Social and Information Networks

University of Toronto CSC303
Winter/Spring 2022

Week 4: Jan 31-Feb 4 (2022)

This week's agenda

Last week we finished discussing Chapter 4 of the text. We discussed the probability of triadic closure (resp. focal closure, membership closure) as a function of the number of common friends (resp. the number of common interests (foci), the number of friends in a given focus) and the Schelling's segregation model. We've covered all the material needed for the first assignment.
This week:

- Chapter 5 and structural balance
- Strong structural balance and weak structural balance.

This week's agenda

- Structural Balance
  - Balanced triangles
  - Strongly balanced networks
  - Strong balance theorem
  - Weak structural balance
  - The signed Laplacian matrix

Critical Review

- Rubric & two examples are on the course website
- Groups of 3-4
- You will be critically reviewing a paper
- **The paper must be recent** (i.e. published on or after January 1st 2019)
- **The paper must be either published in a journal/conference, or have been accepted to be published in a journal/conference**
- Why no arXiv preprints?

Comic from xkcd

Critical Review

- Email me your group & choice of paper by March 5th (use the email on Quercus/syllabus/course website)
  - If I don't approve your paper, then you're getting a zero
  - I'll confirm receipt once I see it, and I'll try to write back to approve or reject your choice within 3 days
- The report is due via MarkUs by March 25th
- We'll now review the rubric
- You have an upper limit of 5 pages, but you're free to make it long or short as you feel is appropriate (the exemplars are around 1000 words, thereabouts or a bit longer is definitely reasonable)

Structural balance: positive and negative links

Now for a new topic, who we like and dislike. As previously mentioned so far we have restricted attention to social networks where all edges reflect some positive degree of friendship, collaboration, communication, etc.

Chapter 5 now explores some interesting aspects of networks where edges can be both positive and negative. This is, of course, quite natural in that people (countries) often have enemies as well as friends (allies). We also have companies that can be aligned in some way or can be competitors.

Following the development stemming from the distinction between strong and weak ties, we would like to see what we can infer about a network given that some edges are positive and some are negative. More specifically, what can be assumed from certain types of triadic closures? How can local properties (e.g., how edges of a triangle are labeled) can have global implications (i.e., provable results about network structure)?

Some initial assumptions

We start with a strong assumption:
Assume the network is a complete (undirected) graph. That is, as individuals we either like or dislike someone. Furthermore, this is not nuanced in the sense that there is no differentiation as to the extent of attraction/repulsion).

[Image modified from *Star Wars: Episode III - Revenge of the Sith*. Directed by George Lucas, Lucasfilm, 2005]

Some initial assumptions

Later in the chapter, the text considers the issue of networks that are not complete networks. The text also reflects a little on the nature of directed networks (when discussing the *weak balance property*) but essentially this chapter is about undirected networks.

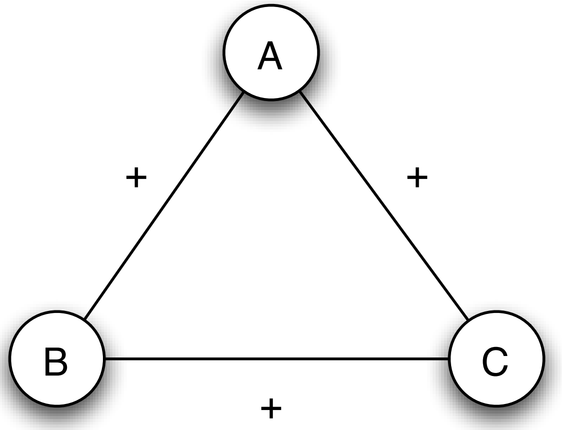
Note: For non-complete networks, we can assume the graph is connected since otherwise we can consider each connected component separately.

Types of instability

Thinking of networks as people with likes and dislikes of other people (rather than some other possible interpretations), we can consider the 4 different types of labelled triangles in the graph, depending on the number of positive (+) and negative (-) edges. That is, any completely labelled triangle can have 0,1,2, or 3 positive edges and due to the symmetry of a triangle that is all the information we have about any particular triangle.

