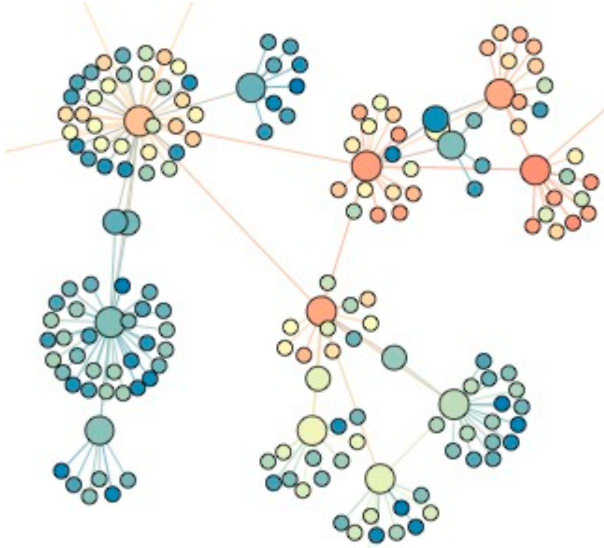


Topic mash II: assortativity, resilience, link prediction

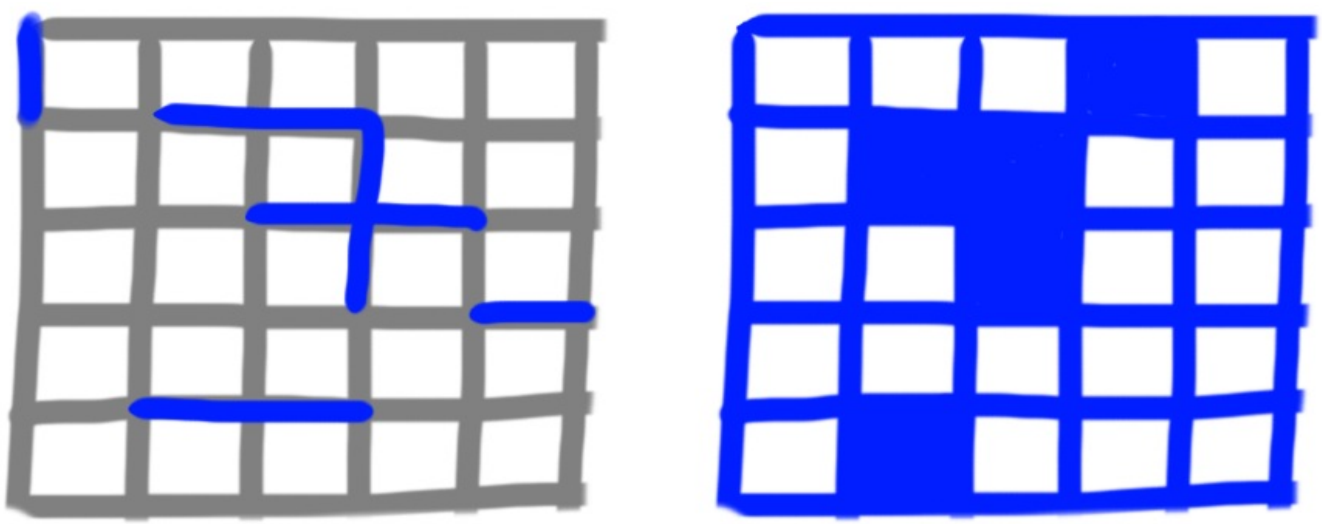
CS224W

Outline

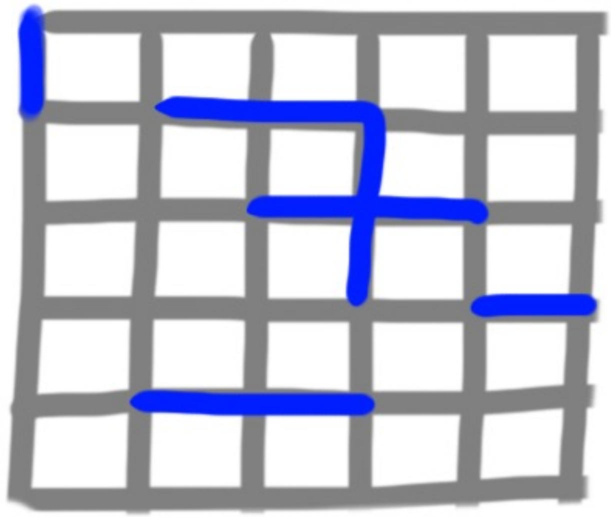
- Node vs. edge percolation
- Resilience of randomly vs. preferentially grown networks
- Resilience in real-world networks

network resilience

- Q: If a given fraction of nodes or edges are removed…
  - how large are the connected components?
  - what is the average distance between nodes in the components
- Related to percolation (previously studied on lattices):

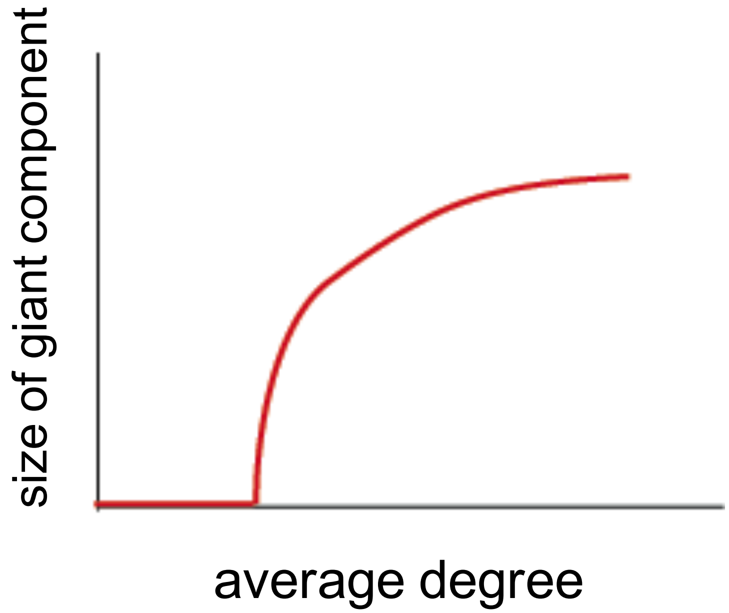


edge percolation

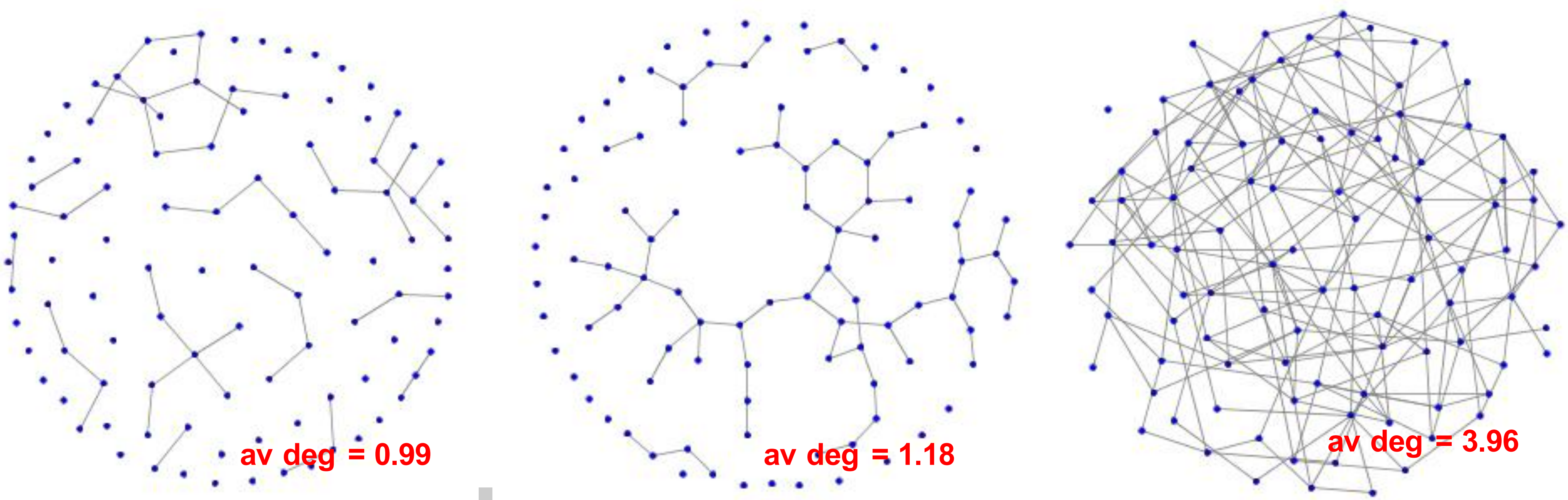


- Edge removal
  - bond percolation: each edge is removed with probability (1-p)
    - corresponds to random failure of links
  - targeted attack: causing the most damage to the network with the removal of the fewest edges
    - strategies: remove edges that are most likely to break apart the network or lengthen the average shortest path
    - e.g. usually edges with high betweenness

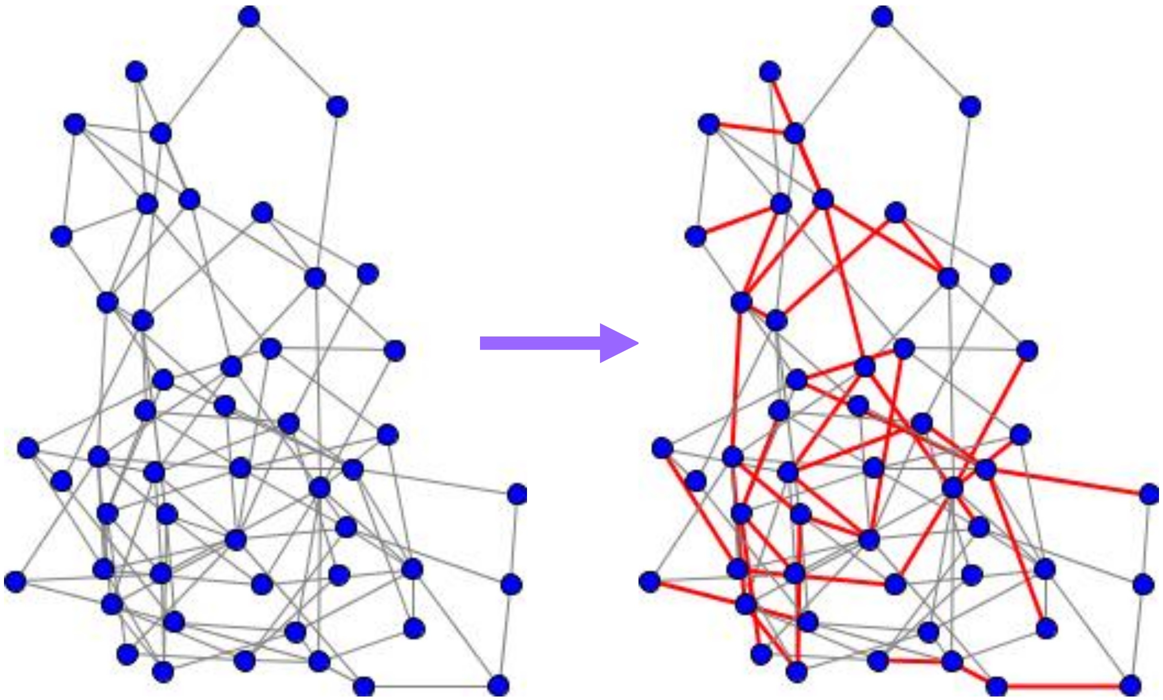
reminder: percolation in ER graphs



- As the average degree increases to z = 1, a giant component suddenly appears
- Edge removal is the opposite process – at some point the average degree drops below 1 and the network becomes disconnected

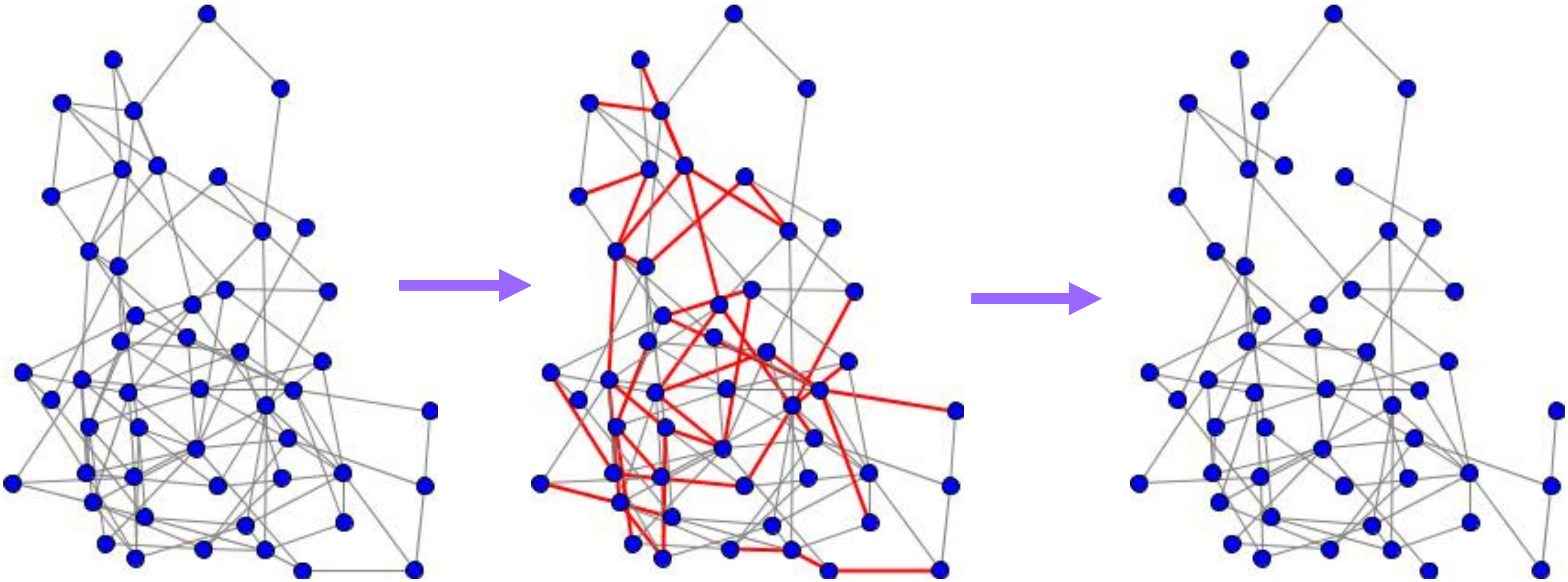


Quiz Q:



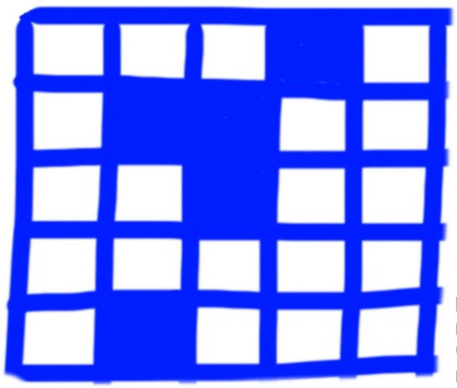
In this network each node has average degree 4.64, if you removed 25% of the edges, by how much would you reduce the giant component?

edge percolation

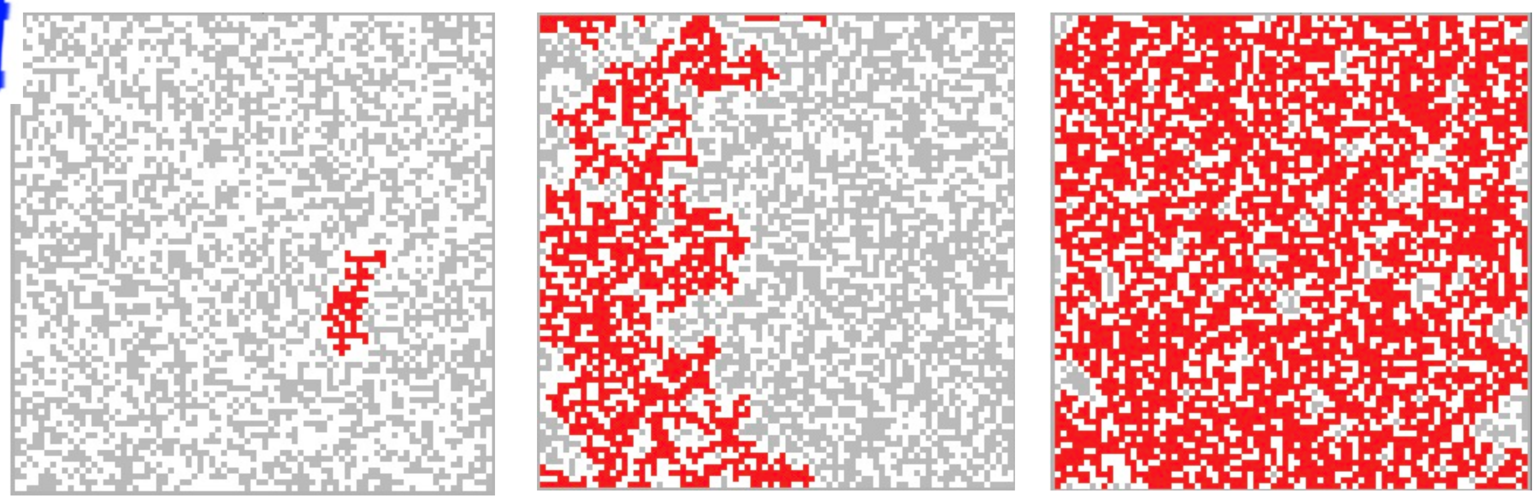


50 nodes, 116 edges, average degree 4.64
after 25 % edge removal
76 edges, average degree 3.04 – still well above percolation threshold

node removal and site percolation



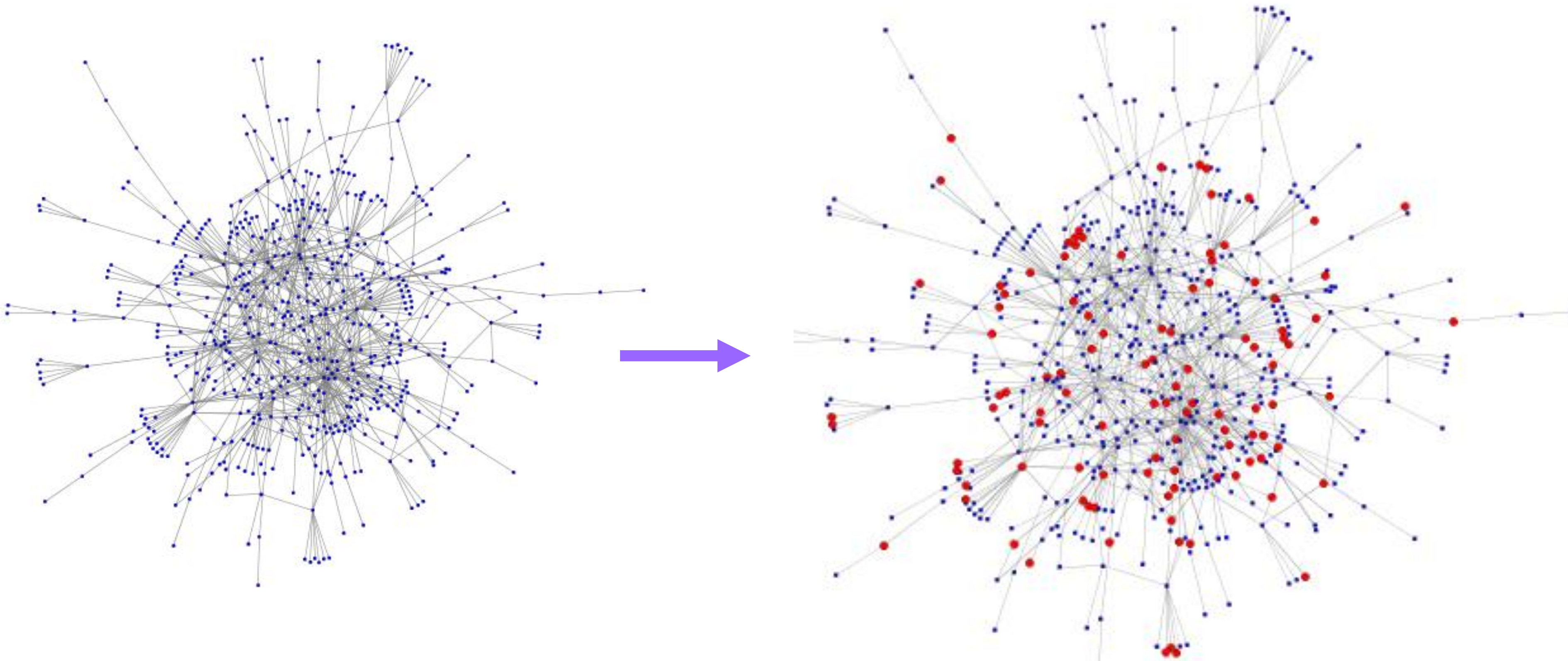
Ordinary Site Percolation on Lattices:
Fill in each site (site percolation) with probability p



- **low p**: small islands
- **p critical**: giant component forms, occupying finite fraction of infinite lattice.
  p above critical value: giant component occupies an increasingly larger portion of the graph

http://web.stanford.edu/class/cs224w/NetLogo/LatticePercolation.nlogo

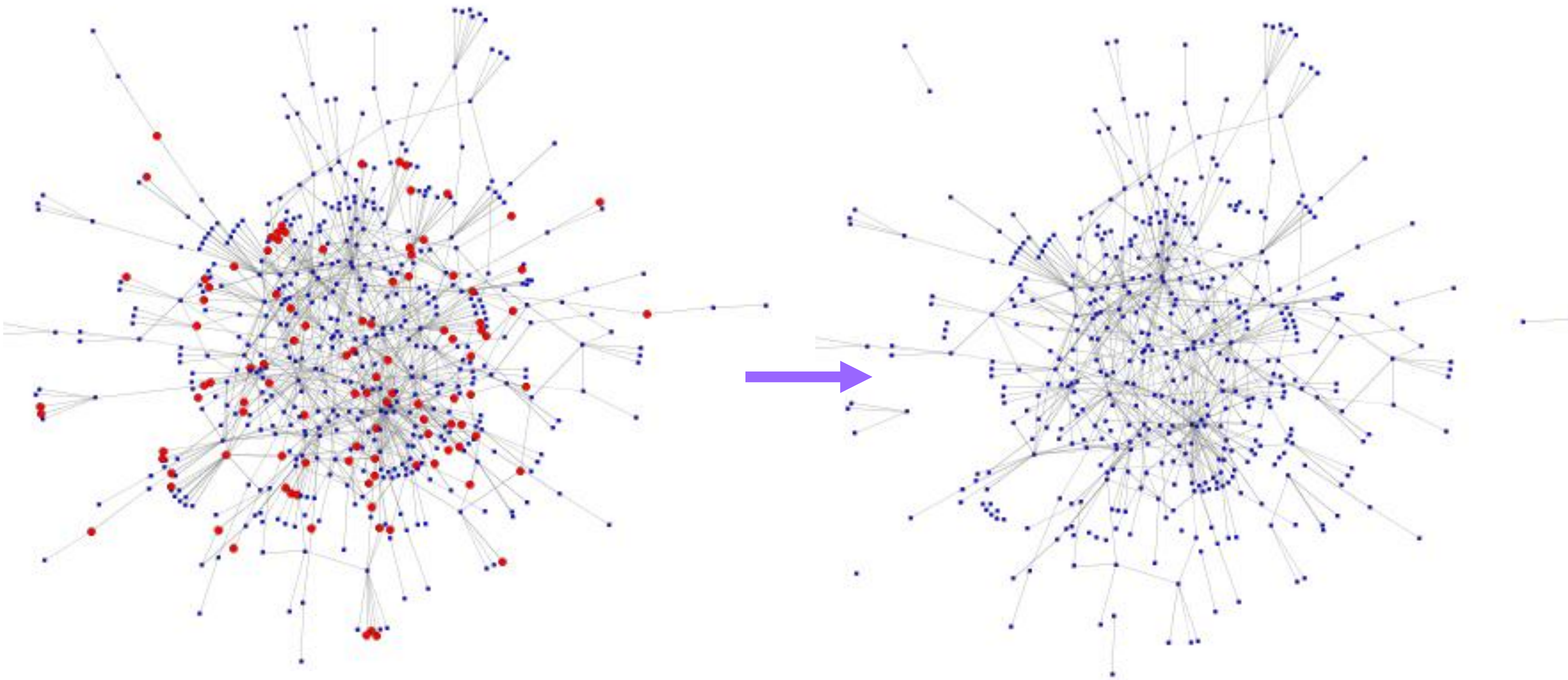
Percolation on networks



- Percolation can be extended to networks of arbitrary topology.
- We say the network percolates when a giant component forms.

