User-Level Microblogging Recommendation Incorporating Social Influence

Daifeng Li
Department of Computer Science and Technology, Tsinghua University, FIT Building 3-308, East Zhongguancun Road, Haidian District, Beijing, 100084, China. E-mail: daifli_3000@163.com

Zhipeng Luo
Beijing University of Aeronautics and Astronautics, No. 37, Xueyuan Road, Haidian District, Beijing, 100191, China. E-mail: patrick.luo2009@gmail.com

Ying Ding
School of Library and Information Science, Informatics West 302, 107 S. Indiana Avenue, Bloomington, IN, 47405-7000, USA, and Tongji University, Shanghai, China. E-mail: dingying@indiana.edu

Jie Tang*
Department of Computer Science and Technology, Tsinghua University, FIT Building 3-308, East Zhongguancun Road, Haidian District, Beijing, 100084, China. E-mail: jery.tang@gmail.com

Gordon Guo-Zheng Sun
Tencent Company, No. 66, China Technical Trading Building, Beijing North Fourth Ring Road, Haidian District, Beijing,100080, China. E-mail: gordon.gzsun@gmail.com

Xiaowen Dai
General Motors, China Science Lab, No. 56, Kim Wan Road, Pudong New Area City, Shanghai, 200120, China. E-mail: xiaowen.dai@gm.com

John Du
General Motors, China Science Lab, No. 56, Kim Wan Road, Pudong New Area City, Shanghai, 200120, China. E-mail: john.du@gm.com

Jingwei Zhang
Department of Electronic Engineering, Tsinghua University, Roma Building 5-301, East Zhongguancun Road, Haidian District, Beijing, 100084, China. E-mail: iceboal@gmail.com

Shoubin Kong
Department of Computer Science and Technology, Tsinghua University, FIT Building 3-308, East Zhongguancun Road, Haidian District, Beijing, 100084, China. E-mail: kongsb09@mails.tsinghua.edu.cn

With the information overload of user-generated content in microblogging, users find it extremely challenging to browse and find valuable information in their first attempt. In this paper we propose a microblogging recommendation algorithm, TSI-MR (Topic-Level Social Influence-based Microblogging Recommendation), which can significantly improve users' microblogging experiences. The main innovation of this proposed algorithm is that we consider social

* Corresponding author

Received June 12, 2013; revised October 13, 2015; accepted October 14, 2015

 • Published online 3 August 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23681

influences and their indirect structural relationships, which are largely based on social status theory, from the topic level. The primary advantage of this approach is that it can build an accurate description of latent relationships between two users with weak connections, which can improve the performance of the model; furthermore, it can solve sparsity problems of training data to a certain extent. The realization of the model is mainly based on Factor Graph. We also applied a distributed strategy to further improve the efficiency of the model. Finally, we use data from Tencent Weibo, one of the most popular microblogging services in China, to evaluate our methods. The results show that incorporating social influence can improve microblogging performance considerably, and outperform the baseline methods.

TABLE 1. Comparison between Tencent and Twitter.

	Tencent Weibo	Twitter
Avg. Degree	71.15407	18.86
Avg. Forward	10.0304	2.3609
Avg. Time	95,875 seconds	102,232 seconds
Avg. Depth/Deepest	1.2898/69	1.1245/22
Original Create	0.63	0.42
Reciprocity	0.25	0.58
Clustering Coefficient	1.38×10^{-5}	0.106
Diameter	15	6
Giant Comp. Percentage	99.95%	93.03%

Introduction

Tencent Weibo (Tencent) is one of the most popular microblogging services in China. It is an important platform that combines both social media and social network, and has 469 million users as of 2014. Tencent allows users to share information with their followers or the public by posting messages of up to 140 Chinese characters, which are called weibos. With an average of 60–150 million weibos generated per day, users can access all weibos generated by a specific person, and forward weibos to friends. The forward behaviors can accelerate the spread of information in social networks more efficiently than traditional social media. Many users consider Tencent as a personalized media center, which provides the newest information about political events, economics, celebrities, and their friends' newest activities at the first search. It has become one of the most important information sources in their daily lives. Different from Twitter, Tencent Weibo has certain unique features, as noted in Table 1 (the data for Twitter are mainly based on the research of Kwak, Lee, Park, & Moon, 2010; the data for Tencent are collected with a similar size of networks at the end of 2011).

In Table 1, compared to Twitter, Tencent users have higher activity levels in creating and sharing information (the values of "Original Create" and Avg. Forward of Tencent are bigger than that of Twitter). Tencent users have a higher Avg. Degree and lower Reciprocity than that of Twitter, which means that influential users are more influential than those of Twitter. The Avg. Forward of Tencent is much higher than that of Twitter, which means Tencent users are more likely to share information with others, especially for information from celebrities. Notably, the users of Tencent are nearly all from the same country, and investigation of their behaviors is an important complement to current social network studies in China and elsewhere. Yet as a result of the rapidly increasing number of weibos posted, most Tencent users encounter a serious problem of information overload. According to our statistical analysis, Tencent users follow 64 people on average, which generates hundreds or even thousands of weibos each day. It is unfeasible for users to easily browse and find useful information in this huge data set, especially for active users who often have more followers than others; they often spend more time checking all weibos for useful information (Chen et al., 2012). Thus, the important challenge of recommending useful weibos to a user is the focus of this paper. Intuitively, a weibo is useful to a user if that user is interested in or willing to read the weibo. Whether a user is interested in a weibo is determined by many factors, such as the quality of the weibo, the author's degree of influence, and so forth. Personal significance is also an important factor in deciding whether a weibo is useful. Chen et al. (2012) considered topic factor, social relations, and users' interest preferences to generate a collaborative ranking framework. Yan, Lapata, and Li (2012) proposed a graph-theoretic model for tweet recommendation, where they generated three networks to connect users and items and to observe user characteristics such as the influence of their preferences, popularity, diversity, and influence of tweets. Similar to their research, our work also considers those important factors, including topic information, user profiles from their historical records, the influence of a user, and the abstracted key content from a weibo. Hong, Doumith, and Davison (2013) concluded that Co-Factorization Machines (CoFM) with ranking-based loss functions is superior to state-of-the-art methods and yields interpretable latent factors. The co-ranking framework makes analysis of an extensive feature set, which is extracted from a certain real-world social network (e.g., Twitter), and the proposed model obtains substantial performance gains over competitive approaches on a large real-world data set. Feng and Wang (2013) proposed a feature-aware factorization model to re-rank the tweets and demonstrated the effectiveness of the proposed model. Tan, Li, Zhang, and Guo (2013) utilized a factor graph-based model to combine both node attributes and network information into a unified framework. Qian, Zhang, Zhang, and Duan (2013) applied community detection algorithms to realize collaborative recommendation. The most important difference is that our research also takes social influence into consideration, where its direct influences are studied by the daily communications between two users, while the indirect influences are learned by applying social status theory (Hopcroft, 2012; Tang, Lou, & Kleinberg, 2012; Tang, Zhuang, & Tang, 2012). We also consider constraints of

DOI: 10.1002/asi

these influential relationships under different topics. The main contributions are as follows:

- Our model incorporates explicit Tencent Weibo features such as the degree of user influence, topic information, the main content of weibos, social relations, and topic information into a unified framework, which can further help improve recommendation results.
- For determining direct social influence, we are able to identify the influential relationships between two users by studying their historical communication records, and for determining indirect social influence, we detect the influential relationships by applying social status theory.
- We add topic information into structural analysis of indirect influence. Experiment results show that this method can improve performance and provide more personalized recommendation services based on users' interests.

We verify our proposed model on a large-scale Tencent data, which help us better understand user behaviors in Tencent Weibo.

Related Work

Social Influence

One main purpose of social influence analysis is to detect and evaluate the existence of social influence (Anagnostopoulos, Kumar, & Mahdian, 2008). Kempe, Kleinberg, and Tardos (2003) constructed an NP-Hard problem to solve influence maximization in social network settings. Tang, Sun, Wang, and Yang (2009) measured social influence in relation to different topics and proposed Topical Affinity Propagation (TAP) to model the topic-level social influence. Liu, Tang, Han, Jiang, and Yang (2010) designed an LDA-based Social Influence model to detect influential relationships among individuals. Crandall, Cosley, Huttenlocher, Kleinberg, and Suri (2008) developed techniques for identifying and modeling the interactions between social influence and selection by using data from online communities. Jiang et al. (2012) used indirect influence to improve the performance of recommendations, but they did not incorporate the structure of social influence into their models. Some research also incorporated social structures from the social network theory (Easley & Kleinberg, 2010) into social influence analysis, For example, Hopcroft, Lou, and Tang (2011) used the social balance theory to predict users' followers on Twitter. Different from their research, we mainly address indirect structural influences by combining social status theory with topic information.