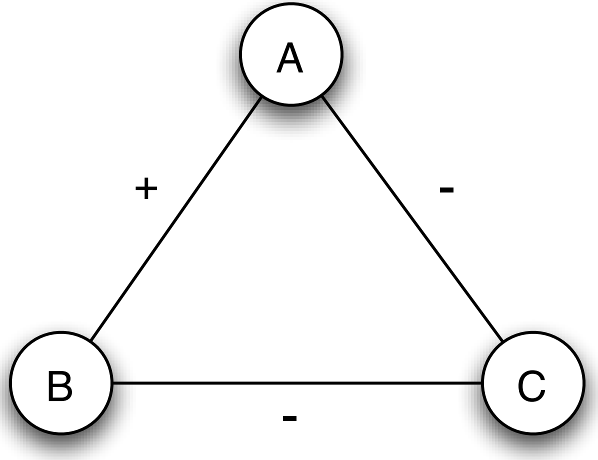
Using a central idea from social psychology, two of the four triangle labellings are considered relatively stable (called *balanced*) and the other two relatively unstable (*not balanced*).
Here follows the four types of triangles as depicted in Figure 5.1 of the text:

A natural stable configuration



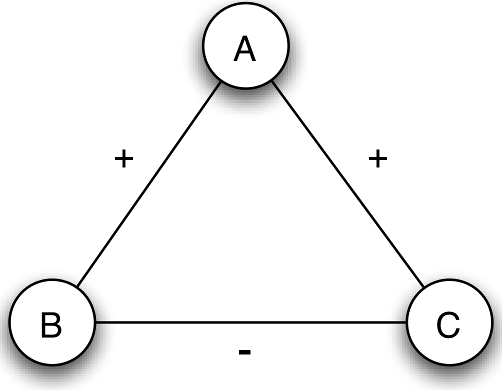
In this case, A, B, C are mutual friends and that naturally indicates that they would likely remain so.

The second stable configuration



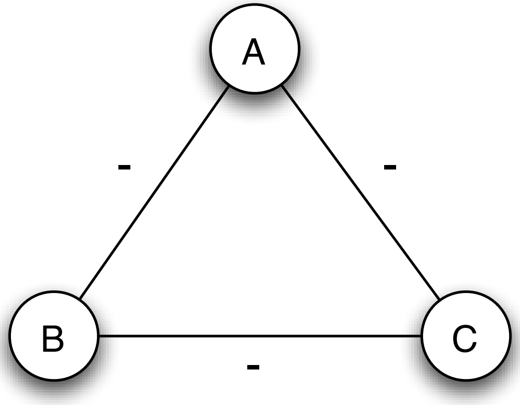
This may be a slightly less obvious stable situation where A and B are friends and if anything that friendship is reinforced by a mutual dislike for C.

A natural unstable configuration



In this case, A has two friends B and C who unfortunately do not like each other. The claim here is that the stress of this situation will encourage A to either try to have B and C become friends or else for A to take sides with B or C and thus eliminate a friendship so as to move toward the previous stable configuration.

A somewhat less obvious unstable configuration



Why is this called unstable? The instability here is sometimes explained by the phenomena that "the enemy of my enemy becomes my friend" as we sometimes see in international relations. This is less convincing than the other type of instability and we will return to this situation soon.

The strong structural balance property

The underlying behavioural theory is that these unstable triangles cause stress and hence the claim that such unbalanced triangles are not common.

In order to try to understand if this theory tells us anything about the global structure of the network, we can make the following strong balance assumption (much as we made the strong triadic closure assumption).

Strong structural balance property: Every triangle in the network is balanced.

Recall that we started off with the assumption that the network is a complete graph with every edge labelled so we are assuming a property for all n choose 3 triangles. Of course, we cannot expect this property to hold but just as the strong triadic closure property was an extreme assumption, we can hope that this strong assumption will also suggest or predict useful information about the network.

Balance as a form of equilibrium

One way to further justify the distinction between balanced and unbalanced triangles is to view balance (resp non balance) as a desirable (undesirable) situation.

In a balanced (resp. unbalanced) configuration, any single change in a relation (i.e. edge label) will lead to an unbalanced (resp. balanced) configuration.

That is, balance is a form equilibrium.

Later in the term, we will discuss stable matchings. (How many have seen this in CSC304 or elsewhere?) We view stable matchings as an equilibrium. In stable matchings (as in balanced triangles), it is a pair of "agents" that we consider in a single change. We discuss stable matchings later in this course.