Random attack on scale-free networks

- Example: gnutella filesharing network, 20% of nodes removed at random

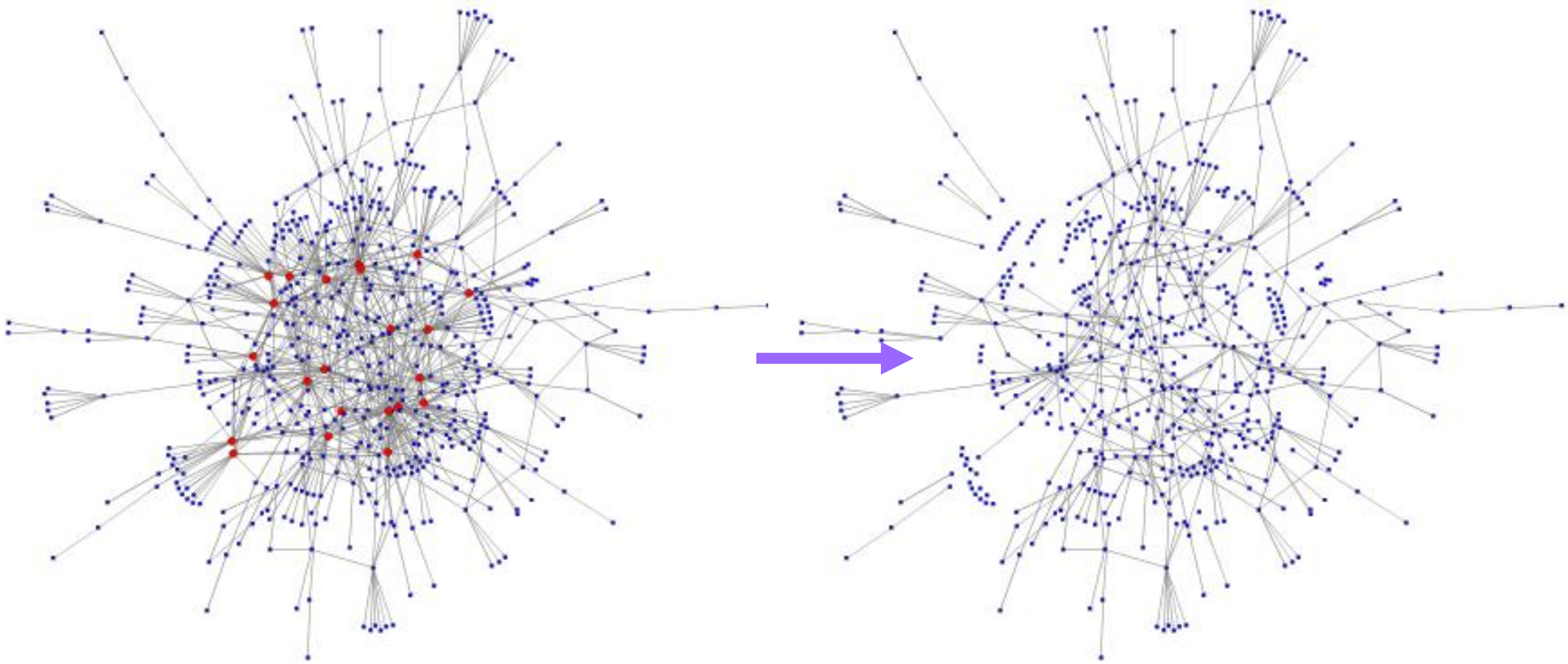


574 nodes in giant component

427 nodes in giant component

Targeted attacks on power-law networks

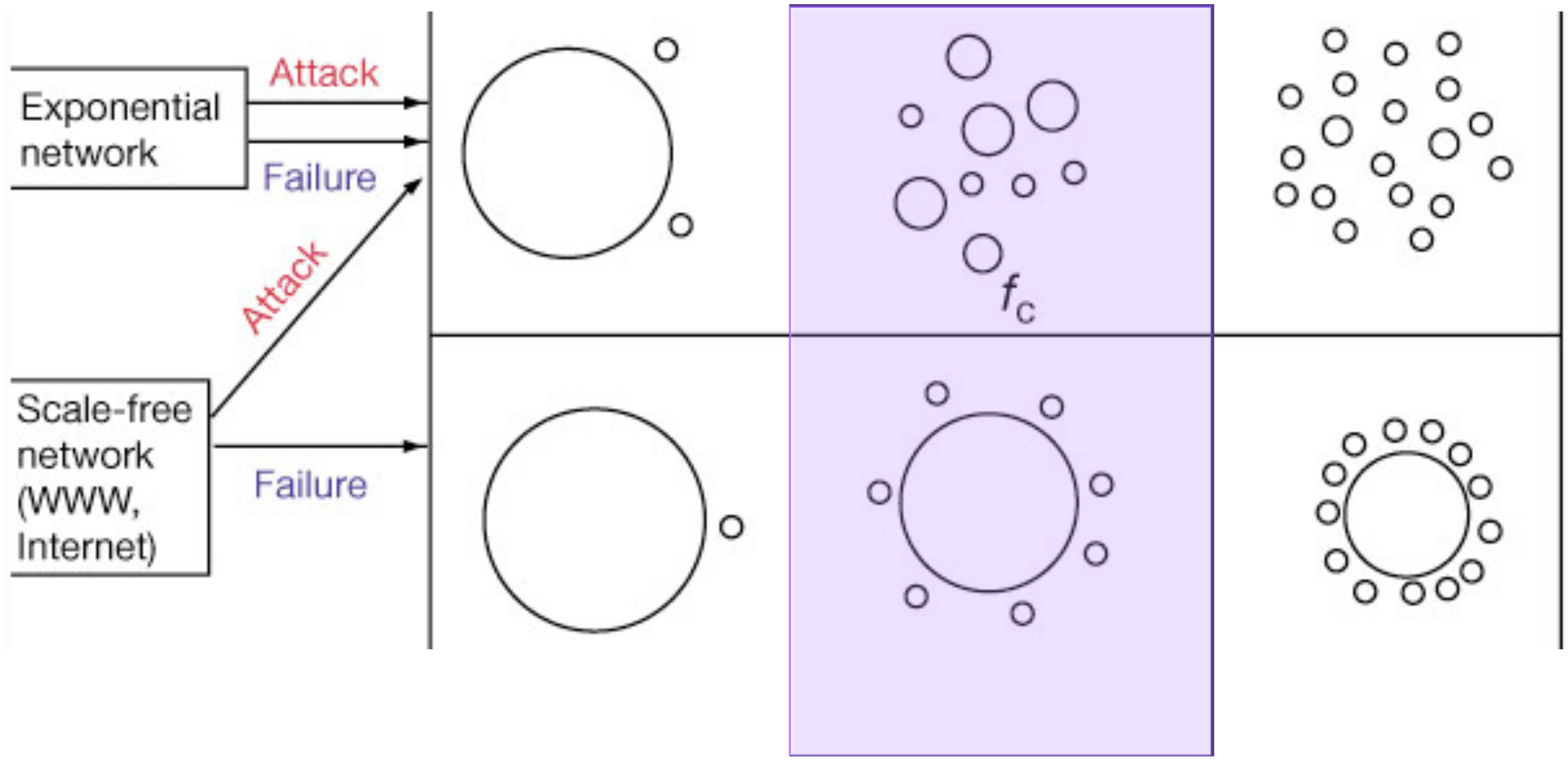
- Power-law networks are vulnerable to targeted attack
- Example: same gnutella network, 22 most connected nodes removed (2.8% of the nodes)



574 nodes in giant component

301 nodes in giant component

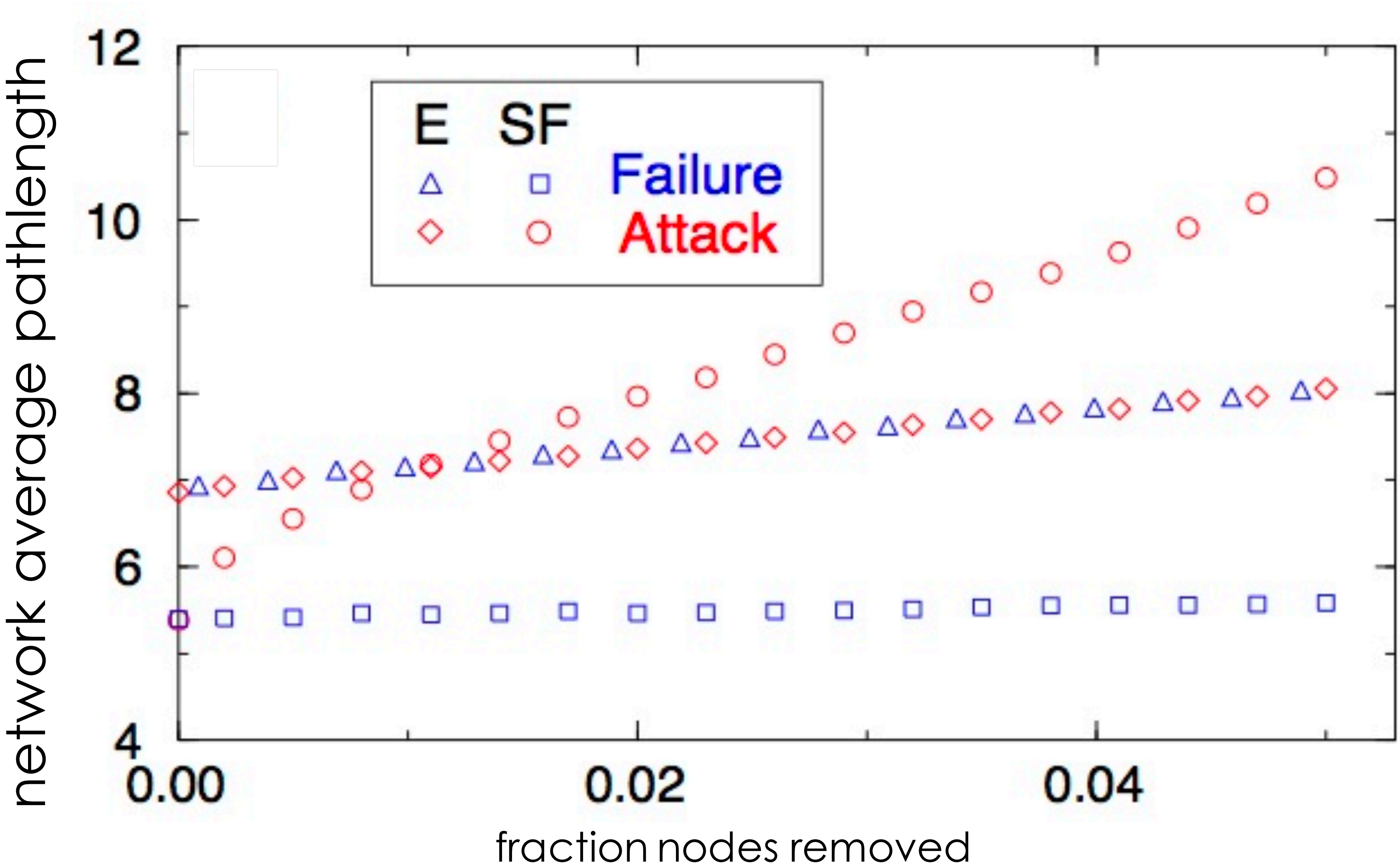
random failures vs. attacks



why the difference here for attack vs. failure?

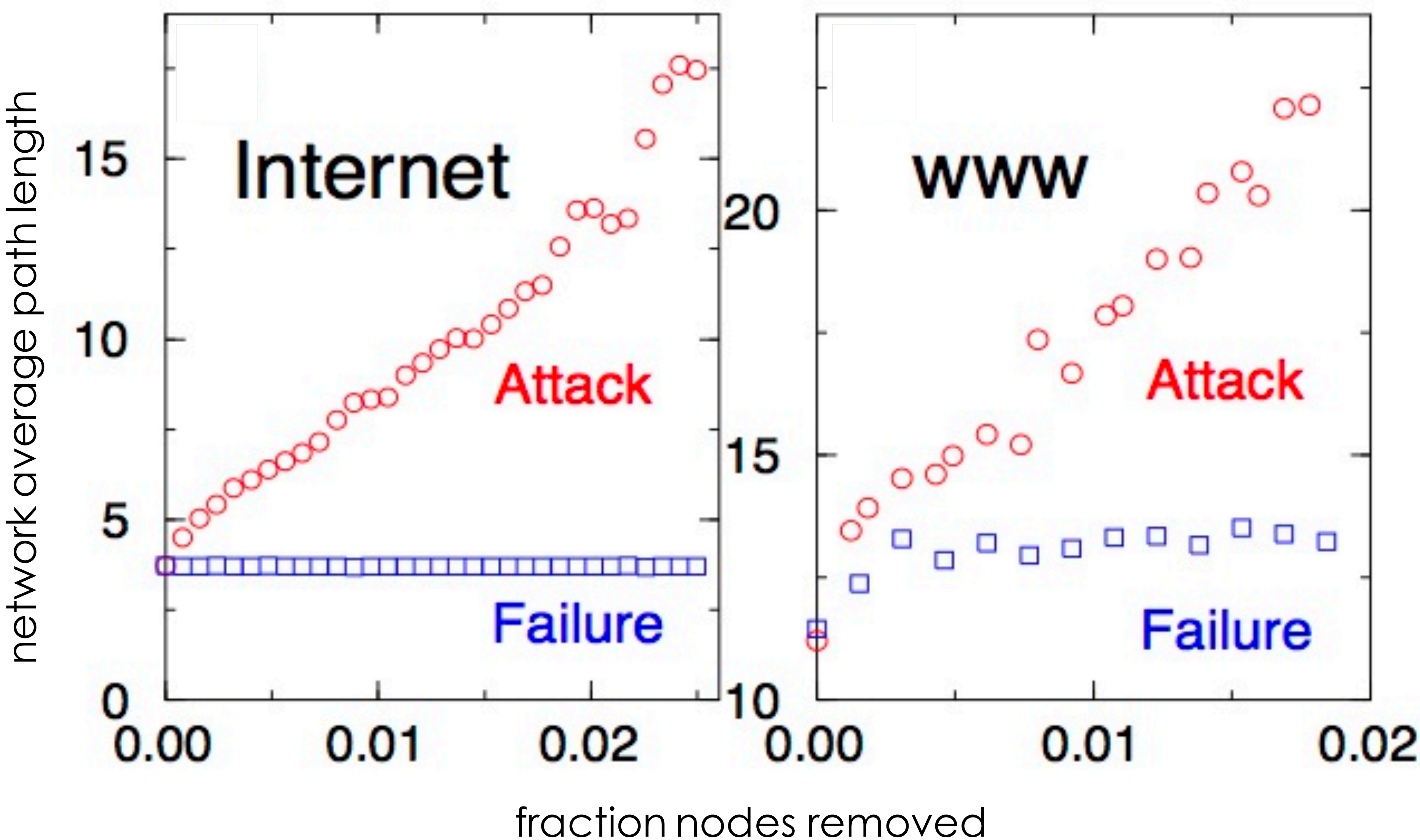
Source: Error and attack tolerance of complex networks. Réka Albert, Hawoong Jeong and Albert-László Barabási. Nature 406, 378-382(27 July 2000); http://www.nature.com/nature/journal/v406/n6794/abs/406378A0.html

effect on path length



Source: Error and attack tolerance of complex networks. Réka Albert, Hawoong Jeong and Albert-László Barabási. Nature 406, 378-382(27 July 2000); http://www.nature.com/nature/journal/v406/n6794/abs/406378A0.html

applied to empirical networks

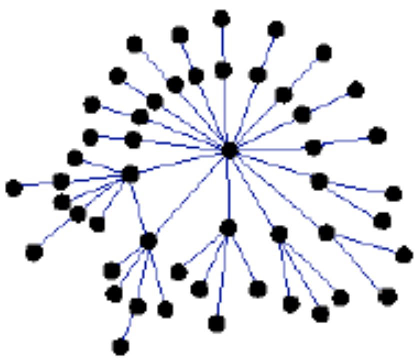


Source: Error and attack tolerance of complex networks. Réka Albert, Hawoong Jeong and Albert-László Barabási. Nature 406, 378-382(27 July 2000); http://www.nature.com/nature/journal/v406/n6794/abs/406378A0.html

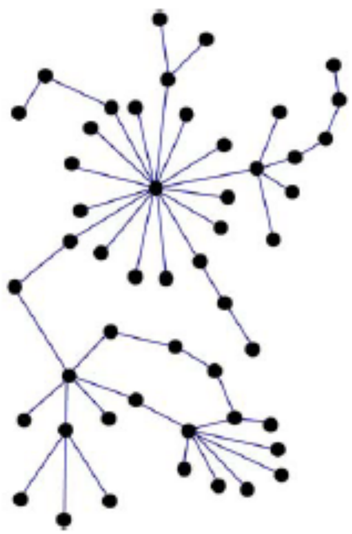
Assortativity

- Social networks are assortative:
  - the gregarious people associate with other gregarious people
  - the loners associate with other loners

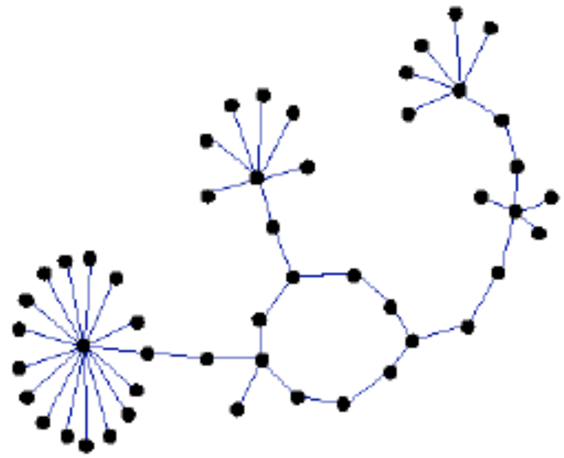
- The Internet is disassortative:



