A Spectral Algorithm for Computing Social Balance

Evimaria Terzi[1] and Marco Winkler[2,⋆]

[1] Boston University, USA
evimaria@cs.bu.edu
[2] Julius Maximilians Universität Würzburg, Germany
marco.winkler@physik.uni-wuerzburg.de

Abstract. We consider social networks in which links are associated with a sign; a positive (negative) sign indicates friendship (animosity) between the connected nodes. Recent work studies such large online signed networks by applying theories that stem from the notion of *social balance*. Computing the social balance of a signed network requires counting the distinct configurations of the signed edges within all possible triangles that appear in the network. A naive algorithm for such counting would require time that is cubic to the total number of nodes; such an algorithm is infeasible for large signed networks that are generated from online applications.

In this paper, we present an efficient spectral algorithm that computes approximate counts of the signed-triangle configurations. The essence of the algorithm lies in associating the eigenvalues of the adjacency matrix of a signed network with its signed-triangle configurations. Our experiments demonstrate that our algorithm introduces only a small error in the computed quantities while, at the same time, it achieves significant computational speedups.

1 Introduction

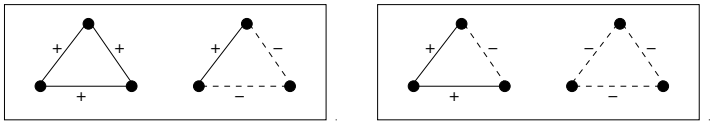
The interplay between positive and negative interactions between people defines the smooth functioning of society. In a similar way, in the online world, positive and negative relationships between users play a significant role in the function and evolution of online social networks. For example, users on Wikipedia can vote for or against the nomination of others as admins [4]. Users on Q&A systems (e.g., Yahoo! answers) get promoted by other users if they answer questions correctly; otherwise they get demoted. Users on Epinions can express trust or distrust of others [9,18]; participants on Slashdot can declare others to be either friends or foes [3,13,14]. Even the links between blogposts of different bloggers can be positive when the one blogger endorses the statements of the other or negative if the users express difference in opinions.

The focus of this paper is on computational problems that arise in *signed networks*. In such networks, every link is annotated with either a positive or

⋆ This was done while the author was visiting Boston University.

A. Frieze, P. Horn, and P. Prałat (Eds.): WAW 2011, LNCS 6732, pp. 1–13, 2011.

a negative sign. The positive links represent friendship while the negative ones represent antagonism. A basic tool in the study of signed networks is the notion of *social balance* also referred to as *structural balance* [11]. The signed triangles of a network are the very basic entities to be considered in social-balance studies. Up to node permutation, there are four possible configurations of a signed triangle. These configurations are shown in Fig. 1. The two triangles with an odd number of plus edges are *balanced*(Fig. 1(a)): they satisfy the adages that "the enemy of my enemy is my friend" and "the friend of my enemy is my enemy". The two triangles with an even number of plus edges (Fig. 1(b)) do not comply with this logic of friendship, and are considered *unbalanced*. For balanced triangles, the product of the edge signs is positive; for unbalanced triangles the same product has a negative sign. This definition of balanced and unbalanced triangles has its origins in social psychology studies [5,11] and it has been used as a basis for studying the dynamics of network formation [1] as well as the behavior of users in online social networks [15,16].



(a) Balanced triangles (b) Unbalanced triangles

Fig. 1. Balanced and unbalanced configurations of signed triangles

Given an undirected and signed graph $G = (V, E)$, a very natural question, also associated with social-balance studies, is to compute the fraction of triangles in G that are balanced. If we denote this quantity by $\widehat{b}$, then, the fraction of unbalanced triangles is $\widehat{u} = 1 - \widehat{b}$.

A naive algorithm for computing $\widehat{b}$ in a given graph counts all possible triangles in the graph and finds the ones that are balanced. For a graph with n nodes, such an algorithm goes over all the possible $\binom{n}{3}$ triangles of the graph and requires time $\mathcal{O}(n^3)$. This cubic running time makes this algorithm impractical for large datasets.

