Random walks and diffusion on networks

Naoki Masuda

Department of Engineering Mathematics
University of Bristol, UK

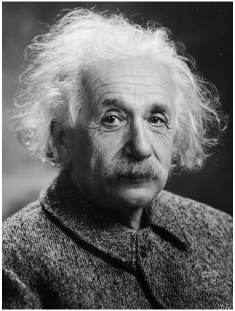
naoki.masuda@bristol.ac.uk
www.naokimasuda.net

Talk plan

- Random walks and diffusion
- 1-dim
- Networks
- Applications of RWs on networks

Diffusion is a broad term

- Diffusion of innovation
- Biological diffusion



Background

- Gamblers ruin: Pascal, Fermat, Huygens, Bernoulli,
- Term random walk was coined by Karl Pearson (1905).
- Brownian motion of particles colliding with atoms and molecules (Einstein, 1905)
- Applications: locomotion and foraging of animals, neuronal firing, decision-making in the brain, population genetics, polymer chains, financial markets, sports statistics
- Relevant theoretical research areas: probability theory (in maths), CS, statistical physics, OR,

Why do we study RWs on networks?

- Theoretical interest
- Applications. Especially, RWs lie at the core of many algorithms
  - PageRank
  - community detection
  - core-periphery structure
  - diffusion map
  - respondent-driven sampling
  - consensus algorithms

1-dim

- Assume the continuous space ($\mathbb{R}^1$)
  - Discrete-space case is similar
- Assume discrete-time RW
- In each time step, a walker at x moves to the interval $[x + r, x + r + \Delta r]$ with probability $f(r)\Delta r$. Then, the master equation is given by

$$p(x; n) = \int_{-\infty}^{\infty} f(x - x')p(x'; n - 1)\mathrm{d}x' \,.$$

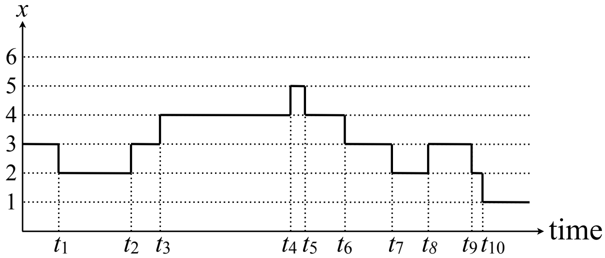
- As time n tends to ∞,

$$p(x; n) = \frac{1}{(2\pi Dn)^{1/2}} e^{-\frac{(x-vn)^2}{4Dn}} \,,$$

where $v \equiv \langle r \rangle$ and $D \equiv \langle (r - \langle r \rangle)^2 \rangle / 2$.

Continuous-time RW

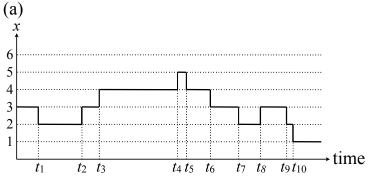
(a)



(b)

steps (n)	0	1	2	3	4	5	6	7	8	9	10
x	3	2	3	4	5	4	3	2	3	2	1

(This fig assumes a discrete space)



(b)

steps (n)	0	1	2	3	4	5	6	7	8	9	10
x	3	2	3	4	5	4	3	2	3	2	1

$$p(x;t) = \sum_{n=0}^{\infty} p(x;n)p(n,t)$$

where x: position, t: time, n: number of moves made

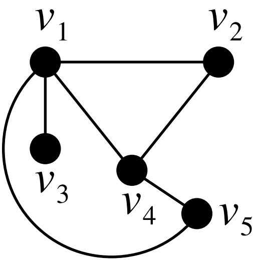
$$\langle n \rangle = \frac{t}{\langle \tau \rangle}, \quad \text{where } \tau \text{ is inter-event time}$$

E.g., $\psi(\tau) = \beta e^{-\beta\tau} \rightarrow p(n,t) = \frac{(\beta t)^n}{n!} e^{-\beta t}$

✓Full soln in terms of the Laplace and Fourier transforms

RWs on networks

Adjacency matrix

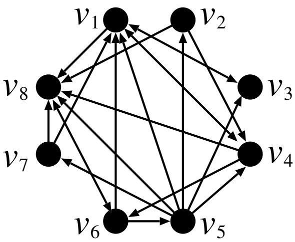


$$\Longleftrightarrow \quad A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- We allow directed and weighted adjacency matrices.

