2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Modeling Social Network Topology with Variable Social Vector Clocks

Ta Yuan Hsu
Dept. of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, IL, 60607-7053, USA
thsu4@uic.edu

Ajay D. Kshemkalyani
Dept. of Computer Science
University of Illinois at Chicago
Chicago, IL, 60607-7053, USA
ajay@uic.edu

***Abstract*—Analyzing social network structures can provide an insight into the character of human interactions and communication mechanisms for solving a variety of social problems. By applying variable social vector clocks and involving weight evolution influence, we construct a coupled-weight and directed link generation algorithm for modeling a social topology in a closed social group. The degree and weight strength distributions of simulation topologies demonstrate the scale-free properties and effectiveness of weight diffusion in the real world.**

I. Introduction

The origin of social networks was a giant transition in the evolution of information spread. The characteristics of network topologies give much insight into the patterns on which the communication connections are based and help us better understand complex social systems, such as human societies [1]. The communication patterns are deeply controlled by information diffusion dynamics and interacting processes in many social networks [2]. Consequently, it is a worthwhile problem to investigate how communication topology structures form with network growth. It can provide unprecedented perspectives on social interaction dynamics.

Many works have given much attention to the influence of complex network topological structures in a variety of fields [3][4]. By applying the approach of complex networks [5][6], the interaction patterns among social groups have been studied extensively. In most real networks, the connection link between a pair of nodes is characterized by a varying weight, like in air traffic or proteomic networks. A weight can be viewed as the strength or frequency of interactions in the social network. It has been shown that the weighting factor strongly influences the characteristics and dynamics of complex networks in many examples [7][8]. In consequence, the coupled weight-topology mechanism is naturally applied to social networks analysis and modeling [9][10]. With this approach, we can figure out various issues, such as terrorist attacks, public safety, and economic problems. For example, Broder et al. [11] analyzed the link structures within web pages and showed that the distribution of degree of web pages obeys the power law property.

In reality, the weight-topology coupling dynamics in social networks is highly affected by social interaction. It is driven by the cyclic and the focal closure for the evolution processes. The cyclic closure mechanism relates to the link connection with neighbors of neighbors. The focal closure process relates to the link connection independently of the local connectivity or geodesic distance [12]. These two fundamental mechanisms are involved in our topology generation model. The stronger the coupling strength of links, the higher the weights of nodes connecting them are. As with the weights, every interaction leads to a stronger connection. Also, the task handling process of individuals plays an important role in the network evolution. In a co-evolutionary network, the time factor also affects the dynamic evolution process of its topological structure [13]. It has been shown that the emergence of bursty dynamics and Granovetter-type weight-topology structure in evolving networks can be formulated by inter-event time factor and weight-topology coupling [14].

There are some well-known topology models in complex networks, such as random type, hierarchical type, or scale-free networks. Scale-free model has been widely proven to be a better framework than others to capture the real-world network topology. The reason is because the character of scale-free networks reflects the social communication behaviors in terms of the growth and preferential attachment characteristics [15]. However, it has been a challenging issue to formulate information diffusion for event-driven communication in a network. Event-driven information diffusion may be highly correlated with timing and ordering of events. The fine-grained temporal framework has been proposed to effectively capture the dynamic inter-communication in social networks [16]. It utilizes the formula of *Vector Clocks* (VC) to realize the temporal infrastructure. The notion of VC from distributed systems was conceptually introduced by Mattern [17]. It can track the most recent state in each node that happens before a given event. Vector-clock-driven frameworks have been widely used in modeling social networks to study many cases, such as self-organization of communication topology [18], and group formation and social navigation [19].

Kossinets et al. [20] applied VC to social networks and proposed a framework of *Social Vector Clocks* (SVC) to capture how information is spread in social networks. However, the conventional SVC has the drawback of poor scalability. Lee et al. [21] proposed a modification to SVCs to formulate the temporal features applicable to social interaction networks with better scalability for link prediction. Hsu et al. [22][23] further extended the modification of SVC to *Variable Social Vector Clocks* (VSVC) to quantitatively model the influence of information diffusion.

