Distributed detection of topological changes in communication networks

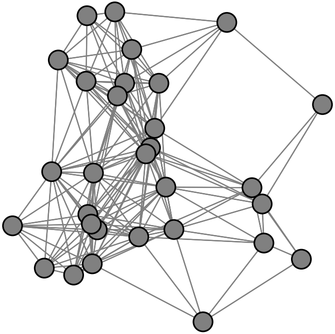
Riccardo Lucchese, **Damiano Varagnolo**, Karl H. Johansson

LULEÅ UNIVERSITY OF TECHNOLOGY

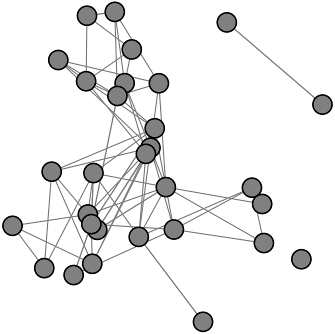
KTH VETENSKAP OCH KONST

Thanks to. . .

The need: detecting changes in topological networks



The need: detecting changes in topological networks



Literature review

I.e., potential solutions

Main idea: *iterate topology estimation routines*

Literature review

I.e., potential solutions

Main idea: *iterate topology estimation routines*

Transmit tables of IDs

Pros: perfect reconstruction / detection

Cons: *not scalable*

Literature review

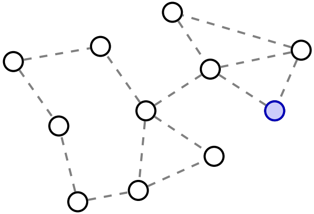
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

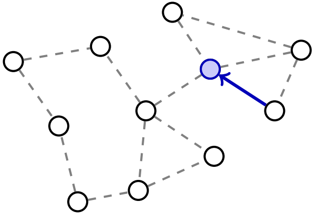
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

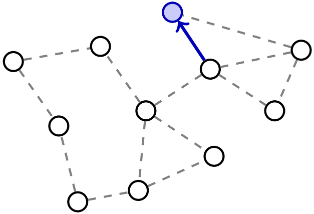
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*

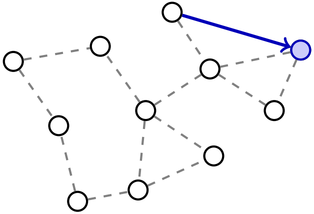


Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

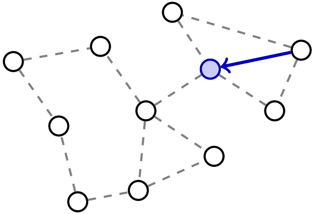
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

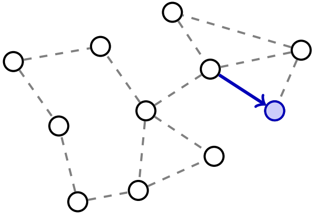
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*

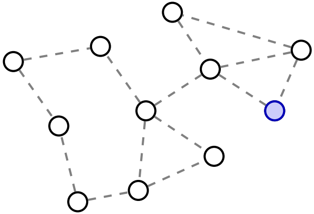


Main idea: *iterate topology estimation routines*

Exploit Random-Walks schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

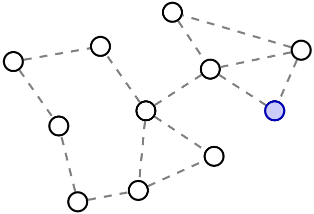
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

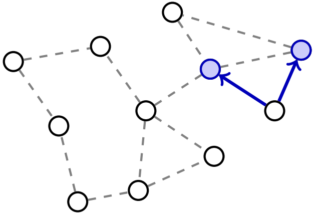
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

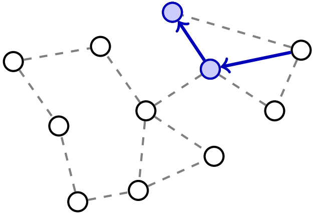
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

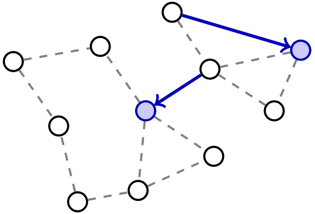
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