Transition-probability matrix (of DTRW)

- $i \to j$ with probability $T_{ij} = \frac{A_{ij}}{s_i^{\text{out}}}$
  where $s_i^{\text{out}} = \sum_{j=1}^{N} A_{ij}$ is the out-strength of node i.



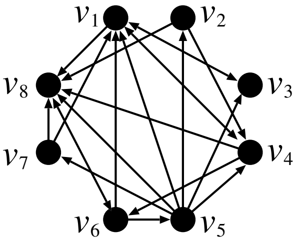
$$A = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- Directed networks allowed
- Weighted networks allowed

Q: Which complicates things more than the other?

Transition-probability matrix (of DTRW)

- $i \to j$ with probability $T_{ij} = \frac{A_{ij}}{s_i^{\text{out}}}$
  where $s_i^{\text{out}} = \sum_{j=1}^{N} A_{ij}$ is the out-strength of node i.



$$B = \begin{pmatrix} 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- Directed networks allowed
- Weighted networks allowed

Q: Which complicates things more than the other?

- Conservation: $\sum_{j=1}^{N} T_{ij} = 1$
- Dynamics: $p_j(n+1) = \sum_{i=1}^{N} p_i(n) T_{ij} \quad (j \in \{1, \ldots, N\})$

Note that $\sum_{i=1}^{N} p_i(n) = 1$

- With vectors, $\boldsymbol{p}(n+1) = \boldsymbol{p}(n) T$
  where $\boldsymbol{p}(t) = (p_1(n), \ldots, p_N(n))$
- $\boldsymbol{p}(n) = \boldsymbol{p}(0) T^n$. That's it.

  Q: linear or nonlinear?

Two main Qs for RWs

- Both in theory and applications, we are mostly concerned with either (or both) of the following questions
  - Stationary density (infinite-time behaviour)
  - Relaxation time (asymptotic behaviour)
  - First-passage time (finite-time behaviour)
- Other important questions:
  - Flow
  - Other finite-time behaviour (esp. in community detection)
  - (Loss of) detailed balance

Two main Qs for RWs

- Both in theory and applications, we are mostly concerned with either (or both) of the following questions
  - Stationary density (infinite-time behaviour)
  - Relaxation time (asymptotic behaviour)
  - First-passage time (finite-time behaviour)
- Other important questions:
  - Flow
  - Other finite-time behaviour (esp. in community detection)
  - (Loss of) detailed balance

Stationary density of DTRWs

$$\boldsymbol{p}(n+1) = \boldsymbol{p}(n)T$$

$$\boldsymbol{p}(n) = \boldsymbol{p}(0)T^n$$

$$\text{Let } p_i^* = \lim_{n\to\infty} p_i(n)$$

$$\boldsymbol{p}^* = \boldsymbol{p}^* T$$

The stationary density is the left eigenvector of T with eigenvalue 1.
The corresponding right eigenvector is: $(1, \ldots, 1)^\top$.
Relaxation: $\boldsymbol{p}(n) \approx \boldsymbol{p}^* + \lambda_2^n \boldsymbol{u}_2^{\mathrm{L}} \langle \boldsymbol{p}(0), \boldsymbol{u}_2^{\mathrm{R}} \rangle$
where λ_2 is the 2nd largest eigenvalue (in modulus) of T

Stationary density for undirected networks

Central result: $p_i^* = \frac{s_i}{\sum_{\ell=1}^{N} s_\ell} \quad (i \in \{1, \ldots, N\})$

where $s_i = \sum_{j=1}^{N} A_{ij} = \sum_{j=1}^{N} A_{ji}$ is the strength (i.e., weighted degree) of node i

Proof: Use $\boldsymbol{p}^* = \boldsymbol{p}^* T$, or $p_i^* = \sum p_j^* T_{ji}$ (for each i).

What does $p_i^* = \frac{s_i}{\sum_{\ell=1}^{N} s_\ell}$ mean?

Stationary density for directed networks

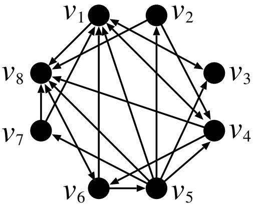
- No such handy results.
- First-order approximation:

$$p_i^* = \sum_{j=1}^{N} p_j^* \frac{A_{ji}}{s_j^{\text{out}}} \approx (\text{const}) \times \sum_{j=1}^{N} A_{ji} \propto s_i^{\text{in}}$$

where $s_i^{\text{in}} = \sum_{j=1}^{N} A_{ji}$ is the in-strength of node i.

