Diffusion of "Following" Links in Microblogging Networks

Jing Zhang, Zhanpeng Fang, Wei Chen, *Member, IEEE,* Jie Tang, *Member, IEEE*

Abstract—When a "following" link is formed in a social network, will the link trigger the formation of other neighboring links? We study the diffusion phenomenon of the formation of "following" links by proposing a model to describe this link diffusion process. To estimate the diffusion strength between different links, we first conduct an analysis on the diffusion effect in 24 triadic structures and find evident patterns that facilitate the effect. We then learn the diffusion strength in different triadic structures by maximizing an objective function based on the proposed model. The learned diffusion strength is evaluated through the task of link prediction and utilized to improve the applications of follower maximization and followee recommendation, which are specific instances of influence maximization. Our experimental results reveal that incorporating diffusion patterns can indeed lead to statistically significant improvements over the performance of several alternative methods, which demonstrates the effect of the discovered patterns and diffusion model.

Index Terms—Link diffusion, Triad formation, Social network

1 Introduction

In a microblogging network such as Twitter, users' "following" behaviors form the "following" links, which is fundamental to the formation of a network structure. The "following" links are observed to be correlated. For example, when a user A follows another user C, this creates a chance for A's follower B to discover C, where A, B and C form a basic triadic structure[1]. We show the link correlations in five different triadic structures in Figure 1, where the observations are based on the dataset described in Section 3. Given the preexisting link between A and B and the new link from A to C added at time t', we present the ratio of new link from B to C, created within time frame δ after t' for each triadic structure. Time t' and t are constrained by $0 \leq t - t' \leq \delta$, where δ is a time delay parameter indicating that the formation of one link can trigger the formation of another link within a short time interval and is empirically set as 7 in units of days. From the figure, we can see that when there is a preexisting link between A and B, the ratio of B following C triggered by A following C will be improved by at least 500 times (Figure 1(c),1(d) and 1(e) vs. 1(a)). The ratios with the neighboring links shown in Figures 1(c),1(d) and 1(e) also present at least 1.54 times of that in a one-hop away link structure shown in Figure 1(b) (We selected the one-hop away link structure with the maximal ratio). Furthermore, a two-way relationship between two users positively affects how likely the new link will propagate (Figure 1(e) vs. 1(c) and 1(d)). The example implies that the formation of A following C influences the formation of B following C. Understanding the diffusion mechanism for such links can give us insight into how a network evolves over time. This can benefit many applications, such as friend recommendation and "word-of-mouth" influence maximization to attract more links in a network. Specifically, there are two potential applications, follower maximization and followee maximization, of link diffusion phenomenon. The two applications aim at activating more links in a network. The requirement is derived from the discussion with real social network companies, who would like to improve user stickiness through encouraging users' behaviors. Even though only 1-2% of triads form new links, given the large scale of the real social network, it still leads to a significant amount of link additions if we can utilize link diffusion in a smart way.

Although the study of link diffusion is related to a number of areas extensively researched, such as link prediction [23], [26], [30], [43], network formation [3], [31], [29], [45] and social influence analysis [1], [6], [10], [18], [24], [27], [49], its objective and methodology are different from these areas of work. The diffusion effect between links influences the evolution of the network structure, while network structure also affects the diffusion strength between links. Existing research on link prediction usually focuses on finding different factors that affect a link to be formed. Network formation models the network evolution to satisfy macroscopic properties such as heavy tails and

- *Jing Zhang is with the Department of Computer Science and Technology,Tsinghua University, Beijing, China, 100084. E-mail:zhangjing12@mails.tsinghua.edu.cn*
- *Zhanpeng Fang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084. E-mail:fzp13@mails.tsinghua.edu.cn*
- *Wei Chen is with Microsoft Research, Beijing, China, 100080. E-mail:weic@microsoft.com*
- *Jie Tang is with the Department of Computer Science and Technology,Tsinghua University, Beijing, China, 100084. E-mail:jietang@tsinghua.edu.cn*

1. Though Twitter does not explicitly provide a function to display such a "following" message to B, B could still have a chance to discover C via browsing A's retweets of C's messages, directly checking the followee list of A, or being recommended of C through the recommendation function of Twitter.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering

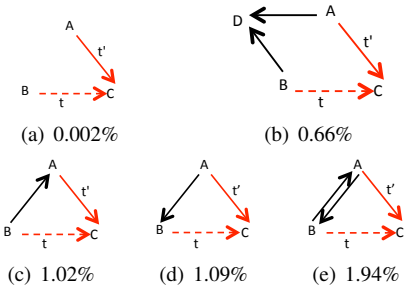


Fig. 1. The ratio of B following C under different triadic configurations. In each triad, the black edge represents a preexisting link; the solid red edge represents a link added at time t', and the dashed red edge represents a possible link created at time t, with $0 \leq t - t' \leq \delta$.

small diameters. Both of them do not consider the dynamic diffusion effect between links (i.e., one link triggers another link in a short time interval). Social influence analysis either aims to verify the existence of social influence [1], [6] or tries to quantify the strength of the influence [18], [27], [49]. However, they focus on studying the influence between users, while we aim to study how the formation of links are influenced and propagated.

In this paper, we study the diffusion phenomenon of "following" links in microblogging networks. In particular, we try to answer the following questions: How to model the diffusion of the links in a network? What are the evident patterns that facilitate such diffusion process? How to quantitatively learn the diffusion strength between links in different patterns? How can the study of link diffusion help real applications? Properly addressing these questions is not an easy task. Although the links are generated by users' behaviors, the diffusion of these links cannot be directly modeled like other actions (e.g., purchase of a product) because the link diffusion process is closely correlated to the dynamic network structure. Thus a principled methodology to model the diffusion process is necessary. We address these issues in this paper and make the following contributions:

1) We propose a "following" link cascade model to depict the link diffusion process through considering the time delay and different diffusion patterns.
2) We find significant triadic structures that affect the diffusion process. For example, a two-way relationship between two users can better (+1%) trigger the propagation of new links than a one-way relationship.
3) We learn the diffusion strength in different triadic structures by maximizing an objective function based on the **"following" link cascade model**.
4) We consider two specific influence maximization applications, follower maximization and followee maximization, to demonstrate the usefulness of the proposed model.
5) We conduct experimental evaluations using a large twitter dataset and a weibo dataset[2]. The results indicate that our method is able to model the dynamic formation of links in microblogging networks more closely than other link prediction or network formation methods.

Organization Section 2 proposes the link diffusion model. Section 3 introduces the dataset and analyzes the link diffusion patterns in different triadic structures. Then, in Section 4, we present how to learn the diffusion strength by maximizing an objective function based on the link diffusion model. Section 5 introduces two applications of the link diffusion model. In Section 6, we show experimental results that validate the effectiveness of our model. Section 7 reviews the related work and Section 8 concludes the paper.

2 "Following" Link Cascade Model

In this paper, we only focus on network structure and ignore user profile features. We propose a "following" link cascade model (FCM) to simulate the diffusion process from one link to its neighboring links in a network. If two links share a common end point, we say that they are *neighboring links* of each other. We ignore the diffusion between links without adjacent relationship because the diffusion between neighboring links is more natural. Figure 1 shows that the two links without direct adjacent relationship (Figure 1(a)1(b)) present an insignificant diffusion effect compared to the neighboring links, i.e., two links in a triadic structure. In addition, the triadic closure is a fundamental concept in social network theory. This was made popular by Grannovetter et al. [19] when studying weak ties and treated by Easley et al. [15] as a useful simplification of reality for understanding and predicting networks. We can therefore safely say that the triad is a basic structure in studying networks.

We first represent a dynamic microblogging network as $G = (V, E, t)$, where each node $v \in V$ represents a user and each directed edge $e_{uv} \in E$ represents a "following" link pointing from user u to v. For each link e pointing from A to B, we call A the follower end point and B the followee end point. Function $t : E \rightarrow \mathbf{N} \cup \{\perp\}$ labels each edge with the timestamp at which the link was formed. Notation $t(e_{uv}) = n \in \mathbf{N}$ indicates that e_{uv} was formed at timestamp n, where time is counted in units of days. Notation $t(e_{uv}) = \perp$ represents the fact that e_{uv} has been formed a long time ago, and its exact formation day is not captured. Henceforth, we abbreviate $t(e_{uv})$ as t_e. We use e' to represent the link from A to C, and use e to represent the potential link from B to C.

Assumption 1: Diffusion effect between links decays over time.

To model the decay assumption, we use a *discovery probability* $g_{e'e}$ to model how early B discovers that the link of A following C is formed, and a *diffusion probability* $h_{e'e}$ to describe how likely it is that A following C affects the formation of B following C after the discovery. Thus, the diffusion strength is represented by discovery probability and diffusion probability together.

2. The most popular Chinese microblogging service.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering

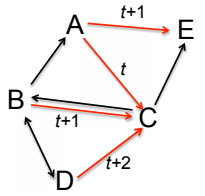


Fig. 2. Diffusion process of "following" links.