The contribution of this paper is to associate the quantity $\widehat{b}$ with the eigenvalues of the network's adjacency matrix. This connection allows us to develop a spectral algorithm that can efficiently approximate the fraction of balanced triangles. Our results highlight the relationship between the structural balance of a network and its underlying structure, as expressed in the eigenvalue decomposition of the matrix representation of the network. Although our methodology applies to arbitrary networks, we present our results for complete graphs first. Then, we show how they generalize to incomplete graphs. In incomplete graphs, apart from the existing edges that are all signed by a "+" or a "-" sign, there are also non-existing (i.e., *neutral*) edges. All signed networks coming from online social networks are incomplete; in such large datasets it is unrealistic to assume

that there is information about the positive or negative relationships between all pairs of nodes. Our experiments show that algorithm works extremely well in practice. As a by-product of our experimental evaluation we get a study of the eigenvalue spectrum of the adjacency matrices of signed networks.

The rest of the paper is organized as follows: in Section 2 we review work related to social-balance studies and spans scientific areas ranging from sociology and physics to computer science. We give some basic definitions and describe some computational tools in Section 3. In Section 4 we describe our methodology for the special case of complete graphs; we extend it for incomplete graphs in Section 5. Our experimental evaluation is given in Section 6 and we conclude the paper in Section 7.

2 Related Work

The principles underlying structural balance are based on theories in social psychology, and they date back to the work of Heider in the 1940s [11,12]. Cartwright and Harary [5,6,10] generalized these principles and expressed them in terms of graphs in the 1950s. All these studies focus on building psychological and sociological models for social balance and, naturally, they do not consider the computational aspects of these models.

More recent work focuses on mathematical models, that attempt to capture how the structural balance of a network can evolve from dynamic changes to the links' signs over time [1,17]. This line of work mostly focuses on questions related to the evolution of friendships and antagonisms between the nodes and does not consider the algorithmic problems related to structural balance.

Analysis of *online* signed social networks, through the lenses of structural-balance theories, has only appeared recently in the work of Leskovec et. al. [15,16]. More specifically, Leskovec et. al. have analyzed theories of balance and status in the context of social-media sites, investigating the extent to which each theory helped explain the linking behavior of users to these sites [16]. In a more recent work, the same authors developed methods for predicting the sign of the edge between users in signed online social networks [15]. Our work is complementary to this; we provide algorithmic tools for computing basic quantities associated with social balance. Such computations will allow the techniques developed in the past to become more efficient.

The idea of associating the triangles of a network with the eigenvalues of its adjacency matrix is not new: it has recently appeared in the work of Tsourakakis et. al. [19,20]. However, Tsourakakis et. al. focus on unsigned networks and on the computation of the total number of distinct (unsigned) triangles in them. On the other hand, we consider signed networks and signed-triangle configurations, and our goal is to count the proportion of balanced and unbalanced triangles in such networks. In that respect, our task is more complicated and the techniques developed by Trourakakis et. al. do not directly apply to our setting.

3 Basics

For the rest of the paper we assume a social network $G = (V, E)$ consisting of a set of n nodes, i.e., $|V| = n$. The edges of the network are *undirected* and *signed*. Each existing edge $\{i, j\} \in E$ is labeled with either a *plus* or a *minus* sign. A plus sign corresponds to feelings of friendship, while a minus sign indicates animosity between nodes i and j. Not all edges need to be present; a non-existing edge between two nodes corresponds to "neutral" feelings.

Given a graph $G = (V, E)$, we use $\mathbf{A}$ to represent the $n \times n$ *adjacency* matrix of G. If edge $\{i, j\}$ is signed with a "plus" ("minus"), then $\mathbf{A}_{ij} = 1$ ($\mathbf{A}_{ij} = -1$). If edge $\{i, j\}$ does not exist, then $\mathbf{A}_{ij} = 0$. Since the graph G is undirected, matrix $\mathbf{A}$ is *symmetric.* In addition to the adjacency matrix $\mathbf{A}$, we also define the *connectivity* matrix of G, which we denote by $\mathbf{G}$. Matrix $\mathbf{G}$ is also symmetric and $\mathbf{G}_{ij} = 1$ if nodes i and j are connected by an edge (irrespective of the sign); otherwise $\mathbf{G}_{ij} = 0$.

Following traditional linear-algebra notation, we use trace $(\mathbf{A})$ (trace $(\mathbf{G})$) to denote the sum of the diagonal elements of $\mathbf{A}$ ($\mathbf{G}$). That is, trace $(\mathbf{A}) = \sum_{i=1}^{n} \mathbf{A}_{ii}$ and trace $(\mathbf{G}) = \sum_{i=1}^{n} \mathbf{G}_{ii}$.

