Graph Models 2 and Random Walks

Sarunas Girdzijauskas

ID2211

March 2019

Recap

- Basic Notations
- Type of graphs
- Paths/cycles/Connectivity/Giant component
- Centrality measures
- Metrics Comparison
- Clustering Coefficient
- Degree Distributions
- Real Networks
- **Graph Models:**
  - **Random Graph, Configuration, Pref. attachment, Watts-Strogatz Small-world, Navigable Networks**

Overview of the Results

Navigable

Search time T:

$$O((\log n)^{\beta})$$

Kleinberg's model

$$O((\log n)^{2})$$

Not navigable

Search time T:

$$O(n^{\alpha})$$

Watts-Strogatz

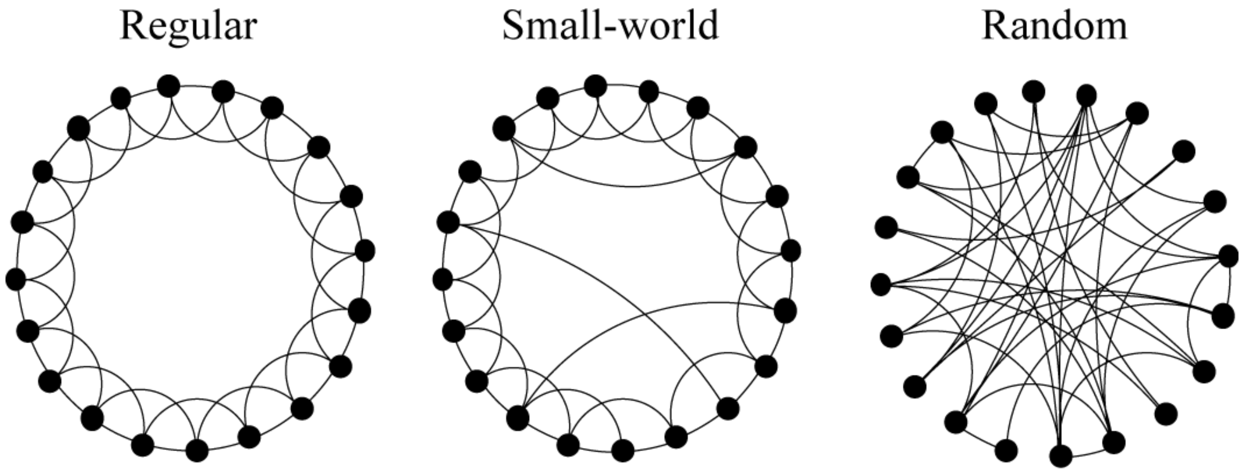
$$O(n^{\frac{2}{3}})$$

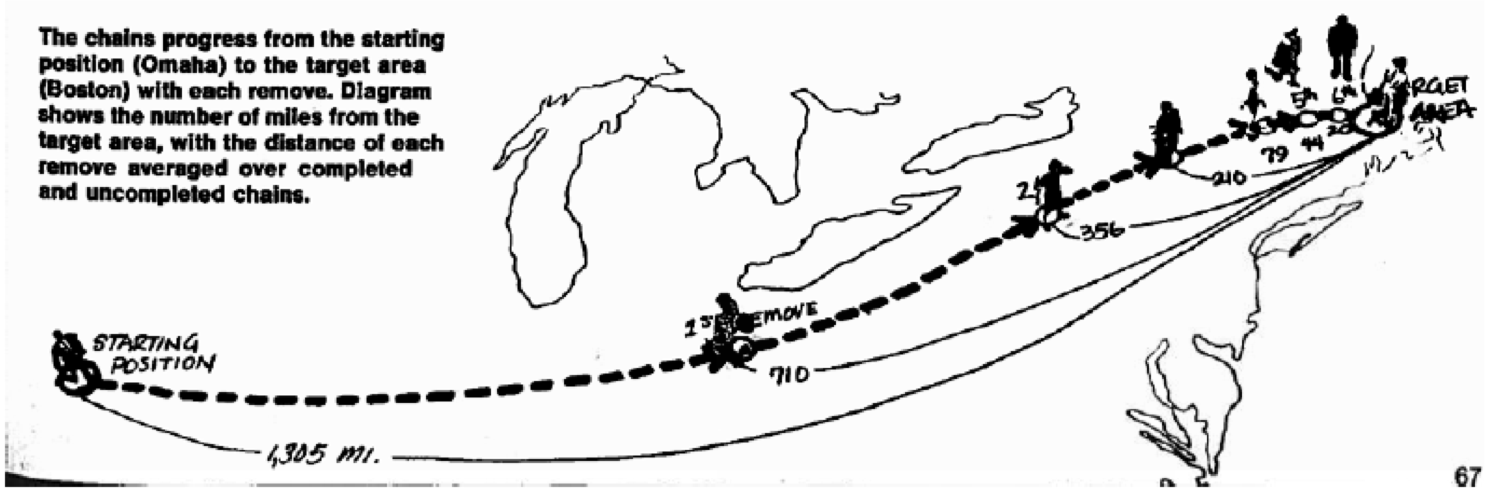
Erdős–Rényi

$$O(n)$$

Note: We know these graphs have diameter *O(log n)*.
So in Kleinberg's model search time is <u>polynomial</u> in *log n*,
while in Watts-Strogatz it is <u>exponential</u> (in *log n*).

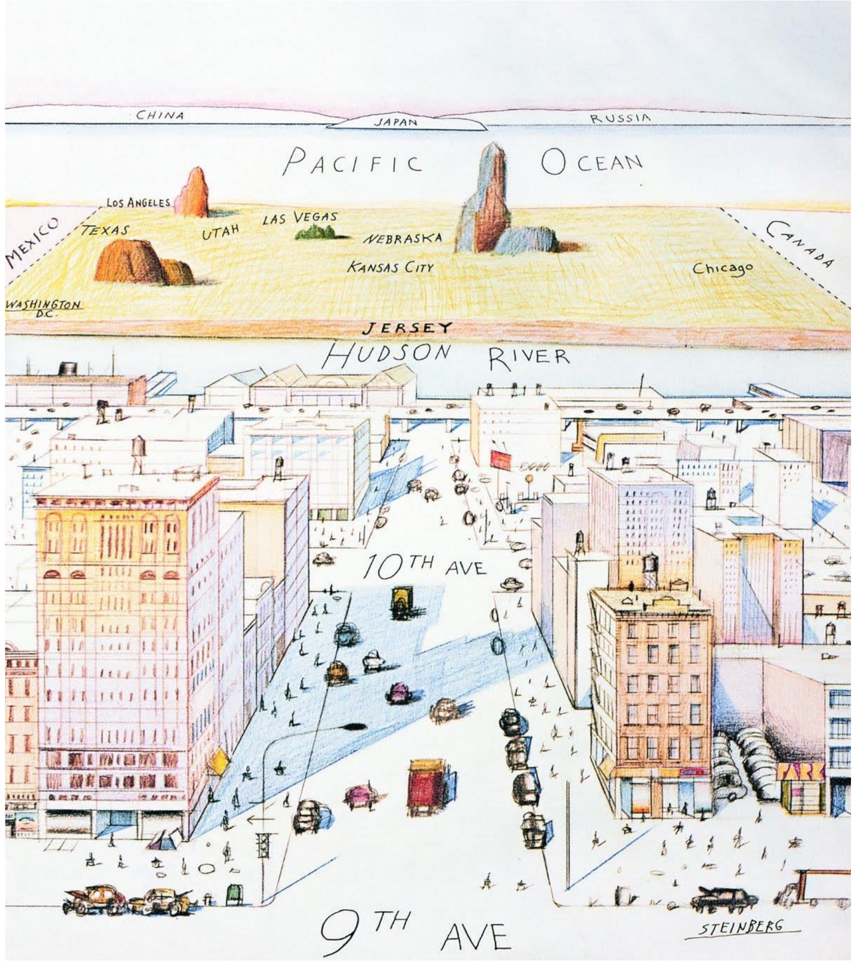
How could we make Small-World Navigable?





