Discovering Polarized Communities in Signed Networks

Francesco Bonchi
francesco.bonchi@isi.it
ISI Foundation, Italy
Eurecat, Barcelona, Spain

Edoardo Galimberti
edoardo.galimberti@unito.it
Dept. of Computer Science., University of Turin, Italy
ISI Foundation, Italy

Aristides Gionis, Bruno Ordozgoiti
{name.surname}@aalto.fi
Dept. of Computer Science, Aalto University, Finland

Giancarlo Ruffo
giancarlo.ruffo@unito.it
Dept. of Computer Science., University of Turin, Italy

ABSTRACT

Signed networks contain edge annotations to indicate whether each interaction is friendly (positive edge) or antagonistic (negative edge). The model is simple but powerful and it can capture novel and interesting structural properties of real-world phenomena. The analysis of signed networks has many applications from modeling discussions in social media, to mining user reviews, and to recommending products in e-commerce sites.

In this paper we consider the problem of discovering polarized communities in signed networks. In particular, we search for two communities (subsets of the network vertices) where within communities there are mostly positive edges while across communities there are mostly negative edges. We formulate this novel problem as a "discrete eigenvector" problem, which we show to be **NP**-hard. We then develop two intuitive spectral algorithms: one deterministic, and one randomized with quality guarantee $\sqrt{n}$ (where n is the number of vertices in the graph), tight up to constant factors.

We validate our algorithms against non-trivial baselines on real-world signed networks. Our experiments confirm that our algorithms produce higher quality solutions, are much faster and can scale to much larger networks than the baselines, and are able to detect ground-truth polarized communities.

ACM Reference Format:
Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. 2019. Discovering Polarized Communities in Signed Networks. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357977

1 INTRODUCTION

The increase of polarization around controversial issues is a growing concern with important societal fallouts. While controversy can be engaging, and can lead to users spending more time on social-media platforms, in disproportionate amounts it can generate a negative user experience, potentially leading to the abandonment of the platform. Excessive polarization, together with the emergence of bots and the spread of misinformation, has thus become an urgent technological problem that needs to be solved. It is not surprising that the last few years have witnessed an uptake of the research on methods for the detection and suppression of these phenomena [21, 25, 36, 37, 43, 54].

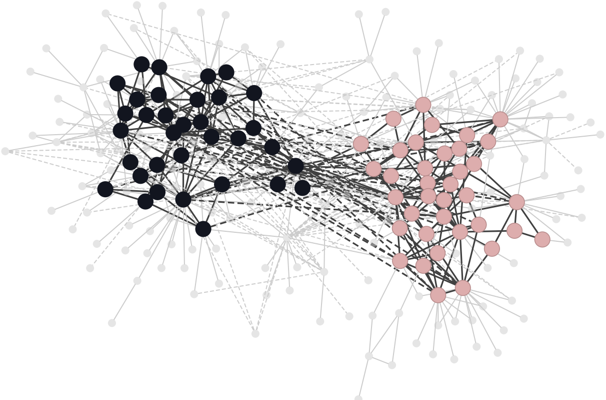

Figure 1: An example of two polarized communities in the Congress network (dataset details in Section 5). Solid edges are positive, while dashed edges are negative.

While polarization is a well studied phenomenon in political and social sciences [5, 8, 18, 19, 23, 55], modern social-media platforms brought it to a different scale, providing an unprecedented wealth of data. The necessity to analyze the available data and gain valuable insights brings new algorithmic challenges.

In order to study polarization in large-scale online data, one first step is to *detect* it. As a step in this direction, in this paper we study a fundamental problem abstraction for this task, i.e., the problem of *discovering polarized communities in signed networks.*

A signed network is a simple, yet general and powerful, representation: vertices represent entities and edges between vertices represent interactions, which can be friendly (positive) or antagonistic (negative) [27]. Signed graphs analysis has many applications from modeling interactions in social media [30], to mining user reviews [7], to studying information diffusion and epidemics [35], to recommending products in e-commerce sites [39, 53], and to estimating the structural balance of a (physical) complex system [4, 41].

In this paper, we introduce the 2-Polarized-Communities problem (2PC), which requires finding two communities (subsets of the network vertices) such that within communities there are mostly positive edges while across communities there are mostly negative edges. Furthermore, we do not aim to partition the whole network,

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357977

so the two polarized communities we are searching can be concealed within a large body of other network vertices, which are neutral with respect to the polarized structure. Our hypothesis is that such 2-community polarized structure accurately captures controversial discussions in real-world social-media platforms.

Figure 1 shows an example of the two most polarized communities found in the Congress network (details in Section 5). The two communities involve 34 and 37 vertices (out of 219), respectively, having more than 98% of positive edges within and 78% negative edges across. The vertices in gray do not participate in any of the two polarized communities: either they have too few connections with any community, or the polarity of their relations are mixed and thus their position within the debate unclear.

Our work is, to the best of our knowledge, the first to propose a spectral method for extracting polarized communities from signed networks. In addition, we present hardness results and approximation guarantees. Our problem formulation deviates from the bulk of the literature where methods typically look for finding many communities while partitioning the whole network [3, 6, 12, 15, 24, 31]. As discussed in more detail in Sections 2 and 3, the closest to our problem statement is the work by Coleman et al. [15], who employ the correlation-clustering framework and search for exactly two communities. However, while in that work all vertices must be included in a cluster, in our setting we allow vertices not to be part of any cluster. This captures the fact that polarized communities are typically concealed within a large body of neutral vertices in a social network. An algorithm that attempts to partition the whole network would fail to reveal these communities. As an additional feature, our methods can be fine-tuned to increase or decrease the size of the discovered communities. Finally, while some spectral techniques promote balanced partitions, we hypothesize that two polarized communities might be of very different sizes, and thus our problem formulation does not enforce evenly sized subgraphs.

Our reliance on spectral methods carries several benefits. First, it is possible to leverage readily available, highly optimized, and parallelized software implementations. This makes it straightforward for the practitioner to analyze large networks in real settings using our approach. Second, even though in this paper we focus on the case of two communities, we can take inspiration from the existing literature on spectral graph partitioning to easily extend our algorithms to the case of an arbitrary number of subgraphs, e.g., by recursive two-way partitioning or the analysis of multiple eigenvectors [47].

In this paper we make the following contributions:

- We formulate the 2-Polarized-Communities problem (2PC) as a "discrete eigenvector" problem (Section 3).
- Exploiting a reduction from classic correlation clustering, we prove that 2PC is **NP**-hard (Theorem 3.3).
- We devise two intuitive spectral algorithms (Section 4), one deterministic, and one randomized with quality guarantee $\sqrt{n}$ (Theorem 4.2), which is tight up to constant factors. We believe these to be the first purely combinatorial bounds for spectral methods. Our results apply to graphs of arbitrary weights. Our algorithms' running time is essentially the time required to compute the first eigenvector of the adjacency matrix of the input graph.
- Our experiments (Section 5) on a large collection of real-world signed networks show that the proposed algorithms discover higher quality solutions, are much faster than the baselines, and can scale to much larger networks. In addition, they are able to identify ground-truth planted polarized communities in synthetic datasets.

Related literature is discussed in the next section, while Section 6 discusses future work and concludes the paper.

2 BACKGROUND AND RELATED WORK

Signed networks. Signed graphs appeared in a work by Harary, who was interested in the notion of *balance* in graphs [27]. In 1956, Cartwright and Harary generalized Heider's psychological theory of balance in triangles of sentiments to the theory of balance in signed graphs [9].

A more recent line of work develops the spectral properties of signed graphs, still related to balance theory. Hou et al. [28] prove that a connected signed graph is balanced if and only if the smallest eigenvalue of the Laplacian is 0. Hou [29] also investigates the relationship between the smallest eigenvalue of the Laplacian and the unbalancedness of a signed graph.

