Centrality measures on Markov chains

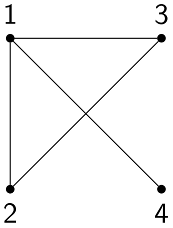
With applications to roads and infection models

Federico Poloni

University of Pisa

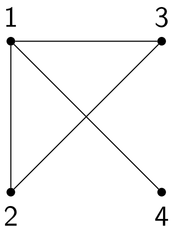
Aalto University
Colloquium, Math & Systems Analysis Dept.
May 9, 2023

Networks



Undirected, unweighted in this talk, for simplicity, but most results generalize.

Networks and matrices



$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad P = D^{-1}A = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$D =$ diagonal matrix with (out)-degrees d_i.

Random walks

Model: a random surfer visits a sequence of nodes taking random edges between them: $x_0, x_1, x_2, \ldots$.

$$P[x_{t+1} = j \mid x_t = i] = \frac{A_{ij}}{d_i} = P_{ij}$$

Even meaningless paths like returning to the state they just came from.

Under mild conditions (e.g., no disconnected parts, dead ends, periodicity…) $P[x_t = i] = \boldsymbol{\pi}_i$, for a certain vector $\boldsymbol{\pi}$ which is the left eigenvector of P corresponding to 1: $\boldsymbol{\pi} P = \boldsymbol{\pi}$

Interpretation: π_i is the average time spent on vertex i.

If A is a symmetric network, $\pi \sim$ degrees.

On our running example: $\boldsymbol{\pi} = [3/8,\ 2/8,\ 2/8,\ 1/8]$.

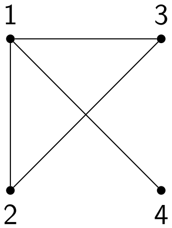
Pagerank

The most famous Markov-chain-based centrality measure: Pagerank

Small modification to the previous model: the random surfer follows a link with probability α, and teleports to another node chosen at random with probability $1 - \alpha$.

This is again a Markov chain model, with

$$P_{pr} = \alpha D^{-1} A + (1 - \alpha) \frac{1}{n} \mathbf{1}\mathbf{1}^\top .$$

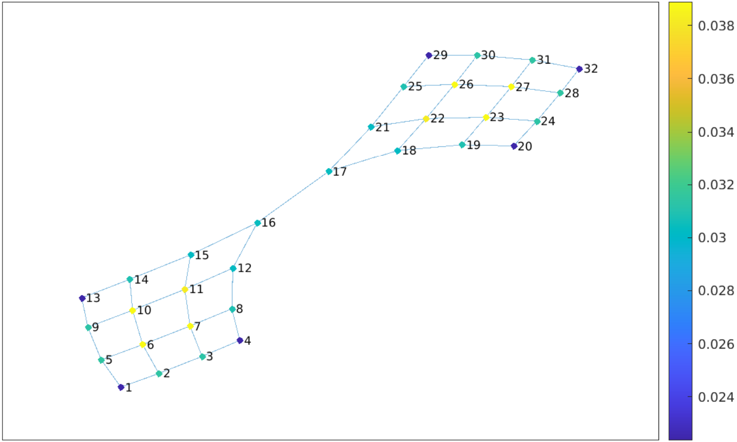


$$PR = \begin{bmatrix} 0.3667 & 0.2459 & 0.2459 & 0.1414 \end{bmatrix}$$

Importance

Pagerank works extremely well in identifying well-connected nodes

But importance is in the eye of the beholder.



Weak ties

Weak ties [Granovetter, '73]

A weak tie is one of the few, sparse links between well-connected clusters.

Example application: road network: identify bottlenecks.

Pagerank does not work well to identify weak ties: random surfers get stuck in clusters.

The Kemeny constant

Probabilistic definition: K the mean time the random walker takes to reach a state j drawn randomly according the invariant distribution $\boldsymbol{\pi}$.

Car-based interpretation: In Random-Walk-Town, each car takes a random walk on the road network.

After a very long number of time steps, a car breaks down. The node j in which it breaks down is distributed according to $\boldsymbol{\pi}$.

The tow truck sets out for repair starting from the car repair shop. But of course it travels randomly, too! How long will it take to reach the location of the broken-down car?

