in a social network if he or she is linked to people in parts of the network that are otherwise not well connected to one another [5].

We first introduce how we calculate the gradient for each parameter. As the network structure in the social network can be arbitrary (may contain cycles), it is intractable to obtain exact solution of the objective function using methods such as Junction Tree [34]. We use Loopy Belief Propagation (LBP) [26] to approximate the solution. Specifically, we first approximate the marginal distribution $P_\theta(y_i|.)$ using LBP. With the marginal probabilities, the gradient can be obtained by summing over all factor functions. Theoretically, the LBP algorithm does not guarantee a convergence and may result in local maximum, but in practice its performance is good. We empirically compare the effectiveness and efficiency of the algorithm in Section 4. After obtained the marginal distribution $P_\theta(y_i|.)$, we use a gradient descent method (or a Newton-Raphson method) to solve the objective function (Eq. 1). We use α as the example to explain how we learn the parameters. Specifically, we first write the gradient of each unknown parameter α with regard to the objective function:

$$\frac{\mathcal{O}(\theta)}{\alpha_j} = \mathbb{E}[f(y_i, x_{ij})] - \mathbb{E}_{P(y_i|G,A)}[f(y_i, x_{ij})] \tag{5}$$

where $\mathbb{E}[f(y_i, x_{ij})]$ is the expectation of the local factor function $f(y_i, x_{ij})$ given the data distribution in the input network and $\mathbb{E}_{P(y_i|G,A)}[g(y_i, x_{ij})]$ represents the expectation under the distribution learned by the model, i.e., $P(y_i|G, A)$. Similar gradients can be derived for parameter β_i, γ_{ij}, and μ_{ik}.

Now we explain how we use distributed learning to approximate the marginal probability. We use a master-slave architecture, i.e., one master machine is responsible for optimizing parameters, and the other slave machines are responsible for calculating the marginal probabilities. At the beginning of the algorithm, the graphical model of Confluence is partitioned into M roughly equal subgraphs, where M is the number of slave processors. The partition can be done by any graph cut software. After the partition, the subgraphs are then distributed over slave processors. Then each slave processor calculates the "local" belief (the marginal probability) on the subgraph G_l according to the following equations (again we use $P(y_i|G, A)$ as the example in the explanation):

$$m^l_{ij}(y_i) = \sigma \sum_{y_i} \psi^l_{ij}(y_i, y_j)\psi^l_i(y_i) \prod_{k\in NB(i)\setminus j} m^l_{ki}(y_i) \tag{6}$$

$$b^l_i(y_i) = \psi^l_i(y_i) \prod_{k\in NB(i)} m^l_{ki}(y_i) \tag{7}$$

$$P(y_i|.) = \sigma \sum_{l=1}^{M} b^l_i(y_i) \tag{8}$$

where σ denotes a normalization constant; $m^l_{ij}(y_i)$ is the "belief" propagated from node y_j to node y_i; $NB(i)\setminus j$ denotes all nodes neighboring node y_i in the subgraph G_l, except y_j; $\psi^l_i(y_i)$ denotes all defined factor functions related to y_i in the subgraph G_l and is calculated by $\psi^l_i(y_i) = \exp(\sum_{k=1}^d f(y_i, x_{ik}) + \beta_i g(y_i, icf(v_i)))$, and $\psi^l_i(y_i, y_j)$ denotes all correlation factor functions related to y_i in the subgraph; notation $b^l_i(y_i)$ denotes the unnormalized "local" belief collected from each subgraph, and finally by combining them together we obtain the marginal probability $P(y_i|.)$.

However, inevitably there will be some correlation factors defined over nodes that are partitioned into different subgraphs. These correlation factors cannot be calculated due to the high communication cost. Simply eliminating those correlation factors may hurt the learning performance. To alleviate this problem, we present a virtual node based method. In particular, suppose three nodes (y_1, y_2, y_3) in the Confluence model are associated with a group conformity factor $g(.)$. If the partition assigns two nodes (e.g., y_1 and y_2) into one subgraph G_1 and the rest one (i.e., y_3) into another subgraph G_2, then we create a virtual node in the first subgraph G_1 so that the group conformity factor can be still calculated in the subgraph. For the virtual node, we do not calculate the local attribute factors $f(.)$. The distributed learning algorithm is summarized in Algorithm 1.

