Navigation and Propagation in Networks

Michael Goodrich

Some slides adapted from slides by Jean Vaucher, Panayiotis Tsaparas, Jure Leskovec, and Christos Faloutsos
Review:
Milgram’s experiment

Instructions:
Given a target individual (stockbroker in Boston), pass the message to a person you correspond with who is “closest” to the target.
Small world phenomenon: Milgram’s experiment

Outcome:
average chain length was between 5 and 6

“Six degrees of separation”
Small world phenomenon:
Milgram’s experiment repeated

email experiment

• 18 targets
• 13 different countries

• 60,000+ participants
• 24,163 message chains
• 384 reached their targets
• average path length 4.0

Milgram’s experiment revisited

- What did Milgram’s experiment show?
  - (a) There are short paths in large networks that connect individuals
  - (b) People are able to find these short paths using a simple, greedy, decentralized algorithm

- Small world models take care of (a)
- Kleinberg: what about (b)?
Kleinberg’s model

- Consider a directed 2-dimensional lattice
- For each vertex $u$ add $q$ shortcuts
  - choose vertex $v$ as the destination of the shortcut with probability proportional to $[d(u,v)]^{-r}$
  - when $r = 0$, we have uniform probabilities
Searching in a small world

- Given a source $s$ and a destination $t$, define a greedy local search algorithm that
  1. knows the positions of the nodes on the grid
  2. knows the neighbors and shortcuts of the current node
  3. knows the neighbors and shortcuts of all nodes seen so far
  4. operates greedily, each time moving as close to $t$ as possible

- Kleinberg proved the following
  - When $r=2$, an algorithm that uses only local information at each node (not 2) can reach the destination in expected time $O(\log^2 n)$.
  - When $r<2$, a local greedy algorithm (1-4) needs expected time $\Omega(n^{(2-r)/3})$.
  - When $r>2$, a local greedy algorithm (1-4) needs expected time $\Omega(n^{(r-2)/(r-1)})$. 
Searching in a small world

- For $r < 2$, the graph has paths of logarithmic length (small world), but a greedy algorithm cannot find them.
- For $r > 2$, the graph does not have short paths.
- For $r = 2$ is the only case where there are short paths, and the greedy algorithm is able to find them.
When $r=0$, links are randomly distributed, $\text{ASP} \sim \log(n)$, $n$ size of grid
When $r=0$, any decentralized algorithm is at least $a_0n^{2/3}$

When $r<2$, expected time at least $\alpha_r n^{(2-r)/3}$
Overly localized links on a lattice

When $r > 2$ expected search time $\sim N^{(r-2)/(r-1)}$

$p \sim \frac{1}{d^4}$
When $r=2$, expected time of a greedy search is at most $C (\log N)^2$.

A geographical small world model links balanced between long and short range.

$p \sim \frac{1}{d^2}$
Extensions

- If there are log n shortcuts, then the search time is $O(\log n)$
  - we save the time required for finding the shortcut

- If we know the shortcuts of log n neighbors the time becomes $O(\log^{1+1/d} n)$
Neighborhood Preferential Attachment (NPA) Model

- KL model: geographic distance ↔ friendship
- BA model: popularity (degree) ↔ friendship

**NPA model**: combines the KL and BA models.

Given road network $G = (V, E)$,

- Start with clique $M$ of size $m + 1$ from $V$
- For each $u \in V \setminus M$, add $m$ random long-range edges where $P(u \rightarrow v) \propto \frac{\text{deg}(v)}{d(u, v)^s}$
- After obtaining edge set $E'$, return $G = (V, E \cup E')$

- Typically choose $s = 2$, following Kleinberg’s proof

Experimental Analysis: Setup

- Datasets: road networks for 50 U.S. states and Washington, D.C.
- Multi-state experiments: combined road networks of multiple states
- Instances of KL, BA and NPA models constructed for each dataset
- Weighted decentralized algorithm ran on 1000 randomly chosen source/target pairs in each dataset
- Only largest connected components considered
- Sizes of road networks range from ~9K to ~2M vertices
Experiment: Hop Counts with Few Long-Range Links

- **m=1**
- **m=2**
- **m=3**
- **m=4**
Experiment: Dropouts

- Original small world experiment: roughly 20% of participants drop out of experiment (e.g. no one to forward package to), at each step of routing
- Longer source-to-target path message more likely to be dropped
- Expect average hop length for successful routes to decrease as increases
Experiment: Getting to Six Degrees of Separation

- Set $m=30$, e.g. assume people have access to an address book when deciding where to forward package
- Can achieve 6 degrees of separation with the NPA model for the majority of road networks, and between 6-7 hops for the largest states when dropout probability $p = 0.2$
Small Worlds
& Epidemic diseases

The SIR Model

- Nodes are living entities
- Link is contact
- 3 States:
  - **Susceptible** (Uninfected and not immune)
  - **Infected**
  - **Recovered** (or dead) and immune
Diffusion in Social Networks

- One of the networks is a spread of a disease, the other one is product recommendations
- Which is which?
Diffusion in Social Networks

- A fundamental process in social networks: Behaviors that cascade from node to node like an epidemic
  - News, opinions, rumors, fads, urban legends, …
  - Word-of-mouth effects in marketing: rise of new websites, free web based services
  - Virus, disease propagation
  - Change in social priorities: smoking, recycling
  - Saturation news coverage: topic diffusion among bloggers
  - Internet-energized political campaigns
  - Cascading failures in financial markets
  - Localized effects: riots, people walking out of a lecture
Failures in networks

- Fault propagation or viruses
- Scale-free networks are far more resistant to random failures than ordinary random networks
  - because of most nodes are leaves
- But failure of hubs can be catastrophic vulnerable or targets of deliberate attacks
  - which may make scale-free networks more vulnerable to deliberate attacks
- Cascades of failures
Effect of peers & pundits (hubs and authorities)

- People’s decisions are affected by what others do and think
  - Pressure to conform?
- Efficient strategy when insufficient knowledge or expertise
  - Ex: picking a restaurant
- Google’s PageRank is a score for influential nodes in a network (the WWW)
PageRank is a link analysis algorithm which assigns a numerical weighting to each Web page, with the purpose of "measuring" relative importance.

- Based on the hyperlinks map
- An excellent way to prioritize the results of web keyword searches
Simplified PageRank algorithm

- Assume four web pages: A, B, C, and D. Let each page begin with an estimated PageRank of 0.25.

  \[ PR(A) = PR(B) + PR(C) + PR(D). \]

  \[ PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}. \]

- L(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

  \[ PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}. \]
PageRank algorithm including damping factor

- Assume page A has pages B, C, D ..., which point to it. The parameter d is a damping factor which can be set between 0 and 1. Usually set d to 0.85. The PageRank of a page A is given as follows:

\[ PR(A) = 1 - d + d \left( \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots \right) \]
Intuitive Justification

- A "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back", but eventually gets bored and starts on another random page.
  - The probability that the random surfer visits a page is its PageRank.
  - The damping factor is the probability at each page the "random surfer" will get bored and request another random page.
- A page can have a high PageRank
  - If there are many pages that point to it
  - Or if there are some pages that point to it, and have a high PageRank.