2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Scalable Social Tie Strength Measuring

Yan Zhong*, Xiao Huang†, Jundong Li‡, Xia Hu*
* Texas A&M University, † The Hong Kong Polytechnic University, ‡ University of Virginia
Email: yanzhong@stat.tamu.edu, xiaohuang@comp.polyu.edu.hk, jl6qk@virginia.edu, xiahu@tamu.edu

***Abstract*—Interpersonal ties describe the intensity of information and activity interactions among individuals. It plays a critical role in social network analysis and sociological studies. Existing efforts focus on leveraging individuals' non-structural characteristics to measure tie strength. With the booming of online social networks (OSNs), it has become difficult to process and measure all the non-structural data. We study the tie strength measuring from the network topological aspect. However, it remains a nontrivial task due to the controversial comprehensions of its definition and the large volume of OSNs. To tackle the challenges, we develop a scalable measuring framework - IETSM. From the network view, we formally define the tie strength of an edge as the inverse of its impact on the similarity between its two nodes' influences in information diffusion. To measure this impact, IETSM constructs a node's influence as the embedding learned from its neighborhoods inductively. It estimates the tie strength of an edge through its impact on its nodes' influences brought by deleting it. The learned tie strength scores could, in turn, facilitate the node representation learning, and we update them iteratively. Experiments on real-world datasets demonstrate the effectiveness and efficiency of IETSM.**

***Index Terms*—Tie Strength, Online Social Networks, Inductive Embedding**

I. Introduction

Interpersonal ties are one of the key concepts in sociology [1]. They describe the intensity of interactions and communications among individuals, and reflect the impact of an edge on the diffusion of information within the network [2]. The edges with relatively strong strengths (called strong ties) are more likely to link similar users, who tend to be clustered together and form communities in the networks [3]. The edges with relatively weak strengths (called weak ties) often work as bridges between different communities and are more important in exchanging information between different communities [1]. Thus, accurately measuring the tie strength, especially weak ties, is an essential problem in sociological studies and many real-world applications such as social recommendation [4] and targeted marketing [5]. For example, tie strength has a significant impact on advertising effectiveness [6]. Advertising messages are more likely to be forwarded through strong ties, while weak ties bridge communities and bring more opportunities. Highly interactive features such as entertaining games could be added into messages on weak ties to make the advertising information more attractive.

Inferring tie strength from the network topological aspect gains in importance with the booming of online social networks (OSNs). Traditional ways to measure the tie strength are to use the non-structural characteristics of the network, such as the intensity of interaction and the intimacy between two people [7]. In OSNs, these non-structural data is often incomplete and difficult to be measured [8]. On the other hand, existing structural information based tie strength measuring methods mainly rely on manually-selected structural features such as node degrees [9] and the overlap between nodes' neighborhoods [10]. They only consider partial topological information and tend to achieve suboptimal results.

The effectiveness of network embedding [11], [12] motivates us to explore whether it could be potentially used to advance the tie strength measuring. Network embedding aims to compress a large-scale network into low-dimensional node representations and preserve the network topological information. It serves as an automatic and efficient feature extraction tool and has achieved significant success in various network analysis tasks [13], [14]. Network embedding could benefit tie strength measuring as it can efficiently provide comprehensive features that reflect the entire network structures.

However, it remains a nontrivial task to take advantage of network embedding to measure the tie strength in OSNs, with three major challenges as follows. First, the definition of tie strength in OSNs is controversial. Various attempts have been made, such as node similarity based definitions [15], [16], hidden effect based definitions [17], and strong triadic closure based definitions [8]. They mainly focus on the property of strong ties rather than weak ties. But it has been shown that weak ties are more important than strong ties for individuals to receive new information [18], [19]. Second, real-world OSNs usually involve a vast number of individuals compared with traditional social networks. It puts demands on the scalability of the measuring methods. Third, network embedding could not be directly applied to tie strength measuring. While tie strength measuring targets at modeling edges, network embedding focuses on preserving node similarities and aims to learn node representations that are general to different applications. Thus, a task-specific network embedding algorithm is desired.

In this paper, we aim at answering two research questions: (i) How to define and measure the tie strength in an adjacency matrix from the network view? (ii) How to jointly perform tie strength measuring and network embedding to make them complement each other towards a better measuring performance? Through studying these questions, we propose an efficient measuring framework named Inductive Embedding based Tie Strength Measuring (IETSM). The main contributions of our work are listed as follows.

- Formally define the tie strength in OSNs in a general way, i.e., purely from the network aspect.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

- Propose an efficient framework IETSM that estimates the tie strength of an edge based on the changes of its two nodes' influences brought by deleting it.
- Design an effective learning algorithm that iteratively learns network embedding and tie strength scores since they could complement each other.
- Empirically demonstrate the effectiveness and efficiency of IETSM in tie strength measuring, especially for weak ties.

II. Problem Statement

Let $G = (V; E; \mathbf{W})$ be a social network, where V is the set of nodes, E is the set of edges, and $\mathbf{W}$ is the adjacency matrix. Without loss of generality, we focus on the binary adjacency matrix $\mathbf{W} \in \{0, 1\}^{n \times n}$. Each edge is denoted by an ordered pair $(i; j)$, and w_{ij} indicates the existence of $(i; j)$ in E. If $(i; j) \in E$, $w_{ij} = 1$, otherwise $w_{ij} = 0$. For undirected G, we have $w_{ij} = w_{ji}$. For directed G, $w_{ij} \neq w_{ji}$. The important symbols are listed in Table I.