The diffusion process of the FCM model unfolds in discrete steps according to the following stochastic rule. When a link e' is added at time t', at each time slot from time t' to $t' + \delta$, the follower end point of e may discover the link e' with discovery probability $g_{e'e}$, and once discovered, there is one chance at that time that e' influences the formation of e with the diffusion probability $h_{e'e}$. If failed, e' will have no chance to activate e again. In other words, the time delay λ for discovery follows a geometric distribution with parameter $g_{e'e}$ and after discovery there is one chance at time $t' + \lambda$ that e' could activate e. The reason we put an upper bound δ on the delay distribution is to follow the observation made in Section 3 that after some time slots (e.g. $\delta = 7$ days), the influence effect almost diminishes. When multiple links activate e, e is activated at the time of the first successful attempt.

Even though our FCM model only allows one chance of activation of a link at the time of the discovery of a neighboring link e', the discovery event allows different interpretations, which compensate the seemingly restricted one-time activation attempt. The discovery of e' by B (the follower end user of e) could mean either that B logs in and first notices that A follows C, or it could also mean that B already notices A following C and already reads some tweets A retweeted from C. The latter can be viewed as B learning more about C before she really "discovers" the value of C and decides whether to follow C. Moreover, the meeting probability between two persons proposed in [1] can be also explained as the discovery probability in our paper. In summary, there can be several interpretations on the discovery mechanism. Essentially, all of them can be understood as a delay on the diffusion process.

Figure 2 shows an example of link diffusion process. e_{AC} is formed at time t, the follower end point B of e_{BC} discovers e_{AC} at $t + 1$ and then e_{AC} affects e_{BC} to be formed at $t+1$. Subsequently, e_{BC} affects e_{DC} to be formed at time $t + 2$. Similarly, the follower end point A of e_{AE} discovers e_{AC} at time $t + 1$ and then e_{AC} affects e_{AE} to be formed at $t + 1$.

The model can be viewed as a variant of the time-delayed independent cascade (IC) model [20], [24], [27]. The IC model diffuses the influence between users in a static network, while our model diffuses the influence between links, which causes the evolution of the network structure. We use the word "influence" for convenience, however, one should not interpret it directly as social influence between users in a social network.

Our goal is to investigate how likely the formation of one link influence the formation of the neighboring links in a short time period. Specifically, we aim at measuring the diffusion strength, i.e., the discovery probabilities and the diffusion probabilities in our model. We categorize the diffusion in a triadic structure into two main categories:

Follower diffusion: If a link of A following C is formed at time t', and this link triggers the formation of B following C at time t with $t' \leq t \leq t' + \delta$, where B is A's follower or followee by time $t' - 1$, we say that there exists an influence of the formation of e_{AC} on the formation of e_{BC}. Since this influence generates further followers of C from a new user, we call it *follower diffusion*[3].

Figure 3(a) shows a typical example of the follower diffusion in the triad (A, B, C) with a preexisting link e_{BA} by time $t'-1$. Later, we will consider other triadic structures of follower diffusion, e.g. by time $t' - 1$, B was already a followee of A, or C was already a follower of B, etc. Symmetrically, we define followee diffusion:

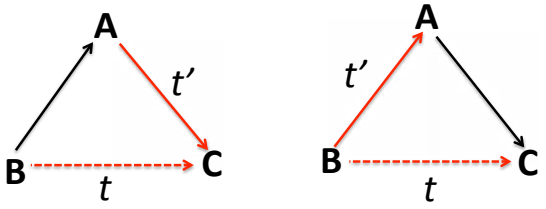
Followee diffusion: If a link of B following A is formed at time t', and this link triggers the formation of B following C at time t with $t' \leq t \leq t' + \delta$, where C is A's followee or follower (by time $t'-1$), we say that there exists an influence of the formation of e_{BA} on the formation of e_{BC}. Since this influence generates further followees of B from a new user, we call it *followee diffusion*. Figure 3(b) shows a typical example of followee diffusion.

The two categories are different. In follower diffusion, the two newly added links share the same followee end point, which results in different users "following" the same user. In followee diffusion, the two newly added links share the same follower end point, which results in different users being followed by the same user. According to the difference, we can design different applications for the two categories. For example, follower diffusion can be used by a user to target a small set of potential followers in order to attract more followers. Followee diffusion, on the other hand, can be naturally used to enhance followee recommendation. Traditional recommender systems mainly consider how likely it is that the recommended users will be accepted by the target user. Based on followee diffusion, the system can instead recommend users who can trigger the maximal number of users the target user will follow subsequently. Traditional recommender systems only focus on encouraging users to follow the one-step recommended followees, while the followee diffusion-based recommendation focuses on triggering more than one step acceptances by the target user. In summary, both follower diffusion and followee diffusion create opportunities of generating more links, which is important for the healthy growth of social networks.

Note that there also exists some other categories of link diffusion patterns, e.g., the diffusion of link e_{AC} to link e_{CB}. However, the physical meanings of those categories are not natural to be understood and explained. Therefore,

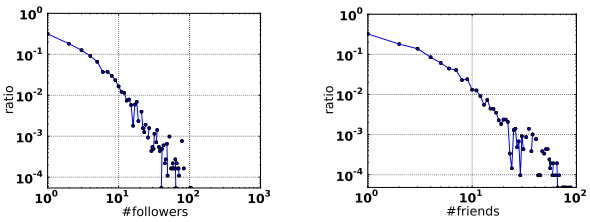
3. To be exact, what is described here is one step of follower diffusion, while we also allow multiple steps of follower diffusion, such as D "following" C triggered by B "following" C. Followee diffusion is equally allowed to have multiple steps.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering



(a) Follower diffusion (b) Followee diffusion

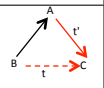
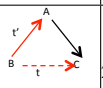
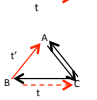
Fig. 3. Two categories of "following" diffusion patterns.



(a) Follower distribution (b) Followee distribution

Fig. 4. Statistics of the crawled Twitter dataset.

TABLE 1
Triad statistics in Twitter.

	Follower Diffusion					Followee Diffusion				
	△	$\|C_{\triangle}\|$	$\|C^{+}_{\triangle}\|$	$r_{\triangle}$		△	$\|C_{\triangle}\|$	$\|C^{+}_{\triangle}\|$	$r_{\triangle}$	
1		22870	233	0.0102 ***	13		24162	2298	0.0951 ***	
2		22527	246	0.0109 **	14		62411	2293	0.0367 ***	
3		33122	642	0.0194 ***	15		63092	3985	0.0632 ***	
4		29830	100	0.0034	16		23099	2314	0.1002 ***	
5		2370	3	0.0013	17		25049	324	0.0129 ***	
6		7283	76	0.0104 *	18		65219	3469	0.0532 ***	
7		116	3	0.0259	19		428	315	0.7360 ***	
8		883	77	0.0872	20		5729	2300	0.4015 ***	
9		730	71	0.0973	21		4372	3427	0.7839 ***	
10		666	46	0.0691 **	22		3889	3267	0.8401 ***	
11		389	42	0.1080 ***	23		8145	3280	0.4027 ***	
12		970	180	0.1856 **	24		27076	23310	0.8609 ***	

Notes: * p-value < 0.05, ** p-value < 0.01, *** p-value < 0.001

we only consider the two defined diffusion categories.

3 Data and Observations

In this section, we employ Twitter to analyze the diffusion effects in the two defined diffusion categories in Section 2. We provide basic observations on the significance of the diffusion patterns and diffusion decay phenomenon. These observations both provide intuitive understanding on the diffusion mechanism, and help guiding learning the parameters in our model.

3.1 Data Collection

We use the Twitter dataset from [41] in our study. Specifically, the dataset is crawled in the following way. To begin with the collection process, we selected the most popular user in Twitter, i.e., "Lady Gaga", and randomly collected 10,000 of her followers. We took these users as seed users and collected all followers of these users by traversing "following" links, which produced in total 13,442,659 users and 56,893,234 links. We then monitored the change of the network structure from 10/12/2010 to 12/23/2010. From the crawled data, we extract a complete subnetwork, in which the links between all users are recorded. The complete subnetwork consists of 112,044 users and 443,399 links between them, in which there are 25,530 dynamic links from 10/12/2010 to 12/23/2010 .

Figure 4 shows the follower distribution and followee (the users being followed) distribution of the crawled complete subnetwork. Both the distributions are drawn in log-log scale. We can see that the two statistics both follow the power law distribution.

3.2 Observations

For both follower and followee diffusion, we define 12 respective categories of different triad structures. Table 1 lists the 24 triads and their statistics in the above Twitter network. Triads 1 to 12 represent follower diffusion and triads 13 to 24 represent followee diffusion.