Surprisingly, this time is independent of the location of the car repair shop!

The Kemeny constant

This quantity can be expressed as a function of the eigenvalues $\lambda_1 = 1, \lambda_2, \ldots, \lambda_n$ of $P \in \mathbb{R}^{n \times n}$:

Kemeny constant [Kemeny,Snell '60]

$$K(P) = \sum_{k=2}^{n} \frac{1}{1 - \lambda_k}.$$

$K(P)$ small $\iff$ A well-connected as a network.

Centralities

We can define a centrality measure for roads (edges) based on the Kemeny constant: a road is important if its removal causes a large increase in $K(P)$:

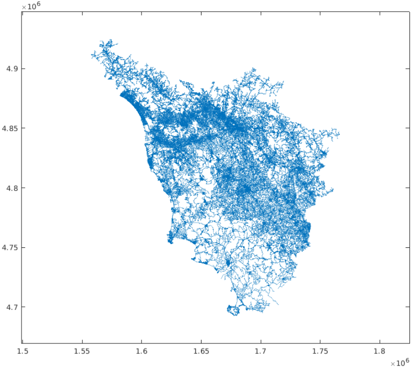
$$c(e) = K(\hat{P}) - K(P).$$

Many other centrality measures are available in literature. [Estrada, book '13]

Main inspirations for us:

- [Estrada, D.Higham, Hatano '09]: communicability betweenness centrality: variation in communicability centrality caused by the removal of an edge.
- [Crisostomi, Kirkland, Shorten '11]: Kemeny constant variation in a Markov chain model of road circulation. Main difference: we do not want to rely on external traffic data, just on the map.

Application

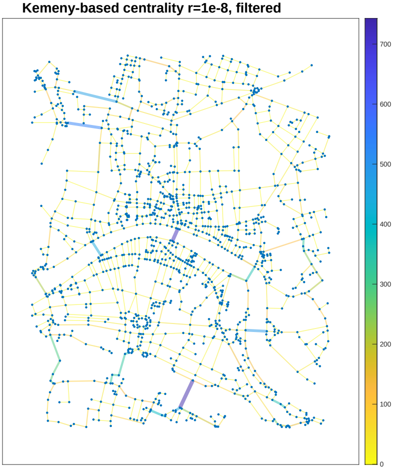


Collaboration with our civil engineering department; research question: is industry location driven by well-connected outskirts?

Large scale maps, e.g., continental Tuscany: 1.56M edges; no traffic data.

Weak ties

Goal: highlight weak ties [Granovetter, '73], i.e., crucial edges that separate (strongly-connected) sections of the map. Example: bridges.



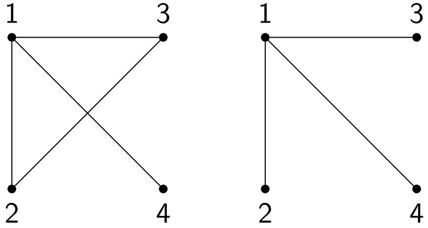
Challenges

- Deal with negative centralities;
- Deal with cut-edges;
- Make it fast enough for $1.5M$ road elements.

Negative centralities

Sometimes, the Kemeny constant decreases when removing an edge!

Example $K(\text{left}) \approx 2.54, K(\text{right}) = 2.5.$



Not ideal: intuition of "connectedness" says more roads are always better.

This phenomenon is known as Braess paradox [Braess '68, Kirkland, Zeng '16].

Analysis

Kemeny constant

$$K(P) = \sum_{i=2}^{n} \frac{1}{1-\lambda_i}.$$

$$\{\lambda_1 = 1, \dots, \lambda_n\} = \mathrm{eig}(D^{-1}A) = \mathrm{eig}(\underbrace{D^{-1/2}AD^{-1/2}}_{:=W,\ \text{symmetrized adjacency matrix}})$$

The edge removal changes W in a non-trivial way.

