Adversarial Learning of Balanced Triangles for Accurate Community Detection on Signed Networks

Yoonsuk Kang*, Woncheol Lee*, Yeon-Chang Lee, Kyungsik Han, and Sang-Wook Kim†

Hanyang University, South Korea
{dyskang, willthor, lyc0324, kyungsikhan, wook}@hanyang.ac.kr

Abstract—In this paper, we propose a framework for embedding-based community detection on signed networks. It first represents all the nodes of a signed network as vectors in low-dimensional embedding space and conducts a clustering algorithm (e.g., k-means) on vectors, thereby detecting a community structure in the network. When performing the embedding process, our framework learns only the edges belonging to balanced triangles whose edge signs follow the balance theory, significantly excluding noise edges in learning. To address the sparsity of balanced triangles in a signed network, our framework learns not only the edges in balanced real-triangles but those in balanced virtual-triangles that are produced by our generator. Finally, our framework employs adversarial learning to generate more-realistic balanced virtual-triangles with less noise edges. Through extensive experiments using seven real-world networks, we validate the effectiveness of (1) learning edges belonging to balanced real/virtual-triangles and (2) employing adversarial learning for signed network embedding. We show that our framework consistently and significantly outperforms the state-of-the-art community detection methods in all datasets.

Index Terms—adversarial learning, balanced triangle, community detection, signed network

I. INTRODUCTION

A community in an unsigned network consists of a set of nodes that have similar characteristics [1]. Community detection (CD) on a given unsigned network is a task of identifying a set of communities by leveraging topological properties of a community structure: (1) intra-connections within each community are dense and (2) inter-connections between communities are sparse [1]. To detect a community structure more accurately, many CD methods have been proposed in the literature [2]–[6].

The emergence of signed networks with both positive and negative edges enables us to better understand complex relationships among nodes in the network based on the edge signs [7]–[12]. However, many existing CD methods targeted for unsigned networks assume networks to have only positive edges, having inherent limitations in utilizing rich information (i.e., edge signs) provided by signed networks [13]. Therefore, CD methods are required to utilize both positive and negative edges in signed networks.

A community structure in a signed network has following topological properties: (1) most intra-connections within each community have positive signs and (2) most inter-connections between communities have negative signs [13], [14]. There have been many attempts to detect a community structure by exploiting these properties in the literature [13], [15]–[19]. However, in a real-world signed network, there are pairs of nodes connected by a negative edge even though they belong to the same community; conversely, there are pairs of nodes connected by a positive edge even though they belong to different communities. Since most CD methods on signed networks learn an edge as a unit of detecting a community structure, it is inevitable to learn these wrong edges. We call such an edge that misleads CD methods in a wrong way a noise edge. The accuracy of CD would decrease as a CD method learns more noise edges.

A signed triangle in a signed network is a set of three nodes, each of which is connected to every other node with a signed edge. There are four types of signed triangles according to the combination of signs: (1) a triangle with three positive edges, denoted as $\Delta_{+++}$; (2) a triangle with two positive edges and a negative edge, denoted as $\Delta_{+--}$; (3) a triangle with one positive edge and two negative edges, denoted as $\Delta_{+-+}$; and (4) a triangle with three negative edges, denoted as $\Delta_{--+}$ [8]. A balance theory [20], a well-known theory in social science, says that social relationships in the real world tend to follow four rules: “a friend of my friend is my friend,” “a friend of my enemy is my enemy,” “an enemy of my friend is my enemy,” and “an enemy of my enemy is my friend.” Under the balance theory, $\Delta_{+++}$ and $\Delta_{+--}$ follow the above rules, thus classified as balanced while $\Delta_{+-+}$ and $\Delta_{--+}$ violate the rules, thus classified as unbalanced. We assume that learning only the edges in balanced triangles can significantly reduces noise edges to be learned, thereby helping improve the CD accuracy. Based on this assumption, in this paper, we propose a framework for network embedding (NE)-based CD on signed networks, named as Adversarial learning of Balanced triangle for Community detection (ABC). It first represents all the nodes in a signed network as vectors in low-dimensional embedding space via our signed NE and conducts a clustering algorithm (e.g., k-means) on vectors, detecting a community structure in the network.