Each triad structure contains links with different directions and timestamps: (a) the black edge without timestamp represents a preexisting link; (b) the solid red edge with timestamp t' represents a link added at time t', and is the cause of the link diffusion under investigation; and (c) the dashed red edge with timestamp t represents the effect of the diffusion to be observed, and it may or may not be presented in an actual triad. The timestamps satisfy $0 \leq t - t' \leq \delta$ (δ is set as 7 days according to the following observations).

In Table 1, notation $C_{\triangle}$ denotes the actual triadic instances with triadic structure $\triangle$, where $C^{+}_{\triangle}$ are the instances with B following C within $[t', t'+\delta]$. $|C_{\triangle}|$ is the number of triadic instances with regard to $\triangle$. Notation $r_{\triangle}$ represents

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering

the rate of B following C in a specific triad, which is calculated as :

$$r_{\triangle} = \frac{|C^{+}_{\triangle}|}{|C_{\triangle}|}. \quad (1)$$

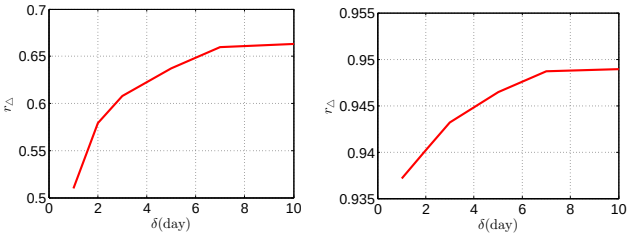
We analyze the diffusion effects in different triadic structures via the following two types of statistics:

- **Pattern significance:** Are the patterns in Table 1 significant or not?
- **Diffusion decay:** Is the diffusion effect between links decay over time?

Pattern significance We conduct a randomization test to demonstrate the significance of the triadic patterns in Table 1. Randomization test is a model-free, computationally intensive statistical technique for hypothesis testing [17], [44]. The key idea is to define a null hypothesis and a test statistic. The main steps include: firstly, compute some test statistic using the set of original observations; secondly, carry out the random shuffle according to the null hypothesis a large number of times, and compute the test statistic for each random data; finally, by the law of large numbers, the permutation p-value is approximated by the proportion of randomly generated values that exceed or equal the observed value of the test statistic. If p-value ≤ 0.05, the null hypothesis is rejected. In our setting, the null hypothesis is defined as: the formation of neighboring links is temporally independent of one another. Under this null hypothesis, we randomly shuffle the formation times of all the newly formed links, and use the test statistic as the rate defined in Eq. (1).

For each triadic structure, we set δ as 7 days and conduct random shuffle 10000 times. In followee diffusion category, the calculated p-values are all 0, which indicate that the followee diffusion patterns are significant. In follower diffusion category, the calculated p-values for triads 4, 5, 7, 8, 9 are larger than 0.05, which denote these patterns are insignificant. The p-values are shown in table 1. In triads 7, 8, 9, the most probable reason of B "following" C is C "following" B before and B "following" back, rather than the influence from A "following" C. However, triads 9, 10, 11, 12 are more significant because there are more two-way links in a triadic closure, which can strengthen the diffusion effect from e_{AC}. In triads 4, 5, 6, the most probable reason why A follows C is "following" back, and thus C is more likely to be an ordinary user. Therefore, the diffusion effect of allowing C to be followed by others is relatively week. However, triad 6 is more significant because it has more two-way links. In triads 1, 2, 3, the link e_{AC} is formed most probably due to the "following" behavior from ordinary user to celebrity user. Thus the diffusion effect in triads 1,2,3 are much stronger. Henceforth, we ignore triads 4, 5, 7, 8, 9 in the following analysis and experiments.

Diffusion decay To observe the effect of link diffusion over time, we vary the value of δ as $1, 2, 3, 5, 7$ and 10 days. For each value of δ, we average the observed $r_{\triangle}$ in follower diffusion category and followee diffusion category respectively, and show the results in Figure 10(a). From the results, we can see that the increasing rate of $r_{\triangle}$ becomes slower over time. When δ is larger than 7 days, $r_{\triangle}$ almost stops increasing, which implies that the diffusion effect persists for about 7 days. We also notice that compared with follower diffusion, the rate $r_{\triangle}$ in followee diffusion is high in the first day and later it increases very slowly. This is because, in follower diffusion (Figure 3(a)), there should be some mechanism for B to discover A following C, such as via browsing A's retweets of C's messages. While in followee diffusion (Figure 3(b)), although B may discover A following C according to the system recommendation after a period of time, B can also discover C immediately after following A via browsing A's retweets of C's messages. Thus, the formation of B following C in followee diffusion is easier than that in follower diffusion. For simplicity, we uniformly choose δ as 7 days in the later analysis and experiments.



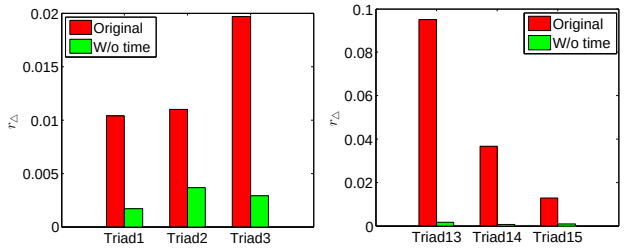
(a) Follower diffusion (b) Followee diffusion

Fig. 5. Diffusion decay in Twitter. X-axis: δ (1,2,3,5,7 and 10 days)

We further conduct another analysis to show that the diffusion effect decays over time. For each triad, we calculate $r_{\triangle}$ with δ as 7 days. For comparison, we change the solid red edge to a preexisting link (i.e., $t = \perp$), instead of a newly formed link, and recalculate $r_{\triangle}$. We select 3 representative triads for each category and report the comparative results in Figure 6. From the results, we can see that in both categories, $r_{\triangle}$ in the original triads (neighboring links are formed within a short period) are significantly higher than those in the comparative triads (neighboring links are formed a long period from each other), which indicates that the diffusion effect between neighboring links decays over time. Moreover, Figure 6(b) presents a higher rate difference than Figure 6(a), which also indicates that the decay effect in followee diffusion is more significant than that in follower diffusion.

Other observations In the category of follower diffusion, we discover that the diffusion strength is more significant when there is a two-way relationship between A and B. We divide triads in this category into four groups (triads 1-3, triads 10-12). In each group, the first triad only has a one-way relationship e_{BA}, the second triad only has a one-way relationship e_{AB}, and the third triad has both e_{AB} and e_{BA}. The statistics in Table 1 show that the triads with a two-way relationship between A and B exert a stronger effect on the formation of B following C than those with only a one-way relationship between A and B (about +1%). This can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering



(a) Follower diffusion (b) Followee diffusion

Fig. 6. Diffusion decay in Twitter. 'W/o time' denotes the results with solid red edges being changed to preexisting links.

TABLE 2
Notations.

SYMBOL	DESCRIPTION
$\mathcal{E}$	the links with observed formation times
S_e	the neighboring links of e added within $[t_e-\delta, t_e]$
R_e	the neighboring links of e not added before $t_e+\delta$
$p(e\|S_e)$	the probability of e being formed at t_e given S_e
$x_{e'e}$	the probability of e' activating e successfully at t_e
$y_{e'e}$	the probability of e' not activating e within $[t_{e'}, t_e]$
$\triangle$	the triadic structure
$h_\triangle$	the diffusion probability of triadic structure $\triangle$
$g_\triangle$	the discovery probability of triadic structure $\triangle$
$C^+_\triangle$	the activated triadic instances with structure $\triangle$
$C^-_\triangle$	the inactivated triadic instances with structure $\triangle$

be explained by the intuition that two-way relationships are much more likely to be actual "social" relationships, rather than "celebrity following", and thus can better facilitate the diffusion of "following" links.

In the category of followee diffusion, we discover that the diffusion effect is more significant when there exists a one-way relationship from A to C. We also divide the triads in this category into four groups (triads 13-15, triads 16-18, triads 19-21, triads 22-24) similar to the division for follower diffusion category. We see that for all the groups, the triads with a one-way relationship from A to C exert a stronger effect on the formation of B following C than those with only a one-way relationship from C to A (+3-40%). This can be explained as a user discovery process: when a link of B following A is formed, this may trigger B to discover A's followee C through immediately browsing A's retweets of C's messages, and A's interest in C may indicate that B would be also interested in C.

Summary We have seen that firstly, the formation of two links in some triads is temporally dependent; secondly, the diffusion effect between two links decays over time; thirdly, a two-way relationship between two users can trigger more links (+1%) than a one-way relationship and a relationship directed from A to C improves the diffusion likelihood from A following C to B following C (+3-40%).

4 Model Learning

In this section, we learn the diffusion strength in different triadic structures in Table 1. We define an objective function based on FCM and propose an EM algorithm to solve it.

