A randomized k-centrality measure & applications to Networks Node Immunization

Luca Avena (Leiden, The Netherlands)

A day on random graphs, TorVergata, 30/05/2022

joint with **Michael Emmerich, Alexandre Gaudillière and Irina Gurewitsch**

Outline

1. **Compartmental models on networks**
2. **Multiple-node immunization**
3. **Our method: rooted forests & randomized k-centrality**
4. **Experiments: the geometry of contagion**

Epidemic spread: a very brief modelling overview

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments + data fitting*).

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments* + *data fitting*).

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments + data fitting*).

SIS model (2-compartments)

$I =$ Infected, $S =$ Susceptible.

$\beta =$ **Infection rate**, $\mu =$ **Recovery rate**.

Interaction as "chemical reactions":

$$S + I \xrightarrow{\beta} 2I, \qquad I \xrightarrow{\mu} S.$$

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments + data fitting*).

SIS model (2-compartments)

$$I = \text{ Infected}, \qquad S = \text{ Susceptible}.$$

$$\beta = \textbf{Infection rate}, \qquad \mu = \textbf{Recovery rate}.$$

Interaction as "chemical reactions":

$$S + I \overset{\beta}{\to} 2I, \qquad I \overset{\mu}{\to} S.$$

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments + data fitting*).

SIS model (2-compartments)

$$I = \text{ Infected,} \qquad S = \text{ Susceptible.}$$

$$\beta = \textbf{Infection rate}, \qquad \mu = \textbf{Recovery rate}.$$

Interaction as "chemical reactions":

$$S + I \xrightarrow{\beta} 2I, \qquad I \xrightarrow{\mu} S.$$

Basic Reprod. Nr. $\Rightarrow r := \beta/\mu =$ infect. rate $\times$ av. durat. infectiousness

Epidemic spread: a very brief modelling overview

- **Compartmental models :** Population of n individuals **divided into interacting compartments** (e.g. infected, susceptible, recovered, immune).
- **Historical note :** First models back to **D. Bernoulli (1760)**; "mature" multidisciplinary area in the XX century **Kermack-McKendrick theory (1927)**; **from the '90 on** (*theory developments + data fitting*).

SIS model (2-compartments)

$I =$ Infected, $S =$ Susceptible.

$\beta =$ **Infection rate**, $\mu =$ **Recovery rate**.

Interaction as "chemical reactions":

$$S + I \xrightarrow{\beta} 2I, \qquad I \xrightarrow{\mu} S.$$

Basic Reprod. Nr. $\Rightarrow r := \beta/\mu =$ infect. rate $\times$ av. durat. infectiousness

SIS on Networks

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x\sim y\}}$.

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

Definition (**SIS evolution** (a non-linear system of n diff. eqs.))

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

Definition (**SIS evolution** (a non-linear system of n diff. eqs.))

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x\sim y\}}$.

Definition (SIS evolution (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t}), x \in \mathcal{V},$ and fix

Infection rate $= \beta > 0,$ **Recovery rate** $= \mu > 0$:

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

Definition (**SIS evolution** (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t})$, $x \in \mathcal{V}$, and fix

Infection rate $= \beta > 0$, **Recovery rate** $= \mu > 0$:

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

Definition (SIS evolution (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t})$, $x \in \mathcal{V}$, and fix

Infection rate $= \beta > 0$, **Recovery rate** $= \mu > 0$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\mu \rho_x(t) + \beta [1 - \rho_x(t)] \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

with some in. cond. $\vec{\rho}(0) := (\rho_1(0), \ldots, \rho_n(0))$.

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x \sim y\}}$.

Definition (SIS evolution (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t})$, $x \in \mathcal{V}$, and fix

Infection rate $= \beta > 0$, **Recovery rate** $= \mu > 0$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\mu \rho_x(t) + \beta[1 - \rho_x(t)] \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

with some in. cond. $\vec{\rho}(0) := (\rho_1(0), \ldots, \rho_n(0))$.

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x\sim y\}}$.

Definition (SIS evolution (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t})$, $x \in \mathcal{V}$, and fix

Infection rate $= \beta > 0$, **Recovery rate** $= \mu > 0$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\mu\rho_x(t) + \beta[1 - \rho_x(t)] \sum_{y\in\mathcal{V}} a_{x,y}\rho_y(t)$$

with some in. cond. $\vec{\rho}(0) := (\rho_1(0), \ldots, \rho_n(0))$.

Remark:

Contact process on $\{0,1\}^V =$ **microscopic Markovian version of SIS**

SIS on Networks

Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, undirected finite graph with adjacency $\mathcal{A} := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = \mathbb{1}_{\{x\sim y\}}$.

Definition (SIS evolution (a non-linear system of n diff. eqs.))

Set $\rho_{\mathbf{x}}(\mathbf{t}) := \mathbb{P}($ node $\mathbf{x}$ is infected at time $\mathbf{t})$, $x \in \mathcal{V}$, and fix

Infection rate $= \beta > 0$, **Recovery rate** $= \mu > 0$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\mu \rho_x(t) + \beta[1 - \rho_x(t)] \sum_{y\in\mathcal{V}} a_{x,y}\rho_y(t)$$

with some in. cond. $\vec{\rho}(0) := (\rho_1(0), \dots, \rho_n(0))$.

Remark:

Contact process on $\{0,1\}^V =$ **microscopic Markovian version of SIS**

Endemic state & largest eigenvalue

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t), \forall x \in \mathcal{V},$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time t), $\forall x \in \mathcal{V}$,

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$,

rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \ldots, \rho_n(t))$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \ldots, \rho_n(t))$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \dots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting **r** $:= \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \ldots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \dots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Max eigenvalue λ_{max} of $\mathcal{A}$:

If $r\lambda_{max} - 1 < 0 \Rightarrow \rho_x(t)$ decays exponentially fast to 0, for any $x \in \mathcal{V}$.

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \dots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Max eigenvalue λ_{max} of $\mathcal{A}$:

If $r\lambda_{max} - 1 < 0 \Rightarrow \rho_x(t)$ decays exponentially fast to 0, for any $x \in \mathcal{V}$.

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \ldots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Max eigenvalue λ_{max} of $\mathcal{A}$:

If $r\lambda_{max} - 1 < 0 \Rightarrow \rho_x(t)$ decays exponentially fast to 0, for any $x \in \mathcal{V}$.
$\Rightarrow$ **critical parameter for the endemic state**:

$$r_c \geq 1/\lambda_{max}.$$

Endemic state & largest eigenvalue

SIS evolution $\rho_x(t) := \mathbb{P}($ node x is infected at time $t)$, $\forall x \in \mathcal{V}$, rescaling time by $1/\mu$ and setting $\mathbf{r} := \beta/\mu$:

$$\frac{\partial \rho_x(t)}{\partial t} = -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t) - r \sum_{y \in \mathcal{V}} a_{x,y} \rho_x(t) \rho_y(t)$$

Outbreak threshold estimate (linearization) :

$$\frac{\partial \rho_x(t)}{\partial t} \leq -\rho_x(t) + r \sum_{y \in \mathcal{V}} a_{x,y} \rho_y(t)$$

From which, if $\vec{\rho}(t) := (\rho_1(t), \ldots, \rho_n(t)) \Rightarrow \vec{\rho}(t) \leq \vec{\rho}(0) e^{(r\mathcal{A} - \mathrm{Id}_n)t}$

Max eigenvalue λ_{max} of $\mathcal{A}$:

If $r\lambda_{max} - 1 < 0 \Rightarrow \rho_x(t)$ decays exponentially fast to 0, for any $x \in \mathcal{V}$.
$\Rightarrow$ **critical parameter for the endemic state**:

$$r_c \geq 1/\lambda_{max}.$$

Multiple-node immunization & max eigendrop

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since** $r_c \geq 1/\lambda_{max}$, **good strategy:** look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency: since $r_c \geq 1/\lambda_{max}$,**
  good strategy: look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since $r_c \geq 1/\lambda_{max}$,**
  good strategy: look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.
  Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k-eigendrop}$$

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since** $r_c \geq 1/\lambda_{max}$**,**
  good strategy: look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.
  Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k-eigendrop}$$

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since** $r_c \geq 1/\lambda_{max}$**,**
  good strategy: look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.
  Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k-eigendrop}$$

Computational issues with max eigendrop:

- No poly exact algorithm for such an optimization problem.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency: since $r_c \geq 1/\lambda_{max}$,** **good strategy:** look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue. Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k-eigendrop}$$

Computational issues with max eigendrop:

- No poly exact algorithm for such an optimization problem.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since $r_c \geq 1/\lambda_{max}$,** **good strategy:** look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue. Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k-eigendrop}$$

Computational issues with max eigendrop:

- No poly exact algorithm for such an optimization problem.
- **Chen Chen et al. (2016)** proved **approx. equivalence with so-called Shield-Value optimization which is NP complete**.

Multiple-node immunization & max eigendrop

- **Goal:** Identify & remove set of k-nodes to empede the "*endemic outbreak*".
- **Problem not well-posed:** 1) limited rigorous analysis 2) heuristics ideas based on simulations with lots of other problems 3) choosing maximal degree nodes is not justified already for $k = 1$.
- **Max eigendrop of adjacency:** **since $r_c \geq 1/\lambda_{max}$,**
  good strategy: look for sets of $k-$nodes from which after removal the reduced network has minimal largest eigenvalue.
  Equivalently: find $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ maximising

$$\Delta_k \lambda(\mathcal{S}) := \lambda_{max}(\mathcal{V}) - \lambda_{max}(\mathcal{V} \setminus \mathcal{S}) = \mathbf{k - eigendrop}$$

Computational issues with max eigendrop:

- No poly exact algorithm for such an optimization problem.
- **Chen Chen et al. (2016)** proved **approx. equivalence with so-called Shield-Value optimization which is NP complete**.

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x) u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x) u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x) u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

- **Shield Value approximates max k-eigendrop**:

$$\Delta_k \lambda(S) \approx SV_k(\mathcal{S}).$$

- **Maximising shieldvalue is NP-complete problem.**

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x)u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

- **Shield Value approximates max k-eigendrop**:

$$\Delta_k \lambda(S) \approx SV_k(\mathcal{S}).$$

- **Maximising shieldvalue is NP-complete problem**.