Signed graphs have also been studied in different contexts. Guha et al. [26] and Leskovec et al. [34] study *directed* signed graphs and develop *status theory*, to reason about the importance of the vertices in such graphs. Other lines of research include edge and vertex classification [10, 50], link prediction [33, 49], community detection [1, 3, 6, 48], recommendation [51], and more. A detailed survey on the topic is provided by Tang et al. [52].

A few recent works explore the problem of finding antagonistic communities in signed networks, though with approaches fundamentally different to ours. Lo et al. consider directed graphs and search for strongly-connected positive subgraphs that are negative bi-cliques [38], which severely limits the size of the resulting communities. A relaxed variant for undirected networks was described in subsequent work [20]. Chu et al. propose a constrained-programming objective to find k warring factions [14], as well as an efficient algorithm to find local optima.

Correlation Clustering. In the standard correlation-clustering problem [6], we ask to partition the vertices of a signed graph into clusters so as to maximize (minimize) the number of edges that "agree" ("disagree") with the partitioning, i.e., the number of positive (negative) edges within clusters plus the number of negative (positive) edges across clusters. In the original problem formulations, such as the ones studied by Bansal et al. [6], Swamy [48], and Ailon et al. [1], the number of clusters is not given as input, instead it is part of the optimization. More recent works study the correlation-clustering problem with additional constraints, e.g., Giotis and Guruswami [24] fix the number of clusters, Coleman et al. [15] consider only two clusters, while Puleo and Milenkovic [45] consider constraints on the cluster sizes.

The problem we study could be seen as a variant of correlation clustering where we search for two clusters, while we allow vertices not to be part of any cluster.

Detecting polarization in social media. A number of papers have studied the problem of detecting polarization in social media Some approaches are based on text analysis [13, 42, 44], while other approaches consider a graph-theoretic setting [2, 16, 22]. However, our work differs significantly from these papers, as we consider signed networks, we provide a correlation-clustering problem formulation, and obtain results with approximation guarantees.

3 PROBLEM STATEMENT

Our setting is reminiscent to the *correlation-clustering* problem [6], which we recall here. Given a signed network $G = (V, E_+, E_-)$, where E_+ is the set of positive edges and E_- the set of negative edges, the goal is to find a partition of the vertices into k clusters, so as to maximize the number of positive edges within clusters plus the number of negative edges between clusters.

An interesting property of the correlation-clustering formulation is that one does not need to specify in advance the number of clusters k, instead it is part of the optimization. In certain cases, however, the number of clusters is given as input. The general problem (given k) has been studied by Giotis and Guruswami [24], while Coleman et al. [15] studied the 2-Correlation-Clustering problem ($k = 2$). The problem arises, for instance, in the domain of social networks, where two well-separated clusters reveal a polarized structure. It can be defined as follows.

Problem 1 (2CC). *Given a signed network $G = (V, E_+, E_-)$, find a partition S_1, S_2 of V so as to maximize*

$$cc(S_1, S_2) = \sum_{\substack{i\in\{1,2\}\\(u,v)\in S_i\times S_i}} \frac{1}{2}\mathbf{1}_{E_+}(u,v) + \sum_{(u,v)\in S_1\times S_2} \mathbf{1}_{E_-}(u,v),$$

where $\mathbf{1}_S$ is the indicator function of the set S.

A crucial limitation of the 2CC problem is that all vertices must be accounted for in one of the two clusters. From an application perspective, however, this may be a strong assumption. For example, in a social network, we may expect two polarized communities on a topic, but there may be many individuals who are neutral.

In order to find communities embedded within large networks, we need to exclude neutral vertices from the solution. Therefore, a first approach might be to consider maximizing agreements including a neutral cluster, that is, finding a partition of V into S_1, S_2, and S_0, so that S_1 and S_2 are the two polarized communities, and S_0 is the neutral community, and the 2CC objective $cc(S_1, S_2)$ is maximized. However, this modification does not change the problem significantly. It is easy to see that it is always no worse to switch a vertex from cluster S_0 to one of the other two clusters.

Proposition 3.1. *Let S_0, S_1, S_2 be any partition of V, with $S_0 \neq \emptyset$. Then there is always a partition S_1', S_2' of V (i.e., $S_1' \cup S_2' = V$ and $S_1' \cap S_2' = \emptyset$) with $S_1 \subseteq S_1'$ and $S_2 \subseteq S_2'$ so that*

$$cc(S_1', S_2') \geq cc(S_1, S_2).$$

A further modification might be to subtract disagreements from the value of the solution, that is, to maximize agreements minus disagreements. In other words, we consider the following problem.

Problem 2 (2CC-Full). *Given a signed network $G = (V, E_+, E_-)$, find a partition S_0, S_1, S_2 of V so as to maximize*

$$\begin{aligned}\overline{cc}(S_1, S_2) \quad = \quad & \sum_{\substack{i\in\{1,2\}\\(u,v)\in S_i\times S_i}} \frac{1}{2}\left(\mathbf{1}_{E_+}(u,v) - \mathbf{1}_{E_-}(u,v)\right) \\ & + \sum_{(u,v)\in S_1\times S_2} \left(\mathbf{1}_{E_-}(u,v) - \mathbf{1}_{E_+}(u,v)\right),\end{aligned}$$

where $\mathbf{1}_S$ is the indicator function of the set S.

Unfortunately, problem 2CC-Full suffers from the same issue as problem 2CC: switching a vertex from the neutral cluster S_0 to one of the polarized clusters S_1 or S_2 (the one that is best) leads to no worse solution according to the objective $\overline{cc}$.

Proposition 3.2. *Let S_0, S_1, S_2 be any partition of V, with $S_0 \neq \emptyset$. Then there is always a partition S_1', S_2' of V (i.e., $S_1' \cup S_2' = V$ and $S_1' \cap S_2' = \emptyset$) with $S_1 \subseteq S_1'$ and $S_2 \subseteq S_2'$ so that*

$$\overline{cc}(S_1', S_2') \geq \overline{cc}(S_1, S_2).$$

A nice property of the $\overline{cc}$ objective is that it can be written neatly in a matrix notation. Let A be the adjacency matrix of the signed network $G = (V, E_+, E_-)$, where positive edges $(i, j) \in E_+$ are indicated by $A_{ij} = 1$, negative edges $(i, j) \in E_-$ are indicated by $A_{ij} = -1$, and non-edges are indicated by $A_{ij} = 0$. A partition S_0, S_1, S_2 of V can be represented by a vector $\mathbf{x} \in \{-1, 0, 1\}^n$, whose i-th coordinate is $x_i = 0$ if $i \in S_0$, $x_i = 1$ if $i \in S_1$, and $x_i = -1$ if $i \in S_2$. Then 2CC-Full can be reformulated as follows.

Problem 3 (2CC-Full). *Given a signed network $G = (V, E_+, E_-)$ with n vertices and signed adjacency matrix A, find a partition S_0, S_1, S_2 of V represented by vector $\mathbf{x} \in \{-1, 0, 1\}^n$ maximizing*

$$\overline{cc}(S_1, S_2) = \mathbf{x}^T A\,\mathbf{x}.$$

Since our goal is to discover polarized communities S_1 and S_2 that are potentially concealed within other neutral vertices S_0, we want to find minimal sets S_1 and S_2. This can be achieved by normalizing $\mathbf{x}^T A\,\mathbf{x}$ with the size of S_1 and S_2, which in vector form is $\mathbf{x}^T\mathbf{x}$. This consideration motivates our last problem formulation, which we dub 2-Polarized-Communities (2PC).

