On Cohesively Polarized Communities in Signed Networks

Jason Niu
University at Buffalo
Buffalo, NY, USA
jasonniu@buffalo.edu

Ahmet Erdem Sarıyüce
University at Buffalo
Buffalo, NY, USA
erdem@buffalo.edu

ABSTRACT

Locating and characterizing polarization is one of the most important issues to enable a healthier web ecosystem. Finding groups of nodes that form strongly stable agreements and participate in collective conflicts with other groups is an important problem in this context. Previous works approach this problem by finding balanced subgraphs, in which the polarity measure is optimized, that result in large subgraphs without a clear notion of agreement or conflict. In real-world signed networks, balanced subgraphs are often not polarized as in the case of a subgraph with only positive edges. To remedy this issue, we leverage the notion of cohesion — we find pairs of cohesively polarized communities where each node in a community is positively connected to nodes in the same community and negatively connected to nodes in the other community. To capture the cohesion along with the polarization, we define a new measure, dichotomy. We leverage the balanced triangles, which model the cohesion and polarization at the same time, to design a heuristic that results in good seedbeds for polarized communities in real-world signed networks. Then, we introduce the electron decomposition which finds cohesively polarized communities with high dichotomy score. In an extensive experimental evaluation, we show that our method finds cohesively polarized communities and outperforms the state-of-the-art methods with respect to several measures. Moreover, our algorithm is more efficient than the existing methods and practical for large-scale networks.

KEYWORDS

balance, cohesion, polarized communities, signed networks

ACM Reference Format:
Jason Niu and Ahmet Erdem Sarıyüce. 2023. On Cohesively Polarized Communities in Signed Networks. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3543873.3587698

1 INTRODUCTION

The growing controversy surrounding the issues of today's society, especially those related to political topics, has led to increased interactions between people with alike and opposite opinions. However, excessive polarization in social media platforms impacts the health of public discourse and democracy. Characterization, detection, and mitigation of polarized groups is a timely and compelling problem which has been thoroughly studied in recent years [6, 10, 19, 25, 26, 35, 42–45, 53]. Polarized groups are often characterized as a pair of communities where nodes form strongly stable agreements in their own community and participate in collective conflicts with the nodes from other community [9, 52, 55]. In addition, the scale of the social media platforms poses new algorithmic challenges for efficient analysis of the available data [9].

Signed networks are a powerful tool to model positive and negative interactions, such as friend-foe and trust-distrust relations [12, 31]. One classical measure to identify polarized groups is the balance, which measures the stability according to the placement of positive and negative edges. Heider defined that a signed graph is balanced if all its cycles are positive—a cycle is positive if it contains an even number of negative edges [31]. A common measure for partial balance is the fraction of the balanced triangles +++ and +−− [5, 12]. There has been some recent studies that suggest balanced subgraphs as a proxy for polarized communities in signed networks [9, 46, 52, 55]. One common disadvantage of these works is that they optimize an ill-defined measure, *polarity*, which results in large subgraphs without a clear notion of agreement or conflict. The main reason for this behavior is that +++ triangles in real-world signed networks are significantly more abundant than +−− triangles, thus +++ dominates the resulting balanced subgraphs in which the conflicts cannot be captured well.

In this work, we propose to leverage the notion of cohesion to find polarized communities. Although finding cohesive subgraphs is a fundamental graph mining problem for all kinds of networks with key applications [2, 3, 20, 23, 27, 28, 38, 47, 51], it is not much studied in signed networks. Existing studies only consider strict models such as cliques [41] or focus on streaming workloads [11] but have not leveraged the cohesion while finding polarized communities. To capture the cohesion along with the polarity to better model the agreement within communities and the conflict across communities, we devise the *dichotomy* measure, which characterizes the extent of polarization for a given pair of communities. We introduce an efficient heuristic that takes advantage of the balanced triangles, which model both the cohesion and the polarization. Inspired by the truss decomposition [16], which finds cohesive regions with hierarchical relations in simple unsigned networks, we propose *atom decomposition* to find signed triangle-specific cohesive subgraphs. We show that not only are balanced triangles more abundant in the real-world signed networks, but they are also significantly closer to each other than expected, thus tend to form good seedbeds for cohesively polarized subgraphs with high dichotomy. Building on this observation, we propose *electron decomposition* which finds polarized communities

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3587698

with high dichotomy in an efficient way. In an extensive experimental evaluation on real-world and synthetic networks, we show that electron decomposition finds better polarized communities than the state-of-the-art with respect to various measures. Our contributions can be summarized as follows:

- **Dichotomy to model polarized communities.** We use polarity and cohesion to define the *dichotomy* measure which targets large communities with proportional sizes.
- **Triangle-based heuristics.** We propose the *atom decomposition* to find signed triangle-specific subgraphs and show that it obtains good seedbeds for polarized communities. Then, we introduce the *electron decomposition* to find cohesively polarized communities.
- **Evaluation.** We compare our results against several baselines on real-world and synthetic networks with respect to several measures. Electron decomposition obtains non-trivial size communities of higher quality than the state-of-the-art methods. We present a case study on the Correlates of War dataset to showcase anecdotal examples. Furthermore, our algorithm is more efficient than the alternative methods and is practical for large networks.

2 PRELIMINARIES

We work on a simple and undirected signed graph $G = (V, E)$ where V is the set of nodes and $E = E^+ \cup E^-$ is the set of edges such that E^+ and E^- are the sets of positive and negative edges, respectively. The neighbors of a node v is denoted by $N(v)$. We define each triangle in a signed network to be of type +++ , +−− , ++− , or −−− , where each + and − is the sign of a unique edge in the triangle.

In this work, we aim to find a subgraph that consists of two polarized communities where the nodes in the same community are connected with positive edges and the nodes from different communities are connected with negative edges. We denote the target subgraph, $S = (V_S, E_S)$, as the union of left and right communities, denoted by (V_L, E_L) and (V_R, E_R), and the edges across the left and right communities, E_{LR}. Hence, $V_S = (V_L \cup V_R)$ and $E_S = (E_L \cup E_R \cup E_{LR})$. Any edge set with a sign superscript denotes the subset of edges with that sign, e.g., E_L^- is the set of negative edges in E_L. Without loss of generality, we assume that the larger of two communities (in number of nodes) is called the left community, hence $|V_L| \geq |V_R|$ by default.

Balance measures. A graph is balanced if its node set can be partitioned into two subsets such that each negative edge joins nodes from different subsets [30]. A popular measure for partial balance is the relative 3-balance — the ratio of the number of balanced triangles to the total number of triangles in the graph [5]. Triangles are preferred when characterizing the partial balance since triangles represent the strongest interactions [8]. However, relative 3-balance, along with other balance measures such as degree of balance and normalized frustration index [5], does not guarantee polarization as in the case of a graph with only positive edges.

