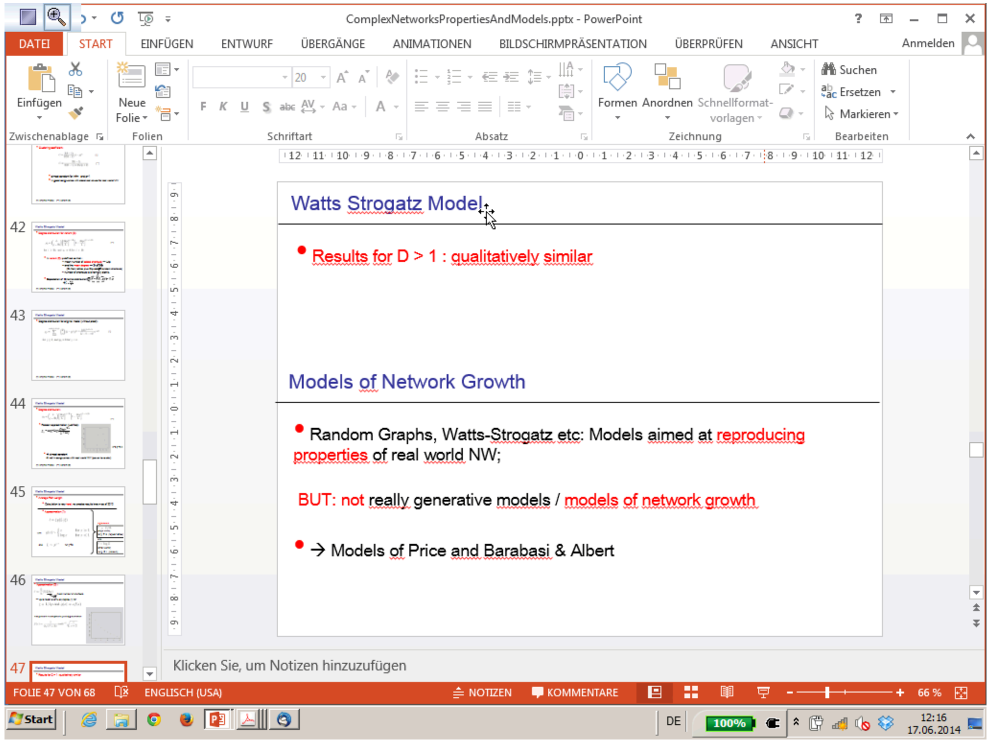
Script generated by TTT

Title: groh: profile1 (17.06.2014)

Date: Tue Jun 17 12:16:21 CEST 2014

Duration: 91:12 min

Pages: 99



Price's Model

- Basic principle:
  „the rich get richer“
  „Matthew effect“ („For to every one that hath shall be given...“ Bible: Mt25:29)
  „preferential attachment“

- Assume directed citation NW:
  - p_k: fraction of nodes with in-degree k,
  - each node (paper) has av. out degree m
  - mean out-deg. $\stackrel{!}{=}$ mean in-deg. → $\sum_k k p_k = m$

- iteratively build graph by adding new vertices (and associated directed (out)edges from these nodes)

Price's Model

- Basic principle:
  „the rich get richer“
  „Matthew effect“ („For to every one that hath shall be given...“ Bible: Mt25:29)
  „preferential attachment“

- Assume directed citation NW:
  - p_k: fraction of nodes with in-degree k,
  - each node (paper) has av. out degree m
  - mean out-deg. $\stackrel{!}{=}$ mean in-deg. → $\sum_k k p_k = m$

- iteratively build graph by adding new vertices (and associated directed (out)edges from these nodes)

Price's Model

- Basic principle:
  „the rich get richer"
  „Matthew effect" („For to every one that hath shall be given..." Bible: Mt25:29)
  „preferential attachment"

- Assume directed citation NW:
  - p_k: fraction of nodes with in-degree k,
  - each node (paper) has av. out degree m
  - mean out-deg. $\stackrel{!}{=}$ mean in-deg. → $\sum_k k p_k = m$

- iteratively build graph by adding new vertices (and associated directed (out)edges from these nodes)

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)
  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1} m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)
  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1} m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)
  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1} m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)

  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1}\, m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)

  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1}\, m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

- probability for a paper X to get cited by a new paper is proportional to number of existing citations of X (X's in-degree)

  - initial „starting in-degree" k_0=1
  - → prob. that new edge attaches to any node with in-deg. k ==

$$\frac{(k+1)p_k}{\sum_k (k+1)p_k} = \frac{(k+1)p_k}{m+1}$$

  - Since mean number of out-edges per added vertex == m → mean number of new in-edges to nodes with current in-degree k is ==

$$x = \frac{(k+1)p_k}{m+1}\, m$$

  - mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1

- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k

- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = \left[kp_{k-1,n} - (k+1)p_{k,n}\right]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

from previous slide:

- mean number of nodes with in-degree k (which is np_k) decreases by x because their in-degree changes to k+1
- mean number of nodes with in-degree k also increases because of nodes having previously k-1 and now have k
- → the net change in the quantity np_k per added vertex satisfies:

$$(n+1)p_{k,n+1} - np_{k,n} = [kp_{k-1,n} - (k+1)p_{k,n}]\frac{m}{m+1}$$

for $k \geq 1$, or

$$(n+1)p_{0,n+1} - np_{0,n} = 1 - p_{0,n}\frac{m}{m+1},$$

for $k = 0$.

Price's Model

- Computing stationary solutions $p_{k,n+1} = p_{k,n} = p_k$ of this equation we find:

$$p_k \sim k^{-(2+1/m)} \quad \text{for } n \to \infty$$

- → the desired power law distribution
- we see: „the rich get richer" → power law

Barabasi-Albert Model

- same principles as Price's but use undirected edges, intended as model for the WWW
- nodes with fixed degree m are added to the network at each iteration
- edges connect to nodes with probability proportional to current degree of node
- → analogous analysis as for Price's leads to

$$p_k \sim k^{-3} \quad \text{for } n \to \infty$$

Barabasi-Albert Model

- same principles as Price‘s but use undirected edges, intended as model for the WWW
- nodes with fixed degree m are added to the network at each iteration
- edges connect to nodes with probability proportional to current degree of node

- → analogous analysis as for Price‘s leads to

$$p_k \sim k^{-3} \qquad \text{for } n \to \infty$$

Barabasi-Albert Model

- same principles as Price‘s but use undirected edges, intended as model for the WWW
- nodes with fixed degree m are added to the network at each iteration
- edges connect to nodes with probability proportional to current degree of node

- → analogous analysis as for Price‘s leads to

$$p_k \sim k^{-3} \qquad \text{for } n \to \infty$$

Barabasi-Albert Model

- same principles as Price‘s but use undirected edges, intended as model for the WWW
- nodes with fixed degree m are added to the network at each iteration
- edges connect to nodes with probability proportional to current degree of node

- → analogous analysis as for Price‘s leads to

$$p_k \sim k^{-3} \qquad \text{for } n \to \infty$$

Barabasi-Albert Model and Price‘s Model

- crucial: linear preferential attachment
- found in a number of real world NW (e.g. citation NW)
- Barabasi-Albert: undirected (not like WWW)
- directed version of Barabasi Albert: attachment prop to sum of out and in- degree: not realistic for e.g. the WWW but for social NW?!
- Price: generates directed acyclic graph: not realistic for SN and WWW
- out-degree of WWW: power-law, Price + BA: constant