Consequence of the strong structural balance property: A provable characterization of networks that satisfy the property

One simple (idealistic) way to construct a network satisfying the property is to assume that that there are no enemies; everyone is a friend. Is this the only way?

Suppose that we had two communities of active political people (e.g. $X =$ the "base" for candidate or political party R, and Y and the "base" for candidate or political party B. In the world of highly politicized politics, it isn't too far of a stretch to think that everyone within a community are friends and everyone dislikes people in the other community. This kind of network would also clearly satisfy the property.

So far then, we have two possibilities, the network is a clique with all positive edges, or the network is composed of two positive cliques with a complete bipartite graph of negative edges between the communities. Are there other possible ways to have the strong balance property?

Harary's Balance Theorem

Are there other possible ways to have the strong balance property?

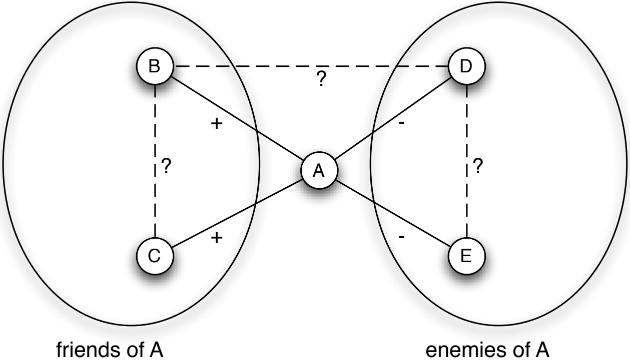
Perhaps surprisingly, in a complete network, these two types of networks (no enemies and two opposing communities) are the only possibilities.

This is a theorem and the proof is not difficult as we will show using the figure 5.4 in the text.

Proof
We assume that the network satisfies the strong balance property. If there are no enemies, then we are done. So suppose there is at least one negative edge and for definiteness lets say that edge is adjacent to node A. Let X be all the friends of A and Y all of its enemies. So every node is in either X or Y since every edge is labelled.

Proof of balance theorem continued

Consider the three possible triangles as in the figure. It is easy to see that in order to maintain structural balance, B and C must be friends as must D and E, whereas B and D (also C and E) must be enemies.



Strong structural balance in networks that are not complete

We will depart from the order of topics in chapter 5, and consider the issue of networks that are not complete. Is there a meaningful sense in which a (non-complete) network is or is not structurally balanced?

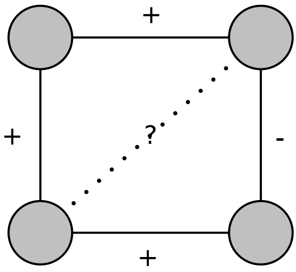
One possibility is to ask whether or not there is a way to complete the graph so that it becomes structurally balanced. Of course, if there is already an unbalanced triangle then there is no way to complete the graph into one satisfying the strong balance property.

Aside: Of course, this immediately raises the question as to how many existing edge labels need to be changed so that a complete network is balanced (or an incomplete network can be made to be balanced)? And will networks tend to dynamically evolve into balanced networks. But for now we will assume that all existing labels are permanent.

How to label missing edges?

Note that when considering the strong triadic property, if all existing triangles satisfied the strong triadic property, then there was always a trivial way to assign labels to unlabelled edges by simply making each unlabelled edge a weak link.

Question: If all existing triangles are balanced, is there always a way to complete a network so as to form a strongly balanced network?



How to label missing edges?

It is easy to see that this is not always possible. For example, consider a network which is a 4 node cycle having 3 positive edges and one negative edge. Any way to label a "diagonal edge" will lead to an imbalance.

We are then led to the following
Question: Can we determine when there is an efficient algorithm to complete the network so as to satisfy the strong balance property? And if there is a completion, how efficiently can one be found?

Determining when and how to complete a network to satisfy the strong balance property

Clearly, if the existing edges are all positive links then there is a trivial way to complete the graph by simply making all missing edges to be positive edges.

So the interesting case is when there are existing negative edges. In this case, the characterization of strongly balanced networks tells us that when the graph is completed, the graph structure must be that of two opposing communities, with only positive edges within each community and only negative edges for links between the communities.