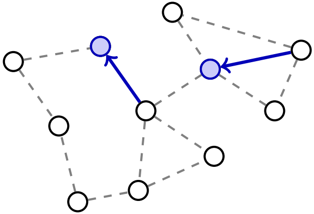
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

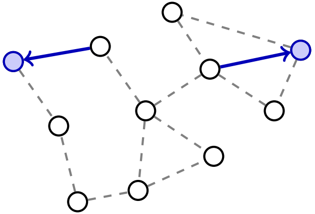
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Literature review

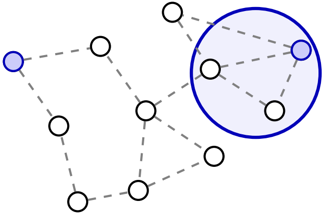
I.e., potential solutions

Main idea: *iterate topology estimation routines*

Exploit Capture-Recapture schemes

Pros: scalable

Cons: *not entirely distributed*



Our contributions

A topology change detector* that is:

- a Generalized Likelihood Ratio (GLR) test
- truly distributed
- scalable and fast *(based on max-consensus)*

Our contributions

A topology change detector* that is:

- a Generalized Likelihood Ratio (GLR) test
- truly distributed
- scalable and fast *(based on max-consensus)*

In this presentation:

- algorithm
- statistical characterization
- experiments

Preliminaries

Preliminaries – Notation on Hypothesis Testing

$$\mathcal{H}_0 = \{f \sim p_\theta \text{ with } \theta \in \Omega_0\}$$
$$\mathcal{H}_1 = \{f \sim p_\theta \text{ with } \theta \in \Omega_1\}$$
$$\Omega_1 = \Omega_0^c$$

Preliminaries – Notation on Hypothesis Testing

$$\mathcal{H}_0 = \{f \sim p_\theta \text{ with } \theta \in \Omega_0\}$$
$$\mathcal{H}_1 = \{f \sim p_\theta \text{ with } \theta \in \Omega_1\}$$
$$\Omega_1 = \Omega_0^c$$

$$g\,(f) : \text{range}(f) \mapsto \{0, 1\}$$

$g(f) = 1$ under $\mathcal{H}_0$: error of type I *(false positive)*

$g(f) = 0$ under $\mathcal{H}_1$: error of type II *(false negative)*

Preliminaries – Notation on Hypothesis Testing

$$\mathcal{H}_0 = \{f \sim p_\theta \text{ with } \theta \in \Omega_0\}$$
$$\mathcal{H}_1 = \{f \sim p_\theta \text{ with } \theta \in \Omega_1\}$$
$$\Omega_1 = \Omega_0^c$$

$$g\,(f) : \text{range}(f) \mapsto \{0, 1\}$$

$g(f) = 1$ under $\mathcal{H}_0$: error of type I *(false positive)*

$g(f) = 0$ under $\mathcal{H}_1$: error of type II *(false negative)*

$$R := \{f \text{ s.t. } g\,(f) = 0\}$$
$$R^c := \{f \text{ s.t. } g\,(f) = 1\}$$

Preliminaries – Notation on Hypothesis Testing

$$\mathcal{H}_0 = \{f \sim p_\theta \text{ with } \theta \in \Omega_0\}$$
$$\mathcal{H}_1 = \{f \sim p_\theta \text{ with } \theta \in \Omega_1\}$$
$$\Omega_1 = \Omega_0^c$$

$$g\left(f\right) : \text{range}(f) \mapsto \{0, 1\}$$

$g(f) = 1$ under $\mathcal{H}_0$: error of type I *(false positive)*

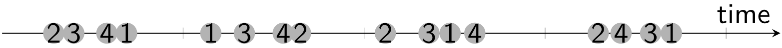
$g(f) = 0$ under $\mathcal{H}_1$: error of type II *(false negative)*

$$R := \{f \text{ s.t. } g\left(f\right) = 0\}$$
$$R^c := \{f \text{ s.t. } g\left(f\right) = 1\}$$

$$\beta_g\left(\theta\right) := \mathbb{P}\left[f \in R^c \;;\; \theta\right]$$
$$\alpha_0\left(g\right) := \sup_{\theta \in \Omega_0} \beta_g\left(\theta\right)$$