Cohesive subgraphs and truss decomposition. The problem of truss decomposition is based on cohesive subgraph discovery. The cohesion of a subgraph is measured in terms of the **edge ratio**, which is the ratio of the number of edges in the subgraph to the number of node pairs. The most intuitive definition for a cohesive subgraph is a clique in which every pair of nodes is connected. However, it is often too rigid, resulting in small subgraphs with trivial significance. Thus, more relaxed forms of a cohesive subgraph have been proposed and k-truss is one such proposal that has been shown to be effective [16]:

Definition 1. *A k-truss of G is a maximal connected subgraph of G where each edge participates in at least k triangles in the subgraph.*

The truss number of an edge $e \in E$ (denoted by $K(e)$) is the largest k for which there is a k-truss that contains e. The edges in the k-truss are triangle-connected to each other, which means any pair of edges e, e' in a k-truss either participates in the same triangle or connected to each other via a series of other edges $e = e_1, e_2, ..., e_k = e'$ such that each consecutive edge pair e_i, e_{i+1} (for $1 \leq i < k$) shares a triangle [32, 49]. Truss decomposition is the process of finding the truss numbers of all the edges in a given graph through a peeling process which iteratively peels the edge with the lowest triangle count. The triangle count of an edge that is being peeled is assigned as its truss number [16]. For a given graph $G = (V, E)$, the space complexity of the truss decomposition is $O(|V| + |E|)$ and the time complexity is $O(\sum_{v \in V} |N(v)|^2)$. An edge can reside in multiple k-trusses with different k values, which results in a hierarchy where lower k-trusses contain (i.e., serve as a parent of) higher k-trusses. The terminal subgraphs which do not have a child in the truss hierarchy are called the leaf trusses. Leaves have the highest edge ratio and thus represent the strongest interactions. A truss has a depth of x from a leaf if its distance to the closest leaf is x. We use the fast hierarchy construction algorithms to obtain the actual subgraphs during the truss decomposition, for which the time complexity is the same as peeling [48].

3 RELATED WORK

Here we review prior works on finding balanced and polarized subgraphs and put them in context of our work.

Controversy in Unsigned Networks. Garimella et al. quantified controversy in unsigned networks by classifying two communities which are strongly separated [26]. They proposed a random walk based method to measure the controversy, which is based on the probability that nodes will end in the same set after a random walk. **Random walk controversy (RWC)** is defined as follows:

$$\frac{(LL)(RR)}{(LL+RL)(LR+RR)} - \frac{(LR)(RL)}{(LR+RR)(LL+RL)} \tag{1}$$

where XY represents the number of walks which start in set X and end in set Y. We use the authors' code to calculate random walk controversy in polarized communities where negative edges are ignored. 50% of the nodes are randomly selected as starting nodes from each set and the walk terminates when another starting node is reached. Our work differs from Garimella et al.'s method by operating on signed networks and considering subgraphs with high cohesion across partitions.

Finding Balanced Communities. A related problem is to find a perfectly balanced subgraph for which the size of the node set is maximized. Figueiredo and Frota proposed a branch-and-cut approach [21] and introduced applications in risk management [22]. Ordozgoiti et al. proposed an algorithm which greedily removes nodes the from the graph until the graph is balanced, followed by adding back the nodes which do not impact balance [46]. The main difference between our work and these approaches is that we allow partial balance and we also target high cohesion in the subgraphs.

For partial balance, Bonchi et al. introduced a spectral algorithm, *EIGENSIGN*, which computes the first eigenvector corresponding to

the largest eigenvalue in the adjacency matrix and then discretizes its entries [9]. *EIGENSIGN* aims to find a pair of communities S that maximizes the **polarity** measure:

$$POL(S) = \frac{2 * (|E_L^+| + |E_R^+| - |E_L^-| - |E_R^-| + |E_{LR}^-| - |E_{LR}^+|)}{|V_L| + |V_R|} \quad (2)$$

where E_{LR} is the set of edges between the left and right sets. Bonchi et al. also adapted the greedy 2-approximation approach proposed by Charikar for finding the most cohesive subgraph [13], herein we refer to as *GREEDY*. It iteratively peels the node with the minimum difference between its positive and negative edge counts and the subgraph with the maximum polarity in this process is returned. **One drawback of the polarity measure is that it measures balance but not polarization**. Contrary to its name, a high value for the polarity measure does not always indicate the existence of two polarized communities. The polarity measure can be high even when one of the communities is empty — a subgraph whose edges are almost entirely positive is close to a complete agreement and there is no polarization at all. In our work, we remedy this issue by studying cohesively polarized communities. We compare our algorithms against *EIGENSIGN* and *GREEDY* in Section 6.

Detecting k Conflicting Groups. In this problem, the objective is to find k subsets of nodes which are positively connected within subsets and negatively connected between subsets. The finding polarized communities problem is simply a special case of this problem for $k = 2$. Chu et al. proposed an algorithm which aims to find all groups which contain k polarized subgraphs in signed networks [15]. However, their algorithm only finds polarized communities within each local region, which often yields subgraphs of lesser quality. A better algorithm for the same problem is proposed by Tzeng et al. [52]. Tzeng et al. proposed two spectral methods, *SCG-MA* and *SCG-R*, which both operate on the leading eigenvector of the adjacency matrix and differ in their rounding schemes [52]. In our problem formulation, prior knowledge of k is not required, unlike [52]. Nevertheless, we compare our algorithms against *SCG-MA* and *SCG-R* in Section 6.

Correlation Clustering. The objective is to partition the nodes of a signed graph into a specific number of clusters such that there are mostly positive edges within clusters and mostly negative edges across clusters [7]. The 2-correlation-clustering problem is a specific case where the number of clusters is two. Bansal et al. introduced a 3-approximation algorithm which considers pairs of clusters for all $v \in V$ such that v and all its positively connected neighbors are in one cluster and all its negatively connected neighbors are in the other cluster [7]. Inspired by Bansal et al.'s 3-approximation algorithm, Bonchi et al. [9] proposed returning the cluster pair which maximizes the polarity measure (Equation 2), which we refer to as *BANSAL*. Coleman et al. proposed an algorithm, *PASTA-TOSS*, for the 2-correlation-clustering problem which iteratively moves nodes across sets and returns the resulting distribution with the highest polarity [17]. We use *PASTA-TOSS* to partition the subgraphs returned by our algorithms into left and right communities. We compare our algorithms against *BANSAL* in Section 6.

Balanced Clique Enumeration. Recently, Sun et al. and Chen et al. studied the problem of maximal balanced k-clique enumeration in signed networks [14, 50], which aims to find maximal cliques with no unbalanced triangles. Gao et al. introduced the maximal multipolarized clique model [24] where cliques are polarized with each other. However, the definition of a clique is often too rigid, resulting in small subgraphs with trivial significance. Here we consider k-truss, a more relaxed model.

k-truss Based Models. There are a few recent works that attempt adapting the k-truss model for signed networks. Zhao et al. defined the signed k-truss as a subgraph where each edge takes part in at least $k-2$ balanced triangles and there is no unbalanced triangles [56]. Their proposed solution, which we refer to as *ZHAO*, iteratively removes the edge which participates in at least one unbalanced triangle and results in the largest subgraph until the remaining graph does not contain any unbalanced triangles. Wu et al. introduced another model, signed (k, r)-truss, where each edge is in at least k balanced triangles and at most r unbalanced triangles [54]. Both Zhao et al.'s and Wu et al.'s truss-based models find a single edge-induced subgraph where only a subset of edges among the selected subset of nodes is considered as part of the subgraph [54, 56]. We compare our algorithms against *ZHAO* in Section 6.

4 MODELING POLARIZED COMMUNITIES

In this paper, we aim to find cohesively polarized pairs of communities with non-trivial size. For cohesion, we use the *edge ratio* (see Section 2). We consider cohesion in our problem formulation to prevent weakly connected communities from being misrepresented as strongly polarized. This makes our algorithms more practical than the existing methods. We consider groups of nodes as a community, which are best modeled as vertex-induced subgraphs in which all the edges among the nodes are considered to be in the subgraph. We believe that this is more realistic than finding edge-induced subgraphs [54, 56] because it is often the set of entities, not specific connections, that one is interested in real-world applications.