In order to filter out noise edges, we first propose a strategy (S1) for signed NE of ABC that learns only the edges in the balanced real-triangles that exist in the network. Real-world networks, however, are very sparse [21]. They will become more sparse in our case since we only consider the edges belonging to balanced real-triangles. To address the sparsity problem, we propose a strategy (S2) that learns not only the

*Two first authors have contributed equally to this work.
† Corresponding author.
edges in balanced real-triangles but also the edges in the balanced virtual-triangles that do not actually exist in the network but are generated by ABC based on the distribution of balanced real-triangles.

Since balanced virtual-triangles are generated based on balanced real-triangles, noise edges can exist in balanced virtual-triangles as well. Thus, to generate balanced virtual-triangles having less noise edges, we propose a strategy (S3) that employs adversarial learning [22]–[24].

Based on the three strategies, ABC learns few noise edges and provides an embedding space that reveals the community structure more clearly. As a result, ABC can easily identify which community each node belongs to by conducting k-means clustering algorithm over the vectors in an embedding space.

Our contributions are summarized as follows:

- We propose a novel embedding-based CD framework, named as ABC, which employs three strategies:
  - (S1) learning only the edges in the balanced triangles to reduce noise edges in learning.
  - (S2) learning not only the edges in balanced real-triangles but also the edges in balanced virtual-triangles to address the data sparsity problem.
  - (S3) employing adversarial learning to generate balanced virtual-triangles with less noise edges.

- Through extensive experiments using seven real-world networks, we show the superiority of ABC in CD as follows:
  - We validate the effectiveness of each of our three strategies.
  - We demonstrate that our ABC consistently and significantly outperforms the state-of-the-art CD methods in all datasets in terms of the CD accuracy.

II. MOTIVATION

In this section, we present the advantages of utilizing signed triangles in CD. Note that, in real-world signed networks, there may be the cases of two nodes with a negative edge in the same community and those of two nodes with a positive edge in different communities. In this paper, we define these edges that have a sign against the community structure as noise edges.

Existing CD methods on signed networks learn all noise edges, which adversely affects CD accuracy.

Suppose that the result of judging whether two nodes are in the same community based on the edge sign can be confirmed from the view point of a third party (i.e., another node). In this case, we can reduce a number of noise edges by considering only the confirmed edges in learning. Since three nodes in a signed triangle are all connected to each other, such a signed triangle can be utilized as a unit to confirm the above judgment as follows: we judge that two nodes \(v_a\) and \(v_b\) are in the same community (resp. different communities) if \(v_a\) and \(v_b\) are connected by a positive (resp. negative) edge. If \(v_a\) and \(v_b\) belong to a balanced triangle (case 1), the actual edge sign between them and the inferred edge sign between them by the balance theory from the view point of a remaining node \(v_c\) are identical. In this case, we will be more confident with the judgment based on the actual edge sign; thus we use the edge \(e_{ab}\) with high confidence for learning. However, if \(v_a\) and \(v_b\) do not belong to any signed triangle (case 2), there is no additional information to confirm the judgment on their edge sign. Also, if \(v_a\) and \(v_b\) are in an unbalanced triangle (case 3), the actual edge sign between them is different from the inferred edge sign according to the balance theory. In the last two cases, we will not be confident with the judgments based on the edge sign between \(v_a\) and \(v_b\); thus, we exclude their edge \(e_{ab}\) from learning, considering \(e_{ab}\) as a noise edge.