Processes on Networks: Percolation

- Assume structure of NW known: what about processes on networks (e.g. spread of info in SN)?
- Percolation: Randomly assign states „occupied“ and „not occupied“ to either edges or vertices → investigate occupied and un-occupied „parts“ separately
- Similarly: Take out nodes / edges, ask for network resilience. E.g. measure resilience via connectednes (e.g. existence of giant component)
- Example: configuration random graph model with power law degree distribution $p_k \sim k^{-\alpha}$; investigate phase transition to / from existing giant component when „occupying“ nodes

Processes on Networks: Percolation

- Assume structure of NW known: what about processes on networks (e.g. spread of info in SN)?
- Percolation: Randomly assign states „occupied“ and „not occupied“ to either edges or vertices → investigate occupied and un-occupied „parts“ separately
- Similarly: Take out nodes / edges, ask for network resilience. E.g. measure resilience via connectednes (e.g. existence of giant component)
- Example: configuration random graph model with power law degree distribution $p_k \sim k^{-\alpha}$; investigate phase transition to / from existing giant component when „occupying“ nodes

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q be the constant fraction of occupied („functional“ / „working“) vertices
- → for vertex with degree k: fraction of occupied neighbors:

$$p(k'|k) = \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → probability that any node is connected to k´ occupied nodes is

$$p_{k'} = p(k') = \sum_k p(k'|k)\, p(k) = \sum_k p(k'|k)\, p_k = \sum_{k=k'}^{\infty} p_k \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → (analysis similar to slide 29 / 30) → for $\alpha \leq 3$: independent of positive q: giant component always exists → random „removal“ of (1-q) nodes leaves NW „unimpressed“

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q be the constant fraction of occupied („functional“ / „working“) vertices
- → for vertex with degree k: fraction of occupied neighbors:

$$p(k'|k) = \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → probability that any node is connected to k´ occupied nodes is

$$p_{k'} = p(k') = \sum_k p(k'|k)\, p(k) = \sum_k p(k'|k)\, p_k = \sum_{k=k'}^{\infty} p_k \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → (analysis similar to slide 29 / 30) → for $\alpha \leq 3$: independent of positive q: giant component always exists → random „removal“ of (1-q) nodes leaves NW „unimpressed“

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q be the constant fraction of occupied („functional“ / „working“) vertices
- → for vertex with degree k: fraction of occupied neighbors:

$$p(k'|k) = \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → probability that any node is connected to k´ occupied nodes is

$$p_{k'} = p(k') = \sum_k p(k'|k)\, p(k) = \sum_k p(k'|k)\, p_k = \sum_{k=k'}^{\infty} p_k \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → (analysis similar to slide 29 / 30) → for α ≤ 3 : independent of positive q: giant component always exists → random „removal“ of (1-q) nodes leaves NW „unimpressed“

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q be the constant fraction of occupied („functional“ / „working“) vertices
- → for vertex with degree k: fraction of occupied neighbors:

$$p(k'|k) = \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → probability that any node is connected to k´ occupied nodes is

$$p_{k'} = p(k') = \sum_k p(k'|k)\, p(k) = \sum_k p(k'|k)\, p_k = \sum_{k=k'}^{\infty} p_k \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → (analysis similar to slide 29 / 30) → for α ≤ 3 : independent of positive q: giant component always exists → random „removal“ of (1-q) nodes leaves NW „unimpressed“

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q be the constant fraction of occupied („functional“ / „working“) vertices
- → for vertex with degree k: fraction of occupied neighbors:

$$p(k'|k) = \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

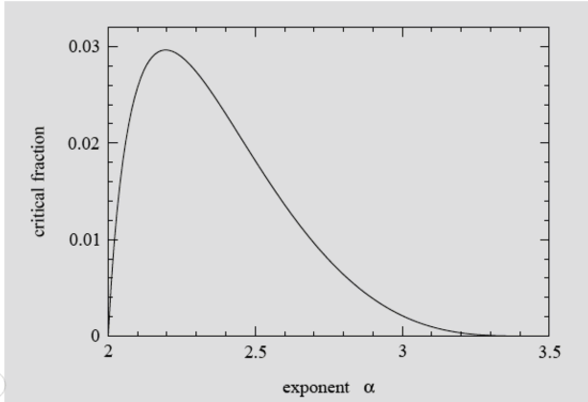
- → probability that any node is connected to k´ occupied nodes is

$$p_{k'} = p(k') = \sum_k p(k'|k)\, p(k) = \sum_k p(k'|k)\, p_k = \sum_{k=k'}^{\infty} p_k \binom{k}{k'} q^{k'} (1-q)^{k-k'}$$

- → (analysis similar to slide 29 / 30) → for α ≤ 3 : independent of positive q: giant component always exists → random „removal“ of (1-q) nodes leaves NW „unimpressed“

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q_k be the fraction of occupied vertices dependent on k (e.g. remove / occupy only high degree nodes)
- → (analysis) → only a small fraction of the high degree nodes needs to be removed to destroy the giant component

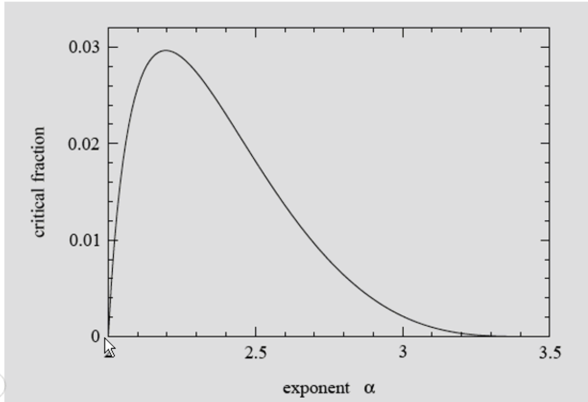


[1]

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q_k be the fraction of occupied vertices dependent on k (e.g. remove / occupy only high degree nodes)

- → (analysis) → only a small fraction of the high degree nodes needs to be removed to destroy the giant component

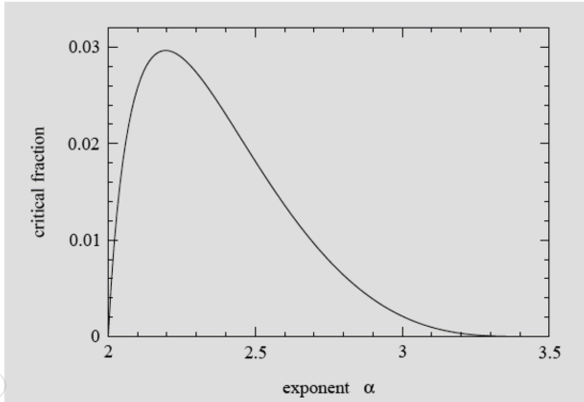


[1]

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q_k be the fraction of occupied vertices dependent on k (e.g. remove / occupy only high degree nodes)

- → (analysis) → only a small fraction of the high degree nodes needs to be removed to destroy the giant component



[1]