As shown in Fig. 1, there are four possible configurations of signed triangles (up to node permutation). In the theory of social balance, the triangle configurations with even number of minus signs are considered as *balanced* (Fig. 1(a)). On the other hand, the two configurations with odd number of minus signs correspond to *unbalanced* triangles (Fig. 1(b)). Notice that the product of the edge signs is positive for a balanced triangle and negative for an unbalanced one. If b and u denote the number of balanced and unbalanced triangles in G, we are interested in computing the following two quantities: (1) the *fraction of balanced triangles in* G, denoted by $\widehat{b}$ and (2) the *fraction of unbalanced triangles in* G, denoted by $\widehat{u}$. Since $\widehat{b} + \widehat{u} = 1$, computing $\widehat{b}$ suffices to achieve our goal.

Given a signed graph $G = (V, E)$ with adjacency matrix $\mathbf{A}$ and connectivity matrix $\mathbf{G}$ we know the following:

Proposition 1 ([19,20]). *The number of distinct triangles that node i participates in is $\delta_i = \frac{1}{2}\mathbf{G}^3_{ii}$.*

Furthermore, we can associate the diagonal elements of $\mathbf{A}^3$ with the balanced and unbalanced triangles a single node participates in. The following proposition summarizes this relationship.

Proposition 2. *For a node i, let b_i be the number of balanced and u_i the number of unbalanced triangles i participates in. Then, $\frac{1}{2}\mathbf{A}^3_{ii} = (b_i - u_i)$.*

Proof. Consider the expansion of the diagonal elements of $\mathbf{A}^3$

$$\mathbf{A}^3_{ii} = \sum_{j=1}^{n}\sum_{k=1}^{n} \mathbf{A}_{ij}\mathbf{A}_{jk}\mathbf{A}_{ki}. \tag{1}$$

For balanced triangles, the product of the edge signs is positive, whereas for unbalanced ones it is negative. For the case that any of the edges is missing

(and therefore the triangle does not exist), the product yields zero. Therefore, the sum over all possible configurations, i.e., the value of $\mathbf{A}^3_{ii}$, is increased by one if the triangle of a path $i \to j \to k$ is balanced. It is decreased by one, if it is unbalanced. However, since the graph is undirected, each unique triangle is counted twice in $\mathbf{A}^3_{ii}$; again, triangle $\{i, j, k\}$ is counted both as $i \to j \to k$ and as $i \to k \to j$. Consequently we have to divide by two.

Propositions 1 and 2 will become useful in proving the main theorems of the paper in Sections 4 and 5. We will also use the following results related to the eigenvalue decomposition of any real symmetric matrix $\mathbf{X}$[1]. First, recall that every real, symmetric $n \times n$ graph has n real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. We also know the following:

Proposition 3 ([8]). *For any $n \times n$ matrix $\mathbf{X}$, the sum of the diagonal elements of $\mathbf{X}$ is equal to the sum of its eigenvalues. That is, if $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $\mathbf{X}$, then we have that trace$(\mathbf{X}) = \sum_{i=1}^{n} \lambda_i$.*

Proposition 4 ([8]). *Let $\mathbf{X}$ be a real, $n \times n$ and symmetric matrix with eigenvalue decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and diag$(\mathbf{\Lambda}) = (\lambda_1, \ldots, \lambda_n)$. Then, for integer k, the eigenvalue decomposition of matrix $\mathbf{X}^k$ is $\mathbf{X}^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$. Where $\mathbf{\Lambda}^k$ is a diagonal matrix with diag$(\mathbf{\Lambda}) = (\lambda_1^k, \ldots, \lambda_n^k)$.*

4 Balanced Triangles in Complete Networks

In this section, we show how the fraction of balanced triangles $\widehat{b}$ of a fully connected signed network can be expressed as a function of the eigenvalues of the network's adjacency matrix. We express this relationship formally in Theorem 1.

Theorem 1. *Let $G = (V, E)$ be a fully connected signed network with adjacency matrix $\mathbf{A}$. If $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $\mathbf{A}$, then the fraction of balanced triangles $\widehat{b}$ of G can be computed as follows:*

$$\widehat{b} = \frac{1}{2} + \frac{1}{4n(n-1)(n-2)} \sum_{i=1}^{n} \lambda_i^3. \tag{2}$$

Before getting into the details of the proof, observe that in a fully-connected signed graph all nodes participate in the same number of signed triangles. That is, we have the following fact.

