CSCI5070 Advanced Topics in Social Computing

Irwin King

The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

Outline

- Scale-Free Networks
  - Generation
  - Properties
  - Analysis
- Dynamic Networks
  - Emergence
  - Epidemics
  - Synchrony
  - Influence Networks

SCALE-FREE NETWORKS

Scale-Free Networks

- A *scale-free network* is a network *G* with degree sequence distribution *g*' obeying a power law of the form $h(k) \sim k^{-q}$, where *k* is degree ($1<k<\infty$) and q is an exponent (typically $2<q<3$)
- Power-law distributions are not exponential distributions
  - The tail of an exponential distribution vanishes much faster

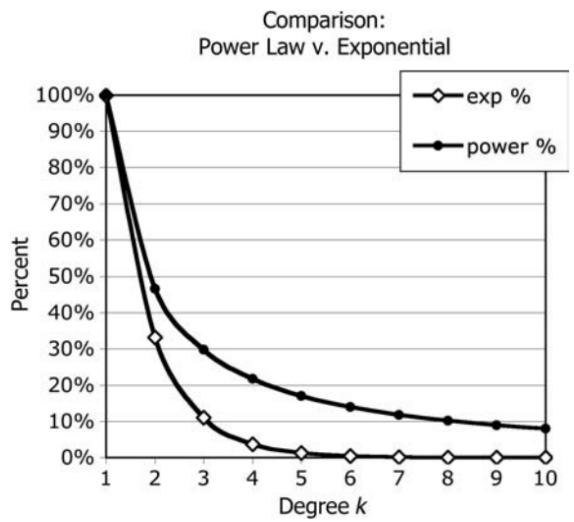


Figure Comparison of $y = 3\exp(-qk)$, versus $y = 1/K^q$, for $q = 1.1$, shows the fat tail of the power law.

Generation

- The Barabasi–Albert (BA) Network
  - Sample from the degree sequence distribution of network *G* at timestep *t*
  - The probability of a high-degreed node obtaining a subsequent link continues to rise, and vice versa
- Procedure

(1) *Growth*: Starting with a small number (m_0) of vertices, at every time step we add a new vertex with $m(\leq m_0)$ edges (that will be connected to the vertices already present in the system).

(2) *Preferential attachment*: When choosing the vertices to which the new vertex connects, we assume that the probability P that a new vertex will be connected to vertex i depends on the connectivity k_i of that vertex, such that $P(k_i) = \frac{k_i}{\sum_j k_j}$.

(3) After t time steps the model leads to a random network with $N = t + m_0$ vertices and mt edges."

BA Generative Procedure

1. Inputs: Δm = number of links to add to each new node; n = network size.
2. Initialize: Designate nodes by enumerating them as $0,1,2,\ldots,(n-1)$.
   a. Given the ultimate number of nodes $n > 3$, initially construct a complete network with nNodes $= n_0 = 3$ nodes and nLinks $= 3$ links. The degree sequence of this complete network is $g = [2,2,2]$, and the degree sequence distribution is $g' = [1]$ because each node is connected to the other two.
3. While nNodes $\leq n$:
   a. New node: Generate a new tail node v.
   b. #New links: Set n_links = minimum $(\Delta m, n)$. Cannot add more links than existing nodes.
   c. Repeat n_links times:
      i. *Preferential attachment*—select an existing head node u by sampling from the degree sequence cumulative distribution function CDF(i) defined by

$$\mathrm{CDF}(i) = \sum_{j}^{i} \frac{\mathrm{degree}(n_j)}{\mathrm{k\_total}}, \text{ where } \mathrm{k\_total} = 2\mathrm{nNodes}$$

      ii. Let r be a uniform random number from $[0,1)$. Then u is a random variate sampled from CDF(i) as follows:

$$\mathrm{CDF}(u-1) \leq r \leq \mathrm{CDF}(u); \;\; u = r\,\mathrm{nNodes}$$

      iii. Connect $v \sim u$, taking care to avoid duplicate links.

BA Network Entropy

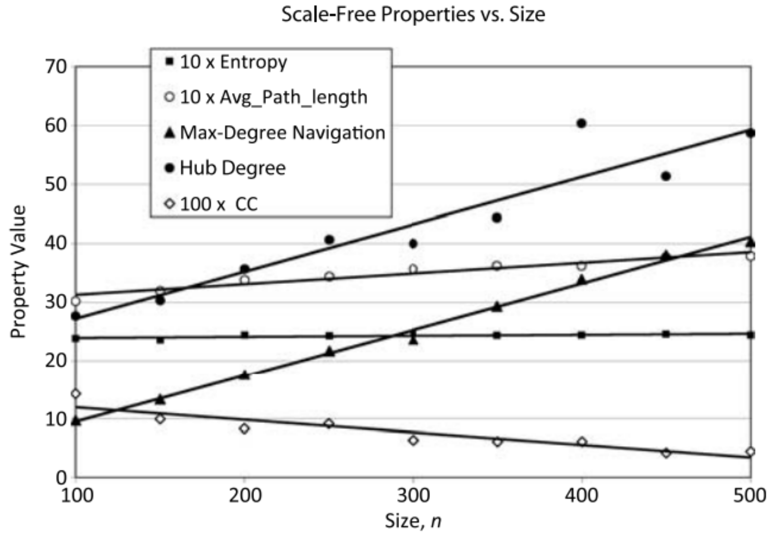


Figure Comparison of BA scale-free network properties versus size, n, for $\Delta m = 2$. Note the scale factors on entropy, average path length, and cluster coefficient.