Previous works on balanced subgraphs [7, 9, 52] consider balance as a direct link to polarization. However, balanced subgraphs do not have to be polarized as in the case of a subgraph with only positive edges. In this case, it is impossible to identify and limit the spread of polarization. Therefore, we consider a new approach based strictly on finding cohesively polarized communities of significant size.

Cohesively polarized communities. Polarized communities are two or more conflicting groups with positive connections in each group and negative connections in between the groups. *Polarized communities are balanced but the opposite is not true: a set of nodes with only positive edges is balanced but not polarized.*

To quantify the polarized communities, various measures have been proposed by earlier works, as explained in Section 3. The polarity measure (Equation 2) has two disadvantages in modeling polarized communities: (1) It does not care about the size of each community separately, i.e., it is perfectly okay if one of the communities does not exist or have a trivial size; (2) It combines two different objectives, agreement inside and conflict across, hence lets one dominate the other when optimizing the measure. Aref et al. proposed two measures to remedy the second issue [4]: **cohesiveness** is defined as the fraction of positive edges to the total number of edges within sets and **divisiveness** is defined as the ratio of negative edges to the total number of edges between sets. Those two measures, however, cannot address the first issue and also do not penalize the negative edges inside each community

and positive edges across the communities. A subgraph with high cohesiveness may have a lower edge ratio compared to another subgraph with similar cohesiveness. Likewise, high divisiveness is trivial if the two communities of the subgraph are separated by a few negative edges.

To address the issues mentioned above, we define **Dichotomy,** which quantifies the quality of a given pair of polarized communities S by using the polarity, cohesion, and the ratio of the community sizes, as follows:

$$POL(S) \cdot \frac{|E_L| + |E_R|}{\binom{|V_L|+|V_R|}{2}} \cdot \frac{min(|V_L|, |V_R|)}{max(|V_L|, |V_R|)} \tag{3}$$

$POL(S)$ represents the polarity (Equation 2) between the communities. Dichotomy measures the polarization of a subgraph by considering all the traits of a cohesively polarized subgraph. We want to find subgraphs which maximize the polarity and cohesion while having similar left and right set sizes so one community does not overwhelm the other. These traits each correspond to a part of the dichotomy formulation. Note that although a simple polarized 4-clique has perfect cohesion and equal left and right set sizes, it has low polarity due to its size, and thus its dichotomy is poor.

Our main problem is defined as follows:

Problem 1. *Given an undirected signed graph G, find a pair of polarized communities with optimal dichotomy.*

Bonchi et al. proved maximizing polarity to be NP-hard [9], and therefore by extension, Problem 1 is also NP-hard. We propose heuristics (Section 5) to find polarized communities with high dichotomy.

5 ALGORITHMS

We start with the *atom decomposition* (Section 5.1), which is at the core of our final algorithm, and give an early empirical evaluation to understand the structure of real-world signed networks (Section 5.1.1). Motivated by our observations, we propose *electron decomposition* for Problem 1 to find pair(s) of cohesively polarized communities (Section 5.2).

5.1 Using Triangles for Cohesion and Balance

Triangles offer a unique opportunity to capture cohesion *and* balance at the same time. The literature is rich with the methods that use triangles to model cohesive subgraphs in various kinds of networks [32, 33, 49]. Here we introduce a new subgraph definition for cohesive subgraphs with respect to a given set of triangle types in signed networks.

Definition 2. *A $(k, \triangle)$-atom of G is a maximal triangle-connected subgraph of G where each edge participates in at least k triangles of type in $\triangle$.*

$\triangle$ is the set of triangle types for which the subgraphs are to be found. If $\triangle$ has all the four signed triangle types, $(k, \triangle)$-atom is equivalent to the k-truss in the unsigned version of the network. For simplicity, we denote balanced triangles ($\triangle = \{+++, +--\}$) by *bal* and unbalanced triangles ($\triangle = \{++-, ---\}$) by *unbal*. We use $\triangle$-atom as shorthand for $(k, \triangle)$-atom when k is not relevant. We also define the **$\triangle$-atom number** of an edge as the largest k for which there is a non-empty $(k, \triangle)$-atom that contains the edge. As in the case of k-truss, all the $(k, \triangle)$-atoms in a graph form a hierarchy where subgraphs with low k values contain the subgraphs with higher k values. The largest k value for which there exists a non-empty $(k, \triangle)$-atom is the **maximum $\triangle$-atom number**. A $(k, \triangle)$-atom with the maximum atom number is the **maximum $\triangle$-atom** and a $(k, \triangle)$-atom that does not contain any $(k', \triangle)$-atom such that $k' > k$ is a **leaf $\triangle$-atom** (i.e., a leaf in the hierarchy).

Algorithm 1: ATOM $(G, \triangle)$

```
Input: G (V, E): graph, △: set of triangle types
Output: K: △-atom numbers
1  T(e) ← 0 ∀ edge e ∈ E
2  foreach triangle t ∈ G do
3      if type(t) ∈ △ then T(e)++ ∀ edge e ∈ t
4  Mark every e ∈ E as unprocessed
5  foreach unprocessed edge e with minimum T(e) do
6      K(e) ← T(e)
7      foreach triangle t ∈ G | type(t) ∈ △ ∧ e ∈ t do
8          if any edge e' ∈ t is processed then continue
9          foreach edge e' ∈ t, e' ≠ e do
10             if T(e') > T(e) then T(e')--
11     Mark e as processed
12 return K
```

To find $(k, \triangle)$-atoms in a given graph (for all k values), we introduce *atom decomposition*, *ATOM* in short, in Algorithm 1. It takes as input a signed graph G and a set of triangle types $\triangle$ and finds the $\triangle$-atom number of all the edges. The triangle count of each edge is initialized to the number of triangles of type in $\triangle$ that the edge participates in (lines 1-3). Then, a peeling process is performed to iteratively peel the edge with the lowest triangle count (of types in $\triangle$) from the graph (lines 5-11). In each iteration, the $\triangle$-atom number of the edge of interest is assigned, triangle count of the neighboring edges (with higher value) is decremented, and the edge of interest is marked as processed. At the end, $\triangle$-atom numbers of all the edges are returned (line 12). To construct the subgraphs and hierarchy, we consider the fast hierarchy construction algorithms in [48] (details are omitted for brevity). *ATOM* finds cohesive subgraphs with maximal number of given triangle types.

Time and space complexity. Unlike truss decomposition, *ATOM* only processes the triangles of a certain type, i.e., it performs truss decomposition on a smaller graph consisting of a subset of triangles from the original graph. This translates to the two additional checks in *ATOM* to ensure that the correct types of triangles are counted (line 3) and the correct types of triangles are used in the peeling (line 7). Those checks are performed in constant time and do not take additional space, thus the time and space complexities of *ATOM* are the same as truss decomposition (see Section 2).

5.1.1 Early Evaluation. We perform an early evaluation of *ATOM* to understand the structure of real-world signed networks in terms of signed triangles (see Table 3 for the datasets). To characterize the significance of the results in real-world networks, we use a null model for comparison, proposed by Kirkley et al. [34]. In this model, the structure of the graph is not changed (i.e., unsigned version stays the same) and only the signs of the edges are randomized while the ratio of negative edges is preserved (e.g., each edge in the randomized version of `Bitcoin` network is assigned a negative sign

Table 1: Maximum △-atom numbers. For each △, real denotes the real value and exp. is the average value of corresponding randomized networks. *bal* denotes {+++ , +−− }.