Fact 1. *Every node $i \in V$ participates in the same number of distinct triangles. That is, if we denote this number by δ_i, we have that $\delta_1 = \delta_2 = \ldots = \delta_n = \delta$ and $\delta = \binom{n-1}{2}$.*

[1] Recall that both the adjacency and the connectivity matrices $\mathbf{A}$ and $\mathbf{G}$ are real and symmetric.

This is simply because every node participates in triangles formed by all combinations of pairs of the remaining $n-1$ nodes (excluding the node itself).

For every node i, the δ triangles it participates in can be partitioned into b_i balanced ones and u_i unbalanced ones. That is,

$$\delta = b_i + u_i \ , \forall i \in \{1, 2, ..., n\}\,. \tag{3}$$

Due to Proposition 2, we have that

$$\frac{\mathbf{A}^3_{ii}}{2} = b_i - u_i \tag{4}$$

By (3) and (4), we obtain that

$$b_i = \frac{1}{2}\delta + \frac{1}{4}\mathbf{A}^3_{ii} \text{ and } u_i = \frac{1}{2}\delta - \frac{1}{4}\mathbf{A}^3_{ii}.$$

Therefore, the total number of balanced and unbalanced triangles, denoted by b and u, will be

$$b = \frac{1}{3}\sum_{i=1}^{n} b_i = \frac{1}{3}\left[\frac{1}{2}n\delta + \frac{1}{4}\text{trace}\left(\mathbf{A}^3\right)\right] \text{ and } u = \frac{1}{3}\sum_{i=1}^{n} u_i = \frac{1}{3}\left[\frac{1}{2}n\delta - \frac{1}{4}\text{trace}\left(\mathbf{A}^3\right)\right].$$

We divide both quantities by 3 because every triangle has three participating nodes, and thus, is triple-counted. Consequently, the fraction of balanced triangles in the network, $\widehat{b}$, is

$$\widehat{b} = \frac{b}{b+u} = \frac{1}{4n\delta}\text{trace}\left(\mathbf{A}^3\right) + \frac{1}{2} = \frac{1}{4n(n-1)(n-2)}\text{trace}\left(\mathbf{A}^3\right) + \frac{1}{2}.$$

The last derivation is due to the fact that $\delta = \binom{n-1}{2}$. Using Propositions 3 and 4, we get the desired result.

Algorithmic implications: Equation (2) implies that we can compute the fraction of balanced triangles by simply computing the eigenvalues of $\mathbf{A}$ and take their third power (this is due to Proposition 4). However, computing all the n eigenvalues of $\mathbf{A}$ still requires time $\mathcal{O}(n^3)$. Fortunately, if one is content with approximations of $\widehat{b}$, Theorem 1 can lead to an algorithm with significantly smaller running time. Assume the permutation of the eigenvalues of $\mathbf{A}$ (i.e., $\lambda_1, \ldots, \lambda_n$) in decreasing order of their magnitude. Then, instead of using all the n eigenvalues in the computation of $\widehat{b}$, we can compute $\widehat{b}_k$ by only using the top-k eigenvalues with the largest magnitude. Therefore, computing these eigenvalues requires time only $\mathcal{O}(n^2k)$ – this is because the standard algorithms for computing eigenvalue decomposition output the eigenvalues sequentially and in decreasing order of their magnitude [8]. For small values of k, the savings in terms of running time can be significant. Our algorithm can further leverage the efficient approximation algorithms for computing eigenvalues of large matrices [7]. Our experiments demonstrate that very small values of k (compared

to n) suffice to give significant computational gains with insignificant accuracy loses. This is because, as our experiments demonstrate, the effective rank of $\mathbf{A}$ is small.

5 Balanced Triangles in Arbitrary Networks

Here we show that even when the signed graph is not a clique, we can express the fraction of balanced triangles as a function of the eigenvalues of the adjacency matrix $\mathbf{A}$ and the connectivity matrix $\mathbf{G}$.

Theorem 2. *Let $G = (V, E)$ be a symmetric signed network (not necessarily fully connected) with adjacency matrix $\mathbf{A}$ and connectivity matrix $\mathbf{G}$. Also let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of $\mathbf{A}$ and $\mu_1, \mu_2, \ldots, \mu_n$ the eigenvalues of $\mathbf{G}$. Then, the fraction of balanced triangles in G can be expressed as follows:*

$$\widehat{b} = \frac{1}{2}\left(1 + \frac{\sum_{i=1}^{n} \lambda_i^3}{\sum_{i=1}^{n} \mu_i^3}\right). \tag{5}$$