- Entropy remains constant as network size increases

$$I = \int_{\Delta m}^{\infty} h(k) \log_2(h(k))\delta x = -1.443 \int_{\Delta m}^{\infty} h(k) \ln(h(k))\delta x$$

Substitution of $h(x) = Ax^{-3}$, where $A = 2\Delta m\ (\Delta m + 1)$ into the integral,

$$I = -1.443A \int_{\Delta m}^{\infty} \left(\frac{\ln(A) - 3\ln(x)}{x^3} \right) \delta x\ = \Delta m + 1\, \frac{1.1645 - \log_2\left(\frac{\Delta m + 1}{\Delta m^2}\right)}{\Delta m}$$

Hub Degree versus Density

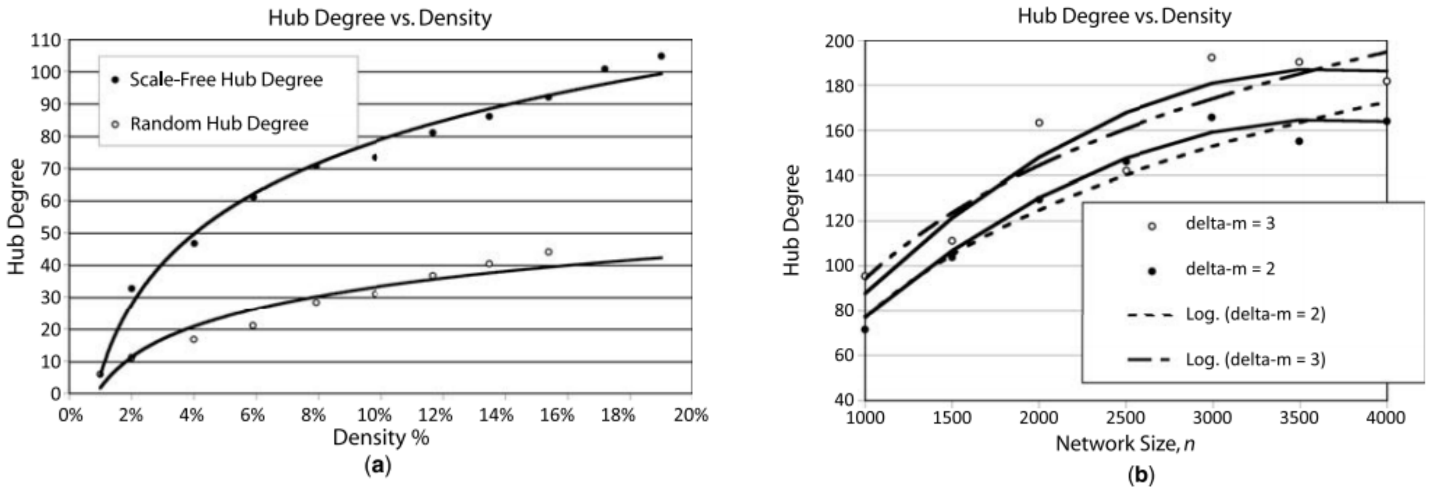


Figure Comparison of hub degree of networks for fixed and variable n: (a) scale-free and random networks versus density, for constant size, $n = 200$, (b) variable number of links m; and hub degree with variable density and size, n, but fixed Δm.

$$\text{Degree(hub)} \sim O\big(\log_2(\text{density})\big);\ 1\% \leq \text{density} \leq 20\%$$

$$\text{Density(scale-free)} = 2\frac{m}{n(n-1)} = 2\frac{\Delta m}{n}$$

- Extremely high hub degree is the predominant property of a scale-free network
- Hub degree grows logarithmically with density for both random and scale-free networks, but the rate of increase is much greater for a scale-free network

Average Path Length

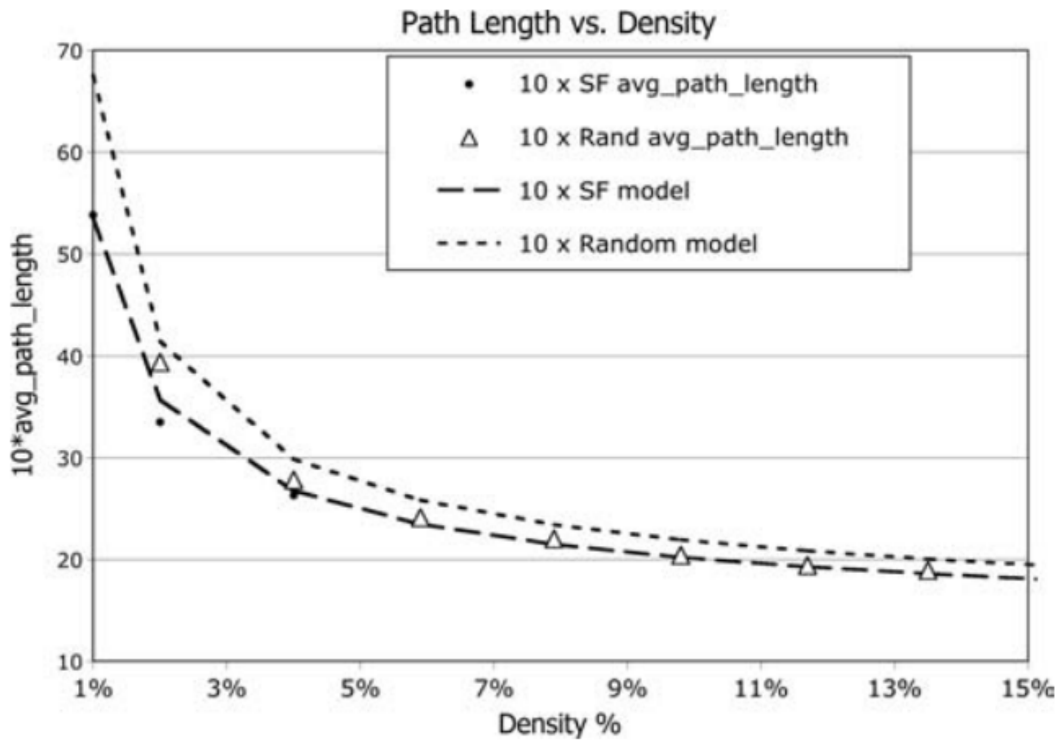


Figure Comparison of average path length of BA scale-free and ER random networks versus density, $n = 200$. The average path length of a scale-free network is slightly less then that of an equivalent random network.