Likelihood function Based on the FCM model, we define a likelihood function to describe the generative probability of all the links in the network. The objective is to estimate the parameters $\theta = \{h_{e'e}, g_{e'e}\}$ through maximizing the likelihood function. Actually, any link only forms once and there is no more than one instance associated to one pair (e', e). Directly estimating the diffusion probability $h_{e'e}$ and discovery probability $g_{e'e}$ for (e', e) therefore results in trivial solution (If e is activated, $h_{e'e}=1$ and $g_{e'e}=1$; if e is inactivated, $g_{e'e}=0$). Instead, according to the observations in Section 3.2, we classify all pairs $\{(e', e)\}$ into the 24 triadic categories (see Table 1). Specifically, a newly added link e at time t and one of its potential neighboring links e' (may form at time t' or not) must satisfy one kind of triadic structures in Table 1. In this way, the parameters $\theta = \{h_{e'e}, g_{e'e}\}$ to be estimated are reduced to 24 parameters $\theta = \{h_\triangle, g_\triangle\}$, where $h_\triangle$ is the diffusion probability and $g_\triangle$ is the discovery probability of a triadic structure $\triangle$. Therefore, the problem of link activating link can be viewed as triad activating link.

The difference between our likelihood function and previous IC model-based likelihood function is that, the influence diffuses from one link to another link instead of the diffusion between nodes in the network. We incorporate the triadic structures between links into the likelihood function and estimate the diffusion strength associated with different triadic structures, which is different from estimating the diffusion strength between two nodes. The symbols used in the section are summarized in Table 2.

We derive the likelihood function based on FCM:

$$\mathcal{L} = \prod_{e\in\mathcal{E}} \left\{ p(e|S_e) \prod_{e'\in R_e} y_{ee'} \right\}. \tag{2}$$

In Eq (2), we formalize the formation of each newly added link $e \in \mathcal{E}$, where $\mathcal{E}$ is the set of links with observed formation times. The formation of a link is correlated with all its recently added neighboring links. As shown in Figure 7, the formation of e_{BC} is jointly influenced by multiple neighboring links e_{BA_1}, e_{BA_2}, $\cdots$, e_{A_nC}. We denote the formation probability of e at time t_e as $p(e|S_e)$, where S_e is the set of neighboring links of e added within $[t_e-\delta, t_e]$. Notice that e being formed at t_e implies that the first time of a link $e' \in S_e$ activating e is t_e.

The formation probability $p(e|S_e)$ is calculated by the joint influence from S_e. A link e is successfully added if at least one of its recently added neighboring links $e' \in S_e$ successfully activated it. However, we do not know which link actually succeeds. Thus, when e is activated (i.e., added), we represent the statuses of e's recent neighboring links by a latent binary vector $\vec{\alpha}_{S_e} = \{\alpha_{e'}\}_{e'\in S_e}$, with each element $\alpha_{e'}=1$ denoting e' tried to activate e and succeeded, and $\alpha_{e'}=0$ denoting e' failed to activate e within $[t_{e'}, t_e]$. For every possible assignment of $\vec{\alpha}_{S_e}$, there is at least one element $\alpha_{e'}$ equaling 1. There is a total of $2^{|S_e|}-1$ possible assignments. According to the law of the total probability, we write $p(e|S_e)$ as:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering

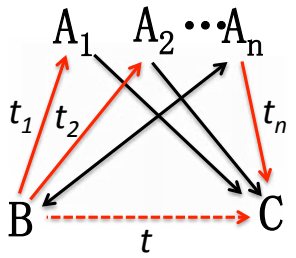


Fig. 7. Diffusion from multiple neighboring links. One link e_{BC} added at time t is jointly influenced by e_{BA_1}, e_{BA_2}, $\cdots$, e_{A_nC} added at time t_1, t_2, $\cdots$, t_n, where triads BA_1C and BA_2C satisfy followee diffusion pattern, and triad BA_nC satisfies follower diffusion pattern.

$$p(e|S_e) = \sum_{\vec{\alpha}_{S_e}} p(e|\vec{\alpha}_{S_e}) p(\vec{\alpha}_{S_e}), \tag{3}$$

Since the actual activation statuses cannot be observed, we assume $p(\vec{\alpha}_{S_e})$ to be uniformly distributed under the assumption of maximal entropy, and focus on calculating conditional probability $p(e|\vec{\alpha}_{S_e})$. Following the assumption of the IC model [20], [24], [27], each neighboring link $e' \in S_e$ activates e independently. Thus, the joint probability $p(e|\vec{\alpha}_{S_e})$ under one possible assignment of $\vec{\alpha}_{S_e}$ is represented as :

$$p(e|\vec{\alpha}_{S_e}) = \prod_{e' \in S_e} x_{e'e}^{\alpha_{e'}} y_{e'e}^{1-\alpha_{e'}}, \tag{4}$$

where $x_{e'e}$ denotes the probability of e' activating e at time t_e successfully. As described in Section 2, when e' is formed at $t_{e'}$, the time delay for the follower end point of neighboring link e to discover e' follows a geometric distribution with parameter $g_\triangle$[4]. After discovery there is one chance at that time that e' could activate e. According to the model, $x_{e'e}$ is defined as follows:

$$x_{e'e} = h_\triangle g_\triangle (1-g_\triangle)^{t_e - t_{e'}}. \tag{5}$$

The notation $y_{e'e}$ denotes the probability of e' not activating e within $[t_{e'}, t_e]$, which means that at each time slot from $t_{e'}$ to t_e, e' does not activate e successfully. In other words, $y_{e'e}$ is the probability that e' activates e after t_e[5]:

$$\begin{aligned} y_{e'e} &= 1 - h_\triangle g_\triangle \sum_{t=t_{e'}}^{t_e} (1-g_\triangle)^{t-t_{e'}} \\ &= h_\triangle (1-g_\triangle)^{t_e - t_{e'}+1} + (1-h_\triangle). \end{aligned} \tag{6}$$

For each newly added link, we also formalize its effect on its unformed neighboring links. A newly added link $e \in \mathcal{E}$ has a chance to activate its unformed neighboring links within the next δ time interval. It fails to activate a neighboring link e' with probability $y_{ee'}$ if $e' \in R_e$, where R_e is the set of neighboring links of e not added before $t_e + \delta$. The probability $y_{ee'}$ is also calculated using Eq. (6), while replace $t_{e'}$ and t_e with t_e and $t_e + \delta$ respectively.

Finally the log-likelihood function can be rewritten as:

$$\log \mathcal{L} = \sum_{e \in \mathcal{E}} \left\{ \log \sum_{\vec{\alpha}_{S_e}} \prod_{e' \in S_e} x_{e'e}^{\alpha_{e'}} y_{e'e}^{1-\alpha_{e'}} + \sum_{e' \in R_e} \log y_{ee'} \right\}.$$

EM algorithm We use an EM algorithm to learn the model parameters.

We introduce a posterior distribution $q(e|\vec{\alpha}_{S_e}) = \frac{p(e|\vec{\alpha}_{S_e})}{p(e|S_e)}$ and use Jensen's inequality to find a lower bound of the log-likelihood function:

$$\begin{aligned} \log \mathcal{L} &= \sum_{e \in \mathcal{E}} \left\{ \log \sum_{\vec{\alpha}_{S_e}} \hat{q}(e|\vec{\alpha}_{S_e}) \frac{p(e|\vec{\alpha}_{S_e})}{\hat{q}(e|\vec{\alpha}_{S_e})} + \sum_{e' \in R_e} \log y_{ee'} \right\} \\ &\geq \sum_{e \in \mathcal{E}} \left\{ \sum_{\vec{\alpha}_{S_e}} \hat{q}(e|\vec{\alpha}_{S_e}) \log \frac{p(e|\vec{\alpha}_{S_e})}{\hat{q}(e|\vec{\alpha}_{S_e})} + \sum_{e' \in R_e} \log y_{ee'} \right\}, \end{aligned}$$

where the notations with $\hat{}$ are the parameters of last iteration. The expression $\hat{q}(e|\vec{\alpha}_{S_e}) \log \hat{q}(e|\vec{\alpha}_{S_e})$ is only related to the parameters of last iteration, which can be viewed as a constant and ignored when maximizing the lower bound. We use $Q(\theta, \hat{\theta})$ to denote the simplified lower bound:

$$Q(\theta, \hat{\theta}) = \sum_{e \in \mathcal{E}} \left\{ \sum_{\vec{\alpha}_{S_e}} \hat{q}(e|\vec{\alpha}_{S_e}) \log p(e|\vec{\alpha}_{S_e}) + \sum_{e' \in R_e} \log y_{ee'} \right\}.$$

By plugging Eq. (4) into the above equation, we obtain:

$$\begin{aligned} Q(\theta, \hat{\theta}) = \sum_{e \in \mathcal{E}} \Bigg\{ \sum_{\vec{\alpha}_{S_e}} \hat{q}(e|\vec{\alpha}_{S_e}) \sum_{e' \in S_e} \big(& \alpha_{e'} \log x_{e'e} \\ & + (1-\alpha_{e'}) \log y_{e'e} \big) + \sum_{e' \in R_e} \log y_{ee'} \Bigg\}. \end{aligned} \tag{7}$$