Fig. 1 demonstrates the process of verifying the edge using a signed triangle, where a signed network consists of seven nodes, nine positive edges, and three negative edges. Suppose that the network has two communities: \(C_1 = \{v_a, v_b, v_c, v_d\}\) and \(C_2 = \{v_e, v_f, v_g\}\). Here, edge \(e_{ab}\) is not regarded as a noise edge since it is in two balanced triangles (i.e., \(\Delta_{+,+}\)) (case 1); thus, we include \(e_{ab}\) for learning. On the other hand, edge \(e_{bf}\) and edge \(e_{cd}\) are noise edges: (1) \(v_b\) and \(v_f\) are connected by a positive edge even though they are in different communities; (2) \(v_c\) and \(v_d\) have a negative edge even though they are in the same community. We can recognize \(e_{bf}\) and \(e_{cd}\) as noise edges to be excluded from learning because \(e_{bf}\) does not belong to any signed triangle (case 2) and \(e_{cd}\) is in two unbalanced triangles (i.e., \(\Delta_{-,+}\)) (case 3).

III. ABC: PROPOSED FRAMEWORK

A. Overview

In this paper, we propose a novel embedding-based CD framework using balanced triangles, namely an ABC. It first represents all the nodes in a signed network as vectors in low-dimensional embedding space and conducts a clustering algorithm (e.g., k-means) on the vectors, thereby detecting a community structure in the network.

The most important challenge in ABC is how to reveal the community structure accurately in the embedding space. Therefore, the main goal of this paper is to solve the problem formally defined as follows: given a signed network \(\bar{S} = (V, E^+, E^-)\), where \(V = \{v_1, v_2, \ldots, v_s\}\) is a set of \(s\) nodes and \(E^+\) and \(E^-\) represent the sets of positive and negative edges, respectively. Note that \(E^+ \cap E^- = \emptyset\), a node pair cannot have both positive and negative edges simultaneously. We use \(A \in \mathbb{R}^{n \times s}\) to denote an adjacency matrix, where \(a_{ab} = 1\) denotes a positive edge between \(v_a\) and \(v_b\), \(a_{ab} = -1\) denotes a negative edge, and \(a_{ab} = 0\) denotes the edge missing. We aim to learn a function \(f: V \rightarrow \mathbb{R}^n\) that maps each node \(v_a \in V\) into a \(n\)-dimensional vector \(\phi_a \in \mathbb{R}^n\) in such a way that two vectors
\(\phi_a\) and \(\phi_b\) are close to each other in \(\mathbb{R}^n\), if \(v_a\) and \(v_b\) are in the same community; on the other hand, two vectors \(\phi_a\) and \(\phi_b\) are far from each other if \(v_a\) and \(v_b\) are in different communities.

Before proceeding, we define some key terminology used throughout this paper.

**Definition 1** (real-edge). An edge \(e_{ab}\) is defined as a real-edge, if \(a_{ab}\) is \(-1\) or \(1\), rather than \(0\).

**Definition 2** (virtual-edge). An edge \(e_{ab}\) is defined as a virtual-edge, if \(a_{ab}\) is \(0\) (i.e., missing edge) and its sign can be easily predicted by the rules of the balance theory.

**Definition 3** (real-triangle). A triangle \((v_a, v_b, v_c)\) is defined as a real-triangle, if \(e_{ab}, e_{bc}, \) and \(e_{ca}\) are all real-edges.

**Definition 4** (virtual-triangle). A triangle \((v_a, v_b, v_c)\) is defined as a virtual-triangle, if at least one of \(e_{ab}, e_{bc},\) and \(e_{ca}\) is virtual-edge.

**Definition 5** (balanced real-triangle \((\triangle)\)). A real-triangle \((v_a, v_b, v_c)\) is defined as balanced, if the signs of \(e_{ab}, e_{bc},\) and \(e_{ca}\) follow the rules in the balance theory.

**Definition 6** (balanced virtual-triangle \((\triangle)\)). A virtual-triangle \((v_a, v_b, v_c)\) is defined as balanced, if the signs of \(e_{ab}, e_{bc},\) and \(e_{ca}\) follow the rules in the balance theory.