Assortative:
hubs connect to hubs



Random

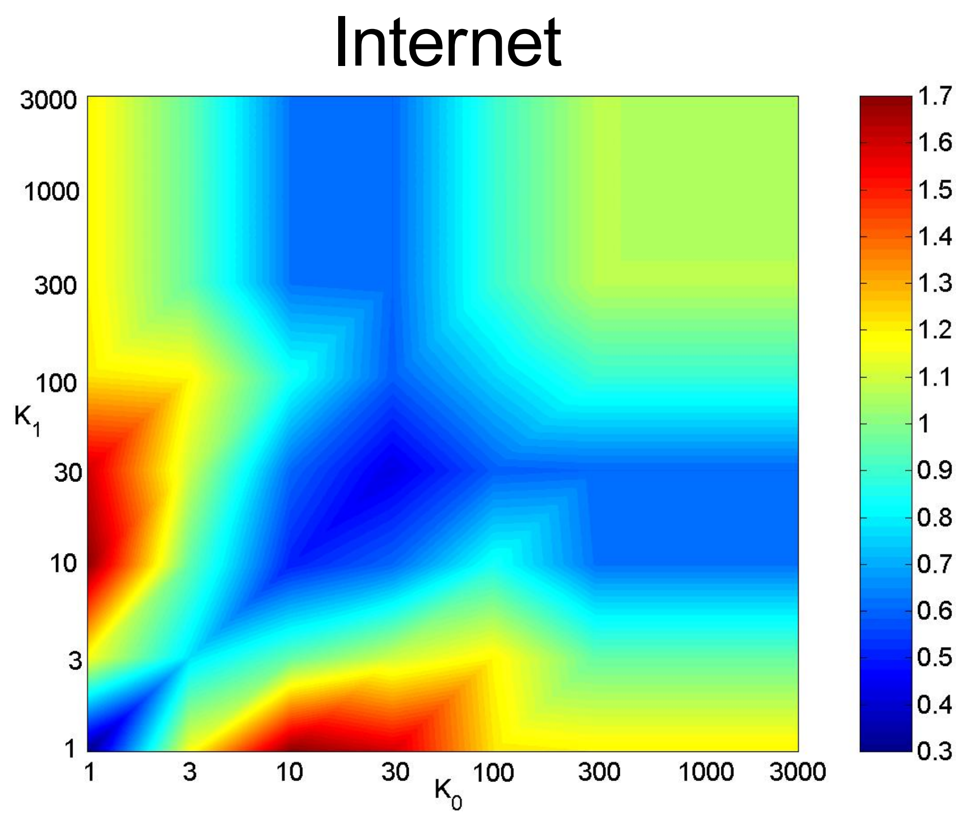


Disassortative:
hubs are in the periphery

Correlation profile of a network

- Detects preferences in linking of nodes to each other based on their connectivity
- Measure $N(k_0,k_1)$ – the number of edges between nodes with connectivities k_0 and k_1
- Compare it to $N_r(k_0,k_1)$ – the same property in a properly randomized network

Degree correlation profiles: 2D

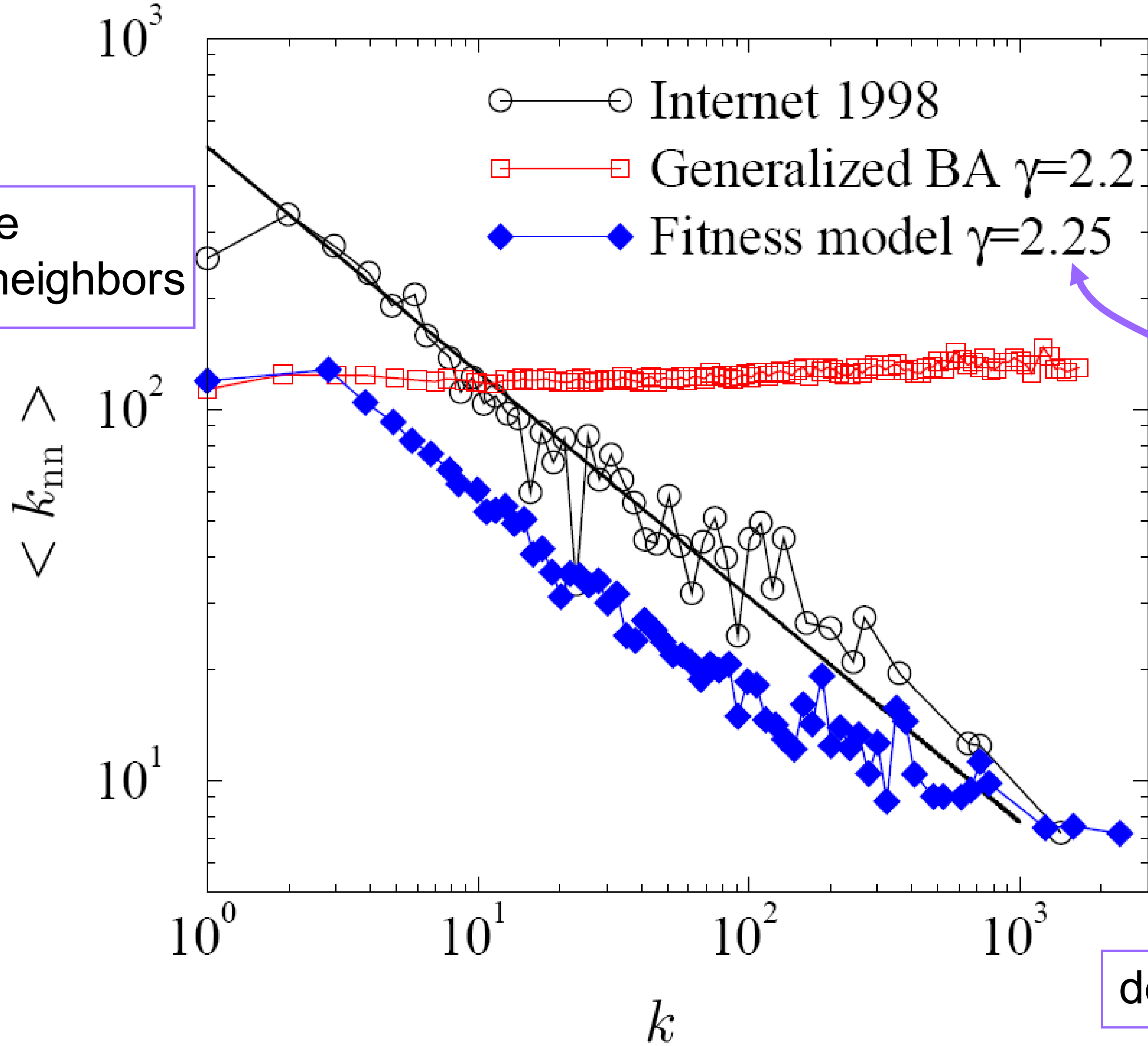


source: Sergei Maslov

Average degree of neighbors

- Pastor-Satorras and Vespignani: 2D plot

average degree
of the node's neighbors



probability of aquiring edges is dependent on 'fitness' + degree Bianconi & Barabasi

degree of node

Single number

- cor(deg(i),deg(j)) over all edges {ij}

$$\rho_{internet} = -0.189$$

The Pearson correlation coefficient of nodes on each side on an edge

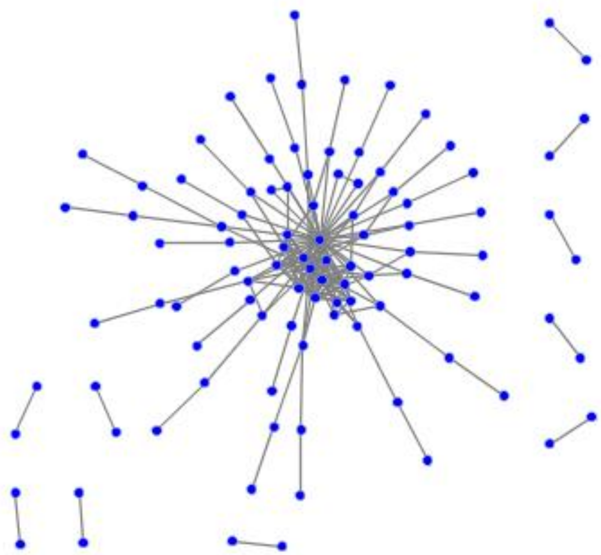
assortative mixing more generally

- Assortativity is not limited to degree-degree correlations other attributes
  - social networks: race, income, gender, age
  - food webs: herbivores, carnivores
  - internet: high level connectivity providers, ISPs, consumers

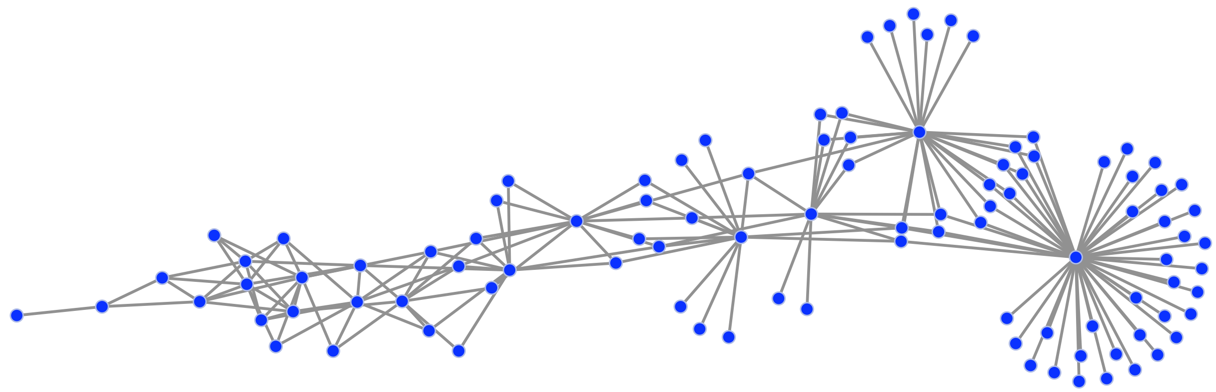
- Tendency of like individuals to associate = ‘homophily’

Quiz Q:

will a network with positive or negative degree assortativity be more resilient to attack?



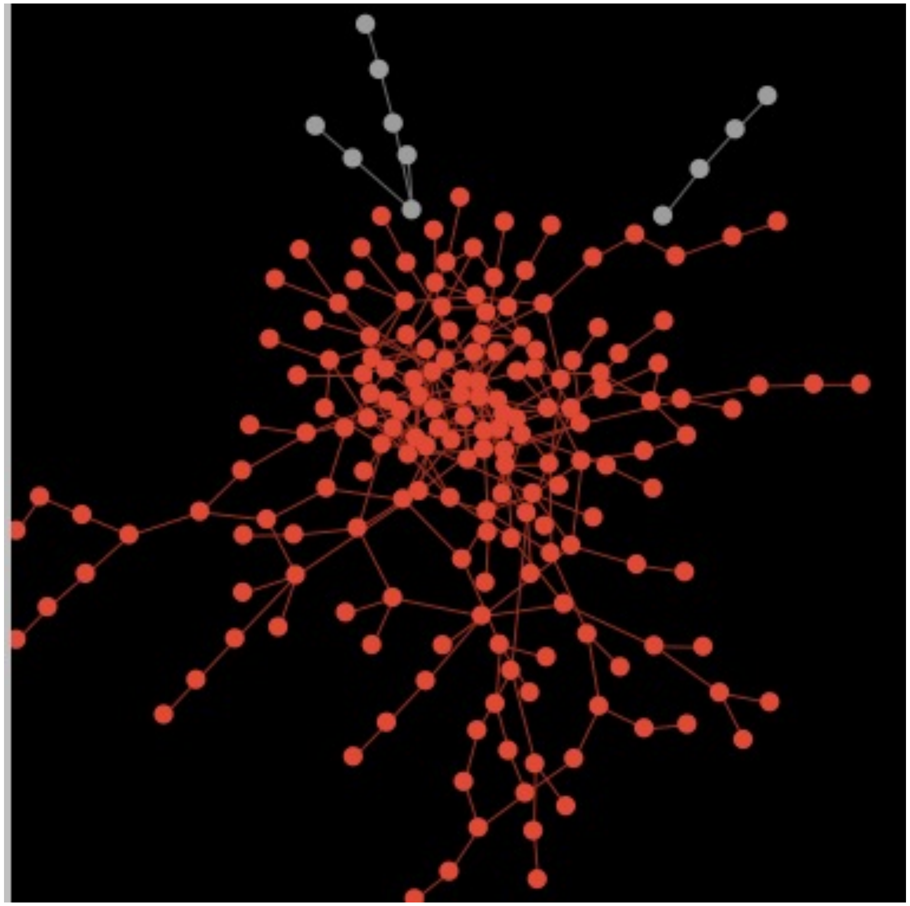
assortative



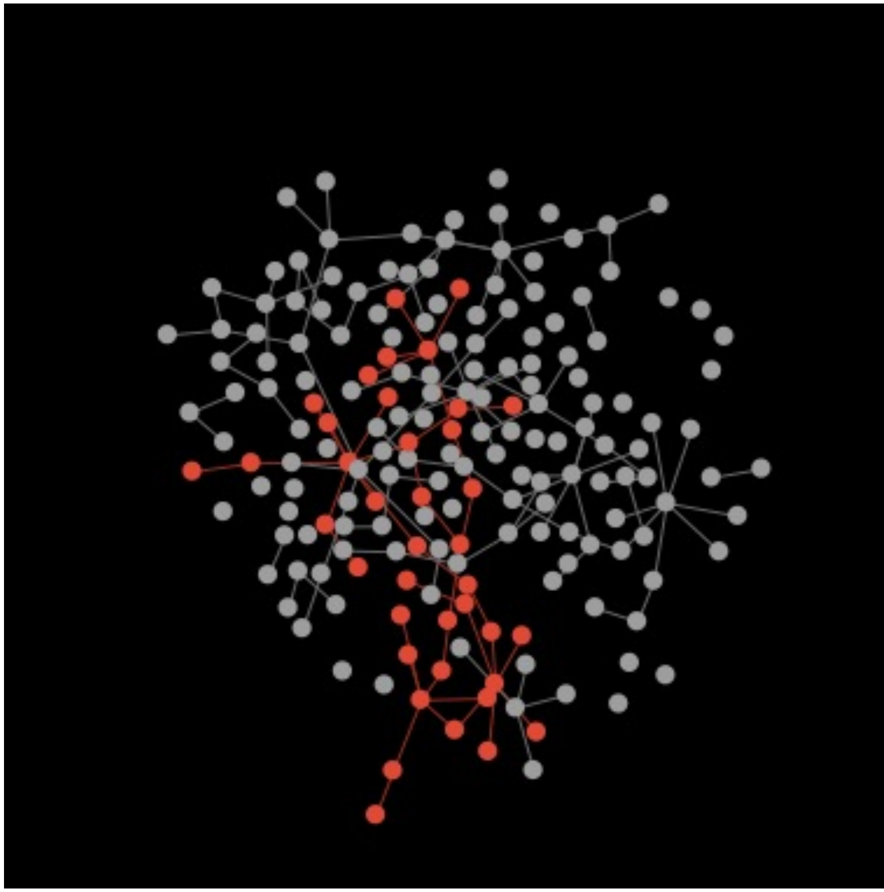
disassortative

Assortativity and resilience

assortative



disassortative



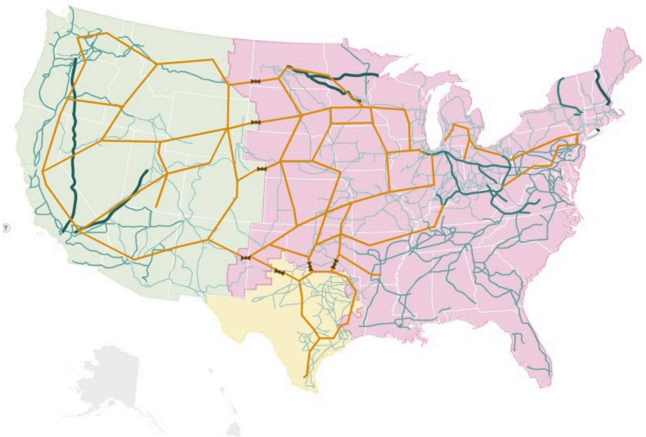
http://web.stanford.edu/class/cs224w/NetLogo/AssortativeResilience.nlogo

Is it really that simple?

