Characterizing attitudinal network graphs through frustration cloud

Lucas Rusnak[1] · Jelena Tešić[1]

Abstract

Attitudinal network graphs are signed graphs where edges capture an expressed opinion; two vertices connected by an edge can be agreeable (positive) or antagonistic (negative). A signed graph is called balanced if each of its cycles includes an even number of negative edges. Balance is often characterized by the frustration index or by finding a single convergent balanced state of network consensus. In this paper, we propose to expand the measures of consensus from a single balanced state associated with the frustration index to the set of nearest balanced states. We introduce the *frustration cloud* as a set of all nearest balanced states and use a graph-balancing algorithm to find all nearest balanced states in a deterministic way. Computational concerns are addressed by measuring consensus probabilistically, and we introduce new vertex and edge metrics to quantify *status*, *agreement*, and *influence*. We also introduce a new global measure of controversy for a given signed graph and show that vertex status is a zero-sum game in the signed network. We propose an efficient scalable algorithm for calculating frustration cloud-based measures in social network and survey data of up to 80,000 vertices and half-a-million edges. We also demonstrate the power of the proposed approach to provide discriminant features for community discovery when compared to spectral clustering and to automatically identify dominant vertices and anomalous decisions in the network.

Keywords Frustration cloud · Balanced graph · Consensus · Status · Controversy · Signed graph

✉ Jelena Tešić
jtesic@txstate.edu

Lucas Rusnak
lucas.rusnak@txstate.edu

[1] Texas State University, San Marcos, TX, USA

1 Introduction

Signed graph network representations of socio-technical networks offer richer modeling of relations between people, AI agents, products, and content. Attitudes captured by edges between two vertices can be agreeable (positive) or antagonistic (negative). Some social signed graph examples include team member evaluations in a company, student evaluations of instructors, movie recommendations based on common interests, or the "trustworthiness" of a product reviewer or seller in online stores. If there is a group decision to be made, consensus or majority voting guides the decisions and the final outcome in such networks. These sentiments have tangible real-world effects, such as annual performance scores and promotions in a corporation. Graph decision algorithms are rarely scrutinized, as consensus and majority voting are established social constructs (Leskovec and Krev 2014). Only when the outcomes are known can an individual's status be perceived as elevated or diminished relative to their peers, and even then only anecdotally questioned or explained. Such algorithms' sensitivity to bias, fraud, and falsehood has been put under a magnifying glass in the last couple of years, as state-of-art research examined the controversy of decisions in cases of status quo (Li et al. 2005) or subgroup mobilization against other groups (Kumar et al. 2018). Research in the domain of consensus in signed graphs focuses on frustration index computation and algorithmic convergence to *some* balanced state (Altafini 2019; She et al. 2020) and is one dimensional, as described in Sect. 2.

In this paper, we focus on finding multiple balanced states of the signed graph, as they represent different consensus outcomes. The main contribution of this paper is that it generalizes the notion of the frustration index to the **frustration cloud**. The *frustration cloud* is a set of all balanced states obtained by a minimal number of edge sign inversions. In comparison, the *frustration index* characterizes the distance between the original signed graph and a single nearest balancing set. We also propose a spanning tree-based graph balancing algorithm that focuses on finding balanced states from spanning trees. The proposed approach works on **any** signed graph, avoids the NP-hardness of finding the frustration index, and focuses on determining a basis of fundamental cycles to produce balanced states (Rusnak 2013). A greedy approach to finding a basis of fundamental cycles is NP-hard in general, and Deo et al. proposed some polynomial-time algorithms (Deo et al. 1982). The *graphB* algorithm introduces a deterministic methodology that finds all the nearest balanced states of a signed graph. To quantify levels of agreement in the network, we sample the frustration cloud via the associated family of balanced matroidal bases (spanning trees) (Zaslavsky 1982). This statistically meaningful sampling of the frustration cloud produces a robust way to handle the brittleness of the data space for signed graph data and avoid challenges presented in Selbst et al. (2018).

The proposed spanning tree-based balancing method combines the requirement for statistical parity across the nearest balanced states with the requirement to consider *all* vertices instead of few selected ones; this method relies on the spanning trees, not random walks (Garimella et al. 2017). The sentiments are reconstructed around a spanning tree to produce a set of nearest balanced states. The resulting balanced states are generalizations of bipartite graphs (Berge 1970; Harary 1959), and the resulting negative edge cut defines two consensus-based sets. We use these consensus-based sets

to characterize the importance of specific vertices and edges necessary to produce a majority consensus: *status* measures an individual's contribution to reaching consensus over the frustration cloud; *agreement* measures the edge's contribution to belonging to the majority consensus; and *influence* measures the vertex based on its edge agreement scores.

1.1 Contribution

Social network analysis has not converged on how to assess the robustness, resilience, and reliability of the network algorithm outcomes, or how to identify anomalies in large signed network graphs. There is a clear need to measure the performance of algorithms that define outcomes, characterize consensus in social or multi-agent attitudinal networks as a unit, and assess vertex and edge contributions to graphs as a whole. We propose a process that characterizes the impact of every vertex and edge in its entirety, and it may be used outside of social network analysis on any binary decision paradigm to examine the reliability of decision-making processes relative to some given ground state. Researchers in multi-agent networks have focused on techniques to produce a *single* convergent balanced state (Alemzadeh et al. 2017; Altafini 2019; Hu and Zheng 2013; Jiang et al. 2016; Pan et al. 2016; She et al. 2020).

We propose a new discrete alternative to Laplacian dynamics, and we identify all nearest balanced states of a signed graph. Our main contribution is a novel signed graph methodology that (1) determines all the nearest balanced states via basis sampling via spanning trees, (2) quantifies the importance of each balanced state relative to the likelihood it will be become the consensus state, (3) quantifies an individual's *status* relative to their peers, (4) characterizes the *potential* maximum status of an individual over tie-break scenarios, (5) provides a constant metric of *controversy* for the entire network that is subject to a Conservation Law, (6) quantifies an individual decision or opinion based on *agreement*, (7) aggregates agreement to each individual to quantify *influence* over others, (8) compares status and influence to quantify the positive/negative relationship of the entire network and provide a spectrum of status-influence, and (9) scales the tree-based balancing algorithm to **graphB:** graph balancing using statistically significant sample of spanning trees (Tešić et al. 2020).

In the event that the sentiment data provided is related to promotions, and the outcome of those promotions are known, the research examines the efficacy of this new methodology by outcomes as status separates "promoted" from "not promoted" and identifies any outliers in either case, and where the influence separates "voters" from those "voted on". We show proof-of-concept implementation results on several large social networks and how status-influence measures of the vertex can discover contentious outcomes on Wikipedia administrator promotions. We also identify anomalous actors when outcomes are not known in the Slashdot dataset (Leskovec and Krev 2014), and we analyze the approach on a small survey dataset (Read 1954) which shows that in new status-influence space, vertices can be fully characterized in terms of community without the need to specify the number of clusters k for spectral clustering.
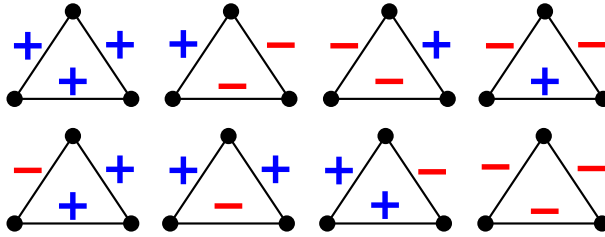
Fig. 1 Sign graph triangle where the top row are balanced states, and the bottom row are unbalanced states

2 Background and related work

In this section, we describe the state-of-art work in mathematical sociology (2.1) and signed graph frustration (2.2), and we present related work in the fields of social network analysis and control (2.3) and signed graph clustering (2.4).

2.1 Related work in mathematical sociology

A *signed graph* consists of a collection of vertices that are linked together with undirected edges; positive sentiment between two vertices is modeled as positive edge "+1", and negative sentiment between two vertices is modeled as a negative edge "−1". Fritz Heider introduced Balance Theory in 1948 (Heider 1946). Balance theory examined consensus in triadic relationships in a signed triangle graph. Figure 1 shows all possible edge signs for a signed triangle graph. These eight graphs are different as they have different edge signs between specific vertices. Each of the eight signings is a *state* of the triangle graph. *Balance Theory* is the base model of attitude change analysis among three persons in a signed graph (Heider 1946). A triangle state is considered *balanced* if the product of the edge signs is positive and *unbalanced* if the product of edge signs is negative. A balanced triadic relationship in a signed graph is captured as "the enemy of my enemy is my friend" paradigm in mathematical social modeling (Leskovec et al. 2010a).

There are four different ways to achieve a balanced state in the triangle, as depicted in Fig. 1, and we emphasize that there is *more than one* balanced state. These multiple consensus scenarios all capture different aspects of reaching consensus. A specific balanced state is a snapshot of the balanced assessment of the network, and it is *not sufficient* as all sentiments are not necessarily equal, as illustrated in Fig. 1 (top row). In this paper, we implement an algorithm that considers the collection of nearest consensus states to characterize network graph behavior. This type of analysis considers all possible consensus outcomes of attitudinal network graphs for a complete consensus characterization.

Mathematical sociology was introduced when Rashevsky characterized large social networks as graphs, where vertices are persons and edges measure the level of acquaintanceship (Rashevsky 1949). The mathematical foundation of signed graphs (Harary 1953) and social balance theory (Abelson and Rosenberg 1958; Heider 1946) introduced the concepts of modeling balance and agreement in social networks using more
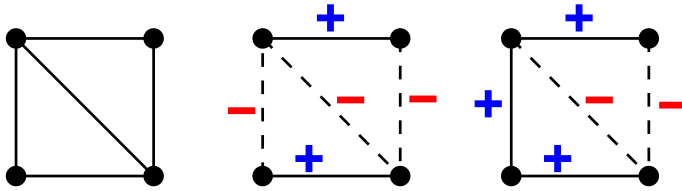
Fig. 2 Left: an underlying graph G; middle: an example balanced signing of G; right: an unbalanced signing of G, Σ. G and Σ are used as signed graph examples in the rest of the paper

complex mathematical models. Harary introduced the *frustration index* of a signed graph as a measure of how far the network graph is from a state of structural balance (Harary 1959). Harary's proposed measure is the smallest number of edges whose negation in the network graph results in a balanced signed graph. Harary's attitudinal balanced model was formalized in graph-theoretic terms (Cartwright and Harary 1956) and fully characterized by Zaslavsky (1982) in matroid-theoretic terms. Davis (1967) studied the necessary and sufficient conditions for clustering of attitudinal graphs. Mathematical sociological modeling has evolved to address sociological phenomena in various fields of social science and helps in understanding, evaluating, and predicting patterns of social relationships and interactions (Hunter et al. 1984).

2.2 Balance and frustration in signed graph

A *signed graph* Σ is a pair (G, σ) that consists of a graph $G = (V, E)$ and an edge-signing function $\sigma : E \to \{+1, -1\}$. For a set of edges E in a signed graph Σ, let E^+ (resp. E^-) denote the set of positive (resp. negative) edges of G—the signs of the edges are regarded as sentiments between two vertices. The *sign of a subgraph* is the product of the signs of the edges in that subgraph. A signed graph is *balanced* if the sign of every circle is positive (Cartwright and Harary 1956; Harary 1953). If the graph Σ is not balanced, there exists a set of edges whose sign reversal produces a balanced signed graph, and that set is called a balancing set, as shown by reversing the signed edge sign in (Fig. 2).

A balancing set is *minimal* if no proper subset is a balancing set. The *frustration index* of a signed graph Σ, denoted $Fr(\Sigma)$, is the smallest number of edges whose change in sign can result in a balanced signed graph (Harary 1959). All balanced signed graphs necessarily have a frustration of 0. The frustration index has applications in various areas including physics (Alava et al. 2001; Barahona 1982), economics (Yoshikawa et al. 2011), negative feedback loops in Boolean networks (Sontag et al. 2008), and statistical mechanics (Sethna 2006). Each balanced state represents a consensus outcome, meaning all paths between two vertices have the same sign. These concepts are related by a Theorem of Harary and motivate our proposed probabilistic consensus model to examine all the nearest balanced states.

Theorem 2.1 (Harary 1953, 1959) *For a signed graph Σ', the following are equivalent:*

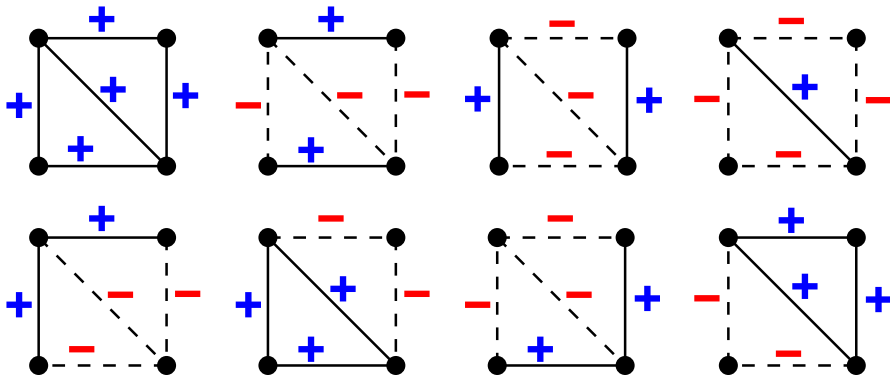1. *Σ' is balanced. (All circles are positive.)*

Fig. 3 All 8 balanced signed graphs of the underlying graph G in Fig. 2. Harary-cut of negative edges is represented by dashed edges

2. *For every vertex pair (v_i, v_j) with $v_i, v_j \in V$ all (v_i, v_j)-paths have the same sign. (Agreement or consensus)*
3. *There exists a bipartition of the vertex set into sets U and W such that an edge is negative if, and only if, it has one vertex in U and one in W. The bipartition (U, W) is called the* Harary-bipartition.*)*
4. *$Fr(\Sigma') = 0$. (0 frustration.)*

Figure 3 shows all possible balanced states Σ' on the given underlying graph G from Fig. 2 (left). Each of the balanced states satisfies all conditions in Theorem 2.1. Once balanced, the set of negative edges whose deletion produces the Harary-bipartition is called the *Harary-cut* of the balanced graph. The Harary-cuts are emphasized by representing the negative edges with dashed edges.

The frustration index is the size of the smallest Harary-cut; for example in Fig. 2, $Fr(\Sigma) = 1$. Computing the frustration index of a signed graph is an NP-hard problem. There exist scenarios that are solvable in polynomial time and for which exact large-scale solutions are possible. Wu and Chen proposed a branch-and-bound algorithm to balance signed graphs by editing edges and deleting vertices, demonstrating its efficiency over trivial and heuristic algorithms on inputs with up to $n = 40$ vertices (Wu and Chen 2013). In control of multi-agent systems, Altafini analyzed the convergence to a balanced state in the decision-making process and presented an effective way to compute the average consensus for a network with up to 100 vertices (Altafini 2019).

Aref et al. developed three binary linear programming models to compute the frustration index quickly and exactly as the solution to a global optimization problem. They demonstrated the efficiency of their techniques for inputs with up to 15,000 edges (Aref et al. 2016, 2020) with an extension that allows for the incorporation of weights in the interval $[-1, +1]$ to determine weighted minimum frustration (Aref et al. 2020). The computational complexity of Aref et al. (2016, 2020) algorithms is bounded by a polynomial function of the size of the underlying graph.