To give a new definition of the tie strength from the network view, we first study information diffusion. Information diffusion is a process by which information spreads across a network through edges (e.g., news spreads in a community) [20]. When a node i receives a new message, i has the chance to spread it to its connected nodes, and the informed nodes might continue to spread it over the network. The nodes affected by i (including i) during this process are called the neighborhoods of i. The overall effect that i makes on its neighborhoods is called i's influence to its neighborhoods.

For an edge $(i; j)$, a conventional way to investigate its tie strength from the network view is to compare the speed of information diffusion over the networks with and without $(i; j)$ [21]. Inspired by this idea, we study the change of i's and j's influences when we remove an edge $(i; j)$ from the network for both cases of strong tie and weak tie. If $(i; j)$ is a weak tie, $(i; j)$ is an important way or even the only way for j to get information from i. The deletion of $(i; j)$ will make it hard for j to get information from i, and the influences of i and j will become much more different than before. On the contrary, if $(i; j)$ is a strong tie, there could exist several paths from i to j. After $(i; j)$ is removed, j still have a high probability to obtain the information from i via other paths, and the similarity between i's and j's influences change little.

To conclude, the deletion of a weak tie will make an apparent change in the similarity between its two nodes' influences, whereas the removal of a strong tie will bring a relatively small change on this similarity. This kind of change caused by deleting an edge is called the impact of an edge. We formally define the tie strength from the network view according to the impact of an edge in Definition 1.

Definition 1 (Tie Strength) In a social network, the tie strength of an edge $(i; j)$ is defined as the inverse of its impact on the similarity between i's and j's influences to their neighborhoods in information diffusion. The weaker strength an edge has, the more impact it has on this similarity.

Based on the terminologies mentioned above, we define the problem of tie strength measuring in OSNs as follows:

TABLE I: Main Symbols and Definitions

Notations	Definitions
n	the number of nodes in the network
d	dimension of the embedding representation
$\mathbf{W} \in \{0, 1\}^{n \times n}$	binary adjacency matrix
$\mathbf{A} \in [0, 1]^{n \times n}$	tie strength score matrix needed to learn
$\mathbf{u}_i \in \mathbb{R}^d$	embedding representation of node i
(i, j)	the edge from node i to node j
E	the set of edges
V	the set of nodes
G	original network, with $G = (V, E, \mathbf{W})$
$G^{i \not\to j}$	reduced network with edge (i, j) removed
N_i	neighborhoods of node i in G
$N_i^{i \not\to j}$	neighborhoods of node i in $G^{i \not\to j}$
$\mathbf{c}_i \in \mathbb{R}^d$	node i's influence in G
$\mathbf{c}_i^{i \not\to j} \in \mathbb{R}^d$	node i's influence in $G^{i \not\to j}$
S_i	random walk beginning from i in G
$S_i^{i \not\to j}$	random walk beginning from i in $G^{i \not\to j}$

Given a social network G, we aim to calculate the tie strength score $a_{ij} \in [0; 1]$ for each $(i; j) \in E$, as defined in Definition 1, such that a_{ij} would be inversely correlated to the impact that $(i; j)$ has on the similarity between node i's and j's influences to their neighborhoods.

Related Work: Link prediction is related to but different from tie strength measuring. The former aims to predict the present probabilities of non-existing links, while the latter focuses on estimating the intensity of existing edges. Whereas, studies [22] show that the predicted score of the existing edge by similarity-based link prediction method such as Katz [16] and SimRank [15] could also be used as an estimation of its tie strength. Thus we include Katz as a baseline method in our experiments.

Edge centrality refers to a group of methods that indicate the importance of edges in the graph and are often used to find the bridge edges [23]. For example, edge betweenness centrality (EBC) calculates the number of shortest paths between linking nodes that pass through the edge [24]. Usually, the weak tie should have a relatively large edge centrality. However, EBC has the computation complexity of $O(|V||E|)$ [24] and cannot be directly applied to large-scale networks.

Recently, graph neural networks have demonstrated remarkable performance in many tasks [25]. The core idea is to aggregate the structural properties and community level information to reveal a node's effect in the graph. For example, GraphSAGE [26] generates node representations by sampling and aggregating features from its local neighborhood. Planetoid [27] is a graph-based semi-supervised learning framework, in which the embedding can be viewed as hidden layers of a neural network.

III. Impact Analysis with Inductive Embedding

In this section, we propose a scalable framework - Inductive Embedding based Tie Strength Measuring (IETSM). Figure 1 illustrates its main idea. IETSM consists of four components. (i) Infer a node's influence as the embedding learned via inductively aggregating its neighbors' representations and itself.

(ii) Measure the tie strength as the change of the similarity of nodes' influences between the original network and the reduced network. (iii) Introduce random walks to accelerate the computation of the inductive embedding. (iv) Adjust node embedding according to the estimated tie strength. These four components clarify the way how tie strength scores and node embedding representations complement each other. Specifically, we build an effective learning algorithm which trains the tie strength scores and node embedding representations iteratively to achieve accurate measuring results.