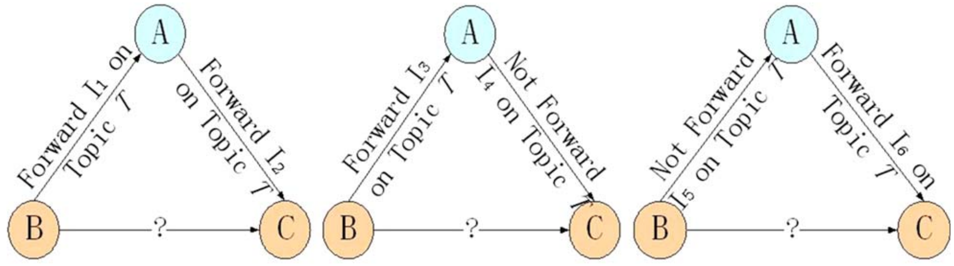
Microblogging Recommendation

As Twitter has become an extremely popular social medium with great impact, plenty of research has focused on analyzing the personal interests of Twitter users and building recommendation algorithms. Michelson and Macskassy (2010) detected the entities of each tweet, and discovered the topics of interest for Twitter users. Ramage, Dumais, and Liebling (2010) applied labeled topic models to analyze the content of each tweet. Yang et al. (2011) established a joint friendship-interest propagation model to present link prediction and tweet recommendation in a unified framework. Chen et al. (2012) proposed a collaborative personalized tweet recommendation algorithm and adopted a latent factor model-based collaborative ranking method to capture users' personal interests in Twitter. Three elements of Twitter, tweet topic level factors, user social relation factors, and other explicit features, are considered major features. Yan et al. (2012) recommended tweets by ranking tweets and their authors simultaneously, using random walk as their basic algorithm to realize co-ranking from three networks: user network, tweet network, and user-tweet network. Hong et al (2013) concluded that Co-Factorization Machines (CoFM) with ranking-based loss functions is superior to state-of-the-art methods and yields interpretable latent factors. The co-ranking framework makes analysis based on an extensive feature set, which is extracted from a real-world social network (e.g., Twitter), and the proposed model obtained substantial performance gains over competitive approaches. Feng and Wang (2013) proposed a feature-aware factorization model to re-rank the tweets. That research achieved excellent performance, but did not provide insights into how social influence is generated according to users' historical records, or how the structure of indirect influence determines the results of tweet recommendation. In this paper, we combine both global Tencent features and topic-level social influence into a unified framework to demonstrate its usefulness in microblogging recommendations.

Factor Graph

Factor graph is a probability-based graph model generated by a Bayesian network or Markov random fields (Tang et al., 2009). The factor graph is performed by passing the "message" along the edges of the graph. Factor graphs are mainly used to model complex real-world systems and to derive practical message-passing algorithms to address association detection and estimation problems (Kschischang, Frey, & Loeliger, 2011; Loeliger, 1998). In recent years, factor graph has also been widely used in different kinds of social networks, such as Twitter (Tan, Tang, Sun, Lin, & Wang, 2010; Tan et al., 2011), Academic Search (Tang, Zhuang, & Tang, 2011), and PatentMiner (Yang et al., 2012). Social structures are also applied in that research, especially for identifying social ties by using social balance and social status. In this study we use factor graph to analyze the communication networks generated by Tencent users, and we also extend the traditional factor graph to incorporate social influence analyses, which abstracts an influential edge into a point, incorporating the social status theory and the topic information. These improved methods can capture the influential relationships more easily and efficiently than standard approaches. Importantly, the results show that incorporating social influence can significantly improve search performance.

DOI: 10.1002/asi



(a) *TP1* (Strong) (b) *TP2* (Weak) (c) *TP3* (Weak)

FIG. 1. Micro indirect influence structures. *TP1* describes a situation of strong indirect influence structure, where *B* has forwarded *A* and *A* has forwarded *B* on the same topic; while *TP2* and *TP3* describe two weak structures. In our research, we found that the number of *TP1*, *TP2*, and *TP3* can significantly influence the performance results. [Color figure can be viewed at wileyonlinelibrary.com]

Problem Definition

In this section we present a formal definition of the problem. A static social network can be represented as $G=(V, E, I)$, where V is the set of $|V|=N$ users, $E \subset V \times V$ is the set of directed links between users, and I is a set of all weibos (similar to tweets in Twitter). In this study we only consider the "Forward" relationship as the links among users, an approach based on the pre-assumption that "The user has a high probability of being interested in a weibo if he/she forwards it." Nowadays, many researchers consider "Forward" as a more important index than "Follow" for evaluating the influential relations between users (Kwak et al., 2010). The main attributes of an original weibo/item I are $I=\{UID\{X\}, KW\{K_1:\omega_1; K_2:\omega_2; K_3:\omega_3\}, T, Time\}$, where KW is the set of the most important list of key words from weibo/item I, and $Ki{:}Wi$ represents the i^{th} key word Ki and its weight Wi in I. The extraction of key words and the calculation of weights can be applied by using FudanNLP,[1] where *UID* is the author *ID* of weibo/item *I*, and *T* is the main topic information of weibo *I*. Users' attributes *X* are mainly organized into three parts: **u**sers' interest in keywords *X(KW)*, topics *X(T)*, and direct influence toward other users *j*: $X(O_{I(UID),j})$. For example, when recommending weibo *I* to user *j*, author of *I* is *I(UID)*, *I(UID)*'s attributes towards user *j* are also important to calculate direct influence. In our research, the main attributes are generally about: *GN* (the number of total replies and comments as well as mentions and forwards of *I(UID)'s* followers); *RN* (the number of weibos-replies between *I(UID)* and *j*); *CN* (the number of weibos-comments between *I(UID)* and *j*); *FN* (the number of weibo-forwards between *I(UID)* and *j*); *MN* (the number of weibo-mentions between *I(UID)* and *j*); and *EN* (the number of weibo mails between *I(UID)* and *j*). We assign $O=\{GN, RN, CN, FN, MN, EN\}$. Given this, we can define the user's influence as follows:

Definition 1. Direct influence between users: The topic-level influence of user *A* towards user *B* $D^k_{A->B}$ can be defined as how *B* will be influenced by *A* on topic *k*. The range of $D^k_{A->B}$ is from -1 to 1, where $D^k_{A->B}<0$ means *A* has a negative influence on *B*, and $D^k_{A->B}>0$ means *A* has a positive influence on *B*. Negative means that *B* has a high probability of disliking *A*'s weibo on topic *k*, and positive means that *B* has a high probability of liking *A*'s weibo on topic *k*.

Direct influence means that the influence can be learned through the communication records of *A* and *B*. In our research, we consider that if user *A* forwards user *B* on a certain topic *T* one time, we assign a value of positive influence from *B* to *A*. Yet evaluating negative influence on Tencent Weibo is intractable in this case, because we do not know whether or not user *A* reads *B*'s weibo. So we use an approximate method to identify negative influence, that is, for each positive influence between user *A* and *B* on a certain topic *T*, we find a negative instance, which is that user *A* did not forward user B on the same topic, as this negative influence.

Definition 2: Indirect influence between users: Indirect influence can be defined by applying the social status theory (Tang et al., 2011), where we define indirect influence using this theory as, if user *B* likes *A*'s weibos related to topic *T*, and *A* likes *C*'s weibos related to topic *T*, then *B* has a high probability of liking *C*'s weibos related to topic *T*. This can be seen as a strong micro-influence structure between *B* and *C*, while for other situations (such as *B* dislikes *A*, *A* likes *C*, wherein we call them weak micro-structures), we assign them a low probability. The main topic-level micro-influence structures are listed in Figure 1.

Proposed Model

TSI_MR Model

Based on the aforementioned definition, we propose a Topic-level Social Influence-based Weibo Recommendation (TSI_MR) model to learn Tencent users' behaviors and make recommendations. Assume we have *U* users and *M* weibos. The objective function is defined as:

$$P(Y|G)=\prod_{i\in M}\prod_{j\in U} IS(y_{ij}, TP_{I(UID),j}, I_i(T)) f(y_{ij}, I_i(T), O_{I_i(UID),j}) \\ g(y_{ij}, I_i(T)) h(y_{ij}, I_i(KW)) \tag{1}$$

where $Y=\{y_{11}, y_{12}, \ldots y_{UM}\}$ represents the results of the recommendation, $y_{ij}=1$ represents user *j* likes the weibo $i \in I$,

[1]http://code.google.com/p/fudannlp/downloads/list

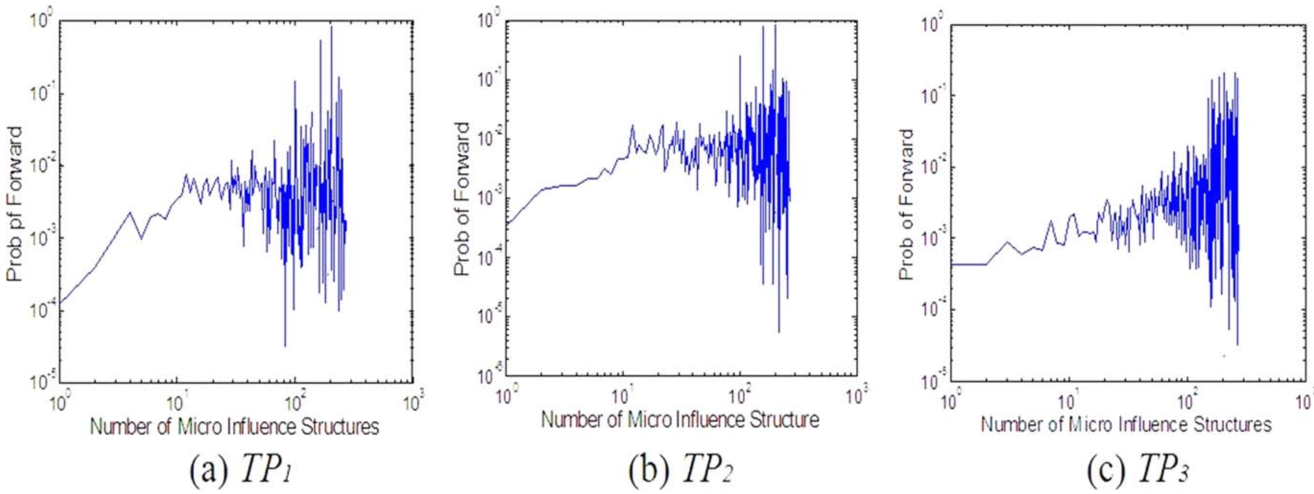


FIG. 2. "Forward" probability versus number of micro influence structures. As seen in the three subfigures, when the number of micro structures is less than a certain value, for example, 40 for *TP1*, there exists a significant linear increase between the number of structures and forward probability for all types *TP1*, *TP2*, and *TP3*. According to our analysis, more than 98% of users have fewer than 40 micro structures. That is the reason why the first half of the curve in each subfigure is a linear increase and the second half exhibits fluctuations. [Color figure can be viewed at wileyonlinelibrary.com]