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x) u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

- **Shield Value approximates max k-eigendrop**:

$$\Delta_k \lambda(S) \approx SV_k(\mathcal{S}).$$

- **Maximising shieldvalue is NP-complete problem**.
- **Netshield** : **greedy fast algorithm** which finds near-to-optimal solutions in

$$O(nk + m) \text{ running time,}$$

with m =number of edges.

Max k-eigendrop, ShieldValue & Netshield (Chen Chen et al. (2016))

- **Shield Value** : for $\mathcal{S} \subset \mathcal{V}$, with $|\mathcal{S}| = k$ define

$$SV_k(\mathcal{S}) := 2\lambda_{max} \sum_{x \in \mathcal{S}} u^2(x) - \sum_{x,y \in \mathcal{S}} a_{x,y} u(x)u(y)$$

(**"eingescore"** − **"repellence"**),

with u principal eigenvector ($\mathcal{A}u = \lambda_{max} u$).

- **Shield Value approximates max k-eigendrop**:

$$\Delta_k \lambda(S) \approx SV_k(\mathcal{S}).$$

- **Maximising shieldvalue is NP-complete problem**.
- **Netshield** : **greedy fast algorithm** which finds near-to-optimal solutions in

$$O(nk + m) \text{ running time,}$$

with m =number of edges.

Our randomized immunization method:

Random Rooted Spanning Forests

&

randomized k-centrality

Graph Laplacian and associated RW

Weighted Directed Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with $|\mathcal{V}| = n$ and weighted adjacency $\mathcal{A}_w := (a_{x,y})_{x,y \in \mathcal{V}}$ s.t. $a_{x,y} = w(x,y)\mathbb{1}_{\{x \neq y\}}$.

Graph Laplacian and associated RW

Weighted Directed Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with $|\mathcal{V}| = n$ and weighted adjacency $\mathcal{A}_w := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = w(x,y)\mathbb{1}_{\{x\neq y\}}$.

Definition (Graph Laplacian)

It is the $n \times n$ matrix $-\mathcal{L}_w$ with :

$$\mathcal{L}_w(x,y) = \begin{cases} w(x,y) & \text{if } x \neq y \in \mathcal{V}, \\ -\sum_{z\neq x} w(x,z) & \text{on the diagonal.} \end{cases}$$

In particular, $\mathcal{L}_w = \mathcal{A}_w - \mathcal{D}_w =$ (weighted) "adjacency" $-$ "degree"

Graph Laplacian and associated RW

Weighted Directed Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with $|\mathcal{V}| = n$ and weighted adjacency $\mathcal{A}_w := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = w(x,y)\mathbb{1}_{\{x\neq y\}}$.

Definition (**Graph Laplacian**)

It is the $n \times n$ matrix $-\mathcal{L}_w$ with :

$$\mathcal{L}_w(x,y) = \begin{cases} w(x,y) & \text{if } x \neq y \in \mathcal{V}, \\ -\sum_{z\neq x} w(x,z) & \text{on the diagonal.} \end{cases}$$

In particular, $\mathcal{L}_w = \mathcal{A}_w - \mathcal{D}_w =$ (weighted) "adjacency" $-$ "degree"

Graph Laplacian and associated RW

Weighted Directed Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with $|\mathcal{V}| = n$ and weighted adjacency $\mathcal{A}_w := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = w(x,y)\mathbb{1}_{\{x\neq y\}}$.

Definition (**Graph Laplacian**)

It is the $n \times n$ matrix $-\mathcal{L}_w$ with :

$$\mathcal{L}_w(x,y) = \begin{cases} w(x,y) & \text{if } x \neq y \in \mathcal{V}, \\ -\sum_{z\neq x} w(x,z) & \text{on the diagonal.} \end{cases}$$

In particular, $\mathcal{L}_w = \mathcal{A}_w - \mathcal{D}_w =$ (weighted) "adjacency" $-$ "degree"

Definition (**Associated Random Walk**)

Let $X = (X(t))_{t\geq 0}$ be the continuous-time Markov chain on $\mathcal{V}$ with infinitesimal generator $\mathcal{L}_w$.

Graph Laplacian and associated RW

Weighted Directed Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with $|\mathcal{V}| = n$ and weighted adjacency $\mathcal{A}_w := (a_{x,y})_{x,y\in\mathcal{V}}$ s.t. $a_{x,y} = w(x,y)\mathbb{1}_{\{x\neq y\}}$.

Definition (**Graph Laplacian**)

It is the $n \times n$ matrix $-\mathcal{L}_w$ with :

$$\mathcal{L}_w(x,y) = \begin{cases} w(x,y) & \text{if } x \neq y \in \mathcal{V}, \\ -\sum_{z\neq x} w(x,z) & \text{on the diagonal.} \end{cases}$$

In particular, $\mathcal{L}_w = \mathcal{A}_w - \mathcal{D}_w =$ (weighted) "adjacency" $-$ "degree"

Definition (**Associated Random Walk**)

Let $X = (X(t))_{t\geq 0}$ be the continuous-time Markov chain on $\mathcal{V}$ with infinitesimal generator $\mathcal{L}_w$.

RW and a determinantal set of nodes

RW and a determinantal set of nodes

Definition (Set of Roots $\mathcal{R}_q$ of intesity q- A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and a parameter $q > 0$, let $\mathcal{R}_q \subseteq \mathcal{V}$ be **a random subset of nodes** with law characterized by:

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{R}_q) = \det [K_q]_{\mathcal{A}}, \qquad \text{for any } \mathcal{A} \subseteq \mathcal{V},$$

with

$$K_q(x, y) := \mathbf{P}_x(X(T_q) = y) = q(q\mathrm{Id} - \mathcal{L}_w)^{-1}(x, y)$$

and T_q **an independent exponential random variable** of parameter q.

RW and a determinantal set of nodes

Definition (Set of Roots $\mathcal{R}_q$ of intesity q- A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and a parameter $q > 0$, let $\mathcal{R}_q \subseteq \mathcal{V}$ be **a random subset of nodes** with law characterized by:

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{R}_q) = \det [K_q]_{\mathcal{A}}, \qquad \text{for any } \mathcal{A} \subseteq \mathcal{V},$$

with

$$K_q(x, y) := \mathbf{P}_x(X(T_q) = y) = q(q\mathrm{Id} - \mathcal{L}_w)^{-1}(x, y)$$

and T_q **an independent exponential random variable** of parameter q.

RW and a determinantal set of nodes

Definition (Set of Roots $\mathcal{R}_q$ of intesity q- A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and a parameter $q > 0$, let $\mathcal{R}_q \subseteq \mathcal{V}$ be **a random subset of nodes** with law characterized by:

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{R}_q) = \det [K_q]_{\mathcal{A}}, \qquad \text{for any } \mathcal{A} \subseteq \mathcal{V},$$

with

$$K_q(x, y) := \mathbf{P}_x(X(T_q) = y) = q(q\mathrm{Id} - \mathcal{L}_w)^{-1}(x, y)$$

and T_q **an independent exponential random variable** of parameter q.

Repelling roots when $w(x, y) = w(y, x)$

$$\mathbb{P}(\{x, y\} \subseteq \mathcal{R}_q) \leq \mathbb{P}(x \in \mathcal{R}_q)\,\mathbb{P}(y \in \mathcal{R}_q)$$

RW and a determinantal set of nodes

Definition (Set of Roots $\mathcal{R}_q$ of intesity q- A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and a parameter $q > 0$, let $\mathcal{R}_q \subseteq \mathcal{V}$ be **a random subset of nodes** with law characterized by:

$$\mathbb{P}\left(\mathcal{A} \subseteq \mathcal{R}_q\right) = \det\left[K_q\right]_{\mathcal{A}}, \qquad \text{for any } \mathcal{A} \subseteq \mathcal{V},$$

with

$$K_q(x, y) := \mathbf{P}_x(X(T_q) = y) = q(q\mathrm{Id} - \mathcal{L}_w)^{-1}(x, y)$$

and T_q **an independent exponential random variable** of parameter q.

Repelling roots when $w(x, y) = w(y, x)$

$$\mathbb{P}\left(\{x, y\} \subseteq \mathcal{R}_q\right) \leq \mathbb{P}\left(x \in \mathcal{R}_q\right)\mathbb{P}\left(y \in \mathcal{R}_q\right)$$

$\mathcal{R}_q$ from Random Rooted Forests

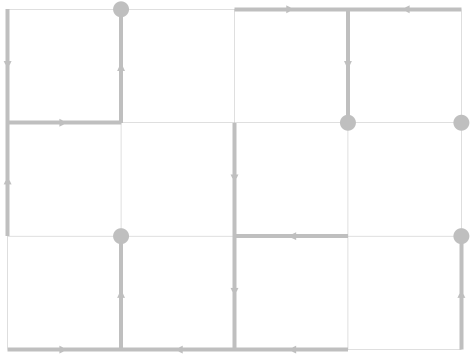
Forest Space: for $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, set

$$\mathcal{F}_{\mathcal{G}} := \{\text{ spanning rooted forests on } \mathcal{G}\}$$

$\mathcal{R}_q$ from Random Rooted Forests

Forest Space: for $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, set

$$\mathcal{F}_{\mathcal{G}} := \{ \text{ spanning rooted forests on } \mathcal{G} \}$$

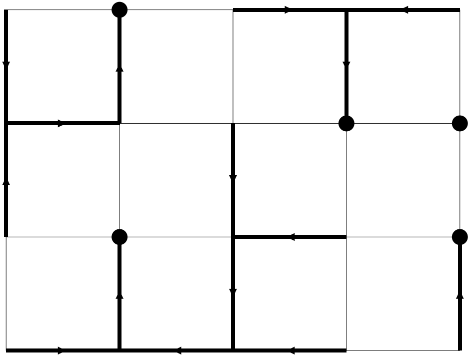


A spanning rooted forest on a square grid.

$\mathcal{R}_q$ from Random Rooted Forests

Forest Space: for $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, set

$$\mathcal{F}_{\mathcal{G}} := \{ \text{ spanning rooted forests on } \mathcal{G} \}$$



A spanning rooted forest on a square grid.

