Winter School on Complex Networks

Navigation

28 January 2016
Navigation

- In Small world nets there are short paths $O((\log(N))^a)$

- But can we find them?
  - Milgram’s experiment suggests nodes can find them using only local information
  - Standard routing algorithms require $O(N)$ information
Kleinberg’s result

- Model: Each node has
  - Short-range connections
  - 1 long-range connection, up to distance $r$ with probability prop. to $r^{-\alpha}$
  - For $\alpha=0$ it is similar to Watts-Strogatz model: there are short-paths
Kleinberg’s result

- If $\alpha=2$ the greedy algorithm (forward the packet to the neighbor with position closest to the destination) achieves avg path length $O((\log(N))^2)$
Kleinberg's result

- If $\alpha<2$ no local information algorithm can take advantage of small world properties
  - avg path length $\Omega(N^{\beta/2})$
    - where $\beta=(2-\alpha)/3$ for $0\leq\alpha\leq2$
    - $\beta=(\alpha-2)/(\alpha-1)$, for $\alpha>2$
Conclusions

- The larger $\alpha$ the less distant long-range contacts move the message, but the more nodes can take advantage of their “geographic structure”
- $\alpha=2$ achieved the best trade-off
Configuration model

- A family of random graphs with given degree distribution
Configuration model

- A family of random graphs with given degree distribution
  - Uniform random matching of stubs
Configuration model

- A family of random graphs with given degree distribution
  - Uniform random matching of stubs
Back to Navigation: Random Walks

- What can we do in networks without a geographical structure?
  - Random walks
Back to Navigation: Random Walks

- How much time is needed in order to reach a given node?
Random Walks: stationary distribution

- \( \pi_i = \sum_{j \in N_i} \frac{1}{k_j} \pi_j \)
- \( \pi_i = \frac{k_i}{\sum_{i=1}^{N} k_j} = \frac{k_i}{2M} \)

- Avg time to come back to node i starting from node i: \( \frac{1}{\pi_i k_i} = \frac{2M}{k_i} \)

- Avg time to reach node i
  - intuitively \( \approx \Theta(M/k_i) \)
Another justification

Random walk as random edge sampling

- Prob. to pick an edge (and a direction) leading to a node of degree $k$ is $\frac{kp_k}{\langle k \rangle}$

- Prob. to arrive to a given node of degree $k$:

  $\frac{kp_k}{p_kN \langle k \rangle} = \frac{k}{2M}$

- Avg. time to arrive to this node $2M/k$

...equivalent to a RW where at each step we sample a configuration model
Distributed navigation  
(speed up random walks)

- Every node knows its neighbors
Distributed navigation (speed up random walks)

- Every node knows its neighbors
- If a random walk looking for \( i \) arrives in \( a \) the message is directly forwarded to \( i \)

\{a,b,c,d\}
Distributed navigation reasoning 1

- We discover \( i \) when we sample one of the links of \( i \)'s neighbors.
- Avg # of these links: \( k_i \sum_k (k-1) \frac{kp_k}{<k>} = k_i \left( \frac{<k^2>}{<k>} - 1 \right) \)
- Prob. to arrive at one of them: \( \frac{k_i}{2M} \left( \frac{<k^2>}{<k>} - 1 \right) \)
Distributed navigation reasoning 2

- Prob that a node of degree \( k \) is neighbor of node \( i \) given that RW arrives to this node from a node different from \( i \):

\[
1 - \left(1 - \frac{k_i}{2M}\right)^{k-1} \approx \frac{k_i(k-1)}{2M}
\]

- Prob that the next edge brings to a node that is neighbor of node \( i \):

\[
\sum_k \frac{k_i(k-1)}{2M} \frac{kp_k}{<k>} = \frac{k_i}{2M} \left(\frac{<k^2>}{<k>} - 1\right)
\]
Distributed navigation

- **Avg. Hop#**
  \[
  \frac{2M}{k_i} \frac{< k >}{< k^2 > - < k >}
  \]

- Regular graph with degree \( d \):
  \[
  \frac{2M}{d(d-1)}
  \]

- ER with \( <k> \):
  \[
  \frac{2M}{k_i(<k>-1)}
  \]

- Pareto distribution
  \[
  P(k) \approx \frac{\alpha x_m^\alpha}{x^{\alpha+1}}
  \]
  \[
  \approx \frac{2M}{k_i} \frac{\alpha-2)(\alpha-1)}{x_m-(\alpha-2)(\alpha-1)}
  \]

  If \( \alpha \to 2 \ldots \)
Distributed navigation

- Application example:
  - File search in unstructured P2P networks through RWs