Multi-order Ego Network Betweenness

Abstract

In this section, we define terms that represent two types of ego networks, ego networks and multi-layered ego networks (Section ??), as well as the betweenness of a vertex in its ego and x-ego networks (Section 3). We then present several properties of the multi-layered ego networks (Section 3.2). Our betweenness computation algorithm in Section 6 takes advantage of these properties.

Keywords: Ego Networks, Betweenness Centrality

1. Introduction

In this section, we define terms that represent two types of ego networks, ego networks and multi-layered ego networks (Section ??), as well as the betweenness of a vertex in its ego and x-ego networks (Section 3). We then present several properties of the multi-layered ego networks (Section 3.2). Our betweenness computation algorithm in Section 6 takes advantage of these properties.

In this section, we define terms that represent two types of ego networks, ego networks and multi-layered ego networks (Section ??), as well as the betweenness of a vertex in its ego and x-ego networks (Section 3). We then present several properties of the multi-layered ego networks (Section 3.2). Our betweenness computation algorithm in Section 6 takes advantage of these properties.

In this section, we define terms that represent two types of ego networks, ego networks and multi-layered ego networks (Section ??), as well as the betweenness of a vertex in its ego and x-ego networks (Section 3). We then present several properties of the multi-layered ego networks (Section 3.2). Our betweenness computation algorithm in Section 6 takes advantage of these properties.

In this section, we define terms that represent two types of ego networks, ego networks and multi-layered ego networks (Section ??), as well as the

Table 1: Summary of notation

Symbol	Description
$V_i(v)$	set of i-hop neighbors of vertex v ($V_0(v) = \{v\}$)
$V_{\leq i}(v)$	set of vertices that are at most i hops away from vertex v (i.e., $V_{\leq i}(v) = \cup_{k=0}^{i} V_k(v)$)
$E_i(v)$	set of edges connecting two vertices in $V_i(v)$ ($E_0(v)$=$\emptyset$)
$E_{\leq i}(v)$	set of edges connecting two vertices in $V_{\leq i}(v)$

betweenness of a vertex in its ego and x-ego networks (Section 3). We then present several properties of the multi-layered ego networks (Section 3.2). Our betweenness computation algorithm in Section 6 takes advantage of these properties.aaaa

2. Multi-order Friendship Networks

In this section, we define new terms that represent *multi-order ego networks* and *multi-order friendship networks*, and then present several properties of the multi-order friendship networks. Our betweenness computation algorithm in Section 3 takes advantage of these properties.

2.1. Definitions

In this paper, we consider a graph $G(V, E)$[1] where V is a set of vertices and E is a set of undirected edges representing social links between vertices. In the literature Marsden (2002); Everett and Borgatti (2005); Nanda and Kotz (2008); Daly and Haahr (2009), given a graph $G(V, E)$ and a vertex $v \in V$, the *ego network* of v is defined as the subgraph of G consisting of v and its 1-hop neighbors (i.e., vertices with an edge to v) as well as the edges between these vertices. Using the notation summarized in Table 1, this ego network can be formally extended as follows:

Definition 2.1. Given a graph $G(V, E)$ and a vertex $v \in V$, the *multi-order ego network* of v is defined as $\mathcal{E}_v^n(V_{\leq n}(v), E_{\leq n}(v))$ where $V_{\leq n}(v)$ is the set of

[1]We use the term *actors* and *social links* to refer to individuals, groups or organizations and their relationships in a social network. On the other hand, the graph representing a social network consists of *vertices* and *edges* representing actors and social links, respectively.

vertices whose shortest distance from v is no longer than n (i.e., $\{v\} \sum V_n(v)$) and $E_{\leq n}(v)$ denotes the set of edges between the vertices in $V_{\leq n}(v)$.

In a given graph shown at Fig. **??** (a), $V_{\leq 1}(v) = V_0(v) \cup V_1(v) = \{v, a, b, c, d\}$ and $E_{\leq 1}(v) = \{\{v, a\}, \{v, b\}, \{v, c\}, \{v, d\}, \{a, b\}, \{c, d\}\}$. The ego network of vertex v, $\mathcal{E}_v(V_{\leq 1}(v), E_{\leq 1}(v))$, is shown in Fig. **??** (b). Ego networks well model the relationships/interactions between an actor and others in a social network. However, ego networks have the limitation that it does not capture a substantial amount of information. For example, in Fig. **??** (a), assume that actor v received from actor a the information about a's 1-hop neighbors (i.e., b, e, f, g, and v). Despite this information, the ego network of v cannot record the social links between a and e, between a and f, and between a and g since it can represent only the social links between v and each 1-hop neighbor of v, and between two 1-hop neighbors of v. On the other hand, *multi-order ego networks*, defined as $\mathcal{E}_v^n(V_{\leq n}(v), E_{\leq n}(v))$, can hold more information than ego networks. The second-order ego network of vertex v, $\mathcal{E}_v^2(V_{\leq 2}(v), E_{\leq 2}(v))$, is shown in Fig. **??** (c).

