Fast graph discovering in anonymous networks

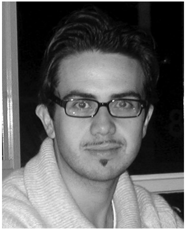
Damiano Varagnolo

KTH Royal Institute of Technology

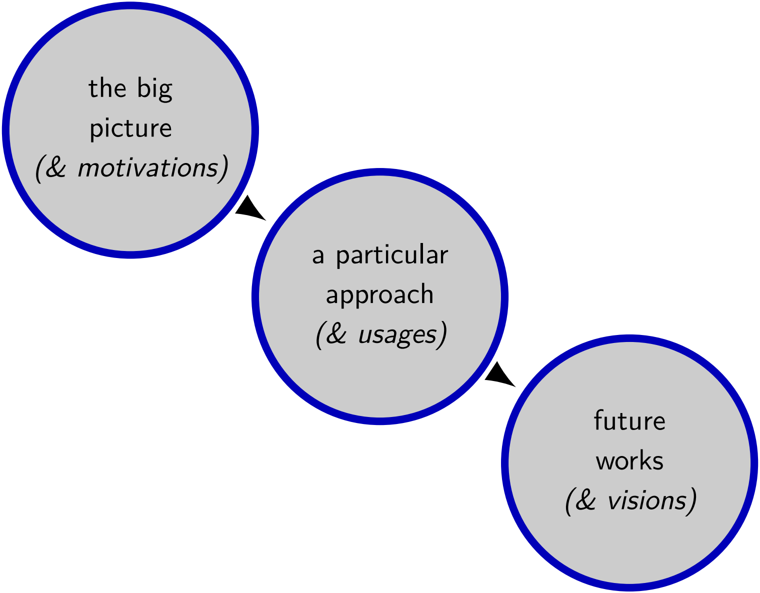
October 19, 2012

Thanks to. . .



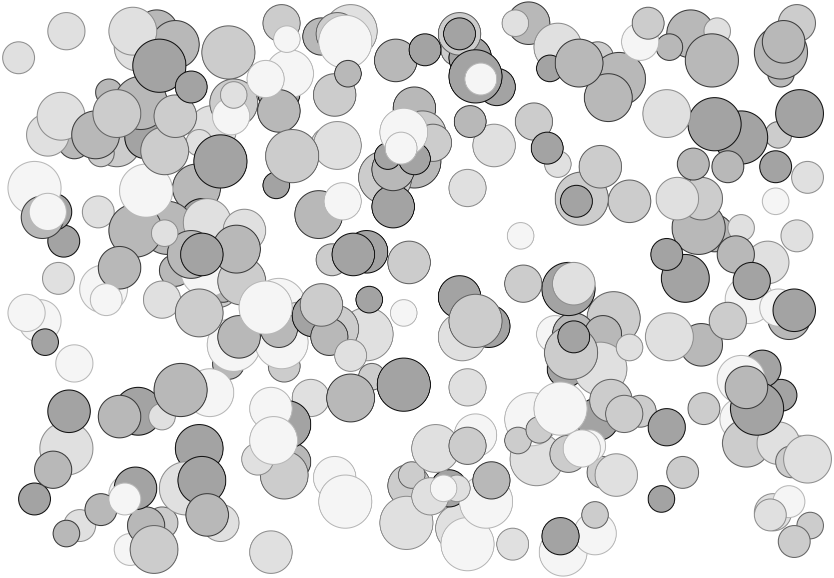

This talk


the big
picture
(& motivations)
a particular
approach
(& usages)
future
works
(& visions)

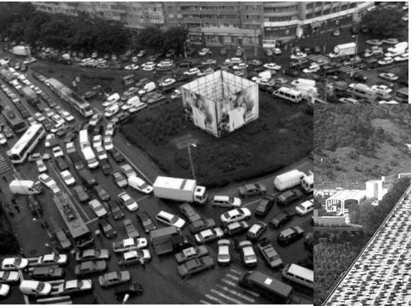
Smart objects. . .