By moving $\sum_{\vec{\alpha}_{S_e}} \hat{q}(e|\vec{\alpha}_{S_e})$ into the inner summation operation and replacing $\hat{q}(e|\vec{\alpha}_{S_e})$ with $\frac{p(e|\vec{\alpha}_{S_e})}{p(e|S_e)}$, we get:

$$\begin{aligned} Q(\theta, \hat{\theta}) = \sum_{e \in \mathcal{E}} \Bigg\{ \sum_{e' \in S_e} \big(& \hat{A}_{e'e} \log x_{e'e} \\ & + (1 - \hat{A}_{e'e}) \log y_{e'e} \big) + \sum_{e' \in R_e} \log y_{ee'} \Bigg\}, \end{aligned} \tag{8}$$

where $A_{e'e}$ is defined as:

$$A_{e'e} = \frac{x_{e'e} \prod_{d \in S_e \setminus \{e'\}} (x_{de} + y_{de})}{\hat{p}(e|S_e)}. \tag{9}$$

In our implementation, $p(e|S_e)$ is actually calculated as

$$p(e|S_e) = \prod_{e' \in S_e} (x_{e'e} + y_{e'e}) - \prod_{e' \in S_e} y_{e'e}, \tag{10}$$

4. To be exact, the subscript notation $\triangle$ should be $\triangle(e', e)$, denoting the particular triadic structure constructed by e' and e. For simplicity, we use notation $\triangle$ instead of $\triangle(e', e)$, since it is clear from the context.

5. For the sake of argument, if e' does not activate e, we can say it activates e at infinity, which is still after t_e.

instead of Eq. (3) to reduce the time complexity from $O(2^{|S_e|})$ to $O(|S_e|)$. In Eq (10), the first product $\prod_{e' \in S_e}(x_{e'e} + y_{e'e})$ is the probability that all edges in S_e activate e at or after time t_e, which indicates none of $e' \in S_e$ activating e before time t_e. The second product $\prod_{e' \in S_e} y_{e'e}$ is the probability that all edges in S_e activate e after time t_e, which indicates none of $e' \in S_e$ activating e before or at time t_e. Therefore, their difference is at least one $e' \in S_e$ activates e at time t_e while all activate e at or after t_e, i.e., t_e is the first time of e being activated by $e' \in S_e$.

Although $\log x_{e'e}$ is a linear combination of $\log h_\triangle$, $\log g_\triangle$ and $\log(1-g_\triangle)$, $\log y_{e'e}$ can not be expressed as such (see Eq. (6)). Therefore, we introduce another posterior distribution $B_{e'e}$ defined in (11) and find a lower bound for $\log y_{e'e}$:

$$\log y_{e'e} \geq \hat{B}_{e'e} \log h_\triangle (1-g_\triangle)^{t_e - t_{e'}+1} + (1-\hat{B}_{e'e}) \log(1-h_\triangle),$$

where $B_{e'e}$ is as defined as:

$$B_{e'e} = \frac{h_\triangle (1-g_\triangle)^{t_e - t_{e'}+1}}{h_\triangle (1-g_\triangle)^{t_e - t_{e'}+1} + (1-h_\triangle)}. \quad (11)$$

By plugging Eqs. (5) and (11) into Eq.(8), we obtain the final lower bound of the original log-likelihood function.

$$\begin{aligned} Q(\theta, \hat{\theta}) = \sum_{e \in \mathcal{E}} \Bigg\{ & \sum_{e' \in S_e} \Big\{ \hat{A}_{e'e} \log h_\triangle g_\triangle (1-g_\triangle)^{t_e - t_{e'}} \\ & + (1-\hat{A}_{e'e}) \big\{ \hat{B}_{e'e} \log h_\triangle (1-g_\triangle)^{t_e - t_{e'}+1} \\ & + (1-\hat{B}_{e'e}) \log(1-h_\triangle) \big\} \Big\} \\ & + \sum_{e' \in R_e} \Big\{ \hat{B}_{ee'} \log h_{ee'} (1-g_\triangle)^{\delta+1} \\ & + (1-\hat{B}_{ee'}) \log(1-h_{ee'}) \Big\} \Bigg\}. \end{aligned}$$

We differentiate $Q(\theta, \hat{\theta})$ with respect to each parameter $h_\triangle$ and $g_\triangle$ and set the partial differential to zero. The link pairs $\{(e', e)\}$ associated with a same triadic structure $\triangle$ are aggregated together. The parameters are calculated as follows:

$$h_\triangle = \frac{\sum_{(e',e) \in C_\triangle^+} \hat{D}_{e'e} + \sum_{(e',e) \in C_\triangle^-} \hat{B}_{e'e}}{|C_\triangle|}, \quad (12)$$

$$g_\triangle = \frac{\sum_{(e',e) \in C_\triangle^+} \hat{A}_{e'e}}{\sum_{(e',e) \in C_\triangle^-} \hat{B}_{ee'}(\delta+1) + \sum_{(e',e) \in C_\triangle^+} \hat{D}_{e'e}(t_e - t_{e'}+1)}. \quad (13)$$

In Eqs. (12) and (13), $C_\triangle^+$ and $C_\triangle^-$ are defined in Table 2. Notations $A_{e'e}$, $B_{e'e}$ and $D_{e'e}$ are the intermediate variables for easy description, where $A_{e'e}$ and $B_{e'e}$ are defined in Eqs. (9) and (11), and $D_{e'e}$ is defined as follows:

$$D_{e'e} = B_{e'e} + A_{e'e} - A_{e'e} B_{e'e}. \quad (14)$$

We summarize the algorithm in Algorithm 1.

Algorithm 1: Model Learning.

Input: A dynamic network $G = (V, E, t)$
Output: $\theta = \{h_\triangle, g_\triangle\}$

```
Initialize h_△ and g_△ with random value within (0,1);
repeat
    E-step : foreach e ∈ E do
        foreach e' ∈ S_e do
            Calculate x_{e'e} using Eq. (5);
            Calculate y_{e'e} using Eq. (6);
        foreach e' ∈ S_e do
            Calculate A_{e'e} using Eq. (9);
            Calculate B_{e'e} using Eq. (11);
            Calculate D_{e'e} using Eq. (14);
        foreach e' ∈ R_e do
            Calculate B_{ee'} using Eq. (11);
    M-step: for △ = 1 to 24 do
        Calculate h_△ using Eq. (12);
        Calculate g_△ using Eq. (13);
until Convergence;
```

5 APPLICATIONS

In this section, we introduce how to use the learned diffusion strength into two applications, follower maximization and followee maximization. The two applications aim at activating more links in a network. We mainly describe how to calculate the expected number of activated links under the assumption of FCM model.

Follower maximization Given a target user v, the goal of follower maximization is to find k initial followers S of v such that the number of new followers is maximized. We use a greedy algorithm [24] to solve the problem. The basic idea is to select the user in round i that maximizes the incremental followers of v. For each user $u \notin S$, the number of activated followers is estimated with R repeated simulations of FCM($S \cup \{u\}$) (Lines 3-8 in Algorithm 2), where FCM(S) returns the set of followers activated by users in S. FCM is a simulated link diffusion process (cf. Section 2 for details), where the links are diffused from the links pointing from the initial users in S to v according to the follower diffusion patterns. In FCM, the probability of e' activating another e is calculated by Eq. (5). After we get the optimal seed set S, the expected number of followers activated by S is estimated by running FCM(S) R times. In the application, we make an assumption that everyone in the seed set will accept the recommendation to follow the target user v with probability 1.0[6]. Although the assumption is not very practical, it does not affect the objective of verifying the effectiveness of our model in the application of influence maximization.

Followee maximization Followee maximization can be

6. The setting is only for the two applications, not for the link prediction task in the experimental section.

Algorithm 2: Follower/Followee Maximization.

Input: A network $G = (V, E)$, user v, seed size k
Output: Initial follower/followee S

```
Initialize S = ∅ and R = 10000;
for i = 1 to k do
    foreach u ∈ V\S do
        s_u = 0;
        for r = 1 to R do
            s_u += |FCM(S ∪ {u})|;
        s_u = s_u/R;
    S = S ∪ {argmax_{u∈V\S} s_u};
```

generally considered to be an extension of friend recommendation. The difference is that friend recommendation mainly focuses on the successes of one-step recommendations, while followee maximization tries to recommend initial one-step followees as seeds to maximize the total number of subsequent followees after the process of followee diffusion ends. Thus, followee maximization can ease the sparsity problem of the network. Specifically, given one target user v, the goal is to recommend k initial followees to v such that the total number of new followees accepted by v is maximized. This application uses the followee diffusion patterns and similar algorithm as follower maximization.

6 Experiments

In this section, we evaluate the proposed FCM in two dynamic networks of Twitter and Sina weibo[7] through the tasks of link formation and influence maximization.