Kleinberg’s model of Small-Worlds

- Kleinberg claims that Watts and Strogatz model **is not effective** for decentralized search;
- presents the **infinite family of SW networks** that generalizes Watts and Strogatz model and shows that decentralized search algorithms can **find short paths** with high probability;
- proves that there exist only **one unique model** within that family for which decentralized algorithms **are effective**.
- **What’s the intuition for his model?**
  - Links are not random, but inversely proportional to the distance

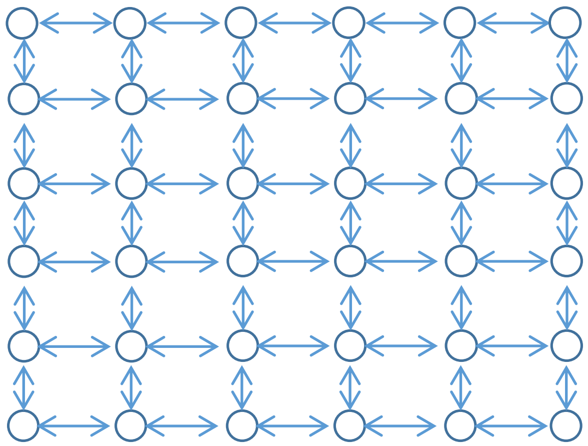


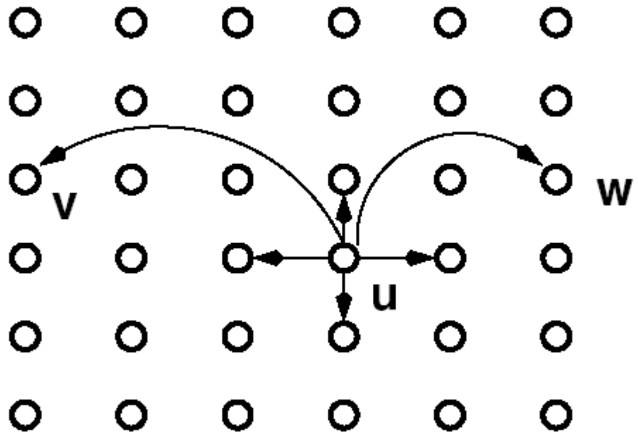
The New Yorker Cover March 29, 1976
Saul Steinberg, “View of the World from 9th Avenue”
https://en.wikipedia.org/wiki/View_of_the_World_from_9th_Avenue

Navigable Small-World networks

- Kleinberg’s Small-World’s model
  - 2-dimensional lattice
  - Lattice (Manhattan) **distance**
  - Two type of links:
    - Short range (neighborhood lattice)
    - Long range
      - Probability for a node *u* to have a node *v* as a long range contact is proportional to

$$P(u \to v) \sim \frac{1}{d(u,v)^r}$$





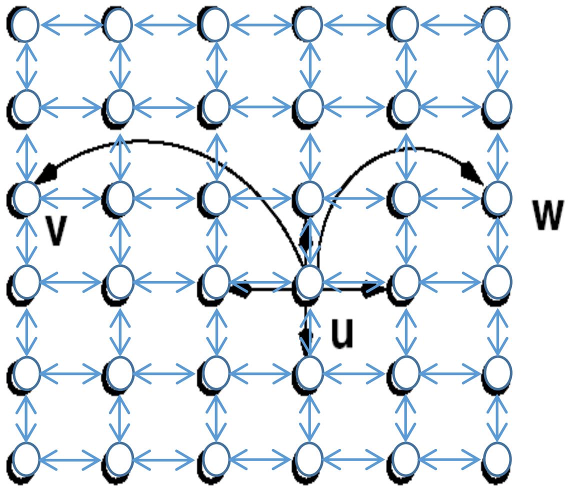
How does it work in practice?

$$P(u \to v) \sim \frac{1}{d(u,v)^r}$$

$$P(u \to v) = \frac{1}{d(u,v)^r} \cdot \frac{1}{Z}$$

- Normalization constant have to be calculated:

$$Z = \sum_{\forall i \neq u} \frac{1}{d(u,i)^r}$$



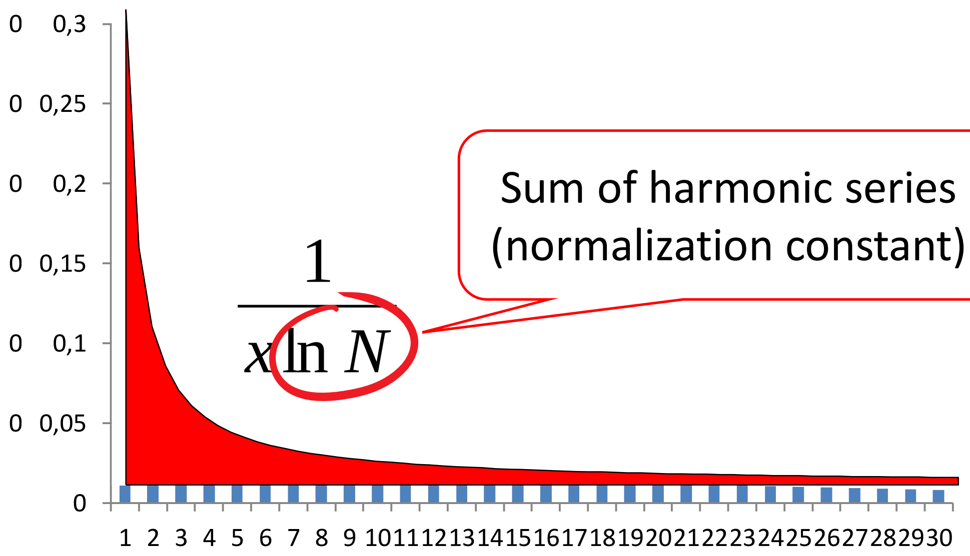
Example

- Choose among 3 friends (1-dimension)
  - A (1 mile away)
  - B (2 miles away)
  - C (3 miles away)
- Normalization constant $\sum_{\forall i \neq u} \frac{1}{d(u,i)} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$

$$P(selectingA) = \frac{\frac{1}{1}}{\frac{11}{6}} = \frac{6}{11}$$

$$P(selectingB) = \frac{\frac{1}{2}}{\frac{11}{6}} = \frac{3}{11}$$

$$P(selectingC) = \frac{\frac{1}{3}}{\frac{11}{6}} = \frac{2}{11}$$



$$\frac{1}{x \ln N}$$

Sum of harmonic series (normalization constant)