The previous example of a 4 node cycle is a clue as to how to proceed. That example can be stated as follows: if a network contains a 4 node cycle with one negative edge then it cannot be completed (to be strongly balanced) . More generally, if a network contains a cycle (of any length) with one negative edge, it cannot be completed. And even more generally, if a network contains a cycle having an odd number of negative edges it cannot be completed. Why?

Consequence of an odd cycle

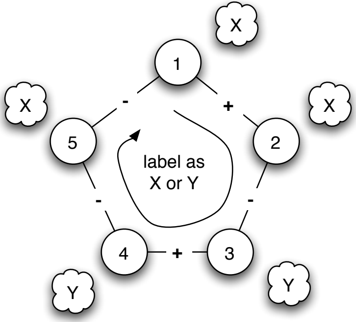


Figure 5.10: If a signed graph contains a cycle with an odd number of negative edges, then it is not balanced. Indeed, if we pick one of the nodes and try to place it in X, then following the set of friend/enemy relations around the cycle will produce a conflict by the time we get to the starting node.

The algorithm for determining if a partially labelled network can be completed to the strongly balanced

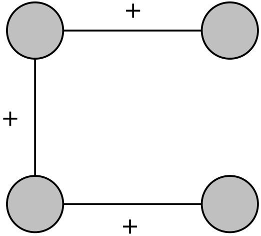
Lets call a cycle with an odd number of edges an odd cycle. The desired algorithm will either find an odd cycle (certifying that the network cannot be completed) or it will return a bipartiton of the nodes satisfying the Balance Theorem. This then also determines if a complete network is balanced.

We proceed as follows:

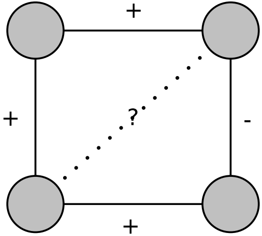
- Suppose $G = (V, E)$ is the given connected network and let $G^+ = (V, E^+)$ where $E^+ = \{e \in E$ such that e is a positive link.$\}$
- We consider the connected components $\mathcal{C} = C_1, \ldots, C_r$ of G^+.
- Note that all edges between any C_i, C_j must be labelled as negative edges (or else they would have been merged into a larger connected component in G^+).
- For every C_i, we must check if there is a negative edge between two nodes in C_i. If so then there is a cycle in C_i with one negative edge, and hence C_i (and thus G) cannot be completed.

The algorithm for determining if a partially labelled network can be completed to the strongly balanced

Connected positive component $\mathcal{C}_i$



Negative edge produces an odd cycle



Completing the algorithm

- Otherwise, consider the graph $G^- = \{\mathcal{C}, E^-\}$ whose nodes are the components of G^+ and whose edges are negative edges in G.
- Since G is connected, G^- is connected.
- if G^- has a cycle with an odd number of negative edges, then by following positive edges in each C_i we have such a cycle in G. We then again have a witness that G cannot be completed.
- Otherwise we are showing that G^- is bipartite and this gives us the bipartition we need for the balance theorem.
- A graph has an odd cycle iff the graph is not bipartite. Breadth first search can be used to determine whether or not a graph is bipartite (equivalently has a 2-colouring). Hence this development is efficient.

We now return to the assumption that our networks are undirected complete graphs.

Friends-enemies vs trust-distrust

There is always an ambiguity in social networks as to how to interpret links. Is a friend as we might traditionally mean a "good friend", or is it a friend as in Facebook friend which is often an acquaintance? And as we have seen we also use social network links to mean collaboration or communication rather than friendship.

This is both the power of network modeling (i.e., that results can carry over to different settings) and also the danger of misinterpreting results for one type of setting to apply to another.

In chapter 5, we see another instance of the ambiguity where instead of the friend-enemy relation, one can interpret an edge label as a trust-distrust relation.

To what extent should we expect intuition for friendship to carry over to trust? As discussed in the text, one distinction between these settings is that trust may be more of a directed edge concept relative to friendship. (Of course, even for friendship the relation may not be symmetric which is why maybe we should reserve the term of "friend" for a good friend.)

The ambiguity in the trust-distrust relation

Ignoring the fact that trust might not be at all symmetric, there is an additional ambiguity in the trust-distrust terminology. Namely, the text considers two possible interpretations that are meaningful even in the context of a simple setting as in the online product rating site Epinions.