6.1 Experimental Setup

We use two datasets. One is the twitter dataset described in Section 3. Another is a Sina Weibo dataset, which, similar to Twitter, allows users to follow each other. The Weibo dataset is crawled in the following way. To begin with, 10 random users were selected as seed users, and then their followees and followees followees were collected, which produced in total 96,882 users and 1,391,432 links. Then we monitored the dynamic changes of the links for the 96,882 users from 8/28/2012 to 9/29/2012 and obtained 30,562 new links.

For each dataset, we construct positive and negative instances from it. The links to be probabilistically generated are the dashed red edges with timestamp t in Table 1. These links may or may not be presented in actual triads. Links presented on the actual dataset are labeled as positive instances, while the others are treated as negative instances. Each positive or negative instance is associated with a list of features based on the empirical counts of the 24 triadic structures in Table 1. Diffusion strength, including the discovery and diffusion probabilities, can be learned from our model. Given the difficulty of directly evaluating the obtained values, we suggest to use the task of link prediction to verify the effectiveness.

We also incorporate diffusion strength into the applications of follower maximization and followee maximization to verify the effectiveness of our model. The goal is not to compare the efficiency of the algorithm of influence maximization, however, we can easily improve the efficiency if applying the methods proposed in [9], [10].

Evaluation metrics To quantitatively evaluate the proposed FCM for estimating the likelihood of a new "following" link, we divide the constructed positive and negative instances from the dataset into training and test set. Since the time dependence only exists between one instance and the neighboring links in its corresponding triadic features, the instances are independent with each other. Thus, we can perform 5-fold cross validation in terms of several alternative metrics. We use FCM to learn $\{h_\triangle\}$ and $\{g_\triangle\}$ in training data, and then estimate the formation probability $p(e|S_e)$ under the influence of recently added neighboring links S_e using Eq.(10) in test set. We first cast the task as a classification problem. The aim is to classify whether a given link e will be formed or not. FCM classifies that e will be formed if $p(e|S_e) > \tau$. We use Precision, Recall, F1-measure and AUC as evaluation measures, where Precision, Recall and F1-measure are set as the optimal values by enumerating different values of τ from 0 to 1 with an interval 0.1, and AUC is obtained by considering all the values of τ. We also cast the task as a ranking problem. The aim is to rank the candidate followee end points for each follower end point. We set candidate followee end points as those two hops away from the given follower end point. We use P@1 (Precision for the top 1 ranking result), P@2, P@5, P@10 and MAP (Mean Average Precision) for evaluating the ranking followee list for a given follower end point and average the metrics for all the follower end points together. The ranking task is to find which followee candidates have the highest probabilities to be followed. FCM naturally calculates a formation probability $p(e|S_e)$ for each candidate link and can easily apply the probabilities for ranking.

Comparison methods We compare our model with several alternative methods. The first category of methods we compare with is based on classification.

Basic: Determines that a link will be definitely formed if it is the edge to be predicted in the 24th triadic structure in Table 1 (i.e, all the three links will become two-way links if the link is formed).

SVM: Uses the same 24 kinds of triadic structures as features and employs SVM-light to train and predict the formation of links.

LRC: Uses the same 24 kinds of triadic structures as features and leverage a logistic regression classification model [30] to train and predict the formation of links.

The second category of methods we compare with is based on ranking.

Collaborative Filtering(CF): Leverages the existing collaborations to make the prediction. Given a follower end

7. The most popular Chinese microblogging service.

point u, we need to find the most possible users that u will make links to them. The basic idea is that if a user u has the similar tendency as a user w, u is then likely to follow the same user as w. We employ a memory-based collaborative filtering algorithm [12], in which the score of u following v is calculated using the following formula:

$$CF\_score(u,v) = \sum_w I(w,v) sim(w,u),$$

where $sim(w,u)$ is the similarity between the users w and u, e.g., cosine similarity based on common followees; the indicator variable $I(w,v)$ is 1 if user w has followed v and 0 otherwise. We rank all the candidates $\{v\}$ to a query user u based on $CF\_score(u,v)$.

SimRank: Calculates the similarity between the given follower end point u and the candidate followee end point v by averaging the similarity between all pairs of their followees [22]. Then the candidates $\{v\}$ to a query follower end point u are ranked based on the similarity.

Katz:: Calculates the similarity between the given follower end point u and the candidate followee end point v by summing over all possible paths from u to v. To improve the efficiency, we only consider the paths with length less than 4. Katz is mentioned as the best link predictor in [35].

Random-random model(RR): Generate networks by proposing a triangle-closing model [29]. The generative process is, when a given follower end point u decides to add a link to some candidate followee v, u first selects a neighbor w uniformly at random, and w then selects a neighbor v uniformly at random. The link e_{uv} is then created and the triangle (u,w,v) is closed. According to the link generation model, the score of u following v is calculated by:

$$RR\_score(u,v) = \frac{1}{|F(u)|} \sum_w I(u,w) I(w,v) \frac{1}{|F(w)|},$$

where $|F(u)|$ means the number of users being followed by user u, and $I(u,w)$ is 1 if user u has followed w and 0 otherwise. We rank all the candidates $\{v\}$ to a query user u based on $RR\_score(u,v)$.

Preferential attachment with communities (PAC): Generate networks by proposing a directed closure process [45]. When a given follower end point u decides to add a link to some candidate followee end point v, with probability β, user u will choose to follow a user from the same community as u; with probability $1-\beta$, u will choose to follow a random user. With probability α, user v will be followed preferentially (i.e., at random from a probability distribution which weights nodes by their current indegree) and with probability $1-\alpha$, user v will be followed uniformly at random. According to the link generation model, the score of u following v is calculated by:

$$PAC\_score(u,v) = \beta\big(\alpha \frac{|N(v)|}{\sum_{v\in C(u)}|N(v)|} + (1-\alpha)\frac{1}{|C(u)|}\big) + (1-\beta)\big(\alpha \frac{|N(v)|}{\sum_{v\in V}|N(v)|} + (1-\alpha)\frac{1}{|V|}\big),$$

where $|N(v)|$ means the number of followers of user v. $C(u)$ is the collection of users belonging to the same community of user u, where the communities are detected initially by the algorithm [11]. V is the collection of followee candidates of user u, which is set as the followees two hops away from user u. MAP is set as the optimal value by enumerating different values of α and β from 0 to 1 with an interval 0.1.

Our work is targeted more general than the recommendation task. So we include baseline methods and evaluation metrics related to link prediction and network formation beyond those typically used for recommendation tasks. For example, several researchers use SVM [21], logistic regress [30], SimRank [35], Katz [35], [52], or preferential attachment [35], [52] to predict links. The two baselines RR and PAC are two advanced preferential attachment models, which consider the formation of triadic closures and perform better than the original preferential attachment as reported in [29], [45].

In addition, we evaluate whether the discovery and diffusion probabilities estimated by our model can improve the performance of link prediction on top of existing features. We adopt the same features and method used in [37]: The features are out-degree based common neighbors, Jaccard's coefficient, Adamic/Adar measure, preferential attachment, unweighted Katz and PropFlow, and the method is random forest. We treat $p(e|S_e)$ output by our model as an additional feature of random forest. In this comparison, we remove the constraint in the previous comparisons that each link to be predicted should be in some triadic structure. Then positive instances are defined as all the newly added links and negative instances are changed to unformed links two-hops away from the positive links [2], [53]. We change the experimental setting to prove the effectiveness of our model on top of existing features in a more general setting.

6.2 Performance Analysis

Table 3, Table 4 and Table 5 show the performance of link formation.

Higher performance From Table 3, we can see that the proposed FCM method clearly outperforms the baseline methods (+1-2% in terms of F1, +3-15 % in terms of AUC). The basic method only considers the 24th triadic structure, and thus under-performs our method because other triadic structures are ignored, though the 24th triadic structure is most significant ($r_{\triangle} = 86\%$ in Table 1). SVM and LRC also under-perform our model, which indicate that our model can better estimate the weights of different triadic structures. We explain why our method outperforms SVM and LRC in Figure 8.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
Performance of link prediction by different classification methods on Twitter. (%)

Model	Precision	Recall	F1-measure	AUC
Basic	74.09	54.66	62.90	77.00
SVM	**73.54**	56.18	63.69	75.28
LRC	63.37	**63.51**	63.43	88.67
FCM	70.58	60.04	**64.88**	**91.95**

TABLE 4
Performance of link prediction by different ranking methods on Twitter. (%)

Model	P@1	P@2	P@5	P@10	MAP
CF	47.69	44.24	35.78	30.26	61.55
SimRank	27.44	30.11	28.90	27.53	46.11
Katz	50.46	45.38	36.22	30.16	62.54
RR	54.57	46.87	36.11	29.99	64.53
PAC	47.69	40.85	33.36	28.99	59.68
FCM	**75.54**	**60.43**	**40.37**	**31.17**	**79.66**

As shown in Table 4, FCM also outperforms the baseline ranking methods (+15-33% in terms of MAP). CF, SimRank, and Katz only consider the static structure information (e.g., the common neighbors between two users u and v) and ignore the dynamic evolution of the network structure (e.g., a new link added at time t' is more likely to trigger its neighboring links to be formed within a short time frame after t). FCM captures the diffusion effect from neighboring links added shortly before, and thus obtains better performance. The network formation methods RR and PAC are proposed to fit the distributions of some macroscopic properties such as clustering coefficient and closure ratio. Besides, they also do not consider the temporal dependence between two links, and thus underperform our approach.