Problem 4 (2PC). *Given a signed network $G = (V, E_+, E_-)$ with n vertices and signed adjacency matrix A, find a vector $\mathbf{x} \in \{-1, 0, 1\}^n$ that maximizes*

$$\frac{\mathbf{x}^T A\,\mathbf{x}}{\mathbf{x}^T\mathbf{x}}.$$

In the rest of this paper we refer to the objective function of Problem 4 as *polarity*. As polarity is penalized with the size of the solution, vertices are only added to one of the two clusters if they contribute significantly to the objective. We show this problem to be **NP**-hard (proof in the Appendix) and propose algorithms with approximation guarantees.

Theorem 3.3. *2PC is* **NP***-hard.*

It should be noted that 2PC does not enforce balance between the communities. This can be beneficial if there exist polarized communities of significantly different size in the input network. In an extreme case, the solution could even be comprised of a single cluster if there is a large, dense community that overwhelms any other polarized formation.

4 ALGORITHMS

The formulation of 2PC is suggestive of spectral theory, which we utilize to design our algorithms. We propose and analyze two spectral algorithms: one is deterministic, while the second is randomized and achieves approximation guarantee $\sqrt{n}$. The running time of both algorithms is dominated by the computation of a spectral decomposition of the adjacency matrix. In practice, this can be done using readily available implementations that exploit sparsity and can run in parallel on multiple cores.

The first algorithm, Eigensign, works by simply discretizing the entries of the eigenvector of the adjacency matrix corresponding to the largest eigenvalue.

To illustrate the difficulty of approximating 2PC, we analyze the following simple algorithm, which we refer to as Pick-an-edge. Pick an arbitrary edge: if it is positive, put the endpoints in one cluster, leaving the other cluster empty; if it is negative, put the endpoints in separate clusters.

Proposition 4.1. *The* Pick-an-edge *algorithm gives an n-approximation of the optimum.*

Proof. The described algorithm outputs a solution $\mathbf{x}$ such that

$$\frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \geq 1.$$

The result now follows from the fact that $OPT \leq \lambda_1 \leq n$, where λ_1 is the largest eigenvalue of A. □

In the case of networks with arbitrary real weights, it can be shown that despite the close relationship between the 2PC objective and the leading eigenvector of A, Eigensign cannot do better than this up to constant factors. Consider a fully connected network with one edge (u, v) of weight $w \gg 0$. The rest of the edges have weight close to zero. The primary eigenvector of the adjacency matrix has two entries — those corresponding to u and v — of the form $1/\sqrt{2} - \epsilon$ for some small ϵ, while the rest are close to zero. We construct a solution vector $\mathbf{y}$ as follows: the two entries corresponding to u and v are set to 1, and the rest to 0. We have $\mathbf{y}^T A \mathbf{y}/\mathbf{y}^T \mathbf{y} \approx w$. On the other hand, the Eigensign algorithm outputs a vector $\mathbf{x}$ for which $\mathbf{x}^T A \mathbf{x}/\mathbf{x}^T \mathbf{x} \approx 2w/n$. It should be noted however, that the focus of this paper is the analysis of the 2PC problem on signed networks. The approximation capabilities of the Eigensign algorithm on signed networks (the adjacency matrix A contains entries with values only −1, 0, and 1) are left open.

Eigensign generally outputs a solution comprised of all the vertices in the graph — unless some components of the eigenvector are exactly zero — which is, of course, counter to the motivation of our problem setting.

To overcome this issue we propose a randomized algorithm, Random-Eigensign, which also computes the first eigenvector, i.e., $\mathbf{v}$, of the adjacency matrix. Instead of simply discretizing the entries of $\mathbf{v}$, it randomly sets each entry of $\mathbf{x}$ to 1 or -1 with probabilities determined by the entries of $\mathbf{v}$. Entries v_i with large magnitude $|v_i|$ are more likely to turn into $sgn(v_i)$ (−1 or 1), while entries v_i with small magnitude $|v_i|$ are more likely to turn into 0. For details see Algorithm 2. Note that if $\mathbf{x}$ is the output of Random-Eigensign, then $\mathbb{E}[\mathbf{x}] = \mathbf{v}$.

The next theorem shows approximation guarantees of Random-Eigensign for signed networks.

Theorem 4.2. *Algorithm* Random-Eigensign *gives a $\sqrt{n}$-approximation of the optimum in expectation.*

Proof. First, observe that we can rewrite the expected value of the objective as follows:

$$\begin{aligned}
\mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] &= \sum_{k=1}^{n} \mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\middle|\mathbf{x}^T\mathbf{x} = k\right] Pr(\mathbf{x}^T\mathbf{x} = k) \\
&= \sum_{k=1}^{n} \frac{1}{k}\mathbb{E}\left[\mathbf{x}^T A\mathbf{x}|\mathbf{x}^T\mathbf{x} = k\right] Pr(\mathbf{x}^T\mathbf{x} = k) \\
&= \sum_{k=1}^{n} \frac{1}{k}\sum_{i\neq j}\mathbb{E}\left[A_{ij}x_i x_j|\mathbf{x}^T\mathbf{x} = k\right] Pr(\mathbf{x}^T\mathbf{x} = k).
\end{aligned}$$

If we define $s_{ij} = sgn(v_i)sgn(v_j)$, where $sgn(x)$ denotes the sign of $x \in \mathbb{R}$, for all i, j we have

$$\begin{aligned}
&\mathbb{E}\left[A_{ij}x_i x_j|\mathbf{x}^T\mathbf{x} = k\right] Pr(\mathbf{x}^T\mathbf{x} = k) \\
&= A_{ij}s_{ij}Pr(x_i = 1, x_j = 1|\mathbf{x}^T\mathbf{x} = k)Pr(\mathbf{x}^T\mathbf{x} = k). \qquad (1)
\end{aligned}$$

We now invoke Bayes' theorem and proceed.

$$\begin{aligned}
&\sum_{k=1}^{n} \frac{1}{k}\sum_{i\neq j} A_{ij}s_{ij}Pr(x_i = 1, x_j = 1)Pr(\mathbf{x}^T\mathbf{x} = k|x_i = 1, x_j = 1) \\
&= \sum_{k=1}^{n} \frac{1}{k}\sum_{i\neq j} A_{ij}v_i v_j Pr(\mathbf{x}^T\mathbf{x} = k|x_i = 1, x_j = 1) \\
&= \sum_{i\neq j} A_{ij}v_i v_j \sum_{k=1}^{n} \frac{1}{k} Pr(\mathbf{x}^T\mathbf{x} = k|x_i = 1, x_j = 1) \\
&= \sum_{i\neq j} A_{ij}v_i v_j \mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}}|x_i = 1, x_j = 1\right].
\end{aligned}$$

Since $1/x$ is a convex function, by Jensen's inequality it is

$$\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}}|x_i = 1, x_j = 1\right] \geq \frac{1}{\mathbb{E}\left[\mathbf{x}^T\mathbf{x}|x_i = 1, x_j = 1\right]}.$$

Furthermore, for any i, j,

$$\mathbb{E}\left[\mathbf{x}^T\mathbf{x}|x_i = 1, x_j = 1\right] \leq 2 + \sqrt{n-2}.$$

To see this, observe that $\mathbb{E}\left[\mathbf{x}^T\mathbf{x}\right] = \|\mathbf{v}\|_1 \leq \sqrt{n}$. So we have

$$\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}}|x_i = 1, x_j = 1\right] \geq \frac{1}{2 + \sqrt{n-2}}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] &= \sum_{i\neq j} A_{ij}v_i v_j \mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}}|x_i = 1, x_j = 1\right] \\
&\geq \sum_{i\neq j} A_{ij}v_i v_j \frac{1}{2 + \sqrt{n-2}} = \frac{\lambda_1}{2 + \sqrt{n-2}}.
\end{aligned}$$

That is,

$$O(\sqrt{n})\mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] \geq \lambda_1 \geq OPT.$$

□

In the appendix we show that this result is tight.