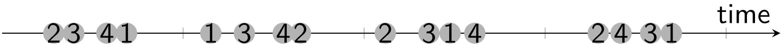
Preliminaries – Notation on Graphs

Very important assumption: synchronous communications



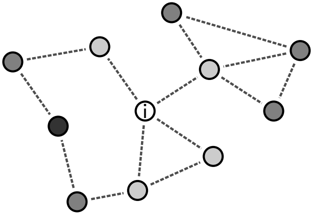
Preliminaries – Notation on Graphs

Very important assumption: synchronous communications



Considered concepts:

- k-steps neighbors
- links among k-steps neighbors



Problem Formulation (simplified)

$S_k^{(i)}(t) :=$ size of k-steps neighborhood of node i at time t

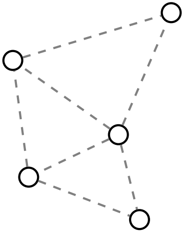
$$\begin{cases} \mathcal{H}_0 : & S_k^{(i)}(t-N) = \ldots = S_k^{(i)}(t-1) = \overline{S}, \quad S_k^{(i)}(t) \geq \sigma \overline{S} \\ \mathcal{H}_1 : & S_k^{(i)}(t-N) = \ldots = S_k^{(i)}(t-1) = \overline{S}, \quad S_k^{(i)}(t) < \sigma \overline{S} \end{cases}$$

Parameters:

- σ *(relative amplitude of change)*
- N *(horizon)*

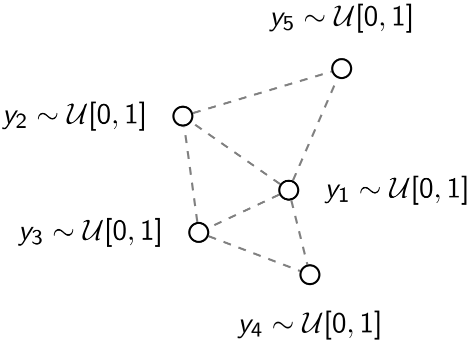
Algorithms

Size Estimation with Max Consensus



Size Estimation with Max Consensus

i.i.d. local generation

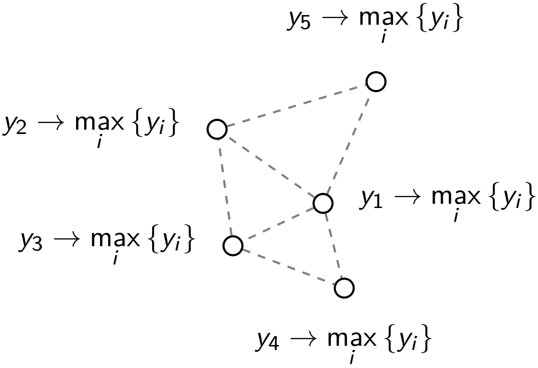


Size Estimation with Max Consensus

i.i.d. local generation

↓

max consensus

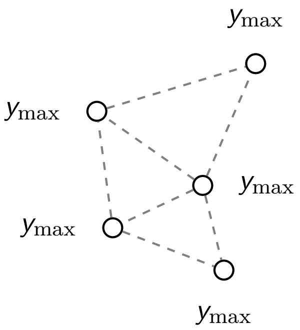


Size Estimation with Max Consensus

i.i.d. local generation

↓

max consensus



Size Estimation with Max Consensus

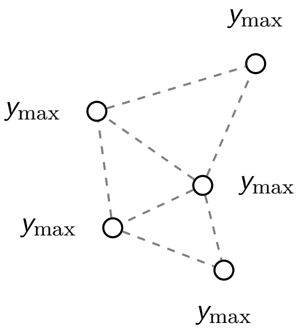
i.i.d. local generation

↓

max consensus

↓

$-\log(y_{\max}) \sim \exp(S)$



Size Estimation with Max Consensus

i.i.d. local generation

↓

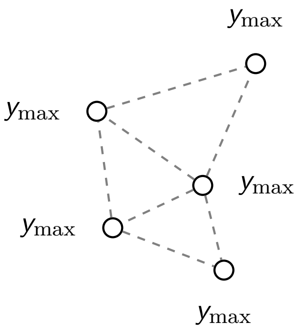
max consensus

↓

$-\log(y_{\max}) \sim \exp(S)$

↓

estimate S with statistical inference



Size Estimation with Max Consensus

i.i.d. local generation

↓

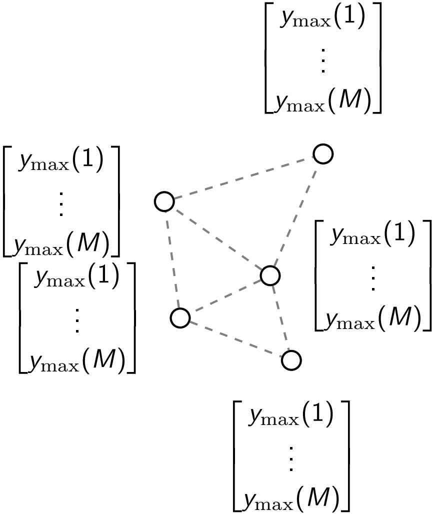
max consensus

↓

$-\log(y_{\max}) \sim \exp(S)$

↓

estimate S with statistical inference



Characteristics

(under no-quantization assumptions)

- ML estimator: $\widehat{S} = \left(-\frac{1}{M} \sum_{m=1}^{M} \log \left(y_{\max}(m) \right) \right)^{-1}$

- $\frac{\widehat{S}}{SM} \sim \text{Inv-Gamma}(M, 1)$

- $\mathbb{E}\left[\frac{\widehat{S}}{S} \right] = \frac{M}{M-1}$

- $\text{var}\left(\frac{\widehat{S} - S}{S} \right) \approx \frac{1}{M}$

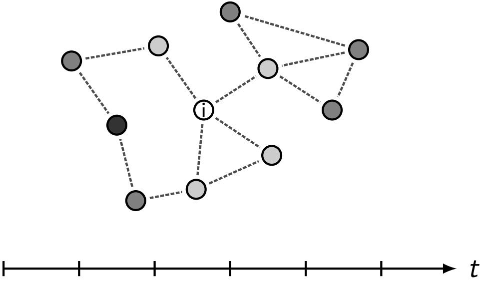
Characteristics

(under no-quantization assumptions)

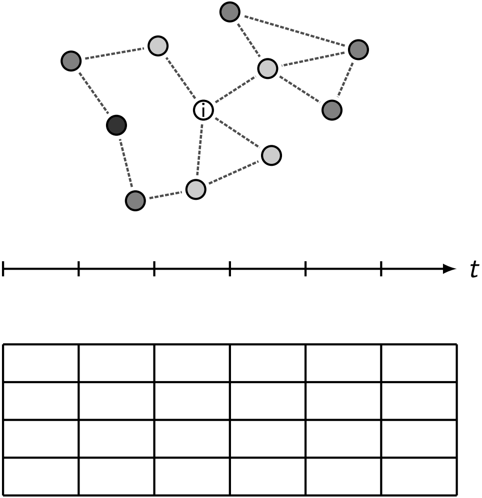
- ML estimator: $\widehat{S} = \left(-\frac{1}{M} \sum_{m=1}^{M} \log\left(y_{\max}(m) \right) \right)^{-1}$
- $\frac{\widehat{S}}{SM} \sim \text{Inv-Gamma}(M, 1)$
- $\mathbb{E}\left[\frac{\widehat{S}}{S} \right] = \frac{M}{M-1}$
- $\operatorname{var}\left(\frac{\widehat{S} - S}{S} \right) \approx \frac{1}{M}$

M trades off performance vs. communications

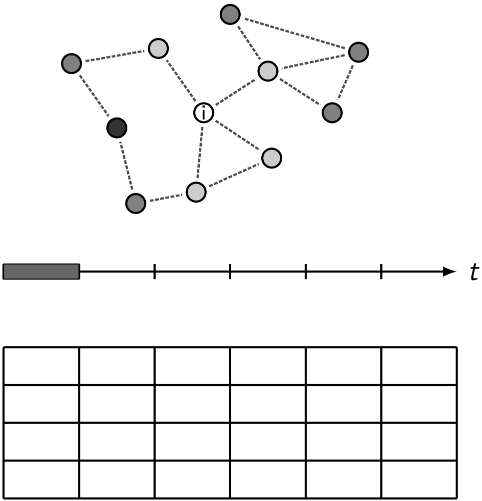
Extension: Size Estimation of k-steps Neighborhoods



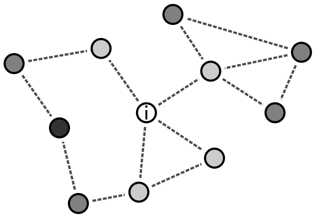
Extension: Size Estimation of k-steps Neighborhoods