Real-world networks	+++		+−−		++−		−−−		*bal*	
	real	exp.	real	exp.	real	exp.	real	exp.	real	exp.
Bitcoin	9.0	7.2	8.0	2.0	3.0	3.2	2.0	1.6	10.0	7.2
Wikielections	17.0	12.1	4.0	3.0	5.0	7.1	4.0	2.6	17.0	12.1
Tw-referendum	51.0	46.3	11.0	2.1	7.0	6.0	0.0	1.2	51.0	46.3
Slashdot	34.0	19.0	7.0	4.0	3.0	12.0	3.0	3.0	34.0	19.0
Epinions	104.0	69.9	10.0	5.7	8.0	26.6	8.0	3.0	104.0	69.9
Wikipolitics	28.0	23.0	3.0	3.0	4.0	7.0	6.0	2.0	28.0	23.0
Wikiconflict	28.0	5.0	14.0	16.0	15.0	9.0	22.0	14.4	28.0	18.0

Table 2: Average proportion of the corresponding triangle type (△) within each leaf in the △-atom hierarchy. Column labels are defined as in Table 1. Tw-referendum returned no applicable subgraphs for △ = {−−− }.

Real-world networks	+++		+−−		++−		−−−		*bal*	
	real	exp.	real	exp.	real	exp.	real	exp.	real	exp.
Bitcoin	0.98	0.69	0.82	0.17	0.48	0.37	0.31	0.59	0.97	0.86
Wikielections	0.88	0.51	0.32	0.13	0.32	0.41	1.00	0.03	0.89	0.77
Tw-referendum	1.00	0.86	0.80	0.01	0.21	0.16	n/a	0.57	1.00	1.00
Slashdot	0.99	0.48	0.63	0.14	0.14	0.42	0.87	0.54	0.99	0.98
Epinions	0.99	0.59	0.50	0.08	0.26	0.36	0.49	0.01	0.98	0.96
Wikipolitics	0.95	0.71	0.18	0.15	0.22	0.31	0.83	0.18	0.96	0.94
Wikiconflict	0.71	0.06	0.58	0.44	0.62	0.28	0.84	0.26	0.73	0.68

with 15% probability). For each real-world network, we generate 10 randomized networks and report the average values.

Placement of signed triangles. Previous studies have shown that balanced triangles (+++ , +−−) are more abundant than unbalanced triangles (++− , −−−) and also more frequent than expected in real-world signed networks [5, 29, 39]. However, there has been limited research on the relative placement of these triangles in relation to each other. We now check how close the signed triangles of same (or similar) type are placed in real-world networks and in their randomized counterparts. To quantify this, we use the maximum △-atom numbers (Table 1) and the average proportion of the selected triangles in leaf atoms (Table 2).

A large maximum △-atom number implies that the edges in the corresponding subgraph participate in a larger number of triangles of type in △, thus the triangles of type △ are placed close to each other. Table 1 presents the maximum atom numbers of real networks and randomized networks (on average) for five settings. If △ has a balanced triangle (first, second, and last settings), the maximum atom number is significantly larger in the real network than in the randomized versions. On the other hand, if △ = {++− }, the real maximum atom number is smaller than the expected value (except Wikiconflict). The +−− results in Table 1 are particularly striking because even in the networks with a smaller fraction of +−− than ++− triangles (Tw-referendum, Epinions), the maximum {+−− }-atom number is larger than the maximum {++− }-atom number in real networks!

A proxy to quantify the closeness of triangles is the proportion of selected triangle types in the resulting subgraph. If the average proportion of triangles is larger than the expected, then the leaves of the real network are typically more cohesive than expected with respect to the corresponding triangle type(s). To measure that, we compute the leaf atoms with various △ settings in both real and randomized networks, compute the fraction of selected triangle types in each, and calculate the average value (e.g., if △ = {+−− }, we check the fraction of +−− triangles in the leaf {+−− }-atoms). Note that leaf atoms have the highest edge ratio when compared to its surroundings. Table 2 presents the results for five settings. +++ and +−− each (and together) yields subgraphs with a higher fraction of selected triangles in real networks than in randomized networks. Also, ++− exhibits smaller proportions than expected for four of seven networks. **We observe that not only are balanced triangles more abundant in the real networks, they are also typically closer to each other than expected, thus tend to form good seedbeds for highly balanced and/or polarized subgraphs.**

5.2 Finding Polarized Communities

△-atom is designed to maximize the number of triangles of desired type(s). However, it does not prevent the undesired type(s) of triangles from forming. In our early evaluation in Section 5.1.1, we found that +++ triangles typically dominate the triangle count and that balanced triangles overall are much more common. Therefore, if we simply use {+++ , +−− }-atoms to find polarized communities, +++ triangles may overwhelm +−− triangles. As mentioned in Section 2, all the subgraph definitions in this work are vertex-induced, implying that all the edges among the chosen set of nodes are considered to be part of the subgraph. If one is looking for balanced subgraphs, *bal*-atoms ({+++ , +−− }-atoms) only ensure that each edge is part of many balanced triangles but does not enforce anything about participations in unbalanced triangles. The most cohesive *bal*-atoms may contain many unbalanced triangles, e.g., 27% of the triangles in leaf *bal*-atoms of Wikiconflict are unbalanced (see the last pair of columns in Table 2). Although △-atom provides a simple model to find subgraphs with many triangles of interest, it is not capable to find cohesively polarized pairs of communities. Therefore, we propose to explicitly avoid unwanted triangles with a pre-processing step.

We propose a conflict-based algorithm which builds upon the $(k, \triangle)$-atom model. Since many previous algorithms often find subgraphs with mainly positive edges, poor conflict is one of the most common reasons for low dichotomy. These algorithms find subgraphs which are balanced but may not be polarized. Ideally, a polarized subgraph is balanced and contain two distinct communities with positive edges within and negative edges in between. As shown in Section 5.1.1, +++ triangles are typically much more common than +−− triangles in real-world signed networks. Hence, the balanced and cohesive subgraphs found by *bal*-atom often feature one large positively connected community and one much smaller opposing community, if at all. In order to find two polarized communities of non-trivial and comparable size, +−− triangles must be given a higher importance than +++ . {+−− }-atom (i.e., polarized atom) may seem like the obvious choice in this context since it maximizes the number of +−− triangles. However, it does not prevent the presence of other types of triangles. Existence of other types of triangles would reduce the polarization between the two communities.