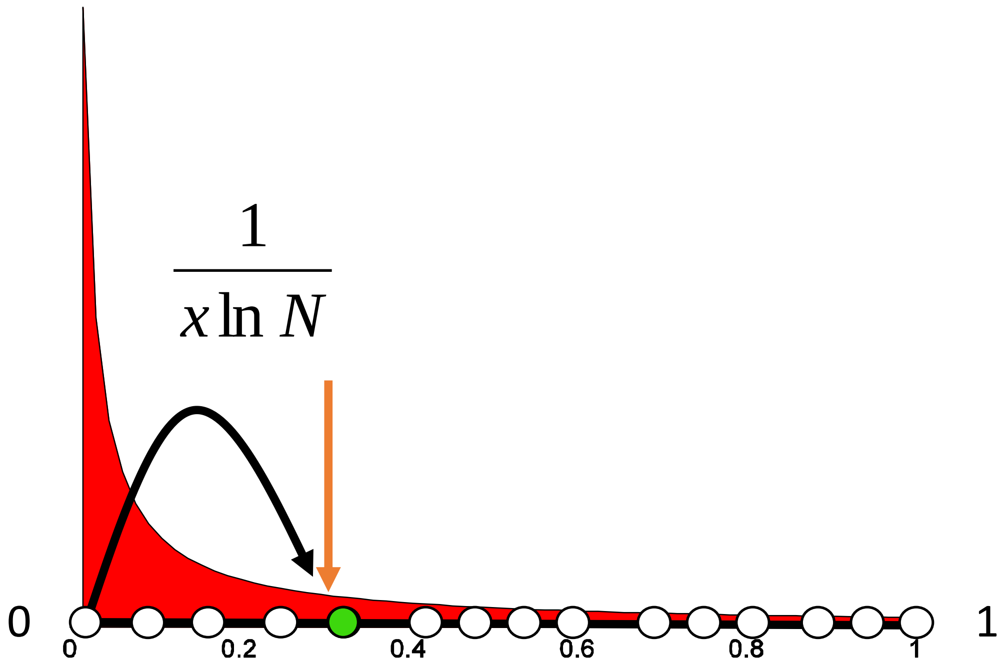
1-dimensional continuous case

- Peers uniformly distributed on a unit interval (or a ring structure)
- Long range links chosen by Kleinberg's small-world principle

- Search cost

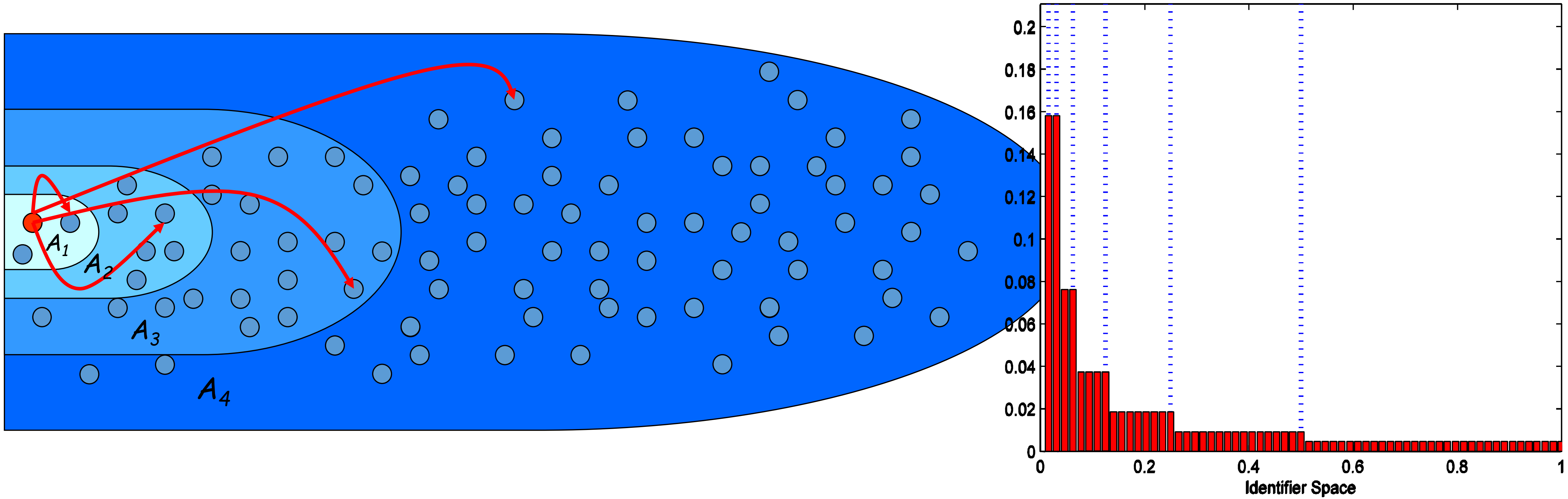
O($\log^2 N/k$) with k long-range links

O(logN) with O(logN) long-range links



Approximation of Kleinberg's model

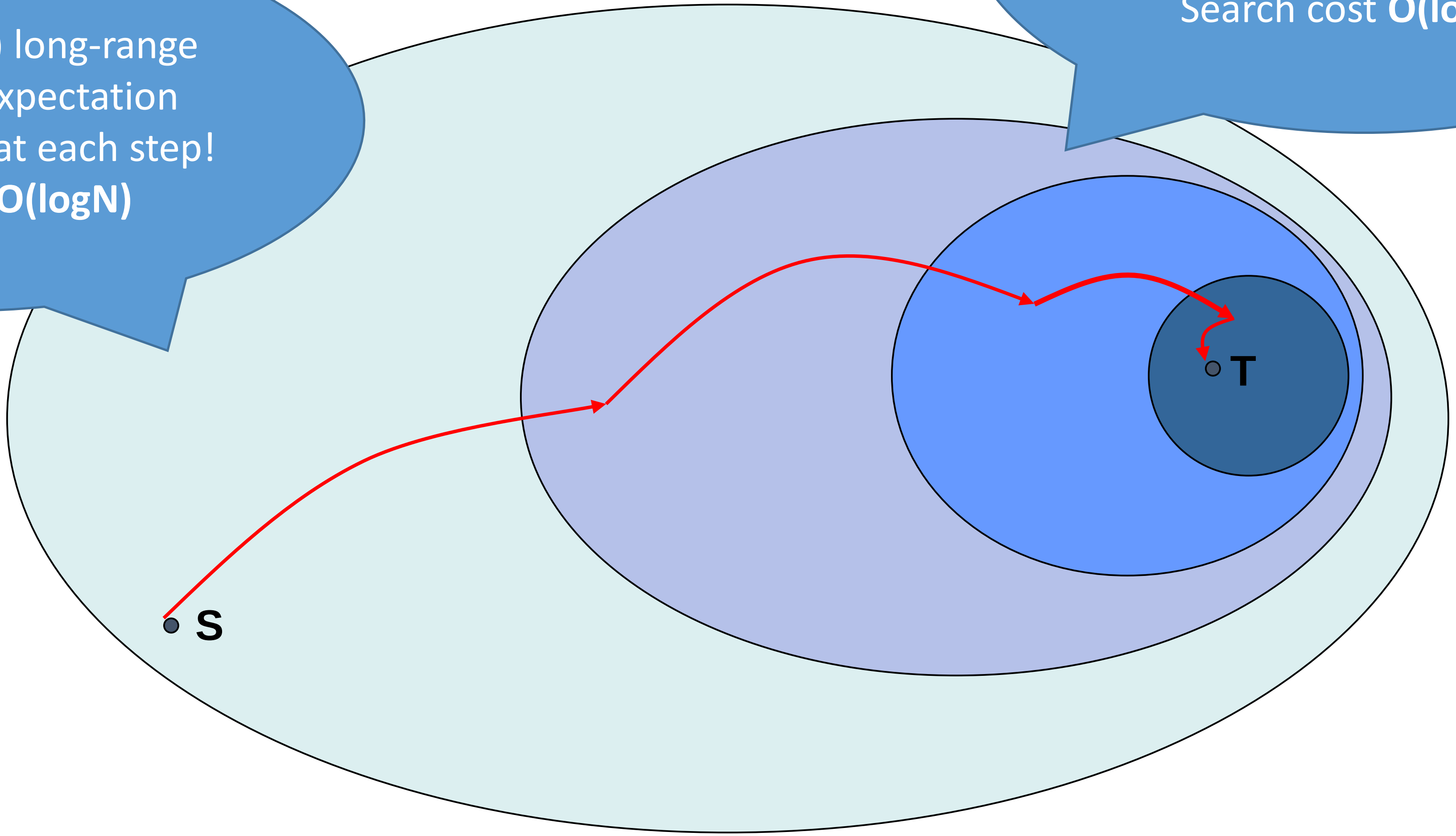
- Given node *u* if we can partition the remaining peers into sets $A_1, A_2, A_3, \ldots, A_{logN}$, where A_i, consists of all nodes whose distance from *u* is between 2^{-i} and 2^{-i+1}.
  - *Then given r=dim each long range contact of u is nearly equally likely to belong to any of the sets* A_i
  - *When q=logN – on average each node will have link in each set of* A_i



Why Does it Work?