To overcome the above limitation, we introduce the following extension to ego networks:

Definition 2.2. Given a graph $G(V, E)$ and a vertex $v \in V$, the *n-order ego network* of v is $\mathcal{E}_v^n(V_{\leq n}(v), E_{\leq n}(v) - E_n(v))$, where $V_{\leq n}(v)$ is the set of vertices that are at most n hops away from v, $E_{\leq n}(v)$ is the set of edges between vertices that are at most n hops away from v, and $E_n(v)$ is the set of edges between n-hop neighbors of v.

Fig. **??** (c) shows the 2-layesubsec:x-egoDefinition v. The 2-layered ego network is different from 2nd order ego networks of which edge sets includes $E_n(v)$. It means that a vertex v can generate its x-ego network by using the neighbor information of its neighbors. In summary, both ego and x-ego networks can be obtained with the same (similar) network overhead. Fig. **??**(b) shows the x-ego network of v from the graph in Fig. 5. As Fig. **??**(a) and Fig. **??**(b) illustrate, the x-ego network of v is different from the ego network of v in that subsec:x-egoDefinitionneighbors of v (i.e., $V_{\leq 2}(v) - V_{\leq 1}(v) = V_2(v)$) as well as the edges between a 1-hop neighbor and a 2-hop neighbor of v. Despite thsubsec:x-egoDefinitionh the ego and x-ego networks of v can be obtained with the same network overhead (since they consume the same messages from 1-hop neighbors of v). The benefits of x-ego networks over ego networks are further verified in Section 4.

Table 2: Comparison of betweenness, ego betweenness, and x-ego betweenness for the graph shown in Fig. 5 (the Pearson correlation is 0.63 between $B(v)$ and $B^{\mathcal{E}}(v)$ and 0.90 between $B(v)$ and $B^{\mathcal{X}}(v)$, and the Spearman correlation is 0.79 between $B(v)$ and $B^{\mathcal{E}}(v)$ and 0.93 between $B(v)$ and $B^{\mathcal{X}}(v)$)

Nodes		v	a	b	c	d	e	f	g	h	i	j	k	l
$B(v)$	Value	0.405	0.383	0.124	0.131	0.286	0.000	0.318	0.049	0.030	0.167	0.000	0.000	0.000
	Rank	1	2	7	6	4	10	3	8	9	5	10	10	10
$B^{\mathcal{E}}(v)$	Value	0.667	0.750	0.167	0.583	0.333	0.000	0.833	1.000	0.167	0.667	0.000	0.000	0.000
	Rank	4	3	8	6	7	10	2	1	8	4	10	10	10
$B^{\mathcal{X}}(v)$	Value	0.383	0.500	0.125	0.269	0.339	0.000	0.524	0.214	0.133	0.400	0.000	0.000	0.000
	Rank	4	2	9	6	5	10	1	7	8	3	10	10	10

3. Betweenness in Multi-order Friendship Networks

In this section, we propose a new algorithm to calculate the betweenness centrality in the multi-oder friendship networks.

3.1. Ego and X-Ego Betweenness

In the literature Freeman (1979); Everett and Borgatti (2005); Brandes (2001), given a graph $G(V, E)$, the betweenness $B(v)$ of a vertex v is defined as:

$$B(v) = \frac{\sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}}{(|V|-1)(|V|-2)} \tag{1}$$