Model inference. The learned model parameters θ can be used to infer users' future actions. In particular, given the network G and the action history A, we aim to predict users action labels Y^{t+1} at time $t+1$. This can be done by performing the model inference on the network to maximize the conditional probability, i.e.,

$$Y^\star = \arg\max_{Y^{t+1}} P_\theta(Y^{t+1}|G, A) \tag{9}$$

Again, we use the distributed loopy belief propagation algorithm to compute the marginal probability $P_\theta(y_i^{t+1}|.)$ and then predict the action of each user at time $t+1$ as the label that has the largest marginal probability. For each user, we define the individual conformity factor according to the estimated individual conformity from the training data. For defining the peer conformity factor $g(y_i, y'_j, pcf(v_i, v_j))$ between v_i and v_j, we first find the latest past time t' when v_j performed the corresponding action y'_j, and calculate the factor according to Eq. 2. The group conformity factor can be similarly defined.

4. EXPERIMENTAL RESULTS

We conduct various experiments to evaluate the Confluence method. The datasets and codes are publicly available.[3]

4.1 Experiment Setup

Data sets. We evaluate the proposed method on four different genres of networks: Flickr, Gowalla, Weibo, and Co-Author. Table 2 lists statistics of the four networks.

Flickr is a photo sharing network. Users on the site can share photos and add comments to other photos. Flickr users can also create and join different groups. The data set spans the period from Apr. 1st, 2012 to Jun. 16th, 2012. We define the action as adding a comment to a specific photo. Thus the action space includes all photos on Flickr. To avoid the sparsity problem, we remove those photos with less than 5 comments. This results in 144,627 unique actions. We try to study how users' commenting actions conform to the other users in the network.

Gowalla is a location-based social network, where users share their locations by checking-in. The data was from [7] and all check-ins of these users over the period of Jul. 10th, 2010 - Jul. 29th, 2010. The action in this data set is defined as whether a user checks in some location (indicated by hashtag or location ID). Thus the dimension of the action space is the number of available locations. We also remove those locations with less than five check-ins and finally obtain 218,811 unique actions. Our goal is to study whether the users will conform to their friends' check-in behavior.

Weibo is the most popular microblogging service in China. We collected a complete network between 1,700,000 users and all the tweets posted by those users between Sep. 28th, 2012 and Oct. 29th, 2012. The action is defined as whether a user posts a message on a specific topic (indicated by hashtag). We choose the ten most popular topics in 2012 and study how users conform to each other in the network on discussing those topics. We aim to study how users conform to each other on discussing those topics.

[3] http://arnetminer.org/conformity/

Table 2: Statistics of the four networks.

Dataset	Flickr	Gowalla	Weibo	Co-Author
#nodes	1,991,509	196,591	1,776,950	737,690
#edges	208,118,719	950,327	308,489,739	2,416,472
#groups	460,888	N/A	N/A	60
#actions	3,531,801	6,442,890	6,761,186	1,974,466

Co-Author is a network of authors. The data set, crawled by Arnetminer.org [33], is comprised of 737,690 CS authors and 2,416,472 co-authorships over 1975 - 2012. Based on the publication venues, authors are categorized into different domains such as Data Mining, Artificial Intelligence, Computer Graphics, etc.[4] The action is defined as whether an author will publish a paper in a specific domain. Thus, in total we have 200 unique actions. Our goal is to study how an author conforms to the other authors on choosing the publication venue.

Evaluation metrics. To quantitatively evaluate the proposed model, we consider the following performance metrics:

- **Prediction accuracy.** We apply the learned model for action prediction and evaluate its performance in terms of Precision, Recall, F1-Measure, and Area Under Curve (AUC).
- **Scalability performance.** We evaluate the computational time as the efficiency metric.
- **Qualitative case study.** We use a case study as the anecdotal evidence to further demonstrate the effectiveness of the proposed model.

All codes are implemented in C++ and JAVA, and all the evaluations are performed on an x64 machine with E7520 1.87GHz Intel Xeon CPU (with 16 cores) and 192GB RAM. The operation system is Microsoft Windows Server 2008 R2 Enterprise. The proposed distributed learning algorithm has a good convergence property. On average, it converges within 100 iterations.