$\mathcal{R}_q$ from Random Rooted Forests

Theorem (Random Forest with Determinantal Roots - A.G. (2018))

Fix $q > 0$ and $\mathcal{G}$. Consider the following probability measure on $\mathcal{F}_{\mathcal{G}}$:

$$\nu_q(F) := \frac{q^{\text{nr. of trees}} \prod_{e \in F} w(e)}{Z(q)}, \quad F \in \mathcal{F}_{\mathcal{G}},$$

$\mathcal{R}_q$ from Random Rooted Forests

Theorem (Random Forest with Determinantal Roots - A.G. (2018))

Fix $q > 0$ and $\mathcal{G}$. Consider the following probability measure on $\mathcal{F}_{\mathcal{G}}$:

$$\nu_q(F) := \frac{q^{\text{nr. of trees}} \prod_{e \in F} w(e)}{Z(q)}, \quad F \in \mathcal{F}_{\mathcal{G}},$$

Then,

the set of roots of the random forest with law ν_q
is the determinatal point process $\mathcal{R}_q$ with kernel K_q

Recall:

- $\mathbb{P}(\mathcal{A} \subseteq \mathcal{R}_q) = \det [K_q]_{\mathcal{A}}$, for any $\mathcal{A} \subseteq \mathcal{V}$,
- $K_q(x, y) = \mathbf{P}_x(X(T_q) = y) =$ RW kernel observed at time-scale $1/q$.

$\mathcal{R}_q$ from Random Rooted Forests

Theorem (Random Forest with Determinantal Roots - A.G. (2018))

Fix $q > 0$ and $\mathcal{G}$. Consider the following probability measure on $\mathcal{F}_{\mathcal{G}}$:

$$\nu_q(F) := \frac{q^{\text{nr. of trees}} \prod_{e \in F} w(e)}{Z(q)}, \quad F \in \mathcal{F}_{\mathcal{G}},$$

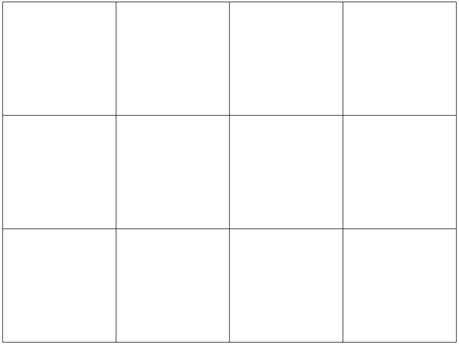
Then,

the set of roots of the random forest with law ν_q
is the determinatal point process $\mathcal{R}_q$ with kernel K_q

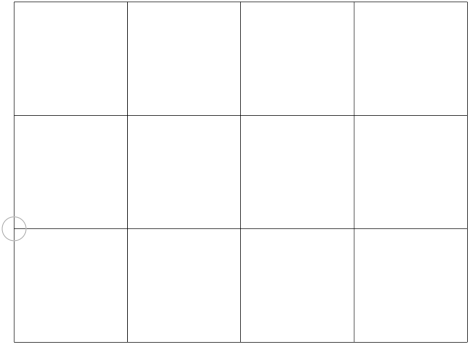
Recall:

- $\mathbb{P}(\mathcal{A} \subseteq \mathcal{R}_q) = \det [K_q]_{\mathcal{A}}$, for any $\mathcal{A} \subseteq \mathcal{V}$,
- $K_q(x, y) = \mathbf{P}_x(X(T_q) = y) =$ RW kernel observed at time-scale $1/q$.

Sampling $\mathcal{R}_q$ with Wilson's algorithm

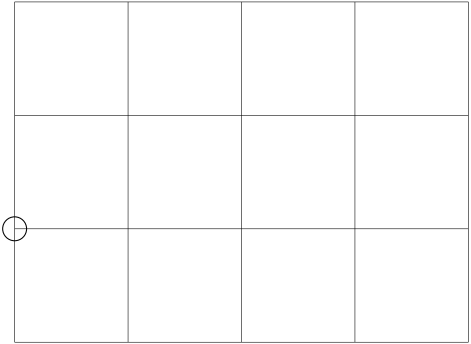


Sampling $\mathcal{R}_q$ with Wilson's algorithm



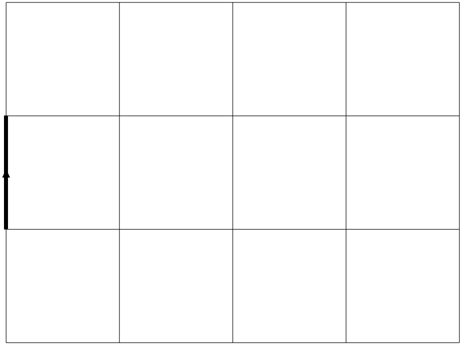
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



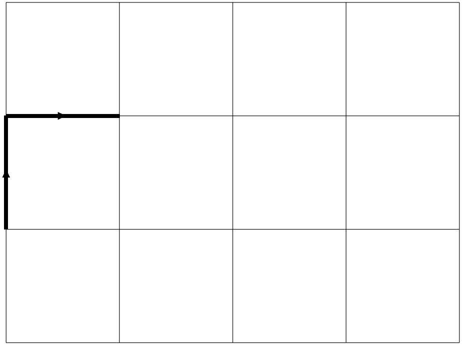
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



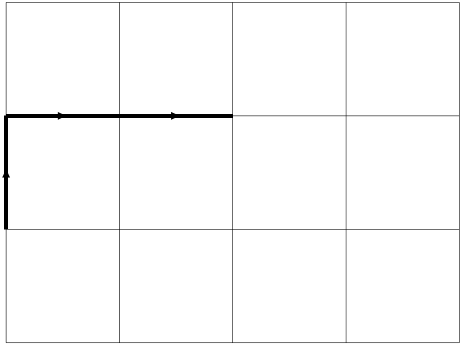
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



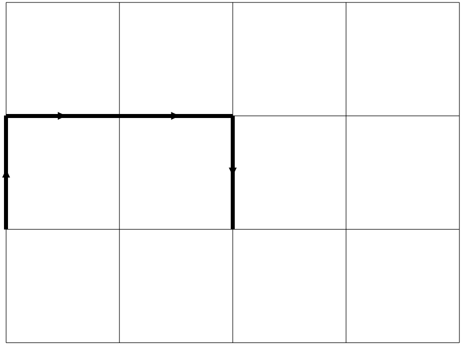
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



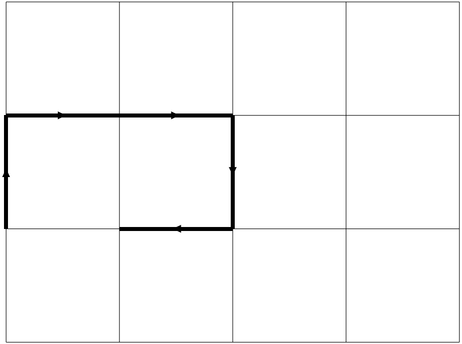
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



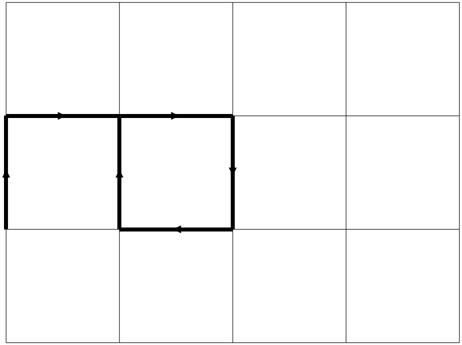
Build dynamically a rooted spanning forest using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



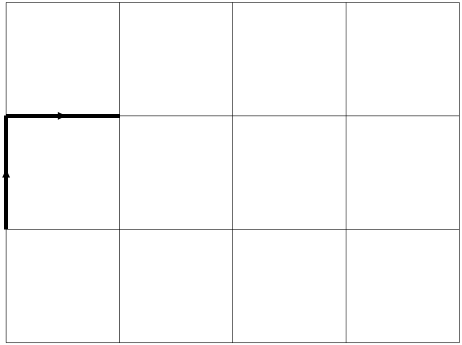
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



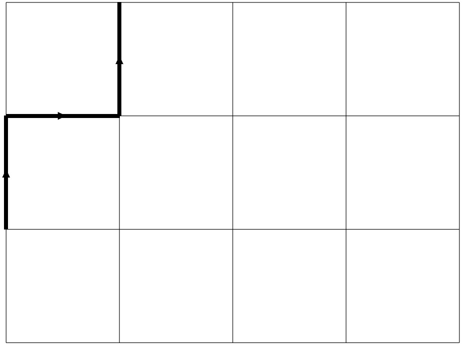
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



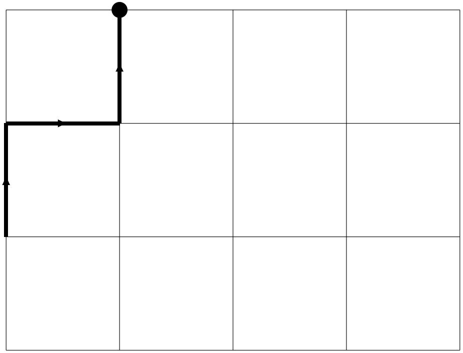
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



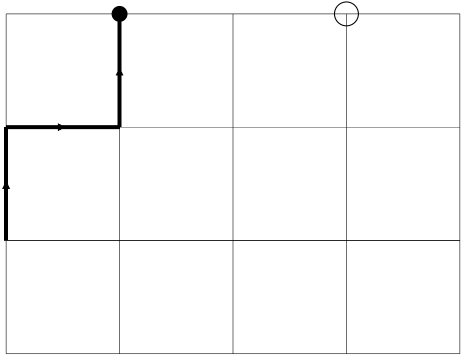
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



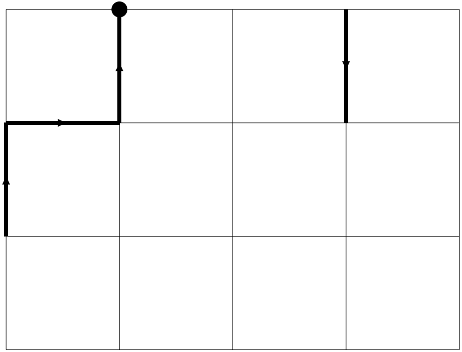
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