Our work is described on directed networks. For undirected networks, we transform them to directed networks by building two directed edges for each undirected edge. The average score of two directed edges is used as the estimated tie strength value for the corresponding undirected edge.

A. Represeting Influences of Nodes

A straightforward way to model a node's influence is to use the node degree, but the node degree cannot sufficiently maintain the topological information in the network. For example, neighborhood overlap (NO) [10], a widely used node-degree based tie strength measure, has a limitation that it could assign zero values for edges located at sparsely connected nodes [28].

We propose to take advantage of network embedding since it could embed the entire topological structure into node vectors. First, we embed the network into the initial d-dimensional embedding $\mathbf{u}_i$ for each node i by preserving the first order proximity. Second, we gather the embedding representations of i's neighborhoods and define their weighted sum as the inductive representation of i's influence.

Initial node embedding by the first order proximity: The first order proximity is the observed pairwise proximity between two nodes in the network [12]. In this problem, the binary adjacency matrix $\mathbf{W}$ is the first order proximity. With a start node i, we sample an arbitrary node j from V and construct an edge sample $(i;j)$. We further assume that the theoretical probability to sample $(i;j)$ is proportional to $\exp(\mathbf{u}_i \quad \mathbf{u}_j)$. Then $p_{ij} = f(i;j)$ is sampled with the start point $ig = \exp(\mathbf{u}_i \quad \mathbf{u}_j)= \sum_{k\in\mathcal{V}} \exp(\mathbf{u}_i \quad \mathbf{u}_k)$. As $\mathbf{W}$ represents the observations of the relationship in the network, the empirical probability of this sample is $\hat{p}_{ij} = w_{ij}= \sum_{k\in\mathcal{V}} w_{ik}$. To preserve the first-order proximity, we expect to find the best representations to minimize the total difference between each pair of distributions $\mathbf{p}_{i\cdot} = (p_{i1};\cdots;p_{in})$ and $\hat{\mathbf{p}}_{i\cdot} = (\hat{p}_{i1};\cdots;\hat{p}_{in})$ of i. With KL divergence to measure the difference, we obtain the following objective function after omitting redundant terms,

$$O_1 = \sum_{i;j\in\mathcal{V}} w_{ij} \log \frac{\exp(\mathbf{u}_i \quad \mathbf{u}_j)}{\sum_{k\in\mathcal{V}} \exp(\mathbf{u}_i \quad \mathbf{u}_k)}: \qquad (1)$$

By maximizing O_1, we get the initial low-dimensional node embedding for each node in the network. Notice that we can choose other node embedding methods that have a differentiable objective function of $\mathbf{u}_i$ and include the first order proximity to construct the initial embedding.

Representing the influences of nodes as the embedding learned from their neighborhoods inductively: For the original network G, we use N_i to denote i's neighborhoods in it, which represents the set of nodes affected by i in information diffusion. We assume that i's effects in information diffusion disappear after K-step spreading, and then N_i becomes the set of nodes that i can contact within K steps through the edges in G. By removing $(i;j)$ from G, we get the reduced network $G^{i\not\to j}$. Similarly, $N_i^{i\not\to j}$ denotes i's neighborhoods in $G^{i\not\to j}$, which is the set of nodes that i can contact within K steps through the edges in $G^{i\not\to j}$.

Now we construct $\mathbf{c}_i$ and $\mathbf{c}_i^{i\not\to j}$ - the node i's influences in G and $G^{i\not\to j}$ as the weighted mean of the node embedding representations in N_i and $N_i^{i\not\to j}$,

$$\mathbf{c}_i = \frac{\sum_{l\in\mathcal{N}_i} {}_{il}\mathbf{u}_l}{\sum_{l\in\mathcal{N}_i} {}_{il}}; \quad \mathbf{c}_i^{i\not\to j} = \frac{\sum_{l\in\mathcal{N}_i^{i\not\to j}} {}'_{il}\mathbf{u}_l}{\sum_{l\in\mathcal{N}_i^{i\not\to j}} {}'_{il}}; \qquad (2)$$

where ${}_{il}$ and ${}'_{il}$ are the weights of node l in N_i and $N_i^{i\not\to j}$. We choose the weight ${}_{il}$ as the $(i;l)^{\text{th}}$ entry of:

$$\mathbf{B} = \mathbf{I} + \mathbf{WD} + (\mathbf{WD})^2 + \quad + (\mathbf{WD})^K; \qquad (3)$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{D}$ is a diagonal matrix whose $(i;i)^{\text{th}}$ element is the inverse of the outer degree of node i. ${}_{il}$ selected in this way reflects the expected times that l is included in a K-step path which starts from i and randomly walks K steps via G. Specifically, $\sum_{l\in\mathcal{N}_i} {}_{il} = K + 1$. ${}'_{il}$ is chosen by the same way with $\mathbf{B}$ calculated in $G^{i\not\to j}$. Similarly, we can calculate $\mathbf{c}_j$ and $\mathbf{c}_j^{i\not\to j}$, the node j's influences in G and $G^{i\not\to j}$.

B. Measuring Tie Strength on the Edge Impact

Based on Definition 1, we investigate the impact of an edge on the similarity between two nodes' influences as follows.

First, we use the inner product of two nodes' influences to reflect their similarity. For example, $\mathbf{c}_i^{i\not\to j} \quad \mathbf{c}_j^{i\not\to j}$ represents the similarity between i's and j's influences in $G^{i\not\to j}$.