Algorithm 1 Eigensign

```
Input: adjacency matrix $A$
Compute $\mathbf{v}$, the eigenvector corresponding to the largest eigenvalue $\lambda_1$ of $A$.
Construct $\mathbf{x}$ as follows: for each $i \in \{1, \dots, n\}$, $x_i = sgn(v_i)$.
Output $\mathbf{x}$.
```

Algorithm 2 Random-Eigensign

```
Input: adjacency matrix $A$
Compute $\mathbf{v}$, the eigenvector corresponding to the largest eigenvalue $\lambda_1$ of $A$.
Construct $\mathbf{x}$ as follows: for each $i \in \{1, \dots, n\}$, run a Bernoulli experiment with success probability $|v_i|$. If it succeeds, then $x_i = sgn(v_i)$, otherwise $x_i = 0$.
Output $\mathbf{x}$.
```

4.1 Enhancements for practical use

When using these algorithms to analyze real-world networks in practical applications, it might be beneficial to apply tweaks to enhance their flexibility and produce a wider variety of results. We propose the following simple enhancements.

Eigensign: As discussed above, Eigensign always outputs a solution involving all the vertices in the network. We can circumvent this shortcoming by including only those vertices such that the corresponding entry of the eigenvector $\mathbf{v}$ is at least a user-defined threshold τ. That is, $x_i = sgn(v_i)$ if $|v_i| \geq \tau$, 0 otherwise.

Random-Eigensign: The $\sqrt{n}$-approximation guaranteed by Random-Eigensign is matched in the extreme case in which all entries of the eigenvector $\mathbf{v}$ are of equal magnitude. Paradoxically, in this situation a solution comprised of all vertices would be optimal, but each vertex is included with a small probability of $1/\sqrt{n}$. We could of course fix this by modifying the probabilities to be $\min\{1, \sqrt{n}|v_i|\}$ for each i. However, in the opposite extreme, where most of the magnitude of $\mathbf{v}$ is concentrated in one entry, modifying the probabilities this way might disproportionately boost the likelihood of including undesirable vertices. An adequate multiplicative factor for both cases is $\|\mathbf{v}\|_1$, modifying the probabilities to be $\min\{1, \|\mathbf{v}\|_1 |v_i|\}$ for each i; in the first case, all vertices are taken with probability 1, while in the second, the probabilities remain almost unchanged. We employed this factor in our experiments with satisfactory results.

An obvious question arising is whether the approximation guarantee of Random-Eigensign could be improved using the modification described above. This question is left for future investigation.

5 EXPERIMENTAL ASSESSMENT

This section presents the evaluation of the proposed algorithms: first (Section 5.1) we present a characterization of the polarized communities discovered by our methods; then (Section 5.2) we compare our methods against non-trivial baselines in terms of objective, efficiency and scalability, and ability to detect ground-truth planted polarized communities in synthetic datasets. Finally, we show a case study about political debates (Section 5.3).

Datasets. We select publicly-available real-world signed networks, whose main characteristics are summarized in Table 1. HighlandTribes[1] represents the alliance structure of the Gahuku–Gama tribes of New Guinea. Cloister[1] contains the esteem/disesteem relations of monks living in a cloister in New England (USA). Congress[1] reports (un/)favorable mentions of politicians speaking in the US Congress. Bitcoin[2] and Epinions[2] are who-trusts-whom networks of the users of Bitcoin OTC and Epinions, respectively. WikiElections[1] includes the votes about admin elections of the users of the English Wikipedia. Referendum[3] [32] records Twitter data about the 2016 Italian Referendum: an interaction is negative if two users are classified with different stances, and is positive otherwise. Slashdot[2] contains friend/foe links between the users of Slashdot. The edges of WikiConflict[2] represent positive and negative edit conflicts between the users of the English Wikipedia. WikiPolitics[1] represents interpreted interactions between the users of the English Wikipedia that have edited pages about politics.

Table 1: Signed networks used: number of vertices and edges; ratio of negative edges ($\rho_- = \frac{|E_-|}{|E_+ \cup E_-|}$); L_1-norm of the eigenvector corresponding the largest eigenvalue of A ($\|\mathbf{v}\|_1$); and, ratio of non-zero elements of A ($\delta = \frac{2|E_+ \cup E_-|}{|V|(|V|-1)}$).

Real-world datasets	$\|V\|$	$\|E_+ \cup E_-\|$	ρ_-	$\|\mathbf{v}\|_1$	δ
HighlandTribes	16	58	0.50	3.61	0.48
Cloister	18	125	0.55	3.71	0.81
Congress	219	521	0.20	10.51	0.02
Bitcoin	5 k	21 k	0.15	31.21	$1.2e-03$
WikiElections	7 k	100 k	0.22	35.96	$3.9e-03$
Referendum	10 k	251 k	0.05	42.66	$4.2e-03$
Slashdot	82 k	500 k	0.23	59.46	$1.4e-04$
WikiConflict	116 k	2 M	0.62	119.66	$2.9e-04$
Epinions	131 k	711 k	0.17	72.20	$8.2e-05$
WikiPolitics	138 k	715 k	0.12	91.48	$7.4e-05$
WikiConflict16$\|V\|$	1 M	67 M	0.62	129.04	$3.4e-05$
Epinions16$\|V\|$	2 M	23 M	0.17	75.99	$9.5e-06$

In order to study scalability, we artificially augment two of the largest datasets to produce networks with millions of vertices and tens of millions of edges (details in Section 5.2).

Implementation. All methods, with the exception of algorithm FOCG (details in Section 5.2), are implemented in Python (v. 2.7.15) and compiled by Cython. The experiments run on a machine equipped with Intel Xeon CPU at 2.1GHz and 128GB RAM.[4]

5.1 Solutions characterization

We first characterize the solutions discovered by our methods Eigensign (for short E) and Random-Eigensign (RE), and we show how the tweaks described in Section 4.1 enhance their flexibility in producing a wider variety of results. In particular, algorithm E evaluates the threshold τ for each $|v_i|$ discretized at the third decimal digit. This operation is carried out efficiently, since $\mathbf{v}$ is computed only once regardless of the number of evaluated values of τ. On the other hand, algorithm RE employs $\|v\|_1$ as multiplicative factor, therefore the probabilities are modified to be $\min\{1, \|\mathbf{v}\|_1 |v_i|\}$. In the following, we refer to the two communities included in the solutions as

[1] konect.cc

[2] snap.stanford.edu

[3] researchgate.net/publication/324517807_Annotated_Corpus_for_Stance_Detection_-_Italian_Constitutional_Referendum_2016

[4] Code and datasets available at github.com/egalimberti/polarized_communities.

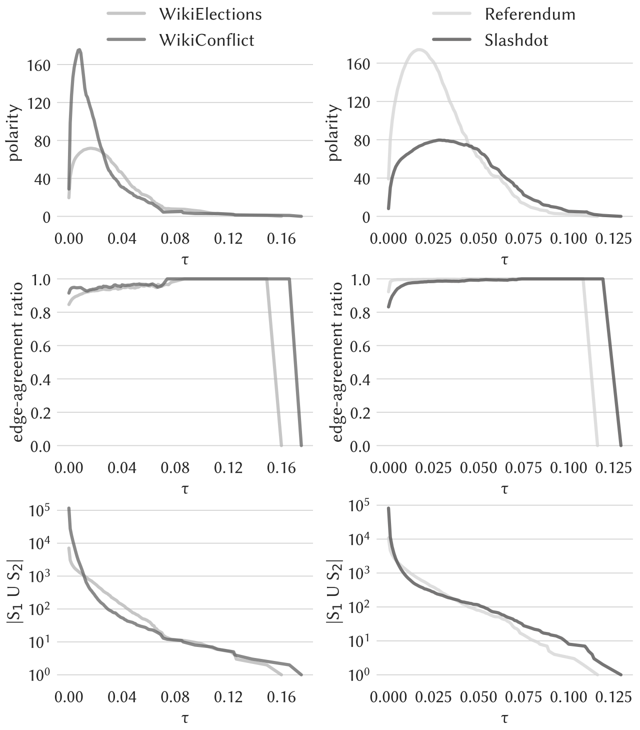

Figure 2: Solutions produced by E **as a function of** τ**.**

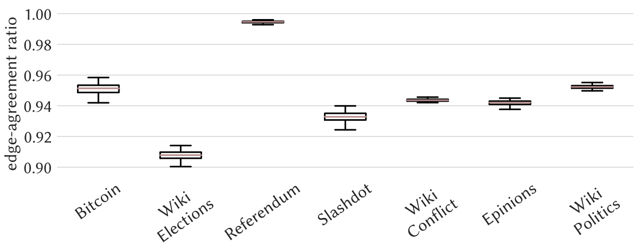

Figure 3: Edge-agreement ratio of the solutions of RE**.**

S_1 and S_2, namely the subsets of vertices that are assigned with 1 and -1, respectively, by the solution vector $\mathbf{x}$.