1. If trust is aligned with agreement on polarized political issues, then the four cases of balanced and unbalanced triangles still seem to apply. In particular, if A distrusts B and B distrusts C, it is reasonable to assume that A trusts C and hence a triangle having three negative labels is not stable.
2. However, if A distrusts B is aligned with A believing that he/she is more knowledgeable than B about a certain product, then a triangle having three negative labels is stable.

This suggests that it is reasonable to study a weaker form of structural balance.

A weaker form of structural balance

It is then interesting to consider a weaker form of structural balance where the only unstable triangles are those having two positive labels.

Definition (Weak Structural Balance)

A network satisfies the *weak structural balance property* if it does not contain any triangles with exactly two positive edges.

Question: Is there a characterization of which (complete) networks satisfy the weak structural balance property?

Since every network that satisfies the strong balance property must also satisfy the weak balance property, the characterization of strongly balanced networks must be a special case of weakly balanced networks. Indeed we have the following characterization:

Theorem: A network $G = (V, E)$ satisfies the weak structural balance property iff $V = V_1 \cup V_2 \dots V_r$ such that all edges within any V_i are positive edges and all edges between V_i and V_j $(i \neq j)$ are negative edges.

Proof of the characterization of weak structural balance

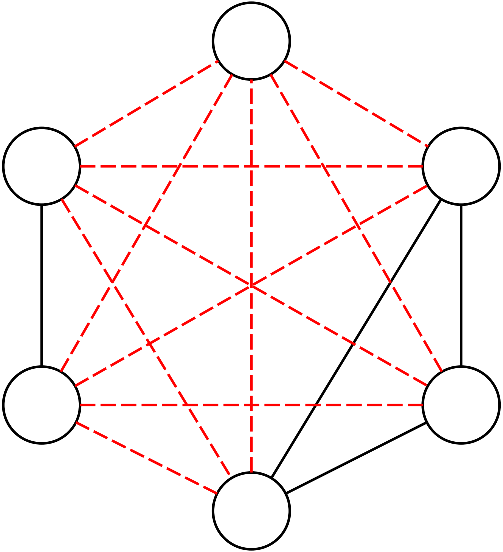
Clearly if the network $G = (V, E)$ has the network structure specified in the Theorem, then the network satisfies the weak balance property. The converse (that the weak balance property implies the network structure) is a reasonably simple inductive argument (say with respect to the number of nodes).

Consider any node A and let X be all the friends of A.
The following two claims are easy to verify:

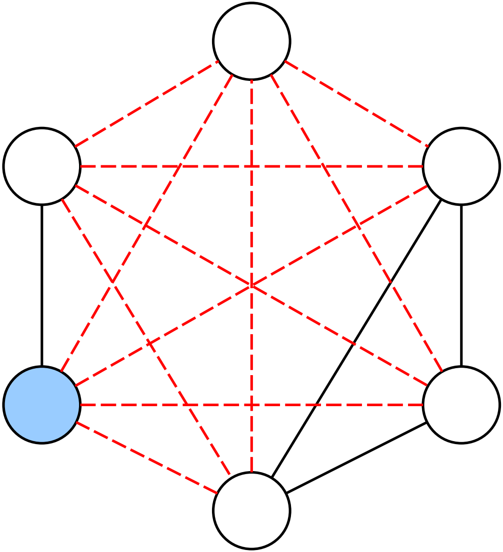
- Any $B, C \in X$ are friends
- If $B \in X$ and $D \notin X$, then B and D are enemies.

Upon removing the nodes in X, the induced network G' of the remaining nodes still must satisfy the weak structure balance property and hence by the induction hypothesis must have the stated network structure.