where σ_{st} is the number of shortest paths from vertex s to vertex t and $\sigma_{st}(v)$ is the number of those shortest paths that pass through vertex v. In the above definition, the denominator represents the total number of pairs of vertices except v. It normalizes $B(v)$ to a value between 0 and 1. Given an undirected graph, $\sigma_{st} = \sigma_{ts}$ and $\sigma_{st}(v) = \sigma_{ts}(v)$ for all vertices s, t, and v. Therefore, it is sufficient to find either σ_{st} or σ_{ts} and either $\sigma_{st}(v)$ or $\sigma_{ts}(v)$. In a large wireless network, obtaining the betweenness of each node (i.e., the betweenness of the vertex representing that node) is costly since all nodes that have limited memory and energy must exchange and consume a substantial number of messages to identify the shortest paths between them. On the other hand, each node can obtain its ego and x-ego networks with much lower overhead since each node needs to broadcast information about its 1-hop neighbors (Section **??**). In this paper, we consider the situations where each node computes its betweenness using either its ego network or x-ego network and then uses the result as an estimate of its true betweenness in the entire network. We refer to the betweenness of v computed from the ego network and x-ego network of v as the *ego betweenness* and *x-ego betweenness* of v (denoted $B^{\mathcal{E}}(v)$ and $B^{\mathcal{X}}(v)$), respectively. Table 2 shows, for every vertex v in Fig. 5, the betweenness ($B(v)$), ego betweenness ($B^{\mathcal{E}}(v)$), and x-ego betweenness ($B^{\mathcal{X}}(v)$). In this table, for most of the vertices, x-ego betweenness is closer to betweenness than ego-betweenness mainly because it is derived from a larger number of vertices and edges. For this reason, the correlation coefficient (also known as the Pearson correlation coefficient) of x-ego betweenness and betweenness (0.9) is higher than that of ego-betweenness and betweenness (0.63). The advantage of x-ego betweenness over ego betweenness can also be observed in terms of Spearman's rank correlation, which indicates, given two series $\mathcal{X} = (X_1, X_2, \cdots, X_N)$ and $\mathcal{Y} = (Y_1, Y_2, \cdots, Y_N)$, the correlation between $(r_{\mathcal{X}}(X_1), r_{\mathcal{X}}(X_2)\ , \cdots, r_{\mathcal{X}}(X_N))$ and $(r_{\mathcal{Y}}(Y_1), r_{\mathcal{Y}}(Y_2), \cdots, r_{\mathcal{Y}}(Y_N))$, where $r_{\mathcal{X}}(X_i)$ and $r_{\mathcal{Y}}(Y_i)$ represent the rank of X_i in $\mathcal{X}$ and that of Y_i in $\mathcal{Y}$, respectively. In Table 2, Spearman's correlation is 0.93 between x-ego betweenness and betweenness and 0.79 between ego betweenness and betweenness. In Section 4, we further demonstrate the benefit of x-ego betweenness using wireless trace data.

3.2. Properties of X-Ego Networks

In this section, we present four properties of x-ego networks. These properties enable efficient x-ego betweenness computation (Section 6). As in Brandes' work Brandes (2001), we denote the *dependency* of vertices s and t on vertex v as $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$. Then, the betweenness of v (Equation (1)) can be computed by adding the dependency values for all pairs of vertices excluding v. To quickly calculate such dependency values in x-ego networks, we have identified the properties explained below.

Theorem 3.1. Assume a vertex v, its x-ego network $\mathcal{X}_v$, and its two different 1-hop neighbors s and t (i.e., $s, t \in V_1(v)$ and $s \neq t$). Then, $\delta_{st}(v) = 0$ if there is an edge between s and t (i.e., $\{s, t\} \in E_1(v)$).

Proof. Since $\{s, t\} \in E_1(v)$, $d(s, t) = 1$. Furthermore, $s, t \in V_1(v)$, meaning that $d(s, v) + d(v, t) = 1 + 1 = 2 > d(s, t) = 1$ (i.e., no shortest path from s to t passes through v). Therefore, $\delta_{st}(v) = \sigma_{st}(v)/\sigma_{st} = 0/\sigma_{st} = 0$. □ □

Example 3.2. In Fig. **??**(b), $\delta_{ab}(v) = 0$ since $a, b \in V_1(v)$ and there is an edge between a and b.

Theorem 3.1 allows us to quickly compute dependencies particularly when x-ego networks exhibit a strong community structure (i.e., 1-hop neighbors of a node tend to have direct social links between them). In Section 4, we show this benefit using wireless trace data. When Theorem 3.1 cannot be applied to v's 1-hop neighbors s and t (i.e., there is no edge between s and t), the dependency of s and t on v can be computed using the following theorem.

Theorem 3.3. Assume a vertex v, its x-ego network $\mathcal{X}_v$, and its two different 1-hop neighbors s and t (i.e., $s, t \in V_1(v)$ and $s \neq t$). Then, $\delta_{st}(v) = 1/|V_1(s) \cap V_1(t)|$ if there is no edge between s and t (i.e., $\{s, t\} \notin E_1(v)$).