To prevent the presence of undesirable triangles while keeping +−− triangles, we can filter out the nodes which are least correlated to the polarization. To this end, we first rank the nodes for removal by using the difference between the count of +−− and unbalanced

Algorithm 2: ELECTRON (G, β)

Input: G (V, E): graph, β: threshold in [-1, 1]
Output: K: processed {+−− }-atom numbers

1 $T(u) \leftarrow 0$ ∀ node $u \in V$
2 **foreach** *triangle* $t \in G$ **do**
3 **if** t *is* +−− **then** $T(u)$++ ∀ node $u \in t$
4 **else if** t is unbalanced **then** $T(u)$-- ∀ node $u \in t$
5 **while** $|V| > 0$ **do**
6 $f \leftarrow \min_{u \in V} \frac{T(u)}{\binom{|V|-1}{2}}$ // Minimum friction in the graph
7 **if** $f >= \beta$ **then** break // Threshold is reached
8 $N \leftarrow \{\text{node } u \in V \mid \frac{T(u)}{\binom{|V|-1}{2}} = f\}$ // Min. friction nodes
9 **foreach** *node* $u \in N$ **do**
10 **foreach** *triangle* $t \in G \mid u \in t$ **do**
11 **if** t *is* +−− **then** $T(v)$-- ∀ node $v \in t \mid u \neq v$
12 **else if** t is unbal. **then** $T(v)$++ ∀ node $v \in t \mid u \neq v$
13 Remove u from G
14 $K \leftarrow ATOM\ (G, \{+--\})$
15 **return** K

triangles {++− , −−− }. This difference favors balanced triangles over unbalanced triangles. Note that we specifically ignore +++ triangles to put more emphasis on the presence of +−− triangles. In addition, we want to factor in cohesion too since it is part of the dichotomy. Let $T_{ub}^{+--}(u)$ be the number of +−− triangles minus the number of unbalanced triangles containing u. The maximum number of +−− triangles a node can participate in is $\binom{|V|-1}{2}$. We define the *friction* of a node $u \in V$ to measure how important it is for the total polarization of the graph (takes values in [-1,1] interval):

$$\frac{T_{ub}^{+--}(u)}{\binom{|V|-1}{2}} \tag{4}$$

We give a tunable algorithm, *electron decomposition* (*ELECTRON* in short), that leverages the friction values of the nodes to find cohesively polarized communities with high dichotomy (Algorithm 2). *ELECTRON* takes as input a signed graph G and a threshold β, and outputs the {+−− }-atom numbers of the filtered graph. *ELECTRON* has a pre-processing stage which filters the input graph according to a specified threshold β (lines 1-13). The threshold β provides a trade-off between the subgraph size and conflict—higher β values typically result in smaller subgraphs with higher degrees of conflict. After the computation of T_{ub}^{+--} values for each node (lines 1-4), the nodes with minimum friction are iteratively removed from the graph until the friction of all the nodes satisfy the threshold β, which is in the interval [-1, 1] (lines 5-13). Note that once a node is removed, the friction of the neighboring nodes may change (lines 9-13). At the end, {+−− }-atom decomposition is performed on the remaining graph (line 14). After the {+−− }-atoms are computed, the two communities (left and right sets) in each subgraph is obtained by Coleman et al.'s *PASTA-TOSS* algorithm, which partitions the nodes into two communities with highest polarity [17]. The top subgraph from our algorithm is the maximum {+−− }-atom. If there are multiple subgraphs with the same highest k, then we choose the subgraph with the best dichotomy.

Time and space complexity. In addition to the atom decomposition (line 14), we perform a filtering process (lines 1-13) as a peeling computation over nodes. T_{ub}^{+--} values of the nodes are maintained in a bucket (instead of friction values, because all have the same denominator), which ensures picking the node(s) with minimum friction value in constant time. This is a variation of the (1,3)-nucleus decomposition, proposed in [49], where the triangle counts of the nodes are used for peeling. The time complexity of this filtering process is the same as truss decomposition and its space complexity is $O(|V|)$. Hence, the time and space complexity of electron decomposition are the same as atom decomposition.

Table 3: Signed Networks.

Real-world networks	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E^-\rvert/\lvert E\rvert$	# triangles
Bitcoin (BI)	5,881	21,492	0.15	33,493
Wikielections (WE)	7,115	100,693	0.22	607,279
Tw-referendum (TW)	10,884	251,406	0.05	3,120,811
Slashdot (SL)	82,140	500,481	0.23	579,565
Epinions (EP)	131,580	711,210	0.17	4,910,076
Wikipolitics (WP)	138,587	715,883	0.12	2,978,026
Wikiconflict (WC)	116,717	2,026,646	0.62	13,831,236

6 EXPERIMENTAL EVALUATION

We evaluate our algorithms on real-world networks against several baselines. We use various measures in evaluation.

Datasets. Important statistics of the used real-world signed networks are given in Table 3. `Bitcoin` and `Epinions` are who-trusts-whom networks of the users of Bitcoin OTC and Epinions.com, respectively [40]. `Tw-referendum` is built from Twitter data about the 2016 Italian Referendum: an interaction is negative if two users are classified with different stances, and is positive otherwise [37]. `Slashdot` contains friend/foe links between the users of Slashdot [40]. `Wikiconflict`, `Wikielections`, and `Wikipolitics` contain links between users from the English Wikipedia [36]. The edges of `Wikiconflict` represent positive and negative edit conflicts between users. `Wikielections` is the network of users that voted for and against each other in admin elections. `Wikipolitics` contains interpreted interactions between users that have edited pages about politics. We also use two large unsigned networks for scalability evaluation: `LiveJournal` and `Orkut`. We randomly assign the edge signs and perform runtime experiments (see Table 7).

Measures. For polarized communities, we consider random walk controversy (RWC) (Equation 1), polarity (Equation 2), cohesiveness, divisiveness (Section 4), and dichotomy (Equation 3)).

Baselines. We compare our algorithms against several state-of-the-art baselines: *BANSAL* [7], *EIGENSIGN* [9], *GREEDY* [9], *SCG-MA* and *SCG-R* (k=2) [52], and *ZHAO* [56] (detailed descriptions are given in Section 3). Since *SCG-MA* consistently finds subgraphs with lower balance compared to the other baselines, it is omitted in the results. We also use the truss decomposition [16], which we refer to as *TRUSS*, as a baseline (it is same as {+++ ,++− ,+−− ,−−− }-atom dec.). For all baselines, we consider the vertex-induced subgraphs that are obtained by including the endpoints of the edges outputted by the baseline.

Setup. All experiments are performed on a Linux operating system (v. Linux 3.10.0-1127) running on a machine with Intel(R) Xeon(R) Gold 6130 CPU processor at 2.10 GHz with 192 GB memory. We implemented our algorithms in C++ and compiled using gcc 6.3.0 at the -O2 level. **Code is available at https://tinyurl.com/polarizedSubgraphs.**

Table 4: Results for the polarized communities obtained by our algorithms and baselines. $|V_L|$ and $|V_R|$ denote the number of nodes in left and right sets (larger is the left set w.l.o.g), RWC is random walk controversy, Pol. is polarity, Coh. is cohesiveness, Div. is divisiveness, and Dic. is dichotomy. For the baselines (*BANSAL*, *EIGENSIGN*, *GREEDY*, *SCG-R*, *ZHAO*, *TRUSS*), we show the one that gives the highest dichotomy and denote it with a trailing asterisk. *P-ATOM* denotes the {+−−}-atom decomposition (i.e., polarized atom decomposition).