Comparison methods. Given the input network G and the action history A, we can construct a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,n}$, where $n = |A|$; $\mathbf{x}_i$ is the feature vector associated with user v_i and $y_i = a$ indicates whether user v_i performs the corresponding action a. In this way, we can use existing methods such as Support Vector Machines (SVMs) or Logistic Regression (LR) to train a classification model and then apply the trained model to predict users' future actions. The difference from our proposed Confluence model is that the classification model does not consider the correlation between users' actions. We also compare with Conditional Random Fields (CRFs) [21].

SVM: it uses all defined features associated with each user to train a classification model and then apply it to predict users' actions in the test data. For SVM, we use SVM-light.[5]

LR: it uses logistic regression (LR) to train the classification model with the same features as those in the SVM method. We also compare with the results of Naive Bayes (NB) and Gaussian Radial Basis Function Neural Network (RBF). For all the three methods, we employ Weka.[6]

CRF: it is a graphical model based on Conditional Random Field (CRF). Comparing with CRFs, the factor graph model provides a more explicit explanation for the factorization of the underlying probability distribution [19]. In addition, it is not easy to incorporate the group conformity factors into the CRF model, as users' group memberships could be arbitrary and one user can belong to multiple groups. Thus in the CRF method, we use attribute-based features, the social-based features, and individual conformity features, but do not use the group conformity features. For CRF, we use Mallet [25].

In all the comparison methods, we try to use the same features. The attribute based features are used in all the methods. The social features defined on social ties and social balance are used in CRF and Confluence only, as SVM and LR cannot capture the correlations. As for the conformity features, individual conformity is defined for each user, and is used in all methods; peer conformity is defined for peer friends and is used in CRF and Confluence; group conformity is defined for groups and is used only in Confluence.

[4]Refer to http://arnetminer.org/topic-browser for a list of domains.

[5]http://svmlight.joachims.org/

[6]http://www.cs.waikato.ac.nz/ml/weka/

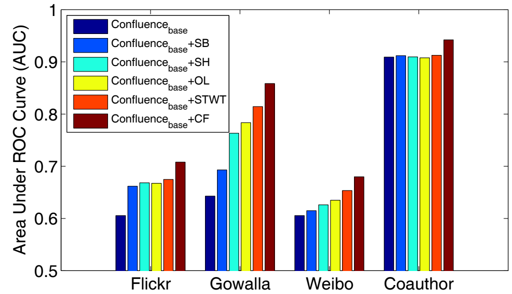


Figure 3: Factor contribution analysis. Confluence$_{base}$ stands for training a Confluence model with only the basic features. Confluence$_{base}$+SB stands for Confluence$_{base}$ with social balance features. Confluence$_{base}$+SH stands for Confluence$_{base}$ with structure hole features. Confluence$_{base}$+OL stands for Confluence$_{base}$ with opinion leader features. Confluence$_{base}$+STWT stands for Confluence$_{base}$ with strong tie/weak tie features. Confluence$_{base}$+CF stands for Confluence$_{base}$ with conformity features.

4.2 Prediction Performance Analysis

On all the four data sets, we use the historic users' actions as the training data in different methods and use the learned model to predict users' action in the next time stamp. Specifically on Flickr each week is a time stamp (which results in 11 time stamps in total), on Gowalla and Weibo each day a time stamp (which result in 20 and 32 time stamps respectively), and on Co-Author each year is a time stamp (which results in 38 time stamps). We perform the prediction for each time stamp and finally report the average performance.

Prediction performance. Table 3 lists the action prediction performance of the different methods on the four data sets. Our method Confluence consistently achieves better performance than the comparison methods. In terms of F1-score, Confluence achieves a 1-17% improvement compared with the SVM, LR, NB, and RBF methods that do not consider the correlation features. CRF also considers some correlation features (such as social tie and social balance based features), thus improves the prediction performance. However it cannot incorporate the group conformity feature, thus still underperforms our method by 0.5-10.5% in terms of F1-score. We produced sign tests for each result, which confirms that all the improvements of our proposed models over the five methods are statistically significant ($p \ll 0.01$).