Contributions: In this paper, we propose a universal framework

ASONAM '15, August 25-28, 2015, Paris, France

DOI: http://dx.doi.org/10.1145/2808797.2809405

of dynamic variable weight evolution based on the influence of information diffusion. We also implement a robotic social closure mechanism and probabilistic system introduced by triad interactions for link creation. Combining with them, we apply VSVCs to propose a Social Network Topology Construction (SNTC) algorithm for a closed social group. SNTC is based on a directed coupling weight network. Our goal is to explore the spreading effect of dynamic evolution of weights for human interaction within a closed social group. The degree and weight strength distributions of simulation topologies demonstrate the scale-free properties and effectiveness of weight diffusion in the real world.

Section II introduces the concept of variable social vector clocks. Section III presents the SNTC algorithm that simulates the construction of network connections. Section IV analyses the simulation results generated by our proposed SNTC algorithm, and then compares with some real-world social network topologies. Section V gives the conclusion.

II. Motivation

In [21], the authors stress that the fine-grained temporal view might provide valuable additional information over and above a series of communication events. Vector clocks have been widely applied to a variety of fields. Although several works have studied vector-clock-driven social network models, they used the traditional social vector clocks (e.g., [18][19]). In our model, we propose a modification of VSVC to model the social network topology.

A. Traditional Social Vector Clocks

Given a set of N vertices in a social network, each vertex can be viewed as a process in a distributed system. Suppose that there is no communication delay and there exists one global synchronous time. Note that the above two constraints are not applicable to the framework of asynchronous distributed systems. A sequence of communication events are organized in terms of the global time ordering within a time interval $[0 \sim T]$. Each event is composed of a multivariate function on a 3-tuple ($timestamp$, $sender$, $receiver$).

The traditional SVC updating approach practically follows the mechanism of the conventional vector clocks. With the assumption of no propagation delay in social networks, when receiving a message sent at timestamp t from sender j at an incoming event (E_t), the timestamp of the receiver i's temporal view of the j^{th} and i^{th} entries is set to the timestamp of E_t. Under the piggyback system of the traditional SVCs, however, each vertex will soon get a large number of indirect updating messages from others, most of whom the receiver does not have any direct communication with ever, or has too far social-connection steps in between them.

Therefore, a modification of the updating framework for SVCs has been addressed in [21]. Here, a parameter μ gives the upper bound on the minimum number of hops between a pair of sending vertex and receiving target along time-respecting paths. This parameter is included in the framework of the traditional SVCs. The semantics of three major different values assigned to μ are as follows:

μ = 1: This only involves direct friendship communication. A receiver can update a component of the local social vector clock based on the incoming message if and only if the corresponding sender for that component ever directly interacts with the receiver (an incoming communication event corresponds to the vector clock piggybacked on a message).

μ = 2: This case further involves friendship-of-friendship indirect communication. A receiver i can update the k^{th} component of the local SVC based on the k^{th} component of the piggybacked timestamp directly sent from a vertex j if and only if the corresponding indirect sender k has ever directly interacted with the sender j.

$\mu = \infty$ (practically it acts as μ being $N-1$ in a N-vertex social group): it is equivalent to the conventional SVC updating approach, considering unlimited indirect communication spread without self-looping updating.

B. Variable Social Vector Clocks and the Modification

Hence, it is without loss of generality to consider overall different reachable distances of friendship. A universal framework of the VSVCs has been presented by Hsu et al. [22][23]. Assume that when μ = c, vertex j sent a direct message to vertex i. Vertex i can receive an indirect update on the k^{th} component of the local SVC based on the k^{th} component of the piggybacked timestamp via a direct update from vertex j, if and only if the maximum number of hops from vertex k to vertex j is $c - 1$. Note that the minimum number of hops in a social network infers the shortest friendship distance. For clarity, d_{ab} is defined as the minimum number of hops from vertex a to vertex b along time-respecting paths. Initially, d is set to $\perp$ for each pair of vertices. If d_{ab} is larger than 1, it means that vertex b has received indirectly the piggybacked information sent from vertex a. If vertex a has sent vertex b a targeted message, d_{ab} should be equal to 1.

In this paper, we propose a modification of VSVCs. It includes four data elements about each entry j at vertex i.