As balancing only requires the sign (sentiment) of the edge, and not the intensity (weight), we demonstrate that the balancing applications introduced in this paper produce a quantitative spectrum of vertex and edge metrics that drive balanced/consensus

outcomes. By maintaining the separation of signs and weights, as suggested by Zaslavsky (2021), we are able to preserve the matroidal structure, and our approach immediately generalizes to any edge weight value by replacing each tree with the weight-product as in Tutte (1984), which is expected in future work. Another critical difference in our approach is the relaxation on determining the frustration index (or weighted frustration index). We are producing a set of balanced states with minimal (in containment) sign alterations, instead of the minimum value (in cardinality) for frustration. We propose an approach in Sect. 3 that focuses on multiple nearest balanced states to avoid the NP-hardness of determining the frustration index while simultaneously analyzing multiple possible nearest consensus outcomes that scale with the size of social network.

2.3 Related work in social network analysis and cybernetics

Large virtual communities and decision networks of the twenty-first century initiated the explosive growth of social network analysis and network science fields. The analysis of largely digital traces of *social networks at scale* expanded well-studied mathematical algorithms for reinforcement, information processing, social judgment, balance, and dissonance (Hunter et al. 1984). Wasserman et al. introduced social network analysis as algebraic graph representations and proposed a series of statistical tests (Wasserman and Faust 1994). The domain research has focused on *predicting* the existence and/or sentiments of edges in the graph, recommending content or a product, or identifying unusual trends. Baseline signed graph theory was used to explain the relative status that individuals hold within in a social network (Leskovec et al. 2010a, b) and focused on socially-conscious science to help understand bias, controversy, conflict, and trust (Guha et al. 2004; Mishra and Bhattacharya 2011). All mathematical models in network science that model intents and trends in online social networks have relied on aspects of well-established consensus-based models in signed graph theory (Chen et al. 2018; Garimella et al. 2017; Yuan et al. 2017) and balanced modeling (Javed et al. 2018; Lu and Zhou 2011; Ruby and Kaur 2017; Tang et al. 2016; Zhao et al. 2018; Zhou et al. 2018).

A multitude of measures have been proposed to access the rich information coded in signed graphs. Mishra and Bhattacharya (2011) introduces *trustworthiness* and *deserve* as local vertex-based measures of bias to reflect the expected weights of out- and in-edges. *Controversy* was introduced by Garimella et al. (2017) as the likelihood a random walk will return to the same side of the network. This method improved the examination of triangles in Leskovec et al. (2010b) by including pendant vertices and proposed to reduce controversy by bridging opposing viewpoints. *Conflict* is defined in Chen et al. (2018) by examining the Laplacian matrix to produce a "Conservation Law of Conflict" reminiscent of Kirchhoff's laws—we provide our own Conservation Law of Controversy in Sect. 4.3. Kumar et al. (2018) discuss group mobilization against other groups to describe conflict in intra-community interactions, and Guha et al. (2004) examines *trust* through an iterative build of belief matrix. Yuan et al. (2017) introduces a sign prediction model for sparse data edge prediction in which they convert the original graph into a edge-dual graph and apply machine learning to

predict signs in sparse graphs. Established methods of network graph analysis focus on endorsement analysis through local topology analytics and strive for agreement by changing (Leskovec et al. 2010b), adding (Garimella et al. 2017) or removing (Guha et al. 2004) edges in the graph. We propose to analyze the signed graph in its entirety and characterize the vertices and edges through **frustration cloud**-based attributes.

In cybernetics, a multi-agent network dynamic is often too complex for existing tools to analyze the entire network and collective dynamic reactions. It is known (Rusnak 2013) that our methodology for balancing a signed graph works on any signed graph and certain classes of hypergraphs. Altafini (2013, 2019) proposes controllability and consensus algorithms in networks by examining the effects of the bipartite consensus of Harary (1953). Pan et al. (2016) examine the bipartite structure of Laplaican dynamics and node decomposition. Hu et al. show that the ideal state of the multi-agent system can modeled as a balanced graph, and that the system converges to the optimal state through the bipartite consensus iterations (Hu and Zheng 2013), while uncontrollability and stabilizability is examined by Alemzadeh et al. (2017). Algorithms for the characterization of the status quo have been examined in Li et al. (2005) for transitive graphs. Jiang et al. propose a sign-driven consensus as a control protocol measured via Laplacian dynamics (Jiang et al. 2016). She et al. (2020) examine consensus in terms of graphical characterizations of the controllability of signed networks and offers a heuristic algorithm for leader selection based on balance theory. This prior work focuses on producing a single balanced state. In this paper, we propose a discrete alternative to find all nearest balanced states of the network via frustration cloud graph analysis.

2.4 Related work in sign graph clustering

We compare the methods introduced in this paper to standard spectral clustering on only positive edges for community detection. Researchers have only recently started mining negative links in networks for community detection (Esmailian and Jalili 2015). Spectral clustering for signed graphs was introduced by Kunegis et al. (2010) by the way of a positive semi-definite modified Laplacian matrix approach. The approach essentially counts positive edges between clusters and negative edges within clusters. Multiple signed graph clustering methods have been proposed since Tang et al. (2016), normalizing Lapacian in different ways. A more recent survey is begin prepared that compares multiple signed spectral clustering methods in terms of effectiveness for the community finding and scalability for large network graphs.

3 The frustration cloud graph analysis

In this section, we expand the notion of the frustration index to frustration cloud analysis and propose a tree-based graph balancing algorithm to discover the nearest balanced states of a signed graph. This methodology improves on the singular focus of the frustration index while avoiding the tedious calculations of finding all balanced states, some of which are only present by passing through another balanced state.

We define *frustration cloud* as a set of all balanced states of an underlying graph that are achievable by a minimal number of edge sign changes. If a balanced state belongs to the frustration cloud, that means no subset of its edges can balance the underlying signed graph. While balanced states for an underlying unsigned graph G are always the same, the nearest balanced states for a signed graph Σ depend on the Σ and the minimal number of edge signs that need to be changed to achieve a balanced state. Balanced states that produce the frustration index are those with a minimal number of edge changes to reach a balance, and are always part of the frustration cloud. The nearness of these states for discovery of the frustration index are discussed in Aref et al. (2016). We use spanning trees as matroidal bases to balance the signed graph. A *spanning tree T* of a graph G is a maximal acyclic subgraph that contains all the vertices of G. For a spanning tree T in graph G and an edge e not in the spanning tree, $e \in E(G) \backslash E(T)$, the *fundamental cycle of e with respect to T in G* is the unique cycle in $T \cup e$. The number of edges outside a spanning tree is a known constant called the *cyclomatic number*. Spanning trees form a basis for the balanced signed-graphic matroid (Zaslavsky 1982). A spanning tree plus an additional edge whose fundamental cycle is negative is the base for the unbalanced signed graph.

3.1 Balancing via spanning trees

For a connected graph G, let $\Sigma = (G, \sigma)$ be the signed graph of G, and $\mathcal{T}_G$ be the set of spanning trees of G. We propose the graph-balancing algorithm that constructs the nearest balanced states of Σ from spanning trees of the underlying graph G. The underlying graph G is assumed to be connected. If it is not, the algorithm is applied to connected components of G. Algorithm 1 produces one balanced state Σ_T per spanning tree T.

Algorithm 1 Signed Graph Tree-Balancing Algorithm:

Input: Input signed graph $\Sigma = (G, \sigma)$.
  for all $T \in \mathcal{T}$, T is a spanning tree of Σ **do**
    for all edges e, $e \in \Sigma \backslash T$ **do**
      if fundamental cycle $T \cup e$ is negative **then**
        change edge sign: $e^- -> e^+$; $e^+ -> e^-$
      end if
    end for
    Construct new balanced signed graph Σ_T
  end for
Output: Set of nearest balanced states Σ_T, $T \in \mathcal{T}$

Algorithm 1 is illustrated in Fig. 4, where the process is illustrated for the signed graph Σ (left) and a single spanning tree (second left). Edges outside the spanning tree are dashed. As the fundamental cycles are found, the edges outside the spanning tree (grey) are examined. The edge sign is not changed if the fundamental cycle is positive (top), and it is changed if the fundamental cycle is negative (bottom) in Fig. 4 (second right). The balanced signed graph is produced with these signing changes in Fig. 4 (right).
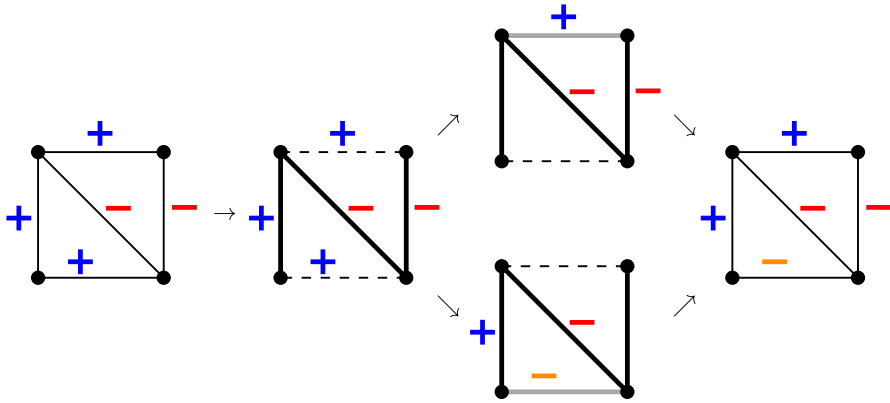
Fig. 4 The spanning tree balancing process via fundamental cycles. Changed edges appear lighter
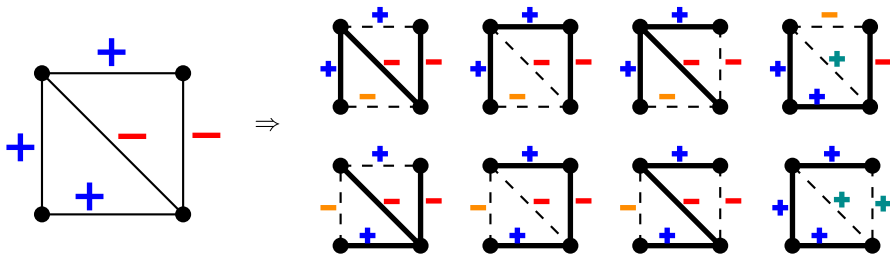

Fig. 5 The spanning trees of a signed graph (bold) produce balanced signed graphs. Edges outside each spanning tree are dashed, and re-signed edges are labelled in orange and teal (lighter). The negative edges form a cut-set in each balanced graph (Color figure online)

The base signed graph Σ in Fig. 4 (left) has a total of 8 spanning trees, marked with darker edges in Fig. 5 (right). The edges outside each spanning tree are indicated by dashed edges. For any spanning tree T and an edge e outside of that tree, the sub-graph $T \cup e$ contains a unique fundamental cycle C. The sign of e is chosen so that C is *positive*. The Algorithm 1 result for 8 spanning trees and signed graph Σ is in Fig. 5 (right).

For an underlying graph G, there are 8 possible balanced graphs as shown in Fig. 3. However, only four of the eight balanced states are achievable by Algorithm 1, as shown in Fig. 6. Not every balanced signed graph is obtainable by a balancing algorithm that uses spanning trees, *only the nearest balanced states* are.

Theorem 3.1 *If $\Sigma = (G, \sigma)$ is a signed graph of G, then the tree-balancing algorithm outlined in Algorithm 1 produces a minimal balancing set for Σ.*

Proof Let Σ be the signed graph of graph G, $T \in \mathcal{T}$ a spanning tree of Σ, and B_T be the balancing set produced by the tree-balancing algorithm (Algorithm 1). If B_T is not minimal, then there exists a smaller balancing set $S \subset B_T$ and an element $e \in B_T \backslash S$ whose reversal is not necessary to balance Σ. However, $T \cup e$ has a unique fundamental circle C, and the only edge of C outside of T is e, so e is required to balance and $B_T \backslash S$ must be empty. ☐
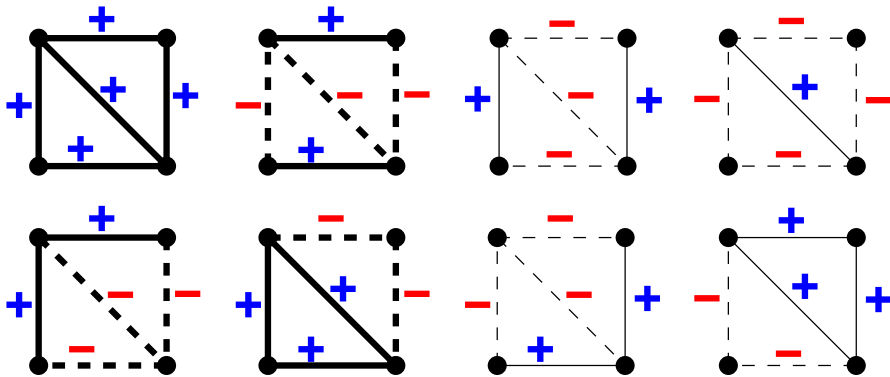
Fig. 6 Out of the eight balanced graphs for signed graph in Fig. 5. The four in bold can be reconstructed using the tree-balancing algorithm (Algorithm 1)

We explain the notion of nearest balanced states and the construction of the frustration cloud in Sect. 3.2.

3.2 The frustration cloud and consensus

In this section we formalize the notion of the *frustration cloud* as the set of nearest balanced states. See Definition 3.1.

Definition 3.1 The *frustration cloud* of a signed graph Σ, denoted $\mathcal{F}_\Sigma$, is the set of all balanced signed graphs obtained by graph B, the tree-balancing Algorithm 1 on Σ.

All balanced states of the underlying signed graph have 0 frustration, per Theorem 2.1. They all represent different views of graph consensus. The frustration index is the smallest number of edge sign switches so the signed graph achieves a balanced state. If the frustration index of signed graph Σ is $Fr(\Sigma)$, that means that Σ is $Fr(\Sigma)$ many sign changes from being balanced.

Let us extend that notion to all balanced states. For a signed graph $\Sigma = (G, \sigma)$, the set of edge-signing functions $\{+, -\}^E$ form a Boolean lattice $\mathcal{L}$ ordered by negative edge subset containment. Thus, the all positive edge signing $(G, +)$ is the **0** element, the all negative edge signing $(G, -)$ is the **1** element, and it is graded by the number of negative edges. Let $\Sigma_1 = (G, \sigma_1)$ and $\Sigma_2 = (G, \sigma_2)$ be two signings of the same underlying graph G. The *distance* between Σ_1 and Σ_2, $d(\Sigma_1, \Sigma_2)$ is the Hamming distance between them in $\mathcal{L}$. This is equivalent to the length of the shortest path between Σ_1 and Σ_2 in $\mathcal{L}$ when regarded as a graph. The Boolean lattice for a signed triangle graph is illustrated in Fig. 7.

Theorem 3.2 *Let Σ be a signed graph, and let Σ' be a balanced state of Σ. $\Sigma' \in \mathcal{F}_\Sigma$ if, and only if, Σ' can be obtained by the minimal balancing set whose size is less than or equal to the cyclomatic number.*

Proof If $\Sigma' \in \mathcal{F}_\Sigma$, it is obtained by the tree-balancing algorithm, which cannot change more edge signs than the cyclomatic number. By Theorem 3.1, this is a minimal balancing set.
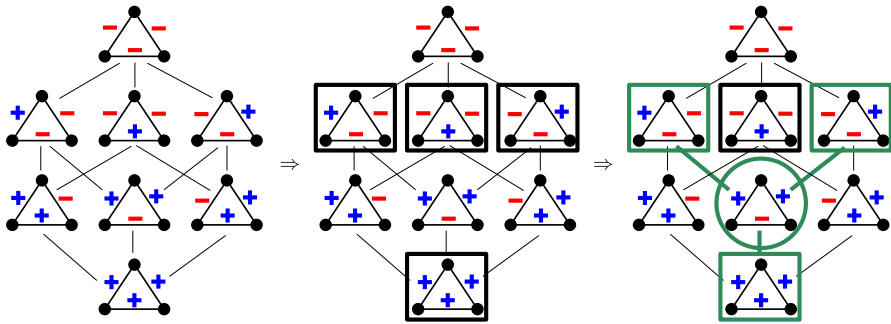
Fig. 7 The Boolean lattice for a signed triangle graph (left); black boxes mark the balanced signed graphs (center); if the underlying signed graph Σ is in green circle, then the green boxes mark a frustration cloud $\mathcal{F}_\Sigma$ (right) (Color figure online)

Now, suppose Σ' is obtained by the minimal balancing B set whose size is less than or equal to the cyclomatic number. Observe that $G \setminus B$ is connected, and any spanning tree in $G \setminus B$ will also be spanning in G. Thus, B is obtained by a spanning tree in $G \setminus B$.   □

The frustration cloud is the set of balanced states resulting from Σ that have no more than the cyclomatic number of edge sign changes. It has a simple interpretation using the Boolean lattice of the signings of the underlying graph G. Consider the eight possible signings of the triangle graph in Fig. 7 (left), ordered by negative edge set containment. Out of these eight signings, exactly four of them are balanced; these are marked with black boxes in Fig. 7 (center).