  - Accurate in many cases, and inaccurate in many other cases

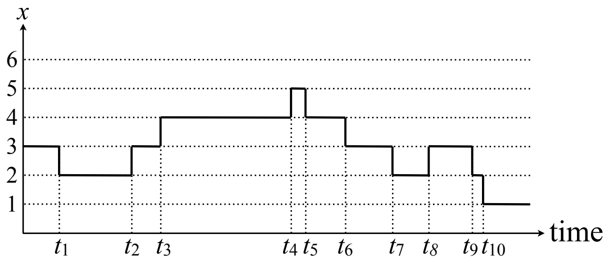
Example



Node	p_i^*	k_i^{in}	From k_i^{in}	k_i^{out}	From k_i^{out}
v_1	0.2275	5	0.2381	3	0.1429
v_2	0.0140	1	0.0476	2	0.0952
v_3	0.0899	2	0.0952	1	0.0476
v_4	0.0969	3	0.1429	3	0.1429
v_5	0.0843	1	0.0476	6	0.2857
v_6	0.2528	2	0.0952	3	0.1429
v_7	0.0140	1	0.0476	2	0.0952
v_8	0.2205	6	0.2857	1	0.0476

CTRWs on networks

(a)

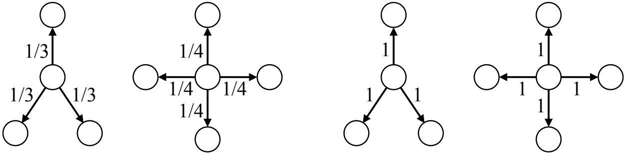


(b)

steps (n)	0	1	2	3	4	5	6	7	8	9	10
x	3	2	3	4	5	4	3	2	3	2	1

- Node-centric CTRW
  1. The walker at node i waits an inter-event time τ distributed according to $\psi(\tau)$
  2. Then, it moves to an (out-) neighbour j with the probability $\propto A_{ij}$
- Edge-centric CTRW:
  1. Each edge (incident to i) is activated with an inter-event time τ distributed according to $\psi(\tau)$
  2. The walker moves the (first) activated edge to a neighbour j
- They are the same only for regular networks (or $s_i^{\text{out}} =$ (const) for directed networks)
- When $\psi(\tau)$ is a non-Poissonian distribution (cf. temporal networks), things are more complicated. Here we assume a Poissonian $\psi(\tau)$.

(a) Node-centric CTRW (b) Edge-centric CTRW



(Poissonian) node-centric CTRW

Master equation:

$$\frac{\mathrm{d}\boldsymbol{p}(t)}{\mathrm{d}t} = \boldsymbol{p}(t)(-I + T) \equiv -\boldsymbol{p}(t)L' = -\boldsymbol{p}(t)D^{-1}L$$

where

$$L' = I - T = \left(\delta_{ij} - \frac{A_{ij}}{s_i^{\mathrm{out}}}\right) \quad \text{(random-walk normalized Laplacian)}$$

$$D = \mathrm{diag}((s_1^{\mathrm{out}}, \ldots, s_N^{\mathrm{out}}) \quad (\mathrm{diag}(s_1, \ldots, s_N) \text{ if undirected networks})$$

$$L \equiv D - A \quad \text{((combinatorial) Laplacian matrix)}$$

Stationary density: $\frac{\mathrm{d}\boldsymbol{p}^*(t)}{\mathrm{d}t} = 0$
$\rightarrow \boldsymbol{p}^* = \boldsymbol{p}^* T$. Same as DTRW.
If the network is undirected, $p_i^* = s_i / \sum_{\ell=1}^{N} s_\ell$.

(Poissonian) edge-centric CTRW

Master equation:

$$\frac{\mathrm{d}\boldsymbol{p}(t)}{\mathrm{d}t} = \boldsymbol{p}(t)(-D + A) = -\boldsymbol{p}(t)L$$