Processes on Networks: Epidemiology

- disease: nodes V = susceptible s ⊎ infective i ⊎ recovered r
- susceptibles: can be infected;
  infective: have the disease and are contageous,
  recovered: have had the disease and are immune (or dead)

- infection probability / rate β , recovering probability γ
- → SIR model („fully mixed“):

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

Processes on Networks: Epidemiology

- disease: nodes V = susceptible s ⊎ infective i ⊎ recovered r
- susceptibles: can be infected;
  infective: have the disease and are contageous,
  recovered: have had the disease and are immune (or dead)

- infection probability / rate β , recovering probability γ
- → SIR model („fully mixed“):

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

Processes on Networks: Epidemiology

- disease: nodes V = susceptible s $\dot\cup$ infective i $\dot\cup$ recovered r
- susceptibles: can be infected;
  infective: have the disease and are contageous,
  recovered: have had the disease and are immune (or dead)

- infection probability / rate β , recovering probability γ
- → SIR model („fully mixed“):

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

Processes on Networks: Epidemiology

- disease: nodes V = susceptible s $\dot\cup$ infective i $\dot\cup$ recovered r
- susceptibles: can be infected;
  infective: have the disease and are contageous,
  recovered: have had the disease and are immune (or dead)

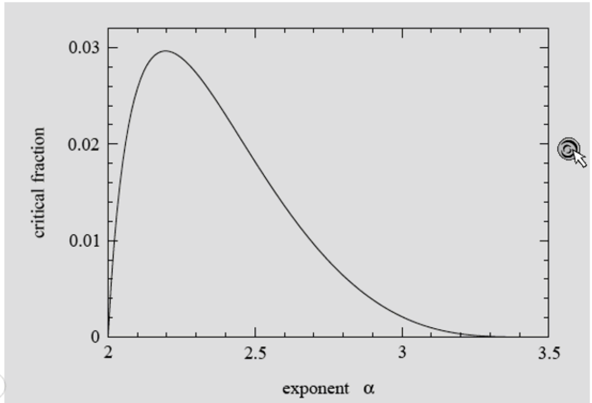
- infection probability / rate β , recovering probability γ
- → SIR model („fully mixed“):

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

Processes on Networks: Percolation

- degree distr.: $p_k \sim k^{-\alpha}$;
- let q_k be the fraction of occupied vertices dependent on k (e.g. remove / occupy only high degree nodes)

- → (analysis) → only a small fraction of the high degree nodes needs to be removed to destroy the giant component



[1]

Processes on Networks: Epidemiology

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

- now: „play“ the model on a network (e.g. human contact network) and investigate perlocation effects:
  - β (infection probability per unit time) and γ (recovery prob. p.u.t.): drawn from probability distributions $P_i(\beta)$ and $P_i(\gamma)$ --> problem is equivalent to edge-percolation problem with edge occupation probability
    $$T = 1 - \int_0^\infty P_i(\beta) P_r(\gamma)\, e^{-\beta/\gamma}\, \mathrm{d}\beta\, \mathrm{d}\gamma.$$
  - investigate dissociation into components (internally connected by unoccupied egdes)
  - corresp. phase transitions: transitions from epidemic outbreak (giant component) vs. controlled state (small components)
  - result: power law with α≤3→ giant component also always exists here

Processes on Networks: Epidemiology

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

- now: „play" the model on a network (e.g. human contact network) and investigate perlocation effects:
  - β (infection probability per unit time) and γ (recovery prob. p.u.t.): drawn from probability distributions $P_i(\beta)$ and $P_i(\gamma)$ --> problem is equivalent to edge-percolation problem with edge occupation probability

$$T = 1 - \int_0^\infty P_i(\beta) P_r(\gamma)\, \mathrm{e}^{-\beta/\gamma}\, \mathrm{d}\beta\, \mathrm{d}\gamma.$$

  - investigate dissociation into components (internally connected by unoccupied egdes)
  - corresp. phase transitions: transitions from epidemic outbreak (giant component) vs. controlled state (small components)
  - result: power law with α≤3→ giant component also always exists here

Processes on Networks: Epidemiology

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

- now: „play" the model on a network (e.g. human contact network) and investigate perlocation effects:
  - β (infection probability per unit time) and γ (recovery prob. p.u.t.): drawn from probability distributions $P_i(\beta)$ and $P_i(\gamma)$ --> problem is equivalent to edge-percolation problem with edge occupation probability

$$T = 1 - \int_0^\infty P_i(\beta) P_r(\gamma)\, \mathrm{e}^{-\beta/\gamma}\, \mathrm{d}\beta\, \mathrm{d}\gamma.$$

  - investigate dissociation into components (internally connected by unoccupied egdes)
  - corresp. phase transitions: transitions from epidemic outbreak (giant component) vs. controlled state (small components)
  - result: power law with α≤3→ giant component also always exists here

Processes on Networks: Epidemiology

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

- now: „play" the model on a network (e.g. human contact network) and investigate perlocation effects:
  - β (infection probability per unit time) and γ (recovery prob. p.u.t.): drawn from probability distributions $P_i(\beta)$ and $P_i(\gamma)$ --> problem is equivalent to edge-percolation problem with edge occupation probability

$$T = 1 - \int_0^\infty P_i(\beta) P_r(\gamma)\, \mathrm{e}^{-\beta/\gamma}\, \mathrm{d}\beta\, \mathrm{d}\gamma.$$

  - investigate dissociation into components (internally connected by unoccupied egdes)
  - corresp. phase transitions: transitions from epidemic outbreak (giant component) vs. controlled state (small components)
  - result: power law with α≤3→ giant component also always exists here

Processes on Networks: Epidemiology

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta i s, \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta i s - \gamma i, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i$$

- now: „play" the model on a network (e.g. human contact network) and investigate perlocation effects:
  - β (infection probability per unit time) and γ (recovery prob. p.u.t.): drawn from probability distributions $P_i(\beta)$ and $P_i(\gamma)$ --> problem is equivalent to edge-percolation problem with edge occupation probability

$$T = 1 - \int_0^\infty P_i(\beta) P_r(\gamma)\, \mathrm{e}^{-\beta/\gamma}\, \mathrm{d}\beta\, \mathrm{d}\gamma.$$

  - investigate dissociation into components (internally connected by unoccupied egdes)
  - corresp. phase transitions: transitions from epidemic outbreak (giant component) vs. controlled state (small components)
  - result: power law with α≤3→ giant component also always exists here

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from" highly (out-degree-)weighted nodes == „Authority" :
- node with high out-degree „to" highly (in-degree-)weighted nodes == „Hub")
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \qquad \rightarrow \qquad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from" highly (out-degree-)weighted nodes == „Authority" :
- node with high out-degree „to" highly (in-degree-)weighted nodes == „Hub")
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \qquad \rightarrow \qquad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from" highly (out-degree-)weighted nodes == „Authority" :
- node with high out-degree „to" highly (in-degree-)weighted nodes == „Hub")
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \qquad \rightarrow \qquad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from" highly (out-degree-)weighted nodes == „Authority" :
- node with high out-degree „to" highly (in-degree-)weighted nodes == „Hub")
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \qquad \rightarrow \qquad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from“ highly (out-degree-)weighted nodes == „Authority“ :
- node with high out-degree „to“ highly (in-degree-)weighted nodes == „Hub“)
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \quad \rightarrow \quad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda\mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from“ highly (out-degree-)weighted nodes == „Authority“ :
- node with high out-degree „to“ highly (in-degree-)weighted nodes == „Hub“)
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \quad \rightarrow \quad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda\mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- We have seen: Feedback/Eigenvector-Centrality / Page Rank: weight of vertex i (neglecting heuristic corrections):

$$x_i = \lambda^{-1} \sum_j A_{ij} x_j \text{ for some } \lambda > 0 \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