Fig. 2 shows the overall process of signed NE in ABC, composed of two components: a generator \((G)\) and a discriminator \((D)\). For adversarial learning, \(G\) and \(D\) act as a pair of opponents in the two-play minimax game. \(G\) and \(D\) have their own embedding space \(\theta_G\) and \(\theta_D\), respectively. As shown in Fig. 2-(a), \(G\) directly samples balanced real-triangles from a given signed network and learns the edges in the balanced real-triangles (Fig. 2-(c)); under the guidance provided by \(D\), \(G\) generates (realistic) balanced virtual-triangles and learns the edges in those virtual-triangles (Fig. 2-(d)); and \(G\) updates \(\theta_G\) based on the two learned results (Fig. 2-(e)). On the other hand, as shown in Fig. 2-(b), \(D\) discriminates balanced real-triangles sampled directly from the given network and balanced virtual-triangles generated from \(G\) (Fig. 2-(f)) and \(D\) updates the discriminated results in \(\theta_D\) (Fig. 2-(g)). Finally, ABC uses \(\theta_D\) for conducting \(k\)-means to detect the community structure in the network. Table I summarizes a list of notations used in this paper.

**B. Generator**

For a strategy (S1), we select three nodes \((v_a, v_b, v_c)\) as a balanced real-triangle containing a target node \(v_a\), where every pair of nodes \((v_a, v_b), (v_a, v_c),\) and \((v_b, v_c)\) is directly connected via a real-edge and the signs of real-edges follow the rules in the balance theory. For a strategy (S2), we generate balanced virtual-triangle containing \(v_a\) as follows. \(G\) first generates a path with a fixed length \(l\) \((l \geq 3)\) starting from \(v_a\), denoted as \(Path_{v_a}\), by using the biased random walk. More specifically, \(G\) first determines the sign of the edge to walk by considering \(\alpha_a\) of \(v_a\), where \(\alpha_a\) (resp. \((1 - \alpha_a)\)) is the ratio of positive (resp. negative) edges to all the edges incident to \(v_a\). Then, it determines a node to walk based on the walking probabilities between \(v_a\) and its neighboring nodes, each of which is connected by the edge with the sign determined above. In other words, the walking probabilities \(p^+(v_a, v_i)\) (resp. \(p^-(v_a, v_i)\)) between \(v_a\) and its neighboring node \(v_i\) connected by the positive edge (resp. negative edge) are computed by (Eq. 1) (resp. (Eq. 2)) below:

---

**Table I**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathcal{S})</td>
<td>Signed network</td>
</tr>
<tr>
<td>(V = {v_1, v_2, \ldots, v_n})</td>
<td>Set of nodes</td>
</tr>
<tr>
<td>(E, E^+, E^-)</td>
<td>Sets of all, positive, and negative edges</td>
</tr>
<tr>
<td>(C = {c_1, c_2, \ldots, c_m})</td>
<td>Set of communities</td>
</tr>
<tr>
<td>(\Delta_-, \Delta^+, \Delta^\perp)</td>
<td>Balanced triangle, balanced real-triangle, and balanced virtual-triangle</td>
</tr>
<tr>
<td>(Path_{v_a, l})</td>
<td>A path starting from (v_a) and length of a path</td>
</tr>
<tr>
<td>(N^-<em>{v_a}, N^+</em>{v_a})</td>
<td>Sets of nodes directly connected to (v_a) by the positive and negative edge in (\mathcal{S})</td>
</tr>
<tr>
<td>(D(\Delta, v_a; \theta_D), G(\Delta, v_a; \theta_G))</td>
<td>Discriminator and generator for (\mathcal{S})</td>
</tr>
<tr>
<td>(\phi_a, \phi_b \in \mathbb{R}^n)</td>
<td>(n)-dimensional representation vectors of (v) in (\mathcal{S})</td>
</tr>
<tr>
<td>(\theta_D, \theta_G \in \mathbb{R}^n)</td>
<td>Union of all (\phi_a) and (\phi_b)</td>
</tr>
</tbody>
</table>
---