1) $VSVC_i[j].time$ captures the latest timestamp of vertex j at vertex i.
2) $VSVC_i[j].Plist$ is a list that holds the predecessors of j and their corresponding timestamps.
3) $VSVC_i[j].Slist$ is a list that holds the successors of j and their corresponding timestamps.
4) $VSVC_i[j].dist$ measures the shortest friendship-respecting distance from vertex j to vertex i.

Whenever a vertex i receives new timestamps, it needs to compute the shortest friendship distance, and the predecessor and successor information with respect to all the other vertices.

III. Methodology

Our motivation seeks to investigate the characteristics and scaling properties of random network topology in a closed social group. We propose a social networking topology construction (SNTC) algorithm to model a weighted, directed random network generated by the probability distribution of the vertex connectivity.

A. Weighted and Directed Social Networks

A vertex represents a human entity. If a directed edge e_{ij} exists, it means that vertex i has sent vertex j a targeted message. The value of w_{ij} is the weight of directed edge e_{ij} and corresponds to the frequency that targeted communication has been issued from vertex i to vertex j. The weight of a directed network can be described by its asymmetric adjacency matrix $(W)_{N\times N}$. Assume that the size of network is N. For clarity, we provide the following definitions.

• the out-degree of vertex i refers to the number of arcs incident from i.

$$OD_i = \sum_j x_{ij} \begin{cases} x_{ij} = 1 & \text{if} \quad w_{ij} > 0 \\ x_{ij} = 0 & otherwise \end{cases} \tag{1}$$

• the in-degree of vertex i refers to the number of arcs incident to i.

$$ID_i = \sum_j x_{ji} \begin{cases} x_{ji} = 1 & \text{if} \quad w_{ji} > 0 \\ x_{ji} = 0 & otherwise \end{cases} \tag{2}$$

• the out-strength of vertex i corresponds to the sum of weights of edges whose outgoing vertex is i.

$$OS_i = \sum_j w_{ij} \tag{3}$$

• the in-strength of vertex i corresponds to the sum of weights of edges whose incoming vertex is i.

$$IS_i = \sum_j w_{ji} \tag{4}$$

• the total strength of vertex i is defined by

$$S_i = OS_i + IS_i \tag{5}$$

B. SNTC Algorithm

SNTC is originally adapted from scale-free random network $ModelB$ proposed by Barabasi et al. [24]. The invariant in $ModelB$ makes the number of vertices constant to eliminate the growth process during the network evolution. This way of holding the size of network mimics some $closed$ social groups, such as *The Telegraph*. It is a list (group) of UK athletes in Twitter. After being created, the size of list hardly varies. However, $ModelB$ explores undirected networks with unweighted edges, which means that a pair of vertices with a connectivity will not be reconnected again. In contrast, most real world social networks are directed and weighted. The SNTC model is defined in the following stages.

1) Initialization: We consider a directed and weighted network with a fixed size of (N) vertices. Initially all vertices are set to be isolated, which means that the initial network is without any connections. Moreover, the weight w_{ij} from sender i to receiver j is zero. In the SNTC model, the dynamics for each timestamp processing consists of the following stages (2)–(4) .

2) Targeted Communication Creation: In our model, there are two kinds of targeted communication. First, a vertex i sends a stranger receiver j (i.e., w_{ij} is zero) a targeted message. It creates a directed link between them. Second, a vertex i transmits a targeted message to its neighbor j (i.e., w_{ij} $>$ 0). Although it does not change the topological link, it increases the strength of w_{ij}. This is viewed as the neighboring interaction (NI). As mentioned above, several studies model the evolution of social networks with cyclic and focal closure mechanisms, such as in [14][25]. The focal closure is regarded as a global attachment (GA) process. It dominates the connections between the random pairing of vertices. The cyclic closure is regarded as a local attachment (LA) process. It implements the connections between vertices having a common neighboring vertex. However, these two mechanisms (GA and LA) are controlled by predefined, constant probabilities (P_{GA} and P_{LA}) for $undirected$ networks in most previous works. Therefore, we utilize GA and LA to formulate a robotic system. It forms the basic rules of $directed$ topological link creation of our model. Contrasted with other studies, P_{GA} and P_{LA} are variable during the network evolution. The motivation comes from the following observation. Initially, the weights of all vertices are identical without any link connections. P_{GA} should be one and P_{LA} should be zero. With the network evolution, P_{GA} should decrease and P_{LA} will increase such that P_{LA} + P_{GA} = 1. In our model, P_{GA} depends on the percentage of the number of isolated vertices in the whole network. There are four cases of selecting a pair of sender and receiver, as follows.