- instead of only looking at in-degrees also look at high out-degree
- node with high in-degree „from“ highly (out-degree-)weighted nodes == „Authority“ :
- node with high out-degree „to“ highly (in-degree-)weighted nodes == „Hub“)
- in-degree based weights: x; out-degree-based weights y

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y} \quad \rightarrow \quad \mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda\mu \mathbf{x}$$

Processes on Networks: Searching and Navigating

- Instead of „Search engine“-type of network search (one big crawl), perform local crawls
- especially suitable in decentralized scenarios
- example: BFS: „do you have the info“? either „yes“ or „no, but will forward to my nighbors“
- variant by Adamic: instead of asking all neighbors : answer will be „no but i have k neighbors → asker can choose highest degree node to „pass on the query baton to“ → if e.g. power law: high degree nodes cover NW very well.
- other variants: see next chapter

Processes on Networks: Searching and Navigating

- Instead of „Search engine“-type of network search (one big crawl), perform local crawls
- especially suitable in decentralized scenarios
- example: BFS: „do you have the info“? either „yes“ or „no, but will forward to my nighbors“
- variant by Adamic: instead of asking all neighbors : answer will be „no but i have k neighbors → asker can choose highest degree node to „pass on the query baton to“ → if e.g. power law: high degree nodes cover NW very well.
- other variants: see next chapter

Processes on Networks: Searching and Navigating

- Navigation in Networks: Milgram experiment showed: Short paths exist and people can find them → some notion of distance / measure of relatedness obviously necessary
- Poisson random graph → easy to achieve: short paths exist;
- Open question: how do people find these paths? nodes i.g. do not „know“ shortest paths to any other node --> routing strategy
- Kleinberg Model [3]: Variation of Watts Strogatz Model respecting spatial distance: investigate question of navigation in networks

Processes on Networks: Searching and Navigating

- Navigation in Networks: Milgram experiment showed: Short paths exist and people can find them → some notion of distance / measure of relatedness obviously necessary
- Poisson random graph → easy to achieve: short paths exist;
- Open question: how do people find these paths? nodes i.g. do not „know“ shortest paths to any other node --> routing strategy
- Kleinberg Model [3]: Variation of Watts Strogatz Model respecting spatial distance: investigate question of navigation in networks

Kleinberg Model

- Put nodes on a n x n grid. Distance: Manhattan:

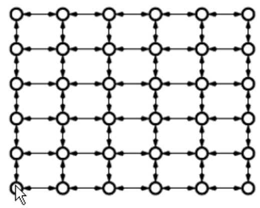
$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range“ edges: Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

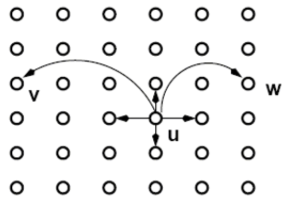
examples:

A)



q1 = 1, q2 = 0

B)



edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- Put nodes on a n x n grid. Distance: Manhattan:

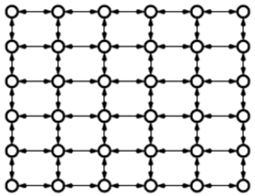
$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range" edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

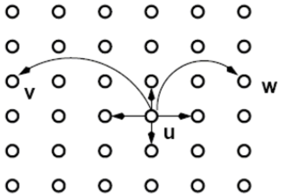
examples:

A)



q1 = 1, q2 = 0

B)



edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- Put nodes on a n x n grid. Distance: Manhattan:

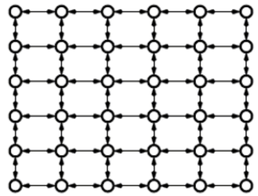
$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range" edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

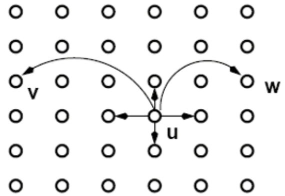
examples:

A)



q1 = 1, q2 = 0

B)



edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- Put nodes on a n x n grid. Distance: Manhattan:

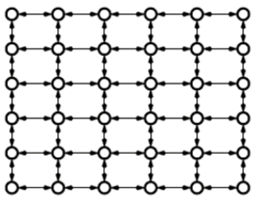
$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range" edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

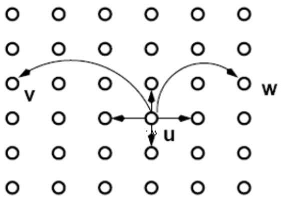
examples:

A)



q1 = 1, q2 = 0

B)



edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- Put nodes on a n x n grid. Distance: Manhattan:

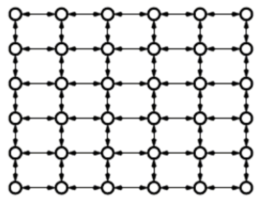
$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range" edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

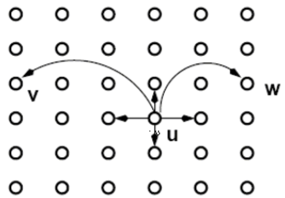
examples:

A)



q1 = 1, q2 = 0

B)



edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

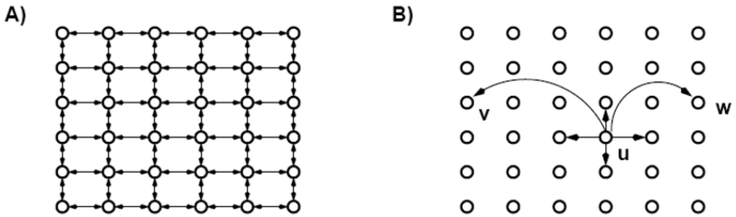
- Put nodes on a n x n grid. Distance: Manhattan:

$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \leq q1$ (regular local contacts)
- Each node i: Additional q2 other „long range" edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

examples:



q1 = 1, q2 = 0 edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- local (decentralized) knowledge:
  - Each node only knows only:
    - Its adjacent nodes
    - The grid's principle structure
    - Position of target node on the grid
    - Positions and long-range contacts of nodes on the message path so far

- (Search-) algorithm with only local knowledge: „decentralized"

Homophily and Distance

- Homophily: supports triadic closure and thus high clustering in SN

- Homophily principle: route questions, information Milgram letters etc. to nodes that are similar to you (socially, geographically, profession-wise) .

- Homophily principle **alone** is not sufficient for routing: if you **only** know „your kind" (socially, geographically, profession-wise) no efficient routing (searching for information, information dissemination etc.) is possible → suitable „distribution", „heterogenity" necessary (compare short-cuts and local (cluster) edges in Watts-Strogatz model)

- Example geographic homophily: geographical distribution of social relations: if too local: too many steps required, if too scattered: no efficient „greedy" decentralized routing possible

Homophily and Distance

- Homophily: supports triadic closure and thus high clustering in SN

- Homophily principle: route questions, information Milgram letters etc. to nodes that are similar to you (socially, geographically, profession-wise) .

