Simulation II: Dynamics on Networks
Dynamics on networks

- Dynamic state changes taking place on a static network topology
  - Regulatory dynamics on gene/protein networks
  - Population dynamics on ecological networks
  - Disease infection on social networks
  - Information/culture propagation on organizational/social networks
Simple example:
Random walk on a network

- An agent (or a set of agents) moving on a network
- An agent jumps randomly to one of the neighbor nodes at each time step
Exercise

• Simulate random walk of an agent on a directed random network made of 50 nodes

• Count how many times each node was visited by the agent over time
TPM and asymptotic probability distribution (review)

- $|\lambda| \leq 1$ for all eigenvalues

- If the original network is strongly connected (with some additional conditions), the TPM has one and only one eigenvalue 1 (no degeneration)

→ This is a unique dominant eigenvalue; the probability vector will converge to its corresponding eigenvector
Exercise

- Construct the transition probability matrix of the random network used in the previous exercise
- Find its dominant eigenvector with $\lambda = 1$
- Compare the results with the previous "counting" results
Dynamics on Networks with Discrete Node States
Opinion formation (Voter model)

- A simple model of opinion formation in society
  - Opinions = discrete states
Three versions of voter models

• **Original voter model**
  - A randomly selected node copies the opinion of one of its neighbors

• **Reverse voter model**
  - A randomly selected node “pushes” its opinion into one of its neighbors

• **Link-based voter model**
  - An opinion is copied through a randomly selected link
Exercise

• Simulate the three different versions of the voter model (original, reverse and link-based) on a Barabasi-Albert scale-free network

• Compare the speed of opinion homogenization between the three models
  - Why different?
Epidemics (SIS/SIR model)

- Initially, a small fraction of nodes are infected by a disease
- If a susceptible node has an infected neighbor, it will be infected with probability $p_i$ (per infected neighbor)
- An infected node will recover and become susceptible (SIS) or recovered (SIR) with probability $p_r$
Exercise

• Study the effects of infection/recovery probabilities on the fixation of a disease on a random social network
  - In what condition will the disease remain within society?
  - In what condition will it go away?
  - Is the transition smooth, or sharp?
Exercise

• Do the same experiments with WS small-world networks and BA scale-free networks

• Compare their properties
Cascade of failure

- Load on a failing node is divided and distributed to its neighbors.
- If the load exceeds capacity of each node, it causes another node failure.
Exercise

• Simulate a cascade of failure on a scale-free network made of 100 nodes with random node capacities and load assignments

• Investigate which node has the most significant impact when it fails
Hopfield network

- A.k.a. “attractor network”
- Neurons connected in a shape of an undirected weighted complete graph
- Each neuron takes either 1 or -1, and updates its state in discrete time
State-transition rule

\[ s_i(t+1) = \text{sign} \left( \sum_j w_{ij} s_j(t) \right) \]

- \( w_{ij} \): connection weight between neuron \( i \) and neuron \( j \)
- \( w_{ij} = w_{ji} \) (symmetric interaction)
- \( w_{ii} = 0 \) (no feedback to itself)
Setting weights by “imprinting”

\[ w_{ij} = \sum_k s^k_i s^k_j \]

- \( k \): index of patterns memorized
- \( s^k_i \): state of neuron \( i \) in pattern \( k \)

- e.g.

```
Pattern 1

Pattern 2

0

2  -2
```
Recovering patterns

- When started with some initial pattern, the network “remembers” the closest pattern in its memory (or its reversal)
  - Can be applied to *content addressable memory*, *pattern recognition*, etc.
Exercise

• Simulate the behavior of the following Hopfield network

2, -2 : weights

-2, -2, 2

2 : +1

-1
Gene regulatory network

- Each gene is activated or inhibited by other genes
  - Forming a network of “logic gates”
  - Each gene takes binary state (on/off)

(from Hasty et al., Nature Reviews Genetics 2, 268-279, 2001)
Boolean network

- Mathematical abstraction of gene regulatory networks
  - Binary node states
  - Each node determines next state using its own Boolean state transition function (referring to neighbors’ states)

- Random Boolean network:
  - Network topology and state transition functions are both randomly generated
Example of transition functions

- 2-input functions \(2^2 = 16\) possibilities

<table>
<thead>
<tr>
<th>(X)</th>
<th>(Y)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
<th>(Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>
Kauffman’s NK networks

- **N**: # of nodes
- **K**: # of inputs to each node
  - Topologies and state-transition functions are both random
  - Similar to, but not the same as, the NK fitness landscape (NK model) often used in mathematical biology and management sciences
NK network’s attractors

- Total # of macro-states: $2^N$
- The network eventually falls into one of its “attractors”
Exercise

• Create a Python code that generates the NK network’s state-transition diagram (i.e., a directed network whose nodes are the network’s macro-states)

• Count how many attractors exist

• Study how # of attractors change when you vary N and K
Dynamics on Networks with Continuous Node States
Simple diffusion

• Individually:

\[
\frac{ds_i}{dt} = D \sum_{j \in N_i} (s_j - s_i)
\]