1) Randomly select a pair of vertices from the whole network with probability P_{GA}*P_{GA}.
2) Randomly select a sender i from the whole network. Choose a receiver j from non-isolated vertices with probability $S_j/\sum_s S_s$. This case is with probability P_{GA}*P_{LA}.
3) Choose a sender i from non-isolated vertices with probability $S_i/\sum_s S_s$. Randomly select a receiver j from the whole network. This case is with probability P_{LA}*P_{GA}.
4) The final case is with probability P_{LA}*P_{LA}. The way of selecting a sender i is the same as the third case. There are two subcases to select a receiver j.
   - The first subcase is to choose a receiver j from the same connected group $G(i)$ of i based on $S_j/\sum_{s\in G(i)} S_s$. This subcase is with probability $P_{LA}^2 * P_{lg}$ (where P_{lg} (the local group probability) = the number of vertices in $G(i)$/the number of vertices from the whole non-isolated vertices).
   - In the second subcase the way of choosing a receiver j is the same approach as the second case. This subcase is with probability $P_{LA}^2 * (1 - P_{lg})$.

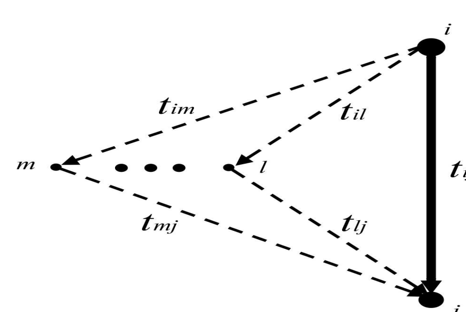

Fig. 1: Schematic representation of triad interaction.

In the fourth case, we need to consider the triad interactions. Whenever i sends a targeted message to j, as shown in Fig 1, we check whether any vertex k exists satisfying that both the edges e_{ik} (from i to k) and e_{kj} (from k to j) have existed.

The approach of selecting a k is based on the observation in [14] that the triad interaction is introduced by a common vertex k when a pair of i and j communicate at t_{ij}. This k needs to satisfy $|t_{ik} - t_{jk}| = 1$ and $t_{ij} - t_{ik} + t_{ij} - t_{jk} = 3$. It implies that after i and j communicate with k consecutively, i and j will communicate with each other at the next timestamp. We generalize the above conditions to propose a probabilistic schema to choose k with the higher chance to introduce the triad interaction between i and j. From all the common vertices, we choose one vertex v preferably through the larger sum $\sum : t_{iv} + t_{vj}$ and the smaller difference $\Delta : |t_{iv} - t_{vj}|$. The triad interaction in [14] is a special case of our schema. Then, the weights of w_{iv} and w_{vj} may be updated by the following formula.

$$w_{iv} \leftarrow w_{iv} + \delta(\frac{1}{OD_i}), w_{vj} \leftarrow w_{vj} + \delta(\frac{1}{ID_j}) \qquad (6)$$

A parameter δ is specified by users to control the change of weights. After selecting sender i and receiver j, the weight w_{ij} is updated by

$$w_{ij} \leftarrow w_{ij} + w_0 \qquad (7)$$

Intuitively, $w_{ij} > 0$ means that i has sent j a targeted message. We assume that w_0 is one in this paper.

3) Updating Variable Social Vector Clocks: Upon vertex j received a targeted message from i, the VSVC of j should be updated. Through VSVC, vertex j can track the latest status and shortest information paths from other vertices. The data structure for VSVCs is the same as presented in Section 2.