Stationary density:

$$\boldsymbol{p}^* L = 0$$

i.e., zero eigenvector of L

- For undirected networks, $\boldsymbol{p}^* = \frac{1}{N}(1 , \ldots , 1)$
- For directed networks, no simple solution but the first-order approximation:

$$p_i^* \approx (\text{const}) \times \frac{s_i^{\text{in}}}{s_i^{\text{out}}}$$

Interim summary

- DTRW
  - driven by transition-probability matrix T
  - $\boldsymbol{p}^* = \boldsymbol{p}^* T$
  - $p_i^* \propto s_i^*$ for undirected networks
- CTRW
  - node-centric
    - driven by random-walk normalized Laplacian $L' = I - T$
    - Same $\boldsymbol{p}^*$ as DTRW
  - edge-centric
    - driven by combinatorial Laplacian $L = D - A$
    - $\boldsymbol{p}^* L = 0$
    - $p_i^* = 1/N$ for undirected networks
- Undirected vs directed

Applications

- PageRank
- Community detection
- Respondent-driven sampling

PageRank

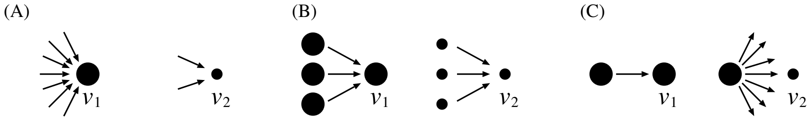
- Sergey Brin & Larry Page (1998) (but see Pinski & Narin 1976)
  Rank websites in Google search!
  They were born on 1973.
  12th and 13rd richest persons in the world (Nov 2016)
- Other applications: ranking of academic journals and papers, professional sports, disease-gene identification, recommendation systems in online marketplaces, prediction of traffic flow ...
- It is the stationary density of a DTRW on a network. Why should it be?

Web graph

Design principle of the PageRank

The rank of node i should be large when

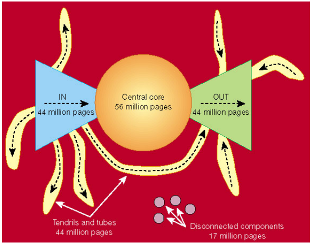
- i receives many edges (with large weights)
- i received edges from important nodes j
- i receives "exclusive" edges from j



$\boldsymbol{p}^* = \boldsymbol{p}^* T$, i.e., $p_i^* = \sum_{j=1}^N p_j^* \frac{A_{ji}}{\sum_{\ell=1}^N A_{j\ell}}$ fulfills these three criteria.

PageRank math

- Empirical networks (e.g., web graph) are usually not “strongly connected”
  - Sinks (dangling nodes) i.e., $s_i^{\text{out}} = 0$
  - Sources, i.e., $s_i^{\text{in}} = 0$
  - Multiple strongly connected components
  - transient nodes and components



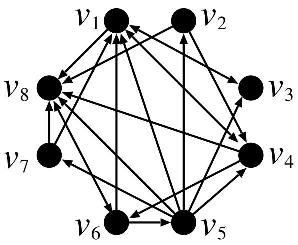
PageRank math

- So, we allow walkers to teleport to other nodes with probability $1-\alpha$ to make an effective network that is strongly connected.
- Master equation:

$$p_i(t+1) = \alpha \sum_{j=1}^{N} p_j(t) T_{ji} + (1-\alpha) u_i$$

  where $(u_1, \ldots, u_N)$ is the "preference vector" $(\sum_{i=1}^{N} u_i = 1)$
  - A popular choice: $u_i = 1/N$
- At dangling nodes, teleport with probability 1
- A popular choice: $\alpha = 0.85$
- Effective network: $T'_{ij} = \alpha T_{ij} + (1-\alpha) u_j$ (and correction for dangling nodes)
- PageRank: $\boldsymbol{p}^* = \boldsymbol{p}^* T'$