If I have O(logN) long-range links I can on expectation halve a distance at each step! Search cost **O(logN)**

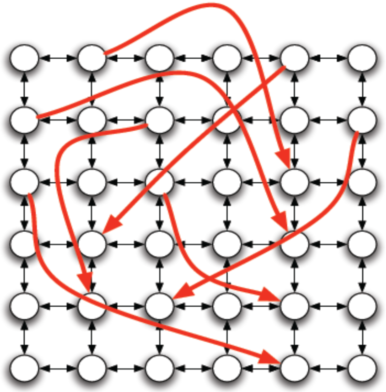
If I have O(1) long-range links will have to spend O(logN) time at each stage until I can halve a distance at each step! Search cost **O($\log^2$ N)**



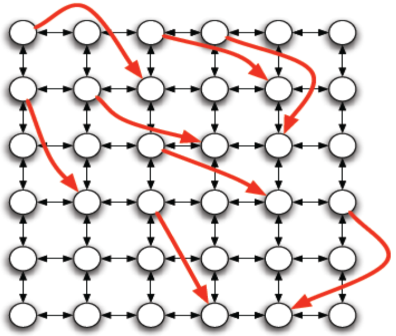
Influence of “r”

- Each peer *u* has link to the peer *v* with probability proportional to $\frac{1}{d(u,v)^r}$ where d(u,v) is the distance between *u* and *v.*

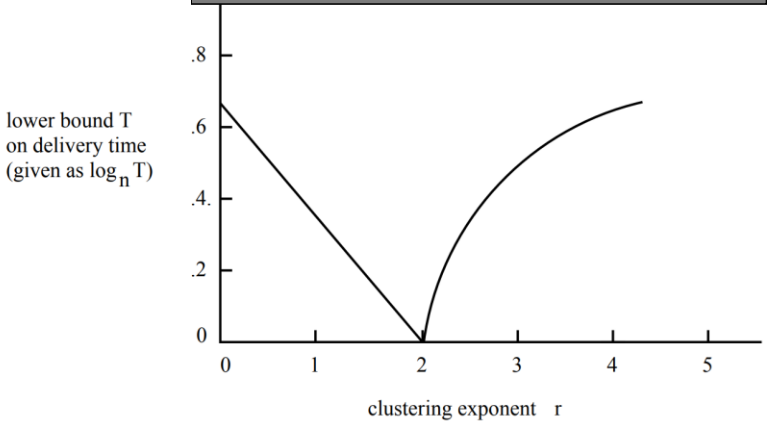
- **Tuning “r”**
  - When **r=0** – long range contacts are chosen uniformly (Similar to Watts-Strogatz). The short paths exist between every pair of vertices, **BUT there is no decentralized algorithm capable finding these paths** efficiently.
  - When **r<dim** we tend to choose more far away neighbors (decentralized algorithm can quickly approach the neighborhood of target, but then slows down till finally reaches target itself).
  - When **r>dim** we tend to choose more close neighbors (algorithm finds q target in it’s neighborhood, but reaches it slowly if it is far away).
  - When **r=dim** (dimension of the space), the algorithm exhibits optimal performance.



Small r: too many long links



Big r: to many short links



Performance with r=dim

- When there is one long range link (q = 1):
  - The expected search cost is bounded **by $O(\log^2 N)$**
- When there are constant number of long range links (q = k ≤ logN):
  - The expected search cost is bounded by

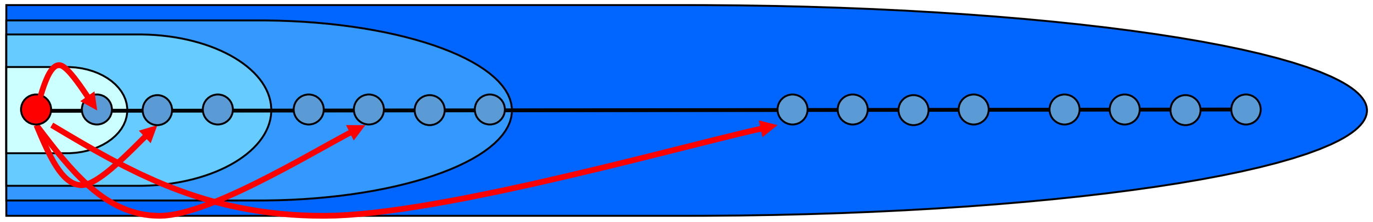
$O(\log^2 N)/k$

- When q = logN:
  - The expected search cost is bounded **by $O(\log N)$**
  - Notice similarity with P2P systems!

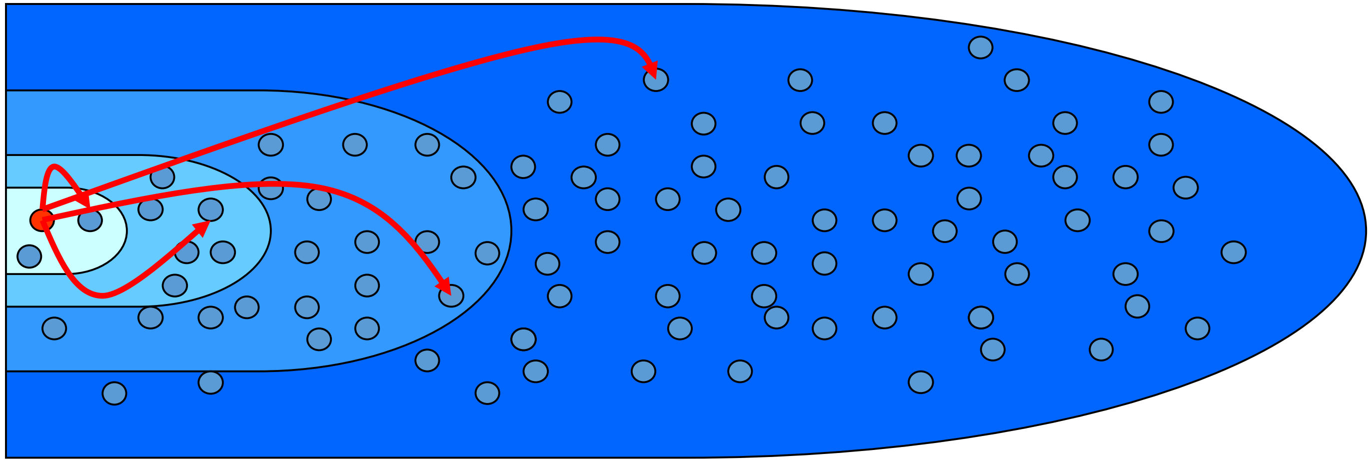
Traditional DHTs and Kleinberg model

- Most of the structured P2P systems are similar to Kleinberg's model and are called logarithmic-like approaches. E.g.
  - Chord (randomized version) q=logN, r=1
  - **When q is O(logN), then "ring-link" is expected to appear from long-range link process.**