Table 5 shows the prediction results of random forest with and without the output of our model FCM, on top of existing features, on two datasets. We can see that the performance is significantly improved by adding the prediction result of FCM as a feature. This is because all the basic features only capture static structures and ignore temporal dependence between neighboring links.

On Weibo dataset, the number of triads is very rare except triad 1, 2, 3, 13, 14, and 15. These six triads are the ones containing much more one-way relationships than the other triads. This phenomenon reflects that, in Weibo, most of the links are formed because of "celebrity following" rather than actual "social" activity. While the discovery and diffusion effects in triad 13, 14, and 15 are stronger than those in triad 1, 2, and 3, which is almost consistent with the parameters learned on Twitter dataset. The experiments of comparing with the predefined classification and ranking-based methods on Weibo dataset present the similar results as on Twitter dataset, which are simply ignored in the paper. We only present the improvement on top of existing features on Weibo dataset.

Per-triad analysis We take a close look at the triadic structures individually to gain a better understanding of which factors affect performance, because as shown in Table 1, different triadic structures occupy different proportions and present different diffusion effects. Figure 8 shows the performances of per-triad breakdown on Twitter dataset. We aggregate the links associated with the same triadic structure together and show the F1-measure in each triad in Table 1 respectively. From the results, we can see that the performances of FCM on most triads are better than SVM and LRC. SVM and LRC perform as well as FCM on the triads presenting strong diffusion effects, such as triads 19-24. However, they perform poorer than FCM on the other triads presenting relatively weak diffusion effects, and the difference is most significant in triads 1, 2, 3, and 6, whose diffusion effects are weakest. SVM and LRC are both discriminative classification methods. Their performances are particularly affected by the distinguishing features, which may dominate the effects from the statistically insignificant triads. FCM is a generative model, which smooths the effects on the formation of links from different factors, and thus improves the performances on the statistically insignificant triads.

TABLE 5
Performance of link prediction on top of existing features on Twitter and Weibo. (%)

Dataset	Model	Precision	Recall	F1-measure	AUC
Twitter	W/o FCM	83.33	60.25	69.93	90.48
	W FCM	**97.82**	**79.67**	**87.82**	**95.50**
Weibo	W/o FCM	**97.20**	63.62	76.90	90.97
	W FCM	95.68	**69.68**	**80.64**	**92.15**

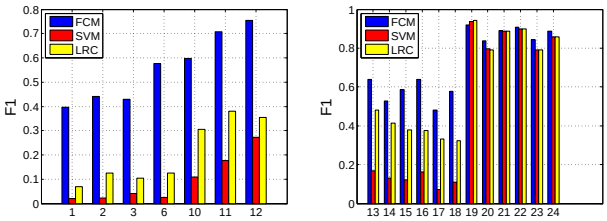
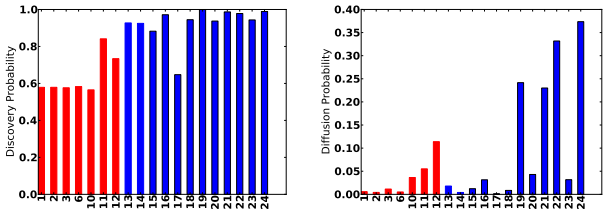


Fig. 8. Performance analysis in different triadic structures on Twitter. X-axis: triadic structure index. Y-axis: F1-measure

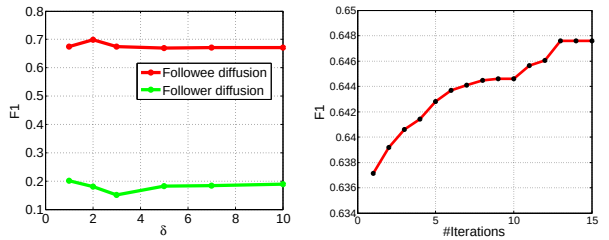
Model parameter analysis We report the learned discovery and diffusion probabilities in Figure 9, where the red bars represent the parameters learned for follower diffusion patterns, and the blue bars represent those learned for followee diffusion patterns. We can see from Figure 9(a) that the discovery probabilities learned for followee diffusion patterns are generally higher than follower diffusion patterns, which indicate that the discoveries in followee diffusion are easier than those in follower diffusion. The learning results are consistent with the observation of diffusion decay in Section 3, which shows that in followee diffusion, the link diffusion happens in almost the first day, while in follower diffusion, the link diffusion decays relatively slowly. Figure 9(b) also shows the learned diffusion probabilities are consistent with the rates $r_{\triangle}$'s in Table

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2407351, IEEE Transactions on Knowledge and Data Engineering



(a) Discovery Probabilities (b) Diffusion Probabilities

Fig. 9. Learned model parameters on Twitter. X-axis: triadic structure index. Y-axis: Discovery/Diffusion probability.



(a) Parameter analysis (b) Convergence analysis

Fig. 10. Performance analysis on Twitter.

1, which suggests that the diffusion effects in followee diffusion are stronger than those in follower diffusion.

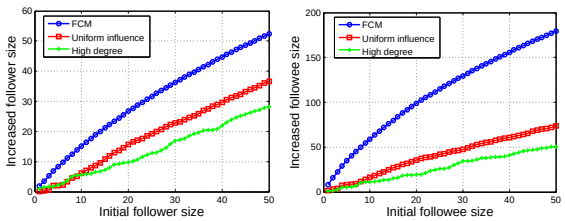
Delay analysis The time delay δ is the time interval between the formation of two links. We study how the parameter δ in FCM affects the performance of link prediction. Figure 10(a) plots the F1-measure of FCM by varying the value of δ in follower and followee diffusion categories respectively. We find that in both categories, the performance becomes relatively stable until the 7th day. This suggests that the diffusion effect persists for about 7 days, which is also consistent with the observations in Section 3. Therefore, we select δ as 7 days in most of the experiments and analysis in this paper.

Convergence analysis We further investigate the convergence of FCM. Figure 10(b) shows the convergence analysis results. We can see that FCM converges within about 13 iterations. This fast convergence property makes the algorithm efficient in large scale dataset.

6.3 Application Improvement

To further verify the effectiveness of the proposed model, we apply the learned discovery and diffusion probabilities to facilitate two applications: follower maximization and followee maximization as described in Section 5. Follower maximization is to select k initial followers to a target user v to maximize the number of subsequent triggered followers to v. Followee maximization is to recommend k initial followees to a target user v to maximize the number of subsequent triggered followees of v.

Our method uses the greedy algorithm in [24]. At each step, the algorithm selects a new follower/followee that can activate maximal number of followers/followees. We



(a) Follower maximization (b) Followee maximization

Fig. 11. Results for "following" influence maximization on Twitter. X-axis: the number of initial users. Y-axis: the number of newly activated users.

compare with two baselines: High degree and Uniform influence. High degree chooses initial followers/followees in order of decreasing degrees (in-degree and out-degree are considered respectively). Uniform influence also use the greedy algorithm in [24] to select the initial follower/followee except that the diffusion probabilities are set uniformly as 0.01. After the initial k followers/followees are selected, we simulate the diffusion process in the network starting from the seeds based on FCM.

From Figure 11, we can see that by using 50 seeds, FCM clearly activates much more followers/followees than the baseline methods (+43-250%). High degree may select the users that can not trigger many diffusions. Uniformly configured diffusion probabilities cannot accurately reflect the correlation between links, and thus weakens the maximization performance. Our method selects the initial followers/followees based on the learned discovery and diffusion probabilities. This demonstrates that distinguishing the diffusion effects in different triadic structures can effectively activate the followers/followees based on the diffusion process of FCM.

7 Related Work

Diffusion model and influence maximization SIR model [25] and SEIR model [33] are two well-known epidemic models that describe the transmission of communicable disease through individuals. To model how users influence each other in a social network, two diffusion models, namely Linear Threshold (LT) Model and Independent Cascade (IC) Model [24], are proposed. Recently, several new diffusion models considering different factors have been proposed, such as time-decayed Independent Cascade Model [8], [27], topic sensitive Independent Cascade Model and Linear Threshold Model [4], diffusion model considering positive and negative opinions together [7], and diffusion model considering friend and foe relationships together [34]. Corresponding algorithms have been designed to efficiently solve the influence maximization problem. The objective is to find k seeds in a network with maximal influence. Domingos and Richardson [14] are the first to study influence maximization as an algorithmic problem. Kempe et al. [24] take the first step to formulate influence maximization as a discrete optimization problem. Leskovec et al. [32] and Chen et al. [9], [10] make efforts

to improve the efficiency of influence maximization. In this paper, we propose a time-decayed link diffusion model, which is different from previous node diffusion models. We use the model in two applications of influence maximization to verify the effectiveness of our model. We do not target at the efficiency of the algorithm and simply leverage the well-known greedy algorithm.