Networks	Algorithms	$\|V_L\|$	$\|V_R\|$	RWC	Pol.	Coh.	Div.	Dic.
Bitcoin	*BANSAL**	21	20	1.00	23.07	0.98	1.00	12.78
	B-ATOM	17	6	1.00	16.70	0.99	1.00	4.52
	P-ATOM	9	8	1.00	14.00	0.98	1.00	11.07
	ELECTRON	19	17	1.00	23.00	1.00	1.00	13.52
Wikielections	*BANSAL**	408	7	0.83	39.49	0.88	0.89	0.08
	B-ATOM	65	0	-	39.08	-	-	0.00
	P-ATOM	155	3	-	3.00	0.53	1.00	0.01
	ELECTRON	6	4	1.00	7.00	1.00	1.00	3.63
Tw-referendum	*BANSAL**	543	100	0.32	52.21	1.00	0.61	0.92
	B-ATOM	128	0	-	99.00	-	-	0.00
	P-ATOM	15	14	1.00	23.66	1.00	1.00	18.66
	ELECTRON	15	14	1.00	23.66	1.00	1.00	18.66
Slashdot	*SCG-R**	26	6	0.64	7.12	1.00	0.92	0.38
	B-ATOM	77	0	-	60.36	-	-	0.00
	P-ATOM	49	7	1.00	7.00	0.55	1.00	0.56
	ELECTRON	30	6	1.00	13.00	0.67	1.00	1.81
Epinions	*BANSAL**	249	5	1.00	157.76	1.00	1.00	1.98
	B-ATOM	142	1	-	135.83	1.00	1.00	0.92
	P-ATOM	134	15	1.00	1.15	0.46	1.00	0.05
	ELECTRON	28	6	1.00	5.29	0.50	1.00	0.80
Wikipolitics	*BANSAL**	599	11	0.58	48.45	0.98	0.41	0.07
	B-ATOM	86	0	-	58.72	-	-	0.00
	P-ATOM	10	3	-	9.23	1.00	1.00	2.13
	ELECTRON	9	2	-	8.91	1.00	1.00	1.76
Wikiconflict	*BANSAL**	369	134	0.91	131.07	0.91	0.96	14.19
	B-ATOM	147	0	-	69.05	-	-	0.00
	P-ATOM	33	31	1.00	40.72	0.90	1.00	26.09
	ELECTRON	40	36	1.00	46.08	0.98	1.00	25.68

6.1 Polarized Communities

Table 4 shows the results for the pairs of polarized communities obtained by {+−− }-atom decomposition (denoted by *P-ATOM*), *ELECTRON* ($\beta = 0.1$), and the best baseline algorithm in terms of dichotomy. For each graph and algorithm, we list the number of nodes in the left and right communities (larger one is the left w.l.o.g.), random walk controversy, polarity, cohesiveness, divisiveness, and dichotomy. Trivial subgraphs with empty right sets do not have a cohesiveness and divisiveness score.

P-ATOM and *ELECTRON* consistently have the best dichotomy scores, often outperforming all other algorithms by several factors. Note that random walk controversy and polarity scores are not in line with the other measures in several cases, which suggests that they are not reliable measures. Since real-world networks typically consist of multiple smaller polarized communities instead of a single large community, finding multiple high quality subgraphs is often desirable instead of misclassifying a larger subgraph.

6.2 Finding Multiple Communities

Our algorithms can find multiple pairs of polarized communities along with the maximal pair, which are ranked by their atom numbers. Each of these non-maximal subgraphs with high dichotomy can tell a unique story about a polarization within the signed network. For the baselines, we can find multiple sets of polarized communities by removing the resulting communities from the graph and reapplying the algorithm on the residual graph. We repeat this process to obtain at most 10 pairs of polarized communities.

Table 5: Average results for the polarized communities obtained by our algorithms and baselines. NP is the number of pairs of polarized communities and the rest is defined as in Table 4.

Networks	Algorithms	NP	$\|V_L\|$	$\|V_R\|$	Pol.	Coh.	Div.	Dic.
Bitcoin	*BANSAL**	10	53.8	5.1	10.76	0.93	0.77	1.79
	P-ATOM	9	18.7	12.4	11.71	0.80	1.00	6.14
	ELECTRON	2	20.0	17.0	23.50	1.00	1.00	13.06
Wikielections	*BANSAL**	10	197.3	13.1	25.00	0.92	0.55	0.33
	P-ATOM	2	431.5	15.0	10.87	0.59	1.00	0.04
	ELECTRON	2	8.0	4.0	8.07	1.00	1.00	3.10
Tw-referendum	*BANSAL**	9	294.1	15.0	54.49	1.00	0.69	0.25
	P-ATOM	3	34.3	25.0	36.17	1.00	1.00	17.69
	ELECTRON	3	29.7	18.7	34.26	1.00	1.00	16.77
Slashdot	*EIGENSIGN**	4	191.0	80.2	38.85	0.96	0.95	0.97
	P-ATOM	7	42.3	5.9	5.76	0.75	1.00	0.37
	ELECTRON	2	34.0	6.0	15.22	0.70	1.00	1.83
Epinions	*EIGENSIGN**	3	199.3	4.0	94.69	1.00	0.99	0.46
	P-ATOM	11	70.2	9.1	5.69	0.83	1.00	0.79
	ELECTRON	3	33.0	6.7	7.73	0.54	1.00	1.08
Wikipolitics	*BANSAL**	10	159.1	1.5	40.85	0.99	0.61	0.04
	P-ATOM	10	51.4	8.1	6.54	0.94	1.00	0.44
	ELECTRON	1	9.0	2.0	8.91	1.00	1.00	1.76
Wikiconflict	*EIGENSIGN**	7	91.6	51.7	46.65	0.86	1.00	9.91
	P-ATOM	154	17.6	3.1	3.21	0.51	1.00	0.56
	ELECTRON	2	43.0	36.0	47.98	0.98	1.00	24.94

Table 5 shows the average results for the pairs of polarized communities obtained by {+−− }-atom decomposition (*P-ATOM*), *ELECTRON* ($\beta = 0.1$), and the best baseline algorithm in terms of dichotomy. For each graph and algorithm, we list the number of obtained community pairs, average number of nodes in the left and right sets (larger set is the left w.l.o.g.), and average polarity, cohesiveness, divisiveness, and dichotomy. Trivial subgraphs with empty right sets are omitted. For our algorithms, only the subgraphs within a depth level of 2 from a leaf are considered.

ELECTRON has the best dichotomy score in 6 of 7 networks, often outperforming the second-best algorithm by a significant margin. `Tw-referendum` is the only dataset where *P-ATOM* slightly outperforms *ELECTRON*. *ELECTRON* is often able to find two communities with non-trivial sizes, thanks to its focus on +−− triangles.

6.3 *ELECTRON* Threshold Experiments

We compare the results of *ELECTRON* for different values of β. Figure 1 shows the number of subgraph nodes and dichotomy when β is 0.1, 0.2, and 0.3. *ELECTRON* avoids finding subgraphs which are harmonious as a whole but contain a small conflicting group hidden within, unlike other algorithms. In general, β value

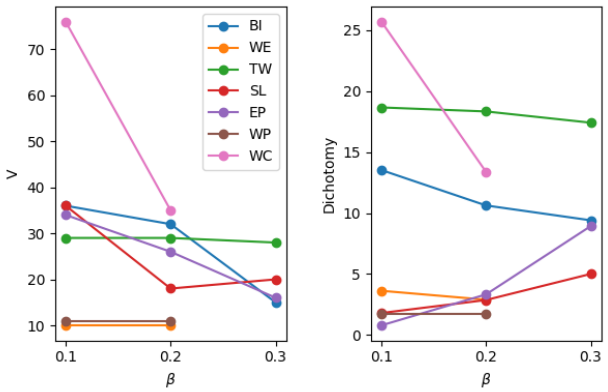

Figure 1: *ELECTRON*'s threshold experiments. For β values of 0.1, 0.2, and 0.3, we plot the number of nodes (left) and dichotomy (right).

Table 6: Average results for the subgraphs obtained by our algorithms and baselines on CoW. NP is the number of pairs of polarized communities and the rest is defined as in Table 4. We only show algorithms which have the best result for Pol., Coh., Div., and/or Dic. (highlighted in blue).