$$\begin{bmatrix} 0 & 1/2 & 6^{-1/2} & 0 \\ 1/2 & 0 & 6^{-1/2} & 1 \\ 6^{-1/2} & 6^{-1/2} & 0 & 3^{-1/2} \\ 0 & 0 & 3^{-1/2} & 0 \end{bmatrix} \to \begin{bmatrix} 0 & 0 & 3^{-1/2} & 0 \\ 0 & 0 & 3^{-1/2} & 0 \\ 3^{-1/2} & 3^{-1/2} & 0 & 3^{-1/2} \\ 0 & 0 & 3^{-1/2} & 0 \end{bmatrix}$$

Solution

Idea Replace the removed edge with two loop edges, $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

This changes the model in an easier-to-predict way:

$$\begin{bmatrix} 0 & 1/2 & 6^{-1/2} & 0 \\ 1/2 & 0 & 6^{-1/2} & 1 \\ 6^{-1/2} & 6^{-1/2} & 0 & 3^{-1/2} \\ 0 & 0 & 3^{-1/2} & 0 \end{bmatrix} \to \begin{bmatrix} 1/2 & 0 & 6^{-1/2} & 0 \\ 0 & 1/2 & 6^{-1/2} & 1 \\ 6^{-1/2} & 6^{-1/2} & 0 & 3^{-1/2} \\ 0 & 0 & 3^{-1/2} & 0 \end{bmatrix}$$

$$W \mapsto \hat{W} := W + \frac{1}{\sqrt{d_i d_j}}(e_i - e_j)(e_i - e_j)^T.$$

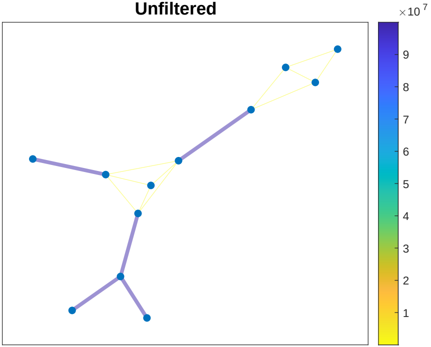
Theorem

With this definition, $c(e) = k(\hat{P}) - k(P) \geq 0$ after each edge removal.

Proof Standard eigenvalue inequalities for symmetric matrices:
$\hat{W} \succeq W \implies \hat{\lambda}_i \geq \lambda_i \implies \sum \frac{1}{1-\hat{\lambda}_i} \geq \frac{1}{1-\lambda_i}$.

Cut-edges

(Color scheme: blue edge = higher = important.)



Problem If the removed edge is a cut-edge, $\hat{G}$ is disconnected, $\hat{\lambda}_2 = 1$, and $K(\hat{P}) = +\infty$.

On a road network, cut-edges are often unimportant dead ends, but sometimes they are crucial for connectivity and cannot be ignored.

Solution

First idea Change the definition to

$$K_r(P) = \sum_{i=2}^{n} \frac{1}{1 + r - \lambda_i}.$$

for a small regularization factor $r > 0$, e.g., $r = 10^{-6}$.

$\leftrightarrow$ replacing the Laplacian $L = D - A$ with $(1 + r)D - A$.

$\leftrightarrow$ adding a small teleport probability à la Pagerank.

Problem Centrality scores $c_r(e) = K_r(\hat{P}) - K_r(P)$ of cut-edges are still $\approx r^{-1}$, artificially high.

Solution: Filtered Kemeny-based centrality

$$\tilde{c}_r(e) = \begin{cases} \frac{1}{r} - c_r(e) & e \text{ is a cut-edge,} \\ c_r(e) & \text{otherwise.} \end{cases}$$

Sign reversal

Why $\frac{1}{r} - c_r(e)$ and not the more natural $c_r(e) - \frac{1}{r}$?

Theorem

If e is a cut-edge, $\frac{1}{r} - c_r(e) \geq 0$.