Second, we test the change of influence between G and $G^{i\not\to j}$. In the directed network, the deletion of $(i;j)$ mainly changes the information received by j rather than i. Thus we focus on the change of j's influence. To capture the change of j's influence from G to $G^{i\not\to j}$, a straightforward way is to use $\mathbf{c}_j \quad \mathbf{c}_j^{i\not\to j}$. Unfortunately, as both $\mathbf{c}_j$ and $\mathbf{c}_j^{i\not\to j}$ are calculated by the node representations in G, and j's neighborhoods are the same in G and $G^{i\not\to j}$, $\mathbf{c}_j$ and $\mathbf{c}_j^{i\not\to j}$ are almost the same that $\mathbf{c}_j \quad \mathbf{c}_j^{i\not\to j}$ by Eq. (2).

To solve this problem, we use $\mathbf{c}_i \quad \mathbf{c}_j^{i\not\to j}$ instead to reflect the change of j's influence by removing $(i;j)$ for two reasons. First, $\mathbf{c}_i \quad \mathbf{c}_j^{i\not\to j}$ can reflect another kind of loss of information of j from i by removing $(i;j)$. $\mathbf{c}_i \quad \mathbf{c}_j^{i\not\to j} \quad \mathbf{c}_i \quad \mathbf{c}_j$, which represents the potential new information that j's neighborhood can get from node i through $(i;j)$. If we delete $(i;j)$ from the network, node j will lose chance to receive $\mathbf{c}_i \quad \mathbf{c}_j^{i\not\to j}$. Second, the relationship between $\mathbf{c}_i$ and $\mathbf{c}_j^{i\not\to j}$ is similar to the relationship between j's influences before and after the

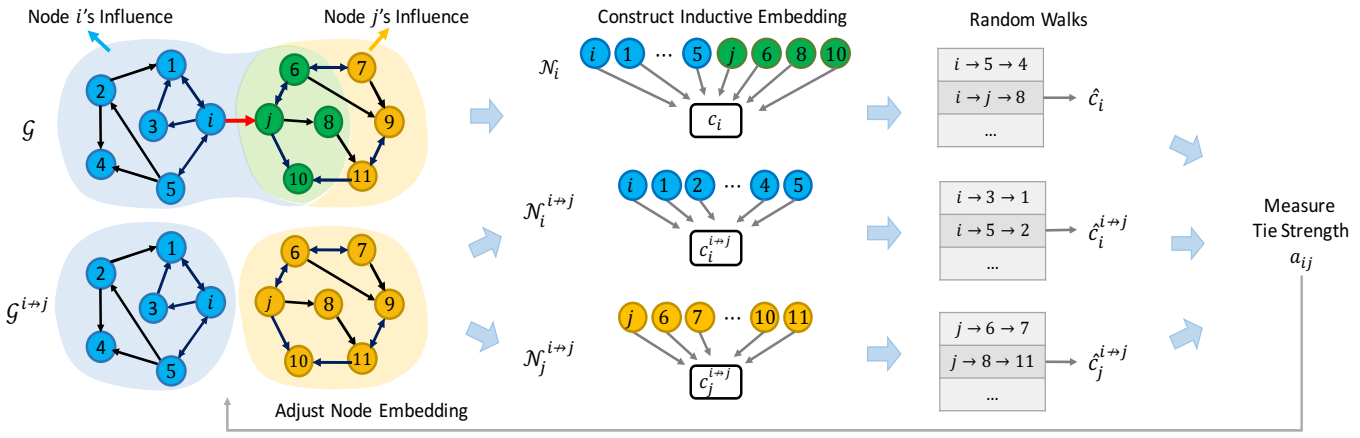

Fig. 1: For $(i;j)$, IETSM compares the difference between the original network G and the reduced network $G^{i\nrightarrow j}$. IETSM constructs $\mathbf{c}_i$, $\mathbf{c}_i^{i\nrightarrow j}$, and $\mathbf{c}_j^{i\nrightarrow j}$ from i's and j's neighborhoods to measures a_{ij}. The estimated a_{ij} can in turn adjust node embedding.

deletion of $(i;j)$. A strong tie's nodes have more paths to share their information than a weak tie's nodes, and thus $\mathbf{c}_i$ and $\mathbf{c}_j$ in the case of the weak tie are more different than the case of the strong tie. Similarly, the influences of j before and after the deletion of $(i;j)$ are more different in the case of the weak tie than the strong tie.

Third, we use $\mathbf{c}_i^{i\nrightarrow j}$ $(\mathbf{c}_i$ $\mathbf{c}_j^{i\nrightarrow j})$ to reflect the loss of the similarity between i's and j's influences to their neighborhoods after removing $(i;j)$. The larger it is, the more impact $(i;j)$ makes, and the weaker the tie strength is. $\mathbf{c}_i^{i\nrightarrow j}$ $(\mathbf{c}_j^{i\nrightarrow j}$ $\mathbf{c}_i)$ can reflect the inverse of the impact of $(i;j)$ on the similarity between i's and j's influences to their neighborhoods. By applying the sigmoid transformation, we could get an estimator of the tie strength of $(i;j)$:

$$\hat{a}_{ij} = \mathbf{c}_i^{i\nrightarrow j} (\mathbf{c}_j^{i\nrightarrow j} \mathbf{c}_i) ; \quad (4)$$

where $(x) = \text{sigmoid}(x) = 1/(1 + exp(x))$.