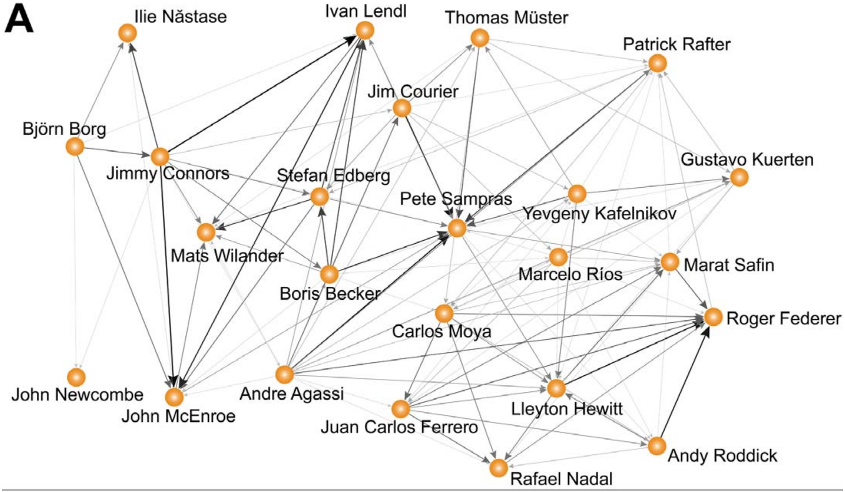
Back to the example



Node	p_i^*	k_i^{in}	k_i^{out}
v_1	0.2275	5	3
v_2	0.0140	1	2
v_3	0.0899	2	1
v_4	0.0969	3	3
v_5	0.0843	1	6
v_6	0.2528	2	3
v_7	0.0140	1	2
v_8	0.2205	6	1

- The three criteria are respected?
  - i receives many edges (with large weights)
  - i received edges from important nodes j
  - i receives “exclusive” edges from j

Tennis players on PageRank



(Radicchi, PLOS ONE, 2011)

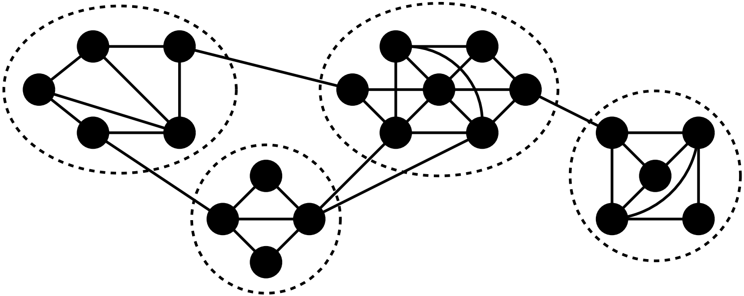
Tennis players on PageRank

Table 1. Top 30 players in the history of tennis.

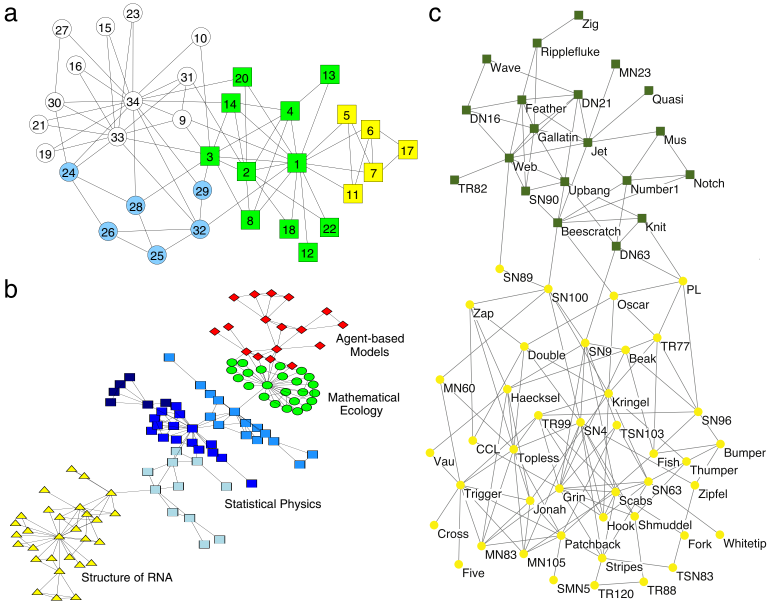
Rank	Player	Country	Hand	Start	End
1	Jimmy Connors	United States	L	1970	1996
2	Ivan Lendl	United States	R	1978	1994
3	John McEnroe	United States	L	1976	1994
4	Guillermo Vilas	Argentina	L	1969	1992
5	Andre Agassi	United States	R	1986	2006
6	Stefan Edberg	Sweden	R	1982	1996
7	Roger Federer	Switzerland	R	1998	2010
8	Pete Sampras	United States	R	1988	2002
9	Ilie Năstase	Romania	R	1968	1985
10	Björn Borg	Sweden	R	1971	1993
11	Boris Becker	Germany	R	1983	1999
12	Arthur Ashe	United States	R	1968	1979
13	Brian Gottfried	United States	R	1970	1984
14	Stan Smith	United States	R	1968	1985
15	Manuel Orantes	Spain	L	1968	1984
16	Michael Chang	United States	R	1987	2003
17	Roscoe Tanner	United States	L	1969	1985
18	Eddie Dibbs	United States	R	1971	1984
19	Harold Solomon	United States	R	1971	1991
20	Tom Okker	Netherlands	R	1968	1981
21	Mats Wilander	Sweden	R	1980	1996
22	Goran Ivaniševic′	Croatia	L	1988	2004
23	Vitas Gerulaitis	United States	R	1971	1986
24	Rafael Nadal	Spain	L	2002	2010
25	Raúl Ramirez	Mexico	R	1970	1983
26	John Newcombe	Australia	R	1968	1981
27	Ken Rosewall	Australia	R	1968	1980
28	Yevgeny Kafelnikov	Russian Federation	R	1992	2003
29	Andy Roddick	United States	R	2000	2010
30	Thomas Müster	Austria	L	1984	1999