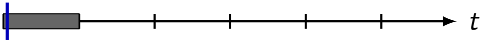


Extension: Size Estimation of *k*-steps Neighborhoods



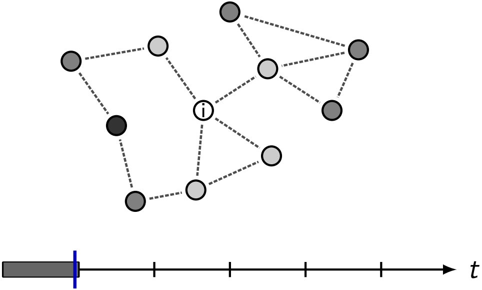
Extension: Size Estimation of k-steps Neighborhoods





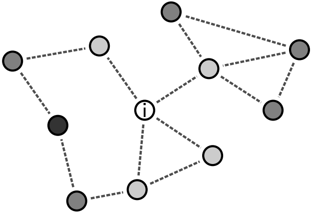
0.1					
0.5					
0.7					
0.3					

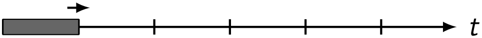
Extension: Size Estimation of *k*-steps Neighborhoods



0.6					
0.7					
0.9					
0.5					

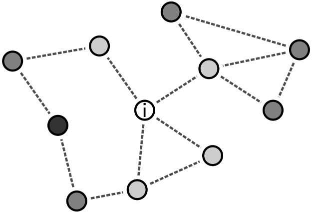
Extension: Size Estimation of k-steps Neighborhoods





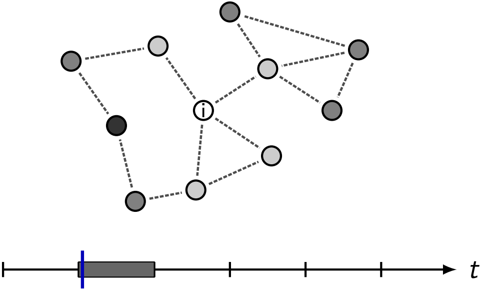
0.6					
0.7					
0.9					
0.5					

Extension: Size Estimation of k-steps Neighborhoods



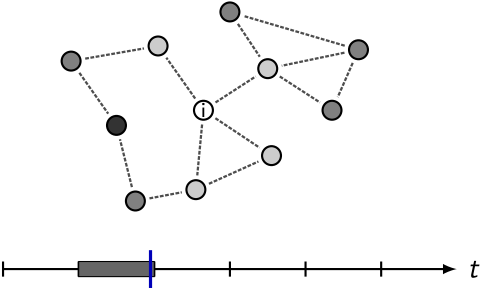
	0.6				
	0.7				
	0.9				
	0.5				

Extension: Size Estimation of *k*-steps Neighborhoods



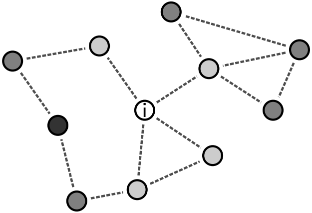
0.4	0.6				
0.3	0.7				
0.6	0.9				
0.5	0.5				

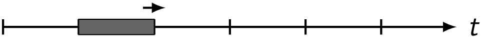
Extension: Size Estimation of *k*-steps Neighborhoods



0.4	0.8				
0.5	0.8				
0.7	0.9				
0.6	0.7				

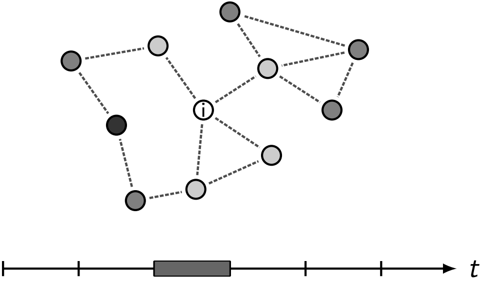
Extension: Size Estimation of *k*-steps Neighborhoods





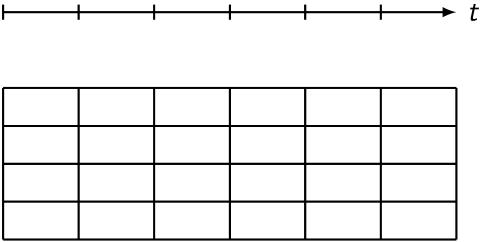
0.4	0.8				
0.5	0.8				
0.7	0.9				
0.6	0.7				