- Homophily principle **alone** is not sufficient for routing: if you **only** know „your kind" (socially, geographically, profession-wise) no efficient routing (searching for information, information dissemination etc.) is possible → suitable „distribution", „heterogenity" necessary (compare short-cuts and local (cluster) edges in Watts-Strogatz model)

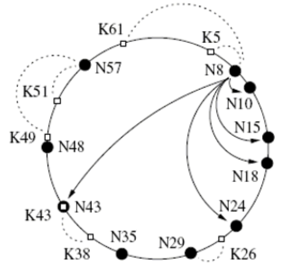
- Example geographic homophily: geographical distribution of social relations: if too local: too many steps required, if too scattered: no efficient „greedy" decentralized routing possible

Homophily and Distance

- Homophily: supports triadic closure and thus high clustering in SN
- Homophily principle: route questions, information Milgram letters etc. to nodes that are similar to you (socially, geographically, profession-wise) .
- Homophily principle **alone** is not sufficient for routing: if you **only** know „your kind“ (socially, geographically, profession-wise) no efficient routing (searching for information, information dissemination etc.) is possible → suitable „distribution“, „heterogenity“ necessary (compare short-cuts and local (cluster) edges in Watts-Strogatz model)
- Example geographic homophily: geographical distribution of social relations: if too local: too many steps required, if too scattered: no efficient „greedy“ decentralized routing possible

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n +* 2^{i-1}*) mod* 2^m*)* → routing in log(N) steps possible



Finger Table of N8

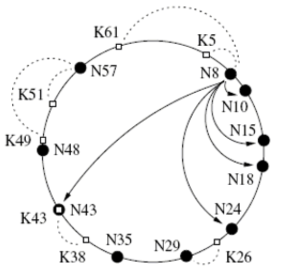
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n +* 2^{i-1}*) mod* 2^m*)* → routing in log(N) steps possible



Finger Table of N8

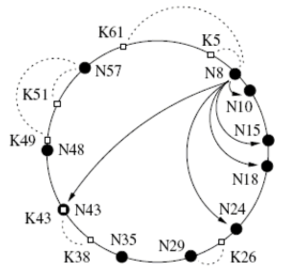
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n +* 2^{i-1}*) mod* 2^m*)* → routing in log(N) steps possible



Finger Table of N8

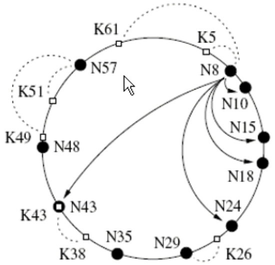
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n + 2^{i-1}) mod 2^m)* → routing in log(N) steps possible



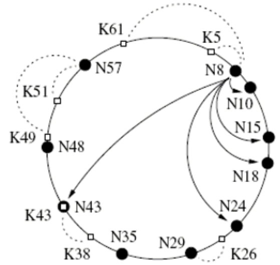
Finger Table of N8		
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n + 2^{i-1}) mod 2^m)* → routing in log(N) steps possible



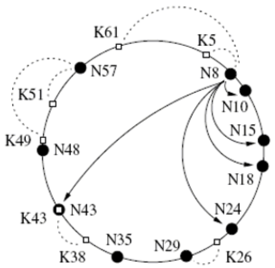
Finger Table of N8		
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n + 2^{i-1}) mod 2^m)* → routing in log(N) steps possible



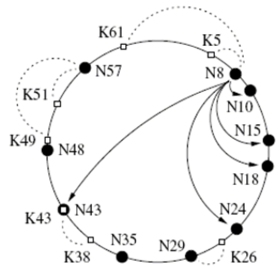
Finger Table of N8		
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor((n + 2^{i-1}) mod 2^m)* → routing in log(N) steps possible



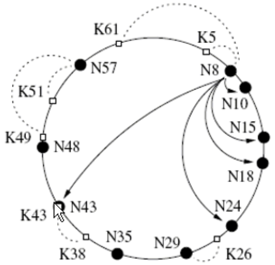
Finger Table of N8		
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8.

[8]

Homophily and Distance

- Example for this principle: Early P2P file sharing protocols based on Distributed Hash Tables: Example: Chord:
  - data (e.g. filenames) („keys“) and host-IDs (e.g. IP-addresses) („nodes“) hashed into the same m-dim key-space,
  - Key *k* is assigned to node *successor(k)*,
  - each node n maintains finger table (length up to m): m-th entry: IP-address of node *successor*$((n + 2^{i-1}) \bmod 2^m)$ → routing in log(N) steps possible



Finger Table of N8		
Idx	Target ID	Successor
0	N8 + 1	N10
1	N8 + 2	N10
2	N8 + 4	N15
3	N8 + 8	N18
4	N8 + 16	N24
5	N8 + 32	N43

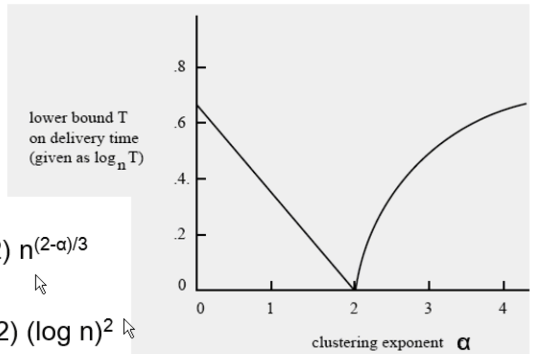
Fig. 8.1: A 6-bit Chord identifier space. Dotted lines indicate which nodes host which keys. Black lines represent the fingers of node *N*8. [8]

Kleinberg Model

- local (decentralized) knowledge:
  - Each node only knows only:
    - Its adjacent nodes
    - The grid's principle structure
    - Position of target node on the grid
    - Positions and long-range contacts of nodes on the message path so far
- (Search-) algorithm with only local knowledge: „decentralized“

Kleinberg Model

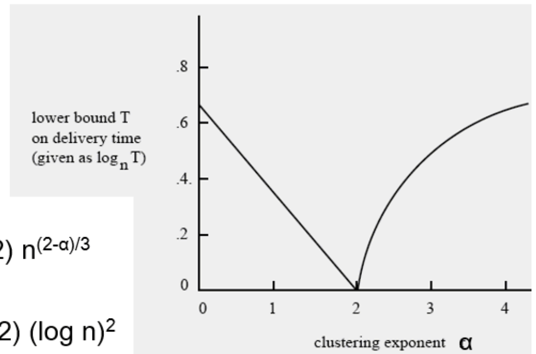
- Now: Send message with local (decentralized) knowledge only
- Given: Decentralized greedy message delivery algorithm: measure number of expected delivery steps s:
  - $0 \leq \alpha < 2$: s at least $\sim c1(\alpha,q1,q2)\, n^{(2-\alpha)/3}$
  - $\alpha = 2$: s at most $\sim c2(\alpha,q1,q2)\, (\log n)^2$
  - $\alpha > 2$ s at least $\sim c3(\alpha,q1,q2)\, n^{(\alpha-2)/(\alpha-1)}$



(closely after [3])