(Radicchi, PLOS ONE, 2011)

Community detection



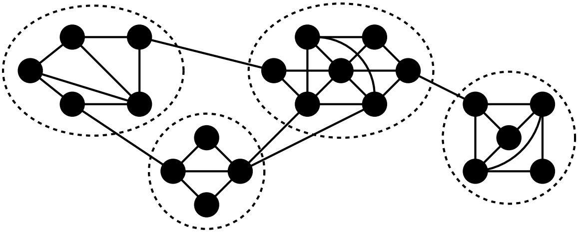
- A very very popular application of network analysis
- Various algorithms to detect communities in unlabeled networks



(Fortunato, Phys Rep, 2010)

Why RWs?

- Idea: A random walker would wander within a community for a long time before transiting to another community.
- Various algorithms based on this idea: Markov stability, Walktrap, InfoMap, Nibble, . . .



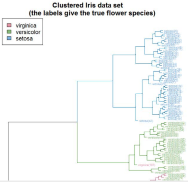
Walktrap (Pons & Latapy, J. Graph Algo., 2006)

- Consider an undirected and unweighted network
- A (DT)RW-based distance between nodes:

$$r_{ij} = \sqrt{\sum_{\ell=1}^{N} \frac{(T^n_{i\ell} - T^n_{j\ell})^2}{k_\ell}}$$

Physical meaning of r_{ij}?

- Nodes i and j with small r_{ij} would belong to the same community
- Then run a hierarchical clustering algorithm
- Discounted by k_ℓ because $p^*_\ell \propto k_\ell$
- n should not be too large because $\lim_{n\to\infty} T^n_{i\ell} = \lim_{n\to\infty} T^n_{j\ell} = p^*_\ell$, implying $r_{ij} \approx 0$



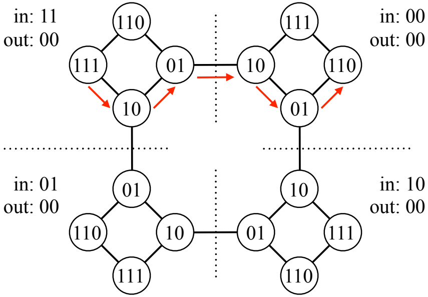
InfoMap (Rosvall & Bergstrom, PNAS 2008)

- Use DTRWs on (generally) directed and weighted networks
- Encode each node into a binary code word
  - Ex: If $v_1 \to 000$, $v_2 \to 001$, $v_3 \to 010$, $v_4 \to 011$, $v_5 \to 100$, $\ldots$, Trajectory $v_3, v_6, v_3, v_1, v_8, \ldots$ is encoded into $010101010000111\cdots$.
- Unique decoding requires a "prefix-free" coding scheme.
  - : Ex: If $v_1 \to 000$ and $v_2 \to 0001$?
  - Huffman code is a famous prefix-free code yielding short binary sequences.
    frequently visited node $\to$ short code word
  - The Shannon entropy gives the lower bound of the mean code word length