C. Acceleration by Random Walks

To accelerate the calculation of $\mathbf{c}_i$, $\mathbf{c}_i^{i\nrightarrow j}$, and $\mathbf{c}_j^{i\nrightarrow j}$, we utilize the random walk technique, in which only a small number of nodes in the neighborhoods of i and j are selected each time to construct their influences.

A K-step random walk S_i of i in G is a $(K+1)$-length path that starts from i and randomly moves K steps according to the edges in G. With S_i, we can compress $\mathbf{c}_i$ by:

$$\hat{\mathbf{c}}_i = \frac{1}{K+1} \sum_{l \in S_i} \mathbf{u}_l; \quad (5)$$

where $E(\hat{\mathbf{c}}_i) = \mathbf{c}_i$ with $_{il}$ in Eq. (2) selected by Eq. (3) [29]. Similarly, by drawing two random walks $S_i^{i\nrightarrow j}$ and $S_j^{i\nrightarrow j}$ of i and j in $G^{i\nrightarrow j}$, we can build $\hat{\mathbf{c}}_i^{i\nrightarrow j}$ and $\hat{\mathbf{c}}_j^{i\nrightarrow j}$. We estimate a_{ij} by

$$\hat{a}_{ij} = \hat{\mathbf{c}}_i^{i\nrightarrow j} (\hat{\mathbf{c}}_j^{i\nrightarrow j} \hat{\mathbf{c}}_i) : \quad (6)$$

With T different groups of random walks S_i, $S_i^{i\nrightarrow j}$, and $S_j^{i\nrightarrow j}$, we can calculate T different estimators $\hat{a}_{ij}^1, : : : ; \hat{a}_{ij}^T$. To combine different $\hat{a}_{ij}^t$, a_{ij} is obtained by minimizing:

$$O_{a_{ij}} = \sum_{t=1}^{T} (a_{ij} \hat{a}_{ij}^t)^2 : \quad (7)$$

The result of a_{ij} is simply the average value of $\hat{a}_{ij}^t$. Eq. (7) is used in our algorithm to calculate the gradient of a_{ij}.

D. Adjusting Node Embedding by Tie Strength

Tie strength scores estimated by the inductive embedding can in turn help adjust the node representations and achieve more accurate estimations. For instance, a small tie strength score a_{ij} of $(i;j)$ could reflect that the i's and j's neighborhoods are heterogeneous [3]. By weakening the link of $(i;j)$, the difference of the similarity between i's and j's influences is augmented, which makes it easier to observe the impact of $(i;j)$. Inspired by this idea, we build a new objective function for node embedding by the neighborhood proximity with a_{ij}.

Combining $\mathbf{W}$ with the tie strength score matrix $\mathbf{A}$, the neighborhood proximity between nodes should be the element-wise product of $\mathbf{W}$ and $\mathbf{A}$, $\mathbf{W}$ $\mathbf{A}$. Thus, the theoretical probability that $(i;j)$ is sampled with the starting point i is $p_{ij} = \exp \hat{\mathbf{c}}_i^{i\nrightarrow j} (\hat{\mathbf{c}}_j^{i\nrightarrow j} \hat{\mathbf{c}}_i) = \sum_{k \in \mathcal{V}} \exp \hat{\mathbf{c}}_i^{i\nrightarrow k} (\hat{\mathbf{c}}_k^{i\nrightarrow k} \hat{\mathbf{c}}_i)$. The empirical probability to sample this edge is $\hat{p}_{ij} = w_{ij}a_{ij} / \sum_{k \in \mathcal{V}} w_{ik}a_{ik}$. Imitating the form of Eq. (1), we build the second objective function for node embedding as:

$$O_2 = \sum_{i;j \in \mathcal{V}} w_{ij}a_{ij} \log \frac{\exp \hat{\mathbf{c}}_i^{i\nrightarrow j} (\hat{\mathbf{c}}_j^{i\nrightarrow j} \hat{\mathbf{c}}_i)}{\sum_{k \in \mathcal{V}} \exp \hat{\mathbf{c}}_i^{i\nrightarrow k} (\hat{\mathbf{c}}_k^{i\nrightarrow k} \hat{\mathbf{c}}_i)} : \quad (8)$$

The overall optimization problem to extract the node representations is $\max(O_1 + O_2)$, where is a hyper-parameter to allocate the importance of two parts. This objective function compromises the topological information between the individual level (O_1) and the neighborhood level (O_2).

Algorithm 1 Effective Learning Algorithm of IETSM

Input: $G = (V; E; W)$, d, M, K, , , and N.
Output: a_{ij} of each edge and u_i of each node.