- Internet?
- terrorist/criminal networks?

Power grid

- Electric power flows simultaneously through multiple paths in the network.
- For visualization of the power grid, check out NPR's interactive visualization: http://www.npr.org/templates/story/story.php?storyId=110997398



Cascading failures

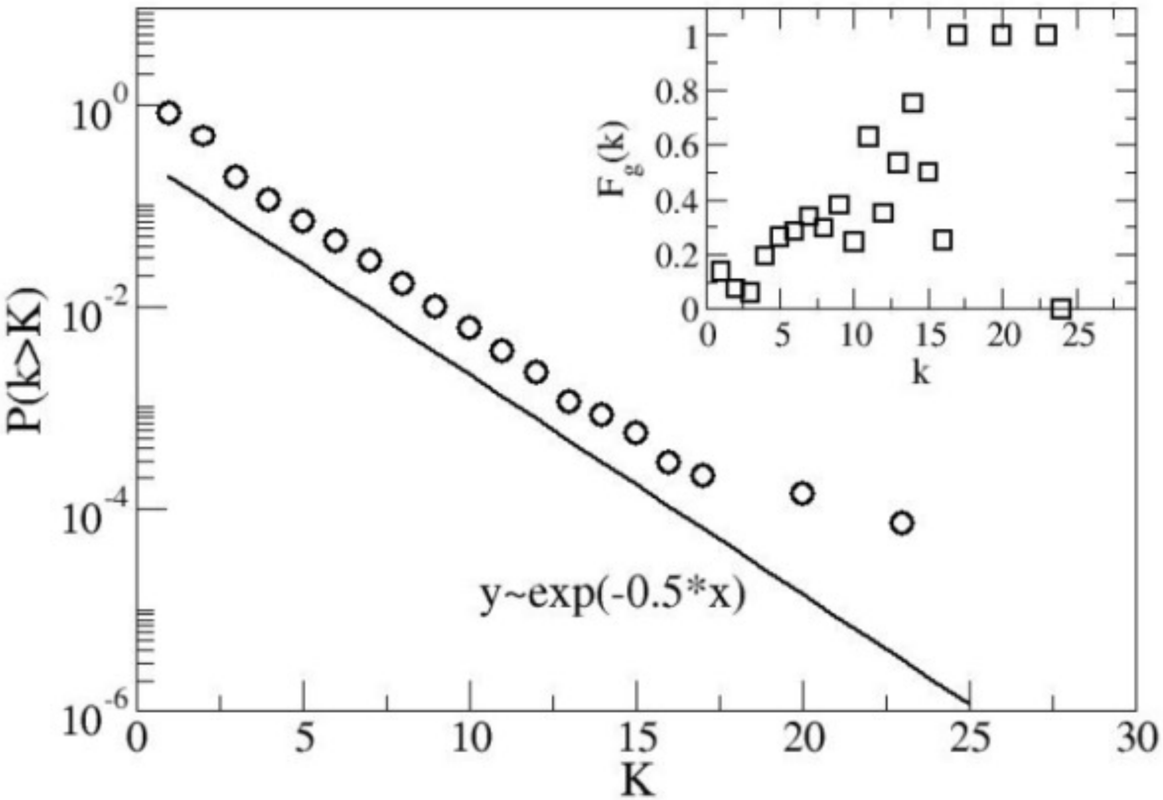
- Each node has a load and a capacity that says how much load it can tolerate.
- When a node is removed from the network its load is redistributed to the remaining nodes.
- If the load of a node exceeds its capacity, then the node fails

Case study: US power grid

Modeling cascading failures in the North American power grid
R. Kinney, P. Crucitti, R. Albert, and V. Latora, Eur. Phys. B, 2005

- Nodes: generators, transmission substations, distribution substations
- Edges: high-voltage transmission lines
- 14099 substations:
  - N_G 1633 generators,
  - N_D 2179 distribution substations
  - N_T the rest are transmission substations
- 19,657 edges

Degree distribution is exponential



$$P(k > K) \approx \exp(-0.5K)$$

Efficiency of a path

- efficiency *e* [0,1], 0 if no electricity flows between two endpoints, 1 if the transmission lines are working perfectly

- harmonic composition for a path

$$e_{path} = \left[\sum_{edges} \frac{1}{e_{edge}} \right]^{-1}$$

  - path A, 2 edges, each with e=0.5, e_{path} = 1/4
  - path B, 3 edges, each with e=0.5 e_{path} = 1/6
  - path C, 2 edges, one with e=0 the other with e=1, e_{path} = 0
- simplifying assumption: electricity flows along most efficient path

Efficiency of the network

- Efficiency of the network:
  - average over the most efficient paths from each generator to each distribution station

$$E = \frac{1}{N_G N_D} \sum_{i \in G_G} \sum_{j \in G_D} \epsilon_{ij}$$

ε_{ij} is the efficiency of the most efficient path between *i* and *j*

capacity and node failure

- Assume capacity of each node is proportional to initial load

$$C_i = \alpha L_i(0) \quad i = 1, 2..N$$

  - L represents the weighted betweenness of a node
  - Each neighbor of a node is impacted as follows

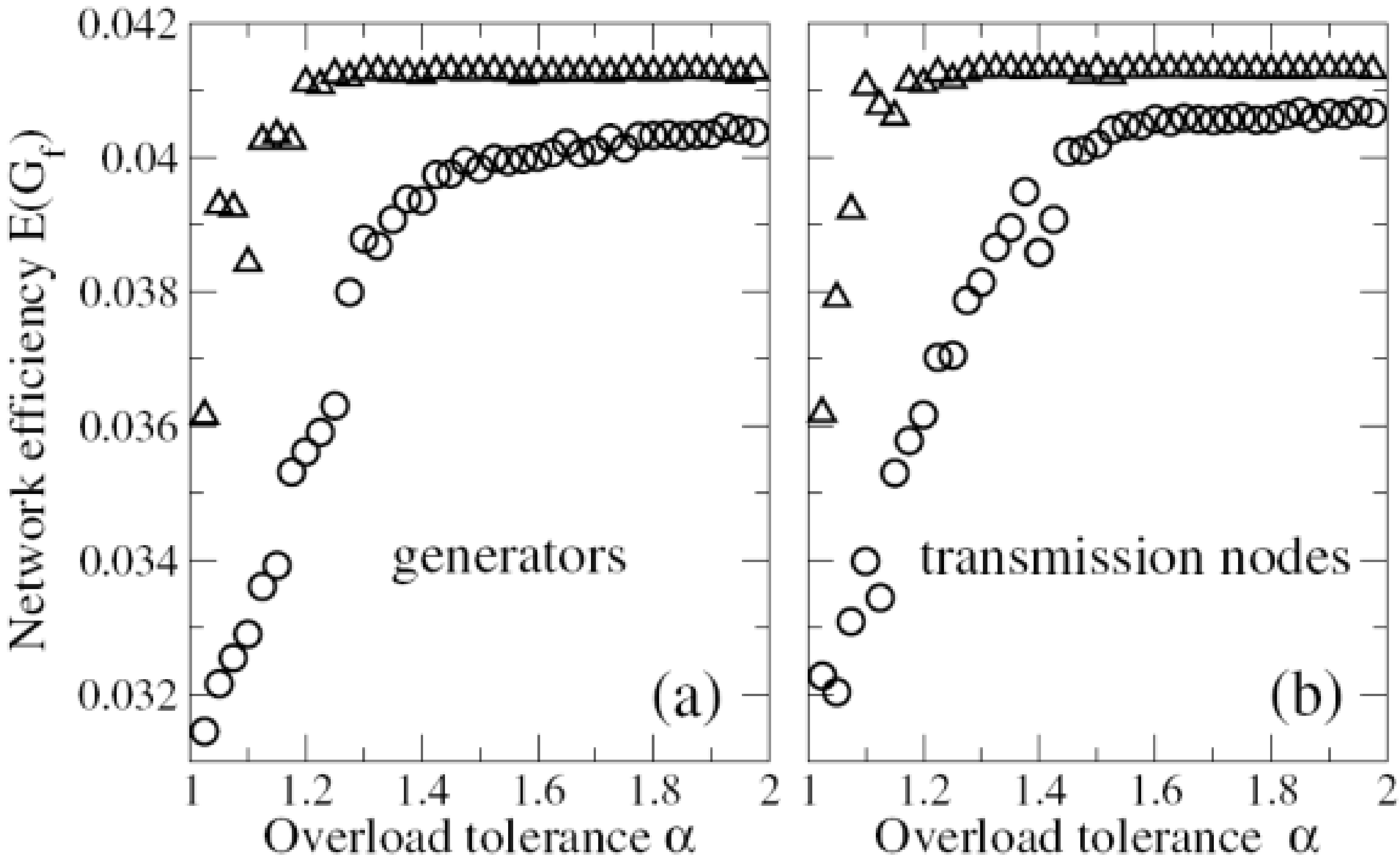
$$e_{ij}(t+1) = \begin{cases} e_{ij}(0) / \frac{L_i(t)}{C_i} & \text{if } L_i(t) > C_i \\ e_{ij}(0) & \text{if } L_i(t) \leq C_i \end{cases}$$

load exceeds capacity

- Load is distributed to other nodes/edges
- The greater α (reserve capacity), the less susceptible the network to cascading failures due to node failure

power grid structural resilience

- efficiency is impacted the most if the node removed is the one with the highest load



○ highest load generator/transmission station removed

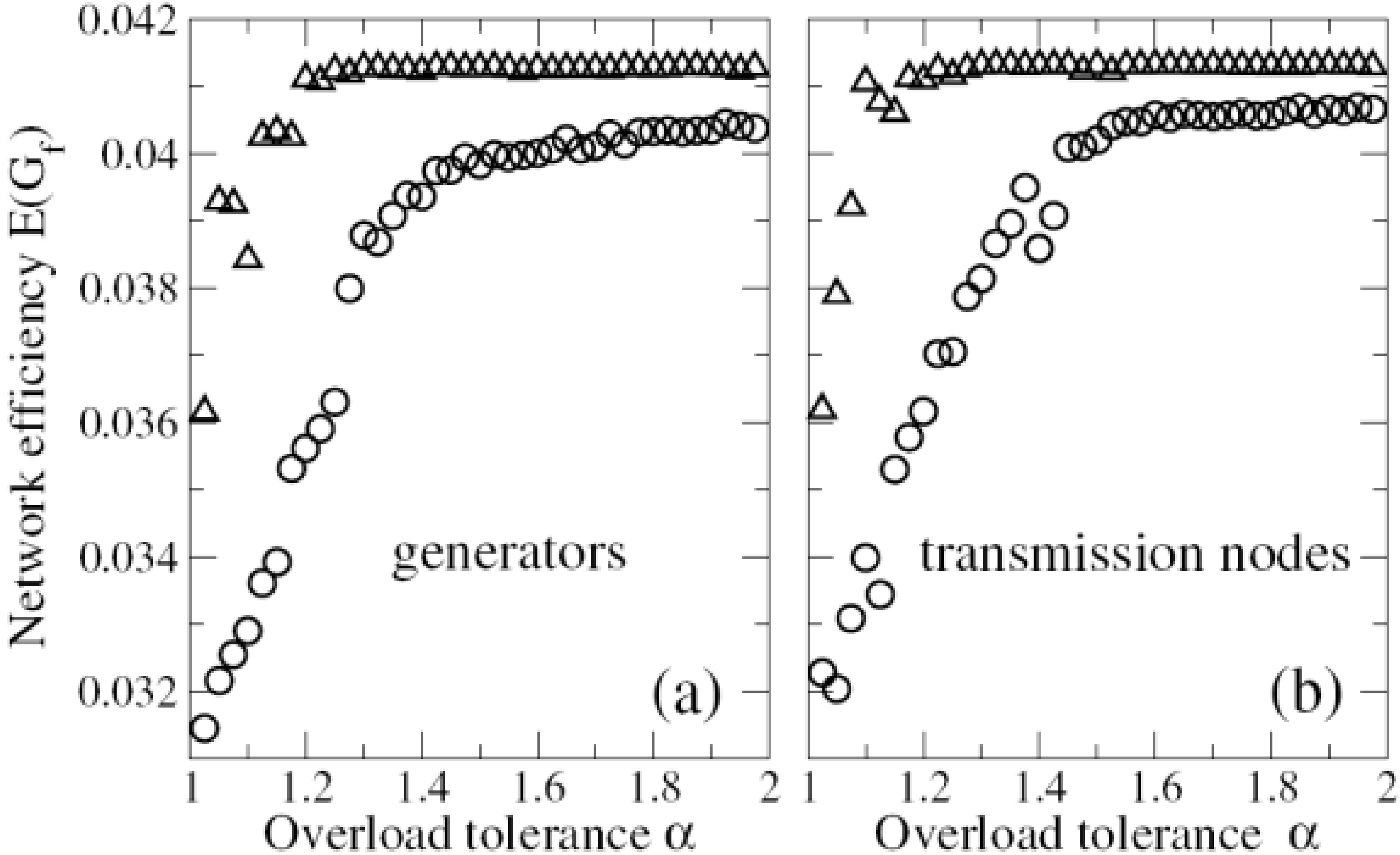
Source: Modeling cascading failures in the North American power grid; R. Kinney, P. Crucitti, R. Albert, V. Latora, Eur. Phys. B, 2005