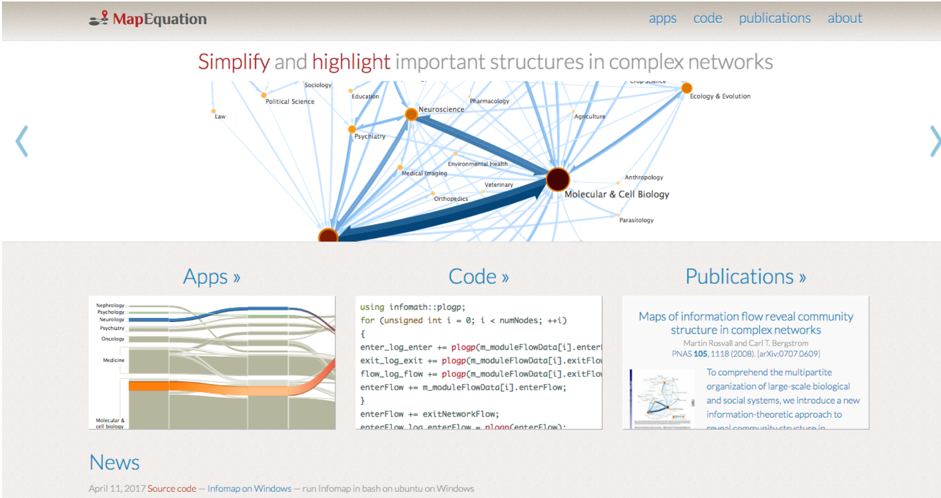
$$H = -\sum_{i=1}^{N} p_i^* \log p_i^*$$

Not about communities so far

- Use coding schemes local to individual communities.
  - Extra workaround to get around entry to and exit from communities
  - But shorter code words if walkers wander within a community
  - Doesn't matter if the same local code word is used in different communities
- Find such a two-layer Huffman code that minimises the mean code word length



11 111 10 01 00 00 10 01 110 . . .


MapEquation
apps code publications about
Simplify and highlight important structures in complex networks
Sociology
Political Science
Law
Education
Neuroscience
Pharmacology
Agriculture
Ecology & Evolution
Psychiatry
Environmental Health
Medical Imaging
Veterinary
Orthopedics
Anthropology
Molecular & Cell Biology
Parasitology
Apps »
Code »
Publications »
Nephrology
Psychology
Neurology
Psychiatry
Oncology
Medicine
Molecular & cell biology
using infomath::plogp;
for (unsigned int i = 0; i < numNodes; ++i)
{
enter_log_enter += plogp(m_moduleFlowData[i].enter
exit_log_exit += plogp(m_moduleFlowData[i].exitFlo
flow_log_flow += plogp(m_moduleFlowData[i].exitFlo
enterFlow += m_moduleFlowData[i].enterFlow;
}
enterFlow += exitNetworkFlow;
Maps of information flow reveal community structure in complex networks
Martin Rosvall and Carl T. Bergstrom
PNAS 105, 1118 (2008). [arXiv:0707.0609]
To comprehend the multipartite organization of large-scale biological and social systems, we introduce a new information-theoretic approach to
News
April 11, 2017 Source code — Infomap on Windows — run Infomap in bash on ubuntu on Windows

Respondent-driven sampling

- Estimate the fraction of infected individuals in a large or difficult-to-reach population.
  What would you do?

Respondent-driven sampling

- Estimate the fraction of infected individuals in a large or difficult-to-reach population.
  What would you do?
- Respondent-driven sampling uses edge-tracing in a social network.

Respondent-driven sampling

- Estimate the fraction of infected individuals in a large or difficult-to-reach population.
  What would you do?
- Respondent-driven sampling uses edge-tracing in a social network.
  1. Start from a seed individual
  2. He/she recruits neighbours to a survey by passing a coupon to each of them.
  3. The successfully recruited individuals then participate in the survey and also pass coupons to their neighbors.
  4. Calculate a weighted mean of the samples to derive an estimate

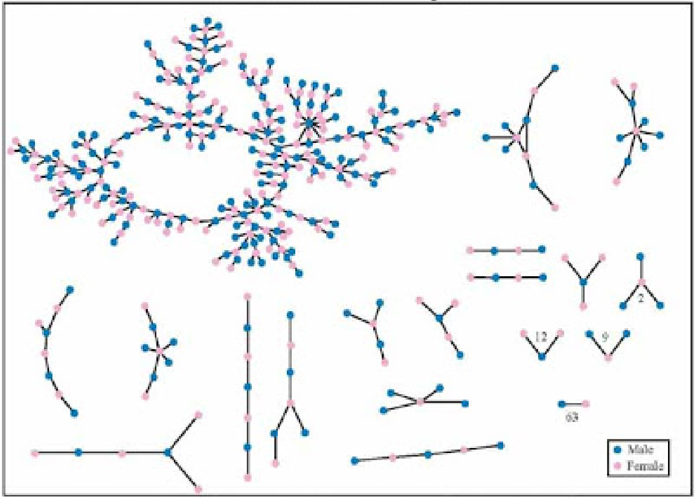
Respondent-driven sampling

- Estimate the fraction of infected individuals in a large or difficult-to-reach population.
  What would you do?
- Respondent-driven sampling uses edge-tracing in a social network.
  1. Start from a seed individual
  2. He/she recruits neighbours to a survey by passing a coupon to each of them.
  3. The successfully recruited individuals then participate in the survey and also pass coupons to their neighbors.
  4. Calculate a weighted mean of the samples to derive an estimate