1: Initial a_{ij} and u_i.
2: **while** less than N edges are sampled **do**
3: Sample an edge $(i; j)$ from E.
4: Sample M negative nodes according to $P_G(v)$.
5: Sample K-step random walks S_i, $S_i^{i \not\to j}$, and $S_j^{i \not\to j}$. Calculate the current $\hat{a}_{ij}$.
6: For each negative sampling v_m, sample another three K-step random walks $S_{i;v_m}$, $S_i^{i \not\to v_m}$, and $S_{v_m}^{i \not\to v_m}$.
7: Calculate the gradient of a_{ij} and all related u_i from Eq. (9), Eq. (12), and Eq. (13). Then update them by

$$a_{ij} \quad a_{ij} \quad \frac{@(O_{a_{ij}})}{@a_{ij}}; \quad u_i \quad u_i + \frac{@(O_1 + O_2)}{@u_i}:$$

8: **end while**

E. Effective Learning Algorithm

Based on the aforementioned four components of IETSM, the tie strength measuring and node embedding could complement each other. We propose the learning algorithm of IETSM in which tie strength scores and node representations are estimated simultaneously in an iterative process. In each step, we update both of them based on their values in the previous step. This is also an adaptive algorithm. Node representations are learned by adaptive weights, and tie strength scores are measured by adaptive node representations. Through this algorithm, we can achieve the estimation of tie strength scores effectively.

The steps of the algorithm are outlined in Algorithm 1. Stochastic gradient descent (SGD) is used to update a_{ij} and u_i. For a_{ij}, we can calculate its gradient from Eq. (7):

$$\frac{@O_{a_{ij}}}{@a_{ij}} = 2(a_{ij} \quad \hat{a}_{ij}); \tag{9}$$

where $\hat{a}_{ij}$ are the tie strength score estimated by the current group of random walks S_i, $S_i^{i \not\to j}$, and $S_j^{i \not\to j}$ via Eq. (6).

For related u_i, it is computationally expensive to calculate their gradients, since the denominators of both O_1 and O_2 require the summation over the entire edge set. To address this problem, we use the technique of negative sampling [30], which only samples a small number of edges to approximate the original objective function and reduce the computational complexity. For each positive sample, we draw M negative samples and approximate O_1 and O_2 by:

$$O_1 = \sum_{i;j \in \mathcal{V}} \Big[w_{ij} \log \sigma(u_i \; u_j) + \sum_{m=1}^{M} E_{v_m} \log \sigma(-u_i \; u_{v_m}) \Big] \tag{10}$$

$$O_2 = \sum_{i;j \in \mathcal{V}} \Big[w_{ij} a_{ij} \log \hat{c}_i^{i \not\to j}(\hat{c}_j^{i \not\to j} \quad \hat{c}_i) + \sum_{m=1}^{M} E_{v_m} \log \hat{c}_i^{i \not\to v_m}(\hat{c}_{v_m}^{i \not\to v_m} \quad \hat{c}_{i;v_m}) \Big]; \tag{11}$$

where v_m is independently sampled from a uniform distribution of the nodes of V. For each v_m, we generate another three K-step random walks $S_{i;v_m}$, $S_i^{i \not\to v_m}$, and $S_{v_m}^{i \not\to v_m}$ to calculate $\hat{c}_{i;v_m}$, $\hat{c}_i^{i \not\to v_m}$, and $\hat{c}_{v_m}^{i \not\to v_m}$.

After approximation, for O_1 part, the corresponding gradients of u_i, u_j, and u_{v_m} are

$$\begin{aligned} \frac{@O_1}{@u_i} &= u_j(1 \quad \sigma(u_i \; u_j)) \quad \sum_{m=1}^{M} u_{v_m} \sigma(u_i \; u_{v_m}); \\ \frac{@O_1}{@u_j} &= u_i(1 \quad \sigma(u_i \; u_j)); \quad \frac{@O_1}{@u_{v_m}} = \quad u_i \sigma(u_i \; u_{v_m}): \end{aligned} \tag{12}$$

For O_2 part, we take the example of u_l with l in $S_i^{i \not\to j}$ but not in the other random walks. The gradient of u_l is:

$$\begin{aligned} \frac{@O_2}{@u_l} &= \frac{@\hat{c}_i^{i \not\to j}}{@u_l} \frac{@O_2}{@\hat{c}_i^{i \not\to j}}; \quad \frac{@\hat{c}_i^{i \not\to j}}{@u_l} = \frac{1}{K+1}; \\ \frac{@O_2}{@\hat{c}_i^{i \not\to j}} &= a_{ij}(\hat{c}_j^{i \not\to j} \quad \hat{c}_i) \; 1 \quad \hat{c}_i^{i \not\to j} \; (\hat{c}_j^{i \not\to j} \quad \hat{c}_i) \; : \end{aligned} \tag{13}$$

The other u_i included in S_i, $S_j^{i \not\to j}$, $S_{i;v_m}$, $S_i^{i \not\to v_m}$, and $S_{v_m}^{i \not\to v_m}$ can be calculated in the similar way.

Complexity: The total number of edge samples N is $O(jEj)$, where jEj denotes the number of edges in E. The process of sampling these edges uses constant time $O(1)$ [12]. For each edge sample, the number of related nodes in random walks and negative sampling is bounded by $3(K+1)(M+1)$. The total complexity of one edge sample to calculate the gradient of all related parameters is $O(dKM)$. Therefore, the overall time complexity of IETSM is $O(dKMjEj)$. IETSM is scalable and can efficiently be applied to large-scale OSNs.

IV. Experiments

Now we perform experiments on five real-world datasets to evaluate the effectiveness and efficiency of IETSM. All datasets are publicly available and widely used in related studies. The details of them are shown as follows. Table II includes the number of nodes and edges of each dataset.