Extension: Size Estimation of *k*-steps Neighborhoods

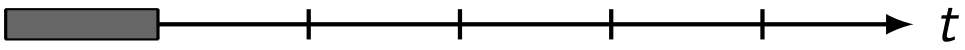


	0.4	0.8			
	0.5	0.8			
	0.7	0.9			
	0.6	0.7			

Extension: k-steps Neighborhood Size Change-Detection



Extension: k-steps Neighborhood Size Change-Detection

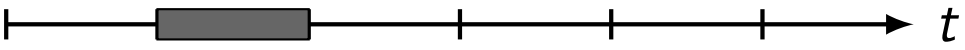


0.4	0.5	0.6	0.7	0.8	0.9
0.3	0.4	0.5	0.6	0.7	0.8
0.4	0.4	0.7	0.7	0.8	0.8
0.6	0.6	0.6	0.9	0.9	0.9

$$\chi := -\frac{1}{M}\sum_{m}\log\left(y_{\max}(m)\right) = \widehat{S}^{-1}$$

0.33

Extension: k-steps Neighborhood Size Change-Detection

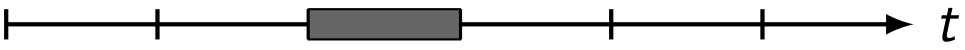


0.3	0.4	0.5	0.6	0.7	0.8
0.4	0.5	0.6	0.7	0.8	0.9
0.6	0.6	0.6	0.9	0.9	0.9
0.4	0.4	0.7	0.7	0.8	0.8

$$\chi := -\frac{1}{M}\sum_{m}\log\left(y_{\max}(m)\right) = \widehat{S}^{-1}$$

0.33 0.33

Extension: k-steps Neighborhood Size Change-Detection



0.3	0.4	0.5	0.6	0.8	0.9
0.6	0.6	0.6	0.9	0.9	0.9
0.2	0.5	0.6	0.7	0.8	0.9
0.4	0.4	0.7	0.8	0.8	0.8

$$\chi := -\frac{1}{M}\sum_{m}\log\left(y_{\max}(m)\right) = \widehat{S}^{-1}$$

0.33 0.33 0.3

Extension: k-steps Neighborhood Size Change-Detection

0.6	0.6	0.6	0.9	0.9	0.9
0.2	0.5	0.6	0.7	0.8	0.9
0.3	0.4	0.5	0.6	0.9	0.9
0.4	0.4	0.7	0.8	0.8	0.8

$$\chi := -\frac{1}{M}\sum_{m}\log\left(y_{\max}(m)\right) = \widehat{S}^{-1}$$

0.33 0.33 0.3 0.3

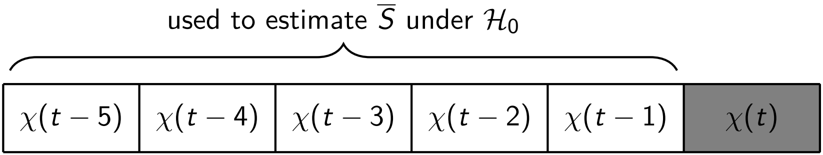
k-steps Neighborhood Size Change-Detection

$$\begin{cases} \mathcal{H}_0 : & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) \geq \sigma \overline{S} \\ \mathcal{H}_1 : & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) < \sigma \overline{S} \end{cases}$$

$\chi(t-5)$	$\chi(t-4)$	$\chi(t-3)$	$\chi(t-2)$	$\chi(t-1)$	$\chi(t)$

k-steps Neighborhood Size Change-Detection

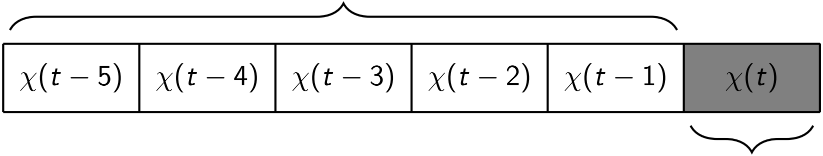
$$\begin{cases} \mathcal{H}_0: & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) \geq \sigma \overline{S} \\ \mathcal{H}_1: & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) < \sigma \overline{S} \end{cases}$$