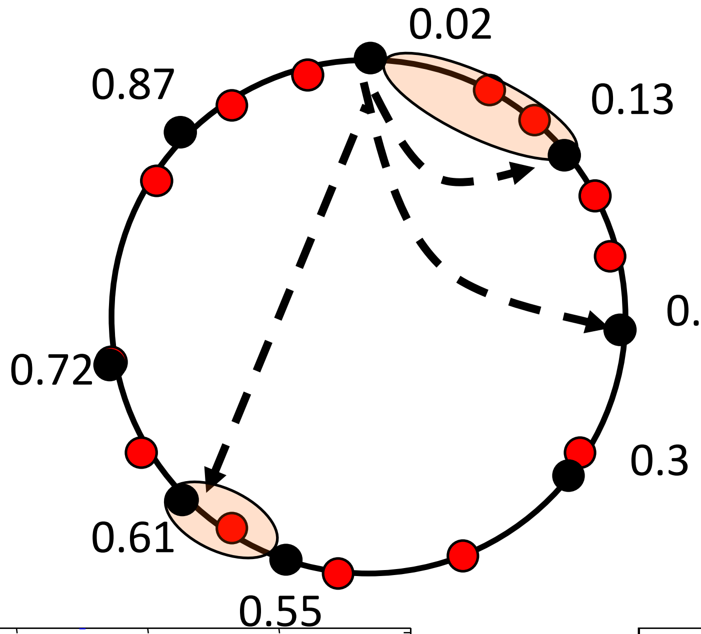
- Randomized Chord's model



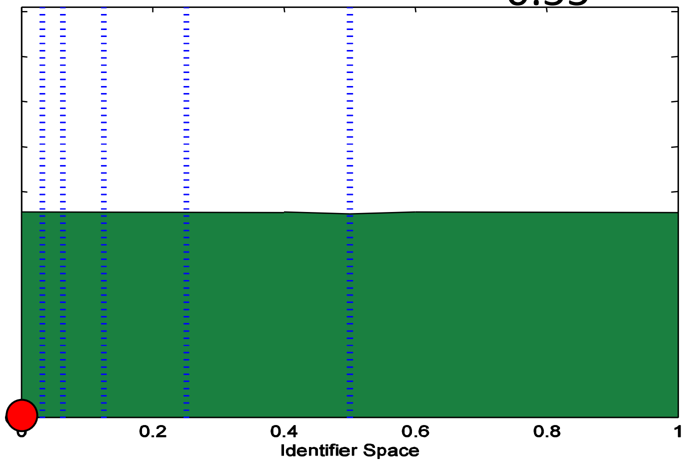
- Kleinberg's model

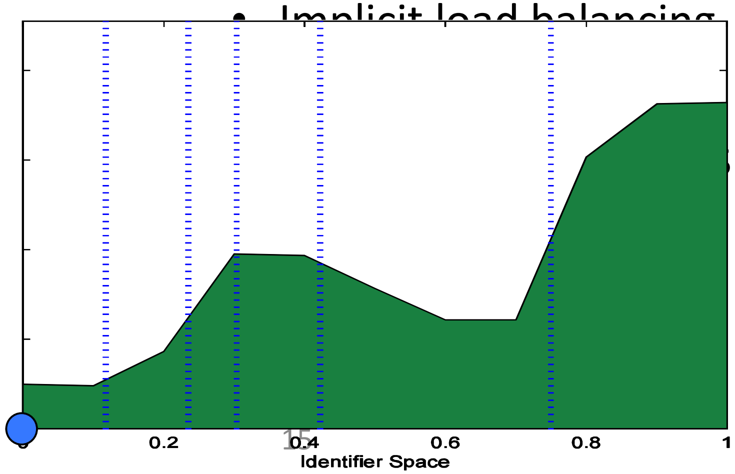


Data on the Overlay



- **Peers** mapped on the ring
  - Uniform hash function (e.g., SHA-1)
- **Resources** mapped on the ring
  - Uniform hash function
- **Connectivity establishment**
  - Based on Kleinbergian Principles
  - E.g., Chord, Symphony etc.
- **Uniform** peer key (id) distribution
  - Implicit load balancing

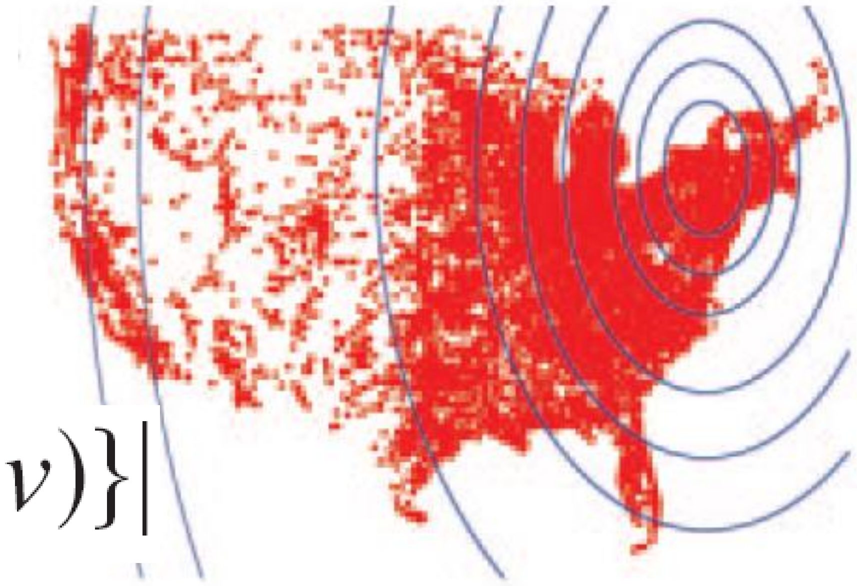




are distributed non-

in real world networks?

Liben-Nowell et a. “Geographic routing in social networks” 2005



- LiveJournal Data
  - Bloggers + Zip Codes

$$rank_u(v) := \left|\{w : d(u,w) < d(u,v)\}\right|$$

- $P(u \rightarrow v) = rank_u(v)^{-\alpha}$
- **What is best α?**
  - For equally spaced pairs: α = dim. of the space
  - In this special case $\alpha = 1$ is best for search

Basic Navigation Principles

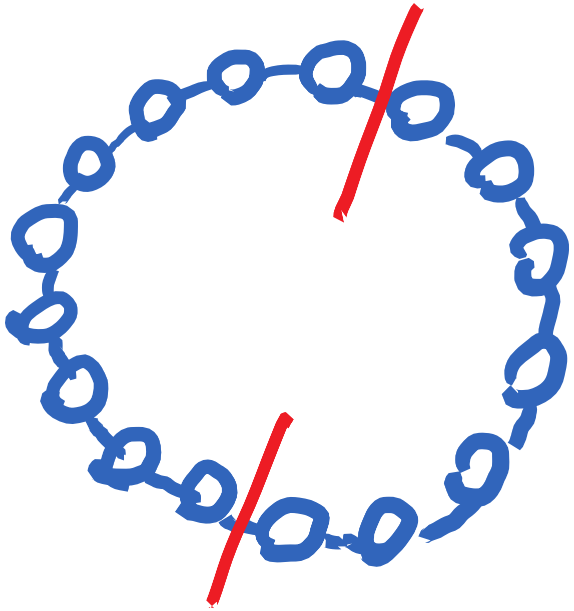
- So **why** can we navigate in the network and **find short paths** without **any global view** of the system?
  - We have **globaly agreed ID space**, with a distance function –>
    - Allows us to make local decisions on minimizing distance to the target
  - (Implicit) Existinace of the **underlying lattice** (i.e., ring) ->
    - Assures us to always be able to make progress navigating towards the target, i.e.,
    - Target will always be reached!
  - Existance of **Kleinbergian long-range links** ->
    - Allows us to progres towards the target rapidly (in polylog steps)
    - Why is it so in ”real-world” it is still an open question...