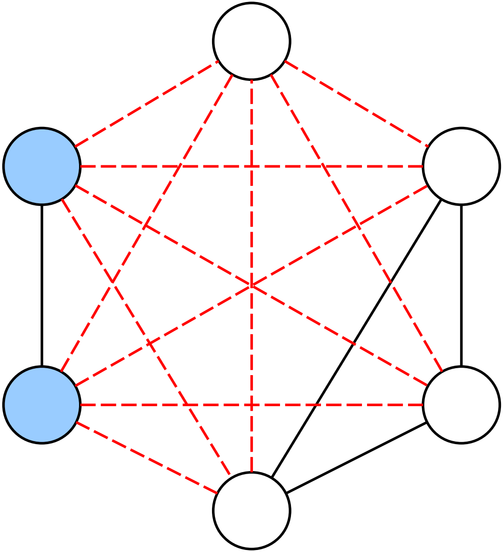
Example: Partitioning a weakly balanced graph



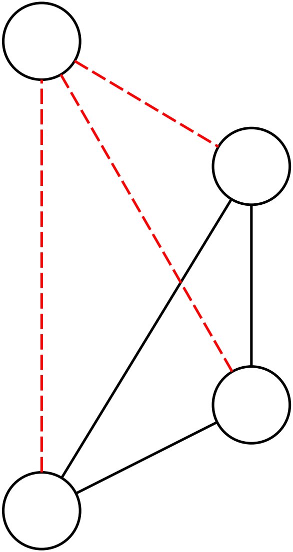
Example: Partitioning a weakly balanced graph



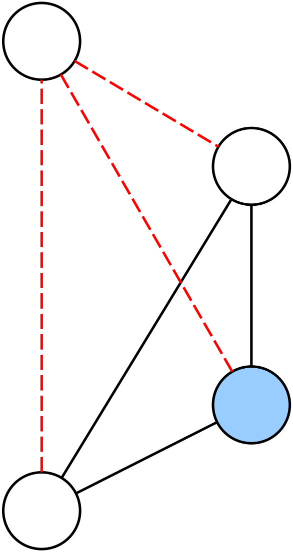
Example: Partitioning a weakly balanced graph



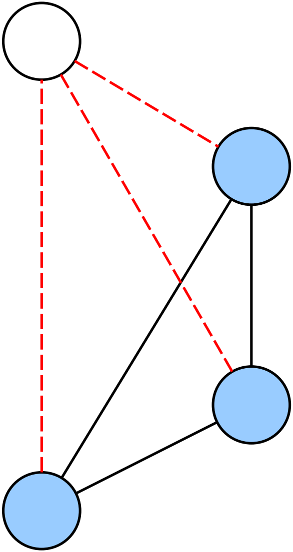
Example: Partitioning a weakly balanced graph



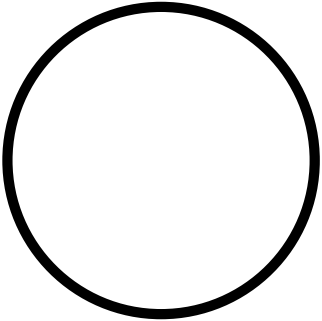
Example: Partitioning a weakly balanced graph



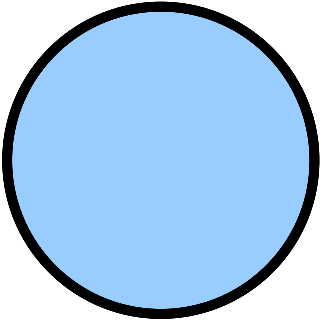
Example: Partitioning a weakly balanced graph