k-steps Neighborhood Size Change-Detection

$$\begin{cases} \mathcal{H}_0: & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) \geq \sigma \overline{S} \\ \mathcal{H}_1: & S(t-N) = \ldots = S(t-1) = \overline{S} \qquad S(t) < \sigma \overline{S} \end{cases}$$

used to estimate $\overline{S}$ under $\mathcal{H}_0$



used to estimate $S(t)$ under no hypotheses or $\mathcal{H}_0$

k-steps Neighborhood Size Change-Detection

1. *(estimation of the pre-change value)*

$$\overline{S} = \left(\frac{1}{N} \sum_{\tau=t-N}^{t-1} \chi(\tau) \right)^{-1}$$

2. *(estimation of the post-change value)*

$$\widehat{S}(t) = \chi(t)^{-1}$$

$$\widehat{S}_0(t) = \begin{cases} \widehat{S}(t) \text{ if } \widehat{S}(t) \geq \sigma \overline{S} \\ \sigma \overline{S} \text{ otherwise} \end{cases}$$

3. *(computation of the log-GLR)*

$$\Lambda = M \log \left(\frac{\widehat{S}_0(t)}{\widehat{S}(t)} \right) - \left(\widehat{S}_0(t) - \widehat{S}(t) \right) \chi(t)$$

4. *(decision between $\mathcal{H}_0$ and $\mathcal{H}_1$)*

$$g(f) = \begin{cases} 0 & \text{if } \Lambda \geq \lambda \\ 1 & \text{otherwise} \end{cases}$$

(how to compute $\lambda \rightarrow$ in 2 slides)

Neighborhood Size Change-Detection – General

$\chi(t-5)$	$\chi(t-4)$	$\chi(t-3)$	$\chi(t-2)$	$\chi(t-1)$	$\chi(t)$

$$\begin{cases} \mathcal{H}_0: & S(t-N) = \ldots = S(t-T) = \overline{S} \\ & \forall \tau \in \{t-T+1, \ldots, 0\} \quad S(t-\tau) \geq \sigma \overline{S} \\ \mathcal{H}_1: & S(t-N) = \ldots = S(t-T) = \overline{S} \\ & \exists \tau \in \{t-T+1, \ldots, 0\} \text{ s.t. } S(t) < \sigma \overline{S} \end{cases}$$

Parameters:

- σ *(relative amplitude of change)*
- N *(outer horizon)*
- T ***(inner horizon)***

Neighborhood Size Change-Detection – General

$\chi(t-5)$	$\chi(t-4)$	$\chi(t-3)$	$\chi(t-2)$	$\chi(t-1)$	$\chi(t)$

$$\begin{cases} \mathcal{H}_0: & S(t-N) = \ldots = S(t-T) = \overline{S} \\ & \forall \tau \in \{t-T+1, \ldots, 0\} \quad S(t-\tau) \geq \sigma \overline{S} \\ \mathcal{H}_1: & S(t-N) = \ldots = S(t-T) = \overline{S} \\ & \exists \tau \in \{t-T+1, \ldots, 0\} \text{ s.t. } S(t) < \sigma \overline{S} \end{cases}$$

Parameters:

- σ *(relative amplitude of change)*
- N *(outer horizon)*
- T ***(inner horizon)***

Algorithm: parallelize the previous one!

Characterization

Computation of the Thresholds - General Case

1. set set $q_1 = 1 - \frac{\Gamma(M, M)}{\Gamma(M)}$ with $\Gamma(a, b) =$ upper incomplete Gamma function, $\Gamma(a) =$ Gamma function
2. set $p_1(\cdot) = \mathrm{Gamma}\left(M, M^{-1}\right)$
3. evaluate the Lambert W-function in the interval $\left(-\frac{1}{e}, 0\right)$
4. set $p_2(a) = p_1\left(-W\left(-e^{a-1}\right)\right) W'\left(-e^{a-1}\right) e^{a-1}$
5. set $p_\nu(a)$ as the mixed probability density and mass function of

$$\nu = \begin{cases} 1 & \text{with mass } q_1 \\ a \in (0, 1) & \text{with density } p_2(a)/q_1 \end{cases}$$

6. compute the mixed probability density and mass function of ω as

$$p_\omega(\cdot) = \overbrace{p_\nu(\cdot) * \cdots * p_\nu(\cdot)}^{T \text{ times}}$$