Consider the signed graph Σ to be the graph in Fig. 7 (right) with a green circle around it. Since the triangle graph has the cyclomatic number 1, we search for all balanced states that are distance 1 or less away from Σ; these are marked with green squares. Observe that the balanced state in the black box in Fig. 7 (right) is not in $\mathcal{F}_\Sigma$, as it requires a path that exceeds the cyclomatic number—one must also travel through another balanced state to reach it.

More complicated graphs may produce balanced states of varying distance from the given signed graph. Consider the underlying graph given in Fig. 8 (left) and its corresponding Boolean lattice of signings in Fig. 8 (middle) where the negative edge sets are listed. Consider the signed graph Σ where edges e_2 and e_5 are negative, and the rest are positive (Fig. 5). This is marked with the open square in Fig. 8 (middle) labeled 25. All the balanced signings are marked with closed squares. Since the cyclomatic number of the underlying graph is 2, we search for all balanced states of distance less than or equal to 2 from the open square; these are indicated by the dark paths in Fig. 8 (right).

The example in Fig. 8 illustrates that the frustration index is obtainable by analyzing the frustration cloud. The signed graph Σ from Fig. 8 has $Fr(\Sigma) = 1$, as the balanced state of minimum distance is of distance 1 away from Σ. It is trivial to verify that the frustration cloud of a balanced graph consists only of itself.
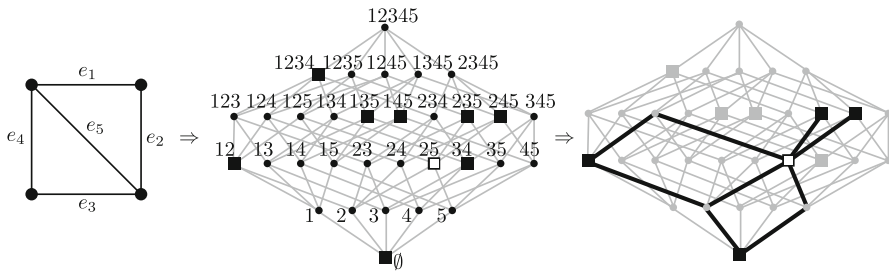
Fig. 8 Left: the underlying graph G from Fig. 5. Middle: the Hasse diagram of all signings of G, with balanced states as closed squares, and the given signed graph from Fig. 5 as the open square. Right: the four elements of $\mathcal{F}_\Sigma$ and their shortest paths to Σ from Fig. 6 (bold)

4 Probabilistic consensus model

Consensus for social networks is community resolution when opposing parties set aside their differences and barely agree on a statement (Hartnett 2011). State-of-art consensus modeling in social network analysis has focused on the locality of the agreement (Garimella et al. 2017; Guha et al. 2004), and it did not consider an entire graph. As illustrated in Sect. 3, there can be multiple balanced states of the same graph, meaning there are multiple ways to achieve global consensus. In this section, we formalize the measures of vertices, edges, and the entire graph stemming from frustration cloud-based analysis.

There are multiple ways in which a minimum set of sentiments can be changed to result in *identical outcomes of consensus*. Different spanning trees in the tree-balancing algorithm (Algorithm 1) can result in the same nearest balanced state, as illustrated in Fig. 5. We weigh each element of the frustration cloud by the number of times it is produced by a basis, as spanning trees are bases for the balanced states of a signed graph (Zaslavsky 1982).

Definition 4.1 For a signed graph $\Sigma = (G, \sigma)$ and balanced signed graph $\Sigma' \in \mathcal{L}$, let $w_{\Sigma'}$ be *weight of Σ' relative to Σ* and defined to be the number of spanning trees of G that balance Σ into Σ'.

An unbalanced signed graph is always assigned a weight of 0. Figure 9 illustrates the balanced signed graphs in Fig. 5 grouped by identical balanced states. The weights of these balanced states are equal to 3, 3, 1, and 1, as indicated by the boxed groupings of the balanced states. The weight captures the frequency of *appearance* of each balanced state using different underlying spanning trees. The weight of the balanced state is a measurement of the likelihood a given consensus will occur, as illustrated in Fig. 10.

4.1 Global vertex status

Let G be a graph whose set of spanning trees is $\mathcal{T}_G$. Given a signed graph $\Sigma = (G, \sigma)$ and a spanning tree $T \in \mathcal{T}_G$, recall that Σ'_T is a balanced signed graph obtained by the tree-balancing algorithm (Algorithm 1). The bipartition (U_T, W_T) from Theorem 2.1 results in two induced subgraphs, as illustrated in Fig. 10, Σ'_{U_T} and Σ'_{W_T}. The sub-
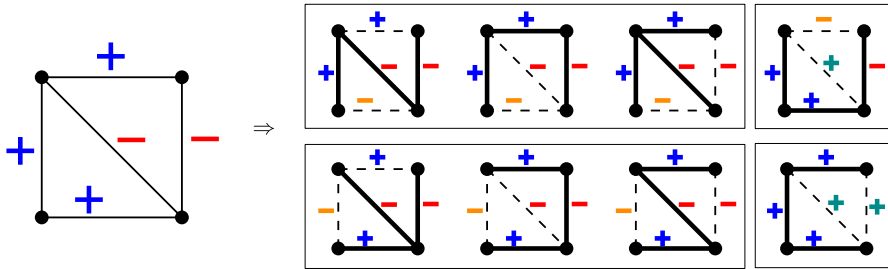
Fig. 9 Tree-balancing algorithm (Algorithm 1) on signed graph Σ produces 4 balanced states. Different spanning trees can produce same balanced state. The edges outside each spanning tree are indicated as dashed lines
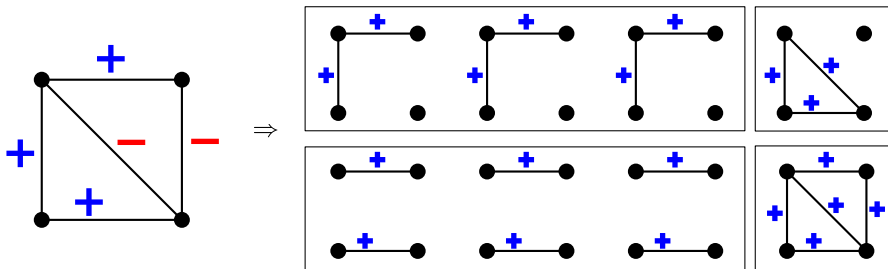

Fig. 10 Harary cuts per balanced state: the deletion of the negative edges in each balanced state in the frustration cloud of the signed graph in Fig. 9

graphs are named so that $|U_T| \leq |W_T|$; thus, Σ'_{W_T} always has the majority of vertices. *Status* (Definition 4.2) is the likelihood that a majority of the vertices in a network can be convinced to agree with a specific node's position over all nearest balanced states, with multiplicity determined by the weight.

Definition 4.2 The *status* of a vertex v in $\Sigma = (G, \sigma)$ is defined as the normalized sum of step functions if vertex v is in the larger subgraph Σ'_{W_T}:

$$status(v) = \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \delta_{\Sigma'_{W_T}}(v), \text{ where } \delta_{\Sigma'_{W_T}}(v) = \begin{cases} 1 & \text{if } v \in \Sigma'_{W_T} \\ 0.5 & |W_T| = |U_T| \quad \text{tie-break} \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 4.1 *Let* $\Sigma = (G, \sigma)$ *be a signed graph with frustration cloud* $\mathcal{F}_\Sigma$. *Status can be defined as a sum of step function if vertex* v *is in larger balanced sub-graph* Σ'_{W_T}, *weighted by* $w_{\Sigma'}$, *the number of spanning trees of G that balance* Σ *into* Σ'.

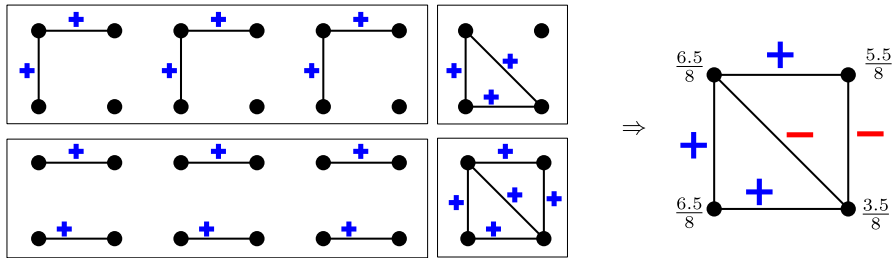$$status(v) = \frac{1}{|\mathcal{T}_G|} \sum_{\Sigma' \in \mathcal{F}_\Sigma} w_{\Sigma'} \delta_{\Sigma'_{W_T}}(v).$$

Fig. 11 Harary-cut for the nearest balanced states weighted by their occurrence (left) and the calculated status values for signed graph Σ

Proof From Definition 4.1, $w_{\Sigma'}$ counts the number of spanning trees that contribute to a given balanced state. Separating these into individual spanning trees and using Definition 4.2 gives the result. □

Figure 11 uses the components of Fig. 10 to determine the status.

The top-left vertex of Fig. 11 has status $[3(1) + 1(1) + 3(0.5) + 1(1.0)]/8 = 6.5/8$; the bottom-left is the same at $6.5/8$; the top right is $[3(1) + 3(0.5) + 1(1)]/8 = 5.5./8$; and the bottom-right is $[1(1) + 3(0.5) + 1(1.0)]/8 = 3.5/8$. Next, let us consider a *sum* of all statuses of all vertices in a graph and Definition 4.2.

Lemma 4.2 *For signed graph $\Sigma = (G, \sigma)$, vertex set V, and tree-spanning set $\mathcal{T}_G$, the sum of statuses of all vertices in Σ equals the normalized sum of cardinality of the larger component of the Harary-cut over all spanning trees $T \in \mathcal{T}_G$.*

$$|\mathcal{T}_G| \sum_{v \in V} status(v) = \sum_{T \in \mathcal{T}_G} |V(\Sigma'_{W_T})|.$$

Proof Summing both sides of the definition of status over all vertices gives

$$\sum_{v \in V} status(v) = \sum_{v \in V} \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \delta_{\Sigma'_{W_T}}(v) = \frac{1}{|\mathcal{T}_G|} \sum_{v \in V} \sum_{T \in \mathcal{T}_G} \delta_{\Sigma'_{W_T}}(v)$$

$$= \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \sum_{v \in V} \delta_{\Sigma'_{W_T}}(v) = \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} |V(\Sigma'_{W_T}))|.$$

The last equality holds even in the event of having two components of equal size, as we have defined status. Since $\delta_{\Sigma'_{W_T}}(v)$ treats them as an equal split (0.5), there are the same number of vertices in both the new majority as well as the minority— which is equivalent to counting the size of the tied majority. The proof is completed by multiplying by $|\mathcal{T}_G|$. □

4.2 Global vertex influence

We now consider the variation of attitudinal strength captured by edge signs in attitudinal networks. When students assign a strong rating score for an instructor evaluation,
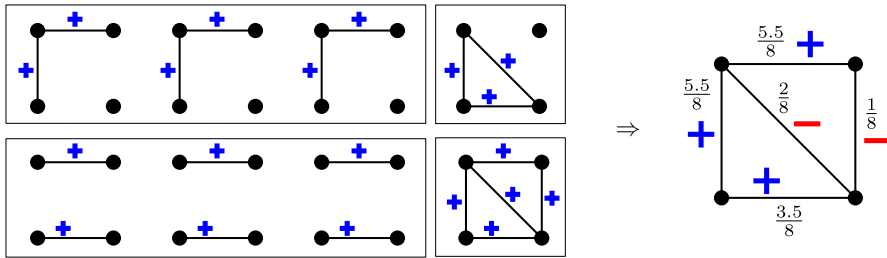
Fig. 12 Left: Harary-cut; right: edge agreement values

it is hard to separate affective, behavioral, and cognitive components of the attitude expressed in that one sentiment. Did the student take all of their other instructors into consideration? What is the subjective evaluation range? How much of the rating is based on students' own subjective performance in the class? How likely is the student to change his mind when he talks to his peers? We examine the strength of the beliefs held within these edges. We propose another new measure for edges similar to status, termed *agreement*, to measure how strongly held a given edge-sentiment is. It is likely that an edge will be positive and contribute to the consensus decision in all near balanced states produced by the tree- balancing algorithm. See Definition 4.3.

Definition 4.3 The *agreement* of an edge e in a signed graph is a normalized sum of all occurrences of an edge in the largest component of a Harary-cut over all spanning trees.

$$agreement(e) = \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \delta_{\Sigma'_{W_T}}(e), \text{ where } \delta_{\Sigma'_{W_T}}(e) = \begin{cases} 1 & \text{if } e \in \Sigma'_{W_T} \\ 0.5 & |W_T| = |U_T| \quad \text{tie-break} \\ 0 & \text{otherwise.} \end{cases}$$

Figure 12 shows agreement for the given example. The larger agreement an edge has, the more likely it will appear in the final, majority decision.

Agreement is calculated dually to status. Figure 11 demonstrated that the bottom-left and top-left vertices both have the largest status values. However, the agreement in Fig. 12 helps to quantify the differences between them.

Parallel to Lemma 4.2 we immediately have:

Lemma 4.3 *For signed graph $\Sigma = (G, \sigma)$, edge set E, and tree-spanning set $\mathcal{T}_G$, the sum of the agreement of all edges e, $e \in E$, in Σ equals the normalized sum of edge cardinality of the larger component of the Harary-cut over all spanning trees $T \in \mathcal{T}_G$.*

$$|\mathcal{T}_G| \sum_{e \in E} agreement(e) = \sum_{T \in \mathcal{T}_G} |E\left(\Sigma'_{W_T}\right)|.$$

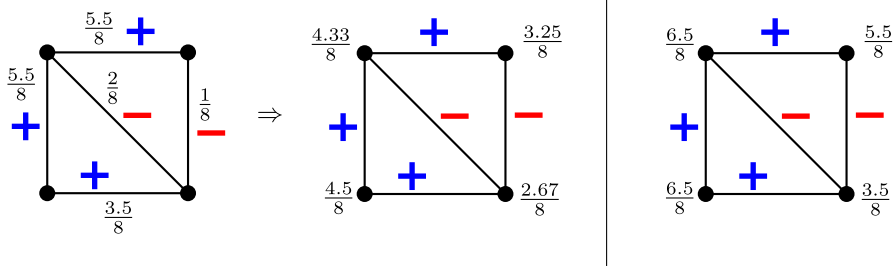Edge agreement is now averaged around each vertex to compare to status. This metric is called *influence*.