1152
\begin{equation}
    p^+(v_a, v_i) = \frac{s(\phi_a, \phi_i)}{\sum_{i' \in N_{v_a}} s(\phi_a, \phi_{i'})},
    \tag{1}
\end{equation}
\begin{equation}
    p^-(v_a, v_i) = 1 - \frac{s(\phi_a, \phi_i)}{\sum_{i' \in N_{v_a}} s(\phi_a, \phi_{i'})},
    \tag{2}
\end{equation}

where $N^+_{v_a}$ and $N^-_{v_a}$ indicate the sets of neighboring nodes connected with $v_a$ via positive and negative edges, respectively. $s(\phi_a, \phi_i)$ indicates the similarity (i.e., the inverse of the Euclidean distance with min-max normalization) between $v_a$ and $v_i$.

After $Path_{b_{v_a}}$ is generated through above process, $G$ generates balanced virtual-triangles of all possible triplets of nodes that contain $v_a$ in $Path_{b_{v_a}}$. At this time, the virtual-edge signs in the balanced virtual-triangles are determined by the rules of the balance theory. Specifically, if there are an even number of negative real-edges between the two nodes to be connected by a virtual-edge in $Path_{b_{v_a}}$, the virtual-edge sign is determined as positive; on the other hand if there are an odd number of negative real-edges, the virtual-edge sign is determined as negative. Note that, since the edge sign is determined in this way, $G$ always generates balanced virtual-triangles (i.e., $v_{\Delta_\ldots}$ and $v_{\Delta_{\ldots\ldots}}$) only.

Now, we discuss the implementation and optimization of $G$. Given balanced virtual-triangles, $G$ aims to minimize the log-probability of correctly classifying these balanced triangles. Since three nodes in the balanced triangle are connected to each other, we learn the three edges of the triangle together, rather than learning a single edge individually [25], [26]. To do this, we define $G$ for a given balanced real/virtual-triangle $\Delta = (v_a, v_b, v_c)$ of a target node $v_a$ as follows:

\begin{equation}
    G(\Delta | v_a) = s(\phi_a, \phi_b)s(\phi_a, \phi_c)s(\phi_b, \phi_c).
    \tag{3}
\end{equation}

The higher value of $G(\Delta | v_a)$ indicates that there are more positive edges in the balanced triangle $\Delta$ containing $v_a$.

Finally, given a balanced real/virtual-triangle $\Delta = (v_a, v_b, v_c)$ for $\mathcal{G}/G$, respectively, we update $\theta_G$ (i.e., $\phi_a$, $\phi_b$, and $\phi_c$) by ascending the gradient of $L(G, D)$ with respect to three nodes $v_a, v_b, v_c$ as follows:

\begin{equation}
    \nabla_{\theta_G} L(G, D) = \nabla_{\theta_G} \log G(\Delta | v_a) \log (1 - D(\Delta, v_a)),
    \tag{4}
\end{equation}

where $\log (1 - D(\Delta, v_a))$ indicates the policy gradient [26], [27], which will be elaborated in the next section.

C. Discriminator

$D$ is designed for a strategy (S3) to generate balanced virtual-triangles having less noise edges. For a target node, $D$ (1) samples a balanced real-triangle containing $v_a$ directly from a given signed network $\mathcal{G}$; (2) receives a balanced virtual-triangle corresponding to the balanced real-triangle from $G$; (3) discriminates the balanced real/virtual-triangles; and (4) updates the discriminated results in $\theta_D$. Let us present a detailed procedure in each step.