Proof. Since $\{s, t\} \notin E_1(v)$, $d(s, t) > 1$. In this case, $d(s, t) = 2$ due to the path $s - v - t$. Furthermore, (i) the number of shortest paths from s to t (i.e., paths whose length is 2) can be expressed as $\sigma_{st} = |V_1(s) \cap V_1(t)|$. On the other hand, (ii) the path $s - v - t$ is the only shortest path from s to t that passes through v (i.e., $\sigma_{st}(v) = 1$) since the length of that path is 2, which must be smaller than the length of any other path from s to t that passes through v. By (i) and (ii), $\delta_{st}(v) = \sigma_{st}(v)/\sigma_{st} = 1/|V_1(s) \cap V_1(t)|$. □ □

Example 3.4. In Fig. **??**(b), vertices a and c are 1-hop neighbors of v and there is no edge between a and c. Furthermore, $V_1(a) = \{b, e, f, g, v\}$ and $V_1(c) = \{d, g, h, v\}$. Therefore, $\delta_{ac}(v) = 1/|V_1(a) \cap V_1(c)| = 1/|\{g, v\}| = 1/2$.

Just like Theorem 3.1, the following theorem quickly computes the dependency of two vertices on a vertex v. While the former applies to a pair of 1-hop neighbors of v, the latter applies to a pair of a 1-hop or 2-hop neighbor and a 2 hop-neighbor of v.

Theorem 3.5. Assume a vertex v, its x-ego network $\mathcal{X}_v$, a vertex $s \in V_{\leq 2}(v)$, and another vertex $t \in V_2(v)$ such that $s \neq t$. Then, $\delta_{st}(v) = 0$ if $\delta_{sn}(v) = 0$ for a 1-hop neighbor vertex n of vertex t (i.e., for $n \in V_1(t)$).

Proof. Since $\delta_{sn}(v) = 0$ (i.e., no shortest path from s to n passes through v), (i) $d(s, n) < d(s, v) + d(v, n)$. Then, (ii) $d(s, t) \leq d(s, n) + d(n, t) < d(s, v) + d(v, n) + d(n, t)$ by (i). Furthermore, $n \in V_1(t)$ and $t \in V_2(v)$, meaning that vertex n is either a 1-hop or a 3-hop neighbor of v. In $\mathcal{X}_v$, however, any vertex including n is at most 2 hops away from v. For this reason, n is a 1-hop neighbor of v. Since $d(v, n)$=$d(n, t)$=1 and $d(v, t) = 2$, (iii) $d(v, n) + d(n, t) = d(v, t)$. By (ii) and (iii), $d(s, t) < d(s, v) + d(v, t)$ (i.e., no shortest path from s to t passes through v). Therefore, $\delta_{st}(v) = \sigma_{st}(v)/\sigma_{st} = 0/\sigma_{st} = 0$. □ □

Example 3.6. In Fig. **??**(b), $b \in V_{\leq 2}(v)$ and $g \in V_2(v)$. Furthermore, for a 1-hop neighbor a of g, $\delta_{ba}(v) = 0$ (Theorem 3.1). Therefore, by Theorem 3.5, $\delta_{bg}(v) = 0$.

Given vertices $s \in V_{\leq 2}(v)$ and $t \in V_2(v)$ such that $s \neq t$, Theorem 3.5 cannot be applied if $\delta_{sn}(v) > 0$ for every 1-hop neighbor n of t. In this case, the dependency of s and t on v can be obtained using the following theorem.

Theorem 3.7. *Assume a vertex v, its x-ego network $\mathcal{X}_v$, a vertex $s \in V_{\leq 2}(v)$, and another vertex $t \in V_2(v)$ such that $s \neq t$. Then, $\delta_{st}(v) = \bar{H}(\{\delta_{sn}(v) : n \in V_1(t)\})$ if $\delta_{sn}(v) > 0$ for every 1-hop neighbor n of vertex t, where $\bar{H}(\{\delta_{sn}(v) : n \in V_1(t)\})$ denotes the harmonic mean computed over $\{\delta_{sn}(v) : n \in V_1(t)\}$.*