Fig. 13 For the signed graph Σ in Fig. 5, its edge agreement values (left) and vertex influence (center) are compared to its vertex status (right)

Definition 4.4 The *influence* of a vertex v in a signed graph is the average agreement of all edges incidental to the vertex v,

$$influence(v) = \frac{1}{deg(v)} \sum_{e \sim v} agreement(e).$$

Comparing the influence to the status in our examples, we see that the influence is always less than or equal to status.

Status and influence provide two measures of vertex influence in the attitudinal network graph, as illustrated in Fig. 13. The relationship of *status* and *influence* measures for vertex v as outlined in Lemma 4.4. Their relation stems from Definition 4.2, Definition 4.4, and from comparing the totality of edge counts around each vertex.

Lemma 4.4 *For an unbalanced signed graph* $influence(v) \leq status(v)$. *Moreover, equality holds when v is a pendant vertex whose edge is positive; influence is 0 when v is a pendant vertex whose edge is negative.*

4.3 Conservation of controversy

Consensus is a general agreement that can be achieved without unanimous voting. If consensus in the signed graph is unanimous, then the Harary-cut produces one partition consisting of the entire connected graph; the nearest balanced state has all positive edges. On the other end of the spectrum, the nearest balanced state can result in a Harary-cut that has bipartitions of equal size, and the entire graph is deadlocked in indecision. Controversy (from Latin *controversia* meaning "turn in opposite direction") occurs anytime there are conflicting opinions in the group. Controversy in balanced graph states occurs when consensus is achieved but the voting was not unanimous. Every balanced state but one (the all positive signed graph) has a certain level of controversy associated with it. The measure of the average status of the nearest balanced states can quantify controversy for the underlining signed graph, and the graph status definition is Definition 4.5.

Definition 4.5 Let $status(\Sigma)$ denote the graph status measure, and $|V(G)|$ is the number of vertices in the graph. Then, an average status of a signed graph Σ is

defined as

$$status(\Sigma) = \frac{1}{|V(G)|} \sum_{v \in V} status(v).$$

Lemma 4.5 *Let $\Sigma = (G, \sigma)$ be a signed graph, then $0.5 \leq status(\Sigma) \leq 1$.*

Proof This lemma sets the bounds of status sum in Lemma 4.2. From the definition of majority, we have that for every spanning tree T we have

$$\frac{|V(G)|}{2} \leq |V\left(\Sigma'_{W_T}\right)| \leq |V(G)|.$$

Summing over all spanning trees and normalizing the sum using Lemma 4.2, we get the result. $\qquad\square$

From Theorem 4.2 we know $|\mathcal{T}_G| \sum_{v \in V} status(v)$ is a sum of the sizes of the majority, so it must be an integer. The bounds are from Theorem 4.5 and multiplying by $|\mathcal{T}_G|$. Combining Lemmas 4.2 and 4.5 we have:

Theorem 4.1 *For a signed graph $\Sigma = (G, \sigma)$ and for all spanning trees T of Σ:*

1. *$status(\Sigma)$ is minimal $(= 0.5)$ if, and only if, $|V(\Sigma'_{W_T})| = |V(\Sigma'_{U_T})|, \forall T$,*
2. *$status(\Sigma)$ is maximal $(= 1)$ if, and only if, $|V(\Sigma'_{W_T})| = |V(G)|, \forall T$.*

We define average status over all vertices in the graph as a measurement of **controversy** (Theorem 4.1). The maximum value of $status(\Sigma)$ is 1.0; this is the case when all nearest balanced states have all positive edges, and everyone agrees all the time. The minimum value of $status(\Sigma)$ is 0.5, and the balancing consistently splits the set in two equally sized subsets. In between, if $status(\Sigma)$ is closer to 1, the entire graph has low controversy, and if it is trending to 0.5, the entire graph has higher controversy. One way to resolve a tie-break in Sect. 4.1 is to assign status and agreement values of 0.5 if the Harary-cut bipartitions are equal size. In the Human Resource Scenario, consider that a "reliable" or "reputable" vertex exists, and have that person (vertex in signed graph) break all ties in its own favor when the Harary-cut bipartitions are of equal size. We define *vertical status* in Definition 4.6.

Definition 4.6 The *vertical status* of a vertex v in $\Sigma = (G, \sigma)$ with respect to designated vertex t is

$$status_t(v) = \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \delta^t_{\Sigma'_{W_T}}(v), \quad \delta^t_{\Sigma'_{W_T}}(v) = \begin{cases} 1 & \text{if } v \in \Sigma'_{W_T}, \\ 1 & |W_T| = |U_T|, \text{ and } v, t \text{ in the same partition,} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.6 states that all tie-breaks increase the status of vertex t and vertices in the same subset as t. Figure 14 illustrates how vertical status compares to status for the signed graph from Fig. 5. Let us consider the top-left vertex (now a closed box) as the tie-breaker in Fig. 14 in case 1, and the bottom-right vertex (now an open box) as
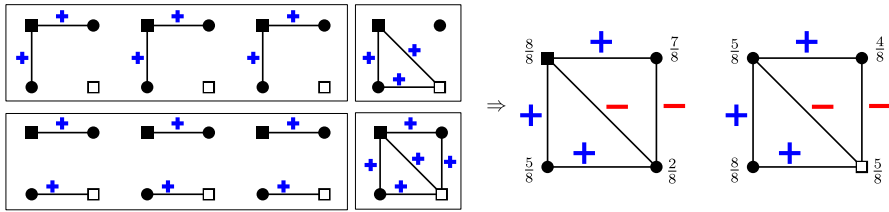
Fig. 14 Calculating the vertical status using the top-left vertex (middle); and bottom-right vertex (right)

the tie-breaker in case 2. *Case 1*: the top-left vertex is used to break any ties, and the vertical status values are now $(8/8, 7/8, 2/8, 5/8)$, as illustrated in Fig. 14 (middle). *Case 2*: the bottom-right vertex is used to break ties, and the vertical status values are now $(5/8, 4/8, 5/8, 8/8)$. In both cases, the status of the chosen vertex increased over the original status in Fig. 11.

Lemma 4.6 *For a signed graph $\Sigma = (G, \sigma)$ and vertex $v \in V$, the $status_t(v)$ is maximized when $t = v$.*

Proof If $t = v$, vertex determines its own tie-breakers, and every 0.5 in $\delta_{\Sigma'_{W_T}}(v)$ is replaced with a 1 in $\delta^t_{\Sigma'_{W_T}}(v)$. □

Definition 4.7 Let $status_t(\Sigma)$ denote the average vertical status of a signed graph Σ with distinguished vertex $t \in V$.

$$status_t(\Sigma) = \frac{1}{|V(G)|} \sum_{v \in V} status_t(v)$$

The average status and the average vertical status over the whole signed graph is a constant, called the *controversy* of the signed graph. This constant provides the following Conservation of Controversy Law.

Theorem 4.2 (Conservation of Controversy Law) *For a signed graph $\Sigma = (G, \sigma)$, graph controversy is equal to its status and any vertical status:*

$$controversy(\Sigma) = status(\Sigma) = status_t(\Sigma), \forall t \in V(G).$$

Proof As in Theorem 4.2,

$$\sum_{v \in V} status_t(v) = \sum_{v \in V} \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \delta^t_{\Sigma'_{W_T}}(v) = \frac{1}{|\mathcal{T}_G|} \sum_{v \in V} \sum_{T \in \mathcal{T}_G} \delta^t_{\Sigma'_{W_T}}(v)$$

$$= \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} \sum_{v \in V} \delta^t_{\Sigma'_{W_T}}(v) = \frac{1}{|\mathcal{T}_G|} \sum_{T \in \mathcal{T}_G} |V(\Sigma'_{W_T})| = \sum_{v \in V} status(v).$$

The second to last equality is a strict count of the size of the majority, while the last equality is from Lemma 4.2. The proof is completed by dividing by $|V|$. □

The **Conservation of Controversy Law** in Theorem 4.2 states that the average status is equal to any average vertical status. Interpreting average status as controversy, we can conclude that *the level of controversy is independent of vertex preference*, but the individual status values may change. The controversy in Fig. 11 (right) is 0.6875, in Fig. 14 (middle) is 0.6875, and in Fig. 14 (right) is 0.6875. The vertical status increase of the chosen vertex in Fig. 14 is *at the cost of status values of other vertices*, so the overall controversy stays the same. Controversy is one of the most important concepts in this paper, as it quantifies the level of controversy in a graph as a whole and does not depend on the tie-breaker decision, as proven by the Conservation of Controversy Law.

5 graphB: spanning tree-sampling balancing algorithm

The proposed measures of status 4.2, agreement 4.3, and controversy 4.1 of a signed graph require the computation of *all* spanning trees for the underlying unsigned graph G. For a real life socio-technical network, this is computationally prohibitive. In this section, we propose to utilize existing spanning tree graph discovery and sampling methods to accurately model measures derived from *a*ll spanning trees as outlined in Sect. 4 with a *s*ample of spanning trees (Russell and Norvig 2009). Note that the Conservation of Controversy Law (Theorem 4.2) holds for any fixed subset of spanning trees; while different subsets of trees may produce different controversy values, the conservation is in tie-break scenarios within the sample.

The upper limit on the number of spanning trees is computed by Cayley's theorem: the complete graph with v vertices has v^{v-2} spanning trees, and a complete bipartite graph with v, q vertices has $v^{q-1} \cdot q^{v-1}$ spanning trees (Buekenhout and Parker 1998; Rusnak et al. 2018). For a small social network such as the Highland Tribes (Read 1954) in Fig. 30 with 16 vertices and 29 positive and 29 negative edges, that number is quite high $16^{14} = 7.21e+16$. The exact number of spanning trees for any graph G can be calculated in polynomial time as the determinant of a matrix derived from the graph, using Kirchhoff's matrix-tree theorem (Tutte 1984). The Tutte polynomial of a graph can be defined as a sum, over the spanning trees of the graph, of terms computed from the "internal activity" and "external activity" of the tree. Its value at the arguments (1, 1) is the number of spanning trees (Tutte 1984), and the computed number of spanning trees for the Highland Tribes is 402,506,278,163. Computing probabilistic consensus measures as outlined in Sect. 4 will require running Algorithm 1 over 400 billion trees, and that is computationally prohibitive. We examine statistical samples of spanning trees to approximate modeling of balanced state coverage with k sampled spanning trees. We propose scaled adjustment of Algorithm 1 termed **graphB**: Spanning Tree-Sampling Balancing Algorithm, and we implement the proof-of-concept (Tešić et al. 2020). The efficiency improvements on the tree-based balancing algorithm are outlined in Algorithm 2, and its implementation complexity is addressed in Sect. 5.1. We model the frustration cloud and balanced state weights using n spanning trees to compute status, influence, and controversy.

Algorithm 2 includes sampling a tree step; instead of looping over all spanning trees, we select a subset of spanning trees $\mathcal{T}_n$ that contains n spanning trees of signed

---

Algorithm 2 graphB: Spanning Tree-Sampling Balancing Algorithm:

Input: Input signed graph $\Sigma = (G, \sigma)$.
Input: Sample n spanning trees T to $\mathcal{T}_k$ set using BFS, random or DFS.
  for all $i \in [1, n]$, $\Sigma_i, T_i \in \mathcal{T}_k$ **do**
    for all edges $e, e \in \Sigma \backslash T$ **do**
      if fundamental cycle $T_i \cup e$ is negative **then**
        change edge sign: $e^- -> e^+$; $e^+ -> e^-$
      end if
    end for
    Construct new balanced signed graph Σ_i
    Create Harary cutset for T_i, (U_i, W_i)
  end for
Output: Compute status for each vertex and agreement for each edge

---

graph Σ. Given a fixed number of vertices and edges, path-like trees have minimal eigenvalues while star-like trees have maximal eigenvalues, per Lovasz' eigenvalue characterization for trees (Lovsaz and Pelikan 1973). Next, we analyze three strategies for sampling spanning trees in the graphB algorithm (Algorithm 2) w.r.t. eigenvalues and the frustration cloud. A breadth-first sampling search favors star-like spanning trees (max eigenvalues), a depth-first spanning tree search favors path-like spanning trees (min eigenvalues), and a random tree selection is used as a baseline as it is the most efficient (random). We examine the sensitivity of our proposed method using random, breadth-first, and depth-first spanning tree samplings and demonstrate how status, influence, and controversy can be perceived in different paradigms.

Random sampling baseline A *uniform spanning tree* is a tree chosen randomly from among all the spanning trees with equal probability, and there are multiple known implementation algorithms. We assumed that a random sample will best represent the frustration cloud and multiplicity. The fastest implementation is a *random minimal spanning* tree algorithm; it generates random trees, but cannot guarantee sampling uniformity. The edges of the graph are assigned random weights, and then the minimum spanning tree of the weighted graph is constructed. We compared the implementation of three different algorithms in NetworkX (Hagberg et al. 2008) for discovery of the *minimal spanning tree*, namely DJP, Boruvka, and Kruskal's algorithms (Hagberg et al. 2008; Russell and Norvig 2009). They are all greedy algorithms that run in polynomial time, and we have not observed much difference in speed or randomization when selecting a specific one.

Breadth-first search (BFS) BFS searches for spanning trees by progressively exploring all the neighborhood vertices from the starting vertex (search key) at present depth before moving to next depth level. As a result, breadth-first spanning trees are the most star-like trees in the graph, and they represent the classes of trees that have maximal eigenvalues (Lovsaz and Pelikan 1973). The BFS algorithm has been used to find the shortest path between two vertices in a graph measured by number of edges, as it allows for the discovery of the shortest fundamental cycles in a graph. Thus, spanning trees resulting from a breadth-first search have the maximal number of pendant vertices and fundamental cycles of minimal length (Russell and Norvig 2009). We use NetworkX (Hagberg et al. 2008) implementation of BFS for proof-of-concept.

Depth-first search (DFS) DFS searches for spanning trees by exploring the branch to the highest depth level possible before backtracking and expanding (Russell and Norvig 2009), effectively delaying cycle feedback as long as possible. Trees determined by a depth-first approach are the most path-like trees with the minimal number of pendant vertices. The process maximizes the length of the fundamental cycle. The DFS algorithm has been used in determining the number of connected components in a graph (Russell and Norvig 2009). Also, the DFS spanning tree search algorithm maximizes the sampling of path-like trees, the classes of trees that have minimal eigenvalues (Lovsaz and Pelikan 1973), and the fundamental cycle length.

Random spanning tree sampling provides a straightforward way to analyze the frustration cloud, as context-driven algorithms such as breadth- or depth-first alter the resolution of the data. We conjecture that breadth-first tree sub-sampling will provide the greatest data resolution for computing measures in Sect. 4, while depth-first tree sub-sampling will produce a noisy interpretation of the same data. We demonstrate the validity of this assumption on a larger Wikipedia administrator dataset in Sect. 6.1.

5.1 graphB implementation and complexity analysis

The graphB algorithm implementation in Python is released as open source (Tešić et al. 2020). The algorithm implementation has the overall complexity of $O(n \cdot v \cdot e)$ for run time and $O(v^2 v)$ for memory consumption, where kn is the number of sampled trees, v is the number of vertices in the signed graph, and e is the number of edges. We keep the adjacency matrix in memory for the entire process ($O(v^2)$). In the pre-process step, we symmetrize the adjacency matrix ($O(v^2)$), find connected components ($O(v+e)$), sort by the number of neighbors in the largest connected component, and write it to a file ($O(v^2)$). In the process step, we find n trees ($O(n \cdot e)$). Then for each tree, we find all fundamental cycles in the graph, as there exists a cycle containing e if, and only if, there exists a fundamental cycle with respect to the selected spanning tree T that contains e. We have adopted a linear time algorithm for finding articulation points (Farina 2015) in ($O(v+e)$) time for a single spanning tree. The complexity of the entire processing step is $O(n \cdot v \cdot e)$. Note that algorithm implementation does not make any assumptions on the signed graph (sparsity, planarity). The only assumption graphB implementation makes is that the graph edge weights are either $+1$ or -1.