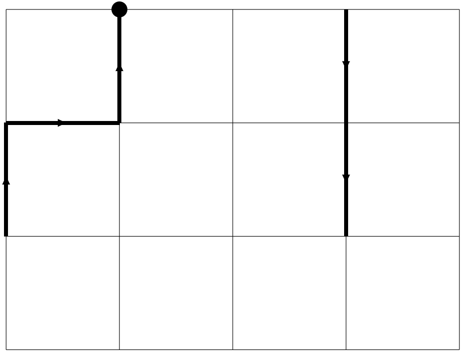
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



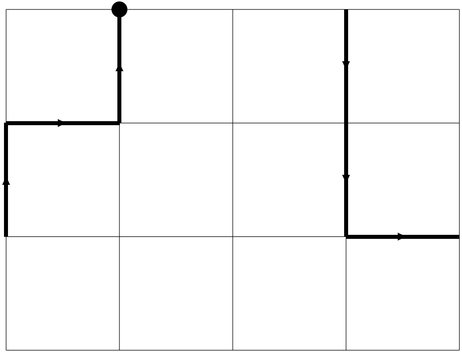
Build dynamically a rooted spanning forest using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



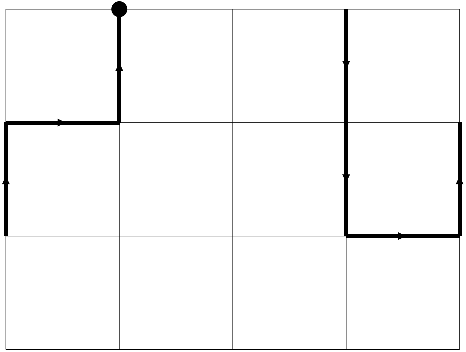
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



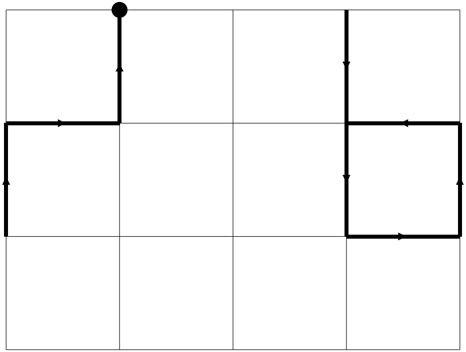
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



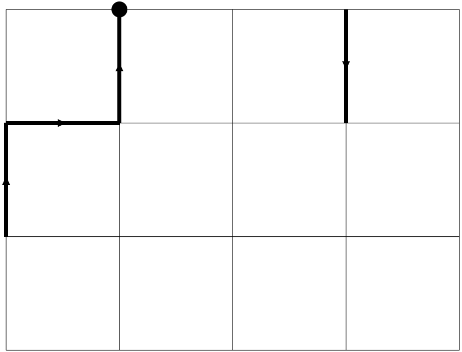
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



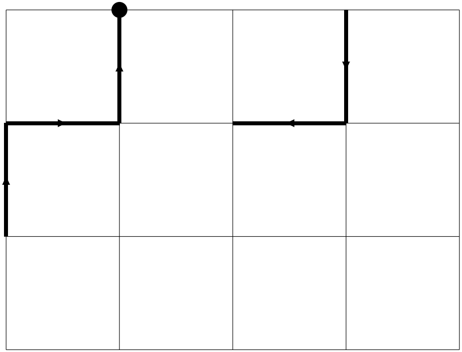
Build dynamically a rooted spanning forest using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



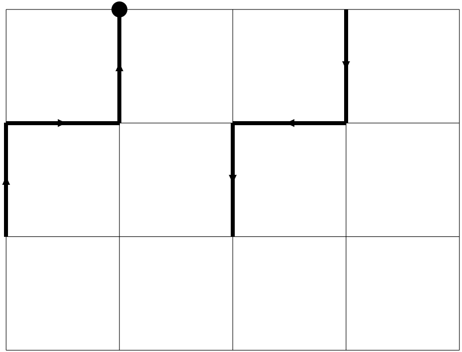
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



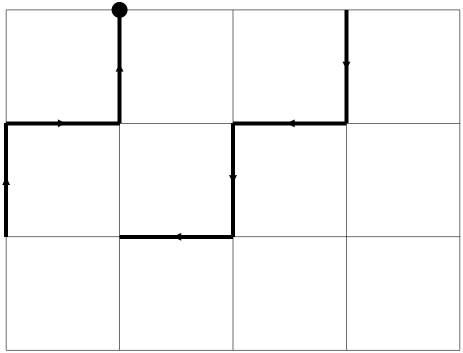
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



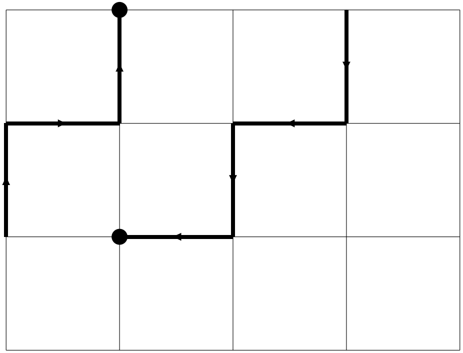
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



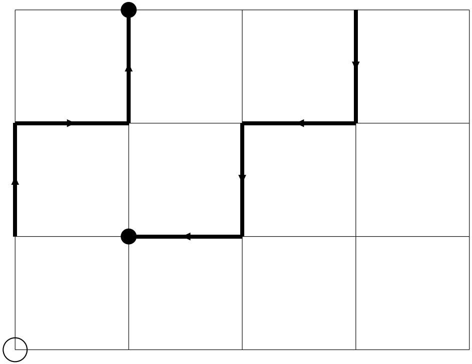
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



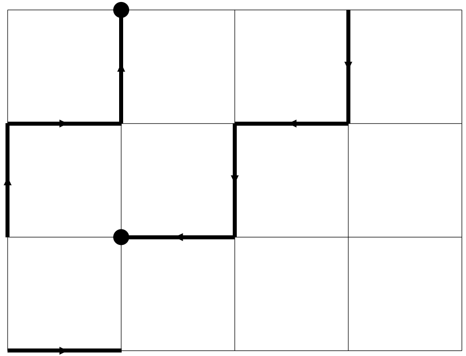
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



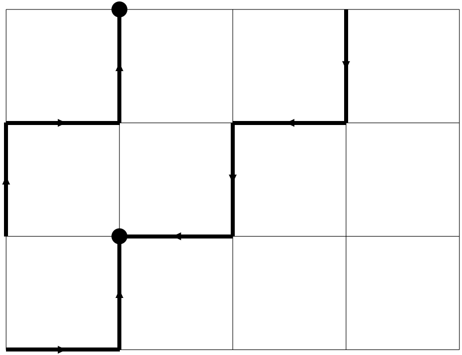
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



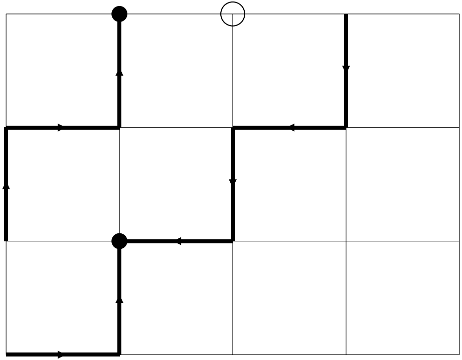
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



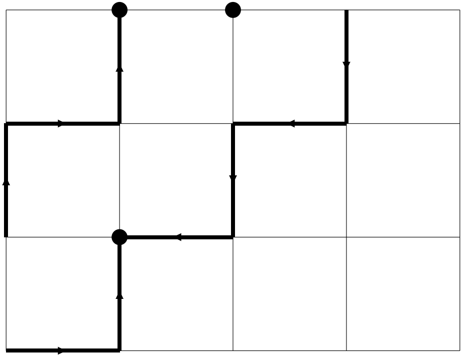
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



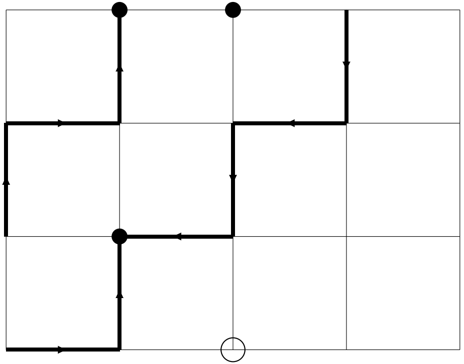
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



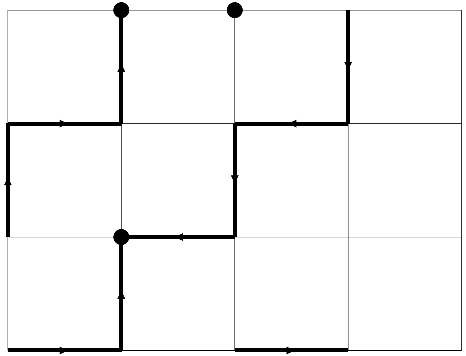
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



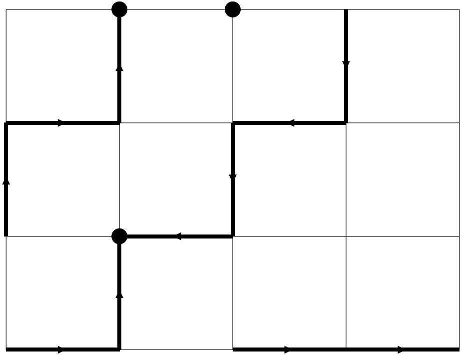
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



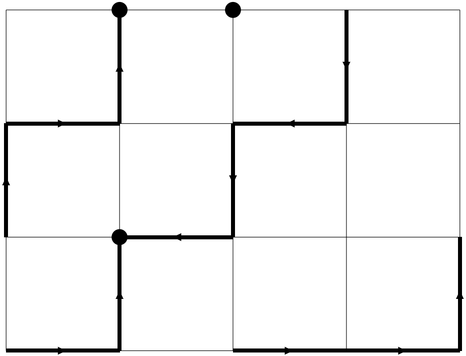
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