$$\text{avg\_ path\_length(BA network)} = \frac{A \ \log(n)}{\log(n) + \log(C(\text{density}))}$$

Average Path Length

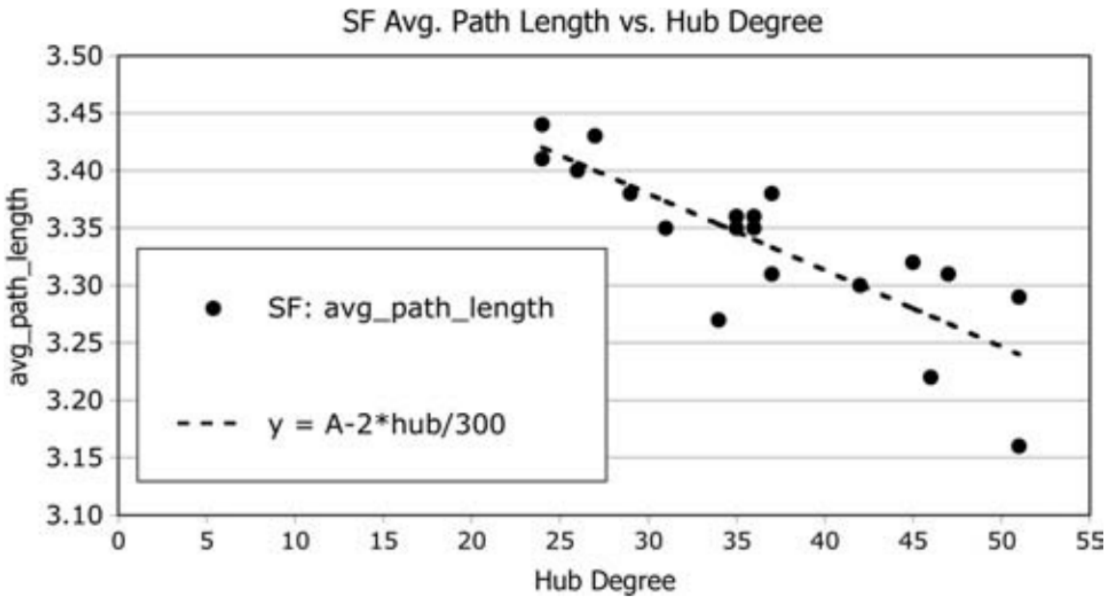


Figure Average path length declines linearly as hub degree increases in a scale-free network of size $n = 200$ and density 2% ($\Delta m = 2$). Least-squares curve fit to these data yields $A = 3.58$ and slope equal to $\frac{-2}{300}$.

$$\text{avg\_ path\_length(BA network)} = A - 2\,\frac{\text{hub\_degree}}{300}$$

Closeness

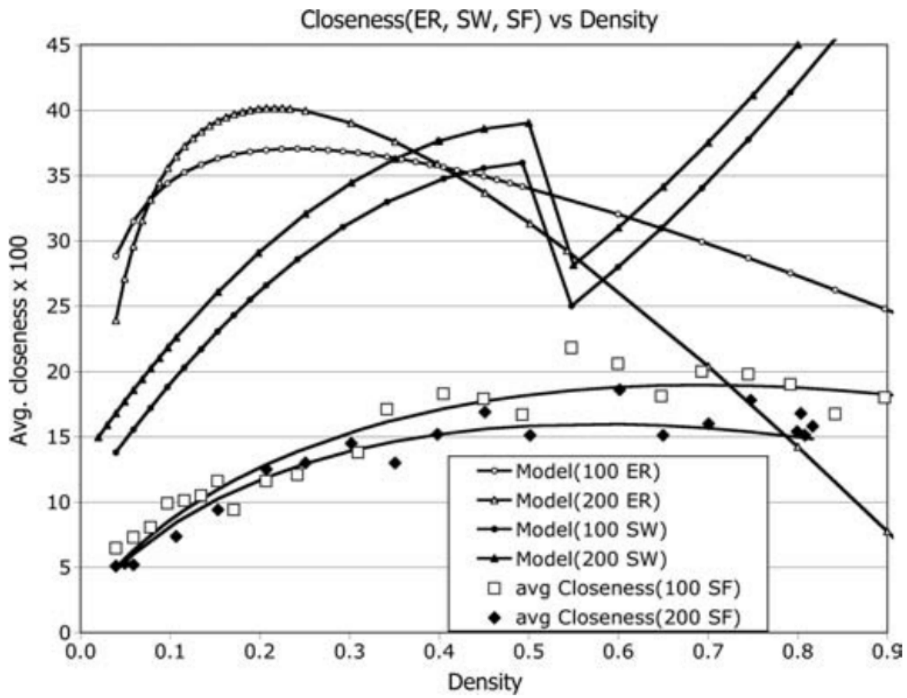


Figure Average closeness versus density for random, small-world, and scale-free networks shows that scale-free networks have weaker intermediaries on average.