Kleinberg Model

- Now: Send message with local (decentralized) knowledge only
- Given: Decentralized greedy message delivery algorithm: measure number of expected delivery steps s:
  - $0 \leq \alpha < 2$: s at least $\sim c1(\alpha,q1,q2)\, n^{(2-\alpha)/3}$
  - $\alpha = 2$: s at most $\sim c2(\alpha,q1,q2)\, (\log n)^2$
  - $\alpha > 2$: s at least $\sim c3(\alpha,q1,q2)\, n^{(\alpha-2)/(\alpha-1)}$



(closely after [3])

Kleinberg Model

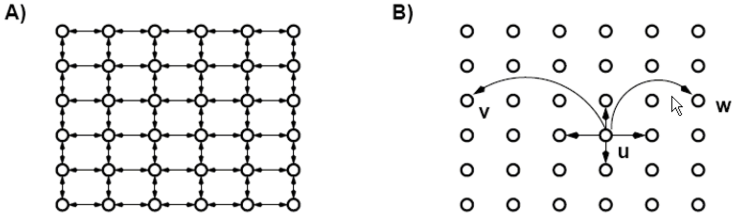
- Put nodes on a n x n grid. Distance: Manhattan:

$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \le q1$ (regular local contacts)
- Each node i: Additional q2 other „long range“ edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

examples:



q1 = 1, q2 = 0   edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

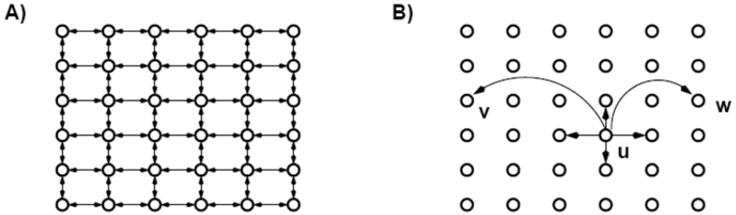
- Put nodes on a n x n grid. Distance: Manhattan:

$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \le q1$ (regular local contacts)
- Each node i: Additional q2 other „long range“ edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

examples:



q1 = 1, q2 = 0   edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

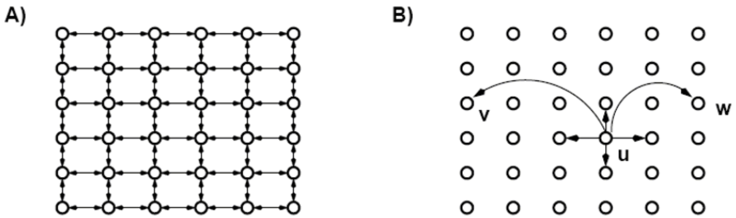
- Put nodes on a n x n grid. Distance: Manhattan:

$$r(i,j) = |x_i - x_j| + |y_i - y_j|$$

- Each node i: Connected to all nodes with $r(i,j) \le q1$ (regular local contacts)
- Each node i: Additional q2 other „long range“ edges:
  Probability of edge to node j:

$$P(j) \sim r(i,j)^{-\alpha}$$

examples:



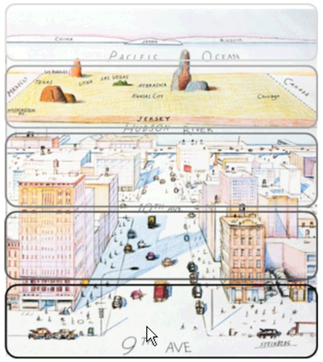
q1 = 1, q2 = 0   edges for one node u: (q1 = 1, q2 = 2)

[3]

Kleinberg Model

- Effectively: Only a notion of distance (not neccess. spatial!) is necessary to route!
- Applications in P2P Systems (see [2])

- Explanation for the dependency of α=D on the grid‘s dimension D for efficient delivery:
  - start from node u
  - partition the set of other nodes into sets $A_0, A_1, A_2, \ldots, A_{\log n}$, where A_i has a distance to u between 2^i and 2^{i+1}
  - proven in [3]: only α=D ensures that the q2 „long-range“ contacts are evenly distributed over the A_i
  - for α>D : bias towards smaller distances; for α<D : bias towards larger distances

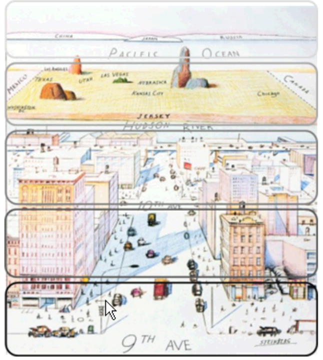


[5,6]

Kleinberg Model

- Effectively: Only a notion of distance (not neccess. spatial!) is necessary to route!
- Applications in P2P Systems (see [2])

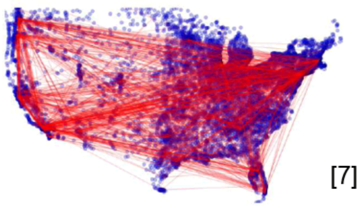
- Explanation for the dependency of α=D on the grid‘s dimension D for efficient delivery:
  - start from node u
  - partition the set of other nodes into sets $A_0, A_1, A_2, \ldots, A_{\log n}$, where A_i has a distance to u between 2^i and 2^{i+1}
  - proven in [3]: only α=D ensures that the q2 „long-range“ contacts are evenly distributed over the A_i
  - for α>D : bias towards smaller distances; for α<D : bias towards larger distances



[5,6]

Geographic Distance as Routing Metric

- In analysis of Milgram‘s experiment:
  - People early in the resp. path: often cite geographic proximity as main forwarding criterion;
  - People late in the path: chose similarity of occupation
- [6]: Study „Kleinberg-like“ distributions / effects on real social NW:
  - LifeJournal.com : locate ~ 10^6 users (long/lat of their hometown)
  - Simulate Milgram on friendship NW with greedy decentralized forwarding and geographical proximity as criterion
  - Result: efficient routing is possible on average (see [6])



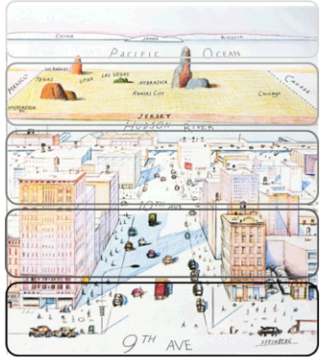
[7]

Fig. 2 The LiveJournal social network [32]. A dot is shown for each geographic location that was declared as the hometown of at least one of the ≈500,000 LiveJournal users whom we were able to locate at a longitude and latitude in the continental United States. A random 0.1% of the friendships in the network are overlaid on these locations.