and f, g, and h are feature functions of the conditional probability distribution of homophily and direct influence $P(y_{ij}|I_i(T), X_j(O_{I_i(UID),j}))$, user j's topic preference of i^{th} weibo $P(y_{ij}|X_j(I_i(T)))$ and user j's keyword preference of i^{th} weibo $P(y_{ij}|X_j(I_i(KW)))$. $IS(y_{ij}, TP_{I(UID),j}, I_i(T))$ is used to calculate indirect influence between user j and $I_i(UID)$, where TP is the structure type, and T is topic of i^{th} weibo. We abstract each "Forward Behavior" as a node, and design our factor graph model based on these abstractions. According to our investigation, the huge number of Tencent users contribute an average of 200 million forwarding behaviors each day, which can be handled by using the distributed high-performance server. But the total indirect influence relationship is bigger than 1,000 billion, which is not easy to handle. As for our selected 1,100 most active users, they totally contribute about 141,000 forwarding behaviors in 3 months, but the indirect influence relationship exceeds 1.5 million, which is 10 times more than forwarding behaviors. Assume there are total U Tencent users, for each Tencent user they have m followees and n followers on average. Then the approximate time complexity for calculating indirect influence is $O(U \times m^3 \times n)$, where U is about 500 million, m is around 70, and n is around 65. While for Loopy Belief Propagation (LBP), which is introduced in this paper for calculating log-likelihood, the time complexity is about $O(2 \times V \times \overline{f})$ for one iteration, where V is the number of forwarding behaviors and $\overline{f}$ is the number of features. In our applications, V and f are usually very big. In order to handle that problem, we proposed a Message Passing Interface (MPI)-based distributed algorithm, which can be seen in the Distributed Learning section to improve the efficiency of our model. The main idea is to first partition the forwarding graph into several subgraphs by applying graph partition algorithms; each subgraph has a strong inner connection and a weak outer connection. Second, assign each subgraph to a certain processor to speed up the learning process. As seen in the experimental result, the distributed model can significantly improve efficiency. While for a larger training data set, for example, 1 million users, an algorithm with one processor cannot work normally, while a distributed algorithm can gain the result within 3 days. Then the direct influence based on "Forward Behaviors" can be defined as:

$$f(y_{ij}, I_i(T), O_{I_i(UID),j}) = \frac{1}{Z_f} \times \exp \{\alpha \times \Phi(y_{ij}, X_{I_i(UID),j}(I_i(T)), X_j(O_{I_i(UID),j}))\} \tag{2}$$

where $\Phi(y_{ij}, X_{I_i(UID),j}(I_i(T)), X_j(O_{I_i(UID),j}))$ is defined as an exponential function:

$$\Phi(y_{ij}, X_{I_i(UID),j}(I_i(T)), X_j(O_{I_i(UID),j})) = \exp\left\{ y_{i,j}^2 + \frac{X_{I_i(UID),j}^2(I_i(T)) + X_j^2(O_{I_i(UID),j}))}{2} - 2 \right\} \tag{3}$$

Formula (2) means that we can predict users' behaviors *yij* based on their preferences and their direct influence relationships. Besides, according to our statistical analysis for 3 months of Tencent data (November–January, 2012), we find that the distribution of "Forward" probability along with micro-influence structures satisfies an exponential increase. In Figure 2, we only consider strong influence structures, as defined in Definition 2. We select three different topics: *T1*: Politics, *T2*: Economics, and *T3*: Fashion as examples, where the number is counted as: If user *A* forwards user *B* on topic *T*, and B forwards user *C* on topic *T*, then the count of micro-influence structures between user *A* and user *C* increases to 1.

As seen in Figure 2, the number of strong micro-influence structures can make a significant contribution towards Forward probability (curves with linear increases include 98% users). Similar research shows that weak structures can make small contributions for improving Forward predictions. Based on the statistical analysis described earlier, we can first design the formula of influential relationships, as noted in Formula (4):

$$IS(y_{ij}|I_i(UID),j,TP,T)=\frac{1}{Z_{IS}}\times\exp\{\lambda\times\Omega(y_{ij},I_i(UID),j,TP,T)\} \tag{4}$$

where $\Omega(y_{ij},I_i(UID),j,TP,T)$ is defined as:

$$\Omega(y_{ij},I_i(UID),j,TP,T)=\frac{e^{Indicator(S_{I_i(UID),j}|TP,T)}}{1+e^{Indicator(S_{I_i(UID)j}|TP,T)}} \tag{5}$$

$S_{I_i(UID),j}$ indicates whether micro-influence structures with type TP exist between user $I_i(UID)$ and j. "Indicator" is an indication function used to describe the existence of $S_{I_i(UID),j}$. We assign different Indicator values for different micro-influence types. We also use an exponential increase function to design the probability distribution formula with other attributes noted in Formulas (6) and (7) as follows:

$$g(y_{ij},X_j(T))=\frac{1}{Z_g}\times\exp\{\beta\times\Theta(y_{ij},X_j(T))\} \tag{6}$$

$$h(y_{ij},X_j(KW))=\frac{1}{Z_h}\times\exp\{\gamma\times\Psi(y_{ij},X_j(KW))\} \tag{7}$$

where $\Theta(y_{ij},X_j(T))$ and $\Psi(y_{ij},X_j(KW))$ are defined as:

$$\Theta(y_{ij},X_j(T))=e^{y_{ij}^2+X_j^2(I_i(T))-2} \tag{8}$$

$$\Psi(y_{ij},X_j(KW))=e^{y_{ij}^2+X_j(I_i(KW))-2} \tag{9}$$

Z can be defined as the integration of the meta-item in Formulas (10)–(13) as:

$$Z_f=\int\exp\left\{\alpha\times e^{y_{ij}^2+\frac{X^2_{I_i(UID),j}(I_i(T))+X_j^2(O_{I_i(UID),j}))}{2}-2}\right\} d\frac{X^2_{I_i(UID),j}(I_i(T))+X_j^2(O_{I_i(UID),j}))}{2} \tag{10}$$

$$Z_{IS}=\int\exp\left\{\lambda\times\frac{e^{Indicator(S_{I_i(UID),j}|TP,T)}}{1+e^{Indicator(S_{I_i(UID)j}|TP,T)}}\right\}dIndicator \tag{11}$$

$$Z_g=\int\exp\{\beta\times e^{y_{ij}^2+(X_j(I_i(T)))^2-2}\}dI_i(T) \tag{12}$$

$$Z_h=\int\exp\{\gamma\times e^{y_{ij}^2+(X_j(I_i(KW)))^2-2}\}dI_i(KW) \tag{13}$$

Formula (4) is used to calculate the indirect influence between two users, while Formulas (6) and (7) are used to calculate the values of users' attributes. In order to obtain the optimized value of the model, which can maximize the log-likelihood derived from Formula (1), we design the vector $\phi=\{\alpha,\beta,\gamma,\lambda\}$, $S=\{\sum\sum\Phi_{ij},\sum\sum\Theta_{ij},\sum\sum\Psi_{ij},\sum\sum\Omega_{ij}\}$ and $Z=\sum_i\sum_j Z_f\times Z_g\times Z_h\times Z_\lambda$. We assign log-likelihood $\Omega=\log(P(Y^{L'}|Y^L,\phi))$, where Y^L is a training instance with labels to indicate whether a current instance has forwarding behaviors, and $Y^{L'}$ is the same training instance with all configurations. For example, for an instance X, the label is $y_x^L=+1$, which means that this instance has a forwarding behavior. $y_x^{L'}$ represents, under the condition of all Y^L, the value of assigning $y_x^{L'}$ as $+1$ or -1. Thus $Y^{L'}|Y^L,\phi$ is the sum of all possible states of users' forwarding behaviors $Y^{L'}$ under the condition of Y^L and ϕ in the forwarding network. $Y^{L'}$ is the sum of all possible states of users' forwarding behaviors without any constraint. Our target is to find the most suitable ϕ to maximize log-likelihood Ω in Formula (14). The target can be expressed in Formula (15):

$$\begin{aligned}\Omega&=\log(\mathrm{P}(\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi))=\log\sum_{\mathrm{Y}^{L'}|\mathrm{Y}^L}\frac{1}{Z}\exp\{\phi^T\mathrm{S}\}\\&=\log\sum_{\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi}\exp\{\phi^T\mathrm{S}\}-\log Z\\&=\log\sum_{\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi}\exp\{\phi^T\mathrm{S}\}-\log\sum_{\mathrm{Y}^{L'}}\exp\{\phi^T\mathrm{S}\}\end{aligned} \tag{14}$$

$$\phi^*=\arg\max(\Omega) \tag{15}$$

$$\begin{aligned}\frac{\partial\Omega}{\partial\phi}&=\frac{\sum_{\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi}\exp\{\phi^T\mathrm{S}\}S}{\sum_{\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi}\exp\{\phi^T\mathrm{S}\}}-\frac{\sum_{\mathrm{Y}^{L'}}\exp\{\phi^T\mathrm{S}\}S}{\sum_{\mathrm{Y}^{L'}}\exp\{\phi^T\mathrm{S}\}}\\&=E_{\mathrm{Y}^{L'}|\mathrm{Y}^L,\phi}(\phi^T\mathrm{S})-E_{\mathrm{Y}^{L'}}(\phi^T\mathrm{S})\end{aligned} \tag{16}$$

The purpose of obtaining optimized parameters from Formula (14) is to derive $\partial\Omega/\partial\phi\sim 0$ in Formula (16). One main solution for the learned process is applying the Gradient Descending Algorithm (Tang et al., 2011) to approach an optimized status, as seen in Algorithm 1.

As seen in Algorithm 1, one main challenge which remains for solving Algorithm 1, is how to calculate

INPUT: Social Network G, Learning Rate η.
OUTPUT: Learned Parameters ϕ.