The main idea of the proof of Theorem 2 is similar to that of Theorem 1. The only difference is that now, the number of triangles δ_i that a node i participates in is different for every node i and can be computed using Proposition 1. If b_i and u_i denote the number of balanced and unbalanced triangle that node i participates in, then we have that

$$\delta_i = b_i + u_i. \tag{6}$$

Every balanced triangle contributes a (+1) (all three edges positive or two edges negative and one positive). Thus, the diagonal entries of $\mathbf{A}^3$ are numbers of the form

$$\frac{\mathbf{A}^3_{ii}}{2} = b_i - u_i. \tag{7}$$

Using Proposition 1 and Equations (6) and (7), we get

$$b_i = \frac{1}{4}\left(\mathbf{G}^3_{ii} + \mathbf{A}^3_{ii}\right) \text{ and } u_i = \frac{1}{4}\left(\mathbf{G}^3_{ii} - \mathbf{A}^3_{ii}\right).$$

The total numbers of balanced and unbalanced triangles of the graph, denoted by b and u respectively, are then given by

$$b = \frac{1}{3}\sum_{i=1}^{n} b_i = \frac{1}{12}\left[\text{trace}\left(\mathbf{G}^3\right) + \text{trace}\left(\mathbf{A}^3\right)\right]$$

$$u = \frac{1}{3}\sum_{i=1}^{n} u_i = \frac{1}{12}\left[\text{trace}\left(\mathbf{G}^3\right) - \text{trace}\left(\mathbf{A}^3\right)\right].$$

As before, the division by 3 is due to the fact, that the summation goes over all nodes in the network, what results in a triple-counting. It follows that the fraction of balanced triangles in the network, is

$$\widehat{b} = \frac{b}{b+u} = \frac{1}{2}\left(1 + \frac{\text{trace}\left(\mathbf{A}^3\right)}{\text{trace}\left(\mathbf{G}^3\right)}\right).$$

Using Propositions 3 and 4, we can replace the quantities $\text{trace}\left(\mathbf{A}^3\right)$ and $\text{trace}\left(\mathbf{G}^3\right)$ by $\sum_{i=1}^{n} \lambda_i^3$ and $\sum_{i=1}^{n} \mu_i^3$ respectively, and get the desired result.

Algorithmic implications: Equation (2) implies that we can compute the fraction of balanced triangles by computing the eigenvalues of $\mathbf{A}$ and take their third power (this is due to Proposition 4). However, computing all the n eigenvalues of $\mathbf{A}$ still requires time $\mathcal{O}(n^3)$. Fortunately, if one is content with approximations of $\widehat{b}$, Theorem 2 can lead to an algorithm with significantly smaller running time. Instead of computing all the n eigenvalues of $\mathbf{A}$ and $\mathbf{G}$, we can evaluate Equation (5) using only the k largest-magnitude eigenvalues of matrices $\mathbf{A}$ and $\mathbf{G}$. We call the algorithm that uses Theorem 2 and the first k eigenvalues of $\mathbf{G}$ and $\mathbf{A}$ for computing $\widehat{b}$, the SPECTRAL(k) algorithm. For small values of k, the running time of SPECTRAL(k) is very small. Further computational improvements can be achieved by leveraging more efficient approximation algorithms for eigenvalue computations [7]. Our experiments demonstrate that the values of k that give satisfactory approximations of $\widehat{b}$ are very small because, as our experiments show, the effective ranks of $\mathbf{A}$ and $\mathbf{G}$ are both small and similar.

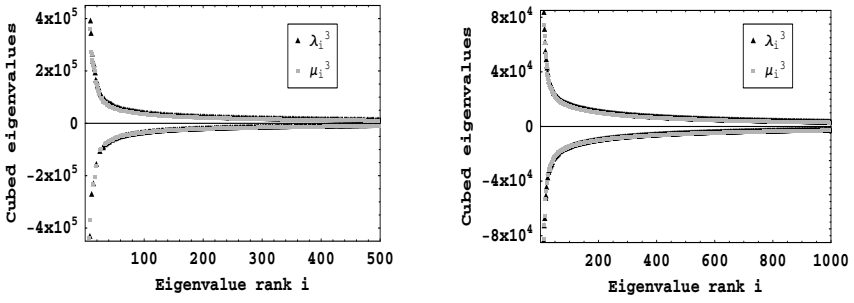
6 Experiments

Our experiments both with real and synthetic datasets demonstrate that a very small number of eigenvalues of the adjacency and the connectivity matrix suffices to provide accurate approximations of $\widehat{b}$.