Kleinberg Model

- Effectively: Only a notion of distance (not neccess. spatial!) is necessary to route!
- Applications in P2P Systems (see [2])

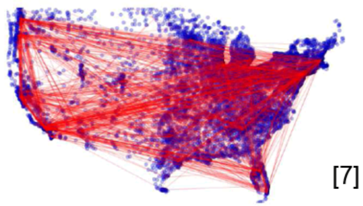
- Explanation for the dependency of α=D on the grid‘s dimension D for efficient delivery:
  - start from node u
  - partition the set of other nodes into sets $A_0, A_1, A_2, \ldots, A_{\log n}$, where A_i has a distance to u between 2^i and 2^{i+1}
  - proven in [3]: only α=D ensures that the q2 „long-range“ contacts are evenly distributed over the A_i
  - for α>D : bias towards smaller distances; for α<D : bias towards larger distances



[5,6]

Geographic Distance as Routing Metric

- In analysis of Milgram‘s experiment:
  - People early in the resp. path: often cite geographic proximity as main forwarding criterion;
  - People late in the path: chose similarity of occupation
- [6]: Study „Kleinberg-like“ distributions / effects on real social NW:
  - LifeJournal.com : locate ~ 10^6 users (long/lat of their hometown)
  - Simulate Milgram on friendship NW with greedy decentralized forwarding and geographical proximity as criterion
  - Result: efficient routing is possible on average (see [6])

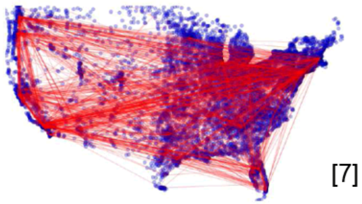


[7]

Fig. 2 The LiveJournal social network [32]. A dot is shown for each geographic location that was declared as the hometown of at least one of the ≈500,000 LiveJournal users whom we were able to locate at a longitude and latitude in the continental United States. A random 0.1% of the friendships in the network are overlaid on these locations.

Geographic Distance as Routing Metric

- In analysis of Milgram's experiment:
  - People early in the resp. path: often cite geographic proximity as main forwarding criterion;
  - People late in the path: chose similarity of occupation
- [6]: Study „Kleinberg-like" distributions / effects on real social NW:
  - LifeJournal.com : locate ~ 10^6 users (long/lat of their hometown)
  - Simulate Milgram on friendship NW with greedy decentralized forwarding and geographical proximity as criterion
  - Result: efficient routing is possible on average (see [6])

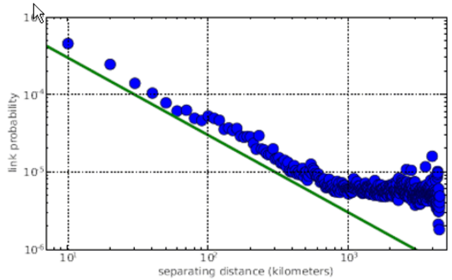


[7]

Fig. 2 The LiveJournal social network [32]. A dot is shown for each geographic location that was declared as the hometown of at least one of the ≈500,000 LiveJournal users whom we were able to locate at a longitude and latitude in the continental United States. A random 0.1% of the friendships in the network are overlaid on these locations.

Geographic Distance as Routing Metric

- But: Investigate Kleinberg's $\alpha \overset{!}{=} 2$ claim: result: probability of friendship as a function of distance reveals $\alpha \approx 1$



$$P(d) = \varepsilon + \Theta(1/d)$$

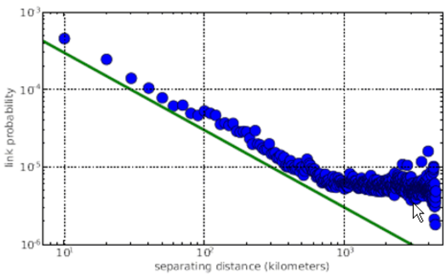
[7]

Fig. 3 The probability $P(d)$ of a friendship between two people in LiveJournal as a function of the geographic distance d between their declared hometowns [32]. Distances are rounded into 10-kilometer buckets. The solid line corresponds to $P(d) \propto 1/d$. Note that Theorem 1 requires $P(d) \propto 1/d^2$ for a network of people arranged in a regular 2-dimensional grid to be navigable.

- Although $\alpha \approx 1$ was found, systematic power law is surprising in "pure virtual" community as LiveJournal

Geographic Distance as Routing Metric

- But: Investigate Kleinberg's $\alpha \overset{!}{=} 2$ claim: result: probability of friendship as a function of distance reveals $\alpha \approx 1$



$$P(d) = \varepsilon + \Theta(1/d)$$

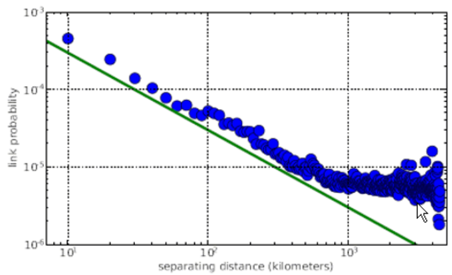
[7]

Fig. 3 The probability $P(d)$ of a friendship between two people in LiveJournal as a function of the geographic distance d between their declared hometowns [32]. Distances are rounded into 10-kilometer buckets. The solid line corresponds to $P(d) \propto 1/d$. Note that Theorem 1 requires $P(d) \propto 1/d^2$ for a network of people arranged in a regular 2-dimensional grid to be navigable.

- Although $\alpha \approx 1$ was found, systematic power law is surprising in "pure virtual" community as LiveJournal

Geographic Distance as Routing Metric

- But: Investigate Kleinberg's $\alpha \overset{!}{=} 2$ claim: result: probability of friendship as a function of distance reveals $\alpha \approx 1$



$$P(d) = \varepsilon + \Theta(1/d)$$

[7]

Fig. 3 The probability $P(d)$ of a friendship between two people in LiveJournal as a function of the geographic distance d between their declared hometowns [32]. Distances are rounded into 10-kilometer buckets. The solid line corresponds to $P(d) \propto 1/d$. Note that Theorem 1 requires $P(d) \propto 1/d^2$ for a network of people arranged in a regular 2-dimensional grid to be navigable.

- Although $\alpha \approx 1$ was found, systematic power law is surprising in "pure virtual" community as LiveJournal

Geographic Distance as Routing Metric

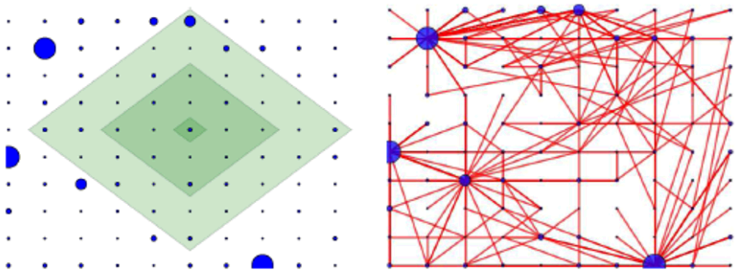
- Other studies: also confirm efficient decentralized routing with geographic proximity as criterion is possible AND $\alpha \approx 1$
- Explanation for this "contradiction": geographic density of people is not uniform (cp. urban vs. mid west) as assumed in the Kleinberg grid.
- Model: Rank-based friendship: modify Kleinberg (D=k):
  - at each grid point more than one person
  - p(v) for long range contact v starting at node u: $p_u(v) \sim 1 / rank_u(v)$ where rank(u,v) = number of people who live at least as close to u as v does. (→ u sorts candidates v according to distance)
  - (if population density is **uniform**: $rank_u(v) \sim \theta(d(u,v)^k)$ → Kleinberg's claim is fulfilled)