6 Proof of concept

The computation of frustration cloud-based measures for signed graphs is implemented using Python and Python libraries, and the code used for proof-of-concept is released on GitHub (Tešić et al. 2020). The analysis of the largest connected component, spanning tree search methods, and statistics is computed using the NetworkX (Hagberg et al. 2008) package. Experiments are run on the Texas State University LEAP system (LEAP 2020): Dell PowerEdge C6320 cluster node consists of two (14-core) 2.4 GHz E5-2680v4 processors 128 GB of memory each, and two 1.5 TB memory vertices with four (18-core) 2.4 GHz E7-8867v4 Intel Xeon processors (LEAP 2020). The

LEAP system allows us to scale the data analysis to support the sampling of $n = 1000$ spanning tree computations and to demonstrate the feasibility of computation on larger graph datasets (Leskovec and Krev 2014). In the *graphB* Algorithm 2 pipeline, spanning trees are generated over the dataset and saved in h5 format; we use the NetworkX (Hagberg et al. 2008) implementation of random minimal tree, breadth-first, and depth-first tree discovery. Next, for each generated spanning tree, the balancing algorithm is executed on the edges not in the spanning tree (Algorithm 2) for more details. We obtained a list of unique paths encompassing the spanning tree and given edge and checked for fundamental cycles. If the product of the cycles is -1, then the given edge completing the cycle by changes signs, resulting in a balanced cycle. We repeat the process for each edge. Once all edges outside the spanning tree were visited and edge signs persisted or flipped, the resulting state was a balanced state of the graph. Next, we take a Harary-cut and split the graph into two components. We repeat the process for $k = 1000$ trees (Algorithm 2). The final step is computing the status for each vertex and the agreement for each edge as the normalized sum over k sampled trees, per Definitions 4.2 and 4.3. Vertex influence is then computed as the normalized sum over the sampled 1000 trees per Definition 4.4 and the controversy of the entire graph per Theorem 4.1.

6.1 Wikipedia administratorship election data

Stanford Network Analysis Project's (SNAP) repository of network data provides good proof-of-concept access to attitudinal network graphs (Leskovec and Krev 2014). *Wikipedia* administrator election data represents votes by Wikipedia users in elections for promoting individuals to the role of administrators from July 2004 to January 2008. Wikipedia administrators are editors who have been granted the ability to perform special tasks. The dataset contains 7118 users (vertices) and 103,747 votes (edges) over 2794 elections with one election per candidate, and the outcome of the elections. Out of 2794 elections, 1235 resulted in the promotion to administrator (44.2%), and 1559 elections did not result in the promotion of the candidate. On the editorial side, 3464 editors cast zero or 1 vote over all elections, 5506 editors cast under 10 votes, and 1612 editors voted 10 times or more. Administrators are chosen through a *community review process* that seeks *consensus* is not a majority rule, as the *editor in charge* reviews editors' votes and rationale.

The signed graph is constructed from Wikipedia administrator election data so that each vertex represents an editor or nominee; if both are running for multiple administrator positions (this is possible as there are different Wikipedia sections), we represent one user with multiple vertices, where each vertex has a final outcome (winner, loser, editor). The edges in the graph model the vote of support or initial nomination ($+1$) or a vote of opposition (-1). We ignore the neutral votes as they are equivalent to no-vote or ambivalent votes per (Abelson and Rosenberg 1958). For k spanning trees and balanced states, k Harary cutsets are found from balanced states as in Algorithm 2. Sampled status and influence are computed as follows:

Definition 6.1 For a signed graph $\Sigma = (G, \sigma)$ and subset of size k of spanning trees $\mathcal{T}_k \subseteq \mathcal{T}_G$, the sampled status, agreement, influence, and controversy are computed as:
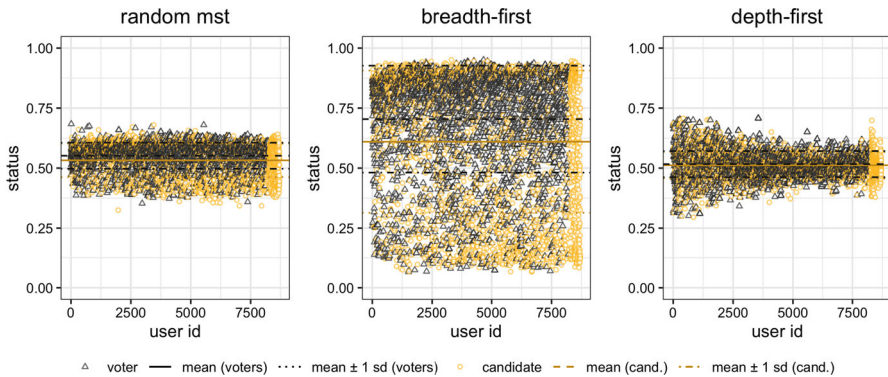
Fig. 15 Status of editors (black triangles) and nominees (yellow circles) in Wikipedia administrative election dataset resulting from different tree sampling methods: random minimal spanning tree (left), breath-first (center), and depth-first (right) (Color figure online)

1. $status(v) = \frac{1}{k} \sum_{T \in \mathcal{T}_k} \delta_{\Sigma'_{W_i}}(v)$, where $\delta_{\Sigma'_{W_i}}(v)$ is defined in Definition 4.2,

2. $agreement(e) = \frac{1}{k} \sum_{T \in \mathcal{T}_k} \delta_{\Sigma'_{W_i}}(e)$ where $\delta_{\Sigma'_{W_i}}(e)$ is defined in Definition 4.3,

3. $influence(v) = \frac{1}{deg(v)} \sum_{e \sim v} agreement(e)$,

4. $controversy(\Sigma) = \frac{1}{|V|} \sum_{v \in V} status(v)$.

Connected components 7066 users (vertices), or 99.3% of all vertices, and 103,663 of all votes (edges), or 99.97% of all edges, belong to the largest connected component of the constructed attitudinal graph. Only 52 users and 26 votes are not in largest connected component, and they represent unsuccessful nominations that fail to gather significant votes. We examine the Wikipedia dataset using a sample size of k spanning trees, where $k \in \{10, 100, 1000\}$.

6.1.1 Experiment: spanning tree discovery

First, we measure the status and influence for three different spanning tree discovery techniques on the Wikipedia dataset, for $k = 1000$ spanning trees are compared: *random* trees as determined by minimal spanning trees with random edge weights, *breadth-first* trees with a random initial vertex, and *depth-first* trees with a random initial vertex. We compute status and influence scores for all participants (vertices) in the Wikipedia election data. First, we analyze the data in id-status and id-influence space, and we color editors as black triangles and nominees as yellow circles. Whether a vertex is an editor or nominee *is not used* to determine status and influence scores; this information is only used in data analysis and the visualization step.

Results for status and influence for Wikipedia data editors and nominees for three spanning tree discovery techniques are shown in Figs. 15 and 16 . Both the editor and nominee means coupled with a 1 standard deviation band are shown as solid and dashed lines, respectively. The computed *s*tatus of the editors (black triangles) and nominees (yellow circles) appear in Fig. 15.

Table 1 Spanning three discovery methods comparison on Wiki data

Type	Mean status (controversy)	St. Dev status	Mean influence	St. Dev influence
Breadth-first	0.6693	0.2564	0.5178	0.2823
Random	0.54446	0.06117	0.3465239	0.15985
Depth-first	0.5149	0.0547	0.3067	0.1528

Status score distribution for each sampling methodology in Fig. 15 produces similar score distribution for editors and for nominees, as 1-SD bands for editors and nominees are close for each sampling method. By observing status measure only, one may conclude the Wikipedia election processes are fair based on the likelihood of landing in a majority (evenly spread out between voters and votees). Status does not discriminate between voters and nominees in Fig. 15. Next, let us examine the influence measure of the editors (grey) and nominees (yellow) in Fig. 16. The editors display a much larger influence value than the nominees. The influence score clearly separates high influence individuals (voters) from low influence individuals (votees) in the network. Note that the conjecture from Sect. 5 is shown valid in practice, as the breadth-first search provides the highest resolution of status and influence metric analysis in both figures. The depth-first experiment consistently produces a biased sample in terms of balanced states and the frustration cloud, and our experiments on other datasets consistently confirm the conjecture.

Overall data statistics are summarized in Table 1. Note that by design, depth-first (DPS) and breadth-first (BFS) represent lower and upper bounds of controversy (mean status) that the experiment confirms. Controversy is computed for all three tree discovery strategies, and the breadth-first value is 0.6693 while the depth-first value is 0.5159. Controversy, as defined in Theorem 4.1, is a constant value when all spanning trees are accounted for, and breadth-first and depth-first give us the range estimate of the actual value if all spanning trees are considered. Controversy is a relative measure of the attitudinal network graph, and if compared to another graph, the same spanning tree sampling method must be used for valid comparison.

Next, we provide analysis of only the *nominees* (yellow circles). While restricted to the nominees and using the known outcomes of the votes, we examine the efficacy of our method. We re-color the id-status and id-influence graphs with nominee outcomes to illustrate the effectiveness and difference in the metrics as follows: blue triangles represent a positive outcome (promoted to administrator) and red circles represent a negative outcome (not promoted or withdrew its nomination). This is depicted in Fig. 17 and captures the different measures of influence and status present. High status individuals are mostly the nominees that won the administrator elections, and low status individuals are mostly the nominees that did not win the elections for random and BFS spanning tree sampling. The sensitivity of status measure for nominees and its strong relation to outcome makes status a good predictor of promotability in random and breadth-first spanning tree discoveries. For DFS, the resolution of the status is too low to make any conclusion.
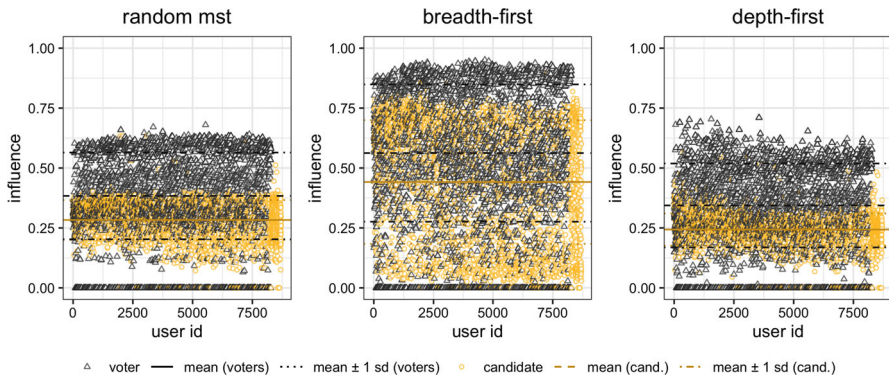
Fig. 16 Influence of editors (black triangles) and nominees (yellow circles) in Wikipedia administrative election dataset resulting from different tree sampling methods: random minimal spanning tree (left), breadth-first (center), and depth-first (right) (Color figure online)
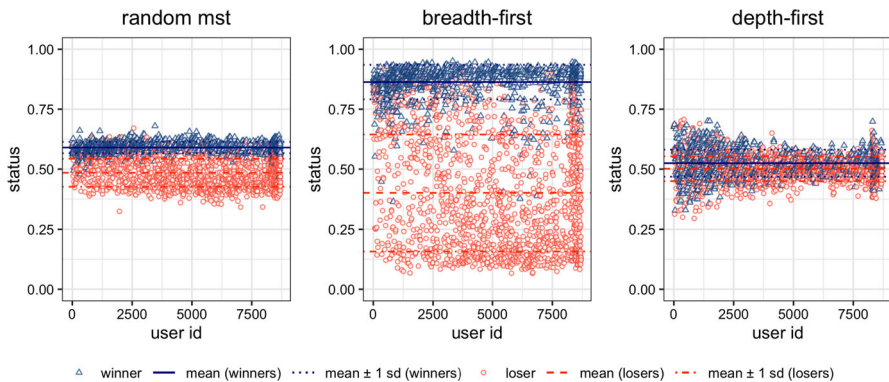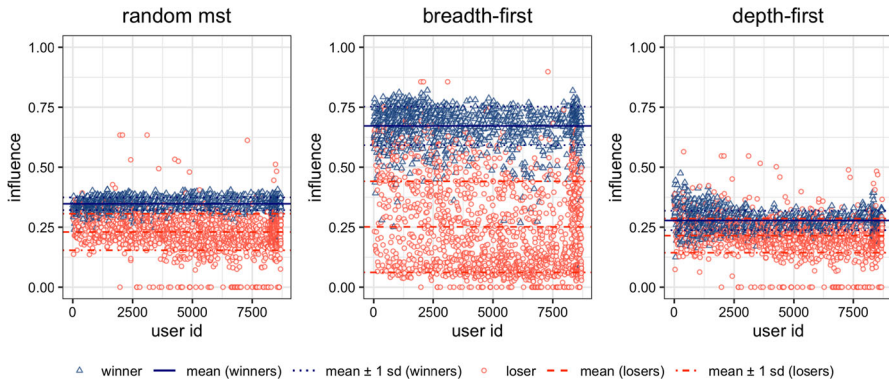

Fig. 17 Status of winners (blue triangles) and losers (red circles) in Wikipedia administrative election dataset resulting from different spanning tree discovery methods: random minimal spanning tree (left), breadth-first (center), and depth-first (right) (Color figure online)

Influence score distribution colored by outcome in Fig. 18 provides a better separation of winners and losers, even for the DFS sampling method. In a promotional network such as a Wikipedia election, status does not distinguish between editors and nominees, but influence does. Nominee-only analysis (people that have been voted on) illustrates a high correlation between status and influence scores prediction. Upon further analysis of Figs. 17 and 18, we identify clear promotional outliers: nominees with high status/influence that did not win the nomination. This deserves further study on anomalous promotion and case-by-case analysis.

Depth-first spanning tree discovery strategy does not allow for a reliable representation of the statistical significance of balanced states. We exclude the depth-first spanning tree search in subsequent experiments for status and influence measure and focus on random and breadth-first search spanning tree sampling.

Fig. 18 Influence of winners (blue triangles) and losers (red circles) in Wikipedia administrative election dataset resulting from different spanning tree discovery methods: random minimal spanning tree (left), breadth-first (center), and depth-first (right) (Color figure online)

6.1.2 Experiment: sufficient number of spanning trees

In this experiment, we offer a heuristic answer to the complex estimation of the sufficient number of spanning trees (n in Definition 6.1) that will produce reliable modeling of balanced state representation. We evaluate the sensitivity of status and influence scores to the number of spanning trees sampled for random minimal tree and breadth-first search spanning tree for $n = 10$, $n = 100$, and $n = 1000$ trees using random minimal sampling and breadth-first search techniques. The results for random spanning tree sampling for status is illustrated in Fig. 19 and for influence in Fig. 20. For both figures, random samples are on top and the breadth-first samples are on the bottom.

Status for $n = 10$ can only take one of 11 different values (as the vertex in majority or not for each of the 10 sampled balanced states), as illustrated in Fig. 19. A shelving effect is visible due to so few samples. Influence (Definition 6.1) has a higher resolution, as it is based on the average of agreement for each vertex, and vertex measure differs. A shelving effect is visible in Fig. 20 for a tight group of editors that behave in a similar fashion. Higher n allows for more diverse samples to contribute to status, and while the overall resolution of both discovery strategies is smaller, it provides better results. Figures 19 and 20 show that the status values have the higher resolution. They also show that $n = 100$ of breadth-first and random sampling spanning trees achieves similar separation results in nominee outcome status as $n = 1000$ trees.

What is the guiding principle for larger graphs? We have tested the method on a larger signed graph for Slashdot, and $n = 1000$ seems to be a good sampling rate for breadth-first spanning tree sampling, as illustrated in Fig. 25.