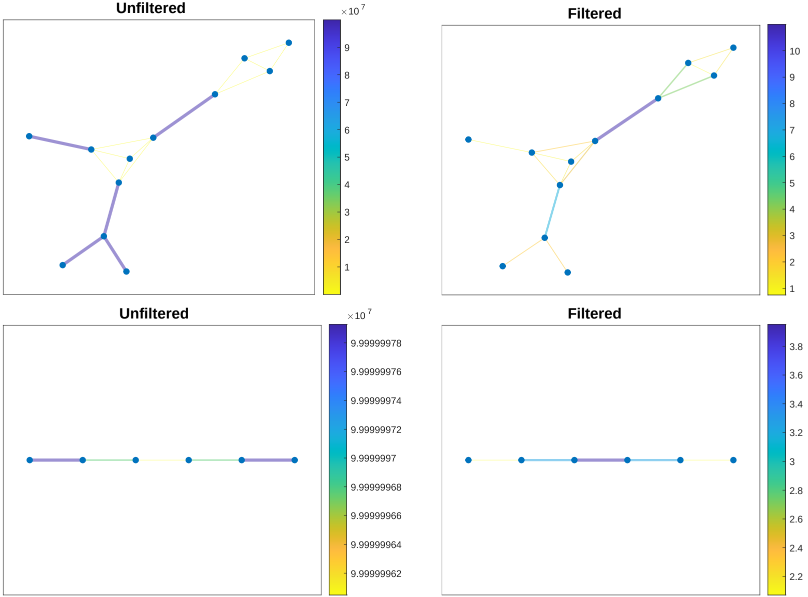
Proof Interlacing inequalities: since $\hat{W} - W \succeq 0$ is rank-1 positive semidefinite,

$$\frac{1}{r} = \hat{\lambda}_2 \geq \lambda_2 \geq \hat{\lambda}_3 \geq \lambda_3 \geq \cdots \geq \hat{\lambda}_n \geq \lambda_n.$$

Hence

$$\frac{1}{r} - c_r(e) = \underbrace{\frac{1}{1+r-\lambda_2} - \frac{1}{1+r-\hat{\lambda}_3}}_{\geq 0} + \underbrace{\frac{1}{1+r-\lambda_3} - \frac{1}{1+r-\hat{\lambda}_4}}_{\geq 0} + \cdots + \underbrace{\frac{1}{1+r-\lambda_n}}_{\geq 0}.$$

Unfiltered vs. filtered



Open problem

Filtered Kemeny-based centrality

$$\tilde{c}_r(e) = \begin{cases} \frac{1}{r} - c_r(e) & e \text{ is a cut-edge,} \\ c_r(e) & \text{otherwise.} \end{cases}$$

Empirical observation

With this definition, centrality scores of cut-edges have centrality scores comparable with non-cut-edges, and they are sorted correctly in order of importance.

We still do not have a good explanation for this observation!

Getting it done

Problem How to reduce the $\mathcal{O}(n^4)$ cost and make it fast enough for large graphs?

Theorem [Kemeny '81, Kirkland '10, Wang-Dubbeldam-Van Mieghem '17]

Let $\mathbf{w} \in \mathbb{R}^n$ be any vector such that $\mathbf{w}^T\mathbf{1} = 1$. Then,

$$K(P) = \mathrm{Trace}(S^{-1}) - 1, \quad S = I - P + \mathbf{1}\mathbf{w}^T.$$

Since $\hat{P} - P$ and $\hat{S} - S$ is a rank-1 update, we can use the

Sherman–Morrison formula

$$(S + \mathbf{u}\mathbf{v}^T)^{-1} - S^{-1} = \frac{-1}{1 + \mathbf{v}^T S^{-1}\mathbf{u}} S^{-1}\mathbf{u}\mathbf{v}^T S^{-1}$$

$$c(e) = K(\hat{P}) - K(P) = \mathrm{Trace}\left(\frac{-1}{1 + \mathbf{v}^T S^{-1}\mathbf{u}} S^{-1}\mathbf{u}\mathbf{v}^T S^{-1}\right) = \frac{-\mathbf{u}^T S^{-2}\mathbf{v}}{1 + \mathbf{v}^T S^{-1}\mathbf{u}}.$$

Final formula

Some more routine manipulations:

- Introduce regularization parameter r;
- Use again Sherman–Morrison to invert $S_r = (1+r)I - P + \mathbf{1}\mathbf{w}^T$
- Express it in terms of "regularized Laplacian" $L_r = (1+r)D - A$;
- Choose w to make the problem symmetric