Factor contribution analysis. In the Confluence model, we define basic features based on the user-associated attributes, and five

Table 3: **Average prediction performance of different methods on the four data sets. The number enclosed in the parenthesis is the standard deviation.**

Data	Method	Precision	Recall	F1-Measure	AUC
Flickr	SVM	0.5921 (±0.0036)	0.5905 (±0.0031)	0.5802 (±0.0012)	0.6473 (±0.0004)
	LR	0.6010 (±0.0052)	0.5900 (±0.0057)	0.5770 (±0.0018)	0.6510 (±0.0008)
	NB	0.6170 (±0.0071)	0.6040 (±0.0083)	0.5920 (±0.0031)	0.6520 (±0.0019)
	RBF	**0.6250 (±0.0039)**	0.5960 (±0.0010)	0.5720 (±0.0024)	0.6700 (±0.0010)
	CRF	0.5474 (±0.0030)	**0.8002 (±0.0009)**	0.6239 (±0.0016)	0.6722 (±0.0010)
	Confluence	0.5472 (±0.0025)	0.7770(±0.0010)	**0.6342 (±0.0010)**	**0.7383 (±0.0006)**
Gowalla	SVM	0.9290 (±0.0212)	0.9310 (±0.0121)	0.9295 (±0.0105)	0.9280 (±0.0042)
	LR	0.9320 (±0.0234)	0.9290 (±0.0234)	0.9310 (±0.0155)	0.9500 (±0.0054)
	NB	0.9310 (±0.0197)	0.9290 (±0.0335)	0.9300 (±0.0223)	0.9520 (±0.0030)
	RBF	0.9320 (±0.0254)	0.9280 (±0.0284)	0.9300 (±0.0182)	0.9540 (±0.0022)
	CRF	0.9330 (±0.0100)	0.9320 (±0.0291)	0.9330 (±0.0164)	0.9610 (±0.0019)
	Confluence	**0.9372 (±0.0097)**	**0.9333 (±0.0173)**	**0.9352 (±0.0101)**	**0.9644 (±0.0140)**
Weibo	SVM	0.5060 (±0.0381)	0.5060 (±0.0181)	0.5060 (±0.0157)	0.5070 (±0.0053)
	LR	0.5190 (±0.0461)	0.6450 (±0.0104)	0.5750 (±0.0281)	0.5390 (±0.0133)
	NB	0.5120 (±0.0296)	0.6700 (±0.0085)	0.5810 (±0.0165)	0.5390 (±0.0132)
	RBF	**0.5240 (±0.0248)**	0.5690 (±0.0098)	0.5460 (±0.0159)	0.5450 (±0.0103)
	CRF	0.5150 (±0.0353)	0.6310 (±0.0121)	0.5720 (±0.0209)	0.6320 (±0.0139)
	Confluence	0.5185 (±0.0296)	**0.9967 (±0.0085)**	**0.6816 (±0.0156)**	**0.7572 (±0.0077)**
Co-Author	SVM	0.7672 (±0.0338)	0.8671 (±0.0145)	0.8256 (±0.0129)	0.8562 (±0.0115)
	LR	0.8700 (±0.0261)	0.7640 (±0.0346)	0.8140 (±0.0221)	0.8500 (±0.0030)
	NB	0.7640 (±0.0177)	0.8510 (±0.0185)	0.8050 (±0.0048)	0.8720 (±0.0074)
	RBF	0.7720 (±0.0182)	0.8830 (±0.0191)	0.8240 (±0.0145)	0.8790 (±0.0031)
	CRF	0.8081 (±0.0252)	0.8771 (±0.0249)	0.8360 (±0.0087)	0.9025 (±0.0025)
	Confluence	**0.8818 (±0.0105)**	**0.9089 (±0.0130)**	**0.8818 (±0.0084)**	**0.9579 (±0.0022)**

types of social features: social balance (SB), structure hole (SH), opinion leader (OL), strong tie/weak tie (STWT), and conformity (CF). Here we examine the contributions of the different social factors defined in our model. Specifically, first we use the basic features to train a model (referred to as Confluence$_{base}$). Then we incrementally add one of the five social features and evaluate its improvement on the prediction performance over that using only basic features. Figure 3 shows the Area Under Curve (AUC) score on the different data sets. We see that different social factors contribute differently in the different networks. For example, the opinion leader based features are very useful in the Gowalla network, but less useful in the Co-Author network. On the other hand, the conformity based features consistently improve the prediction performance on all the networks. In terms of the AUC score, the improvements by adding conformity features range from 2% to 20% in the four networks. This analysis confirms the importance of the conformity phenomena in social networks.

