Structural Balance in Signed Social Networks

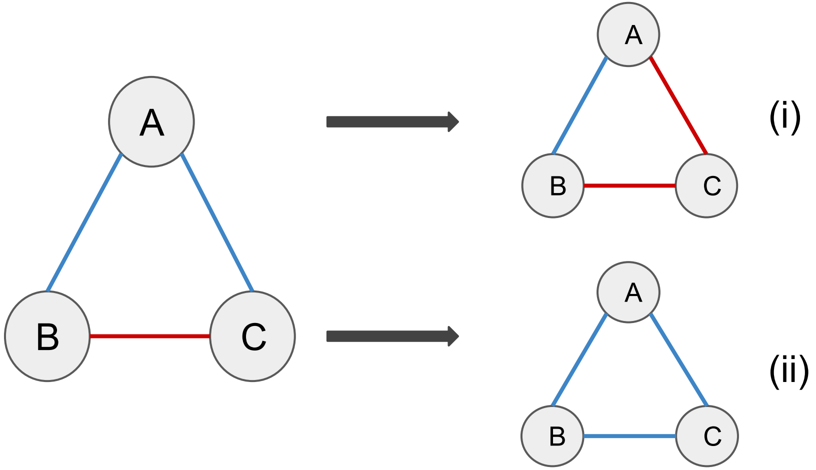
Continuous-time model of structural balance - Mavel et.al

Dyadic Imbalance in networks - Burghardt et.al

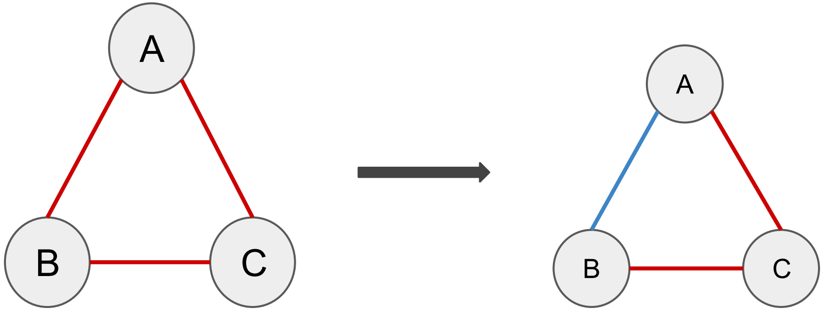
Seoyoung Cho

Mentor: Megan Morrison

Structural Balance



Red line: negative relationship
Blue-line: positive relationship



=> Stability or Balance

The triangles in the social network graph will reach balanced state, having odd number of positive edges.

Terms

Balanced: triangles with odd number of positive edges

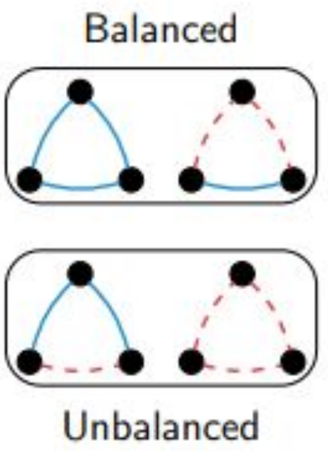
Unbalanced: triangles with even number of positive edges

Complete: Graphs contain edges between all pairs of nodes.