Geographic Distance as Routing Metric

- Other studies: also confirm efficient decentralized routing with geographic proximity as criterion is possible AND $\alpha \approx 1$
- Explanation for this "contradiction": geographic density of people is not uniform (cp. urban vs. mid west) as assumed in the Kleinberg grid.
- Model: Rank-based friendship: modify Kleinberg (D=k):
  - at each grid point more than one person
  - p(v) for long range contact v starting at node u: $p_u(v) \sim 1 / rank_u(v)$ where rank(u,v) = number of people who live at least as close to u as v does. (→ u sorts candidates v according to distance)
  - (if population density is **uniform**: $rank_u(v) \sim \theta(d(u,v)^k)$ → Kleinberg's claim is fulfilled)

Geographic Distance as Routing Metric

- Finding: Even for non-uniform geographic distribution of people: efficient routing possible (choose pairs randomly):
  - For q1= 2k and q2 = 1 : $T \sim O((\log n)^3)$



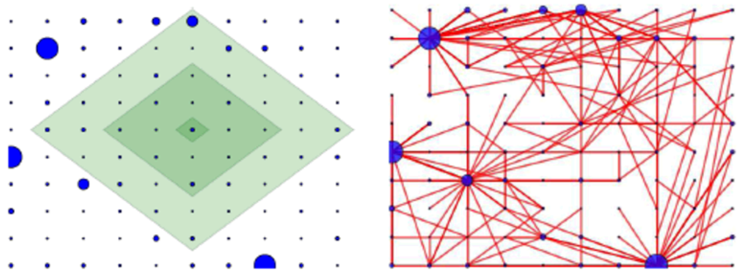
(a) Concentric balls around a city C, where each ball's population increases by a factor of four. A resident of C choosing a rank-based friend is four times more likely to choose a friend at the boundary of one ball than a friend at the boundary of the next-larger ball.

(b) A rank-based social network generated from this population distribution. For visual simplicity, edges are depicted as connecting cities; the complete image would show each edge connecting one resident from each of its endpoint cities.

[7]

Geographic Distance as Routing Metric

- Finding: Even for non-uniform geographic distribution of people: efficient routing possible (choose pairs randomly):
  - For q1= 2k and q2 = 1 : $T \sim O((\log n)^3)$



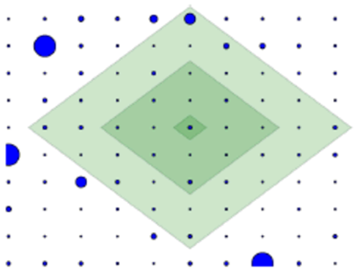
(a) Concentric balls around a city C, where each ball's population increases by a factor of four. A resident of C choosing a rank-based friend is four times more likely to choose a friend at the boundary of one ball than a friend at the boundary of the next-larger ball.

(b) A rank-based social network generated from this population distribution. For visual simplicity, edges are depicted as connecting cities; the complete image would show each edge connecting one resident from each of its endpoint cities.

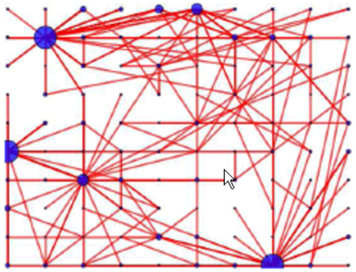
[7]

Geographic Distance as Routing Metric

- Finding: Even for non-uniform geographic distribution of people: efficient routing possible (choose pairs randomly):
  - For q1= 2k and q2 = 1 : T~O((log n)^3)



(a) Concentric balls around a city C, where each ball's population increases by a factor of four. A resident of C choosing a rank-based friend is four times more likely to choose a friend at the boundary of one ball than a friend at the boundary of the next-larger ball.



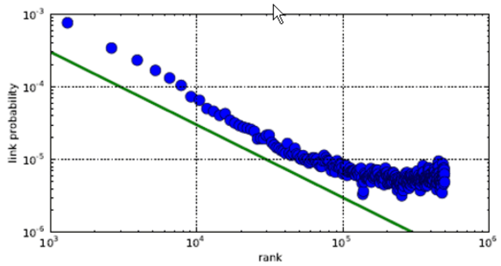
(b) A rank-based social network generated from this population distribution. For visual simplicity, edges are depicted as connecting cities; the complete image would show each edge connecting one resident from each of its endpoint cities.

[7]

Geographic Distance as Routing Metric

- Rank based evaluation of LifeJournal:

$$P(r) = \Theta(1/r) + \varepsilon$$



[7]

Fig. 5 The probability $P(r)$ of a friendship between two people u and v in LiveJournal as a function of the rank of v with respect to u [32]. Ranks are rounded into buckets of size 1300, which is the LiveJournal population of the city for a randomly chosen person in the network, and thus 1300 is in a sense the "rank resolution" of the dataset. (The unaveraged data are noisier, but follow the same trend.) The solid line corresponds to $P(r) \propto 1/r$. Note that Theorem 2 requires $P(r) \propto 1/r$ for a rank-based network to be navigable.

Off-Grid / Other Metrics

- Grid and concentration on geo-proximity alone: not realistic.
- Example: occupation: given taxonomy of occupations: similarity measure: determine tree height of least common ancestor (lca) of u and v
- Kleinberg: if tree is a regular b-ary tree and long range probability goes as $\Pr[u \to v] \propto b^{-\beta \cdot \mathrm{lca}(u,v)}$ then efficient routing only for β=1
- Generalization: tree → graph : Pr[u→v] ~ 1 / f (path-distance(u,v))

Off-Grid / Other Metrics

- Grid and concentration on geo-proximity alone: not realistic.
- Example: occupation: given taxonomy of occupations: similarity measure: determine tree height of least common ancestor (lca) of u and v
- Kleinberg: if tree is a regular b-ary tree and long range probability goes as $\Pr[u \to v] \propto b^{-\beta \cdot \mathrm{lca}(u,v)}$ then efficient routing only for β=1
- Generalization: tree → graph : Pr[u→v] ~ 1 / f (path-distance(u,v))

Off-Grid / Other Metrics

- Grid and concentration on geo-proximity alone: not realistic.
- Example: occupation: given taxonomy of occupations: similarity measure: determine tree height of least common ancestor (lca) of u and v
- Kleinberg: if tree is a regular b-ary tree and long range probability goes as $\Pr[u \rightarrow v] \propto b^{-\beta \cdot \mathrm{lca}(u,v)}$ then efficient routing only for β=1
- Generalization: tree → graph : Pr[u→v] ~ 1 / f (path-distance(u,v))

Off-Grid / Other Metrics

- Grid and concentration on geo-proximity alone: not realistic.
- Example: occupation: given taxonomy of occupations: similarity measure: determine tree height of least common ancestor (lca) of u and v
- Kleinberg: if tree is a regular b-ary tree and long range probability goes as $\Pr[u \rightarrow v] \propto b^{-\beta \cdot \mathrm{lca}(u,v)}$ then efficient routing only for β=1
- Generalization: tree → graph : Pr[u→v] ~ 1 / f (path-distance(u,v))

Off-Grid / Other Metrics

- Kleinberg: Group Model: Partition set of actors into groups: Pr[u→v] ~ [1 / size of smallest common group of u and v] → navigateable with decentralized greedy approach if:
  - Groups form a hierarchy (small contained in large) → „narrowing in“ possible
  - Group-sizes satisfy certain bounds → non-zero „escaping“ probability from a small group
- Advantage: Groups can be formed according to different criteria simultaneously
- Strictly spoken: Even in the 2-dim grid: we have a combination of two elements (long and lat)