Quiz Q:

- Approx. how much higher would the capacity of a node need to be relative to the initial load in order for the network to be efficient? (remember capacity C = α * L(0), the initial load).

power grid structural resilience

- efficiency is impacted the most if the node removed is the one with the highest load



○ highest load generator/transmission station removed

Source: Modeling cascading failures in the North American power grid; R. Kinney, P. Crucitti, R. Albert, V. Latora, Eur. Phys. B, 2005

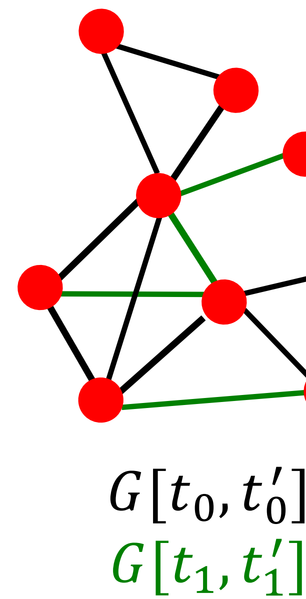
recap: network resilience

- resilience depends on topology
- also depends on what happens when a node fails
  - e.g. in power grid load is redistributed

Link Prediction and Network Inference

Link Prediction in Networks

- **The link prediction task:**
  - Given $G[t_0, t_0']$ a graph on edges up to time t_0', **output** of links (not in $G[t_0, t_0']$) that are predicted to appear in $G[t_1, t_1']$



- **Evaluation:**
  - $n = |E_{new}|$: # new edges that appear during the test period $[t_1, t_1']$
  - Take top n elements of L and count correct edges

Link Prediction via Proximity

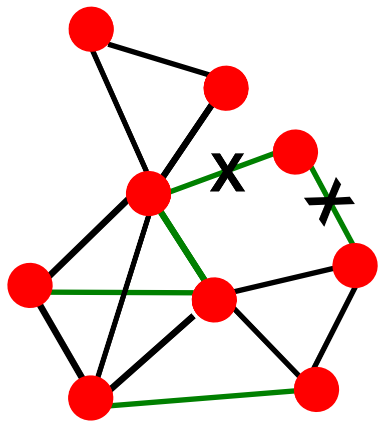
- **Predict links in an evolving collaboration network**

	training period			Core		
	authors	papers	collaborations[1]	authors	$\|E_{old}\|$	$\|E_{new}\|$
astro-ph	5343	5816	41852	1561	6178	5751
cond-mat	5469	6700	19881	1253	1899	1150
gr-qc	2122	3287	5724	486	519	400
hep-ph	5414	10254	47806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

- **Core:** Because network data is very sparse
  - Consider only nodes with degree of at least 3
    - Because we don't know enough about these nodes to make good inferences

Link Prediction via Proximity

- **Methodology:**
  - For each pair of nodes *(x,y)* compute score c*(x,y)*
    - For example, c*(x,y)* could be the # of common neighbors of x and *y*
  - Sort pairs *(x,y)* by the decreasing score c*(x,y)*
    - **Note:** Only consider/predict edges where both endpoints are in the core (*deg.* ≥ 3)
  - **Predict top n pairs as new links**
  - **See which of these links actually appear in $G[t_1, t_1']$**



Link Prediction via Proximity

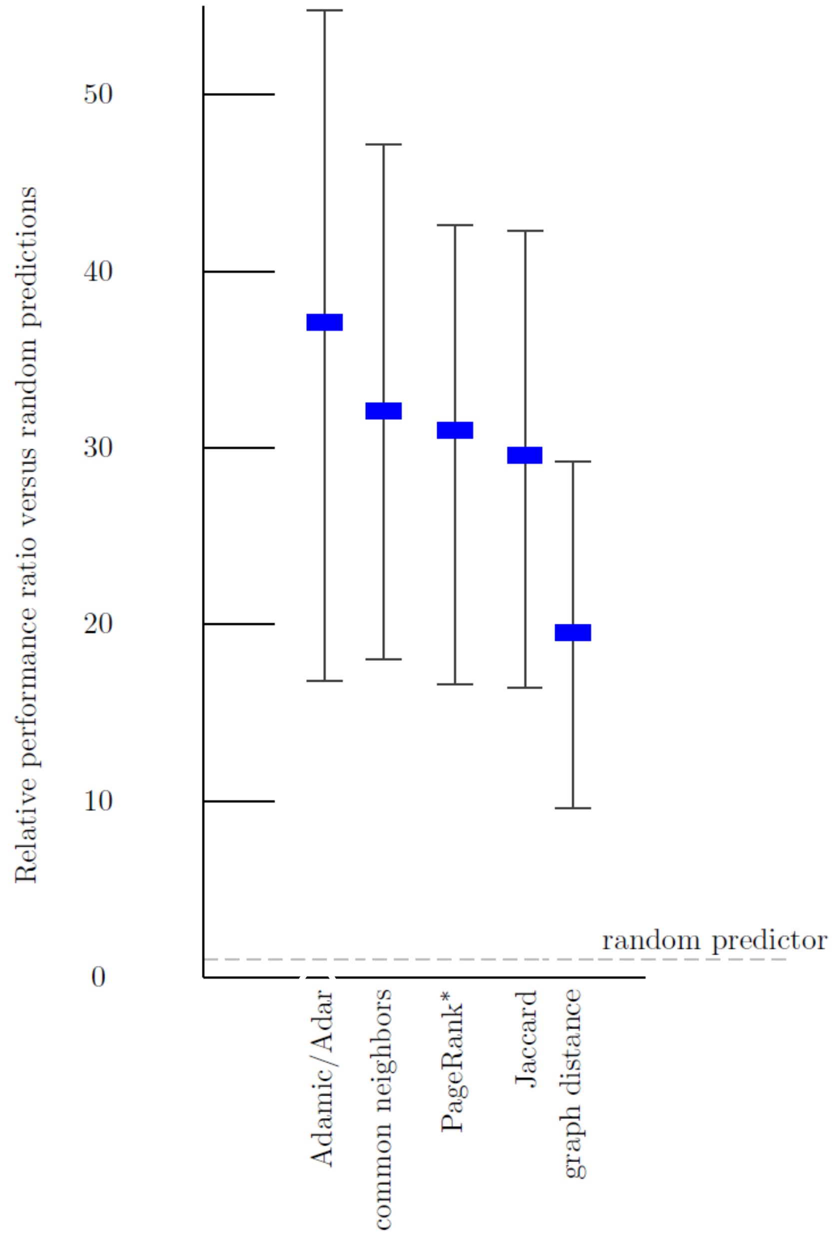
- **Different scoring functions** $c(x,y) =$
  - **Graph distance:** (negated) Shortest path length
  - **Common neighbors:** $|\Gamma(x) \cap \Gamma(y)|$
  - **Jaccard's coefficient:** $|\Gamma(x) \cap \Gamma(y)| / |\Gamma(x) \cup \Gamma(y)|$
  - **Adamic/Adar:** $\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/\log|\Gamma(z)|$
  - **Preferential attachment:** $|\Gamma(x)| \cdot |\Gamma(y)|$
  - **PageRank:** $r_x(y) + r_y(x)$
    - $r_x(y)$ … stationary distribution score of y under the random walk:
      - with prob. 0.15, jump to x
      - with prob. 0.85, go to random neighbor of current node

Γ(x) … neighbors of node x

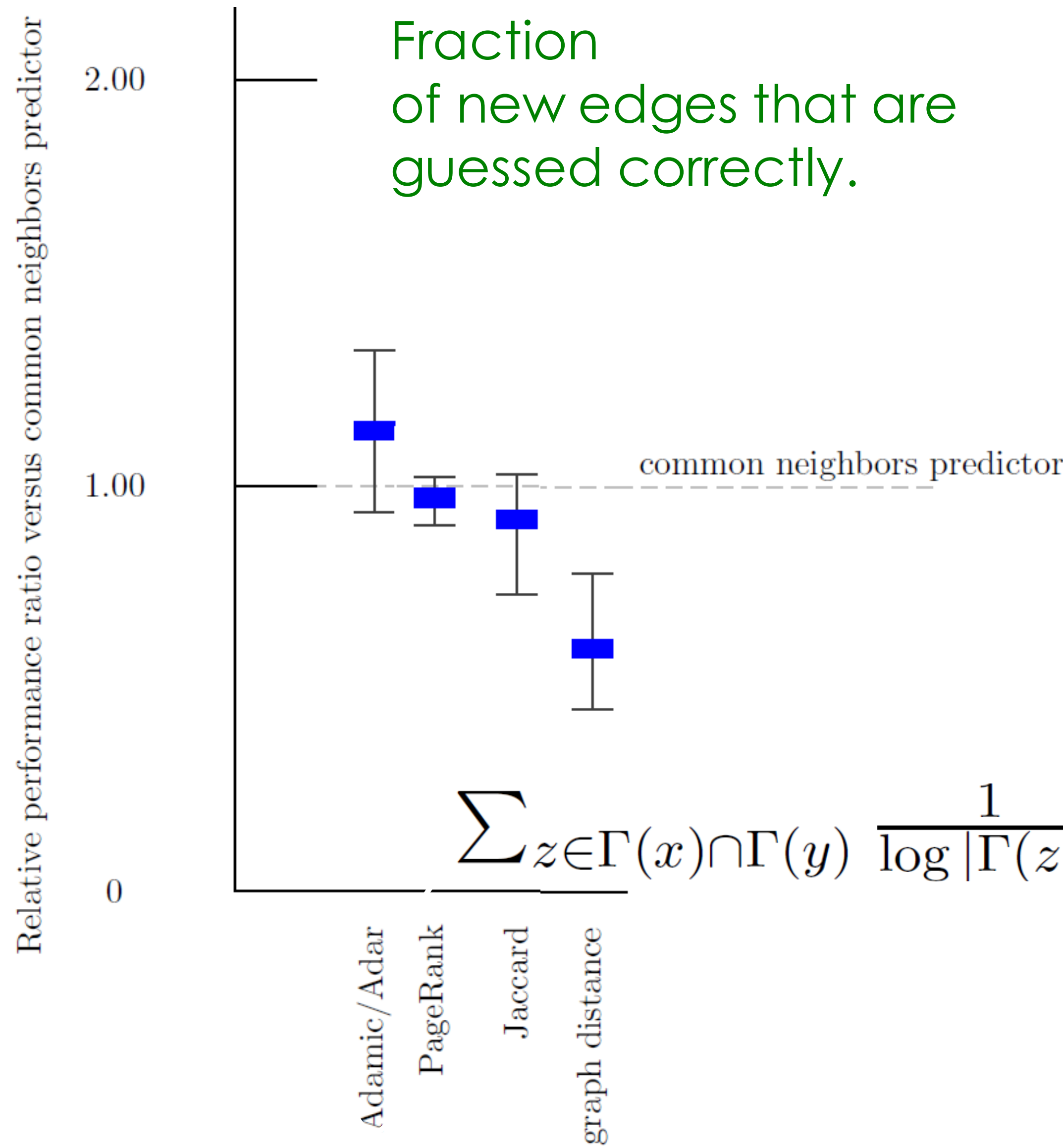
- **Then, for a particular choice of c(·)**
  - For every pair of nodes *(x,y)* compute *c(x,y)*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
  - **Predict top *n* pairs as new links**

[LibenNowell-Kleinberg ’ 03]

Results: Improvement



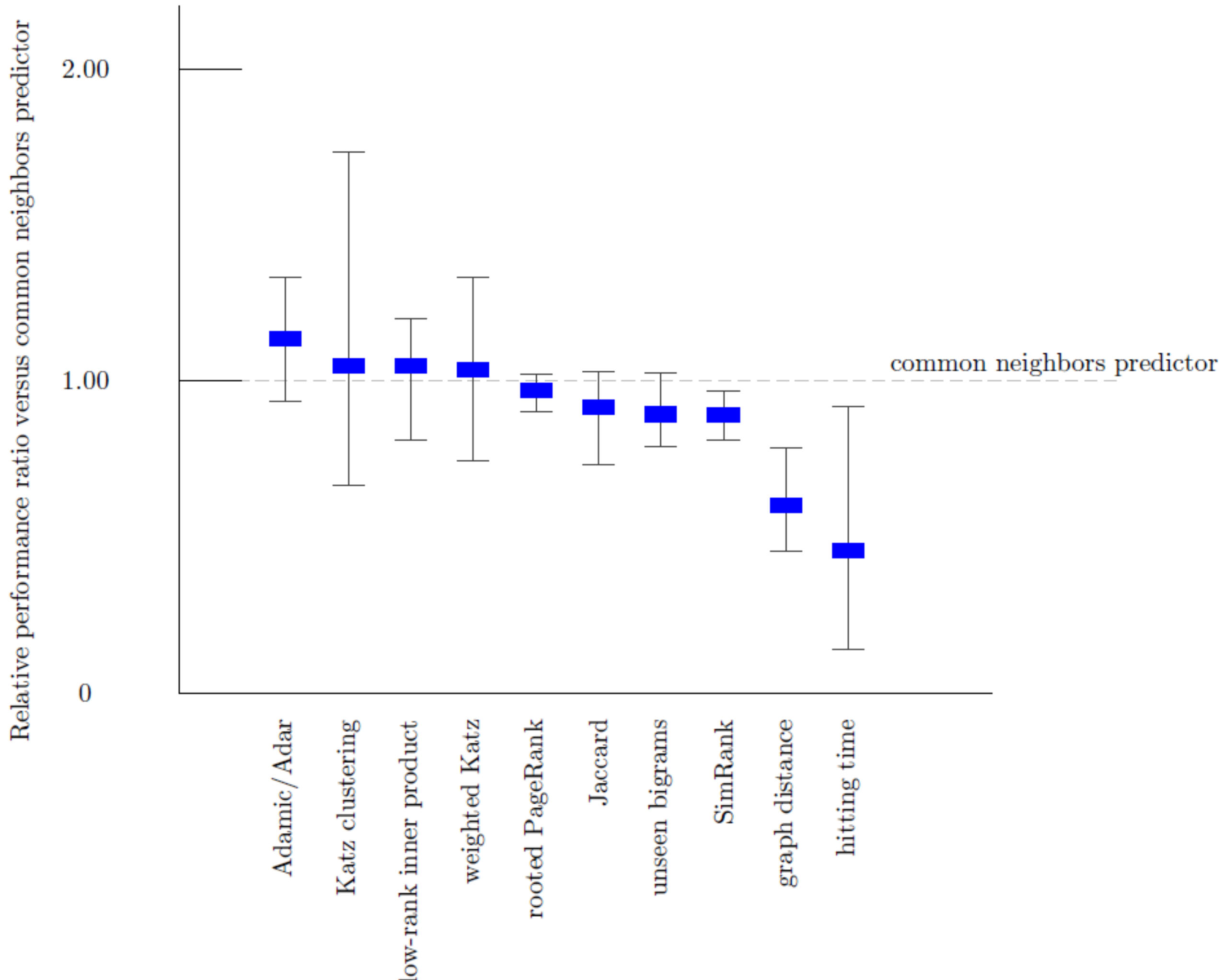
Performance score: Fraction of new edges that are guessed correctly.