TABLE Number of Paths per Node Through Closest Node

Network	Number of Paths *n*
Random	0–10
Small world	0–20
Scale-free	0–38

Cluster Coefficient

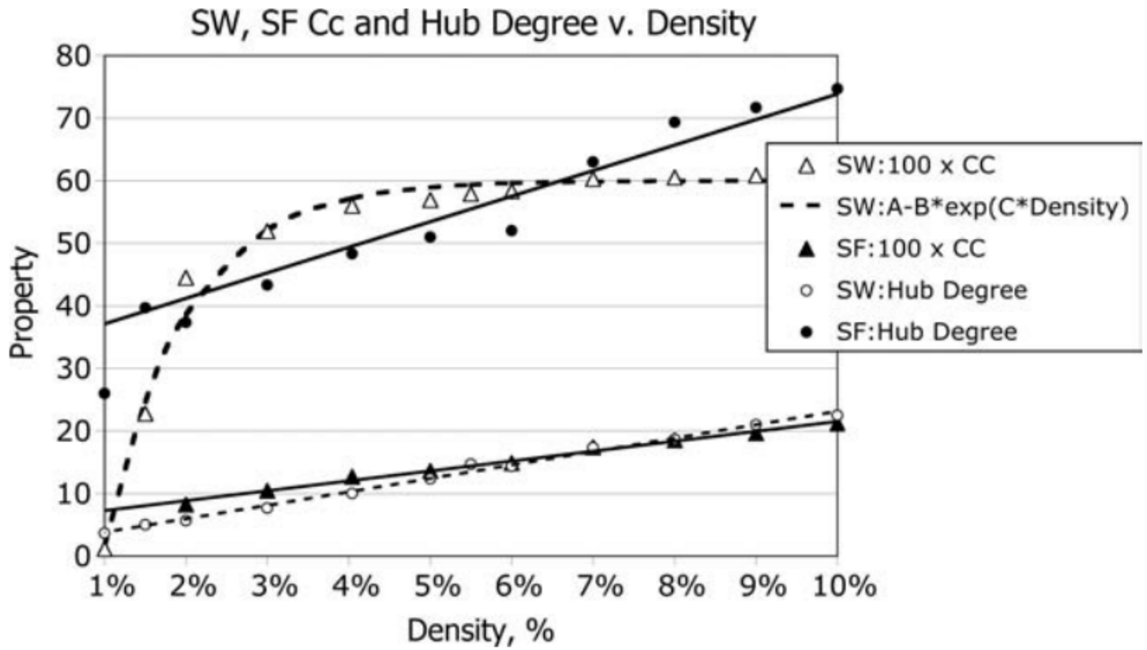


Figure Cluster coefficient versus density of small-world and scale-free networks for small values of density, small rewiring probabilities, and $n \sim 200$, rewiring probability $\sim$ 4% (small world).

Scale-Free CC. $100(CC(\text{scale-free})) \sim O(\text{density})$, $1\% \leq \text{density} \leq 10\%$. Alternatively, $100(CC(\text{scale-free})) \sim O(\Delta m)$ because density $= 2(\Delta m/n)$; $n \gg 1$.

Small-World CC. $100(CC(\text{small-world})) = A - B \exp(C \text{ density})$; $1\% \leq \text{density} \leq 10\%$, where $A = 60$; $B = 158.5$; $C = -100$, for $n = 200$.

Comparison

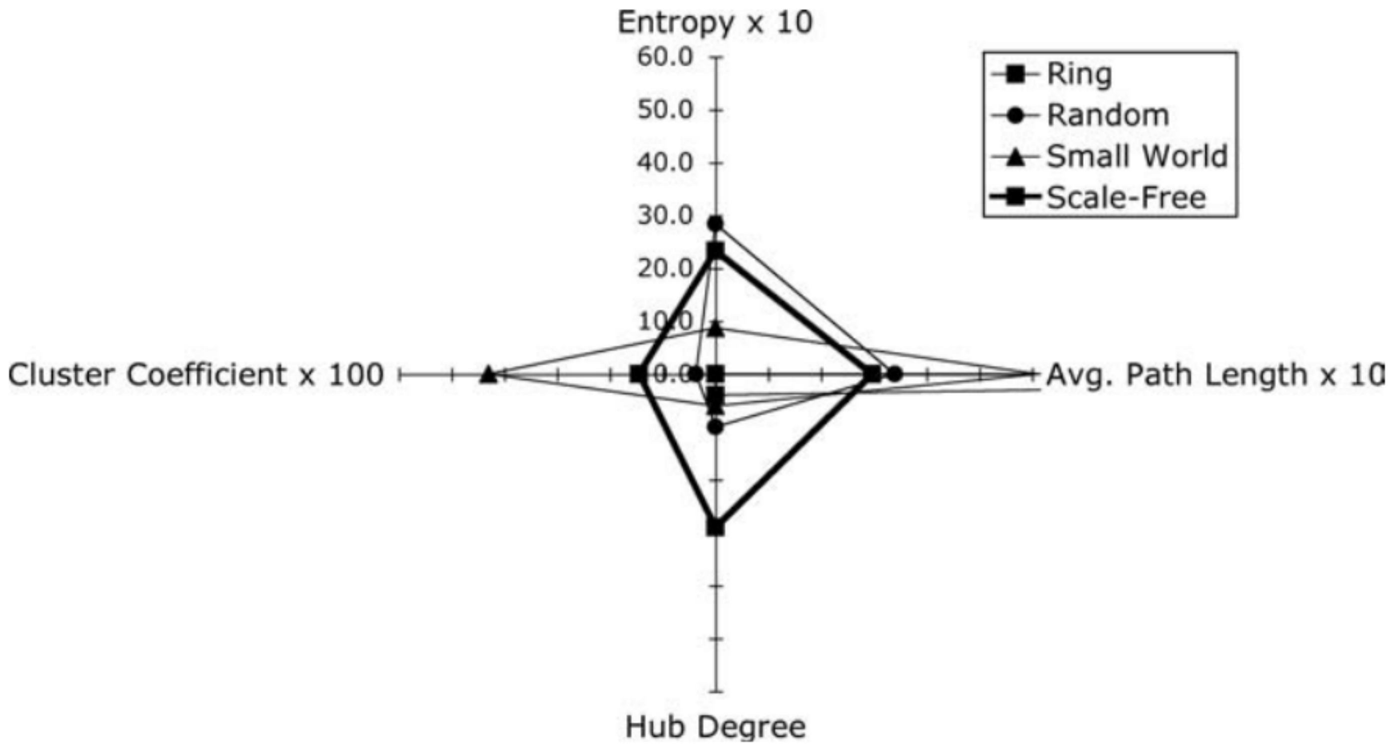


Figure Kiviat graph comparing ring, random, WS small-world, and BA scale-free network properties.