\begin{enumerate}
    \item In step (1), among multiple balanced virtual-triangles generated by $G$, only one is selected for this discrimination. Inspired by generating fake motif in [5] for unsigned networks, we set a balanced virtual-triangle that contains the following three nodes: a target node, the first node visited from the target node, and the last node in the path. Note that one of the three edges in the balanced virtual-triangle produced from this way should be a real-edge and the other two edges should be virtual-edges.
    \item Now, we present how to discriminate the balanced real/virtual-triangles. As mentioned earlier, one of the edges in a balanced virtual-triangle $v_{\Delta}$ is real. In this case, even if $D$ correctly identifies $v_{\Delta}$ as a balanced virtual-triangle, $D$ should not consider the real-edge in $v_{\Delta}$ as a virtual-edge in learning. Thus, we make $D$ discriminate only two virtual-edges (i.e., excluding the real-edge) in a balanced virtual-triangle from two real-edges in a balanced real-triangle. As a result, we do not use the real-edge in the balanced virtual-triangle in the discrimination process. Thus, with two virtual/real-edges, there are three combinations of signs: (1) two positive edges (i.e., ++); (2) one positive and one negative edges (i.e., +−); and (3) two negative edges (i.e., −−).
    \item We discuss the implementation and optimization of $D$. Given the balanced real/virtual-triangles, $D$ aims to maximize the log-probability of correctly classifying these balanced triangles. We present the implementation of $D$ for two edges $(e_{ac}, e_{bc})$ in a given balanced real/virtual-triangle $\Delta = (v_a, v_b, v_c)$ of a target node $v_a$ as follows:
    \begin{equation}
        D(\Delta, v_a)
        \begin{cases}
            s(\phi_d, \phi_e)s(\phi_d, \phi_f), & \text{if two edges in } \Delta \text{ are } (++). \\
            s(\phi_d, \phi_e)(1-s(\phi_d, \phi_f)), & \text{if two edges in } \Delta \text{ are } (+-). \\
            (1-s(\phi_d, \phi_e))(1-s(\phi_d, \phi_f)), & \text{if two edges in } \Delta \text{ are } (-). 
        \end{cases}
        \tag{5}
    \end{equation}
    The higher value of $D(\Delta, v_a)$ indicates that the balanced triangle $\Delta$ containing $v_a$ is more likely to be a balanced real-triangle. Finally, given a balanced real/virtual-triangle $\Delta = (v_a, v_b, v_c)$ for $D$, we update $\theta_D$ (i.e., $\phi_d$, $\phi_e$, and $\phi_f$) by ascending the gradient with respect to three nodes $v_a, v_b, v_c$ as follows:
    \begin{equation}
        \nabla_{\theta_D} L(G, D)
        \begin{cases}
            \nabla_{\theta_D} \log D(\Delta, v_a), & \text{if } \Delta \text{ is a balanced real-triangle from } \mathcal{G}; \\
            \nabla_{\theta_D} (1-\log D(\Delta, v_a)), & \text{if } \Delta \text{ is a balanced virtual-triangle from } G.
        \end{cases}
        \tag{6}
    \end{equation}
\end{enumerate}

D. Overall Optimization

In this section, we present the overall optimization of ABC based on adversarial learning. Given $\mathcal{G}$, we aim to learn the following two components:

- **Generator** $G(v_{\Delta} | v_a; \theta_G)$ aims to generate a realistic balanced virtual-triangle containing $v_a$ by approximating the distribution of the balanced real-triangle containing $v_a$ (i.e., $p_{\text{true}}(v_{\Delta} | v_a)$).
- **Discriminator** $D(\Delta, v_a; \theta_D)$ aims to discriminate labels (i.e., real or virtual) for the balanced triangles ($\Delta$ or $v_{\Delta}$) by estimating the probability that $\Delta$ is a balanced real-triangle.

$G$ and $D$ are combined while playing a minimax game. In $\theta_G$, $G$ tries to embed two nodes in the balanced triangle
connected by a positive real/virtual edge to be close and two nodes in the balanced triangle connected by a negative real/virtual edge to be distant. On the other hand, in \( \theta_D \), \( D \) embeds two nodes in the balanced triangle connected by a positive real-edge (resp. a negative virtual-edge) to be close and those connected by a positive virtual-edge (resp. a negative real-edge) to be distant. Formally, \( G \) and \( D \) act as a pair of opponents in the following two-player minimax game with the joint loss function \( L(G,D) \):

\[
\begin{align*}
\min_{\theta_G} \max_{\theta_D} L(G,D) &= \sum_{v_a \in V} \left[ (E_{\triangle \sim p_{real}(\theta_G)}) \log D_{\theta_D}(\triangle, v_a; \theta_G) \right] \\
&+ (E_{\triangle \sim p_{virtual}(\theta_G)}) \log (1 - D_{\theta_D}(\triangle, v_a; \theta_G))].
\end{align*}
\]