Figure 2 shows how the solutions returned by algorithm E are affected by parameter τ in terms of polarity, edge-agreement ratio (i.e., the portion of edges in the solution that comply with the polarized structure), and size on four datasets. In all of them, the three measures follow very similar trends. The highest polarity is achieved at about a fourth of the domain of τ, when most of the neutral vertices are discarded. The edge-agreement ratio, instead, is consistently close or equal to 1: the solutions have a coherent polarized structure regardless of the chosen τ. Finally, as expected, the number of vertices included in the solutions decreases as τ grows, and presents a substantial decay at the beginning of the domain. Therefore, parameter τ is a powerful enhancement that allows algorithm E to be tuned to return the most suitable solution for the domain under analysis.

For algorithm RE, due to the randomness, we report the best solution with respect to polarity out of 100 runs. We do the same for the baseline LS, that we introduce in Section 5.2. Figure 3 shows the boxplots of the edge-agreement ratio over the larger datasets. It has significant values in all cases, above 0.9, and is stable among the different executions. Polarity and solution size for all datasets are reported in Figure 4. For such measures, we do not show boxplots as they are highly dependent on the specific dataset and very stable over different runs: their index of dispersion is lower than 0.01 and $3.2e{-}05$, respectively, for all datasets. This confirms that algorithm RE is very stable and does not require multiple executions to identify high-quality solutions.

5.2 Comparative evaluation

We next compare algorithms E and RE against non-trivial baselines inspired by methods proposed in the literature for different yet related problems.

FOCG. The first method we compare to, whose objective is to find k *oppositive cohesive groups* (i.e., k-OCG) in signed networks, is taken from [14]. Algorithm FOCG detects p different k-OCG structures within the input signed network, among which we elect the one having highest polarity as the ultimate solution to our problem. We setup the algorithm with the default configuration (i.e., $\alpha = 0.3$ and $\beta = 50$) and $k = 2$. The code is provided by the authors.

Greedy. Our second baseline is inspired by the 2-approximation algorithm for *densest subgraph* [11]. Algorithm Greedy (for short G), iteratively removes the vertex minimizing the difference between the number of positive adjacent edges and the number of negative adjacent edges, up to when the graph is empty. At the end, it returns the subgraph having the highest polarity among all subgraphs visited during its execution. The assignment of the vertices to the clusters is guided by the sign of the components of the eigenvector $\mathbf{v}$, corresponding to the largest eigenvalue of A.

Bansal. A different approach, motivated by the strong similarity to our setting, is inspired by Bansal's 3-approximation algorithm for 2CC on complete signed graphs [6]. For each vertex $u \in V$, this algorithm, which we refer to as Bansal (for short B), identifies u together with the vertices sharing a positive edge with u as one cluster, and the vertices sharing a negative edge as the other. Of these $|V|$ possible solutions, it returns the one maximizing polarity.

LocalSearch. Finally, we consider a local search approach (LocalSearch, for short LS), guided by our objective function. Algorithm LS starts from a set of vertices chosen at random; at each iteration, it adds (removes) to (from) the current solution the vertex that maximizes the gain in terms of polarity, and finally terminates when the gain of moving any vertex is lower than 0.2. Also for this algorithm, the assignment of the vertices to the clusters is guided by the signs of $\mathbf{v}$.

Figure 4 reports the achieved values of polarity for all compared algorithms on all datasets, as well as the size (normalized by $|V|$) of the solutions returned. In most of the cases, algorithm E results the be the most competitive method with respect to polarity; on the other hand, algorithm RE is able to return solutions of high polarity for the small-sized datasets. Algorithm FOCG is instead not competitive since its solutions are of extremely small size (note that the numerator of our objective can be up to quadratic in the size of the denominator, so size matters for reaching high polarity). Algorithm G has, in general, polarity comparable to algorithm E, slightly higher in a few cases (with the exception of WikiConflict, in which algorithm E clearly outperforms algorithm G). However, it must

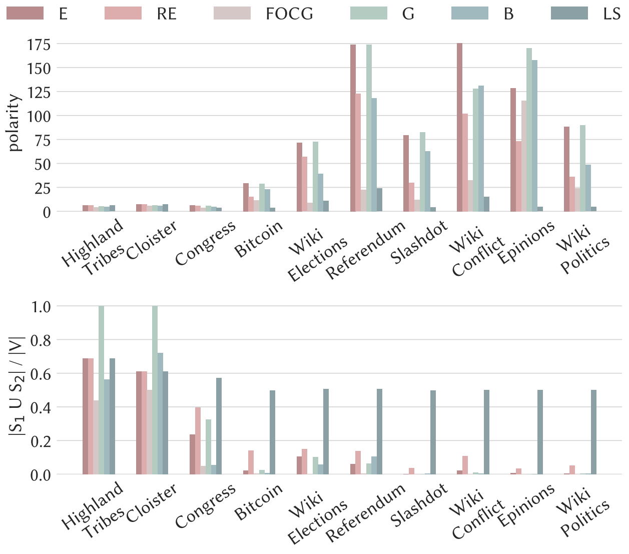

Figure 4: Polarity and solution size (normalized) of the proposed algorithms and baselines.

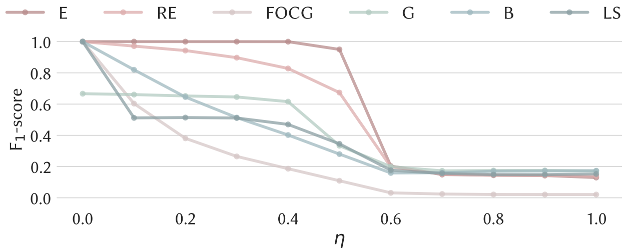

Figure 5: F_1-score as a function of the noise parameter η ($n_c = 100$, $n_n = 800$).

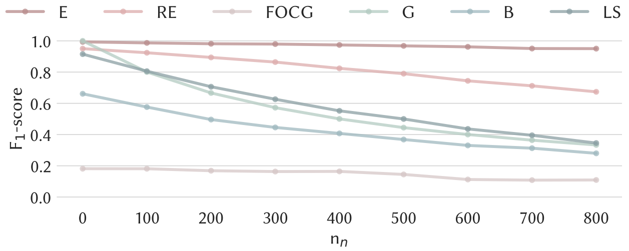

Figure 6: F_1-score as a function of the number noisy vertices n_n ($n_c = 100$, $\eta = 0.5$).

be noted that algorithm G often returns a very dense subgraph as one of the two communities, leaving the second community totally empty, which is, of course, undesirable in our context. Algorithms B and LS, instead, exhibit weak performance in terms of polarity: their search spaces strongly depend on the neighborhood structure of the vertices (for B), or on the random starting sets (for LS). About the solution size, all methods, with the exception of algorithms FOCG and LS, return solutions of reasonable dimension with respect to the number of vertices of the networks. Excluding the small empirical datasets (i.e., HighlandTribes, Cloister, and Congress), the size of the solutions is below 20% of the input.