Algorithm 1 shows the updating process of VSVCs. Lines 1-4 capture the up-to-date timestamp, renew the successor information on the entry i, update the predecessor information on the entry j, and measure the shortest friendship distance from i to j. Lines 5-17 deal with the updating procedure of VSVC of j on the entries $k \neq i$ & $k \neq j$. In lines 11-12, $Merge$ will merge two successor lists ($Slist$), and then, two predecessor lists ($Plist$), respectively. When two $Slists$ have the same successor, $Merge$ chooses the latest timestamp from the two successors' timestamps. When merging two $Plists$, it follows the same way as merging two $Slists$.

4) Dynamic Variable Weight Evolution: The fundamental concept of *dynamic variable weight evolution* originates from the following observation. An individual preferentially uses his social connections [26] as source for social intercourse. This phenomenon feeds a transitivity effect in the network because the friends of his friends are more likely to have a connection from him. Newly added targeted communication for sender i and receiver j might not only cause weight of edge e_{ij} to increase, but also cause weights of edges e_{pi} and e_{js} to increase ($p \in$ the neighboring predecessors of i and s $\in$ the neighboring successors of j). The corresponding update rules of weights have been utilized and referred to as *dynamic evolution of weight* in [27] as follows.

$$w_{pi} \leftarrow w_{pi} + \delta(\frac{w_{pi}}{IS_i}) \qquad (8)$$

$$w_{js} \leftarrow w_{js} + \delta(\frac{w_{js}}{OS_j}) \qquad (9)$$

Algorithm 1: Variable Social Vector Clocks

Input: $Sender\ i, VC_i[]; Receiver\ j, VC_j[]; Timestamp\ t$

1 $V_j[j].time \leftarrow t$; $V_j[i].time \leftarrow t$;
2 $V_j[i].Slist.add(j, t)$;
3 $V_j[j].Plist.add(i, t)$;
4 $V_j[i].dist \leftarrow 1$;
5 **if** V_i *has been active* **then**
6   **for** $k \leftarrow 0$ **to** $N-1$ *but* $k \neq j, i$ **do**
7   **if** $V_i[k] \neq \bot$ *and* $V_j[k] \neq \bot$ **then**
8   **if** $V_i[k].dist < V_j[k].dist$ **then**
9         $V_j[k].dist \leftarrow V_i[k].dist+1$;
10       $V_j[k].time \leftarrow max\{V_i[k].time, V_j[k].time\}$;
11       $Merge(V_i[k].Slist, V_j[k].Slist)$;
12       $Merge(V_i[k].Plist, V_j[k].Plist)$;
13   **else if** $V_i[k] \neq \bot$ *and* $V_j[k]$ *is* $\bot$ **then**
14       $V_j[k].time \leftarrow V_i[k].time$;
15       $V_j[k].Slist.add(V_i[k].Slist)$;
16       $V_j[k].Plist.add(V_i[k].Plist)$;
17       $V_j[k].dist \leftarrow V_i[k].dist+1$;

Here the definition of δ is the same as in equation (6). Without loss of generality, we further propose a universal weighted updating framework (*dynamic variable weight evolution*) adapted to all predecessors p of i and successors s of j when a targeted mention has been sent from i to j. The influence of the weight update in e_{ij} could dynamically cause a transitivity effect for all the direct or indirect predecessors/successors of i/j. For example, when an individual receives a message, his friends and friends of friends might be more likely to send or receive in the future. Therefore, it will be reasonable to update the weights on the time-respecting paths with the shortest friendship distance from p to i and from j to s. For clarity, the definitions of $predecessors$ for vertex i and $successors$ for vertex j are as follows.

- $Predecessor\ p$ of i: a vertex ($d_{pi} = VC_i[p].dist > 0$).
- $Successor\ s$ of j: a vertex ($d_{js} = VC_s[j].dist > 0$).
- P_i: set of all predecessors of i.
- S_j: set of all successors of j.

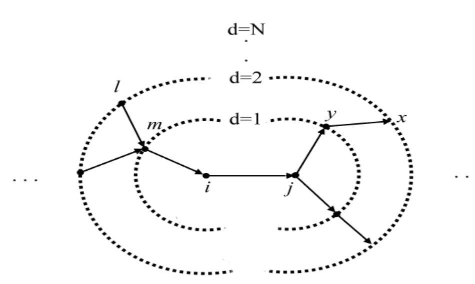

Fig. 2: Illustrative example of dynamic variable weight evolution.