$$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}$$

Results: Common Neighbors

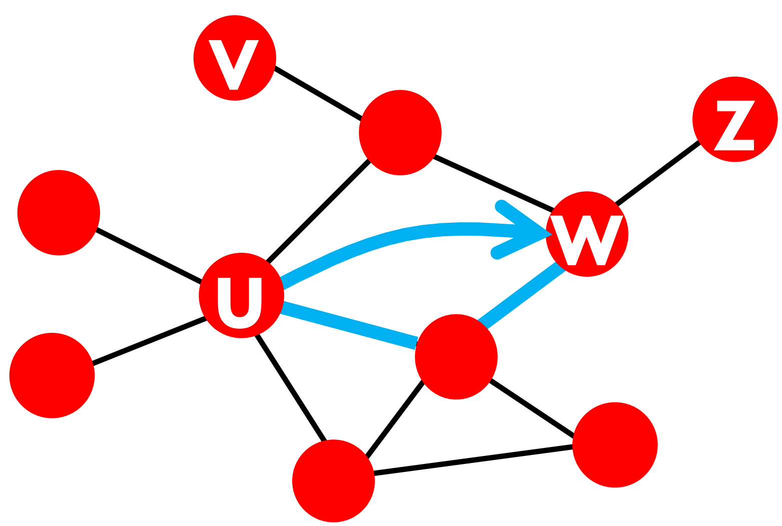
- Improvement over #common neighbors

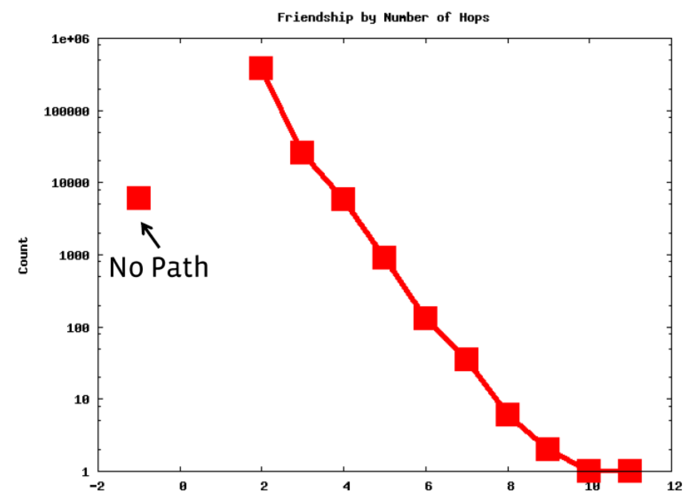


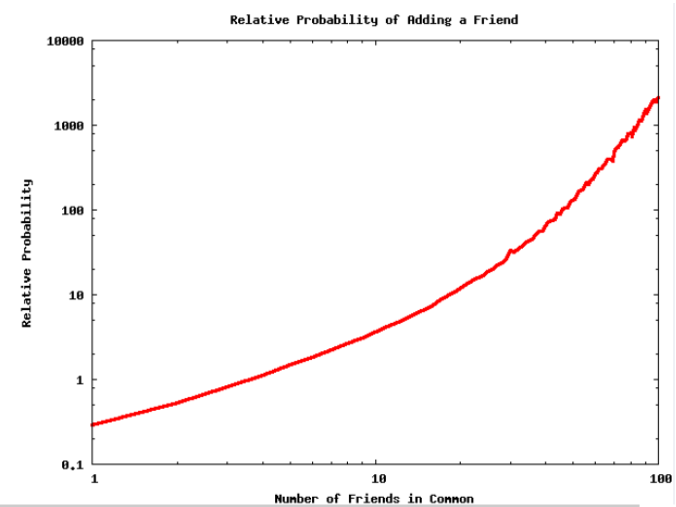
Supervised Random Walks for Link Prediction

Supervised Link Prediction

- **Can we learn to predict new friends?**
  - **Facebook's People You May Know**
  - **Let's look at the FB data:**
    - 92% of new friendships on FB are friend-of-a-friend
    - More mutual friends helps





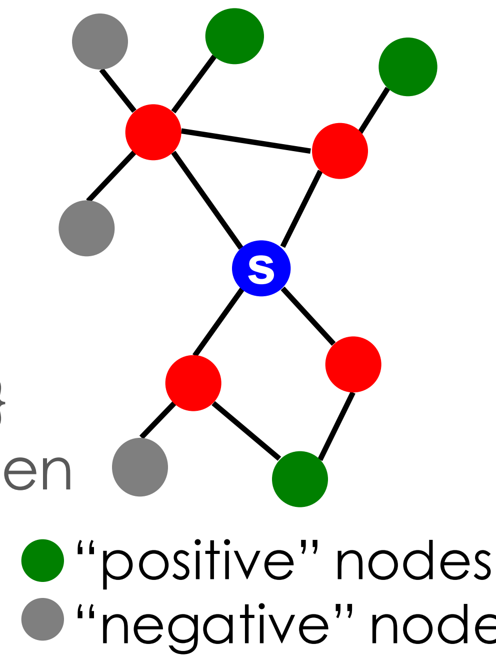


Backstrom & Leskovec, WSDM'11

Supervised Link Prediction

- **Goal: Recommend a list of possible friends**

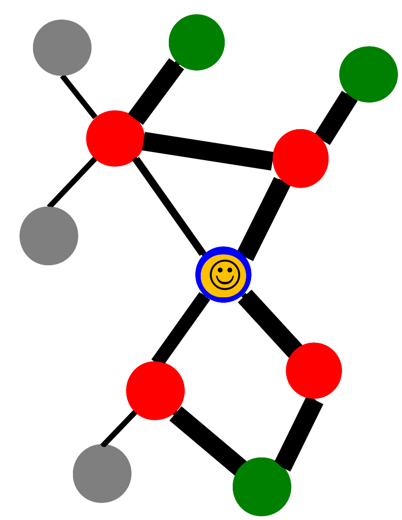
- **Supervised machine learning setting:**
  - **Labeled training examples:**
    - For every user s have a list of others she will create links to $\{d_1 \ \dots \ d_k\}$ **in the future**
      - *Use* FB network from May 2012 and $\{d_1 \ \dots \ d_k\}$ are the new friendships you created since then
      - These are the "positive" training examples
    - Use all other users as "negative" example
  - **Task:**
    - For a given node s, **score** nodes $\{d_1 \ \dots \ d_k\}$ **higher** than any other node in the network



Green nodes are the nodes to which **s** creates links in the future

Supervised Link Prediction

- **How to combine node/edge features and the network structure?**
  - Estimate **strength** of each friendship (u, v) using:
    - Profile of user u, profile of user v
    - Interaction history of users u and v
  - This creates a **weighted graph**
  - Do **Personalized PageRank from s** and measure the "**proximity**" (the visiting prob.) of any other node w from s
  - Sort nodes w by decreasing **"proximity"**



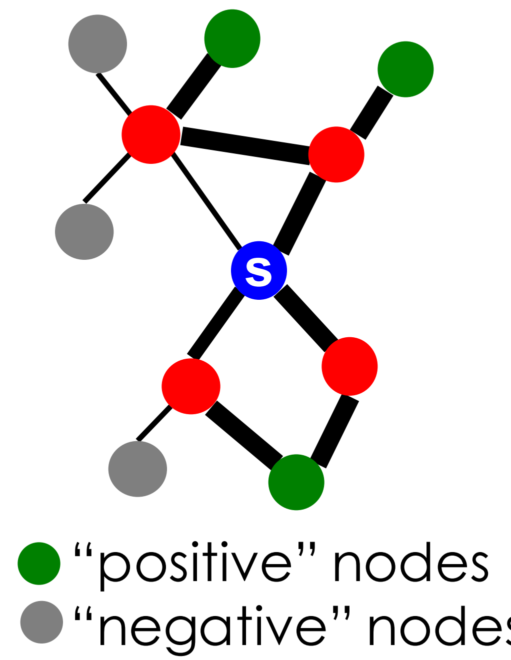
"positive" nodes
"negative" nodes

Supervised Random Walks

- Let s be the starting node
- Let $f_\beta(u,v)$ be a function that assigns **strength a_{uv} to edge (u,v)**

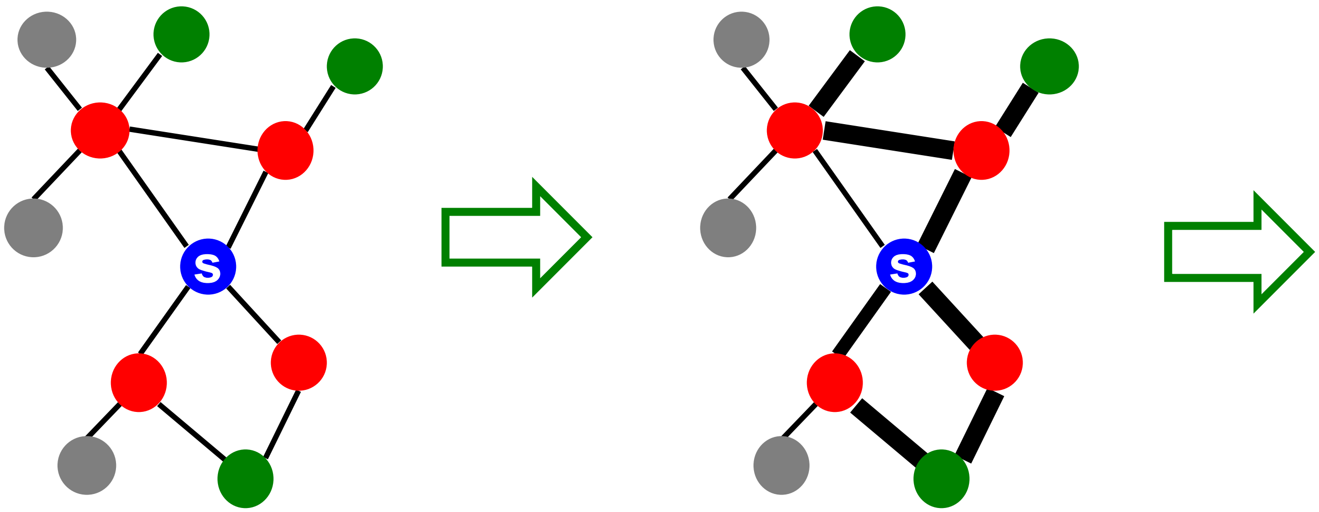
$$a_{uv} = f_\beta(u,v) = \exp(-\sum_i \beta_i \cdot x_{uv}[i])$$

  - **x_{uv} is a feature vector of (u,v)**
    - Features of node u
    - Features of node v
    - Features of edge (u,v)
  - **Note: β is the weight vector we will later estimate!**



- Do **Random Walk with Restarts** from s where transitions are according to edge strengths a_{uv}

SRW: Prediction



Network

Set edge strengths
$a_{uv} = f_\beta(u,v)$

Random Walk with Restarts on the weighted graph.
Each node w has a PageRank proximity p_w

Sort nodes w by the decreasing PageRank score p_w

Recommend top k nodes with the highest proximity p_w to node s

- **How to estimate edge strengths?**
  - **How to set parameters β of $f_\beta(u,v)$?**
- **Idea:** Set β such that it (correctly) predicts the known future links

Personalized PageRank

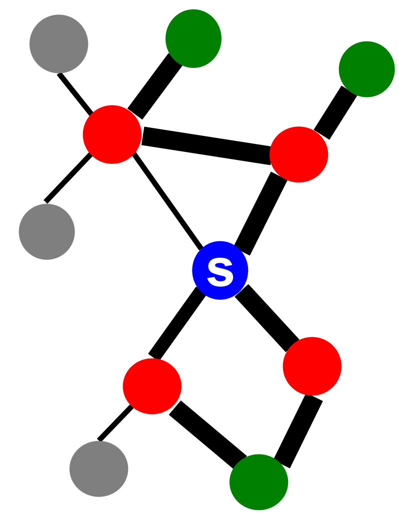
- a_{uv} …. **Strength of edge (u, v)**
- **Random walk transition matrix:**

$$Q'_{uv} = \begin{cases} \frac{a_{uv}}{\sum_w a_{uw}} & \text{if } (u,v) \in E, \\ 0 & \text{otherwise} \end{cases}$$

- **PageRank transition matrix:**

$$Q_{ij} = (1-\alpha)Q'_{ij} + \alpha \mathbf{1}(j = s)$$

  - Where with prob. α we jump back to node s



"positive" nodes
"negative" nodes

- **Compute PageRank vector:** $p = p^T Q$
- **Rank** nodes w by decreasing p_w

The Optimization Problem

- **Positive** examples

  $D = \{d_1, \dots, d_k\}$

- **Negative** examples

  $L = \{other\ nodes\}$

- **What do we want?**

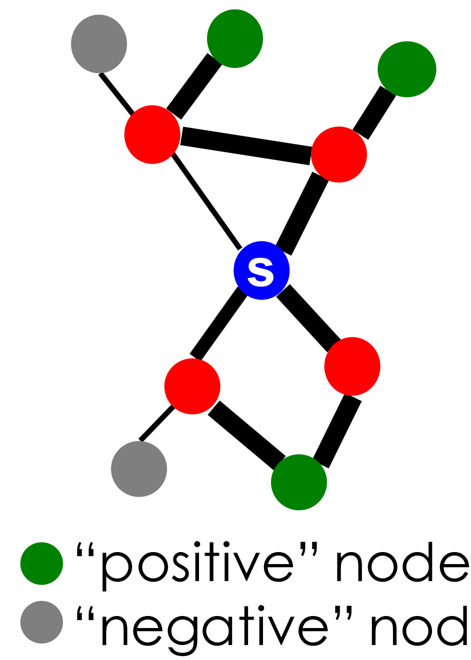
$$\min_{\beta} F(\beta) = ||\beta||^2$$

We prefer small weights β to prevent overfitting

such that

$$\forall\, d \in D, l \in L :\ p_l < p_d$$

Every positive example has to have higher PageRank score than every negative example