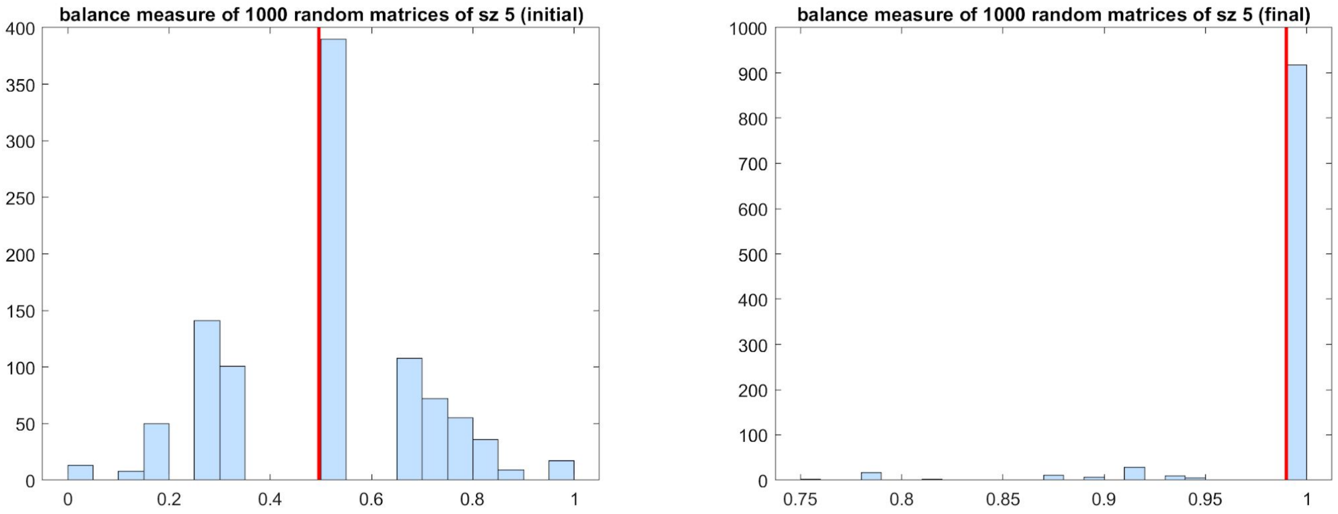
- In Marvel’s paper, all graphs are complete

Balance measure: $\dfrac{\text{Number of balanced triads}}{\text{Number of all triads}}$

* Balance measure = 1 : graph is balanced
= 0 : graph is unbalanced; imbalance measure = 1



Testing Structural Balance outcomes



Continuous-time Model

Motivation: Build dynamic theory of structural balance.

Continuous-time model for structural balance by Kulakowski et al.

$$X_{n,n} = X_{n,n}^T = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,n} \end{pmatrix}$$

* Each entry represents the strength of relationship between corresponding nodes
(+ : friendly - : unfriendly)

Evolution of the Relationship over time

$$\frac{dX}{dt} = X^2 \qquad (1)$$

$$\frac{dx_{ij}}{dt} = \sum_k \underbrace{x_{ik}x_{kj}}_{\text{Guides the sign of triangle \{i,j,k\}}} \qquad (2)$$

Consider: x_{ij} drawn independently from common distribution F that is symmetric about **μ** (mean value of the initial friendliness)

Result: (1) all-positive network if $\mu > 0$
(2) 2 all-positive factions with negative relations connecting if $\mu \le 0$

Evolution to a Balanced State

Solving eq 1:

1. Eigen Decompose X(0) $X(0) = QD(0)Q^T$ $D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$ $Q = \begin{pmatrix} | & | & & | \\ \omega_1 & \omega_2 & \cdots & \omega n \\ | & | & & | \end{pmatrix}$

2.

$$\frac{dx}{dt} = x^2 \Rightarrow \ell_k(t) = \frac{\lambda_k}{1 - \lambda_k t} \quad \Rightarrow \quad D(t) = \begin{pmatrix} \ell_1 & 0 & \cdots & 0 \\ 0 & \ell_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \ell_n \end{pmatrix} \quad \Rightarrow \quad \boxed{\begin{gathered} \frac{dX}{dt} = X^2 \\ \Rightarrow QD(t)Q^T \end{gathered}}$$

* $X(0) = \mathrm{diag}(\lambda_k)$

Behavior of the Model

Behavior 2: If μ ≤ 0, network divides into 2 all-positive factions connected by negative relations

Conditions: (1) $\lambda_1 \geq 0$ (2) $\lambda_1 > \lambda_2$ (3) $\omega_{1k} \neq 0$

$$\ell_k(t) = \frac{\lambda_k}{1-\lambda_k t} \to \infty \text{ as } t \to \frac{1}{\lambda_k}$$