Datasets: For our experiments we use two real signed-graph datasets: **Epinions** and **Slashdot**. We have downloaded both datasets from `http://snap.stanford.edu/data/`[2]. Since both datasets were directed, we obtained their undirected versions as follows: For every directed edge $e(i \to j)$ such that $e(j \to i)$ does not exist, we create an undirected edge that retains the same sign as the original directed edge. If both $e(i \to j)$ and $e(j \to i)$ exist and have the same sign, we again, created undirected edge between i and j with the same sign. If edges $e(i \to j)$ and $e(j \to i)$ have contradicting signs, we leave nodes i and j to be disconnected. The resulting **Epinions** dataset has $n = 131,828$ nodes and $|E| = 711,210$ undirected edges; 83% of these edges are positive. The resulting **Slashdot** dataset contains $n = 77,357$ nodes and $|E| = 468,554$ edges; 76% of these edges are signed positive.

We also generate synthetic scale-free (**SF**) graphs using the generative model proposed by Barabási and Albert [2]. We convert the generated unsigned graphs into signed graphs by randomly assigning a "+' (or a "-") sign to every edge of the graph with probability equal to p_+. The value of p_+ is a parameter that we vary for our experimental evaluation.

[2] We used the file *soc-sign-Slashdot081106.txt.gz* for Slashdot.



(a) **Epinions** dataset (b) **Slashdot** dataset

Fig. 2. Distribution of the cubed eigenvalues λ_i^3 and μ_i^3 of the adjacency matrix $\mathbf{A}$ and the connectivity matrix $\mathbf{G}$ for the Epinions and the Slashdot graph, ordered by magnitude in decreasing order

6.1 Evaluating Spectral on Real-World Signed Networks

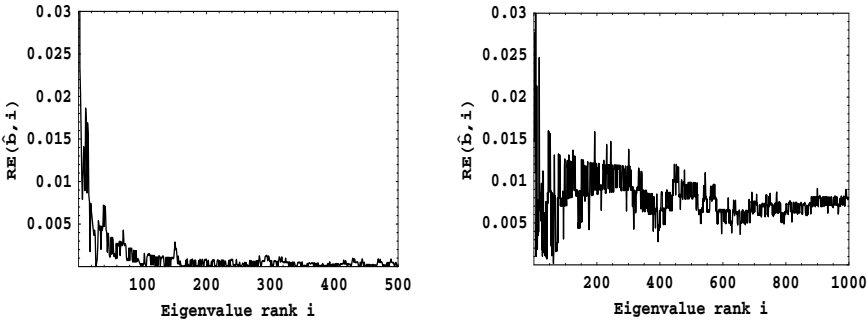
Recall, that Theorems 1 and 2 indicate that, in order to compute $\widehat{b}$, it suffices to know the sum of the cubed eigenvalues of the adjacency ($\mathbf{A}$) and the connectivity ($\mathbf{G}$) matrices. Since our objective is to approximate these sums by considering only some of the eigenvalues, we need to show that the first eigenvalues contribute the major part to the total sum. Therefore, as a first experiment, we examine the distributions of the cubed eigenvalues of $\mathbf{A}$ and $\mathbf{G}$ for the **Epinions** and **Slashdot** datasets. Figures 2(a) and 2(b) show the result for the first cubed eigenvalues of the **Epinions** and **Slashdot** graphs, ordered by magnitude. There are two crucial features, which justify the approximation of the sum over all eigenvalues by the ones with the largest magnitudes. First, the distribution of the magnitude versus the rank is highly skewed and therefore the early summands contribute the most. Notice that the skew is identical for the eigenvalues of $\mathbf{A}^3$ and $\mathbf{G}^3$ matrices, meaning that both matrices have the same effective rank. Second, the signs of the eigenvalues switch between positive and negative. Therefore, many of the eigenvalues in the tail cancel each other out.