Effects of conformity. We further present an in-depth analysis of how different levels of conformities affect the performance of action prediction. Figure 4 shows the prediction performance (in terms of AUC) of the proposed Confluence by considering different levels of conformities. Confluence$_{base}$ stands for the Confluence method by considering only basic features (i.e., ignoring all conformity factors). It can be clearly seen that without the conformity based factors, the prediction performance drop significantly. Co-Author network is most predictable because the co-authorships are stable and predictable in general. Weibo and Flickr are the most difficult to predict because the user behavior is fairly autonomous and independent. Conformity has most significant prediction impact on Gowalla, which suggests conformity plays an important role in geospatial and mobile applications in social networks. By incorporating the conformity features, significant improvements (+20-30%) over the prediction performance can be obtained on Gowalla. Confluence$_{base}$+I (or +P or +G) respectively indicates that we respectively add individual conformity features (or peer conformity features or group conformity features) into the Confluence$_{base}$ method. By incorporating each type of conformity factors, we observe clear improvement compared to the Confluence$_{base}$ method. We can also see that on all the four data sets, the group conformity is more important than the other two types of conformities. This makes sense, as in most cases conformity is a group phenomenon rather than an individual behavior.

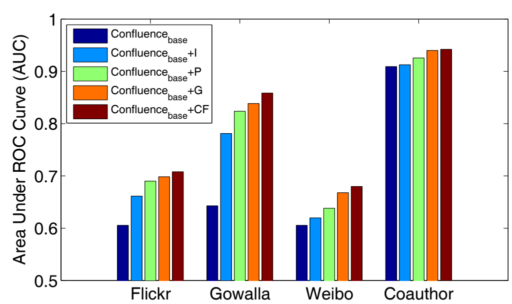


Figure 4: Effect of conformity. Confluence$_{base}$ stands for the Confluence method without any social based features. Confluence$_{base}$+I stands for the Confluence$_{base}$ method plus only individual conformity features. Confluence$_{base}$+P stands for the Confluence$_{base}$ method plus only peer conformity features. Confluence$_{base}$+G stands for the Confluence$_{base}$ method plus only group conformity.

4.3 Scalability Performance

We now evaluate the scalability performance of the distributed learning algorithm on the four networks. In our experiments, we

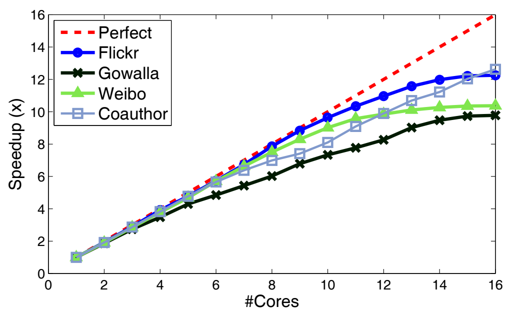


Figure 5: Scalability performance.

Table 4: Running time of the proposed algorithm (hour).

Data Set	Flickr	Gowalla	Weibo	Co-Author
Confluence	1.602	0.245	1.083	0.512
Confluence (single)	19.637	2.395	11.229	6.464
CRF	3.864	0.387	2.547	1.823

use METIS [16] to partition the graph into multiple subgraphs (one for each core). Figure 5 shows the speedup of the distributed algorithm with different number of computer nodes (2, 3, 6, 8, 10, 12, 14, 16 cores) used. The speedup curve is close to the perfect line at the beginning. Though the speedup inevitably decreases due to the increase of the communication cost between the different computer nodes, the distributed learning algorithm can still achieve $\sim 9\times$ speedup with 16 cores. It is noticeable that the speedup curves on different networks present a bit different patterns. This is due to the difference of the network properties (such as densities). Table 4 further gives the running time for learning the proposed Confluence model over 16 computer cores and single compute on different data sets.

Another thing worth noting is that the distributed learning is essentially an approximation of the original learning algorithm on a single machine. We used METIS to partition the graph into multiple subgraphs and distribute the subgraphs onto slave machines. We also evaluate the prediction performance by the distributed learning algorithm. On average, the prediction accuracy by the distributed learning over 16 cores only drops slightly (ranging from 0.5-1.68%), which further demonstrates the effectiveness of the distributed learning algorithm.