• Collectively (with Laplacian L):

\[
\frac{ds}{dt} = -DLs
\]
Exercise

- Simulate a diffusion process of continuous node states on a Barabasi-Albert scale-free networks with $n = 100$ and $m = 1$
Exercise

- Calculate the eigenvalues and eigenvectors of Laplacian matrices of several different network topologies
- Interpret their meanings in the context of diffusion
- Confirm your interpretation by numerical simulation of the diffusion processes
Synchronization

• Linear coupling model:

\[
\frac{ds_i}{dt} = F(s_i) + \sum_j (c_{ij} H(s_j))
\]

- If \( s_i(t) = s(t) \) for all \( i \), then the network is synchronized

• \( F(s) \): internal dynamics
• \( C = (c_{ij}) \): coupling matrix
• \( H(s) \): output function
Synchronization and Laplacian

- If coupling depends only on the difference of outputs across a link:

\[
\frac{ds_i}{dt} = F(s_i) + \sigma \sum_{j \in N_i} (H(s_j) - H(s_i))
\]

- I.e., \( C = -\sigma L \)

- Laplacian’s “spectral gap” (first non-zero eigenvalue) is critical in determining synchronizability of the network
Exercise

• Simulate the following nonlinear Kuramoto model:

\[
\frac{ds_i}{dt} = w_i + \frac{K}{|N_i|} \sum_{j \in N_i} \sin(s_j - s_i)
\]

• \(w_i\): inherent angular velocity
• \(N_i\): neighbors of node \(i\)

- What kind of networks synchronize most easily?
Exercise

- Measure and plot the following “phase coherence” in the simulation of the Kuramoto model:

\[ r = \left| \frac{\sum_j e^{i\theta_j}}{n} \right| \]
Synchronizability
Synchronizability

• Synchronizability of a simple coupled dynamical network can be studied by conducting stability analysis

\[
\frac{dx_i}{dt} = R(x_i) + \alpha \sum_{j \in N_i} (H(x_j) - H(x_i))
\]

\(R(x)\): Local reaction term (homogeneous)  
\(H(x)\): Output function
Exercise

• Consider adding a small perturbation to the general solution of the dynamical equation (w/o interactions)

\[
\frac{dx}{dt} = R(x) \rightarrow x_s(t)
\]

• Conduct stability analysis by assuming:

\[
x_i(t) = x_s(t) + \Delta x_i(t)
\]
Condition for synchronizability

- Solution $x_s(t)$ is stable (i.e., the network is synchronized) if

$$\alpha \lambda_i H'(x_s(t)) > R'(x_s(t))$$

for all $i$ and $t$

(you need to consider only $\lambda_2$ and $\lambda_n$)
Exercise

• Analyze the synchronizability condition of the following coupled oscillator model:

\[
\frac{d\theta_i}{dt} = \beta \theta_i + \alpha \sum_{j \in N_i} (\theta_j - \theta_i)
\]
Mean-Field Approximation
Mean-field approximation

- An approximation to drastically reduce the dimensions of the system by reformulating the dynamics in terms of “a state of one node” and “the average of all the rest (= mean field)”
How MFA works

1. Make an approximated description about how one node changes its state through the interaction with the average of all the rest (= mean field).
2. Assume that 1. uniformly applies to all the nodes, and analyze how the mean field itself behaves.
Mathematical description of MFA (difference equations)

• Original equations:
  \[ x_{i_t} = F_i( \{ x_{i_{t-1}} \} ) \]

• Approximate equations with MFA:
  \[ x_{i_t} = F'_i(x_{i_{t-1}}, \langle x \rangle_{t-1}) \]
  \[ \langle x \rangle_t = \sum_i x_{i_{t-1}} / n \]

Each state-transition function takes only two arguments: its own state and the "mean field"
Example: SIS on a random network

• Infection probability $p_i$
• Recovery probability $p_r$
• Edge probability $p_e$

• Write down a difference equation that describes how the probability of infected nodes, $q_t$, (mean field), changes over time
Example: SIS on a random network

<table>
<thead>
<tr>
<th>Current state</th>
<th>Next state</th>
<th>Probability of this transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (susceptible)</td>
<td>0 (susceptible)</td>
<td>((1 - q)(1 - p_e q p_i)^{n-1})</td>
</tr>
<tr>
<td>0 (susceptible)</td>
<td>1 (infected)</td>
<td>((1 - q)(1 - (1 - p_e q p_i)^{n-1}))</td>
</tr>
<tr>
<td>1 (infected)</td>
<td>0 (susceptible)</td>
<td>(q p_r)</td>
</tr>
<tr>
<td>1 (infected)</td>
<td>1 (infected)</td>
<td>(q(1 - p_r))</td>
</tr>
</tbody>
</table>

- Find equilibrium states
- Study the stability of those equilibrium points
  - When does the equilibrium \(q = 0\) become unstable (i.e., epidemic occurs)?
Example: SIS on a SF network