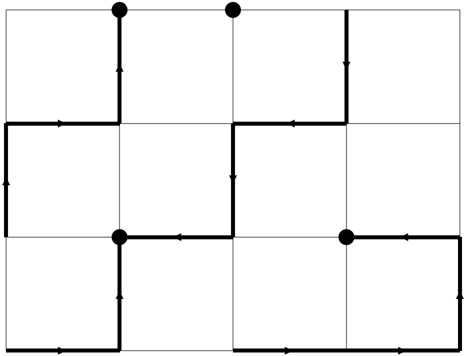
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



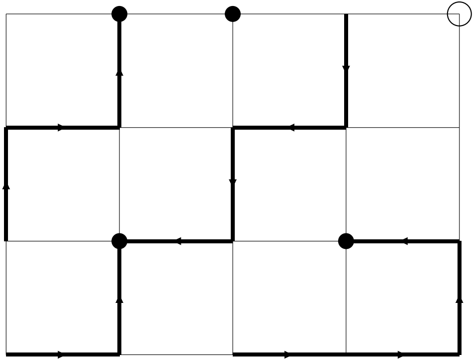
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



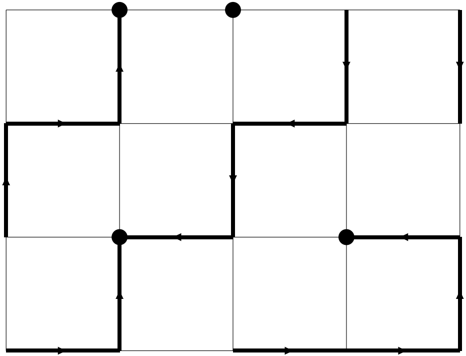
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



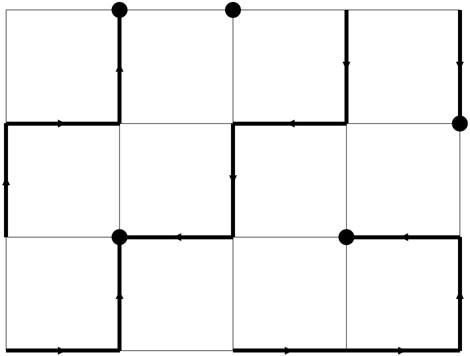
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



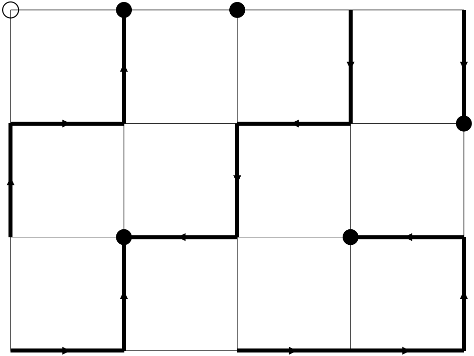
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



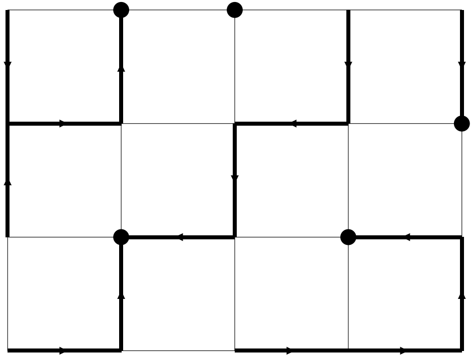
Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

Sampling $\mathcal{R}_q$ with Wilson's algorithm



Build dynamically a rooted spanning forest
using **loop-erased random walks** killed at rate q.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (**Distribution of number of roots - A.G. (2018)**)

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, consider the set of Roots $\mathcal{R}_q$ with kernel K_q. Then **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim Bernoulli\left(\frac{q}{q+\theta_i}\right).$$

with θ_i's **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (Distribution of number of roots - A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, *consider the set of Roots* $\mathcal{R}_q$ *with kernel* K_q. *Then* **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim Bernoulli\left(\frac{q}{q+\theta_i}\right).$$

with θ_i*'s* **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (**Distribution of number of roots - A.G. (2018)**)

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, consider the set of Roots $\mathcal{R}_q$ with kernel K_q. Then **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim \mathit{Bernoulli}\left(\frac{q}{q+\theta_i}\right).$$

with θ_i's **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

Note: $|\mathcal{R}_q| = n$ if $q \to \infty$ and $|\mathcal{R}_q| = 1$ if $q \to 0$.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (**Distribution of number of roots - A.G. (2018)**)

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, consider the set of Roots $\mathcal{R}_q$ with kernel K_q. Then **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim Bernoulli\left(\frac{q}{q+\theta_i}\right).$$

with θ_i's **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

Note: $|\mathcal{R}_q| = n$ if $q \to \infty$ and $|\mathcal{R}_q| = 1$ if $q \to 0$.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (Distribution of number of roots - A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, consider the set of Roots $\mathcal{R}_q$ with kernel K_q. Then **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim Bernoulli\left(\frac{q}{q+\theta_i}\right).$$

with θ_i's **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

Note: $|\mathcal{R}_q| = n$ if $q \to \infty$ and $|\mathcal{R}_q| = 1$ if $q \to 0$.

Efficient sampling of exactly k roots

Wilson's algorithm allows to sample $\mathcal{R}_q$ for fixed q, we developed

- **a coupled algorithm to sample a trajectory** $(\mathcal{R}_q)_{q\in[0,\infty)}$,
- **and $\mathcal{R}_k := \mathcal{R}_q$ conditioned on having k roots** ,
  in $O(n)$ running time if k is a fraction of n.

$\mathcal{R}_q$: a flexible set of well-distributed nodes

Theorem (Distribution of number of roots - A.G. (2018))

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, consider the set of Roots $\mathcal{R}_q$ with kernel K_q. Then **its cardinality** *is a* **non-homogeneous Binomial** *:*

$$|\mathcal{R}_q| = \sum_{i=1}^{n} Y_i, \qquad Y_i \sim Bernoulli\left(\frac{q}{q+\theta_i}\right).$$

with θ_i's **eigenvalues of the graph Laplacian** $-\mathcal{L}_w$.

Note: $|\mathcal{R}_q| = n$ if $q \to \infty$ and $|\mathcal{R}_q| = 1$ if $q \to 0$.

Efficient sampling of exactly k roots

Wilson's algorithm allows to sample $\mathcal{R}_q$ for fixed q, we developed

- **a coupled algorithm to sample a trajectory** $(\mathcal{R}_q)_{q\in[0,\infty)}$,
- **and $\mathcal{R}_k := \mathcal{R}_q$ conditioned on having k roots** ,
  in $O(n)$ running time if k is a fraction of n.

A generalized Page-Rank k-centrality

A generalized Page-Rank k-centrality

Theorem (Roots are at local equilibrium - A.G. (2018))

Fix $k \leq n$, any partition $\{B_1, \ldots, B_k\}$ of $\mathcal{V}$ into k blocks, and any $x_i \in B_i$, for $i = 1, \ldots, k$. Then:

$$\mathbb{P}\Big(\mathcal{R}_k = \{x_1, \cdots, x_k\} \Big| \mathcal{P}_q = \{B_1, \ldots, B_k\}\Big) = \prod_{i=1}^{k} \mu_{B_i}(x_i),$$

where

- $\mathcal{P}_q$ *is the* **partition of $\mathcal{V}$ induced by the forest with law ν_q**,
- μ_{B_i} *is the* **invariant measure** *of the RW X restricted to B_i.*

A generalized Page-Rank k-centrality

Theorem (Roots are at local equilibrium - A.G. (2018))

Fix $k \leq n$, any partition $\{B_1, \ldots, B_k\}$ of $\mathcal{V}$ into k blocks, and any $x_i \in B_i$, for $i = 1, \ldots, k$. Then:

$$\mathbb{P}\Big(\mathcal{R}_k = \{x_1, \cdots, x_k\} \Big| \mathcal{P}_q = \{B_1, \ldots, B_k\}\Big) = \prod_{i=1}^{k} \mu_{B_i}(x_i),$$

where

- $\mathcal{P}_q$ *is the* **partition of $\mathcal{V}$ induced by the forest with law** ν_q,
- μ_{B_i} *is the* **invariant measure** *of the RW X restricted to B_i.*

A generalized Page-Rank k-centrality

Theorem (Roots are at local equilibrium - A.G. (2018))

Fix $k \le n$, any partition $\{B_1, \dots, B_k\}$ of $\mathcal{V}$ into k blocks, and any $x_i \in B_i$, for $i = 1, \dots, k$. Then:

$$\mathbb{P}\Big(\mathcal{R}_k = \{x_1, \cdots, x_k\} \Big| \mathcal{P}_q = \{B_1, \dots, B_k\}\Big) = \prod_{i=1}^{k} \mu_{B_i}(x_i),$$

where

- $\mathcal{P}_q$ *is the* **partition of $\mathcal{V}$ induced by the forest with law** ν_q,
- μ_{B_i} *is the* **invariant measure** *of the RW X restricted to B_i.*

Note:

- For $q \approx 0 \Rightarrow \mathcal{P}_q = \mathcal{V}$ and **the unique root is distributed as the invariant measure** μ of the RW.
- "Freedom to choose" the graph Laplacian.