Influence learning Influence learning is to quantify influence. From the perspective of the measured objects, we can classify the studies into: quantifying influence from topic level [39], [49] or sentiment level [48], quantifying the indirect influence using the theory of quantum cognition [47], measuring the external influence out-of-network sources [42] and measuring the individual, peer and group influence [50].This paper essentially quantifies the pairwise influence between two links.

Link prediction There are both unsupervised and supervised methods for link prediction. Liben-Nowell and Kleinberg [35] survey the unsupervised methods, including preferential attachment [43], random walk with restart [51] , SimRank [22] and Katz [23]. The intuition is that the more similar two users are, the more likely they will be linked. Supervised methods include local Markov Random Field [52], logistic regression [30], and supervised random walk [2]. Lichtenwalter et al. [36] propose a supervised framework to incorporate all the existing unsupervised methods as features and present significant improvement over other methods. The main differences between existing work and our work lie in three aspects. First, existing methods focus on the static reasons (such as common neighbors, social status, and structural balance) that may trigger a link, while we consider dynamic factor of link formation. Lee et al. [28] also predict links based on the temporal information. However, they mainly consider the two-way temporal information between node pairs, and ignore the temporal correlation in a triadic structure. Secondly, most existing work handles undirected links while we address the directed ones. Although Lou et al. [41] also study the directed links, which is only one special case of our patterns. Finally, link prediction mainly focuses on predicting whether a link will be formed or not without caring about how links are diffused under certain effects, while we study the diffusion mechanism of links.

Network formation Network formation aims at proposing network evolution models to generate networks [45], [3], [31], [29]. Barabasi et al. [3] propose preferential attachment to generate scale free networks. Leskovec et al. [31] discover densification powerlaw and shrinking diameters and propose forest fire models to obey these patterns. Leskovec et al. [29] propose a triangle-closing model and Romero et al. [45] propose a variant preferential attachment model to fit their discovered two properties for directed closures. They focus on modeling networks to satisfy macroscopic properties such as heavy tails and small diameters, while we discover microscopic patterns that affect the formation of a network and learn the strength of different patterns.

8 Conclusion

We study the diffusion mechanism of links in microblogging networks, which is also proposed as a challenge in [16], [40]. In the experiment of Twitter, we demonstrate that "following" links propagate according to the triadic structures with different diffusion strength. We mainly study two diffusion categories: follower diffusion and followee diffusion. In follower diffusion, a two-way relationship between the follower end points of two links can better trigger the diffusion of links than a one-way relationship. In followee diffusion, a relationship directed from A to C improves the likelihood that B follows C triggered by B following A. Incorporating the patterns, we propose a generative model to depict the diffusion process of the links and automatically learn the diffusion strength associated with different patterns. Experimental results show that our method by leveraging the learned diffusion strength is superior to alternative baselines.

For future work, it is intriguing to study other triadic structures in addition to those in Table 1, e.g., the triads with C "following" A at time t and the triads with A "following" B at time t. Those triads may represent negative influence between links. Designing and implementing randomized controlled experiments would also be an important direction to validate the causal relationship in the formation of links.

References

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD'08*, pages 7–15, 2008.
[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM'11*, pages 635–644, 2011.
[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
[4] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 81–90. IEEE, 2012.
[5] V. Belak, S. Lam, and C. Hayes. Cross-community influence in discussion fora. In *ICWSM'12*, 2012.
[6] 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.
[7] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincn, X. Sun, Y. Wang, W. Wei, and Y. Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM'11*, pages 379–390, 2011.
[8] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI'12*, 2012.
[9] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD'10*, pages 1029–1038, 2010.
[10] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD'09*, pages 199–207, 2009.
[11] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
[12] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW'07*, 2007.
[13] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *ICML'07*, pages 233–240, 2007.

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

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

[16] H. Fernau, F. V. Fomin, D. Lokshtanov, M. Mnich, G. Philip, and S. Saurabh. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014.

[17] P. I. Good. *Permutation, parametric and bootstrap tests of hypotheses.*, volume 3. Springer, 2005.

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

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

[20] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW'04*, pages 491–501, 2004.

[21] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM'06 workshop*, 2006.

[22] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD'02*, pages 538–543, 2002.

[23] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

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

[25] W. Kermack and A. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. Roy. Soc.*, A 115:700–721, 1927.

[26] C. W. ki Leung, E.-P. Lim, D. Lo, and J. Weng. Mining interesting link formation rules in social networks. In *CIKM'10*, pages 209–218, 2010.

[27] M. Kimura, K. Saito, K. Ohara, and H. Motoda. Learning information diffusion model in a social network for predicting influence of nodes. *Intell. Data Anal.*, 15:633–652, 2011.

[28] C. Lee, B. Nick, U. Brandes, and P. Cunningham. Link prediction with social vector clocks. *arXiv preprint arXiv:1304.4058*, 2013.

[29] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD'08*, pages 462–470, 2008.

[30] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW'10*, pages 641–650, 2010.

[31] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *In KDD*, pages 177–187. ACM Press, 2005.

[32] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD'07*, pages 420–429, 2007.

[33] M. Y. Li, J. R. Craef, L. C. Wang, and J. Karsai. Global synamics of an seir model with a varying total population size. *Math. Biosci.*, 160:191–213, 1999.

[34] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *WSDM'13*, pages 657–666, 2013.

[35] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[36] R. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *KDD'10*, pages 243–252, 2010.

[37] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *KDD'10*, pages 243–252, 2010.

[38] S. Lin, F. Wang, Q. Hu, and P. S. Yu. Extracting social events for learning better information diffusion models. In *KDD'13*, pages 365–373. ACM, 2013.

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

[40] Y. Liu, B. Wu, H. Wang, and P. Ma. Bpgm: A big graph mining tool. *Science and Technology*, 19(1):33–38, 2014.

[41] T. Lou, J. Tang, J. E. Hopcroft, Z. Fang, and X. Ding. Learning to predict reciprocity and triadic closure in social networks. *TKDD*, 7(2), 2013.

[42] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD '12*, pages 33–41, 2012.

[43] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Phys. Rev. E*, 64(2):025102, 2001.

[44] E. Noreen. *Computer Intensive Methods for Testing Hypotheses*. 1989.

[45] D. M. Romero and J. M. Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *ICWSM'10*, 2010.

[46] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES '08*, pages 67–75, 2008.

[47] X. Shuai, Y. Ding, J. Busemeyer, S. Chen, Y. Sun, and J. Tang. Modeling indirect influence on twitter. *IJSWIS*, 8(4), 2012.

[48] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *KDD'11*, pages 1049–1058, 2011.

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

[50] J. Tang, S. Wu, and J. Sun. Confluence: Conformity influence in large social networks. In *KDD'13*, pages 347–355, 2013.

[51] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM'06*, pages 613–622, 2006.

[52] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *ICDM'07*, pages 322–331, 2007.

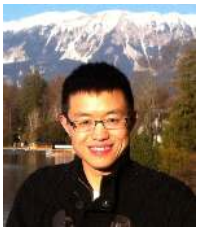
[53] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *KDD'11*, pages 1100–1108, 2011.

[54] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM'10*, pages 261–270, 2010.

Jing Zhang is a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. Before this, Jing got her master degree from the Department of Computer Science and Technology, Tsinghua University. Her research interests include information diffusion, social influence and social representation. She has been visiting student in Hongkong University of Science and Technology and University of Illinois at Urbana-Champaign. She has served as the PC member of ICDM 2014, ASONAM 2015, and the proceeding chair of WSDM 2015.

Zhanpeng Fang received the BE degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, in 2013. He is a master student of the Tsinghua University and Carnegie Mellon University dual degree program in computer science. His research interests include data mining and social network analysis. He has won the first place award in the ICDM 2012 contest and the second place award in the CIKM Cup 2014.

Wei Chen received the bachelors and masters degrees from the Department of Computer Science and Technology, Tsinghua University, and the PhD degree from the Department of Computer Science, Cornell University, Ithaca, New York. He is a lead researcher at Microsoft Research Asia and an adjunct professor of Tsinghua University. His main research interests include distributed computing, fault tolerance, and social network analysis. He is a member of the IEEE.



Jie Tang received the PhD degree from Tsinghua University. He is an associate professor in the Department of Computer Science and Technology, Tsinghua University. His main research interests include data mining algorithms and social network theories. He has been visiting scholar at Cornell University, University of Illinois at Urbana-Champaign, Chinese University of Hong Kong, Hong Kong University of Science and Technology, and Leuven University. He has published over 100 research papers in major international journals and conferences including: KDD, IJCAI, WWW,SIGMOD, ACL, Machine Learning Journal, TKDD, TKDE, and JWS.