INITIALIZE ϕ;
REPEAT
CALCULATE $E_{Y^U|Y^U\cup Y^K}(\phi^T S)$ using LBP;
CALCULATE $E_{Y^U\cup Y^K}(\phi^T S)$ using LBP;
CALCULATE the gradient of ϕ according to Eq.(2):
$\nabla_\phi = E_{Y^U|Y^U\cup Y^K}(\phi^T S) - E_{Y^U\cup Y^K}(\phi^T S)$
UPDATE parameters ϕ with the learning rate η:
UNTIL CONVERGENCE

ALGORITHM 1. Learning TSI_MR

$E_{Y^{L'}|Y^L,\phi}(\phi^T \mathrm{S})$ and $E_{Y^{L'}}(\phi^T \mathrm{S})$. Different from other feature-based algorithms, all required instances in our proposed model (instances can be seen as vertexes) have potential topic-based connections with each other (see section, Problem Definition). Thus, it is intractable to calculate the influence of a new weibo towards all possible users by considering different kinds of connections. In this study we utilize LBP to solve that problem. LBP is a typical algorithm used for calculating the joint probability distributions of a factor graph (Murphy, Weiss, & Jordan, 1999; Tang, Lou et al., 2012; Tang, Zhuang et al., 2012), where its main solution is to first build vertexes and factors based on Factor Graph. For each vertex and factor, we calculate the "probabilities" of all messages passed around until convergence. This algorithm is widely applied in solving Factor Graph. Based on our problem definition, we design a "Forward Behaviors"-based LBP algorithm, as detailed in Algorithm 2. To make the algorithm solvable, we first omit all loops by deleting one edge, then "Root Node" is defined as users without any "Forwarding Records," and "Leaf Node" represents users who have never been forwarded. We assume that all users and their attributes are considered as nodes set X(x), with the probability functions abstracted as factor S(f), then $\mu_{x->f}(\mathrm{x})$ represents the value of messages passing from nodes to factors, while $\mu_{f->x}(\mathrm{x})$ represents the value of messages passing from factors to nodes. User node x employs two kinds of factors: *F(IS, f)* is used for calculating indirect and direct influence, while *F(g,h)* is used for calculating users' attribute values (the concrete calculation methods for this approach can be found in Tang's research [Tang, Lou et al., 2012; Tang, Zhuang, et al., 2012]). After that, we can repeat calculating the sum of all values from root nodes to leaf nodes, and then from leaf to node, until they converge, when the sum of all values no longer changes (details of these algorithms, which use a recursion strategy, are found in Murphy, Weiss, and Jordan's research [Murphy et al., 1999]). By applying LBP, we can build connections between users' attributes and their structural relationships, and it improves the efficiency of the model by applying an approximate method to solve the intractable problem.

INPUT: Factor Graphs based on Network G,
Parameters set ϕ and Factor Function set $S(f)$.
OUTPUT: $E(\phi^T S)$

INITIALIZE: Assign all $\mu_{x->f}(x)$ equal to 1.
REPEAT by using Recursion Strategies:
FOR each user node x:
CALCULATE his values towards attributes factors $F(g,h)$;
CALCULATE his values towards influence factors $F(IS,f)$;
CALCULATE his values from $F(g,h)$ to attributes;
CALCULATE his values from $F(IS,f)$ to other users;
END
CALCULATE $E(\phi^T S)$ according to [3][22].
Until Converge

ALGORITHM 2. LBP Algorithms for Solving Influence Structures.

The topics are derived by applying the topic model developed by Tang's research (Tang et al., 2008). We first use the topic model to process the whole experiment data to gain topic distribution of each weibo, and then use that distribution as the topic feature input for our proposed TSI_MR model. The topic number is assigned as 50 based on our experience (when the number of topics is bigger than 50, it has no significant influence on the prediction results based on our proposed model). For each weibo, we select the top three ranked topics as the topic descriptions. For example, for a weibo Ii, its distribution on K topics is $\{\theta_{1i}, \theta_{2i}, \ldots \theta_{Ki}\}$. We select the top three ranked topics, for example, $\theta_{5i}, \theta_{4i}, \theta_{3i}$, as the topic description of the current weibo. If another weibo j contains the same topic θ_{4j} with i, then we consider that the two weibos are related to the same topic.

Distributed Learning

Scaling up learning algorithms with large-scaled networks is important for obtaining their practical values. To address this, we designed an MPI- based distributed strategy for TSI_MR to study users' forwarding behaviors. The model runs on a server with 15 Intel(R) Xeon(R) processors (2.13 GHz) and total 120 G memory with 15 RAM. We set one processor as master and the other as slaves. For the whole network, we use the graph partition algorithm to divide it into several subgraphs (Karypis & Kumar, 1998). After that, we send each subgraph to different slaves, where each slave uses the assigned subgraph to calculate LBP. We then return the value back to the master processor, where the master integrates and sums up all values from different slaves, and uses the sum value to update parameter ϕ. The algorithm keeps repeating the process until convergence. The distributed strategy is an approximate method, which can lose some performance features, but indeed improves efficiency, which is necessary for practical applications and online recommendations.

Experiment Results

The proposed model for weibo recommendation is general and can be applied to different social networks. In this section we present various experiments to evaluate the performance of the proposed approach.

Experiment Setup

Data Sets. We performed our experiments using Tencent QQ microblogging. The whole data set was collected from November 1, 2011 through January 5, 2012, which contained about 40,000,000 daily microblogs. To better evaluate our methods, we first categorized all users according to their activities, where the most active users with a high number of forwarding behaviors were chosen as experimental objects. Finally, we selected 1,100 users from the top 2,000 ranked as most active users, 1,000 users who were randomly selected from 500,000~5,000,000 ranked user set as normal active users. The reason for choosing normal active users

DOI: 10.1002/asi

TABLE 2. Experimental data summarization.

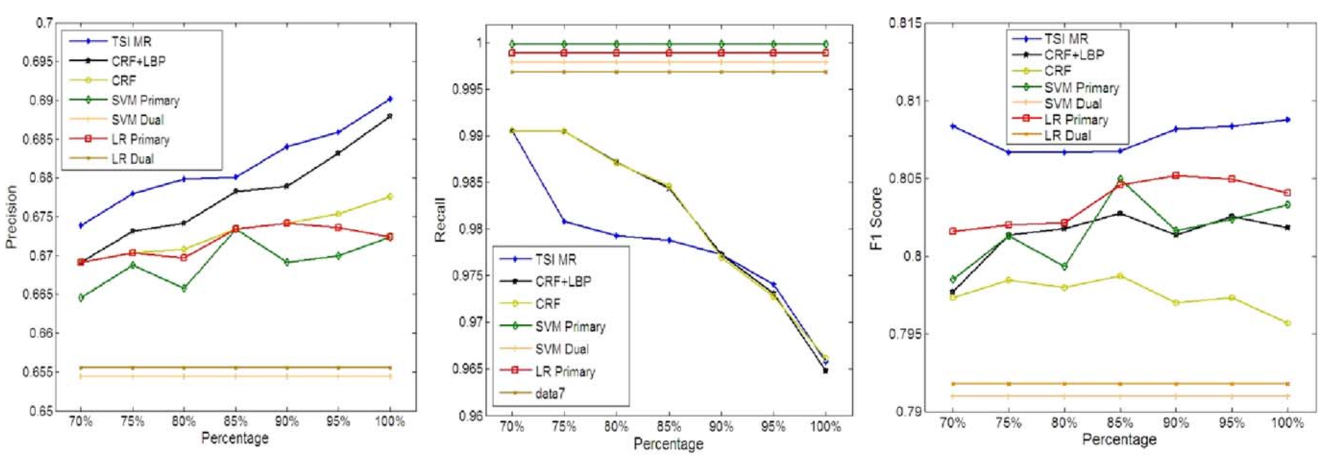
User number	Known behaviors	Unknown behaviors	Total relationships	Key words
High Active: 1,100	292,316	165,053	1,551,621	110,000
Normal Active: 1,000	53,462	26,357	173,233	98,215

was to further prove the validity of TSI_MR; the characteristics of those users are that they keep a level of activity to manage their weibo account to communicate with friends, build social circles, etc. Their monthly forwarding behaviors were from 40 to 200, which can also exhibit their topic preference and social influence. According to official statistical analysis of Tencent, the total number of high active and normal active users is around 40 million as of 2011. A better understanding of their behaviors can create big business opportunities. While for the majority of less active users (ranked after 40 million), due to the very limited information they have provided, TSI_MR cannot work efficiently on learning their behaviors. Thus, currently we do not consider the content of those users as the experimental data. The total statistical information is summarized in Table 2.

As seen in Table 2, the whole training data set is a partially labeled network, in which "Known Behaviors" is defined as what we already know as to whether a user forwarded a weibo or not. "Unknown Behaviors" is defined as we did not know whether a user forwarded a weibo or not. The target of our proposed model is to use known behaviors to infer unknown behaviors. "Total Relationships" is defined as the sum of all direct influence and indirect influence relationships. "Key Words" is the extracted distinct key words from all original weibos. "Key Words" and "Total Relationships" are two important features for us to train the model.

In the experiment, we used known behaviors of 1,100 high active users and 1,000 normal active users as the training data set, unknown behaviors (behaviors that could not be checked before January 5, 2012) as the testing data set. In order to obtain a more comprehensive understanding of the model performance, we made different combinations of the training data and testing data sets. The purpose and solution of all the assignments are as follow:

1. **Purpose**: Test the ability of the proposed model to deal with the sparse training data and handle low deviation data (Figure 3).
   Training and Testing data: use the known behaviors of 1,100 high active users as the training data and the unknown behaviors of 1,100 high active users as the testing data.
2. **Purpose**: Test the ability of the proposed model to deal with the sparse training data and handle high deviation data (Figure 4).
   Training and Testing data: use the known behaviors of 1,100 high active users as the training data and the unknown behavior of 1,000 normal active users as the testing data.
3. **Purpose**: Test the ability of the proposed model to deal with normal active users (Figure 5).
   Training and Testing data: use the known behaviors of 1,000 normal active users as the training data and the unknown behaviors of 1,000 normal active users as the testing data.
4. **Purpose**: Test the generalization ability of the proposed model (Table 6).
   Training and Testing data: use the known behaviors of 1,100 high active users as the training data, use unknown behaviors of high and normal active users as



(a) Precision (b) Recall (c) F1-Score

FIG. 3. Performance on the testing data set with low deviations. Low deviations mean high correlations between the training data and the testing data, which mean that we have plenty of users' communication records in the training data to learn our model and make use of that communication information to predict users' future behaviors in the testing data. [Color figure can be viewed in the online issue, which is available at wileyonline library.com.]