A generalized Page-Rank k-centrality

Theorem (Roots are at local equilibrium - A.G. (2018))

Fix $k \leq n$, any partition $\{B_1, \ldots, B_k\}$ of $\mathcal{V}$ into k blocks, and any $x_i \in B_i$, for $i = 1, \ldots, k$. Then:

$$\mathbb{P}\Big(\mathcal{R}_k = \{x_1, \cdots, x_k\} \Big| \mathcal{P}_q = \{B_1, \ldots, B_k\}\Big) = \prod_{i=1}^{k} \mu_{B_i}(x_i),$$

where

- $\mathcal{P}_q$ *is the* **partition of $\mathcal{V}$ induced by the forest with law** ν_q,
- μ_{B_i} *is the* **invariant measure** *of the RW X restricted to B_i.*

Note:

- For $q \approx 0 \Rightarrow \mathcal{P}_q = \mathcal{V}$ and **the unique root is distributed as the invariant measure** μ of the RW.
- "Freedom to choose" the graph Laplacian.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n - k$ susceptible ones $\diamond$

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.
2. **How many samples?** Sample L many candidate k-sets $\{\mathcal{R}_{k,i}^{\mathcal{C}} : i \leq L\}$ and compute their ShieldValue and Max-degree with L such that total running cost is as in Netshield $O(m+nk)$.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.
2. **How many samples?** Sample L many candidate k-sets $\{\mathcal{R}_{k,i}^{\mathcal{C}} : i \leq L\}$ and compute their ShieldValue and Max-degree with L such that total running cost is as in Netshield $O(m+nk)$.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.
2. **How many samples?** Sample L many candidate k-sets $\{\mathcal{R}_{k,i}^{\mathcal{C}} : i \leq L\}$ and compute their ShieldValue and Max-degree with L such that total running cost is as in Netshield $O(m+nk)$.
3. **Forest k-set $\mathcal{S}_\star$ to be immunized:** Among the L sampled k-sets. Pick the ones with max degree and max Shield Value. Take also the Netshield output set. Set $\mathcal{S}_\star$ to be the set with maximal eigendrop among these 3 sets.

Forest-immunization (on undirected network)

$\diamond$ **Heuristic:** k best spreaders to be removed should be far from each other but also the remaining $n-k$ susceptible ones $\diamond$

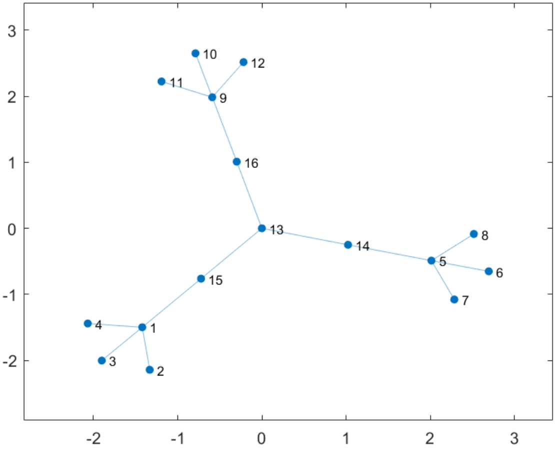
Roots complement-set: $\mathcal{R}_q^{\mathcal{C}} := \mathcal{V} \setminus \mathcal{R}_q$ is **also determinantal** set & it **concentrates around max-degree sets**.

1. **Canidate set of k-spreaders to be removed**: Sample $\mathcal{R}_{n-k}$ roots for the susceptible ones & set its complement $\mathcal{R}_k^{\mathcal{C}}$ to be the candidate k-set of best spreaders.
2. **How many samples?** Sample L many candidate k-sets $\{\mathcal{R}_{k,i}^{\mathcal{C}} : i \leq L\}$ and compute their ShieldValue and Max-degree with L such that total running cost is as in Netshield $O(m+nk)$.
3. **Forest k-set $\mathcal{S}_\star$ to be immunized:** Among the L sampled k-sets. Pick the ones with max degree and max Shield Value. Take also the Netshield output set. Set $\mathcal{S}_\star$ to be the set with maximal eigendrop among these 3 sets.

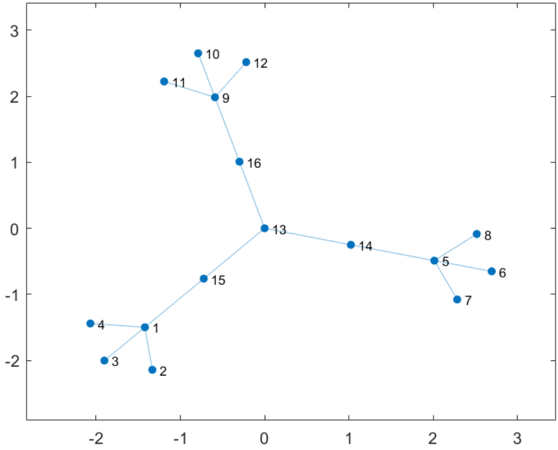
Forest immunization & the geometry of contagion in action:

a few illustrative experiments.

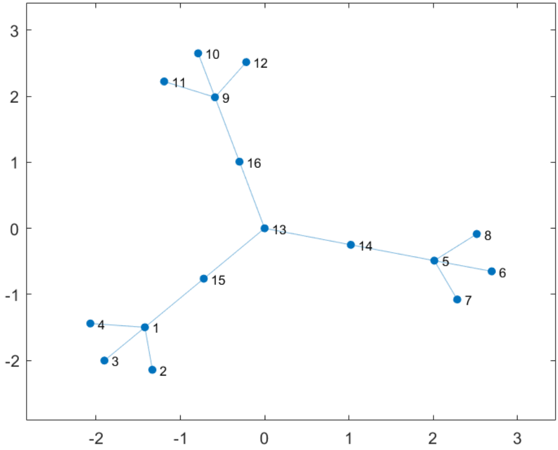
A synthetic insightful example



A synthetic insightful example: $k = 1$

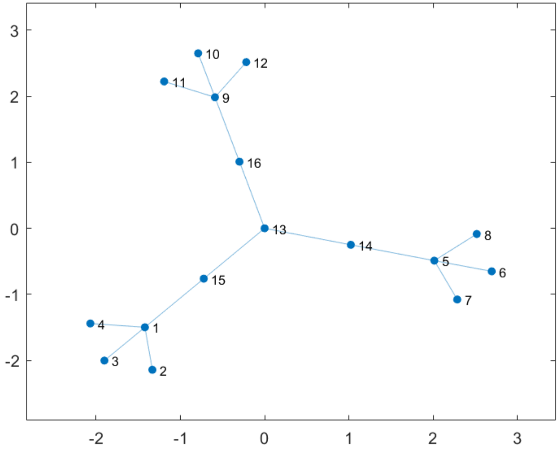


A synthetic insightful example: $k = 1$



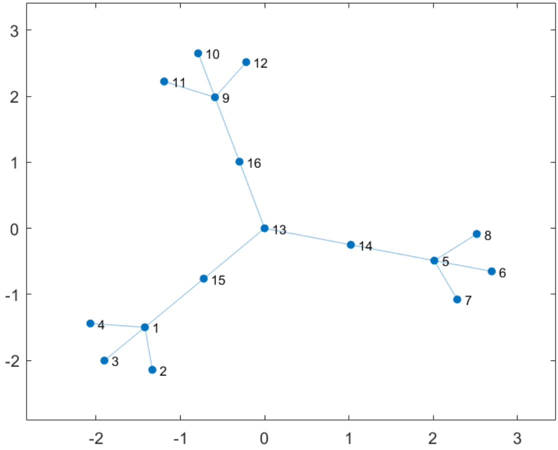
$\Rightarrow$ node with best eigendrop is $\{13\}$: **not the highest degree node.**

A synthetic insightful example: $k = 1$

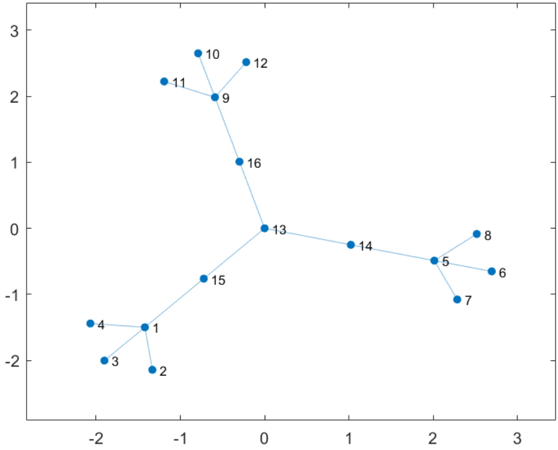


$\Rightarrow$ node with best eigendrop is $\{13\}$: **not the highest degree node.**

A synthetic insightful example: $k = 3$

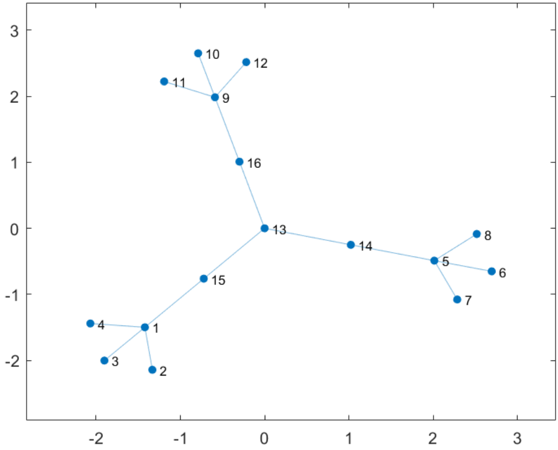


A synthetic insightful example: $k = 3$



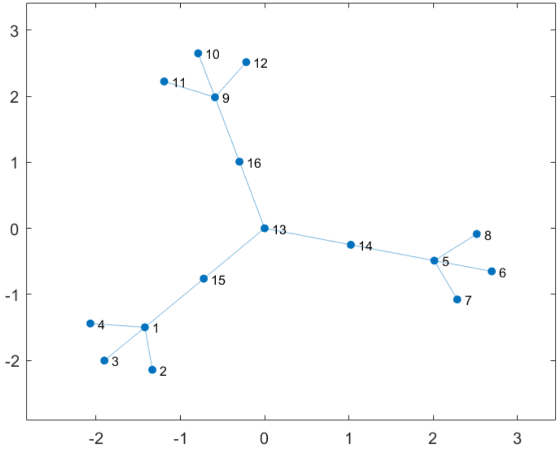
$\Rightarrow$ best triple $\{14, 15, 16\}$: **not the maximal degree set.**

