COMMUNITY DETECTION & RANDOM WALK

Nguyen Truong Duy
Outline

- Random Walk
- Walktrap Algorithm
- Markov Clustering Algorithm
Random Walk on Graphs

- At each time step, a walker is on a vertex. Then he chooses the next vertex randomly and uniformly among its neighbors.
Random Walk on Graphs

- Hitting time: Expected number of steps in a random walk to go from $u$, and ends at $v$ for the first time

- Commute time: Sum of Hitting time from $u$ to $v$, and from $v$ to $u$
Why Random Walk?

- Use random walk to define how community is formed.

- Particularly, use random walk in defining measures of vertex similarity
Why Random Walk?

- Capture well and intuitively the community structure.
- Can be computed efficiently.
Walktrap

- By Pascal Pons and Matthieu Latapy
- Implemented in iGraph library.
- Claim:
  - In real-world cases: time $O(V^2 \log V)$
  - Worst case: time $O(E V^2)$
  - Space $O(V^2)$
Walktrap

- Intuition: random walkers tends to get ‘trapped’ into densely connected parts (communities)
Walktrap

- Consider the probability $P^t$ to go from vertex $i$ to $j$ in $t$ steps.
- Establish a distance measure between vertices (and between clusters) based on $P^t$.
- Use clustering algorithm (Ward’s method) to find communities.
Markov Clustering Algorithm

- By van Dongen
- Simple to implement.
- Time complexity: $O(V^3)$
Markov Clustering Algorithm

- Intuition: More or less same as Walktrap
Markov Clustering Algorithm

- Iterative clustering algorithm.
- Two main operations: expansion and inflation
Markov Clustering Algorithm

- Start with the initial transfer matrix
- Expansion: Calculate the probability of a random walker going between vertices in \( t \) steps (by matrix multiplication)
- Inflation: Highlight probabilities of intra-cluster walks and demote inter-cluster walks
Markov Clustering Algorithm

- Converges to a matrix 0-1 matrix $M$ (like an adjacency matrix).
- $M$ represents a disconnected graph. Each connected component is a community.
Work to Do

- Do testing and benchmarking for the two algorithms
References

Pascal Pons, Matthieu Latapy, Computing communities in large networks using random walks

http://micans.org/mcl/