4.4 Qualitative Case Study

Now we use a case study from Flickr to further demonstrate the effectiveness of the proposed model. Figure 6 shows an example extracted from Flickr. User A joined three groups (denoted as Group 1, 2, 3 respectively). On 03/10/2012, user A added one comment respectively to Picture 1 and Picture 2. The Action 1 (adding comment to Picture 1) was mainly performed in Group 1 and the Action 2 (adding comment to Picture 2) was mainly performed in Group 2. After modeling with the proposed Confluence method, the modeling results suggest that, for performing Action 1, user A has a strong conformity to user B, but very weak conformity to user D and C. By taking a closer look at the data, we found that Group 1 is a loosely connected group and members have very few connections in the group, and the comments to the same photo are very controversial (such as the comments of B and D to Picture 1). Thus the influence between users are mainly at the peer level. For Action 2, the modeling result suggests that user A has a strong group conformity to Group 2. By checking the data, we found that Group 2 is a tourist group, where people posted their photos taken in the trip. Thus it is very likely that many users added comments to some popular photos together.

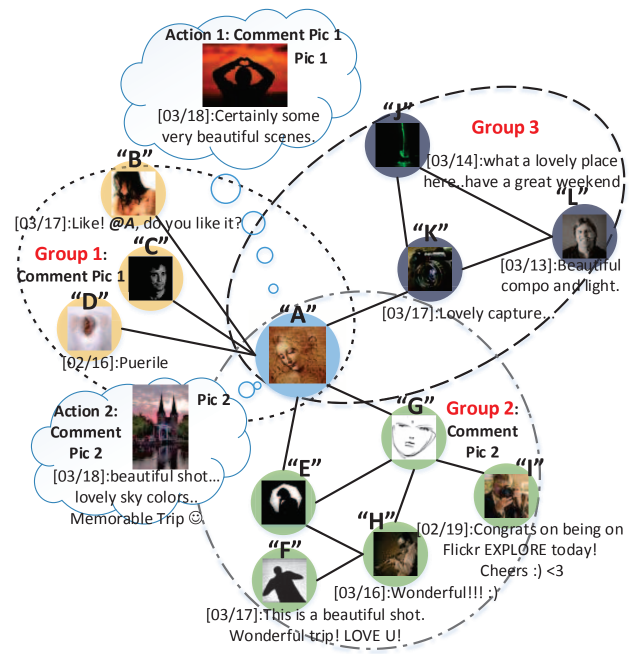


Figure 6: Case Study. User A joined three groups (Group 1, 2, 3). He performed two actions on 03/10/2012 (Action 1: add comment to Picture 1 and Action 2: add comment to Picture 2). Modeling results (by the Confluence model) suggest that User A has a strong conformity to user B and Group 2.

5. RELATED WORK

Conformity was first studied by psychologist through interviews with small groups of participants. Kelman [17] identified three major types of conformity. However, the categorization is mainly from subject and cannot easily be detected by a computational model. In economics, Bernheim [3] proposed the social conformity theory and presented a model for modeling the conformity process. However, due to the lack of real data, he mainly focused on the theoretical aspect of the model.

Considerable work has been conducted for studying the effects of social influence. For example, Bakshy et al. [2] conducted randomized controlled trials to identify the effect of social influence on consumer responses to advertising, and Bond et al. [4] used a randomized controlled trial to verify the social influence on political voting behavior. Anagnostopoulos et al. [1] proposed a shuffle test to examine the existence of social influence. However, most of the methods focus on qualitatively study the existence of social influence in different networks. Tang et al. [32] presented a Topical Affinity Propagation (TAP) approach to quantify the topic-level social influence in large networks. Goyal et al. [13] presented a method to learn the influence probabilities by counting the number of correlated social actions. Tan et a. [30] proposed a model to learn and distinguish the effects of influence, correlation, and uses' action dependency. However, all the aforementioned works mainly consider the peer influence between users and ignore the group conformity effect. Zhang et al. [35] proposed the concept of social influence locality and used a large microblogging network to study how users' behavior is influenced by close friends in their ego networks. Li et al. [22] tried to study the interplay between influence and individual conformity. However, they do not consider the group conformity. Quite a few studies have been done for maximizing the influence spread in social network. Domingos and Richardson [10, 28] formally defined influence maximization as an algorithmic problem and prove its NP-hardness. Chen et al. [6] further developed efficient algorithms to approximately solve the influence maximization problem. While, influence maximization is in nature different from the conformity analysis problem. To the best of our knowledge, this is the first attempt to formally define the problem of conformity influence analysis and to address this problem with a principled method.