DYNAMIC NETWORKS

Definitions

- A *dynamic network*, $G(t) = [N(t), L(t), f(t):R]$, is a time-varying 3-tuple consisting of a set of nodes $N(t)$, a set of links $L(t)$, and a mapping function $f(t): L(t) \rightarrow N(t) \times N(t)$, which maps links into pairs of nodes
- If G reaches a final state, $G(t_f)$, and remains there for all $t \geq t_f$, G is said to be *convergent*
- $G(t_f)$ is a *fixed point*
- If $G(s)$ converges for all initial states s, G is *strongly convergent*

Emergence

- In a *static network*, the properties of nodes, links, and mapping unction remain unchanged over time
- In a *dynamic network*, the number of nodes and links, the shape of the mapping function, and other properties of the graph change over time
- Time-varying changes leading to structural reorganization in a network—is called **emergence**
- An *emergent network* is formed by starting at some predefined *initial state* and then transitioning through a series of small changes into an *end state*

Open-loop & Feedback-loop

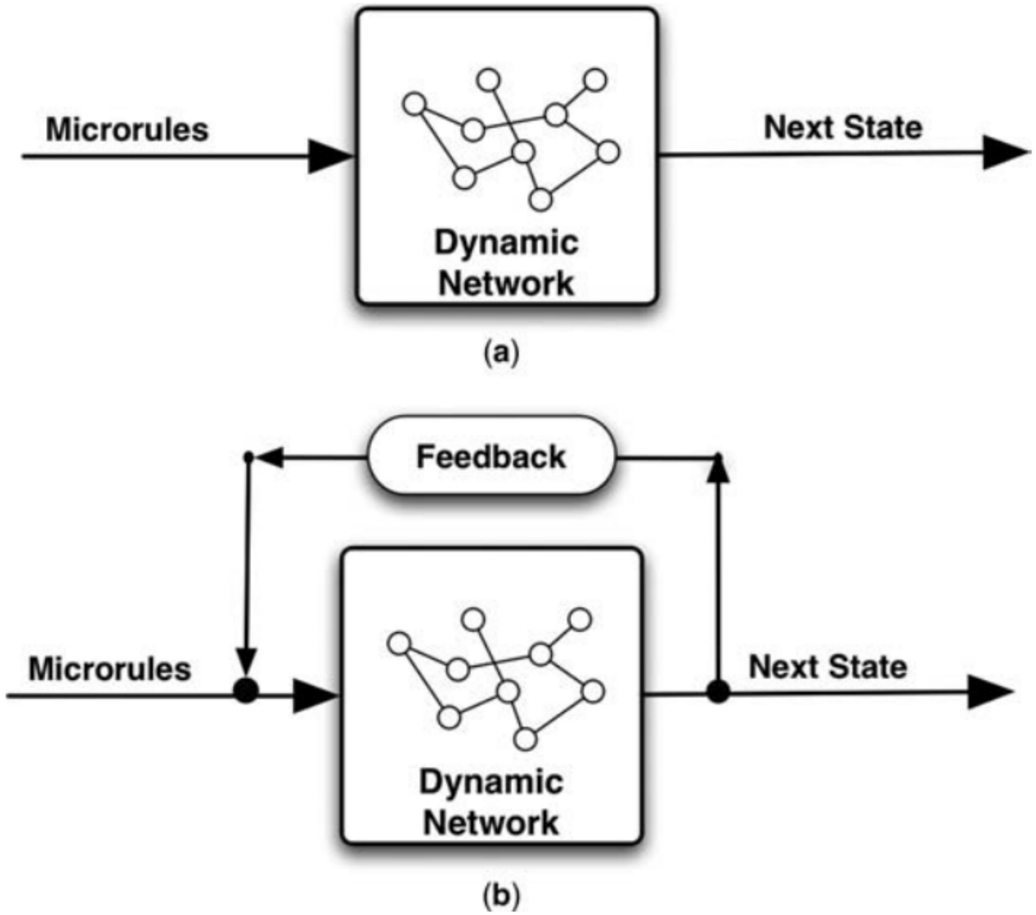


Figure Emergence in a dynamic network: (a) open-loop emergence, typical of intrinsic or genetic evolution; (b) feedback-loop emergence, typical of external or environmental forces shaping a network.

Hub Emergence

- Initially, *G*(0) is a random network with *n* nodes and *m* links
- At each time-step, select a node and link at random
  - The randomly selected node is selected if its degree is higher than that of the randomly selected link's head node
  - Rewire the link

```
public void NW_doIncreaseDegree(){
  //Rewire network to increase node degrees
  int random_node = (int)(nNodes * Math.random()); //A random node
  int random_link = (int)(nLinks * Math.random()); //A random link
  int to_node = Link[random_link].head;          //Link's to_node
  int from_node = Link[random_link].tail;        //Anchor node
  if(node[random_node].degree > node[to_node].degree){
     if(NW_doAddLink(node[from_node].name, node[random_node].name))
        NW_doCutLink(random_link);                //Replace random link
     else  Message = "Warning: Duplicate Links Not Allowed.";
    }
  }//NW_doIncreaseDegree
```