Algorithm	NP	$\|V_L\|$	$\|V_R\|$	Pol.	Coh.	Div.	Dic.
ZHAO	8	24.12	1.62	13.84	0.60	1.00	0.05
TRUSS	8	27.5	0.12	21.35	0.99	1.00	0.11
P-ATOM	3	17.0	4.67	6.64	0.76	1.00	3.50
ELECTRON	2	10.5	6.0	11.18	0.97	1.00	4.99

of 0.1 gives subgraphs with high dichotomy for the majority of the networks, hence suggested as default. One can choose slightly higher β values for higher degrees of conflict at the expense of smaller community sizes, where dichotomy represents the trade-off.

6.4 Case Study: Correlates of War

In this section, we present a case study to evaluate the algorithms on the CoW (Correlates of War) dataset [18]. Nodes in CoW are the countries, negative edges indicate a major conflict such as war, and positive edges represent alliances or peace treaties. The original data has 52 signed networks corresponding to different time periods between 1946 and 1999. Here we aggregate the data by choosing the most common sign for each edge (or the most recent, if tied). In total, there are 180 nodes, 397 negative edges, 1406 positive edges, and 12249 triangles. USA has the highest positive degree and RUS has the highest negative degree.

Table 6 shows the average results for all obtained communities (as described in Section 6.2). We omit the algorithms who do not perform best in any of the measures. *P-ATOM* and *ELECTRON* are the only algorithms that are able to find a subgraph with significant dichotomy. Almost all of the baselines (except *SCG-R*) and *B-ATOM* find the same subgraph consisting of 33 countries strongly aligned with the USA — there are 554 positive and only 7 negative edges. The top subgraph found by *ELECTRON* (the maximum $\{+--\}$-atom of the filtered graph) is shown in Figure 2. One set has the countries in alliance with USA and the other set contains the countries who had a positive relationship with RUS, which is expected since the time range mapped by CoW includes the cold war period. *P-ATOM* finds a smaller version of this subgraph. **None of the other baselines can find a similar subgraph that depicts the cold war**.

Although a large subgraph may have high balance, it may not have high dichotomy. Subgraphs with low cohesion may represent weaker alliances and/or conflicts where some countries may be neutral (not friendly nor hostile) with each other. Subgraphs with high cohesion may represent strong alliances between countries who may not be directly involved in the cold war. Subgraphs with high dichotomy consists of two strong alliances in a significant conflict with each other. **Hence, capturing the polarization in the form of dichotomy is the key to measure the significance of the relationships within a subgraph.**

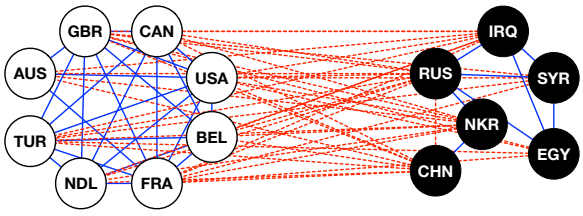

Figure 2: The top subgraph found by *ELECTRON*. White and black nodes denote the left and right communities, respectively. Positive edges are denoted by straight blue lines and negative ones are shown by dashed red lines.

Table 7: Runtime results (in seconds). We denote the computations that exceed 12 hours by -.

	Wikiconflict	LiveJournal			Orkut		
	$\|V\|$=117K	$\|V\|$=4M, $\|E\|$=35M			$\|V\|$=3M, $\|E\|$=109M		
	$\|E\|$=2M	$\|E^-\|/\|E\|$			$\|E^-\|/\|E\|$		
Algorithm		0.05	0.25	0.5	0.05	0.25	0.5
BANSAL	11.5K	-	-	-	-	-	-
EIGENSIGN	344	10.2K	3.2K	607	8.3K	4.3K	5.0K
GREEDY	7.7K	-	-	-	-	-	-
SCG-R	2.8K	6.9K	2.4K	2.6K	2.7K	2.8K	3.4K
ZHAO	496	4.6K	1.9K	2.6K	20.2K	19.9K	21.7K
TRUSS	20	106	105	123	668	671	671
B-ATOM	38	352	315	316	2.0K	2.3K	1.9K
ELECTRON	47	681	632	722	1.6K	2.0K	2.3K

6.5 Runtime Performance

We compare the runtimes of our algorithms and the baselines on the largest network in our dataset, Wikiconflict, and also two large unsigned networks: LiveJournal and Orkut in Table 7. We randomly assign the edge signs in LiveJournal and Orkut, and used three different probabilities for negative edges, 0.05, 0.25, and 0.5, to see if the fraction of signs impact the results. We generate 10 random networks for each configuration and considered the average runtimes. We terminate the computations after 12 hours. *TRUSS*, *B-ATOM*, and *ELECTRON* find multiple polarized communities through their peeling process. For baselines, we find multiple sets of polarized communities as explained in Section 6.2.

For Wikiconflict, our algorithms are faster than the baselines, with the exception of *TRUSS*. All the proposed algorithms finish in about a minute or less. *TRUSS* takes less time because it simply considers all the triangles without any specific checks for edge signs, which makes it ineffective in finding polarized communities. For LiveJournal and Orkut, we observe similar trends. Our algorithms are consistently faster than the baselines, taking a few minutes for LiveJournal and less than an hour for Orkut. No significant difference is observed for different ratios of negative edges.

7 CONCLUSION

We studied the problem of finding polarized communities. We leveraged cohesion to define the dichotomy measure which can better model the polarized communities in real-world signed networks. We then proposed a heuristic, *ELECTRON*, that successfully takes advantage of the signed triangles. Experimental evaluation on real-world networks showed that our algorithms typically yield more polarized and cohesive communities when compared to the state-of-the-art methods. We believe that our algorithms will be beneficial in real-world applications that are engaged with signed networks. For example, cohesively polarized subgraphs in online discussion platforms can point to the set of users that are heavily interested in a topic with strong opinions.

ACKNOWLEDGMENTS

This research was supported by NSF award OAC-2107089 and used resources from the Center for Computational Research at the University at Buffalo [1].

REFERENCES

[1] 2021. Center for Computational Research, University at Buffalo. http://hdl.handle.net/10477/79221.

[2] J. Ignacio Alvarez-Hamelin, Alain Barrat, and Alessandro Vespignani. 2006. Large scale networks fingerprinting and visualization using the k-core decomposition. In *NIPS*. 41–50.

[3] Albert Angel, Nick Koudas, Nikos Sarkas, and Divesh Srivastava. 2012. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *arXiv preprint arXiv:1203.0060* (2012).

[4] Samin Aref, Ly Dinh, Rezvaneh Rezapour, and Jana Diesner. 2020. Multilevel structural evaluation of signed directed social networks based on balance theory. *Scientific Reports* 10, 1 (2020), 1–12.

[5] Samin Aref and Mark C Wilson. 2018. Measuring partial balance in signed networks. *Journal of Complex Networks* 6, 4 (2018), 566–595.

[6] Delia Baldassarri and Peter Bearman. 2007. Dynamics of political polarization. *American Sociological Review* 72, 5 (2007), 784–811.

[7] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning* 56, 1 (2004), 89–113.

[8] Phillip Bonacich and Philip Lu. 2012. *Introduction to mathematical sociology*. Princeton University Press.

[9] Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. 2019. Discovering polarized communities in signed networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 961–970.

[10] Jennifer Brundidge. 2010. Encountering "difference" in the contemporary public sphere: The contribution of the Internet to the heterogeneity of political discussion networks. *Journal of Communication* 60, 4 (2010), 680–700.

[11] Jose Cadena, Anil Kumar Vullikanti, and Charu C Aggarwal. 2016. On dense subgraphs in signed network streams. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 51–60.