Planted polarized communities. In order to better assess the effectiveness of the various algorithms, we test their ability to detect a known planted solution, concealed within varying amounts of noise. For our purposes we create a collection of synthetic signed networks identified by three parameters: the size of each planted polarized community $n_c = |S_1| = |S_2|$ (for convenience, we consider communities having the same size); the number of noisy vertices external to the two polarized communities $n_n = |V \setminus (S_1 \cup S_2)|$; and, a *noise parameter* $\eta \in [0, 1]$ governing the edge density and agreement to the model. In detail:

- edges inside S_1 (respect. S_2) exist and are positive with probability $1 - \eta$, exist and are negative with probability $\eta/2$, and do not exist with probability $\eta/2$;
- edges between S_1 and S_2 exist and are negative with probability $1 - \eta$, exist and are positive with probability $\eta/2$, and do not exist with probability $\eta/2$;
- all other edges (outside the two polarized communities) exist with probability η and have equal probability of being positive or negative.

The higher η, the less internally dense and polarized the two communities are, and the more connected the noisy vertices are, both between themselves and to the communities. Observe how the case with no noise ($\eta = 0$) corresponds to the “perfect” structure.

For each configuration of the parameters, we create 10 different networks and we report the average F_1-score in detecting which vertices belong to S_1 (respect. S_2) and which ones to $V \setminus (S_1 \cup S_2)$[5].

In Figure 5 we fix the size of the synthetic network to 1 000 ($n_c = 100$, $n_n = 800$) and vary η. For $\eta = 0$, all algorithms have, as expected, maximum F_1-score with the exception of algorithm G that, even in the case without noise, is not able to exactly identify the planted structure. As expected, as η increases, the F_1-score decays for all methods; however, our algorithms E and RE clearly outperform the others. Figure 6 shows the F_1-score varying the number n_n of vertices external to the polarized communities, with fixed $n_c = 100$ and $\eta = 0.5$. Again algorithms E and RE stand out, especially E that presents F_1-score close to the maximum in all cases. Algorithm FOCG has the poorest performance: the small size of its solutions penalizes the recall, which is never greater than 0.1.

Runtime and scalability. Figure 7 reports the runtime of all algorithms over all datasets. Algorithms E and RE, with their practical enhancements discussed in Section 4.1, always terminate in less than 40 seconds. The runtime of the baselines is instead more than an order of magnitude higher than algorithms E and RE.

In order to assess the scalability of our methods, we augment two of the larger datasets (i.e., WikiConflict and Epinions) by artificially injecting dummy vertices having a number of randomly-connected edges equal to the average degree of the original network, while maintaining ρ_- (i.e., the ratio of negative edges). The largest datasets created in this way contain up to 2 M vertices and 67 M edges (see Table 1 for details). Note that, as the quantity of noise increases, δ, i.e., the ratio of non-zero elements of the adjacency matrix, decreases. Nonetheless, δ differs with respect to the original datasets less than an order of magnitude in both cases, making the following results about scalability significant.

[5] For instance *recall* is defined as $(|S_1^* \cap S_1| + |S_2^* \cap S_2|)/|S_1 \cup S_2|$, where S_1^* (S_2^*) denotes the first (second) community returned by the algorithm while S_1 (S_2) denotes the corresponding ground-truth one.

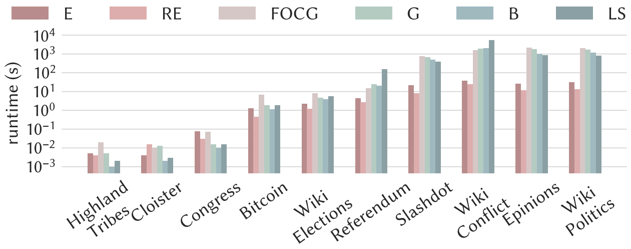

Figure 7: Runtime of the proposed algorithms and baselines.

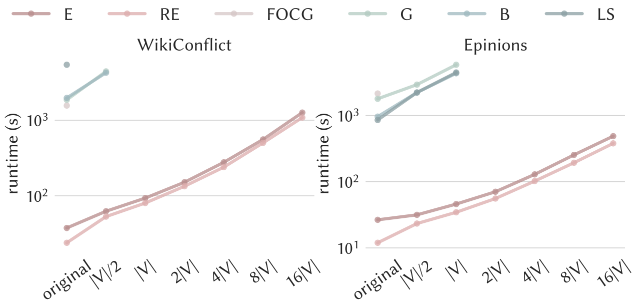

Figure 8: Scalability: runtime of the proposed algorithms and baselines as a function of the number of injected dummy vertices, for WikiConflict and Epinions.

Figure 8, which reports on the x-axis the number of dummy vertices added, shows that the runtime of both algorithms E and RE grows linearly with the number of vertices. Among the two, algorithm E is slightly slower than algorithm RE due to the evaluation of multiple values of the threshold τ. In the worst case, algorithms E returns in about 21 minutes. On the other hand, the baselines cannot complete each computation within the 10 000 seconds timeout that we apply. No baseline terminates for more than $|V|$ additional dummy vertices on both datasets. In particular, algorithm FOCG is able to handle in reasonable time only the original versions, with no dummy vertices. It should be noted that algorithm FOCG recursively finds a polarized structure, removes the corresponding subgraph, and repeats the process on the remaining vertices. While each one of these iterations runs efficiently, most of the found structures are too small to be of interest in our setting. Thus, it is necessary to allow the algorithm to complete many of such iterations in order to find interesting solutions.

5.3 Case study: political debate

We finally analyze the solution extracted by algorithm RE from Referendum to show tangible benefits of our problem formulation and algorithms in identifying the two most polarized communities in a signed network modeling political debates. The Referendum dataset includes Twitter data about the Italian Constitutional Referendum held on December 4, 2016 (more information about the Referendum can be found at this link). The original data seed consists of about 1 M tweets posted between November 24 and December 7, 2016, extended by collecting retweets, quotes, and replies. The users (10 884 in total) are annotated with a stance about their outlook towards the Referendum as favorable (5 137), against (1 510), or none (4 237) when the stance cannot be inferred. An interaction (edge) is considered negative if occurred between two users (vertices) of different stances, and is positive otherwise, i.e., we treat "none" users as neutral, in agreement with both favorable and against users.

The solution output by algorithm RE consists of two communities of 27 and 1 558 users, accounting for 14% of the overall user set. Both communities have more than 99% of positive edges within and 74% of negative edges in-between, and thus, are highly polarized. Interestingly, all the 27 users of the smaller community are classified as favorable to the Referendum, while the users in the larger community as against (75%) or "none" (24%), with the exception of 3 favorables. Moreover, the vertices in the solution have, on average, 183.12 adjacent edges compared to the average 22.85 contacts of the vertices outside, meaning that the solution identifies the "core" of the controversies, i.e., a set of intensely debating users about the Referendum. These results provide evidence of the practical value of our problem formulation and algorithms to identify two communities that are polarized about a certain topic.

6 CONCLUSIONS AND FUTURE WORK

Detecting extremely polarized communities might enable fine-grained analysis of controversy in social networks, as well as open the door to interventions aimed at reducing it [21]. As a step in this direction, in this paper we introduce the 2-Polarized-Communities problem, which requires finding two communities (subsets of the network vertices) such that within communities there are mostly positive edges while across communities there are mostly negative edges. We prove that the proposed problem is **NP**-hard and devise two efficient algorithms with provable approximation guarantees. Through an extensive set of experiments on a wide variety of real-world networks, we show how the proposed objective function can be optimized to reveal polarized communities. Our experiments confirm that our algorithms are more accurate, faster, and more scalable than non-trivial baselines.