Hub Emergence

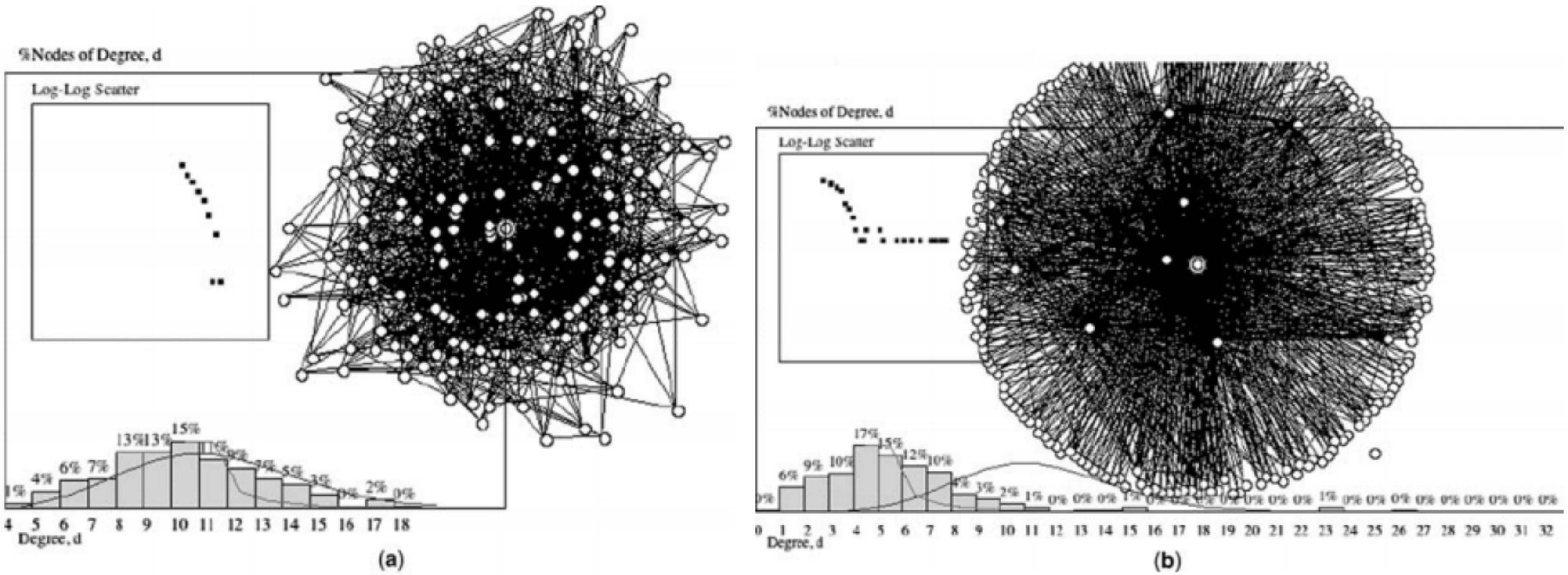


Figure Emergence of a network with high-degreed hubs from a random network: (a) random network, $G(0)$—the degree sequence distribution before emergence; (b) hub emerged network, $G(160{,}000)$—the degree sequence distribution after emergence. The initial network, $G(0)$, $n = 200$, $m = 1000$, was generated by the Erdos–Renye procedure.

Cluster Emergence

- Select a random link and random node
- Rewire the link to point to the new (random) node, if the overall cluster coefficient remains the same or is increased
- If the cluster coefficient decreases as a result of rewiring, revert to the topology of the previous time-step
- Repeat this microrule indefinitely, or until stopped

Cluster Emergence

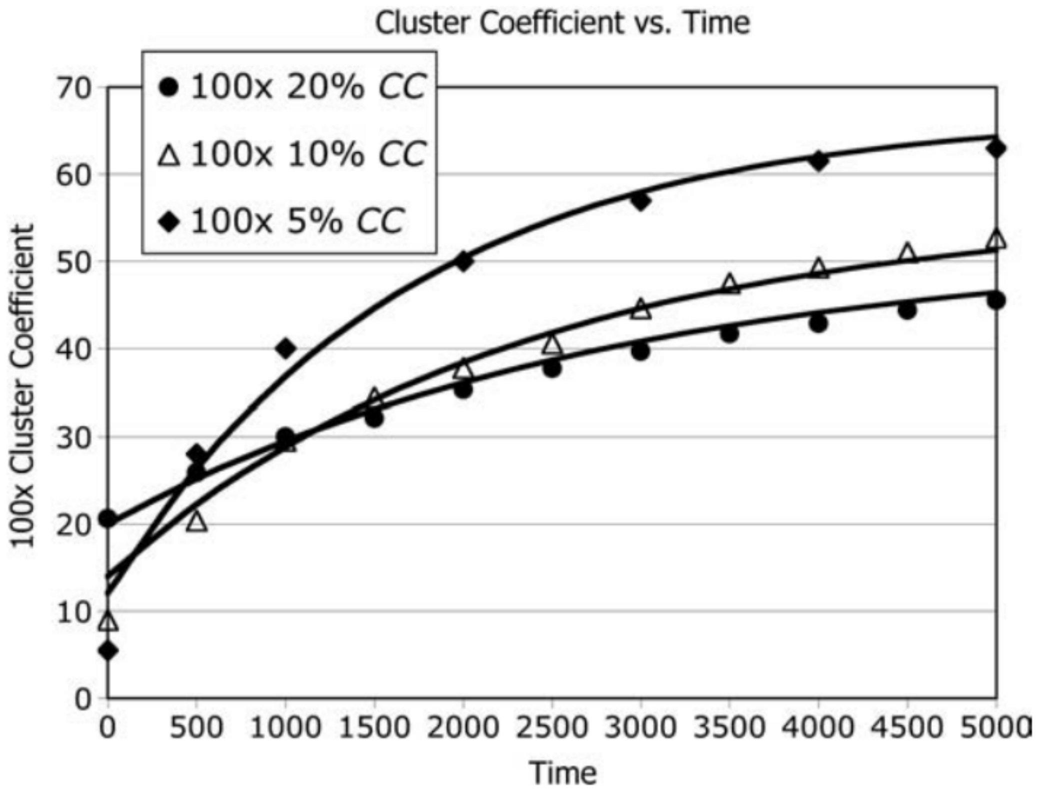


Figure Cluster coefficient versus time for $n = 100$, $m = 249$ (5%), $m = 500$ (10%), and $m = 1000$ (20%).