A synthetic insightful example: $k = 3$

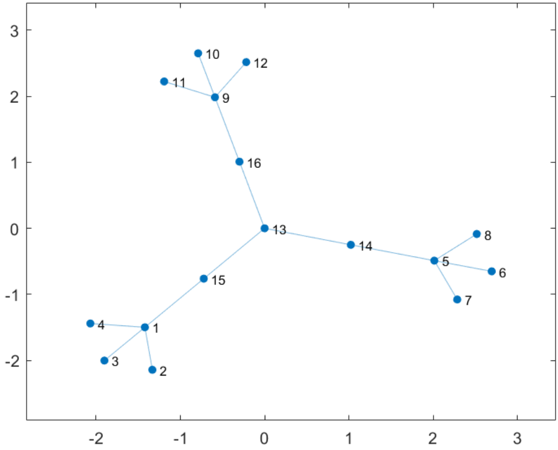


$\Rightarrow$ best triple $\{14, 15, 16\}$: **not the maximal degree set.**

A synthetic insightful example: $k = 4$

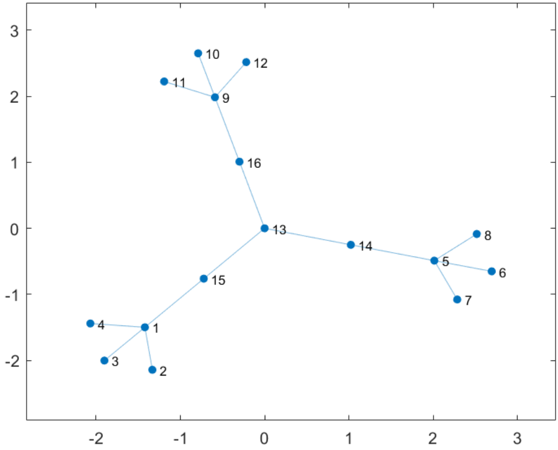


A synthetic insightful example: $k = 4$



$\Rightarrow$ best quaduple $\{13, 1, 5, 9\}$: **global + "periferial" centers.**

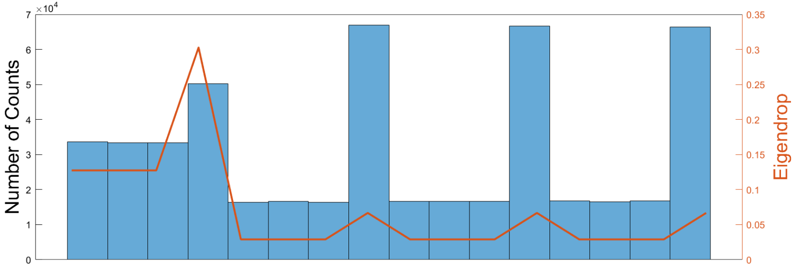
A synthetic insightful example: $k = 4$



$\Rightarrow$ best quaduple $\{13, 1, 5, 9\}$: **global + "periferial" centers.**

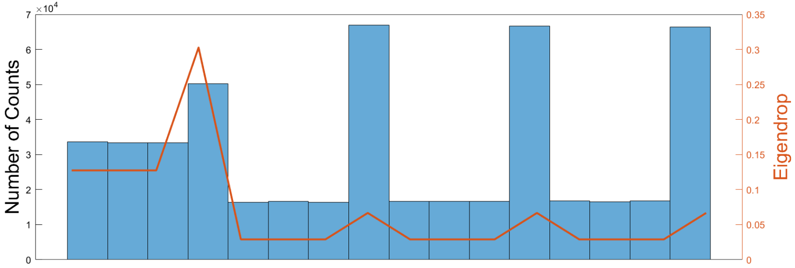
A synthetic insightful example: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$

A synthetic insightful example: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$



$\mathcal{R}_q^{\mathcal{C}}$ **histogram**, **Eigendrop profile**

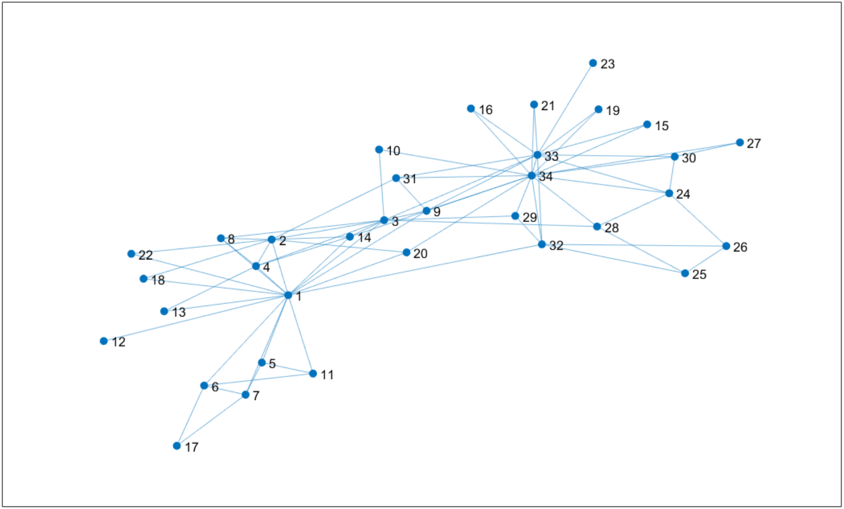
A synthetic insightful example: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$



$\mathcal{R}_q^{\mathcal{C}}$ **histogram**, **Eigendrop profile**

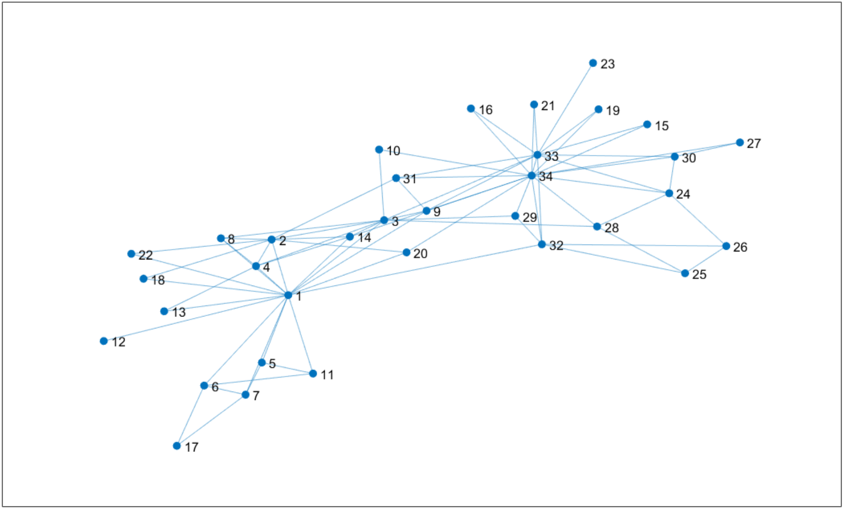
The Karate club: "a fighting group"

The Karate club: "a fighting group"



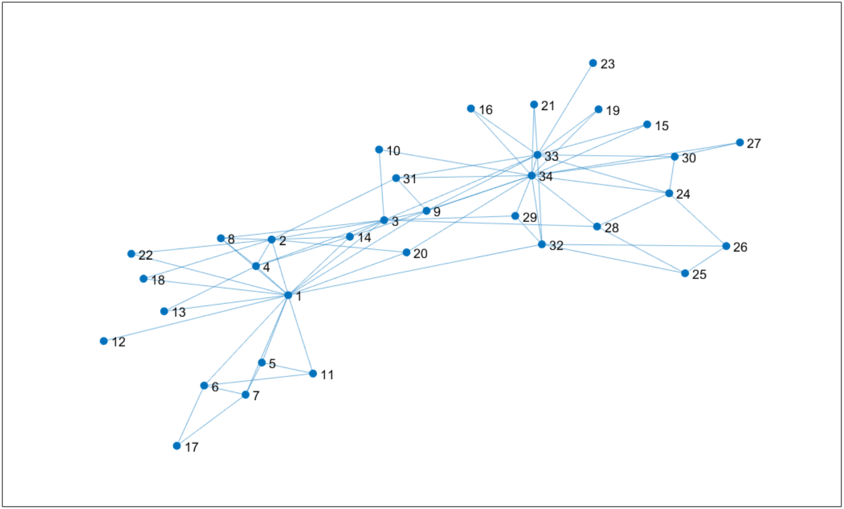
President $\{1\}$ Vs Instructor $\{34\}$ led to a split into 2 groups.

The Karate club: “a fighting group”

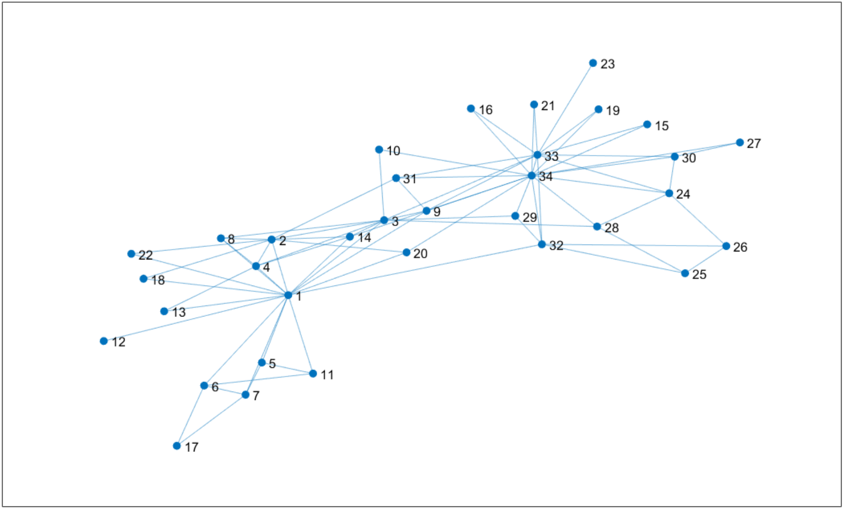


President $\{1\}$ Vs Instructor $\{34\}$ led to a split into 2 groups.