DOI: 10.1002/asi

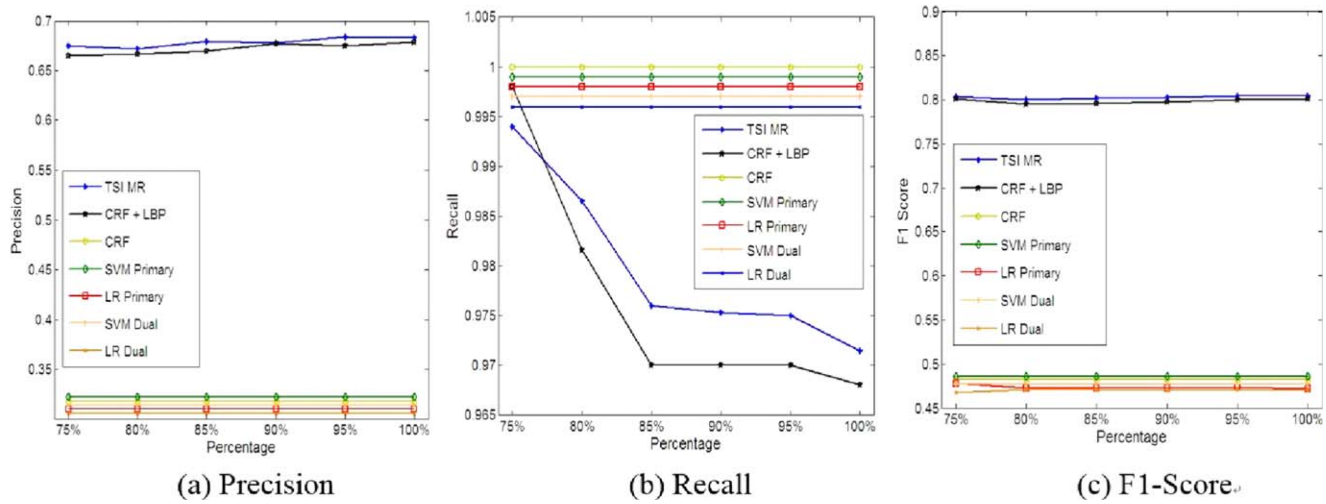


FIG. 4. Performance on the testing data set with high deviations. High deviations mean low correlations between the training data and the testing data, which mean that we do not have enough communication records in the training data to predict users' future behaviors in the testing data. For that situation, we mainly consider using the indirect influence structure among users to make up for the lack of communication information. The results show that TSI_MR can gain similar performance for both the low deviation and high deviation data sets. [Color figure can be viewed at wileyonlinelibrary.com]

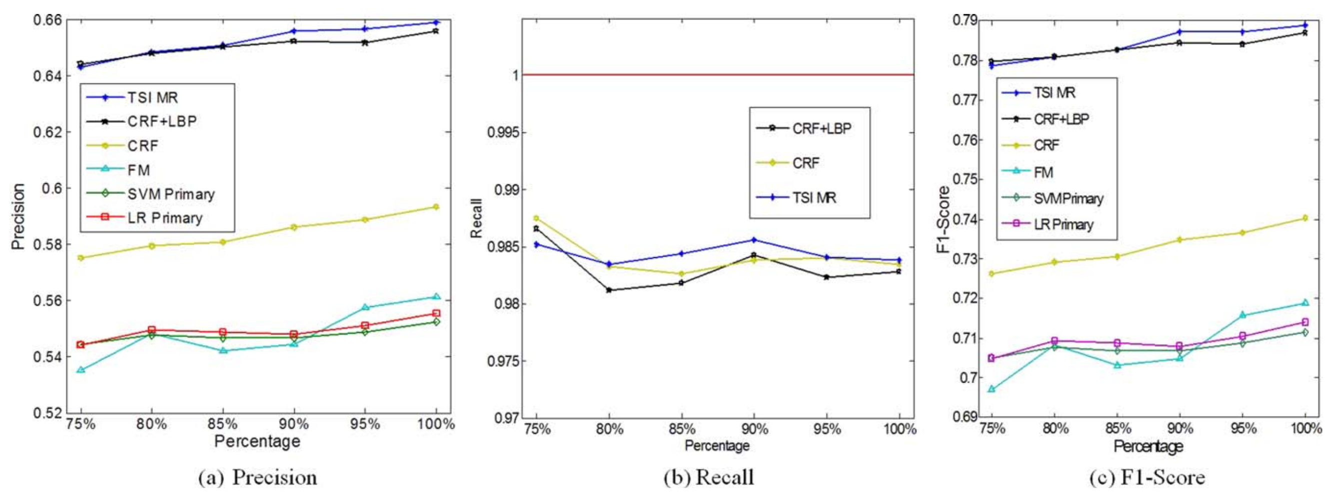


FIG. 5. Performance of the testing data of normal active users with low deviation. The experimental results further prove that incorporating topic-level social influence into the factor graph model can obtain a better performance than other baseline methods. [Color figure can be viewed at wileyonlinelibrary.com]

the testing data (100%, 50%, 0% combination of the two data sets).

5. **Purpose**: Test the performance of the proposed model on the training data of different user groups (Table 7).

   Training and Testing data: use known behaviors of 1,100 high active users and 1,000 normal active users as the training data separately for two models, use the blending of unknown behaviors from two groups of users as one testing data. The two models will be evaluated on the same testing data.

Comparison Method. In our research, we use five classical algorithms for comparison: CRF+LBP, Conditional Random Field (CRF), Factor Model (FM), Support Vector Machine (SVM), and Logistical Regression (LR). The main idea is to predict user interest toward a certain weibo based on their historical behavior records. For CRF, the code is mainly from Wu et al. (2012). CRF+LBP means to apply LBP to calculate the expectation of CRF in each iteration to only incorporate direct influence into consideration, not considering indirect influence. For SVM, we use SVMlight[2]; for Logistical Regression, we use Statistical Toolbox.[3] For CRF+LBP, we adopt the code provided by Tang, Lou et al. (2012), Tang, Zhuang et al. (2012), and for FM, the algorithm is from libFM (http://www.libfm.org/).

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

[3]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

DOI: 10.1002/asi

TABLE 3. Performance of forwarding predictions.

	Precision	Recall	F1-score
TSI_MR	69.87%	96.05%	0.8089
SVM Dual	31.88%	100%	0.4835
LR Dual	66.09%	100%	0.7958
CRF+LBP	68.79%	96.10%	0.8018
CRF	67.81%	96.13%	0.7952
FM	67.78%	100%	0.8080

TABLE 4. *p*-value for model comparison *t* test.

TSI_MR	CRF+LBP	CRF	FM	SVM	LR
P-Value	<0.05	<0.05	<0.05	<0.05	<0.05
Average Deviation	+0.0021	+0.0043	+0.0055	+0.03048	+0.0074

Evaluation Methods. We used precision, recall, F1 score, and area under the curve (AUC) as our evaluation metrics. In the current experimental scene, assume we have N testing data, which include X forwarding behaviors and Y not forwarding behaviors (X + Y = N), model M makes prediction on N testing data, it estimates that T from N is forwarding behaviors, F is not (T + F = N). Then precision, recall, F1-score, true positive and false positive of the AUC are defined as:

$$precision=\frac{T\cap X}{T};\ recall=\frac{T\cap X}{X};\quad f1-score=\frac{2*precision*recall}{precision+recall} \tag{17}$$

$$true\_positive\_rate=\frac{T\cap X}{T\cap X+(X-T\cap X)}=recall \tag{18}$$

$$false\_positive\_rate=\frac{Y-F\cap Y}{(Y-F\cap Y)+F\cap Y}=1-\frac{F\cap Y}{Y} \tag{19}$$

As seen in the formula above, a true positive means that a positive sample is also estimated as positive by our proposed model. The true positive rate is equal to recall.

Prediction Performance

On all training data sets, we used the historic users' behaviors to train the model, and used the learned model to predict users' behaviors for different objects. The comparison results are noted in Table 3.

As seen in Table 3, our proposed model TSI_MR gains a higher F1-score than SVM and CRF, and has a higher accuracy score than CRF, while its recall is a little lower than CRF. This means that it can learn more accurate rules to judge uncertain situations. For example, TSI_MR will drop those nodes with high uncertainty. Another reason for this approach is that by considering indirect influence, we can make recommendations for users without direct connections, while for CRF, mistakes can occur for those situations.

For further evaluation, a statistical significance test was conducted to compare related models. The evaluation was done by calculating *P*-values and average deviation of all test results, which includes both high deviation and low deviation data sets with different ratios. Table 4 shows the experimental results.

TABLE 5. The contribution of each attribute of TSI_MR.