Final formula

$$c(\{i,j\}) = \frac{A_{ij}\mathbf{d}^T(\mathbf{x}.^2)}{1 - A_{ij}(x_i - x_j)}, \quad \mathbf{y} = L_r^{-1}(\mathbf{e}_i - \mathbf{e}_j), \quad \mathbf{x} = \mathbf{y} - \frac{\mathbf{d}^T\mathbf{y}}{\gamma}\mathbf{z}.$$

where $\mathbf{d} = \operatorname{diag}(D)$, $\mathbf{z} = L_r^{-1}\mathbf{d}$, $\gamma = \mathbf{d}^T\mathbf{z} + \mathbf{d}^T\mathbf{1}$.

Practical cost

Final formula

$$c(\{i,j\}) = \frac{A_{ij}\mathbf{d}^T(\mathbf{x}.^2)}{1 - A_{ij}(x_i - x_j)}, \quad \mathbf{y} = L_r^{-1}(\mathbf{e}_i - \mathbf{e}_j), \quad \mathbf{x} = \mathbf{y} - \frac{\mathbf{d}^T\mathbf{y}}{\gamma}\mathbf{z}.$$

where $\mathbf{d} = \operatorname{diag}(D)$, $\mathbf{z} = L_r^{-1}\mathbf{d}$, $\gamma = \mathbf{d}^T\mathbf{z} + \mathbf{d}^T\mathbf{1}$.

1. Precompute Cholesky factorization of $L_r = (1+r)D - A$, and $\mathbf{d}, \mathbf{z}, \gamma$.
2. To compute $c(e)$ for each edge (possibly in parallel), solve one linear system with L_r (using the precomputed factorization) and $\mathcal{O}(n)$ extra operations.

On road networks, often $n \approx m \approx \operatorname{nnz}(\operatorname{chol}(L_r))$ (related to planarity).

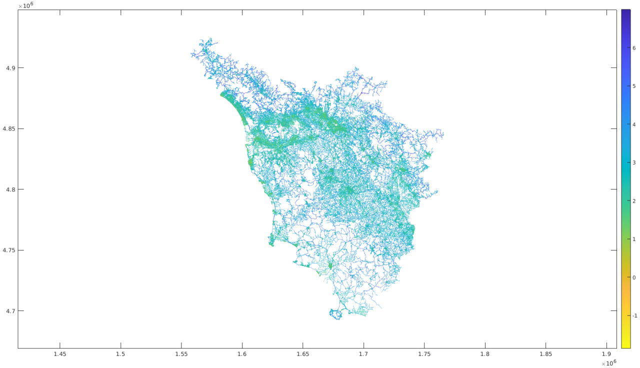
Hence all these operations are $\mathcal{O}(n)$; but the total cost to compute all centralities is still $\mathcal{O}(n^2)$.

Experiment: a large-scale network

Mainland Tuscany map: $n = 1.22M$, $m = 1.56M$, $\text{nnz}(\text{chol}(L_r)) = 3.36M$.

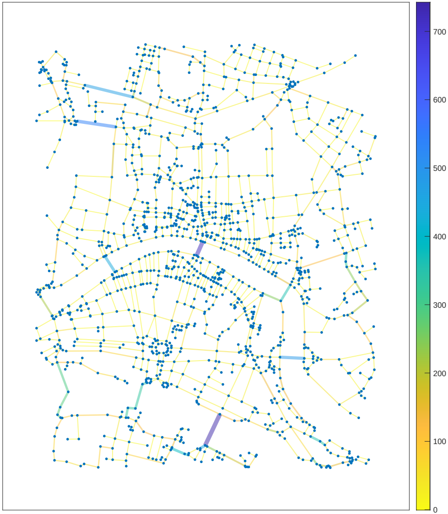
1. Precomputation and `chol` : $< 1s$.
2. `parfor` centrality computation: 18 hours.

On a machine with 12 3.4GHz Xeon physical cores.