This work opens several enticing avenues for further inquiry. Some questions follow immediately from our theoretical results. What are the approximation capabilities of Eigensign in signed networks? Can we improve the $\sqrt{n}$ factor of Random-Eigensign, e.g., by multiplying the probability vectors by $\|v\|_1$ or some other factor? Finally, it would be interesting to extend the problem to detect an arbitrary number of communities.

The application of the proposed algorithms to real-world networks with positive and negative relationships can have implications in computational social science problems. For instance, understanding opinion shifts in data streaming from social media sources can be investigated in terms of polarized communities. Opinions shared within vertices (individuals) belonging to the same community are likely to be reinforced after different interactions; discussions within individuals with antagonistic perspectives may result in both opinion shifts and controversy amplification. The identification of a subgraph made of vertices belonging to 2-Polarized-Communities may lead to novel ways to discover the basic laws behind opinion shift dynamics. Thus, it would be interesting to study extensions of the 2-Polarized-Communities problem in the setting of temporal networks.

REFERENCES

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

[2] L. Akoglu. Quantifying political polarity based on bipartite opinion networks. In *ICWSM*, 2014.

[3] P. Anchuri and M. Magdon-Ismail. Communities and balance in signed networks: A spectral approach. In *ASONAM*, 2012.

[4] T. Antal, P. Krapivsky, and S. Redner. Social balance on networks: The dynamics of friendship and enmity. *Physica D*, 224(130), 2006.

[5] D. Baldassarri and P. Bearman. Dynamics of political polarization. *American sociological review*, 72(5):784–811, 2007.

[6] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.

[7] G. Beigi, J. Tang, and H. Liu. Signed link analysis in social media networks. In *ICWSM*, 2016.

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

[9] D. Cartwright and F. Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

[10] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. A correlation clustering approach to link classification in signed networks. In *COLT*, 2012.

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

[12] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon. Scalable clustering of signed networks using balance normalized cut. In *CIKM*, 2012.

[13] Y. Choi, Y. Jung, and S.-H. Myaeng. Identifying controversial issues and their sub-topics in news articles. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, 2010.

[14] L. Chu, Z. Wang, J. Pei, J. Wang, Z. Zhao, and E. Chen. Finding gangs in war from signed networks. In *KDD*, 2016.

[15] T. Coleman, J. Saunderson, and A. Wirth. A local-search 2-approximation for 2-correlation-clustering. In *ESA*, 2008.

[16] M. D. Conover, B. Gonçalves, J. Ratkiewicz, A. Flammini, and F. Menczer. Predicting the political alignment of twitter users. In *SocialCom/PASSAT*, 2011.

[17] F. Cribari-Neto, N.L. Garcia, and K. Vasconcellos. A note on inverse moments of binomial variates. Brazilian Review of Econometrics, 20:2, 2000.

[18] J.-M. Esteban and D. Ray. On the measurement of polarization. *Econometrica: Journal of the Econometric Society*, pages 819–851, 1994.

[19] L. Feldman, T. A. Myers, J. D. Hmielowski, and A. Leiserowitz. The mutual reinforcement of media selectivity and effects: Testing the reinforcing spirals framework in the context of global warming. *Journal of Communication*, 64(4):590–611, 2014.

[20] M. Gao, E.-P. Lim, D. Lo, and P. K. Prasetyo. On detecting maximal quasi antagonistic communities in signed graphs. *Data mining and knowledge discovery*, 30(1):99–146, 2016.

[21] K. Garimella, G. De Francisci Morales, A. Gionis, and M. Mathioudakis. Reducing controversy by connecting opposing views. In *WSDM*, 2017.

[22] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):3, 2018.

[23] K. Garrett and N. J. Stroud. Partisan paths to exposure diversity: Differences in pro-and counter-attitudinal news consumption. *Journal of Communication*, 64(4):680–701, 2014.

[24] I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. In *SODA*, 2006.

[25] E. Graells-Garrido, M. Lalmas, and D. Quercia. People of opposing views can share common interests. In *WWW*, 2014.

[26] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

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

[28] Y. Hou, J. Li, and Y. Pan. On the Laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, 51(1):21–30, 2003.

[29] Y. P. Hou. Bounds for the least Laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4):955–960, 2005.

[30] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *WWW*, 2009.

[31] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *SDM*, 2010.

[32] M. Lai, V. Patti, G. Ruffo, and P. Rosso. Stance evolution and twitter interactions in an italian political debate. In *NLDB*, 2018.

[33] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, 2010.

[34] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *SIGCHI*, 2010.

[35] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *WSDM*, 2013.

[36] Q. V. Liao and W.-T. Fu. Can you hear me now?: mitigating the echo chamber effect by source position indicators. In *CSCW*, 2014.

[37] Q. V. Liao and W.-T. Fu. Expert voices in echo chambers: effects of source expertise indicators on exposure to diverse opinions. In *SIGCHI*, 2014.

[38] D. Lo, D. Surian, K. Zhang, and E.-P. Lim. Mining direct antagonistic communities in explicit trust networks. In *CIKM*, 2011.

[39] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *RecSys*, 2009.

[40] A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: theory of majorization and its applications*, volume 143. Springer, 1979.

[41] S. A. Marvel, S. H. Strogatz, and J. M. Kleinberg. The energy landscape of social balance. *Physical Review Letters*, 103(19), 2009.

[42] Y. Mejova, A. X. Zhang, N. Diakopoulos, and C. Castillo. Controversy and sentiment in online news. *CJ'14: Computation+Journalism Symposium*, 2014.

[43] S. A. Munson, S. Y. Lee, and P. Resnick. Encouraging reading of diverse political viewpoints with a browser widget. In *ICWSM*, 2013.

[44] A.-M. Popescu and M. Pennacchiotti. Detecting controversial events from twitter. In *CIKM*, 2010.

[45] G. J. Puleo and O. Milenkovic. Correlation clustering with constrained cluster sizes and extended weights bounds. *SIAM Journal on Optimization*, 25(3), 2015.

[46] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.

[47] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[48] C. Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *SODA*, 2004.

[49] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. In *RecSys*, 2010.

[50] J. Tang, C. Aggarwal, and H. Liu. Node classification in signed social networks. In *SDM*, 2016.

[51] J. Tang, C. Aggarwal, and H. Liu. Recommendations in signed social networks. In *WWW*, 2016.

[52] J. Tang, Y. Chang, C. Aggarwal, and H. Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3):42, 2016.

[53] P. Victor, C. Cornelis, M. De Cock, and A. M. Teredesai. Trust-and distrust-based recommendations for controversial reviews. *IEEE Intelligent Systems*, 26(1), 2011.

[54] V. Vydiswaran, C. Zhai, D. Roth, and P. Pirolli. Overcoming bias to learn about controversial topics. *Journal of the Association for Information Science and Technology*, 66(8):1655–1672, 2015.

[55] M. Wojcieszak and D. Mutz. Online groups and political discourse: Do online discussion spaces facilitate exposure to political disagreement? *Journal of communication*, 59(1):40–56, 2009.

A APPENDIX

Hardness (Theorem 3.3). In this section, we refer to a solution of 2PC as S_1, S_2, which denote the subsets of vertices that are assigned a 1 or a -1, respectively, in the solution vector $\mathbf{x}$. Given a vertex $v \in V$ and a subset of vertices $S \subseteq V$, we use $d_+(v, S)$ (respectively $d_-(v, S)$) to denote the number of '+' edges (respectively '−' edges) connecting v to other vertex in S.

We exploit the following result in our proof. It can be easily verified by examining the behavior of the cost functions when moving one vertex from one set to the other, so we omit the proof.

Proposition A.1. *If we require $S_1 \cup S_2 = V$, problem* 2CC-Full *is equivalent to* 2CC*, i.e., their optimal solutions are the same.*

We now prove that 2PC is **NP**-hard by reduction from 2CC, which has been shown to be **NP**-hard by Shamir *et al.* [46].