Item	All	No Edges	No FN	No KW	No TP	No RN	No CN
Accuracy	0.6904	0.6879	0.6702	0.6820	0.6822	0.6746	0.6903
Precision	0.6987	0.6981	0.7026	0.6907	0.6822	0.7086	0.6986
Recall	0.9605	0.9610	0.8958	0.9669	1.0000	0.8884	0.9605
F1-Score	0.8089	0.8087	0.7875	0.8058	0.8111	0.7883	0.8089

In Table 4, all the *P*-values for *t*-test are smaller than 0.05, so the assumption that there exists a difference of performance between TSI_MR and other models is confirmed. In addition, we defined "Average Deviation" to evaluate the performance of our model. The steps of calculating average deviation are:

1. Use TSI_MR to run all data sets separately and get a result set $RT\{rt_1, rt_2, \ldots rt_P\}$. RT means the results set of TSI_MR, rt_i means the result of data set i. Use the testing data set to calculate F1 for each result of RT.
2. Repeat Step 1 by using other models: CRF+LBP, CRF, FM, SVM primary, LR primary, and we get result sets RA, RB, RC, RD, RE for each model.
3. Calculate average deviation for the TSI_MR and each of the other completed models using the formula listed here:

$$AverageDeviation=\frac{\sum_{i=1}^{P}(\mathrm{rt}_i-\mathrm{ra}_i)}{P} \tag{20}$$

4. We found that all deviations are greater than zero, which means that TSI_MR performs better than other models from a statistical viewpoint. This may be due to utilization of the influence mechanism as supervised functions, which can choose more related training data sets and narrow the scope of the recommended items.

From another perspective, we would like to consider all of those features wherein one can make a significant contribution to the performance of our proposed model. We thus designed the experiment as follows (see Table 5):

- For each time of calculation, we omit one attribute from the original TSI_MR model and run it on the training and testing data;
- We calculate and compare accuracy, precision, recall, and F1-score for each trained result.

In Table 5, the contribution of *FN* is larger than others. *RN* is also an effective factor to reflect the latent relationships between users. "Edge," which represents indirect influence, also significantly improves the experiment, which means that the assumed existence of indirect influence is established. But due to the limitation of the sparse data, this improvement did not reach the level of our expectations.

DOI: 10.1002/asi

TABLE 6. Performance on different testing data sets.

	Precision			Recall			F1-score		
	100%	50%	0%	100%	50%	0%	100%	50%	0%
TSI_MR	69.87%	68.68%	68.30%	96.05%	96.83%	97.15%	0.8089	0.8036	0.8021
CRF+LBP	68.79%	68.01%	67.84%	96.10%	96.23%	96.81%	0.8018	0.7969	0.7977
CRF	67.81%	54.83%	32.28%	96.13%	98.82%	100%	0.7952	0.7053	0.4881
SVM Primary	67.24%	54.72%	32.43%	100%	100%	100%	0.8041	0.7073	0.4898
LR Primary	67.22%	54.03%	32.43%	100%	100%	100%	0.8040	0.7016	0.4898
FM	67.78%	54.46%	32.43%	100%	100%	100%	0.8080	0.7052	0.4898

For the third aspect, we propose to verify the capability of the TSI_MR model to handle the sparse data and testing data with deviations. We first select two testing data sets T1 and T2, where the first is highly related with the training data set, while the second is related at a low level. For example, if we have abundant communication information for users A and B, we can have high confidence in predicting behaviors between users A and B in the future; if not, then the prediction confidence is low. Low deviation means that for a small amount of high active and popular users, they frequently create plenty of weibos and their weibos are widely forwarded, so we can easily learn their interesting distributions and use that distribution to infer their future behaviors under a certain condition. Furthermore, similar to existing research, the behavior pattern of other users who have strong connections with those high active users can also be learned. Different from those direct connection-based learning models, we step into a further stage to use the social status theory to find inner correlations of indirect social influence among users, which is introduced as high deviation, which means that we use the proposed model to learn the behavior pattern of high active and popular users, and then use the learned model to infer another set of users, who have indirect social influence relationships with current users.

We then randomly select 55%, 65%, 75%, 85%, 95%, and 100% of data from the original training data set as the new training data set, which is applied to verify the capability of TSI_MR for handling the sparse data (noted in Figures 3 and 4).

As seen in Figures 3 and 4, there is no significant distinction between low deviation and high deviation testing data for the TSI_MR model, while for other baselines they cannot work normally when making predictions based on the testing data with high deviation. High deviation means for a target user, to whom we want to recommend weibos, if we know a little about their historical behavior records in the training data, then we cannot gain a better performance to predict their future behaviors by applying general methods. The aim of exhibiting experiment results in Figure 4 is to illustrate that for a high deviation problem, if we know the users' connections with other high active users in the training data, for example, forwarding behaviors, we can also infer those users' certain behavior patterns with a high confidence. The reason is that TSI_MR can better make use of the information of indirect influence between two users, who do not have frequent communication records with each other, to infer their correlations. While according to our statistical analysis, for most of the users in a similar topic domain, they on average contribute 25 indirect influence structures, which provides plenty of information for us to train the TSI_MR model and make a more accurate prediction.

TABLE 7. Performance on training data from two types of user groups.

	Testing data (BTD)		
	Precision	Recall	F1-score
TSI_MR (HAU)	68.68%	96.83%	0.8036
CRF+LBP (HAU)	68.01%	96.23%	0.7969
SVM Primary (HAU)	47.72%	100%	0.6461
LR Primary (HAU)	47.03%	100%	0.6397
TSI_MR (NAU)	66.74%	98.54%	0.7958
CRF+LBP (NAU)	66.32%	98.22%	0.7918
SVM Primary (NAU)	45.68%	100%	0.6288
LR Primary (NAU)	44.79%	100%	0.6187

In order to further validate the proposed model, we used the data from normal active users as our experimental data set and repeated the same experiment with the low deviation assignment. The experimental results will be shown here.

As seen in Figure 5, the probability model with social influence mechanism (TSI_MR and CRF+LBP) significantly outperforms other baselines without considering social influence. Because the training data from normal active users is less plentiful than that of high active users, the performance of TSI_MR and CRF_LBP on normal active users is not as good as that on high active users in Figure 3. While compared with TSI_MR and CRF+LBP, indirect influence can also provide a positive improvement to make TSI_MR outperform CRF+LBP.

In Table 6, we summarize the performance of TSI_MR (100% training data set) on different testing data sets. We assigned three different types of testing data, 100% user coverage, 50% user coverage, and 0% user coverage. 100% user coverage means that all the users in the testing data set can be found in the 100% training data, 50% user coverage means that only 50% users in the testing data can be found in the training data, 0% user coverage means that no users in the testing data can be found in the training data. While for

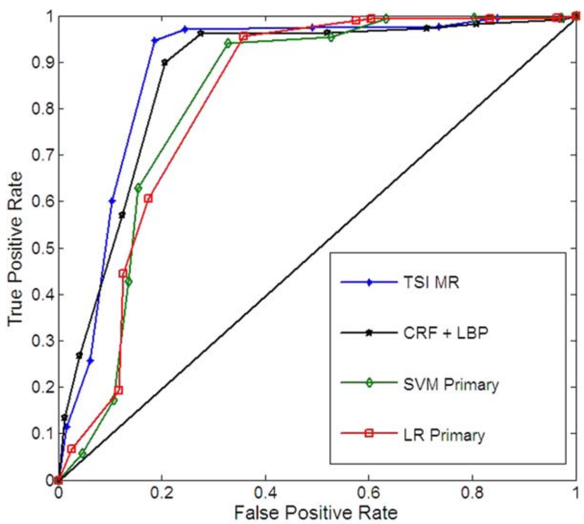


FIG. 6. ROC curves for a 100% training data set of known behaviors from 1,100 high active users; the testing data set is from the unknown behaviors of 1,100 high active users for low deviation evaluation. Three representative algorithms were used as baseline: CRF+LBP, SVM Primary, and LR Primary. For all results, the AUC of TSI_MR is bigger than the other baselines. [Color figure can be viewed at wileyonlinelibrary.com]

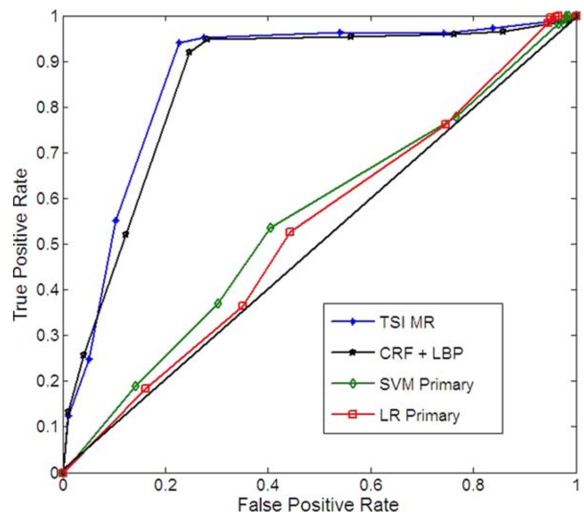


FIG. 7. ROC curves for 100% training data set of known behaviors from 1,100 high active users; the testing data set is from the unknown behaviors of 1,000 normal active users for high deviation evaluation. Three representative algorithms were used as baselines: CRF+LBP, SVM Primary, and LR Primary. For all results, the AUC of TSI_MR is bigger than the other baselines. The AUC of SVM and LR, which have no influence mechanism, reduce significantly. [Color figure can be viewed at wileyonlinelibrary.com]

those users who cannot be found in the training data, they have direct or indirect connections with users in the training data. The aim for processing the current experiment is to observe the generalization capability of our proposed model and other baselines. In order to obtain a high confidence result, we use 10-fold cross-validation to evaluate each test result. The experimental results can be seen in Table 6.