Epidemics

- A *network epidemic* is a process of widespread and rapid propagation of a contagion through a network
  - Typically, the contagion is a condition of network nodes (working, failing, dormant, active, etc.) brought about by adjacent nodes through propagation along one or more links
  - Infection rate: the probability that propagation of the contagion at node *v* successfully infects adjacent nodes

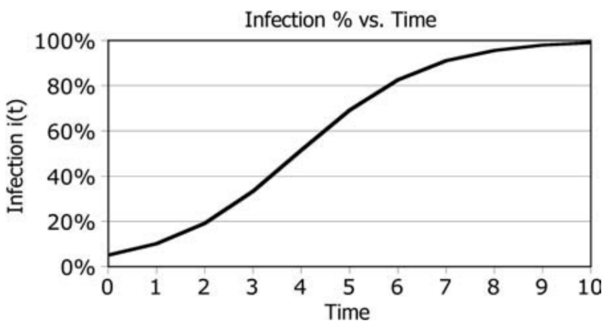
Kermack–McKendrick Model

- Assumes homogeneous and a very large population
  - Everyone has an equal chance of contracting the disease from anyone else
  - n is unbounded
- Each node is classified to one of four states
  - *Susceptible*: it can possibly become infected
  - *Infected*: it has contracted the contagion
  - *Recovered*: it has recovered from the infection and is now immune to future infection
  - *Removed*: it has died from the effects of the infection

Kermack–McKendrick Equations

We define γ as the rate of infection, Δ as the rate of removal or recovery, and t_r as the duration of a node's infected state before transition to a removed or recovered state.

Let $S(t)$ be the number of susceptible actors, $I(t)$ the number of infected actors, and $R(t)$ the number of actors removed from the population as a result of death or immunity after recovering from the illness, at time t. In a finite population, $n = S(t) + I(t) + R(t)$. The Kermack–McKendrick equations relate S, I, and R to one another through their time rate of change and initial conditions, as follows:



$$\frac{\delta S(t)}{\delta t} = -\gamma S(t)I(t); \;\; S(0) = S_0 \tag{I}$$

$$\frac{\delta I(t)}{\delta t} = \gamma S(t)I(t) - \Delta I(t); \;\; I(0) = I_0 \tag{II}$$

$$\frac{\delta R(t)}{\delta t} = \Delta I(t); \;\; R(0) = R_0 \tag{III}$$

$$S(t) + I(t) + R(t) = n \tag{IV}$$

Other Models

- Susceptible–Infected–Removed (SIR) Model
- Susceptible–Infected–Susceptible (SIS) Model

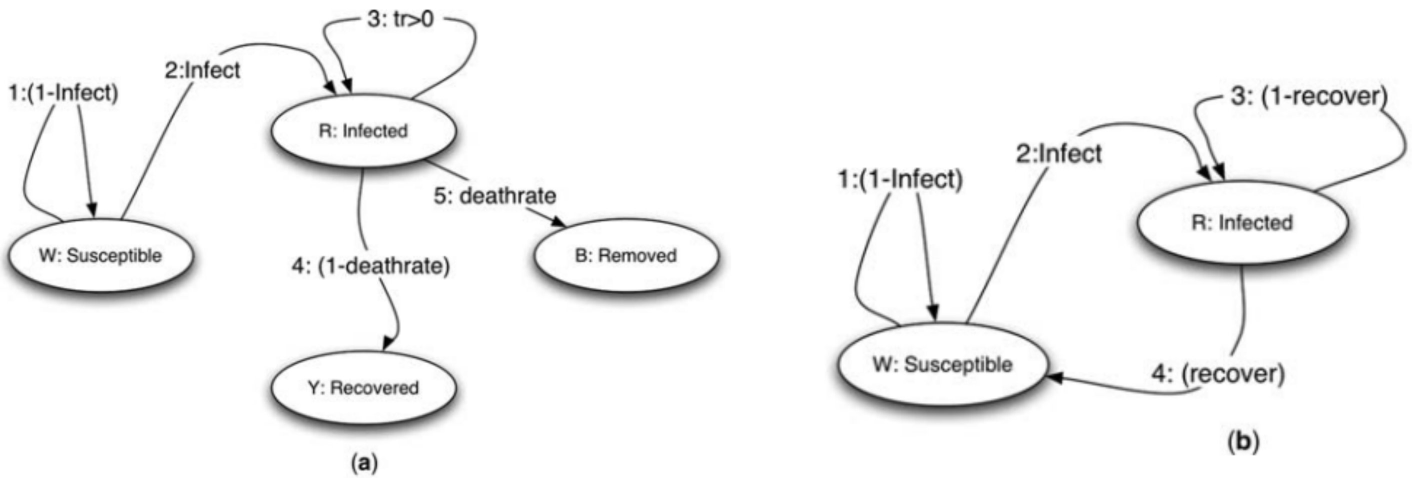


Figure Models of epidemics: (a) SIR; (b) SIS (color code : W—susceptible = white; R—infected = red; B—removed = black; Y—recovered = yellow).

Synchrony

- A network is said to *synchronize* if the values of all of its nodes converge to some constant as the time rate of change of all of its node values approaches zero
- A dynamic network is
  - *stable* if the value of its nodes synchronize
  - *transient* if its node values oscillate
  - *bistable* if its nodes oscillate between fixed values
  - *unstable* or *chaotic*, otherwise

Chaotic Maps

- A dynamic network is said to *sync* if, starting at some initial state *G*(0), it evolves in finite time to another state, *G*(t*), and stays there, forever.
- *G*(t*) is called a strange attractor
- Networks that appear to bounce around from one state to another in no apparent pattern are considered *chaotic*
- Networks that oscillate between two or more strange attractors are called *oscillators*

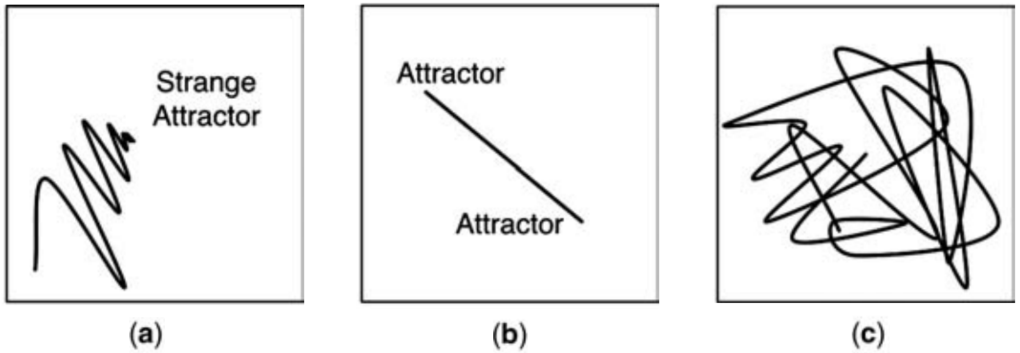


Figure Chaotic maps: trajectories of (a) a stable node as it reaches its strange attractor, (b) a bistable node as it oscillates between two attractors, and (c) a chaotic node as it wanders around in state space.