6.1.3 Experiment: outcome analysis

Measures of status and influence can be used to access the outcome for a vertex in an attitudinal graph. Requests for adminship (RfA) is the process by which the Wikipedia community decides to promote nominees into administrators (Wikipedia 2020). Here,
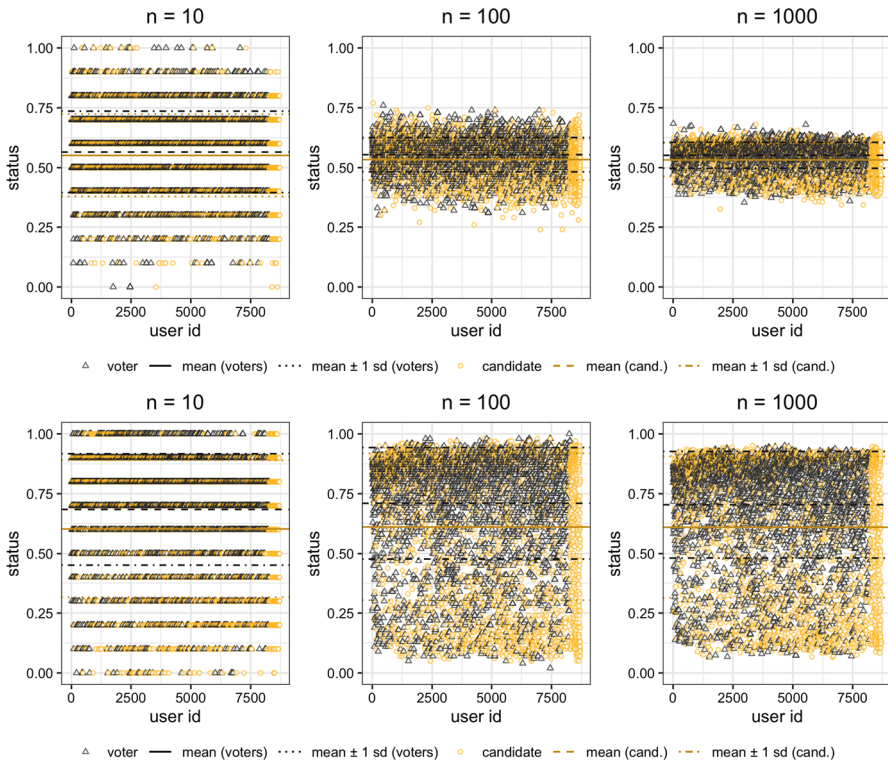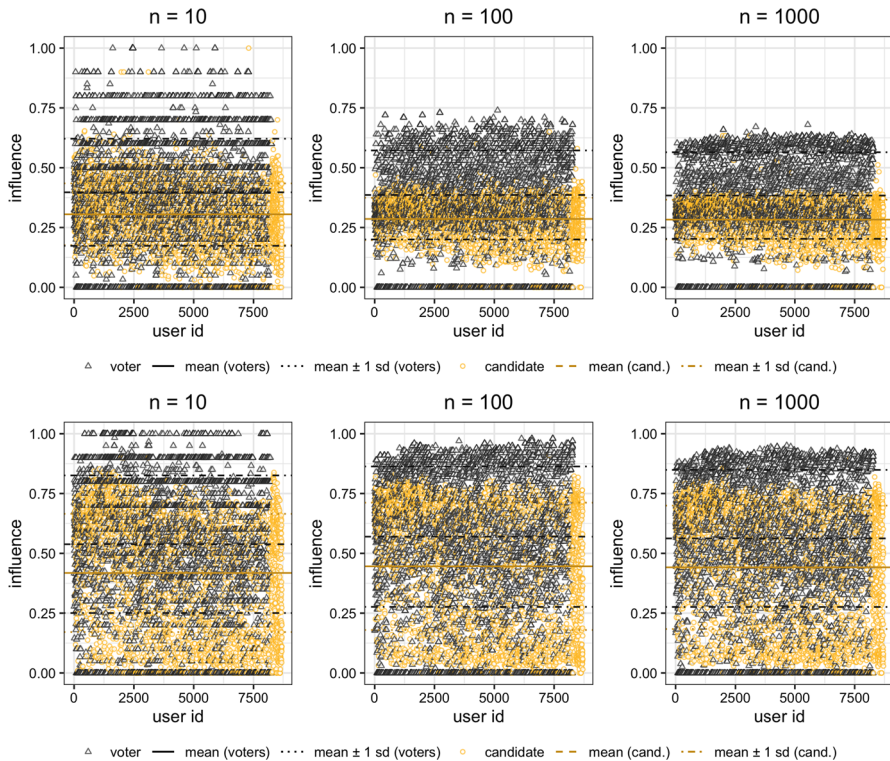
Fig. 19 Status score distribution for random (top row) and breadth-first (bottom row) search tree sampling to number of n sampled trees, $n = 10$, $n = 100$ and $n = 1000$ for editors and nominees for status

we use *RfA* as the ratio of total votes for the nominee overall votes. Election outcomes are colored by blue triangles representing candidates that won the election and red circles representing candidates that lost the election. Nominee status is obtained by users submitting their own requests for adminship or being nominated by editors.

The final outcome for Wikipedia is a complex process that involves majority voting (RfA in Fig. 21) and a vetting process. In general, if the number of positive votes is under 65%, the nominee is rejected; if the number of votes is over 75%, the nominee is selected. A vetting process and discussion determine the final outcome. Burke proposed a model of the behavior of candidates for promotion to administrator status in Wikipedia (Burke and Kraut 2008). He analyzes multiple measurable features of the nominee (strong edit history, varied experience, user interaction, helping with scores) and highlights similarities and differences in the community's stated criteria for promotion decisions to those criteria that actually correlated with promotional success. In this experiment, we examine the use of status and influence scores per vertex as vertex features. Status and influence do not consider any of Burke's candidates' features, only their position in the signed graph. The relationship of status and influence to RfA is illustrated in Fig. 21.

Fig. 20 Influence score distribution for random (top row) and breadth-first (bottom row) search tree sampling to number of N sampled trees, $N = 10$, $N = 100$ and $N = 1000$ for editors and nominees for influence

Figure 21 shows that both status and influence are highly positively correlated with RfA, as the RfA outcomes mirror those of status and influence, and all measures are highly correlated to the outcome as well. A summary of aggregate findings is in Table 2. Next, we study the nominees whose status and influence scores are different (Fig. 22).

6.1.4 Experiment: status/influence cone

We now analyze outcomes and RfA in a status-influence feature space. The vertices only appear under $x = y$ line in the graph, as shown in Fig. 23 and Lemma 4.4. Editors, the black triangles in Fig. 23 (left) have the higher status and influence as leaders in swaying opinions. In Fig. 23 (right), we restrict the analysis to nominees and color by outcome only: blue triangles are elected and red circles are rejected. Significant separation between the elected and rejected nominees is apparent. Note they are further away from the 45° line representing the most influential users, which are almost exclusively editors.

Next, we analyze RfA values in the status-influence space in Fig. 24. The left graph shows continuous scores, while the right one uses the Wikipedia admin score scale.
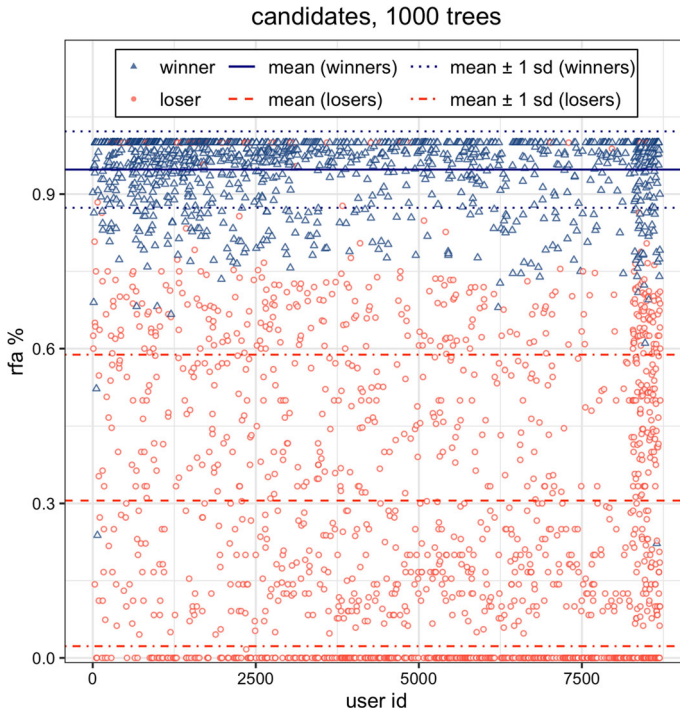
Fig. 21 Wikipedia data analysis of the request for adminship (RfA) (Wikipedia 2020). Blue triangles are users that won the election, and red circles are users that lost the election (Color figure online)
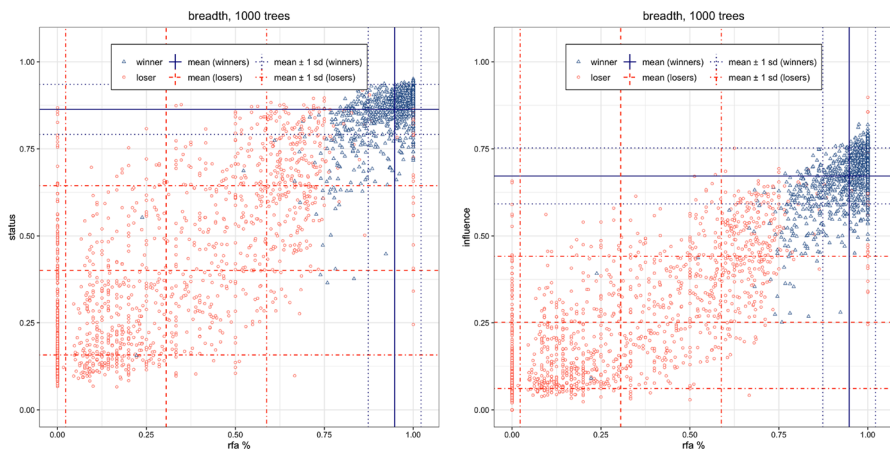
Fig. 22 Wikipedia data analysis of (left) status versus RfA and (right) influence versus RfA

Table 2 Measure distribution in Wiki adminship dataset for $N = 1000$ breadth-first spanning tree discovery

Mean (St. Dev)	Nominees	Promoted	Not Promoted
RfA	0.9476 (0.0742)	0.3055 (0.2826)	
Status	0.6097 (0.2962)	0.8632 (0.0719)	0.4009 (0.2433)
Influence	0.4414 (0.2579)	0.6721 (0.0803)	0.2514 (0.1898)
Mean (St. Dev)	All	Editors	
RfA	0.5958 (0.3852)	N/A	
Status	0.6693 (0.2564)	0.7041 (0.2226)	
Influence	0.5178 (0.2823)	0.5624 (0.2864)	
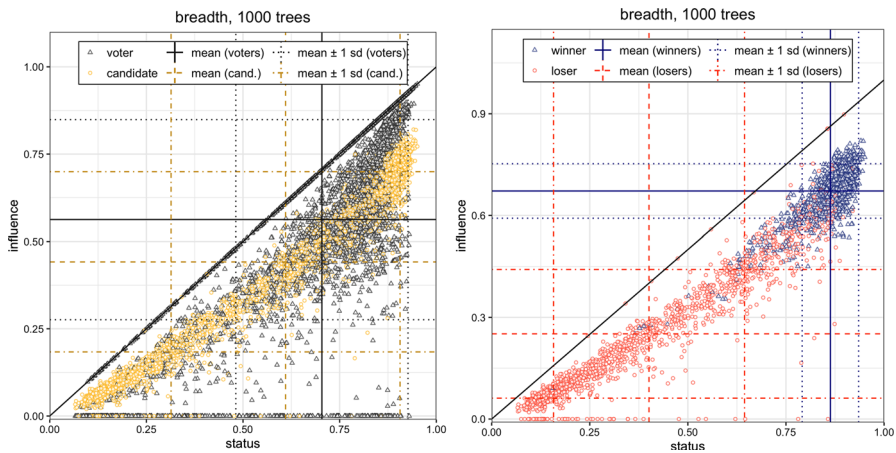
Fig. 23 Wikipedia data analysis of the voting process from status versus influence perspective: (left) editors (black triangles) versus nominees (yellow circles); (right) by Wikipedia nominee outcome: blue triangles are elected, red circles are rejected (Color figure online)

Here, we flag spam users, privileged users, narrow domain users and all anomalies by examining red circle distribution, the RfA in [65,75)% over status-influence graph in Fig. 24 (right).

Our algorithm for Wikipedia adminship uncovered several interesting cases. Wiki ID *80-man* had a status of 0.919, an influence of 0.622, and lost the elections. The user primarily edited a lot of music pages, and was deemed "not ready" for adminship yet based on other criteria. Wiki ID *bozmo* had a status of 0.405, an influence of 0.277362637, and won the elections. He self-nominated after around 3 years of contributions, and he received a fair amount of opposition due to being less active around the time of nomination. It is unclear why he won. Wiki ID *tjstrf* had a status of 0.905, an influence of 0.649, and lost. Due to some discussion of sensitive topics, he rejected the promotion. Wiki ID *dmn* had a status of 0.448, an influence of 0.279753623, and won the election. Further research showed they have been on Wikipedia for over 16 years, made regular edit contributions for a year, and have a history of conflicts and controversial comments. All four cases show that our algorithm uncovered atypical
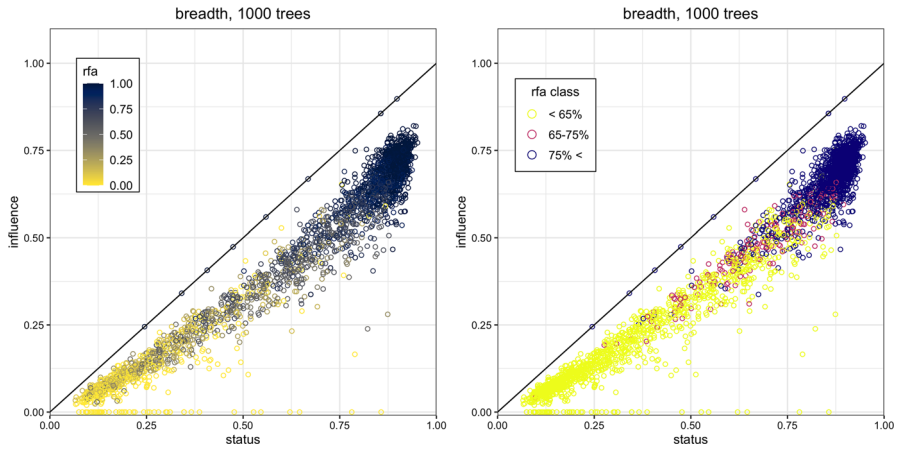
Fig. 24 Wikipedia data analysis of the voting process from status vs. influence perspective. Left: RfA score for nominees ramped from yellow to blue (light-to-dark). Right: binned by $< .65\%$ in yellow, $[.65, .75)$ red, and $[.75, 1]$ blue (Color figure online)
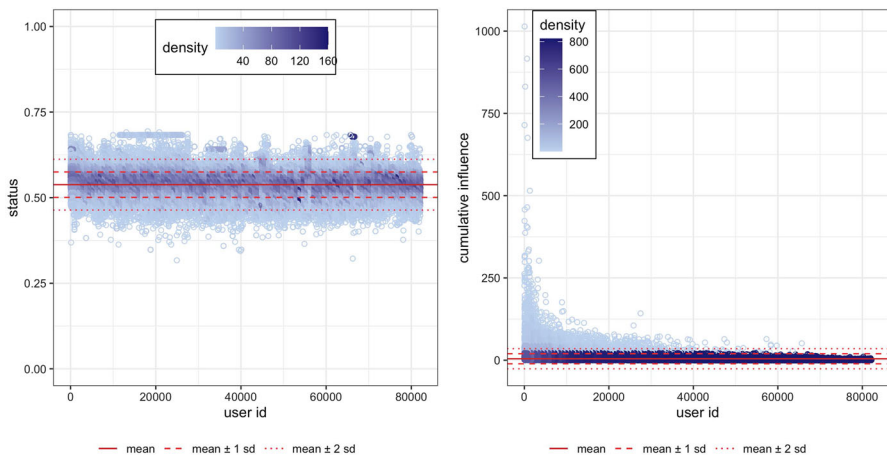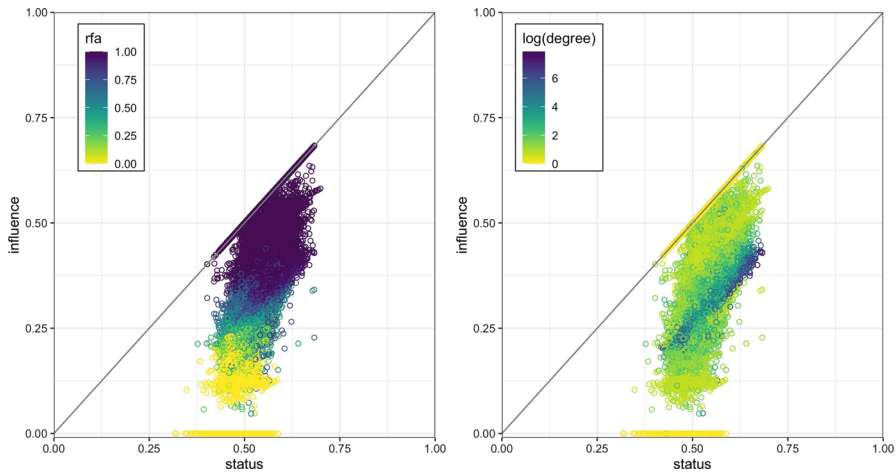