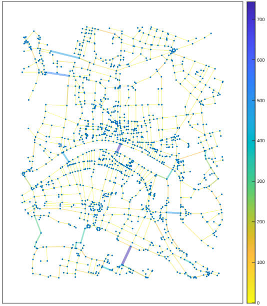


Experiment: the bridges of Pisa

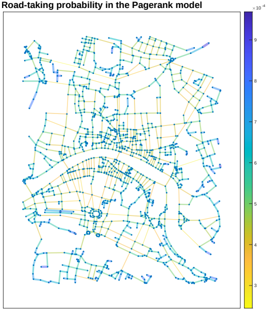
Kemeny-based centrality r=1e-8, filtered



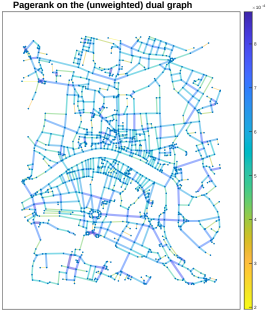
Kemeny-based centrality r=1e-8, filtered



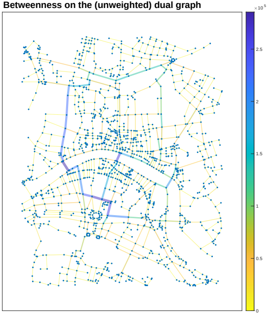
Road-taking probability in the Pagerank model



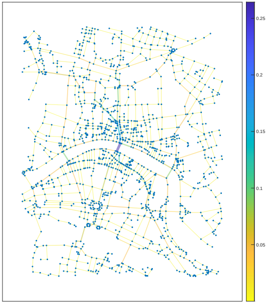
Pagerank on the (unweighted) dual graph



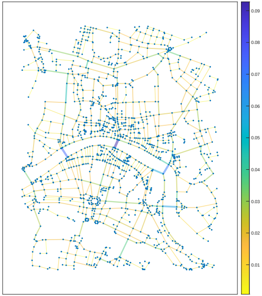
Betweenness on the (unweighted) dual graph



Edge betweenness (from Networkx)



Edge current flow betweenness (from Networkx)



Conclusions

- The Kemeny constant variation works well to highlight bottlenecks and weak ties.
- Connectivity/positivity issues can be solved.
- Computationally feasible even in large scale.
- Interesting results for our collaborators in civ-eng.

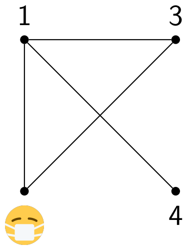
D. Altafini, D. Bini, V. Cutini, B. Meini, F. Poloni. *An edge centrality measure based on the Kemeny constant*.
Arxiv:2203.06459. To appear in SIAM J. Matrix Anal. Appl.

Infectivity

Another application: spreading on networks.

Models certain phenomena such as news, or diseases.

At each time step, the infection may spread from any infected individual to a non-infected neighbor with probability β (independently on each edge).



No recovering probability, so eventually everyone gets infected (SI model).

Mean infection time

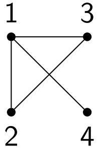
Idea: a person (node) is important if the infection takes little time to spread from them to the rest of the network.

$$MIT_i = E[\text{time to go from (only } i \text{ infected) to (everyone infected)}]$$

The spread of the infection is not a random walk: multiple people can be infected at the same time.

Different models

More experiments show that it is impossible to reconcile the two models:



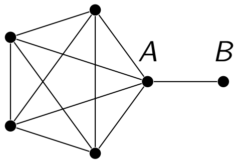
$$M_{RW} = \begin{bmatrix} 0 & 3.3333 & 3.3333 & 7 \\ 2 & 0 & 2.6667 & 9 \\ 2 & 2.6667 & 0 & 9 \\ 1 & 4.3333 & 4.3333 & 0 \end{bmatrix},$$

$$M_{inf} \approx \begin{bmatrix} 0 & 7.7922 & 7.7766 & 10.0051 \\ 7.7503 & 0 & 7.7358 & 17.7442 \\ 7.7383 & 7.7284 & 0 & 17.7707 \\ 10.0059 & 17.7454 & 17.7628 & 0 \end{bmatrix}.$$

Example: the random walker takes more time to go $2 \to 1$ than $4 \to 1$, but the infection takes less time.

Problems

Repeat: not a random walk!