Recap: Network properties

- Each network is unique "***microscopically***", but in large scale networks one can observe ***macroscopic*** properties:
  - **Diameter** (six-degrees of separation);
  - **Clustering coefficient** (triangles, friends-of-friends are also friends);
  - **Degree distribution** (are there many "hubs" in the network?);
  - **Largest Component** (how many "islands" and how big?)
  - **Navigability** (can you do efficient routing?=
- Network models:
  - Erdos-Renyi
  - Configuration
  - Preferential attachment
  - Watts-Strogatz
  - Kleinberg

Expanders

- ***Expanders are graphs with very strong connectivity properties***.
  - sparse yet very well-connected
- Example
  - N nodes, E edges N=E
  - ***Does this graph have good connectivity properties?***
  - 2 edges fail -> isolates up to N/2 nodes



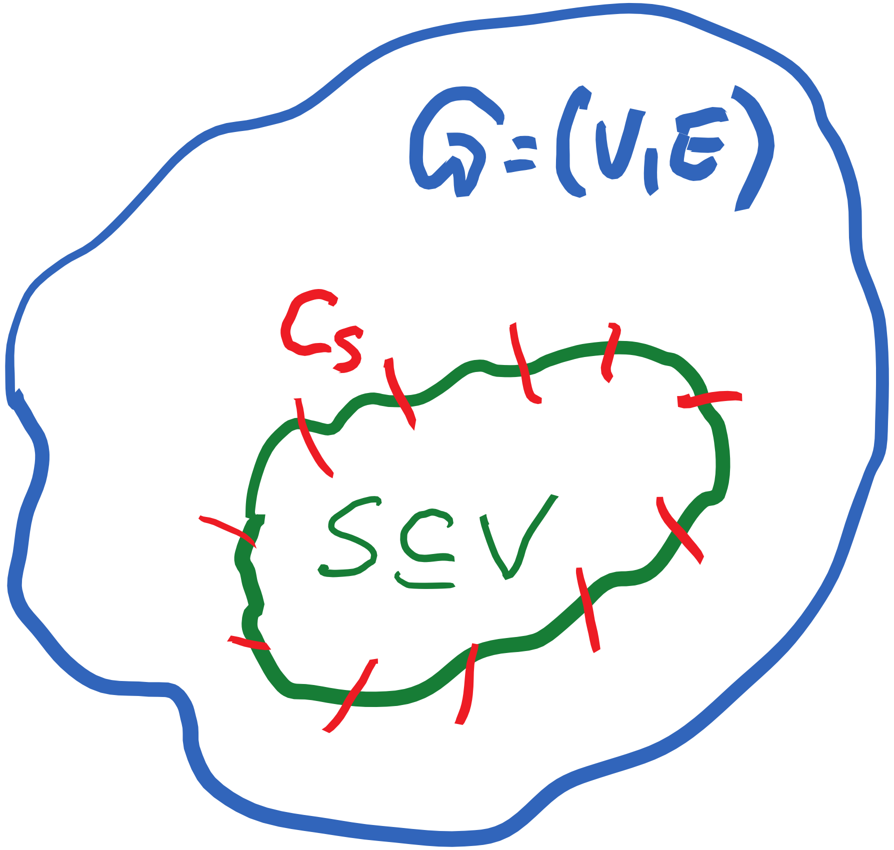
Can one isolate large number of nodes by removing small number of edges?

Expansion

- **Expansion *α*:**

$$\alpha = \min_{S \subseteq V} \frac{c_s}{\min(n_s, n - n_s)}$$

- How robust are your graphs?
- ***To isolate* k *nodes one needs to remove at least* α*k *edges***

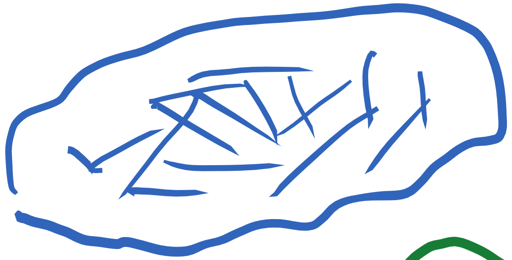


$|V| = n \quad |S| = n_s$

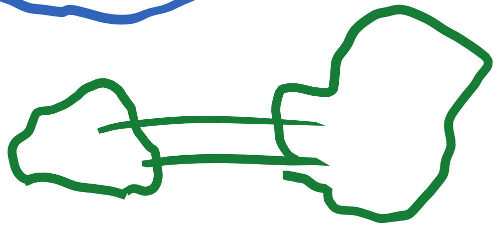
Examples

- Which networks do you think have good expansion (random graph, tree, grid)?

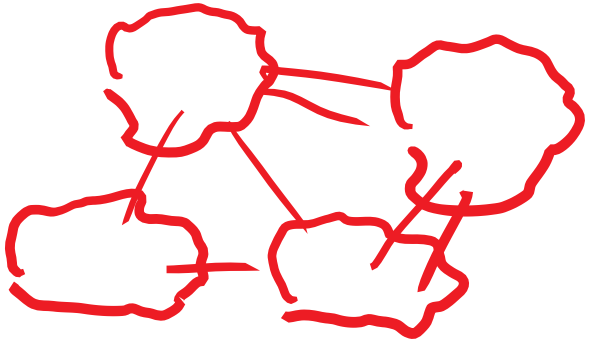
High expansion



Low expansion



Social networks

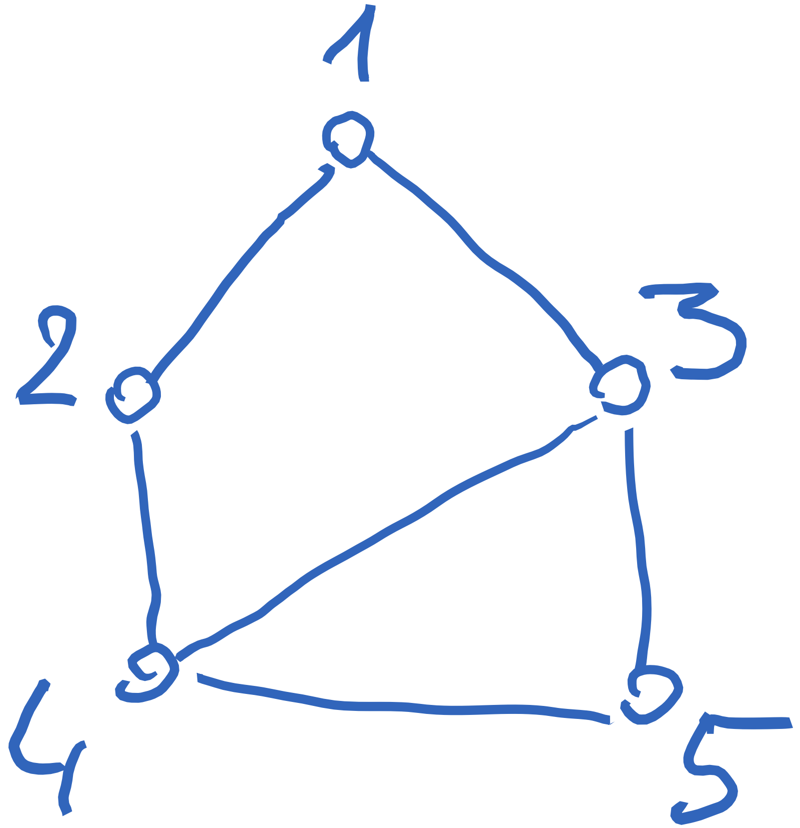


Properties of Expander Graphs

- A graph **is an expander** if the number of edges originating from every subset of vertices is larger than the number of vertices at least by a constant factor (more than 1).
    - Sparse yet very well-connected (no small cuts, no bottlenecks)
    - Small second eigenvalue λ_2(we will talk about it later)
    - Rapid convergence of random walk
- For all practical reasons, random walk of TTL = O(logN) on an expander graph with fixed node degree gives an uniform random node from the population

