Too big to fail
Robustness of different networks

Financial networks and systemic risk

Overview of Targeting strategies
Percolation
Percolation
Percolation
Percolation
2d lattice
2d lattice
2d lattice
2d lattice

a. $p = 0.1$

b. $p = 0.7$
2d lattice

Average cluster size
\[ \langle s \rangle \sim |p - p_c|^{-\gamma_p} \]

Order parameter
\[ p_\infty \sim (p - p_c)^{\beta_p} \]

Correlation length
\[ \xi \sim |p - p_c|^{-\nu} \]

Critical probability
\[ p_c \]

Critical exponents
\[ \gamma_p, \beta_p, \nu \]
2d lattice

Average cluster size
\[ \langle s \rangle \sim |p - p_c|^{-\gamma_p} \]

Order parameter
\[ p_\infty \sim (p - p_c)^{\beta_p} \]

Correlation length
\[ \xi \sim |p - p_c|^{-\nu} \]

Critical probability
\[ p_c \]

Critical exponents
\[ \gamma_p, \beta_p, \nu \]

Depends on lattice geometry

Depend on lattice dimension (eg 2d, 3d) up to 6d
2d lattice

Average cluster size
\[ \langle s \rangle \sim |p - p_c|^{-\gamma_p} \]

Correlation length
\[ \xi \sim |p - p_c|^{-\nu} \]
2d lattice

Order parameter

\[ p_\infty \sim (p - p_c)^{\beta_p} \]
OMG Who The Hell Cares!
Systemic risk
2d lattice

Order parameter

\[ p_\infty \sim (p - p_c)^{\beta_p} \]
Attacks

\[ P_\infty \]

\[ f = 0.1 \]
\[ f = f_c \]
\[ f = 0.8 \]

\[ f < f_c : \]
There is a giant component.

\[ f = f_c : \]
The giant component vanishes.

\[ f > f_c : \]
The lattice breaks into many tiny components.
Network structure comparison

What network do you think is more robust?
Network structure comparison

Scale-free networks are more robust

Most nodes have low degrees

Hubs are highly connected and central
Targeted removal

Robustness of different networks

Targeting strategies

Financial networks and systemic risk
Targeted removal

If we consider targeted attacks everything changes!
Hubs are highly connected and central
Network structure comparison
Network structure comparison
Network structure

\[ f_c \]

\[ \gamma \]

- \( k_{\text{max}} = 10^2 \)
- \( k_{\text{max}} = 10^3 \)
- \( k_{\text{max}} = 10^5 \)
- \( k_{\text{max}} = 10^8 \)
- \( k_{\text{max}} = \infty \)
Example: Systemic risk

risk that default or stress of one or more financial institutions ("banks") will trigger default or stress of further banks.
Systemic risk

Interbank network

- banks
- interbank loans

Balance Sheet

Assets
- Interbank Loans
- Derivatives
- Mortgages
- Bonds
- Others

Liabilities
- Interbank Liabilities
- Customer Deposits
- Other
- Equity
Systemic risk
Systemic risk

A 0.5% shock increases 3-6 times the probability of default
Systemic risk

A 0.5% shock increases 3-6 times the probability of default

Think of a topic you like
Think of a topic you like

Think of an example of maximising/minimising propagation
Influence maximisation

Selection of $k$ nodes that best trigger a cascade
Heuristic strategies

Rule of thumb strategies that make sense
Kempe et al

First “influence maximisation” algorithm

Greedy algorithm - Theoretical guarantee

Works well with unrealistic assumptions
Kempe et al

Algorithm 1 Greedy Approximation Algorithm

1: Start with $A = \emptyset$.
2: while $|A| \leq k$ do
3:  For each node $x$, use repeated sampling to approximate $\sigma(A \cup \{x\})$ to within $(1 \pm \varepsilon)$ with probability $1 - \delta$.
4:  Add the node with largest estimate for $\sigma(A \cup \{x\})$ to $A$.
5: end while
6: Output the set $A$ of nodes.
Algorithm 1: Greedy Approximation Algorithm

1: Start with $A = \emptyset$.
2: while $|A| \leq k$ do
3:   For each node $x$, use repeated sampling to approximate $\sigma(A \cup \{x\})$ to within $(1 \pm \varepsilon)$ with probability $1 - \delta$.
4:   Add the node with largest estimate for $\sigma(A \cup \{x\})$ to $A$.
5: end while
6: Output the set $A$ of nodes.
Algorithm 1: Greedy Approximation Algorithm

1: Start with $A = \emptyset$
2: \textbf{while} $|A| \leq k$ \textbf{do}
3: \hspace{1em} For each node $x$, use repeated sampling to approximate $\sigma(A \cup \{x\})$ to within $(1 \pm \varepsilon)$ with probability $1 - \delta$.
4: \hspace{1em} Add the node with largest estimate for $\sigma(A \cup \{x\})$ to $A$.
5: \textbf{end while}
6: Output the set $A$ of nodes.
Set of nodes

Maximum n. of nodes in seed

Algorithm 1: Greedy Approximation Algorithm

1. Start with $A = \emptyset$.
2. while $|A| \leq k$ do
3. For each node $x$, use repeated sampling to approximate $\sigma(A \cup \{x\})$ to within $(1 \pm \varepsilon)$ with probability $1 - \delta$.
4. Add the node with largest estimate for $\sigma(A \cup \{x\})$ to $A$.
5. end while
6. Output the set $A$ of nodes.

Influence of set of nodes $a+x$
Competitive im

Two or more parties compete for influence

Classical setting: 2 parties, opposite sides

Easy to study on voter model
Competitive im Zealot Zealot
Competitive im on voter model

\[ \Delta_i \frac{dx_i}{dt} = (1 - x_i)(\sum_j a_{ji}x_j + p_{A,i}) - x_i(\sum_j a_{ji}(1 - x_j) + p_{B,i}) \]
Competitive IM on voter model

Probability being in state A

$$\Delta_i \frac{dx_i}{dt} = (1 - x_i)(\sum_j a_{ji}x_j + p_{A,i}) - x_i(\sum_j a_{ji}(1 - x_j) + p_{B,i})$$

Normalisation factor

$$\Delta_i = \sum_j a_{ji} + p_{A,i} + p_{B,i}$$

Influence of zealot A

Influence of neighbours
Competitive IM on voter model

(a) 

\[ \langle K_{\text{controlled}} \rangle \]

vs.

\[ \text{time horizon } T \]

\[ \times 10^n \]
Temporary influence

$q = \text{probability of flipping back to pre-influence state}$
Temporary influence

Star network

Scale-free network
Summary

Percolation and its implications
Systemic risk and instability of finance
Influence maximisation