Example: clique of k nodes + a lone branch B. The mean infection time from A to B is $\frac{1}{\beta}$, independently of k, but the mean first passage time of a random walker from A to B is $k^2 + k + 1$, increasing sharply with k.

In particular, random walk-based centrality measures will not work well here.

What is the correct model, then?

2^n possible states: each individual can be infected or not.

Initial state: one individual i infected, e.g., 0100.

Transition between states according to who gets infected in a time step; e.g., $0100 \to 1110$.

This is an absorbing Markov chain: each initial state $\neq 0000$ eventually reaches full infection 1111.

$$P = \begin{bmatrix} T & t \\ 0 & 1 \end{bmatrix}$$

$T =$ upper triangular matrix. Interesting layer structure: non-trivial transitions increase the number of infected individuals.

The full matrix

$$
\begin{matrix}
\ast & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast \\
0 & 0 & \ast & \ast & 0 & 0 & \ast & \ast & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \ast & 0 & 0 & 0 & \ast & 0 & 0 & 0 & \ast & 0 & 0 & 0 & \ast \\
0 & 0 & 0 & 0 & \ast & \ast & \ast & \ast & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \ast & 0 & \ast & 0 & 0 & 0 & 0 & 0 & \ast & 0 & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & \ast & \ast & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & \ast & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & 0 & \ast & 0 & \ast & 0 & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & \ast & 0 & 0 & \ast & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & 0 & 0 & 0 & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & \ast & \ast & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & 0 & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast & \ast \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ast
\end{matrix}
$$

First-hitting time

The Markov chain concept we need here: first-passage time (also: first-hitting time).

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

Lemma

Starting from state j in the first block, the mean time needed to reach the absorbing state is $\mathbf{e}_j^\top (I - P_{11})^{-1}\mathbf{1}$.

To compute the mean infection time MIT_j for all j, it is enough to solve a linear system with an upper triangular matrix $T = P_{11}$. Easy?

Problem

…we need to solve a linear system with an upper triangular matrix T of size $(2^n - 2) \times (2^n - 2)$.

Even for a relatively small network of $n = 100$ nodes, this is unfeasible.

Tensor train / DMRG techniques have been used for smaller problems [Dolgov, Savostyanov '22]

Idea (explained very shortly): see MIT_{1001} as the output of a linear dynamical system under inputs $1, 0, 0, 1$. Hopefully the size of the state (a sort of rank of the tensor) stays small.

Alternative: simulate the system many times, and approximate the infection time with sample means.

Can we improve on this method?

Our proposal

Our proposal: simulation via shortest-potential-infection-path.

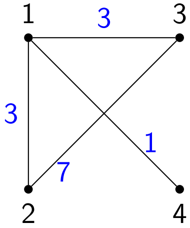
Observation: if i is infected and j is not, the infection spreads along the edge $i \to j$ with a time that has a geometric distribution.

k	1	2	3	4	...
P[spread in time k]	β	$(1-\beta)\beta$	$(1-\beta)^2\beta$	$(1-\beta)^3\beta$	...

Idea: first draw at random potential infection times along each edge, then reconstruct the real behavior.

"Folklore in some circles" [Goering, Albin et al, 2015]

Example



Time t	Infected set	$\hat{\mathbf{X}}(t)$
0	$\{2\}$	(0,1,0,0)
1	$\{2\}$	(0,1,0,0)
2	$\{2\}$	(0,1,0,0)
3	$\{1,2\}$	(1,1,0,0)
4	$\{1,2,4\}$	(1,1,0,1)
5	$\{1,2,4\}$	(1,1,0,1)
6	$\{1,2,3,4\}$	(1,1,1,1)
7	$\{1,2,3,4\}$	(1,1,1,1)
8	$\{1,2,3,4\}$	(1,1,1,1)

On the algorithm

At the cost of computing one all-to-all shortest-path matrix, we can sample infections starting from each of the n nodes at the same time.

With sufficiently many samples (possibly in parallel), we can compute sample mean times.