(7)

IV. EVALUATION

We designed our experiments, aiming at answering the following key research questions (RQs):

- **RQ1**: Does learning the edges in balanced real-triangles and those in balanced virtual-triangles help to improve the accuracy of CD?
- **RQ2**: Does our adversarial learning of balanced triangles help to improve the accuracy of CD?
- **RQ3**: Does ABC provide CD accuracy higher than the state-of-the-art CD methods?

A. Experimental Settings

For the experiments, we use seven real-world signed networks provided by [19]: Medical, Enron, Rcv1., Science, Bibtex, Health, and Eurlx as shown in Table II. We compare ABC with the following four state-of-the-art CD methods for signed networks: FEC [13], UAC [17], GML [15], and PML [16]. We use the source codes provided by the authors [13], [15]–[17]. Following [5], [15]–[17], we set the number of communities to detect as the number of ground-truth communities contained in each dataset. For parameters for each method, we use the best setting found via extensive grid search in the ranges suggested in its respective paper. For ABC, the best parameter setting is found as follows: \( n = 128 \), learning rate=0.001, \( e = 35 \), \( i_e = 3 \), \( l = 5 \). For initializing \( \theta_G \) and \( \theta_D \), we use SLF [25].

For the accuracy measure, we use normalized mutual information (NMI), which is the most widely used metric for evaluating the accuracy of CD methods [28]. It is an information-theoretic metric that measures the similarity between ground-truth communities and detected communities by a CD method, producing a value between 0 and 1. A higher score of NMI indicates that the detected communities match the ground-truth communities more closely.

B. Experimental Results

**RQ1: Effectiveness of learning not only the edges belonging to balanced real-triangles but also those in balanced virtual-triangles.** To this end, we first make two variants of ABC according to learning strategies as follows: (1) \( ABC_{(r)} \) learns only the edges belonging to balanced real-triangles; (2) \( ABC_{(r+)} \) learns only the edges belonging to balanced real/virtual-triangles. Note that both \( ABC_{(r)} \) and \( ABC_{(r+)} \) are generative models that do not employ adversarial learning. Fig. 3 shows the accuracy of the two variants in terms of NMI. We observe that \( ABC_{(r+)} \) consistently outperforms \( ABC_{(r)} \) in all datasets, which indicates that learning the edges belonging to both of balanced real/virtual-triangles helps detect the community structure more accurately, compared to learning the edges belonging to balanced real-triangles only.

**RQ2: Effectiveness of employing adversarial learning.** To do this, we compare ABC with employing adversarial learning, denoted as \( ABC_{(r+v)} \), and ABC without employing adversarial learning (\( ABC_{(r+v-a)} \)). Fig. 3 shows that \( ABC_{(r+v-a)} \) consistently and significantly outperforms \( ABC_{(r+v)} \) in all datasets, which validates that our adversarial learning generates balanced virtual-triangles having less noise edges, which results in improving the accuracy of CD.

**RQ3: Comparison of ABC with state-of-the-art CD methods.** We conduct comparative experiments to demonstrate greater accuracy of ABC than that of the following four competing CD methods for signed networks: FEC [13], UAC [17], GML [15], and PML [16]. We summarize the result shown in Table III as follows. First, the best performer among the competing methods is different, depending on datasets: GML for Medical, Bibtex, and Health; PML for Enron, Rcv1., Science, and Eurlx.\(^1\) Second, ABC consistently, and signifi-