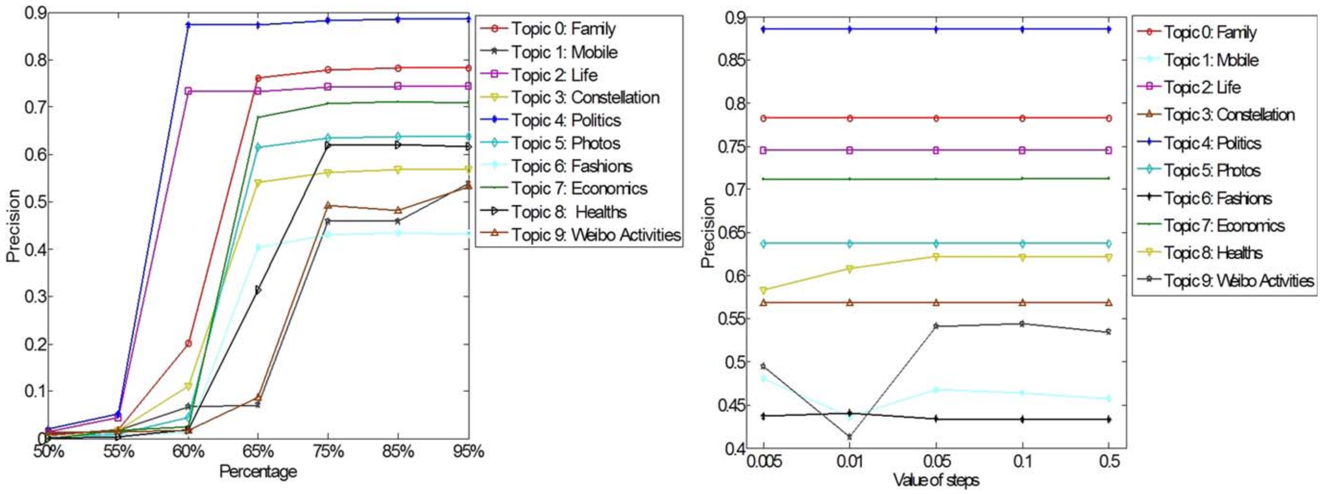
In Table 6, TSI_MR outperforms the other baselines for all three testing data sets, and especially for 50% users coverage and 0% users coverage, the improvement is more significant. The reason is that TSI_MR and CRF+LBP can take direct and indirect influence as new features, and the new features can obtain a closer performance than other features (there is no big change for TSI_MR to make prediction on 100% coverage and 0% coverage testing data). The phenomenon shows that a user's behavior pattern can be approximately fitted by learning her/his high influential neighbors. We also find that TSI_MR outperforms CRF+ LBP with 1% improvement, because we use topic information to divide the users' behaviors. Forwarding behaviors within a similar topic towards one user can be extracted and calculated separately. The topic-based mechanism can further guarantee the performance.

In Table 7, we summarize the performance of our model on two different training data sets: known behaviors from high active users and known behaviors from low active users. We first use the two training data sets to train two models: HAU and NAU; second, we use the 50% blending of unknown behaviors from high active and normal active users as the testing data BTD. Then we evaluate the performance of the two proposed models on the same testing data BTD. A 10-fold cross-validation was applied to guarantee the confidence of the results.

The experimental results show that for the same testing data set BTD, both HAU and NAU, which are trained by TSI_MR, obtain the highest score compared with other baselines. In another aspect, the trained model on HAU outperforms NAU, which means HAU has plenty of information for the model to learn more patterns from user behaviors. In particular, influential relationships play an important role in reducing the gap between different models; this phenomenon further proves that a user's behavior patterns can be learned not only from their own records, but also from the whole network.

In order to better illustrate the performance of TSI_MR, we assign the threshold as {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}, and draw an ROC (receiver operating characteristic) for our proposed model and other baselines. We use known behaviors of 1,100 high active users as the training data. We prepare two testing data groups for low and high deviation tests, which is similar to the assignment in Figures 3 and 4. For the low deviation, we use the unknown behaviors from 1,100 high active users as the testing data; for high deviation, we use the unknown behaviors from 1,000 normal active users as the testing data.

In Figures 6 and 7, we plot the ROC curves to further evaluate the performance of our proposed model for both low deviation and high deviation situations. The ROC curve is used to observe the performance of classifiers under different conditions. Particularly, AUC is the area under ROC curve, whose value is an important indicator to evaluate the performance of a certain classifier. We can see that the AUC of TSI_MR is significantly bigger than other methods, and its ROC curves are all above the diagonal line, implying that TSI_MR is a better method to make weibo recommendations



(a) Percentage VS Topics   (b) Value of Steps VS Topics

FIG. 8. Performance of TSI_MR on different topics. We selected 10 different topics, and evaluated the performance of TSI_MR on different topics. As seen in both Figure 8 (a) and (b), TSI_MR can gain better performance on some topics, such as Politics, Family, Life, and Economics; while for other topics, such as Fashion, Weibo Activities, and Mobile, it can have a relatively low performance. In Figure 8 (a), when the training data are sparse (low percentage of the whole training data), the performance is low, and when the percentage is larger than 60%, the value of precision tends to converge. In Figure 8 (b), the precision of most topics is not significantly influenced by the value of steps. [Color figure can be viewed at wileyonlinelibrary.com]

toward a certain topic than other baseline methods. While for SVM and LR, the ROC curve exhibits a weak confidence for more than 50% of recommended items (the maximum correlation weight of them towards target user is less than 0.1, which is less than that of TSI_MR and CRF+LBP, which are around 0.2). Compared with Figure 6, the ROC curves of SVM and LR is close to the diagonal line in Figure 7. But the ROC curves of TSI_MR and CRF+LBP show no big changes. This phenomenon illustrates that incorporating social influence can significantly improve the weights of true positive items, and distinguish them from the others.

Popular topics are noted in Figure 8. The left subfigure shows the performance on different percentages of the training data set. Topics related to Politics gain the highest scores, yet Family, Life, and Economics also gain a high score, while the topic “Fashion” gains the lowest score. The reason for this is that users’ behaviors may be more predicable on some topics, because their behavior patterns are not easily changed, while for other topics, such as Fashion, accurately capturing their interest changes is intractable. The right subfigure shows the performance on different iteration steps. We assign the step value η from 0.005 to 0.1 using different interval steps, and run it for different topics, where, for most topics, the fluctuations are small. This means that different values of η do not have significant effects on performance results for most situations, but for some other topics, again for Fashion, the fluctuation for different assignments of η is large.

Distributed Performance

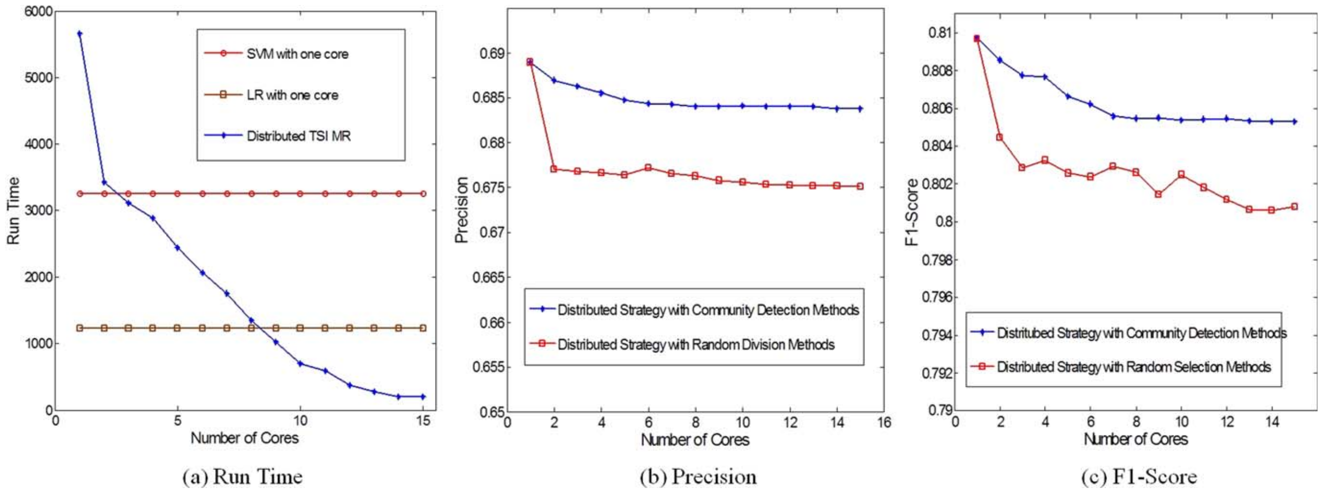
In this section we evaluate the performance of Distributed Strategy on our experiment data set. Figure 9 shows the running time by adopting the Distributed Strategy with a different number of processors (the number is from 1 to 15):

The left subfigure shows the run time of Distributed Strategy with different numbers of cores. Run time is significantly decreased when the number of cores is increased. The middle and right subfigures show the performance of TSI_MR with two Distributed Strategies, where the first is the graph partition algorithm (introduced earlier), while the second is the Random Division method, which separates the whole graphs by randomly eliminating edges. The experiment results show that the first strategy significantly outperforms the second one for both precision and F1-score. The reason for this is that the first one can obtain subgraphs with the lowest connections with others, which means that it can reserve the original information to the maximum extent. The performance of both Distributed Strategies also decreases with the increase in core numbers. The reason for this is that Distributed Strategies are approximate methods, which aim to improve efficiency by losing user connections of the original graph. But the decreasing range is also acceptable (the precision loss is around 0.9%, while the F1-score loss is around 0.7%). In the future, a theoretical study will be promoted for obtaining more optimized results.