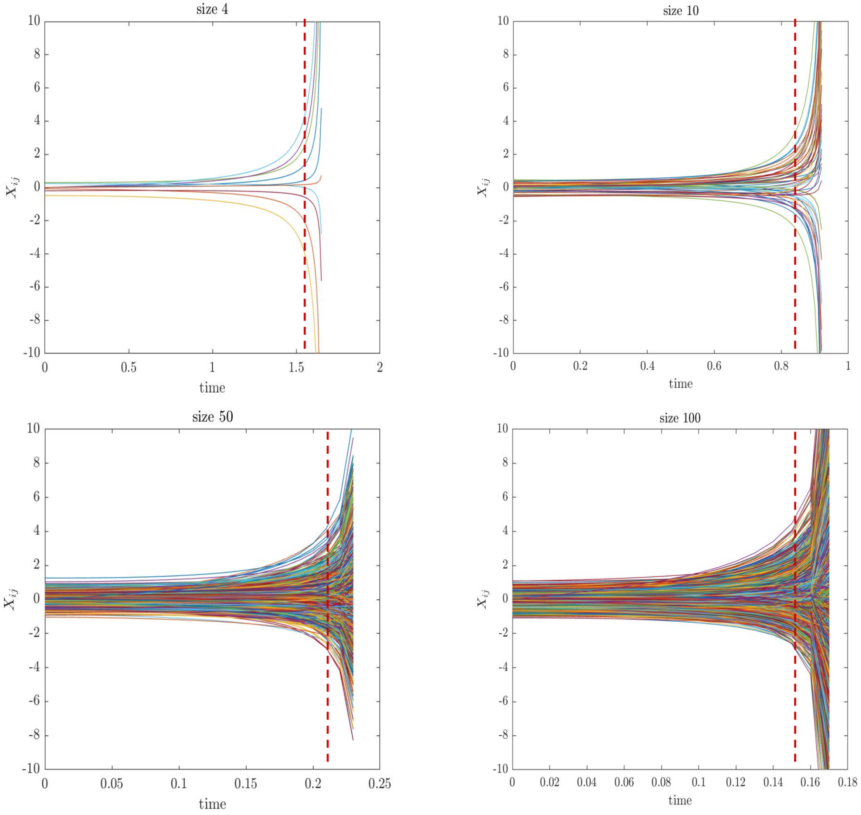
then all $x_{ij} \to \infty$ when ℓ_1 does

$$\Rightarrow \quad t^* = \frac{1}{\lambda_1}, \quad X(t) \to X(t^*)$$

$$X(t^*) \to Q diag[1, 0, \cdots, 0] Q^T = w_1 w_1^T$$

* $\omega_{1k} \neq 0$, so either > 0 or < 0
The signs of ω_{1k} determine the final cliques of each node.

What happens when the matrix size gets bigger? Does it take longer to hit the blow-up time?

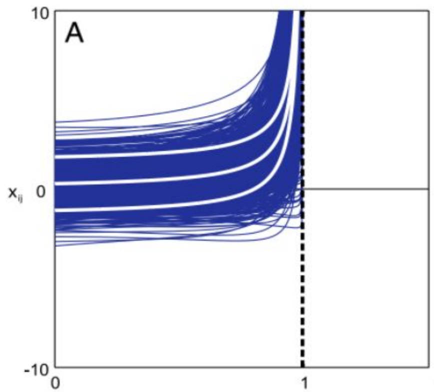
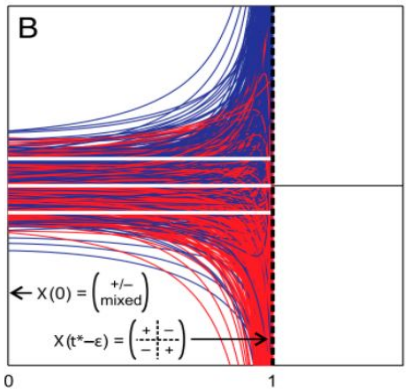
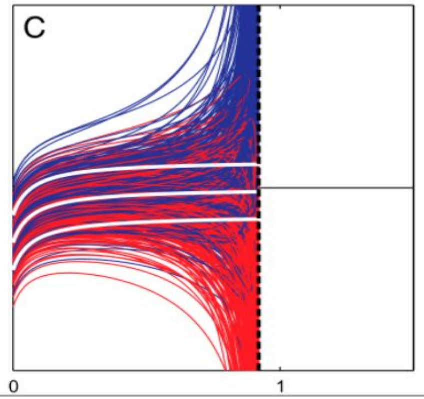