Proof. We prove this theorem considering the following two cases. [Case I. s is a 1-hop neighbor of v (i.e., $s \in V_1(v)$)] Let n_i $(i = 1, 2, \cdots, m)$ denote the ith 1-hop neighbor of t. Then, each n_i is a 1-hop neighbor of v since, in $\mathcal{X}_v$, any 2-hop neighbor (including t) of v can be connected only to a 1-hop neighbor of v (Definition **??**). Thus, by Theorem 3.3, $\delta_{sn_i}(v) = 1/|V_1(s) \cap V_1(n_i)|$. Since $|V_1(s) \cap V_1(n_i)|$ represents the number of shortest paths from s to n_i (i.e., σ_{sn_i}) and each n_i has an edge to t, the total number of shortest paths from s to t can be expressed as (i) $\sigma_{st} = \sum_{i=1}^{m} |V_1(s) \cap V_1(n_i)| = \sum_{i=1}^{m} 1/\delta_{sn_i}(v)$. On the other hand, the path $s - v - n_i - t$ is the only shortest path from s to t via both v and n_i since the length of the path is 3 and any other path from s to t via both v and n_i must be longer. For this reason, (ii) there are m shortest paths from s to t via v (i.e., $\sigma_{st}(v) = m$). By (i) and (ii), $\delta_{st}(v) = \frac{m}{\sum_{i=1}^{m} 1/\delta_{sn_i}(v)} = \bar{H}(\delta_{sn}(v))$. [Case II. s is a 2-hop neighbor of v (i.e., $s \in V_2(v)$)] Let n_i $(i = 1, 2, \cdots, m)$ denote the ith 1-hop neighbor of t. Let also p_j $(j = 1, 2, \cdots, k)$ denote the jth 1-hop neighbor of s. In $\mathcal{X}_v$, 2-hop neighbors (including s and t) of v can be connected only to a 1-hop neighbor of v. For this reason, any shortest path from s to t must contain a shortest path from p_j to n_i for some j and i (i.e., $\sigma_{st} = \sum_{i=1}^{m} \sum_{j=1}^{k} |V_1(n_i) \cap V_1(p_j)|$). For a pair of i and j, on the other hand, the path $n_i - v - p_j$ is the only shortest path from n_i to p_j via v and so is the path $t - n_i - v - p_j - s$ (i.e.,

$\sigma_{st}(v) = mk$). Therefore,

$$\begin{aligned}
\delta_{st}(v) &= \frac{mk}{\sum_{i=1}^{m}\sum_{j=1}^{k}|V_1(n_i)\cap V_1(p_j)|}\\
&= \frac{mk}{\sum_{i=1}^{m}\sum_{j=1}^{k}1/\delta_{n_ip_j}(v)}\\
&= \frac{m}{\sum_{i=1}^{m}\frac{\sum_{j=1}^{k}1/\delta_{n_ip_j}(v)}{k}}\\
&= \frac{m}{\sum_{i=1}^{m}1/\delta_{n_is}(v)}\\
&= \frac{m}{\sum_{i=1}^{m}1/\delta_{sn_i}(v)}\\
&= \bar{H}(\delta_{sn}(v)).
\end{aligned}$$

□ □

Example 3.8. In Fig. **??**(b), $V_1(h) = \{c, d\}$. By Theorem 3.3, $\delta_{ac}(v) = \frac{1}{2}$ and $\delta_{ad}(v) = 1$. Therefore, by Theorem 3.7, $\delta_{ah}(v) = \bar{H}(\{\delta_{an}(v) : n \in V_1(h)\}) = \bar{H}(\{\delta_{ac}(v), \delta_{ad}(v)\}) = \frac{2}{\frac{1}{\frac{1}{2}}+\frac{1}{1}} = \frac{2}{3}$.

4. Evaluation

5. EEGO

6. Computation

P. V. Marsden, Egocentric and Sociocentric Measures of Network Centrality, in: Social Networks, volume 24, 2002, pp. 407–422.

M. Everett, S. P. Borgatti, Ego Network Betweenness, Social Networks 27 (2005) 31–38.

S. Nanda, D. Kotz, Localized Bridging Centrality for Distributed Network Analysis, in: Proc. of IEEE ICCN, 2008.

E. M. Daly, M. Haahr, Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs, IEEE Trans. Mobile Comput. 8 (2009) 606–621.

L. C. Freeman, Centrality in Social Networks: Conceptual Clarification, Social Networks 1 (1979) 215–239.

U. Brandes, A faster algorithm for betweenness centrality, Journal of Mathematical Sociology 25 (2001) 163–177.