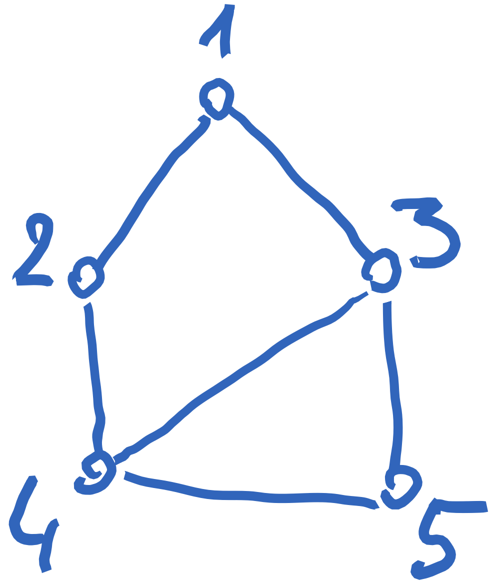
Random Walks on Graphs

Random Walks



- Let G(V,E) be connected graph.
- Consider random walk on G from node v.
  - We move to a neighboring node with probability 1/d(v)
  - The sequence of random walks is Markov chain

Random Walks



Time	Node 1	Node 2	Node 3	Node 4	Node 5
0	1	0	0	0	0
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

- The initial node can be fixed, but also can be drawn from some initial distribution P_0

Convergence?

- For any ***connected non-bipartite*** bidirectional graph, and any starting point, the random walk converges
  - Converges to **unique** stationary distribution
  - **Power Iteration**

t	1	0	0	0	0
1	0.00	0.50	0.50	0.00	0.00
2	0.42	0.00	0.00	0.42	0.17
3	0.00	0.35	0.43	0.08	0.14
4	0.32	0.03	0.10	0.39	0.17
5	0.05	0.29	0.37	0.13	0.16
6	0.27	0.07	0.15	0.35	0.17
7	0.08	0.25	0.33	0.17	0.17
8	0.24	0.10	0.18	0.32	0.17
9	0.11	0.22	0.31	0.19	0.17
10	0.22	0.12	0.20	0.30	0.17
11	0.13	0.21	0.29	0.21	0.17
12	0.20	0.13	0.22	0.28	0.17
13	0.14	0.19	0.28	0.22	0.17
14	0.19	0.14	0.23	0.27	0.17
15	0.15	0.19	0.27	0.23	0.17
16	0.18	0.15	0.23	0.27	0.17
17	0.15	0.18	0.26	0.24	0.17
18	0.18	0.16	0.24	0.26	0.17
19	0.16	0.18	0.26	0.24	0.17
20	0.17	0.16	0.24	0.26	0.17
21	0.16	0.17	0.26	0.24	0.17
22	0.17	0.16	0.24	0.26	0.17
23	0.16	0.17	0.25	0.25	0.17
24	0.17	0.16	0.25	0.25	0.17
25	0.16	0.17	0.25	0.25	0.17
26	0.17	0.16	0.25	0.25	0.17
27	0.16	0.17	0.25	0.25	0.17
28	0.17	0.16	0.25	0.25	0.17
29	0.17	0.17	0.25	0.25	0.17
30	0.17	0.17	0.25	0.25	0.17

Stationary Distribution

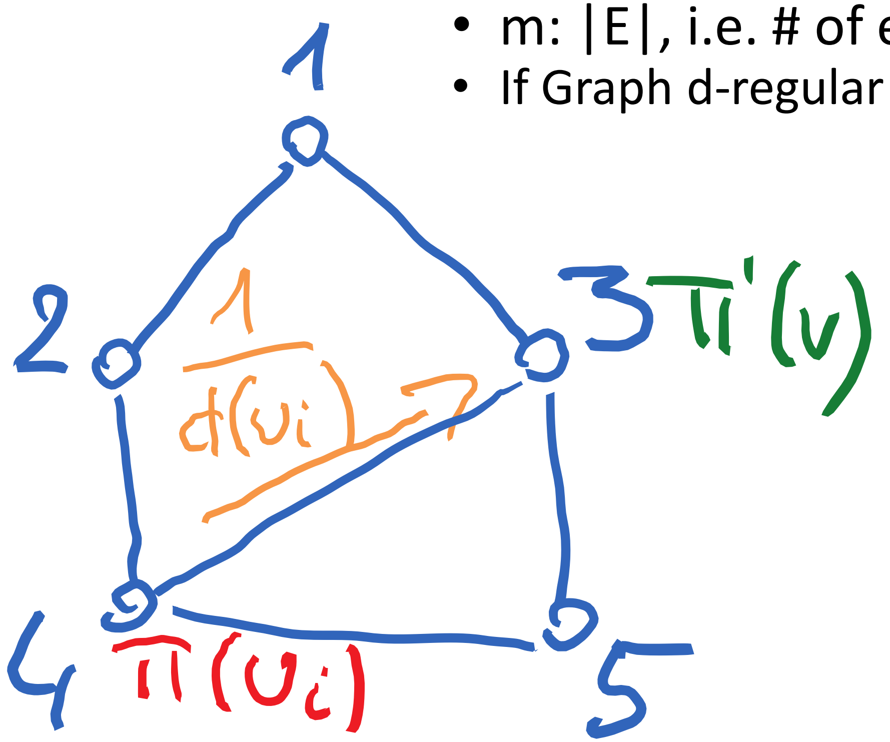
- Which distribution does the random walk converge in our graph?

π	0.17	0.17	0.25	0.25	0.17
π'	0.17	0.17	0.25	0.25	0.17

- Random walk converges to the stationary distribution:

$$\pi(v) = d(v)/2m$$

  - $d(v)$ = degree of v, i.e. # of neighbors.
  - m: $|E|$, i.e. # of edges.
  - If Graph d-regular then to uniform distribution



$$\pi'(v) = \sum_{u:\,(u,v)\in E} \pi(u) \frac{1}{d(u)}$$

$$= \sum_{u:\,(u,v)\in E} \frac{d(u)}{2m} \cdot \frac{1}{d(u)}$$

$$= \sum_{u:\,(u,v)\in E} \frac{1}{2m}$$

$$= \frac{d(v)}{2m}$$

$$= \pi(v)$$

Implications

- The stationary distribution
  $$\pi(v) = d(v)/2m$$
  is **proportional to the degree of v**.
  - What's the intuition?
  - The more neighbors you have, the more chance you'll be visited.
    - We'll talk about it later in ID2222 Data Mining course