Network Stability

- *Stability* describes nodes and networks that recover from disruptions in their state
- A stable network will recover—perhaps very slowly—while an unstable network will not
- Two general techniques used for analyzing the stability of a dynamic system
    - The Laplace or Z-transform (Lyapunov) method (Lago-Fernandez, 1999)
    - The Laplacian eigenvalue (spectral decomposition) method (Wang, 2002)

TABLE Sync Results for Regular and Nonregular Networks

Network	Bistable?	Comments
Barbell	No	Bistable oscillator
Line	No	50%–50% oscillations
2-Regular	Yes	Odd-sized cycles
Complete	Yes	Odd-sized cycles
Toroid	Yes	Odd-sized cycles
Binary tree	No	No cycles
Hypercube	No	Even-sized cycles
Random	Nearly always	Likely has odd-sized cycle
Small-world	Yes	Has clusters
Scale-free	Nearly always	Likely has odd-sized cycle
Star	No	No cycles
Ring	$N =$ odd number	Odd-sized cycle

Influence Networks

- An *influence network* is a (directed or undirected) network, $G = \{ N, L, f \}$ containing nodes N; directed, weighted links L; and mapping function $f : N \times N$ that defines the topology
- Nodes are called *actors*
- Links are called *influences*
- Two major types
  - Undirected networks such as the buzz network (Rosen, 2000)
  - directed networks such as the social networks formed by negotiating parties in a human group

A Buzz Network

- Let *S*(*t*) be the state vector representing an actor's position (a product, idea, political belief, etc.)
  - *S*(*t*) = -1 if the position is negative
  - *S*(*t*) = 0 if neutral
  - *S*(*t*) = +1 if positive

$$-1 \leq S(t) \leq 1$$

$$S(t) = [s_1, s_2, s_3, \ldots, s_n]^{\mathrm{T}} = \text{state vector of } G; \quad S_j(t) = \text{row } j \text{ of } S(t)$$

Buzz State Equation

$$S(t+1) = \phi S(t) + (1-\phi)\sum_{j\sim i} \frac{S_j(t)}{\text{degree}(i)}$$

where the summation $\sum_{j\sim i}$ is over the neighbors of node i, specifically, $j \sim i$ and $s_i = [-1, 1]$, $i = 1,2,\ldots,n$. $0 \leq \phi \leq 1$ is the *stubbornness factor*.

Properties

- Each seed actor spreads his or her product endorsement to adjacent neighbors, which in turn is spread to their neighbors, and so forth, much like the spread of an infection
- The state of each actor depends on the strength of the individual's convictions and ϕıe positions of adjacent neighbors
- Stability
  - Buzz networks reach a consensus that is influenced by the dominant (hub) node
  - Consensus: all nodes reach the same final state

Two-Party Negotiation

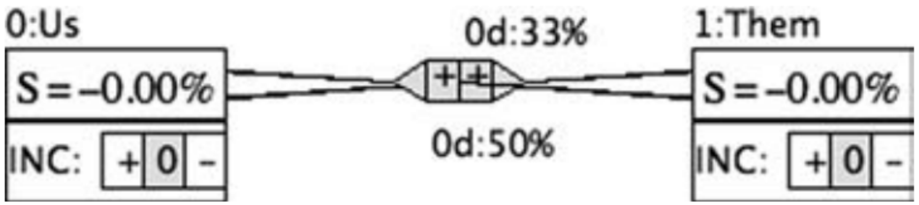


Figure I-net representation of a two-party negotiation: actor Us versus actor Them, and influences of $\frac{1}{3}$ and $\frac{1}{2}$, respectively.

- Two-party negotiation is a process of interactive compromise
  - The two actors must narrow the difference in positions as perceived by each other if the network is to reach a consensus

For example, after one timestep:

$$\text{Us}(1) = \text{Us}(0) + 50\%(\text{Them}(0) - \text{Us}(0)) = 1 + 50\%(-2) = 0$$
$$\text{Them}(1) = \text{Them}(0) + 33\%(\text{Us}(0) - \text{Them}(0)) = -1 + 33\%(2) = -33\%$$

Two-Party Negotiation

$$s_i(t+1) = s_i(t) + \sum_j [s_j(t) - s_i(t)]\phi^T i,\ \ j;$$

$$\text{Us}(2) = \text{Us}(1) + 50\%(\text{Them}(1) - \text{Us}(1)) = 0 + 50\%(-0.33) = -0.167$$
$$\text{Them}(2) = \text{Them}(1) + 33\%(\text{Us}(1) - \text{Them}(1)) = -0.33 + 33\%(0.33) = -22\%$$

$$\text{Us}(3) = \text{Us}(2) + 50\%(\text{Them}(2) - \text{Us}(2)) = 0.167 + 50\%(-0.056) = -19\%$$
$$\text{Them}(3) = \text{Them}(2) + 33\%(\text{Us}(2) - \text{Them}(2)) = -22 + 33\%(0.056) = -20\%$$

- The two actor's states converge in the end
- Differences drop to zero and both actors reach a consensus state equal to negative 20%, which favors the initial position of Them
- Therefore, 1:Them is more influential than actor 0:Us.

References

- Lago-Fernandez, L. F., R. Huerta, F. Corbacho, and J. Siguenza, Fast response and temporal coding on coherent oscillations in small-world networks, available as *Condensed Matter/9909379* (1999).
- Wang, X. and G. Chen, Pinning control of scale-free dynamical networks, *Physica* A 310:521–531 (2002).
- Rosen, E., *The Anatomy of Buzz, Doubleday-Random House*, 2000.