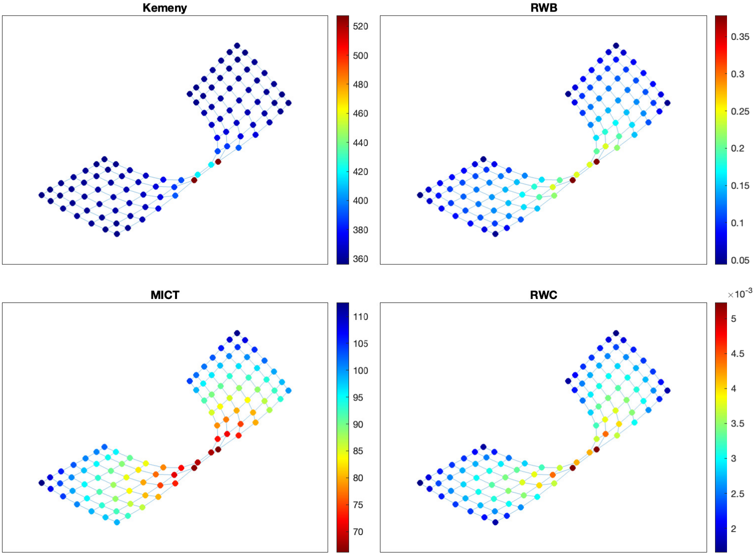
This strategy is cheaper than direct simulation of infection.

As with all sampling methods, convergence is rather slow: error $\mathcal{O}(1/k^2)$ with k samples.

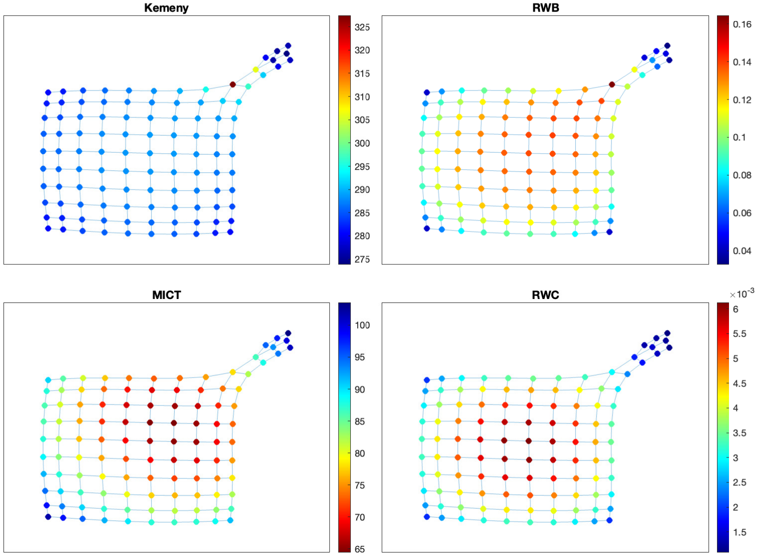
Interesting corollary: the (limit) infection time matrix is symmetric.

$$M_{inf} \approx \begin{bmatrix} 0 & 7.7922 & 7.7766 & 10.0051 \\ 7.7503 & 0 & 7.7358 & 17.7442 \\ 7.7383 & 7.7284 & 0 & 17.7707 \\ 10.0059 & 17.7454 & 17.7628 & 0 \end{bmatrix}.$$

Example


Kemeny
520
500
480
460
440
420
400
380
360
RWB
0.35
0.3
0.25
0.2
0.15
0.1
0.05
MICT
110
105
100
95
90
85
80
75
70
RWC
$\times 10^{-3}$
5
4.5
4
3.5
3
2.5
2

Example


Kemeny
325
320
315
310
305
300
295
290
285
280
275
RWB
0.16
0.14
0.12
0.1
0.08
0.06
0.04
MICT
100
95
90
85
80
75
70
65
RWC
×10^-3
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5

Conclusions

This shortest potential-infection path idea can be generalized to many settings:

- directed networks;
- edge weights;
- recovered individuals (SIR)
- . . .

Can we beat it with "exact" tensor methods?

Main message

Do not use walking to model spreading!

Sooyeong Kim, Jane Breen, Ekaterina Dudkina, Federico Poloni, Emanuele Crisostomi. *On the use of Markov chains for epidemic modeling on networks* PLOS One 2023.