---

**TABLE II**

| Dataset    | \(|V|\) | \(|E|\) | \(|E^+|\) | \(|E^-|\) | \(|C|\) |
|------------|--------|--------|--------|--------|--------|
| Medical    | 978    | 31,000 | 23,196 | 7,804  | 31     |
| Enron      | 1,702  | 52,156 | 37,367 | 14,789 | 51     |
| Rcv1       | 6,000  | 183,728| 134,203| 49,525 | 99     |
| Science    | 6,428  | 218,010| 156,993| 61,017 | 40     |
| Bibtex     | 7,395  | 220,056| 171,836| 48,220 | 159    |
| Health     | 9,205  | 355,826| 254,154| 101,672| 27     |
| Eurlx      | 19,348 | 676,210| 554,374| 121,836| 372    |

**Fig. 3.** Accuracy comparison between ABC variants.

**TABLE III**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( \text{FEC} )</th>
<th>( \text{UAC} )</th>
<th>( \text{GML} )</th>
<th>( \text{PML} )</th>
<th>( \text{ABC} )</th>
<th>( \text{Gain (%)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical</td>
<td>0.538 ( \times 10^{-16} )</td>
<td>0.5739</td>
<td>0.4764</td>
<td>0.4595</td>
<td>0.4954</td>
<td>9.98</td>
</tr>
<tr>
<td>Enron</td>
<td>0.0111</td>
<td>0.0634</td>
<td>0.1076</td>
<td>0.1136</td>
<td>0.1372</td>
<td>20.77</td>
</tr>
<tr>
<td>Rcv1</td>
<td>0.0001</td>
<td>0.1685</td>
<td>0.1875</td>
<td>0.1933</td>
<td>0.2087</td>
<td>7.99</td>
</tr>
<tr>
<td>Science</td>
<td>0.0098</td>
<td>0.0898</td>
<td>0.1154</td>
<td>0.1352</td>
<td>0.1853</td>
<td>37.06</td>
</tr>
<tr>
<td>Bibtex</td>
<td>0.0016</td>
<td>0.4037</td>
<td>0.4603</td>
<td>0.4577</td>
<td>0.4763</td>
<td>3.48</td>
</tr>
<tr>
<td>Health</td>
<td>0.0005</td>
<td>0.1598</td>
<td>0.1756</td>
<td>0.1685</td>
<td>0.2120</td>
<td>20.73</td>
</tr>
<tr>
<td>Eurlx</td>
<td>0.O.O.T</td>
<td>0.4533</td>
<td>0.O.O.T</td>
<td>0.5486</td>
<td>0.5671</td>
<td>3.38</td>
</tr>
</tbody>
</table>

\(\text{O.O.T. out-of-time}\)

---

\(^1\)FEC and GML do not work on the Eurlx dataset within 24 hours.
cantly outperforms all competing methods in all datasets. From the result, we conclude that (1) our (S1) enables ABC to avoid learning of many noise edges, in such a way to consider only the edges in balanced triangles; (2) our (S2) considers the edges in balanced virtual-triangles as well, thus helping address the data sparsity problem; and (3) our (S3) employs adversarial learning, which enables to generate balanced virtual-triangles having less noise edges. Thanks to the three strategies, our ABC framework achieves great improvement in the CD accuracy, compared with the state-of-the-art methods.

V. CONCLUSIONS

In this paper, we proposed an embedding based community detection framework, named as ABC. Specifically, ABC first represents all the nodes of a given signed network as a low-dimensional vectors based on the following three strategies: (S1) learning only the edges in the balanced triangles; (S2) learning the edges in the balanced real/virtual-triangles; and (S3) employing adversarial learning. With these strategies, ABC can provide embeddings that reflect the community structure in a network. Finally, ABC conducts k-means to detect community structure. Through comprehensive experiments using seven real-world signed networks for CD, we demonstrated that each of our strategies is effective and ABC integrating all strategies is the most effective in achieving the high accuracy of CD. In addition, our experimental results showed that ABC consistently and significantly outperforms all the state-of-the-art CD methods in all datasets.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (NRF-2020R1A2B5B03001960 and 2018R1A5A7059549) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korean government (MSIT) (NRF-2020R1A2B5B03001960) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the MSIT (No. 2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)).

REFERENCES