  - Don’t forget to reward participants

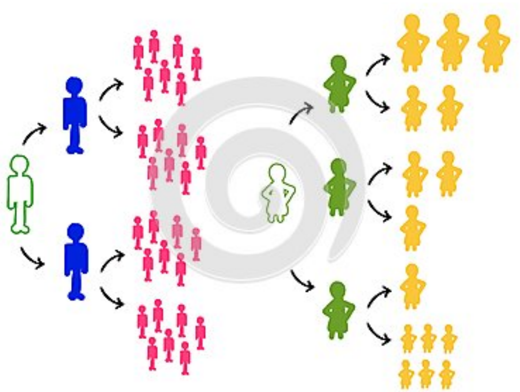
- Useful for difficult-to-reach populations

The Structure of Romantic and Sexual Relations at "Jefferson High School"



Each circle represents a student and lines connecting students represent romantic relations occuring within the 6 months preceding the interview. Numbers under the figure count the number of times that pattern was observed (i.e. we found 63 pairs unconnected to anyone else).

- RDS $=$ “snowball sampling” $+$ mathematics



(www.dreamstime.com)

Q: Why a “weighted” mean?

- y_i: quantity assigned to node i (e.g., infected or not, age, opinion)
- N_S: number of samples S
- $p_i^* \propto k_i$
- Estimator: $\langle \hat{y} \rangle = \frac{1}{N_S} \sum_{v_i \in S} \frac{y_i}{N\hat{p}_i^*}$
  because $N\hat{p}_i^*$ is normalised ($\langle N\hat{p}_i^* \rangle = 1$)
- Use $\hat{p}_i^* = \frac{k_i}{N\langle \hat{k} \rangle}$ to rewrite

$$\langle \hat{y} \rangle = \frac{1}{N_S} \sum_{v_i \in S} \frac{y_i \langle \hat{k} \rangle}{k_i}$$

- What's all this mess?

- y_i: quantity assigned to node i (e.g., infected or not, age, opinion)
- N_S: number of samples S
- $p_i^* \propto k_i$
- Estimator: $\langle \hat{y} \rangle = \frac{1}{N_S} \sum_{v_i \in S} \frac{y_i}{N\hat{p}_i^*}$
  because $N\hat{p}_i^*$ is normalised ($\langle N\hat{p}_i^* \rangle = 1$)
- Use $\hat{p}_i^* = \frac{k_i}{N\langle \hat{k} \rangle}$ to rewrite

$$\langle \hat{y} \rangle = \frac{1}{N_S} \sum_{v_i \in S} \frac{y_i \langle \hat{k} \rangle}{k_i}$$

- What's all this mess?
  We do not know the value of p_i^* or $\langle k \rangle$. We have to estimate them.

We use

$$\langle \hat{k} \rangle = \frac{\sum_{v_i \in S} \frac{k_i}{Np_i^*}}{\sum_{v_i \in S} \frac{1}{Np_i^*}} = \frac{N_S}{\sum_{v_i \in S} (k_i)^{-1}}$$

$$\langle \hat{y} \rangle = \frac{1}{N_S} \sum_{v_i \in S} \frac{y_i \langle \hat{k} \rangle}{k_i} = \frac{\sum_{v_i \in S} (k_i)^{-1} y_i}{\sum_{v_i \in S} (k_i)^{-1}}$$

(RDS II estimator. Volz & Heckathorn, J Official Stat, 2008)

E.g., proportion of nodes that have a discrete type A (e.g., infected):

$$\hat{P}_A = \frac{\sum_{v_i \in A \cap S} (k_i)^{-1}}{\sum_{v_i \in S} (k_i)^{-1}}$$

Resources

Masuda, Porter & Lambiotte, arXiv:1612.03281

- Other topics covered:
  - Fourier and Laplace transforms
  - First-passage times
  - Fractal networks
  - Multilayer networks
  - Temporal networks
  - Discrete-choice models
  - More on community detection
  - Diffusion maps
  - Opinion formation models