- **KDD** [8]: KDD is an author collaboration network of papers published in KDD. The number of collaboration between two authors reflects the tie strength.
- **Bitcoin-Alpha** [31]: Bitcoin-Alpha is an online who-trusts-whom network that records the trust score made to members of Bitcoin Alpha by other members.
- **Youtube1** [32]: Youtube1 is an online contact network of users of Youtube. The number of shared favorite videos between two nodes is taken as the ground truth of tie strength.
- **Flickr** [33]: Flickr is an online relationship network collected from Flickr, which is a photo management website. Users of Flickr share photos with each other and their interactions form a network. Each user is labeled with one of nine different interest groups.
- **Youtube2** [32]: Youtube2 is another online contact network of users of Youtube that includes more than one

TABLE II: Description of the Datasets

Dataset	Nodes	Edges	Ground Truth
KDD	2,892	22,416	# Collaboration
Bitcoin-Alpha	3,783	24,186	Trust Score
Youtube1	15,088	76,765	# Shared Favor Videos
Flickr	7,575	239,738	Community
Youtube2	1,138,499	2,990,443	Community

million nodes and over two million edges. Users in Youtube2 has pre-defined community information.

Baseline methods: We include five groups of methods as baseline methods. (i) To evaluate the contribution of inductive embedding in gathering the topological information, we include NO and Adamic-Adar (AA). (ii) To compare the impact analysis with the similarity analysis, we include Katz and Node Similarity (NS). (iii) To test the effectiveness of focusing on weak ties rather than strong ties to measure tie strength, we include STC-LP2. (iv) To compare our framework with the edge centrality methods, we include edge betweenness centrality (EBC). (v) To demonstrate the effectiveness of the iterative training of tie strength scores and node representations, we include IETSM-na, in which the estimated tie strength scores are not used to adjust the node representations. Their descriptions are listed below.

- **NO** [10]: Use d_i to denote the degree of node i and O_{ij} to indicate the set of nodes directly connecting with both i and j. NO index of an edge $(i; j)$ is defined as
$$\mathrm{NO}_{ij} = \frac{|O_{ij}|}{d_i + d_j \quad 2 \quad |O_{ij}|}.$$
- **AA** [34]: AA index of an edge $(i; j)$ is defined as
$$\mathrm{AA}_{ij} = \sum_{l \in \mathcal{O}_{ij}} \frac{1}{\log(d_l)}.$$
- **Katz** [16]: Katz counts the number of paths between two nodes and uses it as the estimated tie strength.
- **NS**: NS is calculated as the inner product of two nodes' representations learned by LINE of the first order proximity [12]. We can treat NS as an example to directly use the network embedding method without the impact analysis to measure tie strength.
- **STC-LP2** [8]: STC-LP2 infers the tie strength by solving a linear programming problem on strong triadic closure (STC) property.
- **EBC** [24]: EBC is calculated as the number of the shortest paths between nodes that go through an edge in the network.
- **IETSM-na**: In IETSM-na, the estimated tie strength scores are not used to adjust the node representations with = 0.

As some baseline methods only work on undirected networks, we transform all datasets to undirected networks for a fair comparison.

Experimental settings: To evaluate the performance of tie strength measuring, we follow the widely-used method described in [35] to build our evaluation metric, *mean frequency*. First, given a network as the input, we get the predicted tie strength scores of all edges of a measuring method. Based on the tertiles of the predicted scores, we divide the edges into three groups with equal size, i.e., weak, medium, and strong groups. Then the empirical tie strength such as the number of social communications through the edge is used as the ground truth of tie strength. We define *mean frequency* as the arithmetic mean of the empirical tie strength scores of edges in a group. For a good tie strength measuring method, a group of edges with low (or high) predicted tie strength scores should have relatively low (or high) empirical tie strength scores, which leads to a low (or high) mean frequency.

We set $N = 100|E|$, = 5, $M = 5$, and $K = 5$ for IETSM. The embedding dimension d is set as 100 for both NS and IETSM.

A. Effectiveness of IETSM

We use two types of empirical tie strength as the ground truth of tie strength, one based on social intimacy such as the number of collaborations and the other based on community categories. Both of them are widely used in the existing works [8], [35].

Evaluation based on social intimacy: From Table III, we observe that on the weak group of all three datasets, IETSM has the lowest mean frequency and IETSM-na has the second-lowest mean frequency. Particularly, comparing IETSM-na with the best method among the other baseline methods, measuring the tie strength based on the changes of nodes' influences via inductive embedding achieves an average of 4.81% improvement on the weak group. Comparing IETSM with IETSM-na, the iterative learning algorithm achieves an average of another 5.16% improvement on the weak group. NS does not perform well in most cases, which shows that directly using the network embedding method does not ensure to make good tie strength measuring.

For the strong group, IETSM achieves the highest mean frequency on Bitcoin-Alpha and the second-highest mean frequency on KDD but does not perform as well as NO and AA on Youtube1. Meanwhile, we observe that the mean frequency of IETSM increases over three groups of edges on all three datasets, whereas other methods could have a higher value in the weak group than the medium group. These experiments demonstrate that IETSM performs better in discovering the edges with weak tie strength than other methods, which is reasonable since we focus on the impact of an edge on the network to measure the tie strength. The impacts of weak ties are more evident than strong ties to recognize.