Example: Partitioning a weakly balanced graph

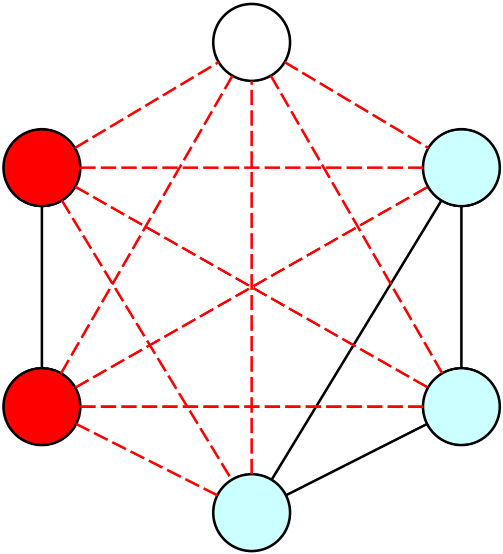


Example: Partitioning a weakly balanced graph

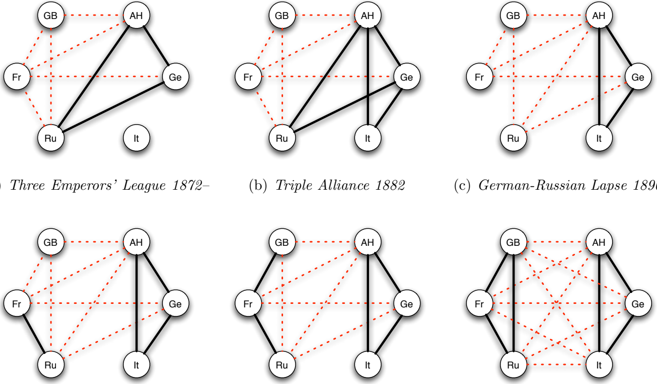


Example: Partitioning a weakly balanced graph

Example: Partitioning a weakly balanced graph



The evolution of European alliances preceding WWI



(a) *Three Emperors' League 1872–81*

(b) *Triple Alliance 1882*

(c) *German-Russian Lapse 1890*

(d) *French-Russian Alliance 1891–94*

(e) *Entente Cordiale 1904*

(f) *British Russian Alliance 1907*

Figure 5.5: The evolution of alliances in Europe, 1872-1907 (the nations GB, Fr, Ru, It, Ge, and AH are Great Britain, France, Russia, Italy, Germany, and Austria-Hungary respectively). Solid dark edges indicate friendship while dotted red edges indicate enmity. Note how the network slides into a balanced labeling — and into World War I. This figure and example are from Antal, Krapivsky, and Redner [20].

Efficiently finding balanced subgraphs

- Real social networks are unlikely to be strongly balanced
- What if we want to find the largest balanced subnetwork?
- Question: Why might we want to do this?
  - Identify opposing blocs in geopolitics
  - Identify polarized communities on social media

Definition (MBS)

Given a signed graph $G = (V, E, w)$, MBS is the problem of finding the *maximum balanced subgraph*. i.e. finding the largest $V' \subseteq V$ such that $G' = (V', \{(v_1, v_2) \in E | v_1, v_2 \in V'\}, w)$ is strongly balanced.

- Problem is NP-Hard, so we have to approximate
- We're going to do this, by studying the properties of the Laplacian matrix

Signed Laplacian Matrix of a Signed Graph

- For our signed graph $G = (V, E, w)$ with n nodes, the Signed Laplacian is:

$$L(G) := D - A$$

- D is the degree matrix:

$$D_{ij} = \begin{cases} |\{a : (v_i, a) \in E\}|, & i = j \\ 0, & else \end{cases}$$

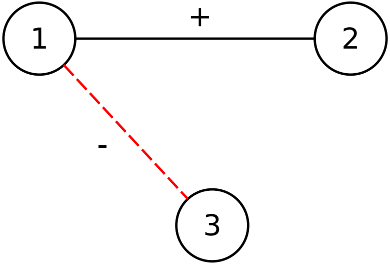
- A is the signed adjacency matrix:

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \ \& \ w((v_i, v_j)) = 1 \\ -1, & (v_i, v_j) \in E \ \& \ w((v_i, v_j)) = -1 \\ 0, & else \end{cases}$$

- Aside: The Laplacian matrix of general edge weighted undirected graphs is $L = D - A$ where D and A are the weighted degree and adjacency matrices respectively. This is a similar but fundamentally different definition than the Signed Laplacian

Signed Laplacian Matrix of a Signed Graph

Consider the following graph G:



$$D = \begin{bmatrix} 2 & & \\ & 1 & \\ & & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad L(G) = D - A = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Properties of the Signed Laplacian

- $L = D - A$, therefore L is a real symmetric matrix
- By Spectral Theorem we therefore have an orthonormal eigenbasis $\mathbf{b}_1, \mathbf{b}_2, \dots \mathbf{b}_n$ with corresponding eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.
  - $L\mathbf{b}_i = \lambda_i \mathbf{b}_i$
  - $\langle \mathbf{b}_i, \mathbf{b}_i \rangle = 1$
  - $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$ for $i \neq j$
- It can also be shown that the signed Laplacian is also positive semi-definite
  - $\forall \mathbf{x} : \mathbf{x}^T L \mathbf{x} \geq 0$

Properties of the Signed Laplacian

From positive semi-definiteness, we know that $\lambda_1 \geq 0$ (Exercise: Prove this!). But why do we care about the eigenvalues of the signed Laplacian?