Blow-up time gets shorter

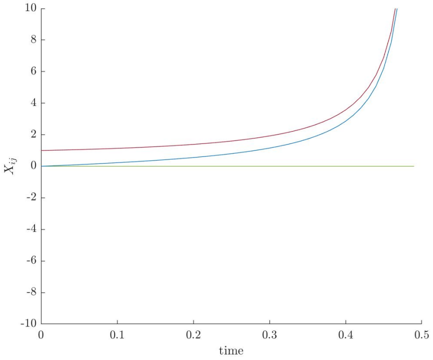
Behavior of the Model in Large n limit

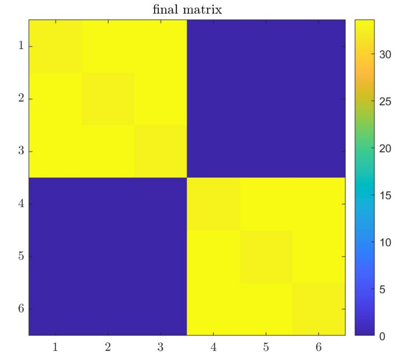
Characterizing the behavior of X(t) in large n limit - Asymptotic behavior gives a good approximation of the outcome of med-large group network

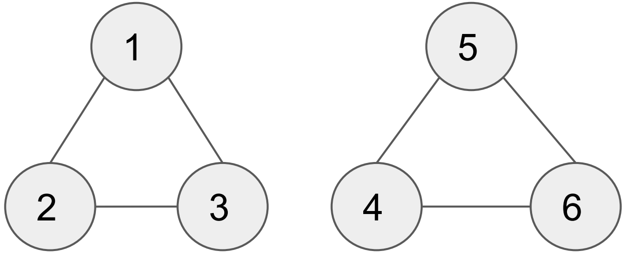
Behaviors determined by μ (mean value of the initial friendliness) :

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>μ > 0 :
One all-positive group</td>
<td>μ = 0 :
Two factions (positive / negative) of equal size</td>
<td>μ < 0 :
Two factions (positive / negative) of approximately equal size</td>
</tr>
</table>

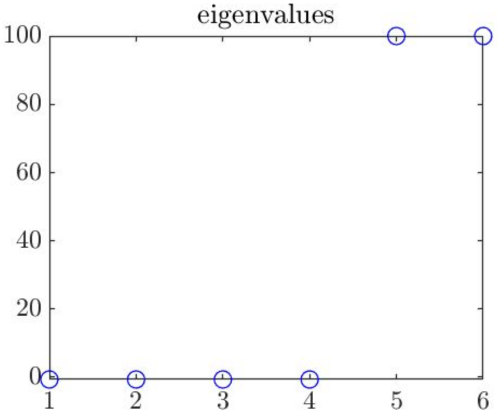
What happens when the matrix isn't fully connected?