Fig. 25 Slashdot friend-foe network analysis using frustration cloud approach and $N = 1000$ breadth-first spanning trees: status density and influence density

promotion or lack thereof, even if RfA was within guiding limits. The Wikipedia administrator election outcome analysis using the graphB approach allows for a fast, objective snapshot of the outcome, as it allows users to flag nominees whose outcome is not in balance with the rest of the attitudinal network. If the outcome is known, as it is in Wikipedia adminship, graphB is used to flag unexpected outcomes for editor review.

Fig. 26 Slashdot friend-foe network analysis using the frustration cloud approach and $N = 1000$ breadth-first spanning trees: influence versus status by RfA, and influence versus status by log vertex degree

6.2 Slashdot Zoo

Slashdot Zoo is a signed social network with 82,144 users (vertices) and 549,202 edges; 77.4% edges are positive (Leskovec and Krev 2014). Edge direction and weight annotates that the origin user tagged the target user as a friend (weight $+1$) or foe (target -1) (Leskovec and Krev 2014). The largest connected component of this data contains 82,052 (99.89%) users. The type of analysis presented in Sect. 6.1 can be expanded to any attitudinal dataset in light of status and influence, with or without the outcome. There is no outcome for Slashdot Zoo data.

We construct the attitudinal graph from the friend-foe relationships and analyze the status and influence of users. Results are presented in Figs. 25 and 26 , with the frustration cloud and $n = 1000$ breadth-first balance tree discovery. In the Slashdot analysis, we consider vertex degree and remove normalization from Definition 4.4 to analyze cumulative influence. Figure 25 shows the density distribution of the status and cumulative influence over the entire network. The second image in Fig. 25 clearly shows higher influence for early adopters of the network.

Figure 26 illustrates Lemma 4.4 for the influence and status relation. The vertices on the slope status = influence line are a single pendant vertex whose edge is positive (RfA is one, degree is 1); the slope influence = 0 line is a single pendant vertex whose edge is negative (RfA is 0, degree is 1); the influence is always smaller than the status value by Definitions 4.2 and 4.4. The angular outliers corresponding to single-decision outcomes (positive slope 1, and negative slope 0) and the radial outliers are the most/least influential nodes in the network. This influence-status cone analysis allows us to analyze the measurements as a function of node degree. See Fig. 26 colored by node degree. The users with high node degree, overwhelmingly positive votes, mid-range influence, and high status are excellent moderator candidates in this set.
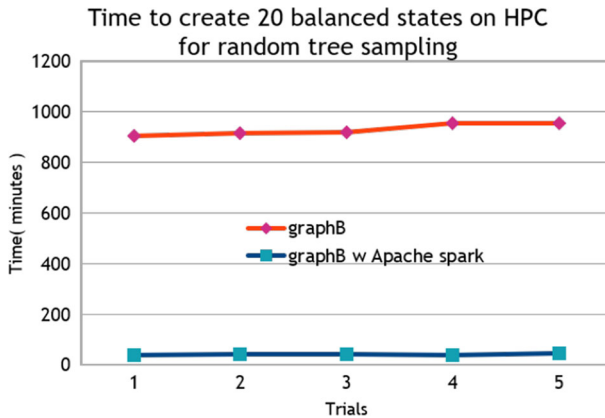
Fig. 27 Timing of balancing graphs using 20 spanning trees and Slashdot data on LEAP cluster

6.2.1 Scaling graphB implementation to large signed graphs

The overall complexity of the implemented graphB algorithm (https://github.com/DataLab12/graphB) is $O(n \cdot v \cdot e)$ for run time and $O(v^2)$ for memory consumption, where n is number of spanning trees, v is number of vertices, and e is number of edges. The Slashdot dataset (Leskovec and Krev 2014) is the largest dataset we have processed to date with over 82,000 vertices. The memory requirement to keep and process $O(v^2)$ matrices required us to upgrade to high memory nodes on HPC (LEAP 2020) to run the code for Slashdot data.

Scaling bottleneck in our graphB implementation (Tešić et al. 2020) was the tree discovering and tree balancing step with $O(n \cdot v \cdot e)$ complexity. The number of discovered cycles is linear with the number of edges and vertices, and the time to balance a graph per spanning tree became prohibitively high. We have implemented Apache Spark parallelization for finding spanning trees and fundamental cycles (as the process is independent for each spanning tree) to utilize the computing cluster and overcome computing issues. The released graphB code (Tešić et al. 2020) allows the user to measure the timing of each step, and with Apache Spark parallelization we have achieved speedup of 22.3 times, as shown in Fig. 27.

6.3 Highland Tribes

The frustration cloud-based approach allows for a more robust way to analyze the perceived outcome in an attitudinal network graph, as it is based on the mathematical sociology model for a balanced system. The Highland Tribes datasets captures the alliance structure of a network of tribes in the Eastern Central Highlands of New Guinea (Read 1954). The network contains sixteen tribes (vertices), and the edges represent agreement ("rova") or animosity ("hina") between two tribes, as illustrated in Fig. 28 with solid lines for agreement and dashed lines for animosity. Read's ethnography portrayed an alliance structure among three tribal groups containing balance as a
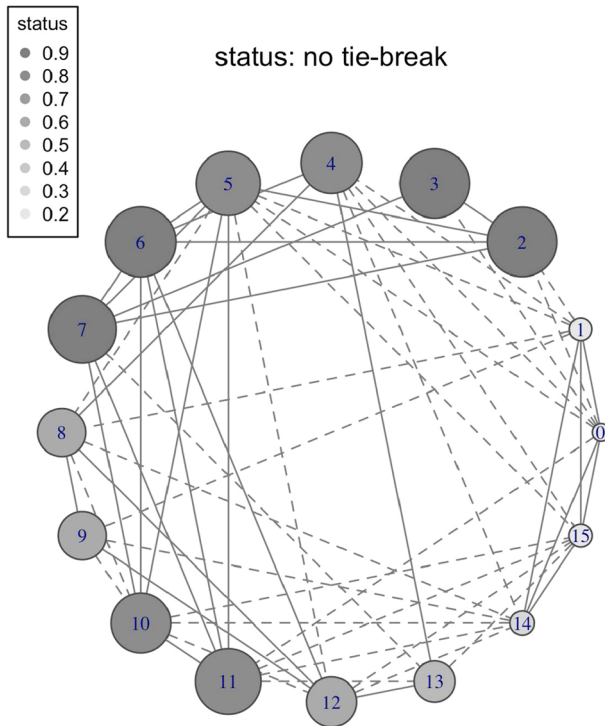
Fig. 28 Highland Tribe status computation for breadth-first sampled 1000 trees: solid circles are tribes, solid lines are agreeable relations, dashed lines are antagonistic relations between two tribes, the size of the vertex circle illustrates computed status for that circle

special case, as the enemy of an enemy can be either a friend or an enemy (Hage and Harary 1983). There are 16 vertices (tribes) and 58 signed edges (tribe relations): 29 positive (sign +1) and 29 negative (sign −1). The signed graph Σ for the Highland Tribes dataset is constructed by adding the two provided matrices.

6.3.1 Experiment: vertical status

The Highland Tribes graph has 402,506,278,163 spanning trees, and we sample 1000 spanning trees using the breadth-first approach for this experiment. Highland Tribes relations in Fig. 28 separate two groups of tribes. The gray shade and size of the vertex circle correspond to the computed status per Definitions 4.2 and 4.6 vertices 0, 1, 14, and 15 form a smaller group, and it is reflected in the lowest status scores for those 4 vertices and lower overall status. This *agrees* with the spectral clustering analysis in Sect. 6.3.2. We also examine the Conservation Law of Controversy from Sect. 4.3 by examining tie-break rule changes and calculating the status of the vertices: one for vertex 0, which has minimum status, and one for vertex 6, which has maximum status. These new status values represent the hypothetical maximum that vertex 0 or vertex 6 may achieve. The corresponding temperature graph shows how the vertical status
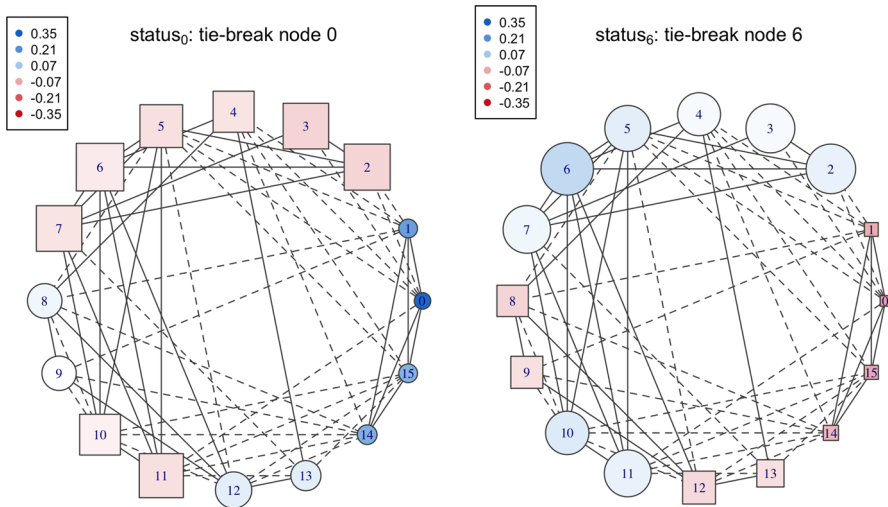
Fig. 29 Highland Tribe verticial status computation for breadth-first sampled 1000 trees. Left: vertex 0 breaks ties; right: vertex 6 breaks ties. Blue circles represent an increased status and red squares represent a decreased status relative to the original status (Color figure online)

maximizes status for the selected vertex and connected vertices in Fig. 29. Figure 29 (left) illustrates the change in status when the tie-break node is from the smaller cluster. While the status of all 4 nodes in that community grows significantly at the expense of the reduced status of vertices in the majority group, Fig. 28 (right) illustrates the maximization of the status in the majority group if vertex 6 is selected as a tie breaker.

Figure 30 illustrates the status difference per vertex ID for each of $status_6$, $status_0$, and $status$. The solid black line demonstrates the Conservation Law, as *controversy* for the Highland graph is constant at 10/16. Also, the status/influence correlation for the Highland dataset has an $R^2 = 0.81$. This may indicate an isolated system, as demonstrated in Figs. 15 and 17 on the Wikipedia dataset—the correlation between status and influence is high when restricted to just the nominees. This means there is no tribe outside of the system acting in a supervisory role.

6.3.2 Experiment: signed spectral clustering versus frustration cloud

Figure 30 shows influence as a function of vertex ID, and the same 4 nodes have the lowest computed influence under each tie-break scenario. An interesting observation on nodes 4, 6, and 7 is that they all have high status; the influence of vertex 4 (*Nagam* tribe) is less than its status in the network, and the influence of vertices 6 and 7 (*Masil* and *Ukudz* tribes) is higher than their status in the network. graphB analysis provides a simple, unbiased view into Highland data and flags 3 out of 16 tribes to be re-examined more deeply by anthropology experts. The clusters in Fig. 31 are calculated only using positive edge spectral clustering to detect nearly-connected non-adversarial relationships. We demonstrate that status is a spectrum of spectral clustering, as anticipated in Sect. 5. The circled blue cluster is the same as the low status group we originally iden-
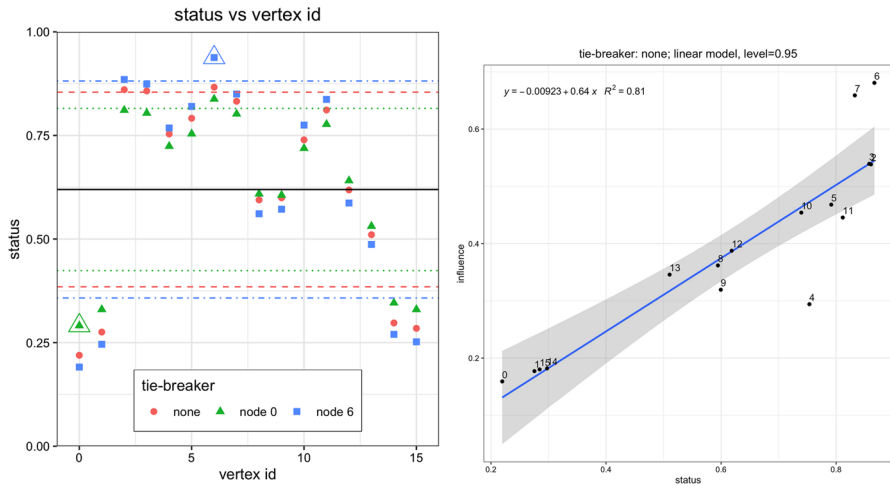
Fig. 30 Left: the status of each vertex for no-tie-break (red circle), vertex 0 breaks ties (green triangle), and vertex 6 breaks ties (blue square). Tie-break vertices are outlined with an open triangle. Right: Highland Tribes status/influence correlation for no 0 tiebreak (Color figure online)
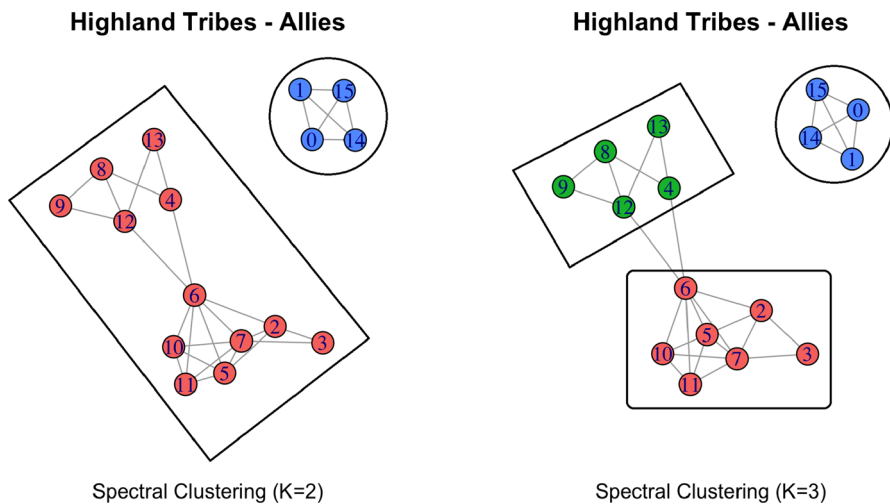

Fig. 31 Spectral clustering of Highland Tribes data using positive edges for $k = 2$ and $k = 3$

tified. The inclusion of the negative edge information for meaningful signed spectral clustering must be examined (Kunegis et al. 2010), especially in highly adversarial networks. For a dataset like Highland, signed spectral clustering (Kunegis et al. 2010) produces the same result for $k = 2$ and $k = 3$, as illustrated in Fig. 32.