Evaluation based on community structure: Previous studies have shown that weak ties often work as bridges between different communities, while strong ties are more likely to connect nodes in the same community [1], [3]. Following the existing work [35], we now evaluate the performance of IETSM from the community aspect on Flickr and Youtube2 networks. We label the edges within the community as the strong ties with value 1 and the edges connecting different communities as the weak ties with value 0. Then the mean

TABLE III: Mean Frequency for Different Groups

Dataset	Method	Weak (#)	Medium	Strong (″)
KDD	NO	1.54	1.70	2.40
	AA	1.54	1.28	**2.81**
	Katz	1.81	1.36	2.46
	NS	1.52	1.97	2.14
	STC-LP2	1.86	1.89	1.89
	EBC	1.79	1.92	1.92
	IETSM-na	**1.36**	1.65	2.63
	IETSM	**1.30**	1.65	**2.69**
Bitcoin Alpha	NO	2.47	2.23	**2.82**
	AA	2.50	2.31	2.71
	Katz	2.56	2.27	2.69
	NS	2.66	2.17	2.69
	STC-LP2	2.50	2.51	2.51
	EBC	2.35	2.45	2.72
	IETSM-na	**2.31**	2.52	2.69
	IETSM	**2.21**	2.50	**2.82**
Youtube1	NO	0.94	1.44	**2.76**
	AA	0.94	1.45	**2.75**
	Katz	1.07	1.48	2.59
	NS	1.24	1.46	2.43
	STC-LP2	1.70	1.71	1.72
	EBC	0.91	1.95	2.28
	IETSM-na	**0.89**	1.62	2.62
	IETSM	**0.83**	1.78	2.51

TABLE IV: Percentage of Strong Tie for Different Groups

Dataset	Method	Weak (#)	Medium	Strong (″)
Flickr	NO	0.22	0.23	0.27
	AA	0.23	0.26	0.22
	Katz	0.24	0.26	0.22
	NS	0.17	0.22	0.32
	STC-LP2	0.23	0.24	0.24
	EBC	0.22	0.23	0.27
	IETSM-na	**0.14**	0.22	**0.35**
	IETSM	**0.14**	0.21	**0.36**
Youtube2	NO	0.48	0.54	**0.58**
	AA	0.49	0.53	0.57
	NS	0.54	0.54	0.51
	IETSM-na	**0.47**	0.53	0.56
	IETSM	**0.43**	0.54	**0.59**

frequency metric becomes the percentage of the strong tie in each group. Table IV includes the results. Katz, STC-LP2, and EBC do not have the results on Youtube2 since they are not scalable and cannot finish within a reasonable time.

We observe that IETSM has the lowest percentage of the strong tie in the weak group and the highest percentage of the strong tie in the strong group for both datasets. IETSM-na achieves the second-best performance except for the strong group of Youtube2. Compared with the second-best method except for IETSM-na, IETSM achieves 17.6% and 12.5% improvement on weak and strong groups for Flickr, and 10.4% and 1.7% improvement for Youtube2. These results show that IETSM can provide a proper order of edges in identifying the weak and strong ties to reflect the natural community similarity among the users in the network.

Case Study: To intuitively inspect the performance of IETSM, we show the estimated tie strength scores of different methods via visualization. Figure 2 shows the results of different methods on a sub-network of the Flickr dataset with randomly selected 400 users from two communities (colored by red and green respectively). For each method, we draw the edges with the top 10% (blue solid edges) and the last 10% (grey dashed edges) of the estimated tie strength scores. For a good measuring method, most of the estimated-strong ties should be the within-community edges, and most of the estimated-weak ties should be the between-community edges. Among all methods, IETSM has the most estimated-weak ties between two communities and the most estimated-strong ties within one community. NS performs the second-best but does not recover the tie strength scores of edges in the green community well. NO, AA, Katz, and EBC misidentify many between-community edges as strong ties. STC-LP2 is not shown since it performs the worst. The visualization demonstrates the effectiveness of IETSM.

B. Efficiency of IETSM

Theoretically, IETSM has a complexity of $O(dKMjEj)$. Now we test the efficiency of IETSM in practice. We construct ten sub-networks of Youtube2 by randomly selecting $10\%; 20\%; :::; 100\%$ of nodes and process IETSM to them with ten parallel threads. The relationship between the number of edges and the equivalent computation time of one thread is shown in Figure 3. We observe that as the network grows, the computation time of IETSM increases in a linear speed of the number of edges, which verifies the efficiency of IETSM.

V. Conclusion

Tie strength reflects the impact of an edge in information diffusion and benefits various real-world applications. Measuring the tie strength in OSNs is still a challenging task. We develop a novel framework, IETSM, to measure the tie strength in OSNs from the network view. We formally define the tie strength according to its properties in information diffusion. Then, IETSM measures tie strength based on the impact of an edge on the similarity between its two nodes' influences built on inductive embedding. An effective algorithm is proposed by iteratively updating learn network embedding and tie strength scores. Experiments on real-world datasets demonstrate the effectiveness and efficiency of IETSM. In future work, we plan to advance IETSM to the dynamic OSNs. In practice, the OSNs might change dynamically, which would further affect the tie strength. Another direction is to involve node attributes such as tweets and reviews to boost the tie strength measuring.

Acknowledgment

This work is, in part, supported by NSF (#IIS-1718840 and #IIS-1750074). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.