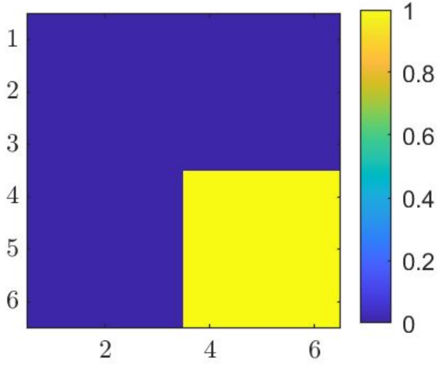






$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

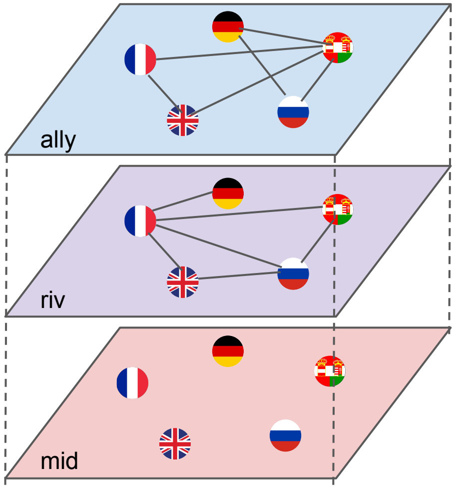


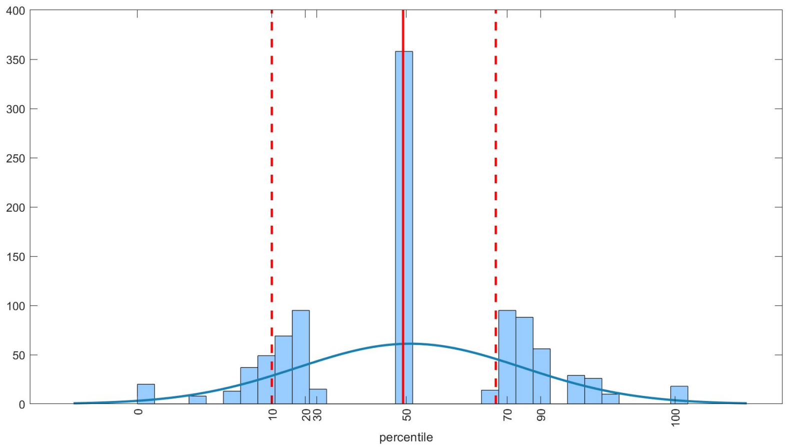


European Power network (1873-1922)

5 Countries: United Kingdom, France, Germany, Austria-Hungary, Russia

Relationship:
1. Alliance - signed treaty; positive measure ranging from 0 to 4
2. Rivalry - negative measure ranging from -2 to 0
3. Militarized Dispute - measure ranging from -5 to 5; < 0 hostile relationship; > 0 cooperating to fight other





Code

```
%%

sz = 6;  % size of the connectivity matrix 4 nodes, so matrix is 4x4

X0 = 0.3*randn(sz);
 % initial condition connectivity matrix (symmetric)
X0 = triu(X0,1) + triu(X0,1)';
X0_vec = reshape(X0,[],1);
tspan = 0:0.01:2;  %timespan for ode integration
[t,X_vec] = ode45(@(t,X) ode_struc_bal(t,X,sz), tspan, X0_vec);

X_final = reshape(X_vec(end,:),sz,sz); %look at the final connectivity matrix

[V,D] = eig(X_final);   %eigenvalues and eigenvectors of final connectivity matrix
[m, i] = max(diag(D));
%% make figures

%timeseries of matrix weights
fig = figure('position', [0, 0, 300, 200]); hold on;
plot(t,X_vec);
ylim([-10,10]);
xlabel('time');
ylabel('$X_{ij}$')

% final connectivity matrix values
figure();
imagesc(X_final);
pbaspect([1 1 1]);
colorbar();
title('final matrix');

% eigenvalues of final connectivity matrix and sign of outer product of
% leading eigenvector
fig = figure('position', [0, 0, 600, 200]); hold on;
subplot(1,2,1);
plot(diag(D),'bo');
title('eigenvalues');


subplot(1,2,2);
plot(diag(D),'bo');
title('$u_1$ outer product');
imagesc(sign(V(:,i)*V(:,i)'));
pbaspect([1 1 1]);
colorbar();
```

```
%% disconnected graph
sz = 6;  % size of the connectivity matrix 4 nodes, so matrix is 4x4

% X0 = 0.3*randn(sz);
%  % initial condition connectivity matrix (symmetric)

X0 = [0,-1,-1,0,0,0;
    -1,0,-1,0,0,0;
    -1,-1,0,1,0,0;
    0,0,1,0,1,1;
    0,0,0,1,0,1;
    0,0,0,1,1,0];
```

```
%% disconnected with 1 positive tie between
X0 = [0,1,1,0,0,0;
    1,0,1,0,0,0;
    1,1,0,-1,0,0;
    0,0,-1,0,1,1;
    0,0,0,1,0,1;
    0,0,0,1,1,0];
```

```
%% 3 disconneced clusters
sz = 9;
X0 = [0 1 1 0 0 0 0 0 0;
      1 0 1 0 0 0 0 0 0;
      1 1 0 0 0 0 0 0 0;
      0 0 0 0 1 1 0 0 0;
      0 0 0 1 0 1 0 0 0;
      0 0 0 1 1 0 0 0 0;
      0 0 0 0 0 0 0 1 1;
      0 0 0 0 0 0 1 0 1;
      0 0 0 0 0 0 1 1 0];


%% functions

function dXdt = ode_struc_bal(t,X,sz)

    X = reshape(X,sz,sz);  %must reshape
    dXdt = X^2;

    dXdt = reshape(dXdt,[],1);
end
```