Figure 32 marks vertices by cluster belongings (color and shape) in status/influence space. Clusters are computed using signed spectral clustering implementation, and it is clear that status and influence capture spectrum of spectral clustering, as illustrated in Fig. 32. This experiment indicates that the robustness of our status/influence model
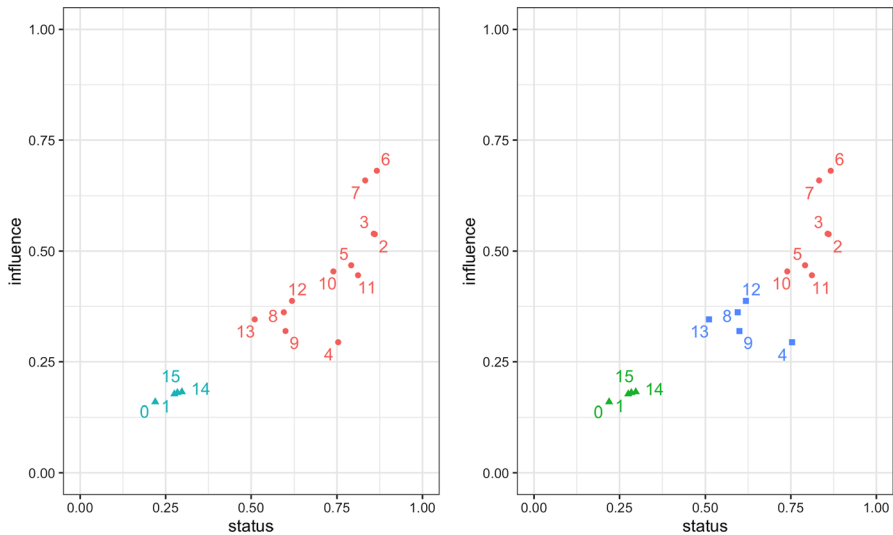
Fig. 32 Signed spectral clustering (Kunegis et al. 2010) using symmetric Laplacian results for Highland Tribes data for $k = 2$ and $k = 3$ in status/influence space demonstrate correspondence of radial distance in status/influence space to clustering. Each clusters is represented with a different shape/color

means we do not need the matrix or the eigenvalues to cluster vertices. Moreover, here is no need to specify k for spectral clustering, as the status/influence cone groups nodes in 2-D space. A study on more degenerate, adversarial networks is necessary to determine if status can provide insight into networks where spectral clustering fails.

7 Conclusion and future work

In this paper, we propose consensus-based quantification of the vertices and edges in a graph. Nearest balanced states of a given network model attitudinal strength and influence in the network through various measures of a network's ability to reach a balanced state with a minimal number of edge sign changes. We introduce the concept of "frustration cloud" and vertex scores that capture vertex relation to the nearest balanced state of the network. The frustration cloud replaces the necessity of determining a balanced state with the least number of sentiment changes by determining a set of nearest states with minimal sentiment disruption. This also trades the NP-hardness of the frustration index for determining fundamental cycles and spanning trees.

We introduce new metrics that quantify the importance of each balanced state relative to the likelihood it will be become the consensus state. The tree-search methods are shown to provide resolution to the data, while the quality of the resolution appears to be indistinguishable beyond $n = 1000$ spanning trees, as seen in Figs. 19 and 20. The measures of vertex status and influence both demonstrate differences in power dynamics when they are not correlated, as seen in Figs. 15 and 17. These vertex scores provide an alternative to examine existing promotional practices, as indicated in Fig. 21, and flag anomalous users.

The status/influence cone provides a view of fairness and power for the data as shown in Fig. 23. The social network at scale produces a different shaped status/influence cone in Fig. 26 that indicates a large separation of power as status remains relatively constant when compared to influence. The smaller dataset of Highland Tribes demonstrates quickly reproducible results for the Conservation Law of Controversy (average status is constant regardless of tie-break node) in Fig. 30, as well as the efficacy of our proposed method for spectral clustering in Fig. 32.

Consensus modeling has gained traction as a way to model agreement in multi-agent networks in the presence of antagonistic interactions (Altafini 2013; She et al. 2020). We plan to apply proposed balancing theory to a multi-agent network agreement to see if we can identify and tune existing policies and detect biased agents (AI modules) in the network. Future research will also focus on comparison of proposed approach to signed and weighted sign graph spectral clustering (Aref et al. 2016; Mercado 2019). We are developing metrics to measure the strength of vertex and edge interactions, a measure that utilizes known promotional outcomes to detect and quantify bias for the majority/minority, and a toolkit for deeper analysis of the proposed metrics.

References

Abelson RP, Rosenberg MJ (1958) Symbolic psycho-logic: a model of attitudinal cognition. Behav Sci 3(1):1–13

Alava MJ, Duxbury PM, Moukarzel CF, Rieger H (2001) Exact combinatorial algorithms: ground states of disordered systems. In: Domb C, Lebowitz JL (eds) Phase transitions and critical phenomena, vol 18. Academic Press, San Diego, pp 143–317

Alemzadeh S, de Badyn MH, Mesbahi M (2017) Controllability and stabilizability analysis of signed consensus networks. In: 2017 IEEE Conference on control technology and applications (CCTA), pp 55–60

Altafini C (2013) Consensus problems on networks with antagonistic interactions. IEEE Trans Autom Control 58(4):935–946

Altafini C (2019) A dynamical approach to privacy preserving average consensus. In: 2019 IEEE 58th Conference on decision and control (CDC), pp 4501–4506

Aref S, Mason AJ, Wilson MC (2016) An exact method for computing the frustration index in signed networks using binary programming, pp 1–30. CoRR, arXiv:1611.09030

Aref S, Mason AJ, Wilson MC (2020) A modeling and computational study of the frustration index in signed networks. Networks 75(1):95–110

Barahona F (1982) On the computational complexity of Ising spin glass models. J Phys A Math Gen 15:3241–3253

Berge C (1970) Sur certains hypergraphes généralisant les graphes bipartites. In: Combinatory theory and its applications, I (Proceedings of Colloquium, Balatonfüred, 1969). North-Holland, Amsterdam, pp 119–133

Buekenhout F, Parker M (1998) The number of nets of the regular convex polytopes in dimension ≤ 4. Discrete Math 186(1):69–94

Burke M, Kraut R (2008) Mopping up: modeling wikipedia promotion decisions. In: Proceedings of the 2008 ACM conference on computer supported cooperative work, CSCW '08. ACM, pp 27–36

Cartwright D, Harary F (1956) Structural balance: a generalization of Heider's theory. Psychol Rev 63:277–293

Chen X, Lijffijt J, De Bie T (2018) Quantifying and minimizing risk of conflict in social networks. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '18, pp 1197–1205

Davis JA (1967) Clustering and structural balance in graphs. Hum Relat 20(2):181–187

Deo N, Prabhu G, Krishnamoorthy MS (1982) Algorithms for generating fundamental cycles in a graph. ACM Trans Math Softw (TOMS) 8(1):26–42

Esmailian P, Jalili M (2015) Community detection in signed networks: the role of negative ties in different scales. Nat Sci Rep 1(5):14339

Farina G (2015) A linear time algorithm to compute the impact of all the articulation points. arXiv:1504.00341 [cs.DS]

Garimella K, De Francisci Morales G, Gionis A, Mathioudakis M (2017) Reducing controversy by connecting opposing views. In: Proceedings of the tenth ACM international conference on web search and data mining, WSDM '17, pp 81–90

Guha R, Kumar R, Raghavan P, Tomkins A (2004) Propagation of trust and distrust. In: Proceedings of the 13th international conference on world wide web, WWW '04. ACM, pp 403–412

Hagberg A, Swart P, Chult DS (2008) Exploring network structure, dynamics, and function using networkX. https://networkx.github.io/documentation/stable/reference/algorithms/index.html

Hage P, Harary F (1983) Structural models in anthropology. Cambridge University Press, S. Cambridge, pp 1–220

Harary F (1953) On the notion of balance of a signed graph. Michigan Math J 2(2):143–146

Harary F (1959) On the measurement of structural balance. Behav Sci 4:316–323

Hartnett T (2011) Consensus-oriented decision-making: the CODM model for facilitating groups to widespread agreement. New Society Publishers, Gabriola Island, pp 1–192

Heider F (1946) Attitudes and cognitive organization. J Psychol 21:107–112

Hu J, Zheng WX (2013) Bipartite consensus for multi-agent systems on directed signed networks. In: 52nd IEEE Conference on decision and control, pp 3451–3456

Hunter JE, Danes JE, Cohen SH (1984) Mathematical models of attitude change: change in single attitudes and cognitive structure. Academic Press, London, pp 1–356

Javed MA, Younis MS, Latif S, Qadir J, Baig A (2018) Community detection in networks: a multidisciplinary review. J Netw Comput Appl 108:87–111

Jiang Y, Zhang H, Chen J (2016) Sign-consensus of linear multi-agent systems over signed graphs using a fully distributed protocol. In: 2016 IEEE 55th Conference on decision and control (CDC), pp 3537–3541

Kumar S, Hamilton WL, Leskovec J, Jurafsky D (2018) Community interaction and conflict on the web. In: Proceedings of the 2018 world wide web conference, WWW'18, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee, pp 933–943

Kunegis J, Schmidt S, Lommatzsch A, Lerner J, De Luca EW, Albayrak S (2010) Spectral analysis of signed graphs for clustering, prediction and visualization. In: Proceedings of the 2010 SIAM international conference on data mining, pp 559–570

Leskovec J, Krev AL (2014) SNAP datasets: stanford large network dataset collection. http://snap.stanford.edu/data

Leskovec J, Huttenlocher D, Kleinberg J (2010a) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '10, pp 1361–1370

Leskovec J, Huttenlocher D, Kleinberg J (2010b) Predicting positive and negative links in online social networks, In: Proceedings of the 19th international conference on world wide web, WWW '10. ACM, pp 641–650

Li KW, Kilgour DM, Hipel KW (2005) Status quo analysis in the graph model for conflict resolution. J Oper Res Soc 56(6):699–707

Lovász L, Pelikán J (1973) On the eigenvalues of trees. Period Math Hung 3:175–182

Lu L, Zhou T (2011) Link prediction in complex networks: a survey. Phys A Stat Mech Appl 390(6):1150–1170

Mercado P, Tudisco F, Hein M (2019) Spectral clustering of signed graphs via matrix power means. In: Proceedings of the 36th international conference on machine learning, vol 97, pp 4526–4536

Mishra A, Bhattacharya A (2011) Finding the bias and prestige of nodes in networks based on trust scores. In: Proceedings of the 20th ACM international conference on world wide web (WWW), pp 567–576

Pan L, Shao H, Mesbahi M (2016) Laplacian dynamics on signed networks. In: 2016 IEEE 55th Conference on decision and control (CDC), pp 891–896

Rashevsky N (1949) Mathematical theory of human relations: an approach to mathematical biology of social phenomena, 2nd edition. Principia Press, Bloomington, pp 1–202

Read K (1954) Cultures of the central highlands, New Guinea. Southwest J Anthropol 10(1):1–43

Ruby, Kaur I (2017) A review of community detection algorithms in signed social networks. In: 2017 International conference on energy, communication, data analytics and soft computing (ICECDS), pp 413–416

Rusnak LJ (2013) Oriented hypergraphs: introduction and balance. Electron J Combin 20(3)(#P48):1–29

Rusnak LJ, Robinson E, Schmidt M, Shroff P (2018) Oriented hypergraphic matrix-tree type theorems and bidirected minors via Boolean ideals. J Algebr Combin 49:1–13

Russell S, Norvig P (2009) Artificial intelligence: a modern approach, 3rd edn. Prentice Hall Press, Hoboken, pp 1–1114

Selbst AD, Boyd D, Friedler S, Venkatasubramanian S, Vertesi J (2018) Fairness and abstraction in sociotechnical systems. In: Proceedings of ACM conference on fairness, accountability, and transparency, pp 59–68

Sethna JP (2006) Statistical mechanics: entropy, order parameters, and complexity, volume 14 of master series in physics. Oxford University Press, Oxford, pp 1–349

She B, Mehta S, Ton C, Kan Z (2020) Controllability ensured leader group selection on signed multiagent networks. IEEE Trans Cybern 50(1):222–232

Sontag E, Veliz-Cuba A, Laubenbacher R, Jarrah AS (2008) The effect of negative feedback loops on the dynamics of Boolean networks. Biophys J 95(2):518–526

Tang J, Chang Y, Aggarwal C, Liu H (2016) A survey of signed network mining in social media. ACM Comput Surv 49(3):1–37

Tešić J, Mitchell J, Hull E, Tomasso M, Rusnak LJ (2020) graphB: Python software package for graph analysis at scale. https://github.com/DataLab12/graphB

Texas State University Division of Information Technology (2020) Leap—high performance computing cluster. https://doit.txstate.edu/rc/leap.html

Tutte WT (1984) Graph theory, volume 21 of encyclopedia of mathematics and its applications. Addison-Wesley Publishing Company Advanced Book Program, Reading, pp 1–360. With a foreword by C. St. J. A. Nash-Williams

Wasserman S, Faust K (1994) Social network analysis: methods and applications, vol 8. Cambridge University Press, Cambridge, pp 1–857

Wikipedia (2020) Requests for Adminiship, Requests for adminship (RFA). https://en.wikipedia.org/wiki/Wikipedia:Requests_for_adminship

Wu BY, Chen JF (2013) Balancing a complete signed graph by editing edges and deleting nodes. In: Chang R-S, Jain LC, Peng S-L (eds) Advances in intelligent systems and applications—volume 1. Springer, Berlin, pp 79–88

Yoshikawa T, Iino T, Iyetomi H (2011) Market structure as a network with positively and negatively weighted links. In: Watada J, Phillips-Wren G, Jain LC, Howlett RJ (eds) Intelligent decision technologies. Springer, Berlin, pp 511–518

Yuan W, He K, Guan D, Han G (2017) Edge-dual graph preserving sign prediction for signed social networks. IEEE Access 5:19383–19392

Zaslavsky T (1982) Signed graphs. Discrete Appl Math 4(1):47–74. MR 84e:05095a. Erratum, ibid. 5 (1983), 248. MR 84e:05095b

Zaslavsky T (2021) A mathematical bibliography of signed and gain graphs and allied areas. Electron J Combin #DS8, 10th edn, pp 1–606

Zhao X, Liu X, Chen H (2018) Network modelling and variational Bayesian inference for structure analysis of signed networks. Appl Math Modell 61:237–254

Zhou J, Li L, Zeng A, Fan Y, Di Z (2018) Random walk on signed networks. Phys A Stat Mech Appl 508:558–566