7. compute the quantile function of ω, $F_\omega^{-1}(\cdot)$
8. set $\lambda_T = F_{\boldsymbol{\omega}}^{-1}(\alpha_0)$

Computation of the Power - General Case

$$\beta_g^r(\kappa, M) := \mathbb{P}\left[\boldsymbol{f} \in R^c \; ; \; \left[\overline{S}, \ldots, \overline{S}, \kappa\sigma\overline{S}, \ldots, \kappa\sigma\overline{S}\right]\right]$$

Computation of the Power - General Case

$$\beta_g^r(\kappa, M) := \mathbb{P}\left[\boldsymbol{f} \in R^c\ ;\ \left[\overline{S}, \ldots, \overline{S}, \kappa\sigma\overline{S}, \ldots, \kappa\sigma\overline{S}\right]\right]$$

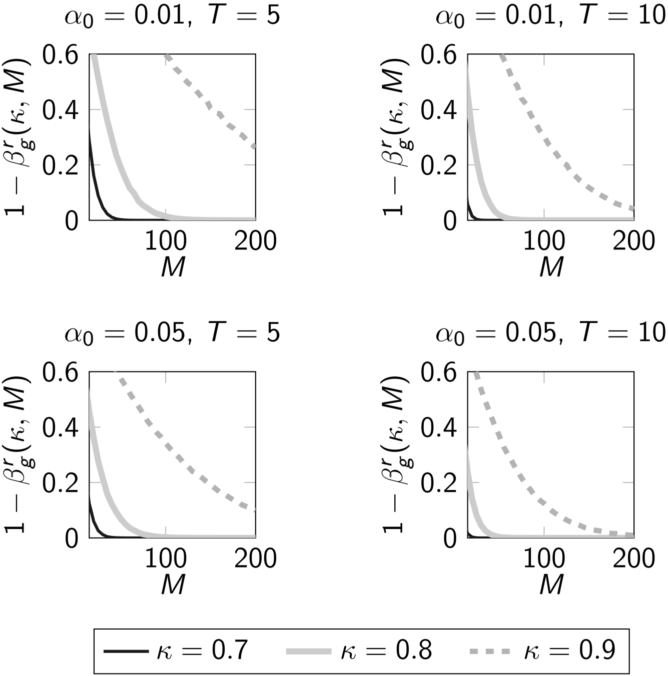
1. compute
$$q_1 = 1 - \frac{\Gamma(M, \kappa M)}{\Gamma(M)}$$
as before
2. set $p_1(a) = \mathrm{Gamma}\left(M, (\kappa M)^{-1}\right)$
3. compute $F_\omega(\lambda_T)$ as before
4. compute
$$\beta_g^r(\kappa, M) = F_\omega(\lambda_T)$$

Computation of the Power - General Case

$$\beta_g^r(\kappa, M) := \mathbb{P}\left[\boldsymbol{f} \in R^c \; ; \; \left[\overline{S}, \ldots, \overline{S}, \kappa\sigma\overline{S}, \ldots, \kappa\sigma\overline{S}\right]\right]$$

1. compute
$$q_1 = 1 - \frac{\Gamma(M, \kappa M)}{\Gamma(M)}$$
as before
2. set $p_1(a) = \mathrm{Gamma}\left(M, (\kappa M)^{-1}\right)$
3. compute $F_\omega(\lambda_T)$ as before
4. compute
$$\beta_g^r(\kappa, M) = F_\omega(\lambda_T)$$

no UMP test exists for this problem!


$\alpha_0 = 0.01,\ T = 5$
$\alpha_0 = 0.01,\ T = 10$
$\alpha_0 = 0.05,\ T = 5$
$\alpha_0 = 0.05,\ T = 10$
$1 - \beta_g^r(\kappa, M)$
0.6
0.4
0.2
0
100
200
M
$\kappa = 0.7$
$\kappa = 0.8$
$\kappa = 0.9$

Experiments

video

Conclusions

- using max-consensus (fast scheme!) is meaningful for topology change detection purposes
- main tradeoff = performance vs. communication requirements
- characterization can be used parameters selection

future direction: adapt for traffic management purposes

Distributed detection of topological changes in communication networks

Riccardo Lucchese, **Damiano Varagnolo**, Karl H. Johansson

damiano.varagnolo@ltu.se

entirely written in LaTeX2ε using Beamer and Tik Z