To quantify how well the actual result of $\widehat{b}$ can be approximated, we evaluate the relative error

$$\text{Re}\left(\widehat{b}, i\right) = \frac{|\widehat{b}_i - \widehat{b}|}{\widehat{b}}. \tag{8}$$

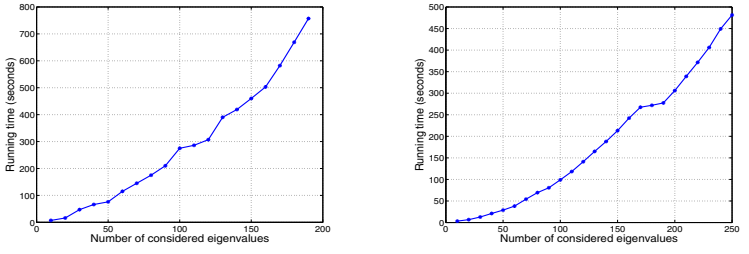
The term $\widehat{b}_i$ is the result of Spectral that considers the first i eigenvalues of $\mathbf{A}$ and $\mathbf{G}$. The exact result $\widehat{b}$ is obtained by counting all the existing triangles using the exact cubic algorithm that we call Naive.

The relative error of the output of Spectral, as a function of the number of considered eigenvalues, i, for the **Epinions** and the **Slashdot** datasets, are shown in Figures 3(a) and 3(b) respectively. For the **Epinions** data, taking into account 10, out of a total of 131,828 eigenvalues, gives an approximation with



(a) **Epinions** dataset (b) **Slashdot** dataset

Fig. 3. Relative error for the approximation of $\widehat{b}$ obtained by SPECTRAL for the **Epinions** and the **Slashdot** datasets, when considering only the i eigenvalues with the largest magnitude



(a) **Epinions** dataset (b) **Slashdot** dataset

Fig. 4. Running time (in seconds) of SPECTRAL as a function of the number of considered eigenvalues

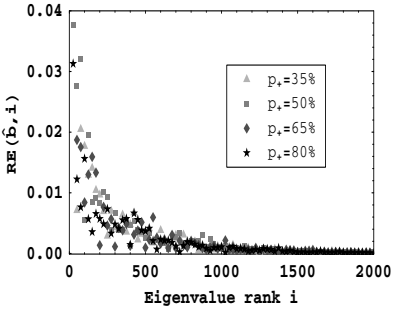
relative error less than 2%. Similarly, for the **Slashdot** data, taking into account ≈ 25 eigenvalues, out of the $77,357$ total eigenvalues, leads to an approximation that has less than 2% relative error.

Figures 4(a) and 4(b) show the running time (in seconds) of the SPECTRAL algorithm on **Epinions** and **Slashdot** data, as a function of the number of considered eigenvalues of $\mathbf{G}$ and $\mathbf{A}$. One can see, that the running time of the algorithm scales almost linearly to the number of eigenvalues. In numbers: for the **Epinions** dataset, NAIVE needs 753 seconds for exact computation, while SPECTRAL gives an estimate with relative error less than 0.9% in 7 seconds. For the **Slashdot** dataset, SPECTRAL with only 10 eigenvalues runs in just 3 seconds and provides a solution with relative error less than 0.7%. Compare this to the running time of NAIVE that is 262 seconds for the same dataset. Note that although the error is not necessarily dropping as the number of

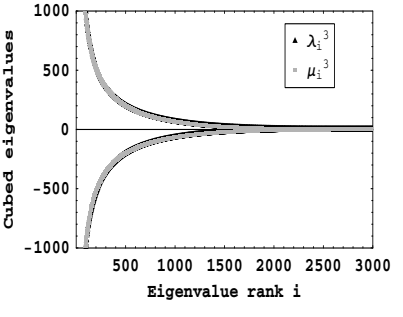
considered eigenvalues increases, it never exceeds 1%. Since more eigenvalues increase the running time of SPECTRAL, using the top-10 magnitude eigenvalues is a reasonable rule of thumb.[3] Neither of our implementations were optimized for run-time experiments, however, the times we report here are indicative of the computational savings achieved by the SPECTRAL algorithm.

6.2 Evaluating SPECTRAL on Synthetic Signed Networks

Here, we further investigate the accuracy of SPECTRAL for signed networks that have different underlying topologies. For that, we generate **SF** graphs with $n = 3000$ nodes and $|E| = 30,000$ edges. We also examine the impact of the parameter p_+ on the overall performance of the algorithm. Figure 5(a) shows the relative error obtained by SPECTRAL, as a function of the number of considered eigenvalues. Every point is an average over four different network realizations with the same set of generation parameters. As we can see, p_+ hardly affects the accuracy of the approximation. Consideration of approximately 100 eigenvalues (out of the total of $3,000$) leads to an error of less than 2%. In Figure 5(b), we observe that the eigenvalues of a **SF** graph with $p_+ = 80\%$ exhibit the same two crucial properties that were pointed out in real graphs. That is, the magnitude of the eigenvalues are highly skewed. This skew is almost identical for both **G** and **A** matrices and similar both for eigenvalues with positive and negative signs. It turns out that, for **SF** graphs, the shape of these distributions is independent of the value of p_+.