6. CONCLUSION

In this paper, we study a novel problem of conformity influence analysis in large social networks. We formally define three major types of conformities, precisely formulate the problem of conformity influence analysis, and propose a Confluence model to model users' actions and conformity. Three factor functions are defined to capture the different levels of conformities. A distributed learning algorithm is presented to efficiently learn the proposed model. We validate the effectiveness and efficiency of the proposed model on four networks. Our experimental results show that the proposed method significantly outperforms several alternative methods. We also present a case study to further demonstrate the effectiveness of the method.

Understanding the fundamental mechanism of social conformity is very important for social network analysis and represents a new and interesting research direction. As for the future work, it would be intriguing to connect the conformity phenomenon with some other social theories such as social status and structural holes so as to understand the formation and dynamic change of the network structure. It is also interesting to design some other model, for example a game theory based model, to model the conformity phenomenon. As for the proposed Confluence model itself, it has many parameters. We also consider adding regularization to control the sparsity of those parameters.

Acknowledgements. The work is supported by the Natural Science Foundation of China (No. 61222212, No. 61073073, No. 61170061), Chinese National Key Foundation Research (No. 60933013, No.61035004), and a fund for Fast Sharing of Science Paper in Net Era by CSTD.

7. REFERENCES

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD'08*, pages 7–15, 2008.

[2] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *EC'12*, pages 146–161, 2012.

[3] B. D. Bernheim. A theory of conformity. *Journal of Political Economy*, 1027(5):841–877, 1994.

[4] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489:295–298, 2012.

[5] R. S. Burt. *Structural Holes: The Social Structure of Competition*. Harvard University Press, 1992.

[6] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD'09*, pages 199–207, 2009.

[7] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD'11*, pages 1082–1090, 2011.

[8] R. B. Cialdini and N. J. Goldstein. Social influence: Compliance and conformity. *Annual Review of Psychology*, 55:591–621, 2004.

[9] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD'08*, pages 160–168, 2008.

[10] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, pages 57–66, 2001.

[11] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[12] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD'10*, pages 1019–1028, 2010.

[13] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM'10*, pages 241–250, 2010.

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

[15] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. *Unpublished manuscript*, 1971.

[16] G. Karypis and V. Kumar. *MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*, Sept. 1998.

[17] H. C. Kelman. Compliance, identification, and internalization: Three processes of attitude change. *Journal of Conflict Resolution*, 2(1):51–60, 1958.

[18] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146, 2003.

[19] F. R. Kschischang, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE TOIT*, 47:498–519, 2001.

[20] T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW'10*, pages 601–610, 2010.

[21] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*, pages 282–289, 2001.

[22] H. Li, S. S. Bhowmick, and A. Sun. Casino: towards conformity-aware social influence analysis in online social networks. In *CIKM'11*, pages 1007–1012, 2011.

[23] L. Liu, J. Tang, J. Han, and S. Yang. Learning influence from heterogeneous social networks. *Data Mining and Knowledge Discovery*, 25(3):511–544, 2012.

[24] T. Lou and J. Tang. Mining structural hole spanners through information diffusion in social networks. In *WWW'13*, pages 837–848, 2013.

[25] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[26] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI'99*, pages 467–475, 1999.

[27] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(066133), 2004.

[28] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, pages 61–70, 2002.

[29] X. Song, Y. Chi, K. Hino, and B. L. Tseng. Identifying opinion leaders in the blogosphere. In *CIKM'06*, pages 971–974, 2007.

[30] C. Tan, J. Tang, J. Sun, Q. Lin, and F. Wang. Social action tracking via noise tolerant time-varying factor graphs. In *KDD'10*, pages 1049–1058, 2010.

[31] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogeneous networks. In *WSDM'12*, pages 743–752, 2012.

[32] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD'09*, pages 807–816, 2009.

[33] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[34] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *UAI'00*, pages 626–633, 2000.

[35] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI'13*, 2013.