- Infection probability $p_i$
- Recovery probability $p_r$
- Degree distribution $P(k)$

- Write down a difference equation that describes how the probability of infected nodes with degree $k$, $q_t(k)$ (many mean fields), changes over time
Degree-dependent infection

- Probability for a node with degree $k$ to get infected from its neighbor:

\[ \sum_{k'} P_n(k' \mid k) q(k') p_i \]

$P_n$ : neighbor degree probability distribution

If the network is nonassortative:

\[ P_n(k') = \frac{k'}{\langle k \rangle} P(k') \]
FYI: Friendship paradox

- “Your friends have more friends than you do, on average”

\[
\sum_{k'} k' P_n(k') = \sum_{k'} \frac{k'^2 P(k')}{\langle k \rangle} = \frac{\langle k^2 \rangle}{\langle k \rangle}
\]

\[
= \frac{\langle k \rangle^2 + \sigma(k)^2}{\langle k \rangle} = \langle k \rangle + \frac{\sigma(k)^2}{\langle k \rangle}
\]
### Calculation...

<table>
<thead>
<tr>
<th>Current state</th>
<th>Next state</th>
<th>Probability of this transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (susceptible)</td>
<td>0 (susceptible)</td>
<td>$(1 - q(k)) \left( 1 - \sum_{k'} \frac{k'}{\langle k \rangle} P(k') q(k') p_i \right)^k$</td>
</tr>
<tr>
<td>0 (susceptible)</td>
<td>1 (infected)</td>
<td>$(1 - q(k)) \left( 1 - \left( 1 - \sum_{k'} \frac{k'}{\langle k \rangle} P(k') q(k') p_i \right)^k \right)$</td>
</tr>
<tr>
<td>1 (infected)</td>
<td>0 (susceptible)</td>
<td>$q(k) p_r$</td>
</tr>
<tr>
<td>1 (infected)</td>
<td>1 (infected)</td>
<td>$q(k)(1 - p_r)$</td>
</tr>
</tbody>
</table>

\[
q_{t+1}(k) = (1 - q(k)) \left( 1 - \left( 1 - \sum_{k'} \frac{k'}{\langle k \rangle} P(k') q(k') p_i \right)^k \right) + q(k)(1 - p_r) \\
= (1 - q(k)) \left( 1 - (1 - q_n p_i)^k \right) + q(k)(1 - p_r) ,
\]

\[
q_n = \frac{\sum_{k'} k' P(k') q(k')}{\langle k \rangle}
\]
Calculation...

\[ q_{t+1}(k) = (1 - q(k))(1 - (1 - kq_n p_i)) + q(k)(1 - p_r) \]
\[ = (1 - q(k))kq_n p_i + q(k) - q(k)p_r = f(q(k)) \]

\[ q_{eq}(k) = \frac{kq_n p_i}{kq_n p_i + p_r} \]

With this:

\[ q_n = \frac{1}{\langle k \rangle} \sum_{k'} k' P(k') \frac{k' q_n p_i}{k' q_n p_i + p_r} \]
• For BA SF networks, this becomes:

\[ q_n = \frac{1}{2m} \sum_{k'=m}^{\infty} k' \cdot 2m^2 k'^{-3} \frac{k'q_n p_i}{k'q_n p_i + p_r} \]

\[ 1 = mp_i \sum_{k'=m}^{\infty} \frac{1}{k'(k'q_n p_i + p_r)}. \]

\[ q_n \approx \frac{p_r}{pr} \frac{p_r}{(e^{mp_i} - 1)mp_i} \]
Calculation...

- Final stability analysis:

\[
\frac{d f(q(k))}{dq(k)} \bigg|_{q(k) = \frac{k q_n p_i}{k q_n p_i + p_r}} = -k q_n p_i + \frac{p_r}{k q_n p_i + p_r} \frac{k^2 P(k) p_i}{\langle k \rangle} + 1 - p_r = r(k)
\]

\[
r(k) = -k \frac{p_r}{(e^{mp_i} - 1) m p_i} p_i + \frac{p_r}{k \frac{p_r}{(e^{mp_i} - 1) m p_i}} p_i + p_r \frac{k^2 \cdot 2 m^2 k^{-3} p_i}{2m} + 1 - p_r
\]

\[
= -\frac{k p_r}{(e^{mp_i} - 1) m} + \frac{m p_i}{k \frac{p_r}{(e^{mp_i} - 1) m}} + k + 1 - p_r
\]
Conclusion

• If $p_i \to 0$:

$$\lim_{p_i \to 0} r(k) = -\frac{kpr}{\left(\frac{pr}{e^{mpi}} \to \infty\right) - 1} m + \frac{m [p_i \to 0]}{\left(\frac{pr}{e^{mpi} \to \infty}\right) - 1} m + k + 1 - pr$$

$$= 1 - pr$$

• Since $0 < 1 - pr < 1$, the non-zero equilibrium state (i.e., epidemic) is still stable even if $p_i \to 0$ on scale-free networks!!
Take-home lesson

• Dynamics on networks can be influenced significantly by network topology