(a) Relative error of the approximation of $\widehat{b}$ obtained by SPECTRAL for $n = 3000$ and $p_+ \in \{35\%, 50\%, 65\%, 80\%\}$, as a function of the number of eigenvalues considered.



(b) Distribution of the cubed eigenvalues λ_i^3 and μ_i^3 of matrices **A** and **G**; values ordered by magnitude in decreasing order and $p_+ = 80\%$

Fig. 5. Experiments with Scale-free graphs

[3] We implemented SPECTRAL using Mathematica and NAIVE using Java. The running times of NAIVE reported here are generous, since we rearrange the input so that it allows NAIVE to do efficient in-memory computations.

7 Conclusions

Given a signed undirected network, we have presented a spectral algorithm for computing the fraction of balanced triangles in the network. The basis of our algorithm lies in the dependency between the the fraction of balanced triangles of the network and the eigenvalues of the adjacency and the connectivity matrices that describe the signed network. After establishing the form of this connection theoretically, we have exploited it in order to device Spectral, a fast algorithm that approximates the fraction of the balanced triangles in the network. In an extensive experimental evaluation, both on real and generated signed networks, we have demonstrated that the fraction of balanced and unbalanced triangles can be approximated very efficiently using our algorithm. In addition to that, our experiments also revealed certain interesting properties of the eigenvalues of the adjacency and the connectivity matrices of signed networks. We view our methodology as a useful computational tool for the studies of structural balance in large online signed social networks.

References

1. Antal, T., Krapivsky, P.L., Redner, S.: Social balance on networks: The dynamics of friendship and enmity. Physica D 130 (2006)
2. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286 (1999)
3. Brzozowski, M.J., Hogg, T., Szabá, G.: Friends and foes: ideological social networking. In: Human Factors in Computing Systems, CHI (2008)
4. Burke, M., Kraut, R.: Mopping up: modeling wikipedia promotion decisions. In: ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 27–36. ACM Press, New York (2008)
5. Cartwright, D., Harary, F.: Structure balance: A generalization of heider's theory. Psychological Review 63(5), 277–293 (1956)
6. Davis, J.A.: Structural balance, mechanical solidarity, and interpersonal relations. American Journal of Sociology 68, 444–462 (1963)
7. Drineas, P., Kannan, R., Mahoney, M.W.: Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. SIAM J. Comput. 36(1), 158–183 (2006)
8. Golub, G., Loan, C.V.: Matrix Computations. Johns Hopkins Press, Baltimore (1989)
9. Guha, R.V., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: International Conferece on World Wide Web (WWW), pp. 403–412 (2004)
10. Harary, F.: On the notion of balance of a signed graph. Michigan Math. Journal 2(2), 143–146 (1953)
11. Heider, F.: Social perception and phenomenal causality. Psychological Rev. 51(6), 358–374 (1944)
12. Heider, F.: Attitudes and cognitive organization. Journal of Psychology 21, 107–112 (1946)
13. Kunegis, J., Lommatzsch, A., Bauckhage, C.: The slashdot zoo: Mining a social network with negative edges. In: International Conference on World Wide Web (WWW), pp. 741–750 (2009)

14. Lampe, C., Johnston, E., Resnick, P.: Follow the reader: filtering comments on slashdot. In: Human Factors in Computing Systems, CHI (2007)
15. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: International CIonference on World Wide Web (WWW), pp. 641–650 (2010)
16. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: Human Factors in Computing Systems (CHI), pp. 1361–1370 (2010)
17. Marvel, S., Kleinberg, J., Strogatz, S.: The energy landscape of social balance. Physical Review Letters 103(19) (2009)
18. Massa, P., Avesani, P.: Controversial users demand local trust metrics: an experimental study on epinions.com community. In: National Conference on Artificial Intelligence (AAAI), pp. 121–126 (2005)
19. Tsourakakis, C.E.: Fast counting of triangles in large real networks without counting: Algorithms and laws. In: International Conference on Data Mining (ICDM), pp. 608–617 (2008)
20. Tsourakakis, C.E., Drineas, P., Michelakis, E., Koutis, I., Faloutsos, C.: Spectral counting of triangles in power-law networks via element-wise sparsification. In: International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 66–71 (2009)