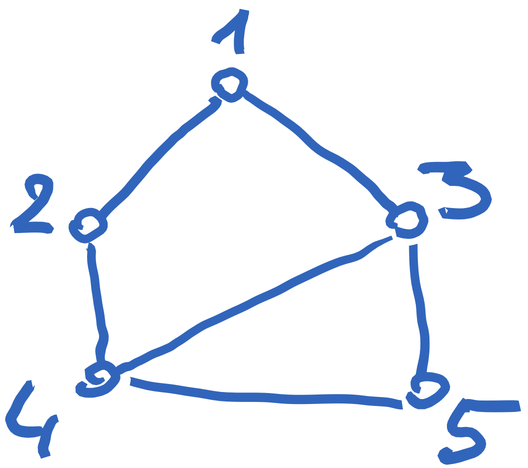
Definitions

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency Matrix

$$D = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \end{pmatrix}$$

Diagonal matrix with $D_{i,i} = 1/d(i)$



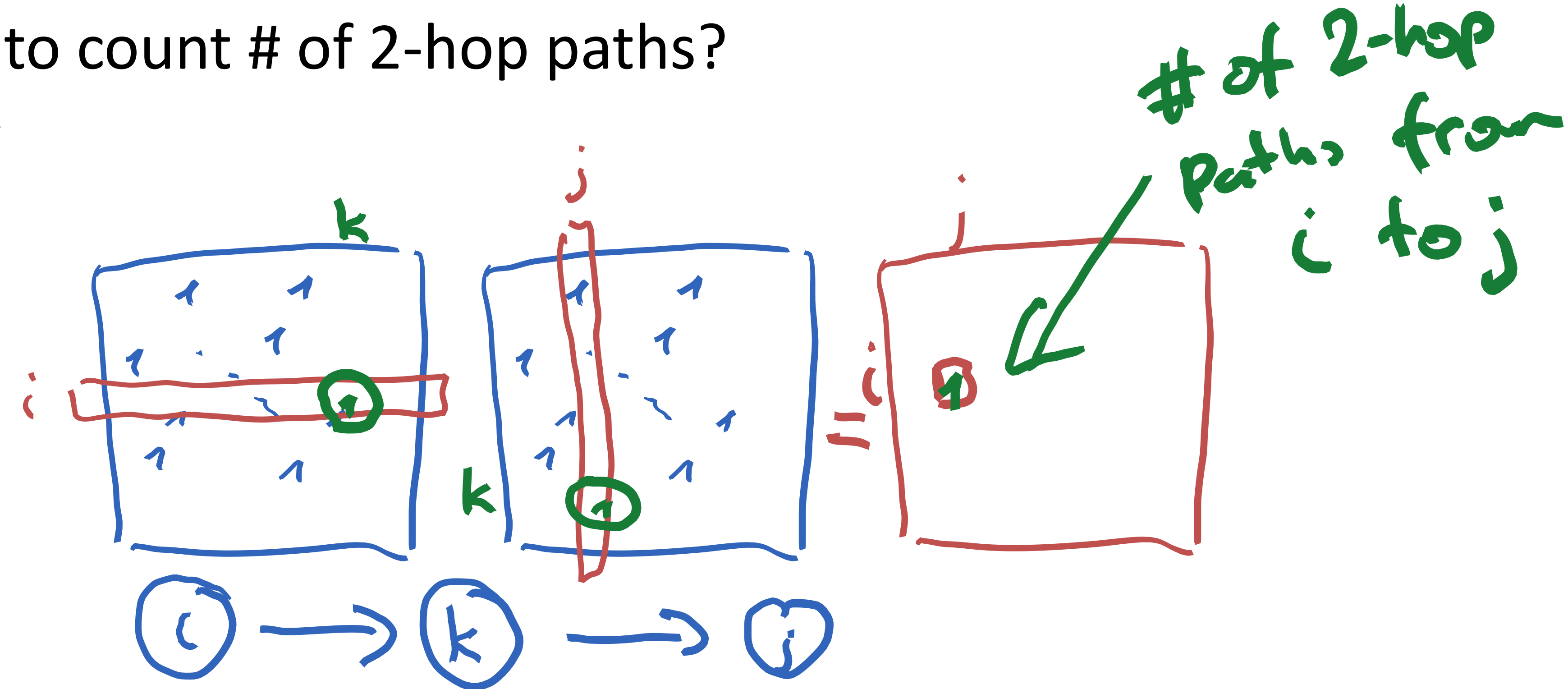
$$M = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

Transition (random walk) Matrix

M=DA

Adjacency Matrix

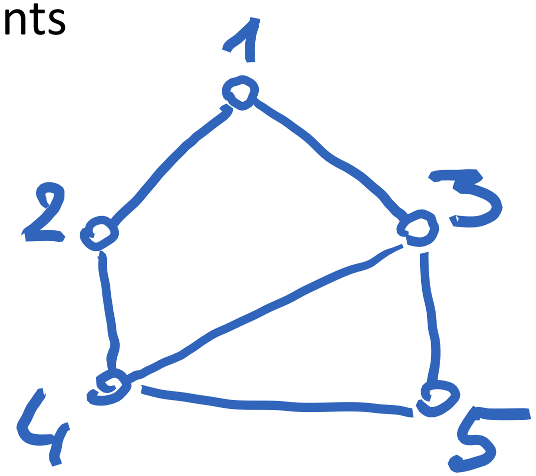
- A is *n x n* adjacency matrix of G=(V,E)
  - A_{ij} is 1 if there is a link between i and j nodes and 0 otherwise
- Gives us all 1-hop paths.
- How to count # of 2-hop paths?
  - A^2



Matrix manipulations

- A^2 gives us # of 2-hop paths
- A^3 gives us ?
  - # of 3-hop paths, etc.
    - Technically it is not a path, but a walk (not simple paths, you can visit the same node several times).
- What about taking a vector v=(1 0 0 0 0) that represents a message at the first node and multiplying it by A?

$$vA = \begin{pmatrix}1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0 & 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 1\\ 0 & 1 & 1 & 0 & 1\\ 0 & 0 & 1 & 1 & 0\end{pmatrix}$$



- vA=(0 1 1 0 0)
  - indicates how many walks of length 1 from node 1 end up in node i.
- vA^2 = (2 0 0 2 1)
  - Indicates how many walks of length 2 from node 1 end up in node i.
- vA^3 = (0 4 5 1 2)
  - Indicates how many walks of length 3 from node 1 end up in node i.

Matrix manipulations (cont.)

- What about multiplying by a Random Walk Matrix?

$$M=\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

Transition (random walk) Matrix

M=DA

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}=\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}=\begin{pmatrix} 0.42 & 0 & 0 & 0.42 & 0.17 \end{pmatrix}$$

What about Random Walk Matrix M? (cont.)

- Recap: Random walk on a graph G: we start at a node v_0 and at the t-th step we are at a node v_t. We move to a neighbor of v_t with probability $1/d(v_t)$.
  - The sequence of random nodes (v_t :t=0,1,2…) is a Markov chain
- We start from the initial state of the system, e.g. P_0: [1 0 0 0 0];
  - Can also be drawn from some initial distribution
- **$P_t = P_0 M^t$**
  - Or can be written as $P_t = (M^T)^t P_0$ if we represent P as a column vector

Again the same Example

$\mathbf{P_t = P_0 M^t}$

$$(1 \quad 0 \quad 0 \quad 0 \quad 0)\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} = (0 \quad 1/2 \quad 1/2 \quad 0 \quad 0)$$

$$(0 \quad 1/2 \quad 1/2 \quad 0 \quad 0)\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} = (0.42 \quad 0 \quad 0 \quad 0.42 \quad 0.17)$$

- When $P_{t+1} = P_t = \pi$, we have reached stationary distribution, i.e. $\boldsymbol{\pi M = \pi}$
- Recall: that v is **eigenvector** of matrix M and λ its eigenvalue if $\mathbf{vM = \lambda v}$
  - so ***π is eigenvector of M with eigenvalue λ=1***