Consider the backward information pathways of sender i. When an edge e_{lm} exists (l, $m \in P_i$) and d_{li} is equal to d_{mi} plus one, the weight w_{lm} of e_{lm} needs to be updated as follows.

$$w_{lm} \leftarrow w_{lm} + \delta(\frac{w_{lm}}{IS_m})^{d_{li}} \quad (10)$$

On the other hand, consider the forward information paths of receiver j. Similarly, when an edge e_{yx} exists ($y, x \in S_j$) and d_{jx} is equal to d_{jy} plus one, the weight w_{yx} of e_{yx} needs to be updated as follows.

$$w_{yx} \leftarrow w_{yx} + \delta(\frac{w_{yx}}{OS_y})^{d_{jx}} \quad (11)$$

Note that the value d_{li} and d_{jx} must be larger than or equal to 1 in (10) and (11). Obviously, *dynamic evolution of weight* in (8) and (9) is a special case of *dynamic variable weight evolution* in (10) and (11). Figure 2 illustrates an example of dynamic variable weight evolution based on the above scenario.

The algorithm 2 represents the procedure of *dynamic variable weight evolution*. The input μ is defined as the upper bound of the shortest friendship distance *d*. Obviously, the weight updating evolution would be manipulated by μ. Based on the direction of information pathways, there are two weight updating functions – BACKWARD and FORWARD, respectively.

In BACKWARD, sender i is the pivot point. Lines 2-4 will capture all predecessors of the sender i and classify them into predecessor lists $predecessor[d].lists$ based on each predecessor's shortest friendship distance d to i (i.e., $VC_i[k].dist$). Line 1 saves the pivot i in the list $ancestor[0]$. Line 6 is used to detect the termination condition. Note that $V_i[l].Slist$ maintains the successors of l from the local view of i. Line 8 checks whether the shortest path distance from each vertex in $V_i[l].Slist$ to i is $d-1$. Line 9 deals with the dynamic weight updates based on equation (10).

Similarly, in FORWARD, receiver j is the pivot vertex. Lines 11-13 get all successors of receiver j. Then, classify them into successor lists $successor[d].lists$ based on each successor's shortest friendship distance d from j (i.e., $VC_k[j].dist$). Line 10 saves the pivot j in the list $successor[0]$. Line 15 does termination detection. Line 17 checks whether y is on the shortest path from j to x with distance $d-1$. Line 18 deals with the dynamic weight updates based on equation (11).

Algorithm 2: Dynamic Variable Weight Evolution

Input: $Sender\ i$; $Receiver\ j$; μ

BACKWARD(i):

1 $predecessor[0].add(i)$;
2 **for** *each index* k **do**
3   **if** $k \neq i \bigwedge (\mu \geq VC_i[k].dist > 0)$ **then**
4     $predecessor[VC_i[k].dist].add(k)$;
5 **for** $d \leftarrow 1$ **to** μ **do**
6   **if** $predecessor[d].list\ is\ \emptyset$ **then**
7     $break$;
8   **for** $\forall\ m \in V_i[l].Slist \bigwedge m \in predecessor[d-1].list \mid l \in predecessor[d].list$ **do**
9     $w_{lm} \leftarrow w_{lm} + \delta(\frac{w_{lm}}{IS_m})^d$

FORWARD(j):

10 $successor[0].add(j)$;
11 **for** *each index* k **do**
12   **if** $k \neq j \bigwedge (\mu \geq VC_k[j].dist > 0)$ **then**
13     $successor[VC_k[j].dist].add(k)$;
14 **for** $d \leftarrow 1$ **to** μ **do**
15   **if** $successor[d].list\ is\ \emptyset$ **then**
16     $break$;
17   **for** $\forall\ y \in V_x[x].Plist \bigwedge y \in successor[d-1].list \mid x \in successor[d].list$ **do**
18     $w_{yx} \leftarrow w_{yx} + \delta(\frac{w_{yx}}{OS_y})^d$