The Karate club: $k = 1$

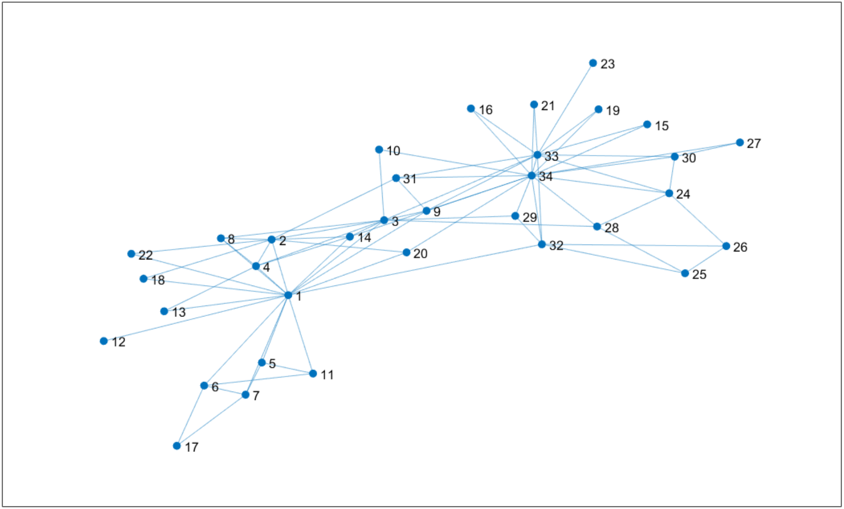


The Karate club: $k = 1$



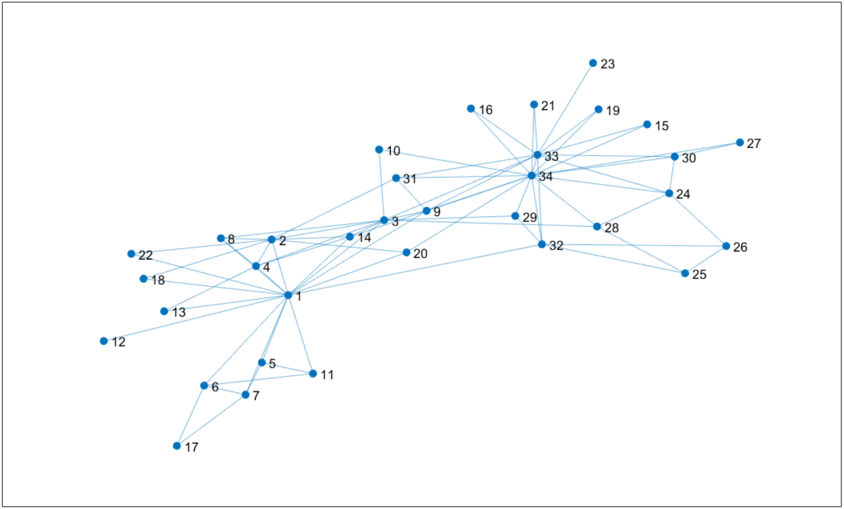
$\Rightarrow$ best node is $\{3\}$: **neither the president nor the instructor.**

The Karate club: $k = 1$



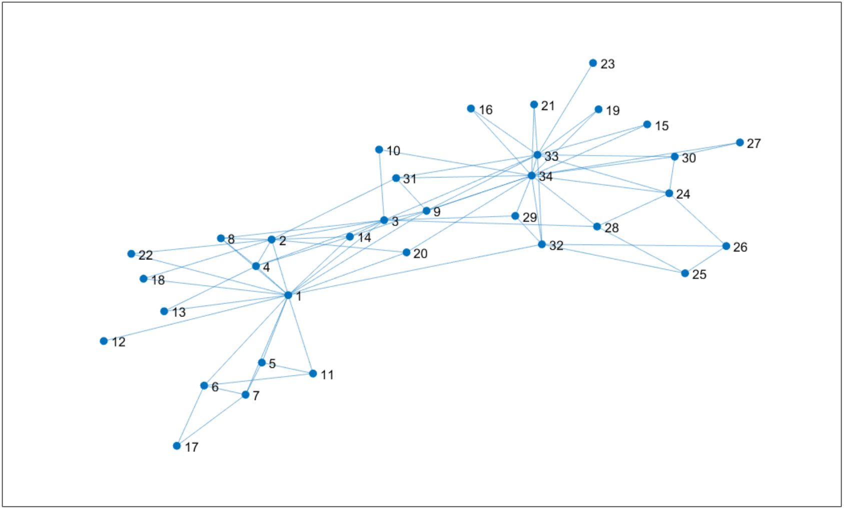
$\Rightarrow$ best node is $\{3\}$: **neither the president nor the instructor.**

The Karate club: $k = 2$



$\Rightarrow$ best pair is $\{1, 34\}$: **the president and the instructor.**

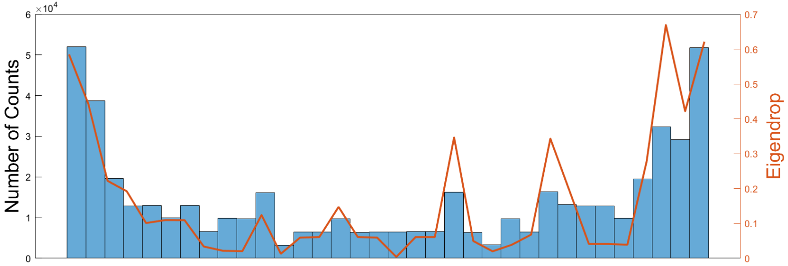
The Karate club: $k = 3$



$\Rightarrow$ best triple is $\{1, 3, 34\}$: **"the fighters and the philanthropist".**

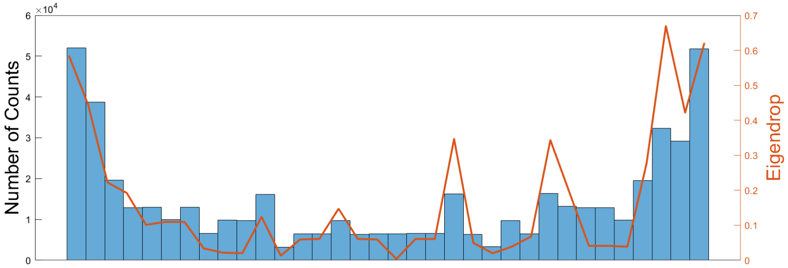
The Karate club: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$

The Karate club: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$



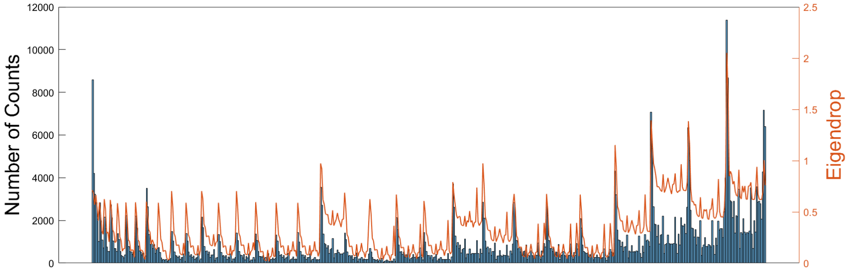
$\mathcal{R}_q^{\mathcal{C}}$ **histogram**, **Eigendrop profile**

The Karate club: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 1$



$\mathcal{R}_q^{\mathcal{C}}$ **histogram**, **Eigendrop profile**

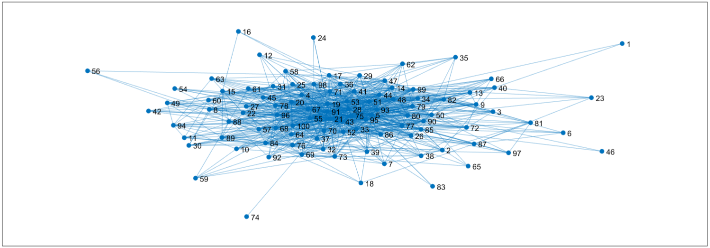
The Karate club: $\mathcal{R}_q^{\mathcal{C}}$ for $k = 2$



$\mathcal{R}_q^{\mathcal{C}}$ **histogram**, **Eigendrop profile**

Conference Interaction: a weighted example

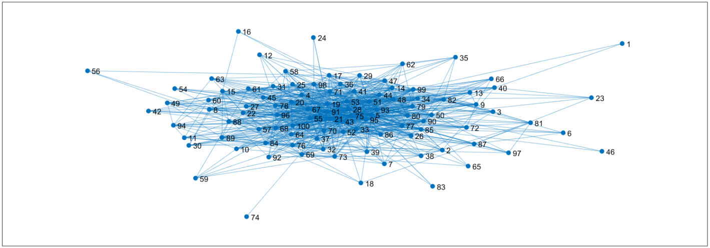
Conference Interaction: a weighted example



$|\mathcal{V}| = 10^2$ individuals, $\quad |\mathcal{E}| \approx 10^3$ interactions in one day,

On such large non-regular networks benchmarks (and similar ones, e.g. airport, etc...) the **forest-immunization can find better solutions than Nethsield in comparable running time.**

Conference Interaction: a weighted example



$|\mathcal{V}| = 10^2$ individuals, $\qquad |\mathcal{E}| \approx 10^3$ interactions in one day,

On such large non-regular networks benchmarks (and similar ones, e.g. airport, etc...) the **forest-immunization can find better solutions than Nethsield in comparable running time.**

References & THANK YOU

• **(Survey on epidemics on networks)** **R. Pastor-Satorras, C. Castellano, P. Van Mieghem and A. Vespignani**, *Epidemic processes in complex networks*, Reviews of Modern Physics 87 (2015).

• **(Max eigenvalue for SIS and SIR)** **D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec and C. Faloutsos**, *Epidemic thresholds in real networks*, ACM Transactions on Information and System Security (2008).

• **(Netshield)** **C. Chen, H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos and D. H. Chau**, *Node immunization on large graphs: theory and algorithms*, IEEE Transactions on Knowledge and Data Engineering 28 (2016).

• **(Roots theory)** **L. A. and A. Gaudillière**, *Two applications of random spanning forests*, Journal of Theoretical Probability 31 (2018).

• **(Survey on rooted forests)** **L. A. , F. Castell, A. Gaudillière and C. Mélot**, *Random forests and networks analysis*, Journal of Statistical Physics 173 (2018).

• **(Forest immunization)** **L. A. , M. Emmerich, A. Gaudillière and I. Gurewitsch** to appear (2022+).