Proof of Theorem 3.3. Given a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ with n vertices as instance of 2CC, we construct a graph $G = (V, E)$ to be an instance of 2PC as follows. For every vertex in $\tilde{V}$ we create a corresponding vertex in V, and for every edge in $\tilde{E}$ we add an edge in E between the corresponding vertices in $\tilde{V}$, and having the same sign. Furthermore, for every vertex v in $\tilde{V}$ we introduce a clique of $m > 3n$ vertices (and positive edges) and a '+' edge between v and every vertex in the clique. The strategy to prove hardness is the following. We first restrict ourselves to *complete* solutions of 2PC (i.e. $S_1 \cup S_2 = V$), which can of course be mapped to solutions of 2CC. We prove that if one such complete solution S_1, S_2 optimizes 2PC, the corresponding solution of 2CC is also a maximizer. Second, we show that any optimal solution of the constructed instance of 2PC is complete.

We denote the objective of the problems 2CC and 2PC, on instances $\tilde{G}$ and G, by W_{2CC} and W_{HPC}, respectively. We consider a solution $\tilde{S}_1, \tilde{S}_2$ of 2CC, and a solution S_1, S_2 of 2PC, such that $\tilde{S}_1 \subseteq S_1$ and $\tilde{S}_2 \subseteq S_2$. Let us first restrict our attention to complete solutions of 2PC. Observe that

$$W_{HPC}(S_1, S_2) = \frac{1}{n+nm}\left(W_{2CC}(\tilde{S}_1, \tilde{S}_2) - D(\tilde{S}_1, \tilde{S}_2)\right) + \frac{1}{n+nm}\left(|S_1|m + |S_2|m + n\binom{m}{2}\right),$$

where $D(\tilde{S}_1, \tilde{S}_2) = \sum_{v \in \tilde{S}_1} d_+(v, \tilde{S}_2) + \sum_{v \in \tilde{S}_2} d_+(v, \tilde{S}_2) + d_-(v, \tilde{S}_1) + d_-(v, \tilde{S}_2)$, that is, the sum of disagreements in the resulting clustering. Note that $W_{2CC}(\tilde{S}_1, \tilde{S}_2) - D(\tilde{S}_1, \tilde{S}_2)$ is exactly the objective of the 2CC-Full problem. In other words, the obective of 2PC on G is proportional to the objective of 2CC-Full on $\tilde{G}$ plus a constant. By Proposition A.1, the first part of the proof is complete.

We now consider a complete solution S_1, S_2 and show that removing vertices leads to no further improvement. Suppose we remove a set R of r vertices from the solution. We want to show

$$\frac{v(W_{HPC}(S_1, S_2)) - \Delta(R)}{n + nm - r} < \frac{v(W_{HPC}(S_1, S_2))}{n + nm},$$

where $v(W_{HPC}(S_1, S_2))$ is the numerator of $W_{HPC}(S_1, S_2)$, and $\Delta(R)$ is the net change after removing the vertices in R (i.e., the number of agreements minus disagreements that are removed). Equivalently, we want to show $\Delta(R)(n+nm) > rv(W_{HPC}(S_1, S_2))$. We first consider that the removed vertices are in $\tilde{V}$. Observe that

$$\Delta(R) \geq rm - \binom{r}{2} - r(n-r),$$

$$v(W_{HPC}(S_1, S_2)) \leq \binom{n}{2} + nm + n\binom{m}{2}.$$

This upper bound holds because the right hand side simply counts all possible '+' and '−' edges, the edges between each actual vertex and its clique, and the edges within cliques. It is therefore sufficient to show

$$rm - rn + r^2 - \binom{r}{2} > r\frac{\binom{n}{2} + nm + n\binom{m}{2}}{n + nm}.$$

After some manipulations and relaxing the condition to remove the dependence on r, we arrive at the following sufficient condition:

$$(m-n)(n+nm) > \binom{n}{2} + nm + n\binom{m}{2},$$

which holds for $m > 3n$. The case where the removed vertices are not in $\tilde{V}$ can be analyzed in the same manner. We have shown that we can reduce an instance of 2CC to a polynomially-sized instance of 2PC. □

Tight example for Random-Eigensign**.** We consider a complete graph where all edges are positive, except for one Hamiltonian cycle comprised of negative edges. Without loss of generality, we can order the vertices so that the adjacency matrix is

$$A = \begin{pmatrix} 0 & -1 & 1 & \dots & 1 & -1 \\ -1 & 0 & -1 & 1 & \dots & 1 \\ & & \vdots & \vdots & & \\ 1 & 1 & \dots & -1 & 0 & -1 \\ -1 & 1 & \dots & 1 & -1 & 0\dots \end{pmatrix}.$$

That is, matrix A is comprised entirely of ones, save for the subdiagonal and superdiagonal entries, which are -1, and $A_{n1} = A_{1n} = -1$. It is easy to see that a constant vector $\mathbf{v}$, i.e., satisfying $v_i = v_j$ is an eigenvector of eigenvalue $n - 5$. Since $\sum_i \lambda_i^2(A) = \|A\|_F^2 = n(n-1)$, the eigenvalue $n - 5$ will be the largest if

$$\frac{n(n-1)}{2} < (n-5)^2,$$

which holds for $n > 16$. Note that $\sqrt{n}\mathbf{v}$ is a feasible solution for 2PC. We now show that Random-Eigensign attains a value of $\Theta(\sqrt{n})$.

We first rely on Equality (1) to obtain the following:

$$\begin{aligned} &\mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] \\ &= \sum_{k=1}^{n} \frac{1}{k} \sum_{i \neq j} A_{ij} s_{ij} Pr(x_i = 1, x_j = 1) Pr(\mathbf{x}^T\mathbf{x} = k | x_i = 1, x_j = 1) \\ &= \sum_{k=1}^{n} \frac{1}{k} \sum_{i \neq j} A_{ij} v_i v_j Pr(\mathbf{x}^T\mathbf{x} = k | x_i = 1, x_j = 1) \end{aligned}$$

Now, observe that given k, $Pr(\mathbf{x}^T\mathbf{x} = k | x_h = 1, x_l = 1)$ is constant for all $i \neq j$. Thus, for arbitrary h, l,

$$\begin{aligned} \mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] &= \sum_{k=1}^{n} \frac{1}{k} Pr(\mathbf{x}^T\mathbf{x} = k | x_h = 1, x_l = 1) \sum_{i \neq j} A_{ij} v_i v_j \\ &= (n-5)\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}} \middle| x_h = 1, x_l = 1\right]. \end{aligned} \quad (2)$$

Observe that when all entries of $\mathbf{v}$ are equal in absolute value, $\mathbf{x}^T\mathbf{x}$ is a binomial variable with parameters $(n, |v_i|) = (n, 1/\sqrt{n})$. Thus, by Jensen's inequality we have

$$\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}} \middle| x_h = 1, x_l = 1\right] \geq \frac{1}{\mathbb{E}\left[\mathbf{x}^T\mathbf{x} | x_h = 1, x_l = 1\right]} = \Omega(1/\sqrt{n}).$$

Furthermore, it is known [17] that

$$\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}} \middle| x_h = 1, x_l = 1\right] = O(1/\sqrt{n}).$$

That is,

$$\mathbb{E}\left[\frac{1}{\mathbf{x}^T\mathbf{x}} \middle| x_h = 1, x_l = 1\right] = \Theta(1/\sqrt{n}).$$

Combining this with Equality (2) we get

$$\mathbb{E}\left[\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right] = \Theta(\sqrt{n}) = \Theta\left(\frac{OPT}{\sqrt{n}}\right).$$

Acknowledgments. Francesco Bonchi acknowledges support from Intesa Sanpaolo Innovation Center. Aristides Gionis and Bruno Ordozgoiti were supported by three Academy of Finland projects (286211, 313927, and 317085), and the EC H2020RIA project "SoBigData" (654024).