IV. RESULTS

A. Scale-free Network Topology

In the simulation setting, we use N = 3,500 which is the size of the network, $time\ steps$ = 20N, and δ = 1.0 to generate a directed and weighted social network topology. When μ = 0 (i.e., the influence of dynamic evolution of weight does not exist), the in-degree and out-degree distributions are both between Gaussian and a power law. This obeys the observation in $Model\ B$. When μ is greater than zero, they roughly show the power law property. The probabilities of higher-degree vertices are exponentially suppressed. As a result, the network topology structure could be roughly viewed as being homogeneous (i.e., most vertices have similar degrees, distributed approximately with the average degree). As μ increases, the shapes of $Prob(degree)$ change to tend to that of $Prob(degree)$ with μ being ∞. Interestingly, the distributions for μ = 4 and μ = ∞ are very close. It infers that the influence of dynamic evolution of weight is bound in 4 friendship links. Note that the parameter δ is used to control the strength of dynamic evolution of weight. As δ increases, the shape of $Prob(degree)$ obviously changes to the power law distribution in small degrees.

We further explore the probability distribution of the in-strength and out-strength, respectively. They show the power law property for $\mu > 0$. Likewise, the strength distributions for μ = 4 and ∞ are very close. The above results demonstrate that a large number of vertices communicate with a few vertices, while only a small number of vertices communicate with plenty of vertices.

B. Real Network Topology Examples

We turn to investigate what value of μ may be reasonably involved in some real closed social groups. We analyze four Twitter lists. A Twitter list is a organized group of Twitter users. Since most of the users in the four lists are subscribed to the lists during a short interval of time, each of the four lists can be regarded as an individual closed group.

- *London 2012 UK Olympics:* It covers Twitter communication events among 492 UK Olympic athletes over four years.
- *MLB:* It includes Twitter communication events among 563 Major League Baseball players in 2013.
- *Mashable:* There are 480 speakers and attendees to stay in the know on the latest by mashable.com.

• *NY Times Journalist:* It includes reporters, editors, photographers and producers curated by *The New York Times.*

To compare with the above social group topologies, we simulate the topology distributions for different μ with corresponding N, δ, and the number of targeted messages in a social group. Simulation results are figured out by different values $\mu : 0 \sim 3$. Interestingly, the fitting lines obtained from the corresponding real social groups are almost covered by the simulation distribution results. In other words, the influence of dynamic evolution is bound to be three friendship links for the above four Twitter groups. It demonstrates that dynamic evolution indeed exists in real-world social networks. Note that the parameters δ are set to be 2.0 and 6.0 for $Mashable$ and $NYtimes$, respectively. It implies that the strength of dynamic weight evolution in these two groups is stronger. Indeed, highly connected people in $Mashable$ groups have sent and received more messages than those highly connected members in other groups. The simulation result shows that the connection strength plays an important role in a social topology.

V. CONCLUSION

In this paper, we proposed an algorithm (SNTC) to construct a complex network for simulating a closed social network topology. We formulated an approach to dynamic variable weight evolution and applied the framework of VSVCs in a coupled-weight and directed network. In addition, we incorporated a robotic social closure mechanism and a probabilistic system for triad interactions in realizing targeted mention link generation. The simulation results show that the network topological structures generated by SNTC have the characteristics of scale-free properties and are effectively controlled by weight diffusion extent (μ) and strength (δ). If μ = 4 and ∞, their topological characteristics are very close. Likewise, the strength distributions for different μ satisfy the power law properties. Besides, if $\mu < 4$, the strength and degree distributions are obviously influenced and distinct. Finally, we clearly see that the simulating topological distributions by SNTC with corresponding N, δ, and the number of communication events are consistent with real social topologies when $\mu < 4$. It seems to imply that the effectiveness of information diffusion exists within friends, friends-of-friends, and friends-of-friends-of-friends.

REFERENCES

[1] G. Caldarelli, *Scale-Free Networks : Complex Webs in Nature and Technology*. Oxford University Press, 2007.

[2] Y. Li, W. Chen, Y. Wang, and Z. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," *CoRR*, vol. abs/1111.4729, 2011.