[12] Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider's theory. *Psychological review* 63, 5 (1956), 277.

[13] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 84–95.

[14] Zi Chen, Long Yuan, Xuemin Lin, Lu Qin, and Jianye Yang. 2020. Efficient Maximal Balanced Clique Enumeration in Signed Networks. In *Proceedings of The Web Conference 2020*. 339–349.

[15] Lingyang Chu, Zhefeng Wang, Jian Pei, Jiannan Wang, Zijin Zhao, and Enhong Chen. 2016. Finding gangs in war from signed networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1505–1514.

[16] Jonathan Cohen. 2008. Trusses: Cohesive subgraphs for social network analysis. *National Security Agency Technical Report* 16, 3.1 (2008).

[17] Tom Coleman, James Saunderson, and Anthony Wirth. 2008. A local-search 2-approximation for 2-correlation-clustering. In *European Symposium on Algorithms*. Springer, 308–319.

[18] Patrick Doreian and Andrej Mrvar. 2019. Structural balance and signed international relations. *Journal of Social Structure* 16, 1 (2019).

[19] Joan-Maria Esteban and Debraj Ray. 1994. On the measurement of polarization. *Econometrica: Journal of the Econometric Society* (1994), 819–851.

[20] Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. 2019. Efficient algorithms for densest subgraph discovery. *PVLDB* 12, 11 (2019), 1719–1732.

[21] Rosa Figueiredo and Yuri Frota. 2013. An improved Branch-and-cut code for the maximum balanced subgraph of a signed graph. *arXiv preprint arXiv:1312.4345* (2013).

[22] Rosa Figueiredo and Yuri Frota. 2014. The maximum balanced subgraph of a signed graph: Applications and solution approaches. *European Journal of Operational Research* 236, 2 (2014), 473–487.

[23] Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. 2006. MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics* 22, 14 (2006), e150–e157.

[24] Jie Gao, Fei Hao, Geyong Min, and Zhipeng Cai. 2021. Maximal Multipolarized Cliques Search in Signed Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2227–2231.

[25] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2017. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 81–90.

[26] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Quantifying controversy on social media. *ACM Transactions on Social Computing* 1, 1 (2018), 1–27.

[27] David Gibson, Ravi Kumar, and Andrew Tomkins. 2005. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st international conference on Very large data bases*. Citeseer, 721–732.

[28] Aristides Gionis and Charalampos E. Tsourakakis. 2015. Dense Subgraph Discovery: Tutorial. In *KDD*. 2313–2314.

[29] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*. 403–412.

[30] Frank Harary. 1953. On the notion of balance of a signed graph. *Michigan Mathematical Journal* 2, 2 (1953), 143–146.

[31] Fritz Heider. 1946. Attitudes and cognitive organization. *The Journal of Psychology* 21, 1 (1946), 107–112.

[32] Xin Huang, Hong Cheng, Lu Qin, Wentao Tian, and Jeffrey Xu Yu. 2014. Querying K-truss Community in Large and Dynamic Graphs. In *SIGMOD*. 1311–1322.

[33] Xin Huang, Wei Lu, and Laks V.S. Lakshmanan. 2016. Truss Decomposition of Probabilistic Graphs: Semantics and Algorithms. In *SIGMOD*. 77–90.

[34] Alec Kirkley, George T Cantwell, and MEJ Newman. 2019. Balance in signed networks. *Physical Review E* 99, 1 (2019), 012320.

[35] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2018. Community interaction and conflict on the web. In *Proceedings of the 2018 world wide web conference*. 933–943.

[36] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection *(WWW Companion)*. 1343–1350.

[37] Mirko Lai, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. 2018. Stance evolution and twitter interactions in an italian political debate. In *International Conference on Applications of Natural Language to Information Systems*. Springer, 15–27.

[38] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. 2010. A Survey of Algorithms for Dense Subgraph Discovery. In *Managing and Mining Graph Data*. Vol. 40.

[39] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1361–1370.

[40] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[41] Ronghua Li, Qiangqiang Dai, Lu Qin, Guoren Wang, Xiaokui Xiao, Jeffrey Xu Yu, and Shaojie Qiao. 2019. Signed Clique Search in Signed Networks: Concepts and Algorithms. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[42] Q. Vera Liao and Wai-Tat Fu. 2014. Can You Hear Me Now? Mitigating the Echo Chamber Effect by Source Position Indicators. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) *(CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 184–196. https://doi.org/10.1145/2531602.2531711

[43] Q Vera Liao and Wai-Tat Fu. 2014. Expert voices in echo chambers: effects of source expertise indicators on exposure to diverse opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2745–2754.

[44] Yelena Mejova, Amy X Zhang, Nicholas Diakopoulos, and Carlos Castillo. 2014. Controversy and sentiment in online news. *arXiv preprint arXiv:1409.8152* (2014).

[45] Alfredo Jose Morales, Javier Borondo, Juan Carlos Losada, and Rosa M Benito. 2015. Measuring political polarization: Twitter shows the two sides of Venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25, 3 (2015), 033114.

[46] Bruno Ordozgoiti, Antonis Matakos, and Aristides Gionis. 2020. Finding large balanced subgraphs in signed networks. In *Proceedings of The Web Conference 2020*. 1378–1388.

[47] Ahmet Erdem Sariyuce. 2021. Motif-driven Dense Subgraph Discovery in Directed and Labeled Networks. In *Proceedings of the Web Conference 2021*. 379–390.

[48] Ahmet Erdem Sariyuce and Ali Pinar. 2016. Fast Hierarchy Construction for Dense Subgraphs. *Proceedings of the VLDB Endowment* 10, 3 (2016).

[49] A. Erdem Sarıyüce, C. Seshadhri, A. Pınar, and Ü. V. Çatalyürek. 2015. Finding the Hierarchy of Dense Subgraphs Using Nucleus Decompositions. In *WWW* (Florence, Italy). 927–937.

[50] Renjie Sun, Qiuyu Zhu, Chen Chen, Xiaoyang Wang, Ying Zhang, and Xun Wang. 2020. Discovering cliques in signed networks based on balance theory. In *International Conference on Database Systems for Advanced Applications*. 666–674.

[51] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli. 2013. Denser Than the Densest Subgraph: Extracting Optimal Quasi-cliques with Quality Guarantees *(KDD '13)*.

[52] Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. 2020. Discovering conflicting groups in signed networks. *Advances in Neural Information Processing Systems* 33 (2020).

[53] VG Vinod Vydiswaran, ChengXiang Zhai, Dan Roth, and Peter Pirolli. 2015. Overcoming bias to learn about controversial topics. *Journal of the Association for Information Science and Technology* 66, 8 (2015), 1655–1672.

[54] Yanping Wu, Renjie Sun, Chen Chen, Xiaoyang Wang, and Qiuyu Zhu. 2020. Maximum Signed (k, r)-Truss Identification in Signed Networks. In *Proc. of the 29th ACM International Conf. on Information & Knowledge Management*. 3337–3340.

[55] Han Xiao, Bruno Ordozgoiti, and Aristides Gionis. 2020. Searching for polarization in signed graphs: a local spectral approach. In *Proceedings of The Web Conference 2020*. 362–372.

[56] Jun Zhao, Renjie Sun, Qiuyu Zhu, Xiaoyang Wang, and Chen Chen. 2020. Community Identification in Signed Networks: A K-Truss Based Model. In *Proc. of the 29th ACM International Conf. on Information & Knowledge Management*. 2321–2324.