Theorem

For a signed graph G, let λ_1 be the smallest eigenvalue of the corresponding signed Laplacian, $L(G)$. Then G is strongly balanced iff $\lambda_1 = 0$.

- Furthermore, it can be shown that signed graphs that are "close" to being balanced have "small" values of λ_1

Finding large balanced subgraphs

- We can show that for the Signed Laplacian $L(G)$ with smallest eigenvalue λ_1, then $\lambda_1 = 0$ iff G is strongly balanced
- There is a result indicating that graphs which are “close” to being balanced have “small' values of λ_1
- Question: Assuming that we can compute λ_1 easily, how could we use this to find a large balanced subgraph?

- Greedy approach: Repeatedly remove the nodes that cause the greatest decrease in λ_1 until the graph becomes strongly balanced
- This is the approach used by Ordozgoiti et al. (see `https://arxiv.org/abs/2002.00775`)

Finding large balanced subgraphs

- Let $\lambda_1(M)$ denote the smallest eigenvalue of the matrix M
- As calculating λ_1 is too expensive to be done $|V|$ times per removed node. Ordozgoiti et al. instead calculate $\lambda_1(L(G))$, and approximate λ_1 when choosing which node to remove from G
- Through a simple (but a bit long) derivation, the authors show that:

$$\lambda_1(L^{(i)}) \leq \frac{\lambda_1(L) + (\mathbf{b}_1)_i^2(d(i) - 2\lambda_1(L(G))) - \sum_{j \in \mathcal{N}(i)} (\mathbf{b}_1)_j^2}{1 - (\mathbf{b}_1)_i^2}$$

- In the above: $L^{(i)}$ is the signed Laplacian after the removal of the node v_i, $\mathbf{b}_1$ is the first eigenvector of $L(G)$, $\mathcal{N}(i)$ are the neighbours of the node v_i, and $d(i)$ is the degree of the node v_i.
- The derivation is straightforwards but a bit long, the details can be found in the paper

Finding large balanced subgraphs

- The author's algorithm uses this bound to greedily remove nodes until a balanced subgraph is found
- After a balanced subgraph is found, we check if the removed nodes can be re-introduced

Finding large balanced subgraphs

Algorithm 1 TIMBAL Algorithm

Input: signed graph G
$R \leftarrow \varnothing$
while G is not balanced **do**
  Compute $L(G)$, $\lambda_1(L(G))$, and corresponding $\mathbf{b}_1$
  $k \leftarrow \arg\min_i \frac{\lambda_1(L)+(\mathbf{b}_1)_i^2(d(i)-2\lambda_1(L(G)))-\sum_{j\in\mathcal{N}(i)}(\mathbf{b}_1)_j^2}{1-(\mathbf{b}_1)_i^2}$
  $G \leftarrow$ largest connected component in $G \setminus \{v_k\}$
  $R \leftarrow R \cup \{v_k\}$
end while
for $v \in R$ **do**
  if $G \cup \{v\}$ is balanced **then**
    $G \leftarrow G \cup \{v\}$
  end if
end for
return G

Finding large balanced subgraphs

Table 2: Largest balanced subgraph found by each method for each dataset

	HighlandTribes		Cloister		Congress		Bitcoin		TwitterReferendum	
method	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V\rvert$	$\lvert E\rvert$
Timbal	13	35	10	33	208	452	4 208	10 158	8 944	166 243
Grasp	10	18	6	11	115	145	2 167	3 686	5 425	49 105
Ggmz	10	21	5	7	153	238	1 388	1 683	2 501	2 821
Eigen	12	37	8	27	11	16	7	17	132	6 140
	WikiElections		Slashdot		WikiConflict		WikiPolitics		Epinions	
Timbal	3 786	18 550	42 205	96 460	48 136	356 204	63 252	218 360	62 010	169 894
Grasp	1 752	4 416	23 289	40 511	18 576	82 726	31 561	81 557	28 189	63 250
Ggmz	713	771	16 389	17 867	6 137	9 145	23 342	37 098	21 009	25 013
Eigen	11	41	35	491	11	28	10	45	6	14

[Table from Ordozgoiti]

- Under various optimizations, the algorithm is able to process the Epinions dataset (containing 1 millions nodes and 12 million edges) in 1.5 hours