Discussion

As introduced earlier, one main contribution of our proposed model is to combine topic-level social influence into our weibo recommendation algorithm. According to the experimental result in the section, the Proposed Model, incorporating the topic-level social influence into our proposed model can significantly improve the performance. In

DOI: 10.1002/asi



(a) Run Time (b) Precision (c) F1-Score

FIG. 9. Performance of the Distributed Strategy on TSI_MR. Figure 9 (a) compares the run time of distributed TSI_MR with other baselines. Figure 9 (b) shows the precision changes with a different number of cores. We also consider the introduced graph partition as a simple community detection strategy, and compare the strategy with random division methods. Figure 9 (c) shows the F1-score changes with a different number of cores. We also compare the strategy with random division methods. [Color figure can be viewed at wileyonlinelibrary.com]

our research, social influence consists of direct and indirect influence. Direct influence can be measured by the selected features and information propagation theory; we used CRF+LBP (use LBP to calculate the expectation of CRF in each iteration) to realize direct influence. Indirect influence is largely based on the social status theory; it is another aspect to supplement the description of weak connections among users in Tencent Weibo. As seen in Table 3, incorporating direct influence (CRF+LBP) can significantly improve the performance compared with other baselines without direct influence, such as CRF, FM, SVM, and LR. The reason is that other baselines do not deal with social network features in an efficient way, while in LBP, the expectation calculation of current vertex (instance) also considers its father vertex set and children vertex set, and further spread the whole network (as seen in Algorithm 1 and Algorithm 2). The advantage of adopting LBP is that it can make use of information of a vertex's neighbors to understand current forwarding behaviors. Furthermore, in many cases, the direct influence may contain not enough information to better understand a user's behavior. For example, user *A* only forwarded user *B*'s weibo one time; thus, the recommendation confidence for *B*'s new weibo towards *A* is low. If we consider the indirect influence of *A* and *B*, for example, *A* is a favorite with user *C* on topic *Z*, *C* is a favorite with *B* on the same topic *Z*, then recommending *B*'s new weibo related with topic *Z* to *A* will gain a higher confidence. More indirect influence structures may further improve the confidence. As seen in Tables 3 and 4, considering indirect influence can make a 0.25% improvement compared with direct influence. According to our real case studies, we found that for a certain type of relationship, which may contain seldom direct forwarding behaviors, but include more indirect influence structures, the prediction result can be significantly improved. While for some cases with low confidence (there is not enough information, which also includes indirect influence between two users, the only information is that one user *A* may forward another user *B's* message several times), the proposed model will fail to recommend *B's* new weibo to *A*, because without indirect influence structure, the positive weight between *B* and *A* can be further reduced.

Another main advantage is that the proposed model can better solve a data sparse problem to a certain extent. For many Tencent users, their forwarding behaviors are very limited and hard to understand. In order to solve that problem, we can use the influence propagation theory to calculate the influence between any two users from the perspective of the whole social network. As summarized in Tables 3 and 4, a well-trained model for a certain amount of active users can make a contribution for other users' weibo recommendations. In Figure 3, low deviation means that we mainly use a user's historical behavior patterns to predict their future behaviors, the main idea of which is similar to the traditional data mining method. In Figure 4, high deviation means that we use other users' well learned models to predict current user's future possible behaviors; during that process, direct and indirect influence both play important roles.

Above all, the main contribution of our proposed model is that it can build an accurate description of latent relationships between two users with weak connections, which can help to improve the performance of the model. Our research illustrates that topic-level social influence can help to better understand users' behaviors in microbloggings. Furthermore, it can also solve data sparsity problems of training data to a certain extent.

Conclusion

In this paper we proposed a TSI_MR model for solving online microblogging recommendation problems in Tencent Weibo. Different from many previous studies, our algorithm applies a supervised algorithm to incorporate direct and

DOI: 10.1002/asi

indirect topic-level social influence into the proposed model to obtain a high performance. The reason for the performance improvement is that topic-level social influence can build an accurate description of latent relationships between two users with weak connections. The experimental results show that incorporating "Social Influence" into a multi-attribute factor graph model can help detect the indirect influence among Tencent users and can clarify users' forwarding behaviors, which can be leveraged for improving weibo recommendations. Furthermore, the topic-level social influence mechanism can be considered a new solution for the data sparsity problem. Second, we used the proposed TSI_MR model to analyze the contributions of different features, which can provide a good foundation for feature selection in the future. And lastly, we designed a Distributed Strategy for handling large-scale data sets, and the experimental results demonstrated a gain in efficiency based on this strategy.

Acknowledgments

This work is supported by the China Post Doc Funding (2012M510027), the National Basic Research Program of China (No.2011CB302302), the He Gaoji Project, the Tencent Company (No.2011ZX-01042-001-002), and the National Natural Science Foundation of China (NSFC Program No.71072037).

References

Anagnostopoulos, A., Kumar, R., & Mahdian, M. (2008). Influence and correlation in social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '08). Las Vegas, USA.ACM. (pp. 7–15).

Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., & Yu, Y. (2012). Collaborative personalized tweet recommendation. Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12). Portland Oregon, USA. ACM. (pp. 661–670).

Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J., & Suri, S. (2008). Feedback effects between similarity and social influence in online communities. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '08). Las Vegas, USA. ACM. (pp. 160–168).

Easley, D., & Kleinberg, J. (2010). Networks, crowds, and markets: Reasoning about a highly connected world. Cambridge, UK: Cambridge University Press.

Feng, W., & Wang, J. (2013). Retweet or not?: Personalized tweet re-ranking. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13). Rome. ACM. (pp. 577–586).

Hong, L., Doumith, A.S., & Davison, B.D. (2013). Co-factorization machines: Modeling user interests and predicting individual decisions in twitter. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13). Rome. ACM. (pp. 557–566).

Hopcroft, J., Lou, T., & Tang, J. (2011). Who will follow you back? Reciprocal relationship prediction. In Proceedings of 20th Conference on Information and Knowledge Management (CIKM '11). Glasgow, Scotland, UK. ACM. (pp. 1247–1252).

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: Understanding microblogging usage and communities. In Proceedings of the Ninth WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis (WebKDD/SNA-KDD '07) (pp. 56–65). New York: ACM.

Jiang, M., Cui, P., Wang, F., Yang, Q., Zhu, W., & Yang, S. (2012). Social recommendation across multiple relational domains. In Proceedings of the 21th ACM International Conference on Information and Knowledge Management (CIKM '12). Hawaii, USA. ACM. (pp. 1422–1431).

Karypis, G., & Kumar, V. (1998). MeTis: Unstructured graph partitioning and sparse matrix ordering system. Side Effects of Drugs Annual. (pp. 206–213).

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD '03). Chicago, IL, USA. ACM. (pp. 137–146).

Kschischang, F.R., Frey, B.J., & Loeliger, H.A. (2011). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), (pp. 498–519).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a new media?. In Proceedings of 19th International Conference on World Wide Web (WWW '10). North Carolina USA. ACM. (pp. 591–600).

Liu, L., Tang, J., Han, J., Jiang, M., & Yang, S. (2010). Mining topic-level influence in heterogeneous networks. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10). Toronto, Canada. ACM. (pp. 199–208).

Loeliger, H.-A. (1998). An introduction to factor graphs. Oxford University Press. IEEE. 21(1). (pp. 28–41).

Michelson, M., & Macskassy, S. (2010). Discovering users' topics of interest on Twitter: A first look. In Proceedings of Fourth Workshop on Analytics for Noisy Unstructured Text Data (pp. 73–80). ACM.

Murphy, K., Weiss, Y., & Jordan, M. (1999). Loopy belief propagation for approximate inference: An empirical study. In UAI (9) (pp. 467–475).

Qian, F., Zhang, Y., Zhang, Y., & Duan, Z. (2013). Community-based user domain model collaborative recommendation algorithm. *Tsinghua Science and Technology*, 18, 353–359.

Ramage, D., Dumais, S., & Liebling, D. (2010). Characterizing microblogs with topic models. In International AAAI Conference on Weblogs and Social Media (AAAI '10). Atlanta, USA. ACM. (pp. 130–137).

Tan, C., Lee, L., Tang, J. (2011). User-level sentiment analysis incorporating social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11). San Diego, CA, USA. ACM. (pp. 1397–1405).

Tan, F., Li, L., Zhang, Z., & Guo, Y. (2013). Latent co-interests' relationship prediction. Tsinghua Science and Technology, 18, 379–386.

Tan, C., Tang, J., Sun, J., Lin, Q., & Wang, F. (2010). Social action tracking via noise tolerant time-varying factor graphs. In Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10). Washington, DC, USA. ACM. (pp. 1049–1058).

Tang, J., Lou, T., & Kleinberg, J. (2012). Inferring social ties across heterogeneous networks. In Proceedings of the fifth ACM International Conference on Web Search and Data Mining (WSDM '12). Seattle, Washington, USA. ACM. (pp. 743–752).

Tang, J., Sun, J., Wang, C., & Yang, Z. (2009). Social influence analysis in large-scale networks. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '09). Pairs, France. ACM. (pp. 807–816).

Tang, J., Zhang, J., Yao, L., Li, J., & Su, Z. (2008). ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '08). Las Vegas, USA. ACM. (pp.990–998).

Tang, W., Zhuang, H., & Tang, J. (2011). Learning to infer social ties in large networks. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge

Discovery in Databases (ECML/PKDD '2011). Athens, Greece. ACM. (pp. 381–397).
Tang, W., Zhuang, H., & Tang, J. (2012). Learning to infer social ties in large networks. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in DataBase Bristol, UK. (PKDD '12). ACM. (pp. 381–297).
Wu, S., Fang, Z., & Tang, J. (2012). Accurate product name recognition from user generated content. (ICDM Contest) In Proceedings of ICDM 2012 Contest. New York, USA. IEEE. (pp. 874–877).
Yan, R., Lapata, M., & Li, X. (2012). Tweet recommendation with graph co-ranking. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12). Korea. ACM. (pp. 516–525).
Yang, S., Long, B., Smola, A., Sadagopan, N., Zheng, Z., & Zha, H. (2011). Like alike: Joint friendship and interest propagation in social networks. In Proceedings of the 20th International Conference on World Wide Web (WWW '11). Lyon, France. ACM. (pp. 537–546).
Yang, Y., Tang, J., Keomany, J., Zhao, Y., Ding, Y., Li, J., & Wang, L. (2012). Mining competitive relationships by learning across heterogeneous networks. In Proceedings of 21th Conference on Information and Knowledge Management (CIKM '2012). Hawaii, USA. ACM. (pp. 1432–1441).