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

[4] A.-L. Barabasi, *Bursts : The Hidden Pattern Behind Everything We Do*. New York, N.Y. Dutton, 2010.

[5] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[6] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks : Structure and dynamics," *Phys. Rep.*, vol. 424, no. 4-5, pp. 175–308, 2006.

[7] J. P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, M. A. de Menezes, K. Kaski, A. L. Barabasi, and J. Kertesz, "Analysis of a large-scale weighted network of one-to-one human communication," *New Journal of Physics*, vol. 9, p. 179, JUN 28 2007.

[8] A. Barrat, M. Barthlemy, and A. Vespignani, "Weighted evolving networks: Coupling topology and weight dynamics," *Physical Review Letters*, vol. 92, no. 22, 2004.

[9] J. Staddon, A. Acquisti, and K. LeFevre, "Self-reported social network behavior: Accuracy predictors and implications for the privacy paradox." in *SocialCom*. IEEE, 2013, pp. 295–302.

[10] Z. Huang and Y. Qiu, "Construction and aggregation of citation semantic link network," in *Fourth International Conference on Semantics, Knowledge and Grid, SKG '08, Beijing, China, December 3-5, 2008*, 2008, pp. 247–254.

[11] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Comput. Netw.*, vol. 33, no. 1-6, pp. 309–320, Jun. 2000.

[12] G. Kossinets and D. Watts, "Empirical analysis of an evolving social network," *Science*, vol. 311, no. 5757, pp. 88–90, 2006.

[13] P. Holme and M. E. J. Newman, "Nonequilibrium phase transition in the coevolution of networks and opinions," *Physical Review E*, vol. 74, no. 5, 2006, qC 20100525.

[14] H.-H. Jo, R. K. Pan, and K. Kaski, "Emergence of bursts and communities in evolving weighted networks," *PLoS ONE*, vol. 6, no. 8, p. e22687, 08 2011.

[15] M. Panda, N. El-Bendary, M. Salama, A. E. Hassanien, and A. Abraham, *Social Networks Analysis: Basics, Measures and Visualizing Authorship Networks in DBLP Data*. London: Series in Computer Communications and Networks, Springer Verlag, 2012.

[16] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[17] F. Mattern, "Virtual time and global states of distributed systems," *Proceedings of the Parallel and Distributed Algorithms Conference*, pp. 215–226, 1988.

[18] M. Rosvall and K. Sneppen, "Modeling self-organization of communication and topology in social networks," *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 1, pp. 016 108–, 2006.

[19] ——, "Reinforced communication and social navigation generate groups in model networks," *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, vol. 79, no. 2, pp. 026 111–026 118, 2009.

[20] G. Kossinets, J. Kleinberg, and D. Watts, "The structure of information pathways in a social communication network," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 435–443.

[21] C. Lee, B. Nick, U. Brandes, and P. Cunningham, "Link prediction with social vector clocks," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, Apr. 2013, pp. 784–792.

[22] T. Hsu, A. Kshemkalyani, and M. Shen, "Modeling user interactions in social communication networks with variable social vector clocks," in *COLLABES '14: Proceedings of IEEE International Conference on Advanced Information Networking and Applications*, 2014.

[23] T.-Y. Hsu and A. D. Kshemkalyani, "Variable social vector clocks for exploring user interactions in social communication networks," *Int. J. of Space-Based and Situated Computing*, vol. 5, no. 1, pp. 39 – 52, 2015.

[24] A.-L. Barabasi, R. Albert, and H. Jeong, "Mean-field theory for scale-free random networks," *Physica A*, vol. 272, pp. 173–187, Jul. 1999.

[25] G. Kossinets and D. J. Watts, "Origins of homophily in an evolving social network," *American Journal of Sociology*, vol. 115, no. 2, pp. 405–450, 2009.

[26] G. Miritello, *Temporal Patterns of Communication in Social Networks*. Springer, Berlin, 2013.

[27] F. Da, Y. Liu, and X. Sun, "Modeling algorithm for the topology of weighted directed network based on the triad formation rule," in *2011 International Conference on Network Computing and Information Security*, 2011, pp. 189–193.