- **Note:**
  - Exact solution to this problem may not exist
  - So we make the constraints “soft” (i.e., optional)

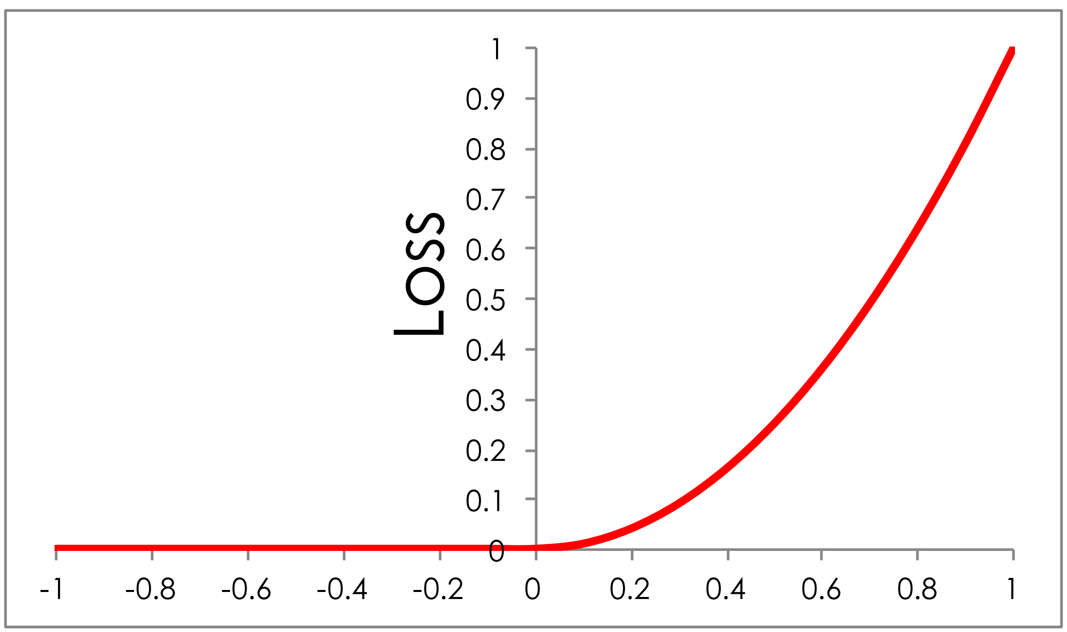
Making Constraints "Soft"

- **Want to minimize:**

$$\min_{\beta} F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda ||\beta||^2$$

  - **Loss:** $h(x) = 0$ if $x < 0$, or x^2 else

Penalty for violating the constraint that $p_d > p_l$

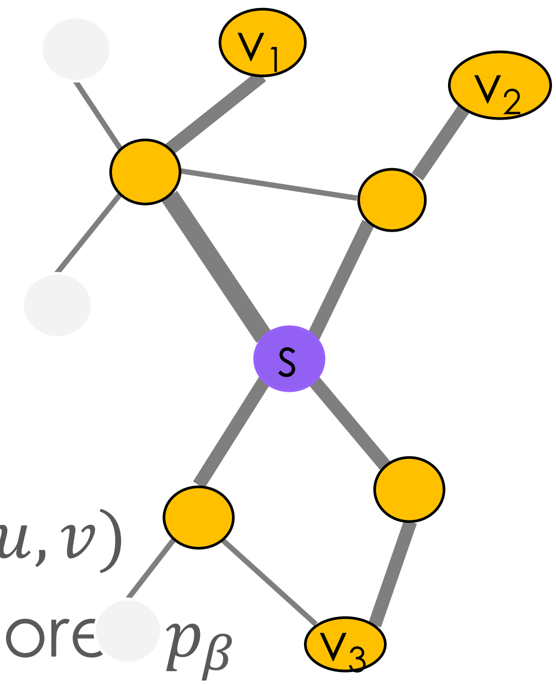


$p_l < p_d$ $\quad$ $p_l = p_d$ $\quad$ $p_l > p_d$

Solving the problem: Intuition

- **How to minimize *F*?**

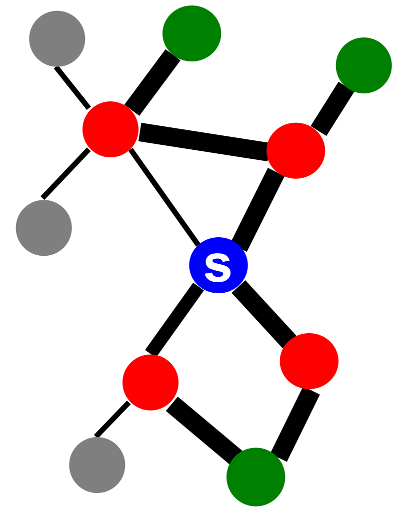
$$\min_{\beta} F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda ||\beta||^2$$



- **Both p_l and p_d depend on β**
  - Given β assign edge weights $a_{uv} = f_\beta(u,v)$
  - Using $Q = [a_{uv}]$ compute PageRank score p_β
  - Rank nodes by the decreasing score

- **Goal: Want to find β such that $p_l < p_d$**

Solving the Problem: Intuition



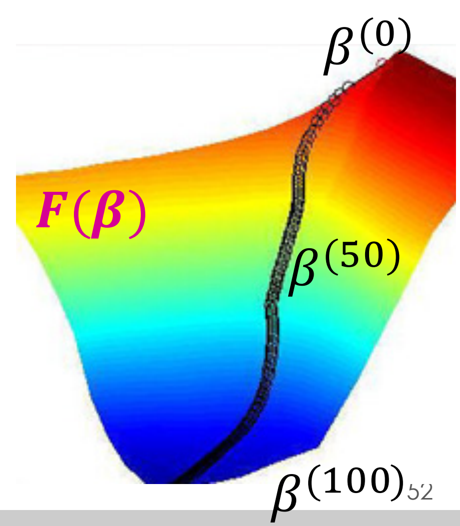
- **How to minimize $F(\beta)$?**

$$\min_{\beta} F(\beta) = \sum_{d \in D, l \in L} h(p_l - p_d) + \lambda ||\beta||^2$$

- **Idea:**
  - Start with some random $\beta^{(0)}$
  - Evaluate the derivative of $F(\beta)$ and do a small step in the opposite direction

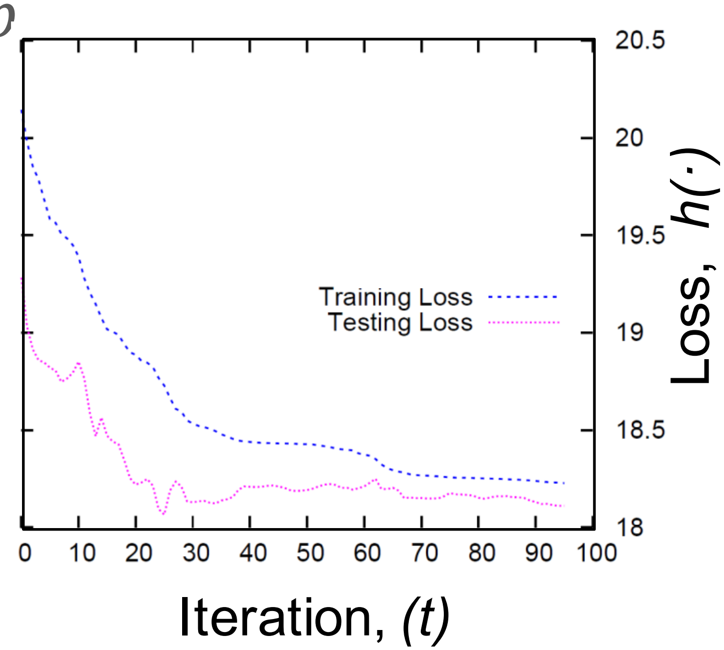
$$\beta^{(t+1)} = \beta^{(t)} - \eta \frac{\partial F(\beta^{(t)})}{\partial \beta}$$

  - Repeat until convergence

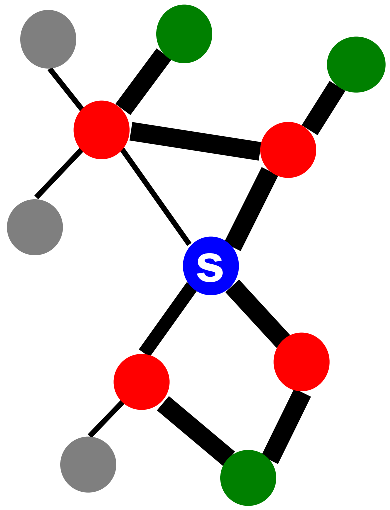


Optimizing $F(\beta)$

- **To optimize $F(\beta)$, use gradient descent:**
  - Pick a random starting point $\beta^{(0)}$
  - Using current $\beta^{(t)}$ compute edge strenghts and the transition matrix Q
  - Compute PageRank scores p
  - Compute the gradient with respect to weight vector $\beta^{(t)}$
  - Update $\beta^{(t+1)}$



Data: Facebook



- **Facebook Iceland network**
  - 174,000 nodes (55% of population)
  - Avg. degree 168
  - Avg. person added 26 friends/month
- **For every node s:**
  - **Positive examples:**
    - D = { new friendships s created in Nov '09 }
  - **Negative examples:**
    - L = { other nodes s did not create new links to }
  - **Limit to friends of friends:**
    - On avg. there are 20,000 FoFs (maximum is 2 million)!

Experimental setting

- **Node and Edge features for learning:**
  - **Node:** Age, Gender, Degree
  - **Edge:** Age of an edge, Communication, Profile visits, Co-tagged photos
- **Evaluation:**
  - **Precision at top 20**
    - We produce a list of 20 candidates
      - By taking top 20 nodes x with highest PageRank score p_x
    - Measure to what fraction of these nodes s actually links to

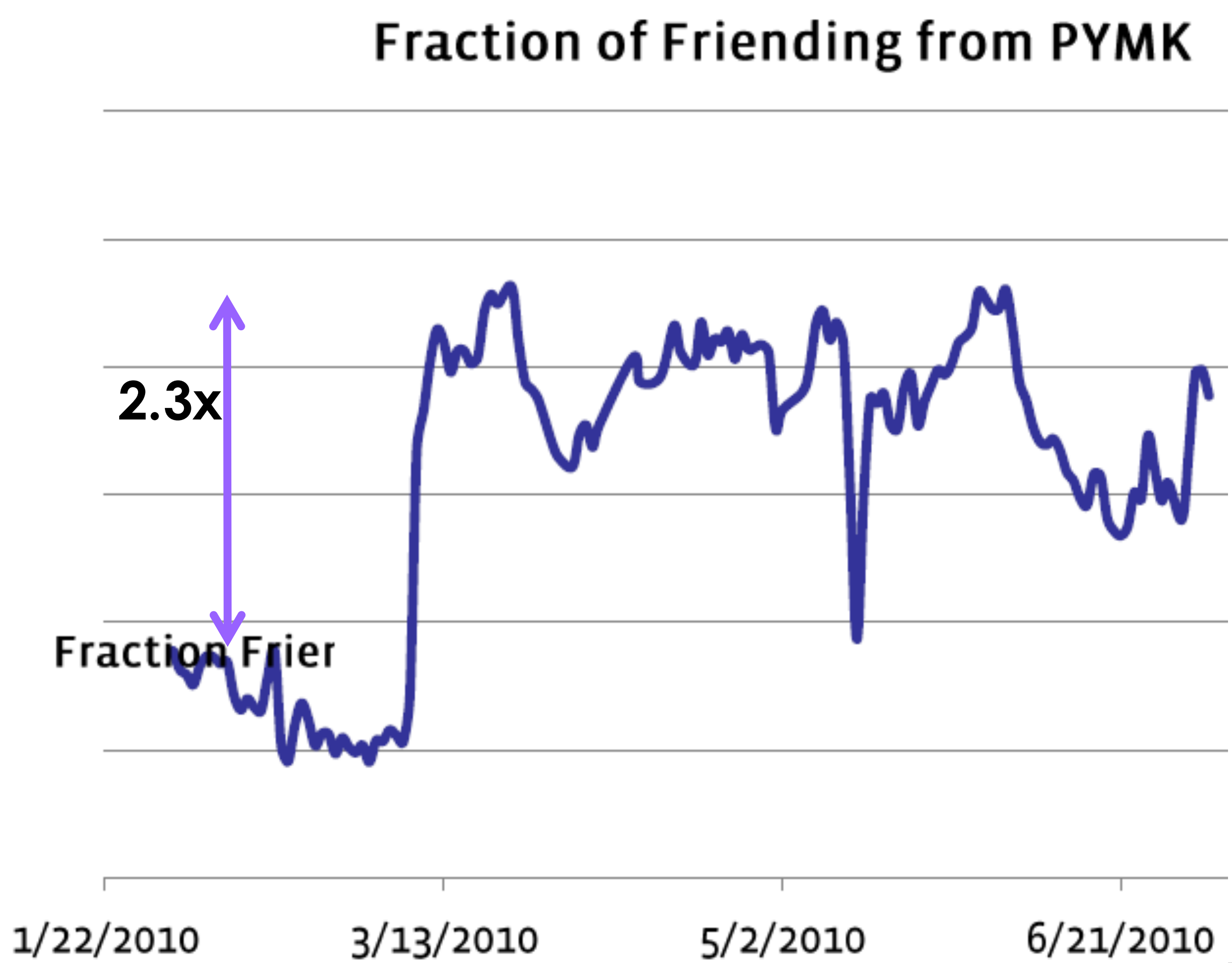
Results: Facebook Iceland

- **Facebook:** Predict future friends
  - Adamic-Adar already works great
  - Supervised Random Walks (SRW) gives slight improvement

Learning Method	Prec@Top20
Random Walk with Restart	6.80
Adamic-Adar	7.35
Common Friends	7.35
Degree	3.25
SRW: one edge type	6.87
SRW: multiple edge types	**7.57**

Results: Facebook

- 2.3x improvement over previous FB-PYMK (People You May Know)



Results: Co-Authorship

- **Arxiv Hep-Ph collaboration network:**
  - Poor performance of unsupervised methods
  - SRW gives a boost of 25%!

Learning Method	Prec@Top20
Random Walk with Restart	3.41
Adamic-Adar	3.13
Common Friends	3.11
Degree	3.05
SRW: one edge type	4.24
SRW: multiple edge types	**4.25**

Topic mash-up write-up

- Network structure matters to resilience
  - and assortativity too
- Link prediction is an interesting and useful task