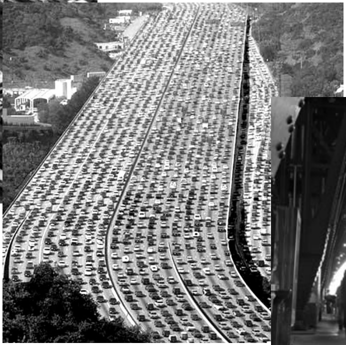
. . . for a distributed world?

(cf. the Metcalfe law)



An example: the transportation system



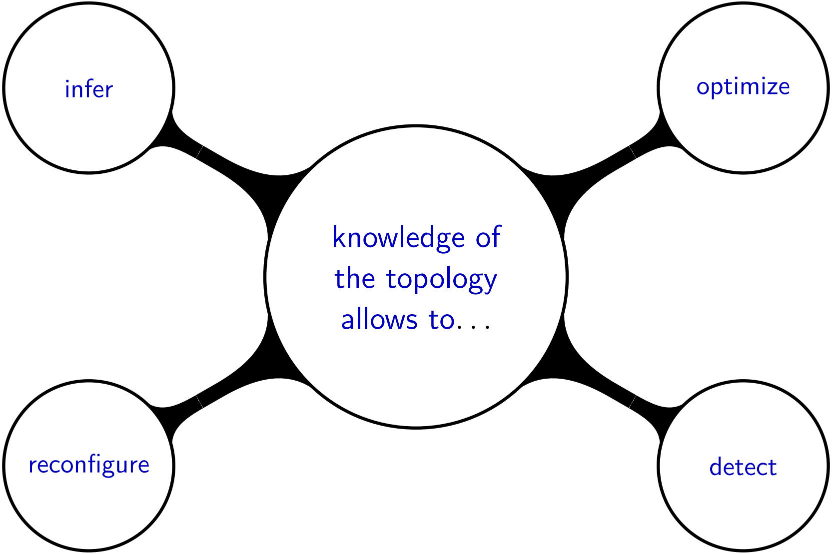


Distributed systems: requirements

- easy implementability
- sufficiently fast convergence
- contained bandwidth requirements
- robustness w.r.t. failures
- robustness w.r.t. churn
- scalability
- ...

Topology Matters

Examples of (high-level) applications



Graph discovering literature – some dichotomies

- static networks ***vs*** dynamic networks
- identity-based ***vs*** privacy-aware algorithms
- information-aggregation ***vs*** information-propagation algorithms

Graph discovering literature – some dichotomies

- static networks ***vs*** dynamic networks
- identity-based ***vs*** privacy-aware algorithms
- information-aggregation ***vs*** information-propagation algorithms

Examples:

- construction of graph views
- random walks
- capture-recapture

Big question:

what can we estimate?

Big question:

what can we estimate?

- without any constraint $\Rightarrow$ infer the whole graph perfectly

Big question:

what can we estimate?

- without any constraint $\Rightarrow$ infer the whole graph perfectly
- with anonymity constraints $\Rightarrow$ infer the whole graph w.h.p.

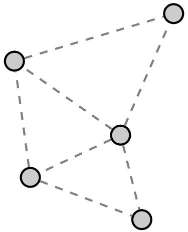
Today's niche

time & consensus & scalability constraints

(i.e., highly dynamic networks where convergence speed is crucial)

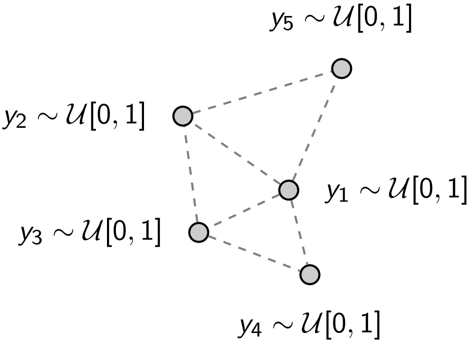
$\Rightarrow$ analyze **max-consensus**

A building block: size estimation with max-consensus



A building block: size estimation with max-consensus

i.i.d. local generation

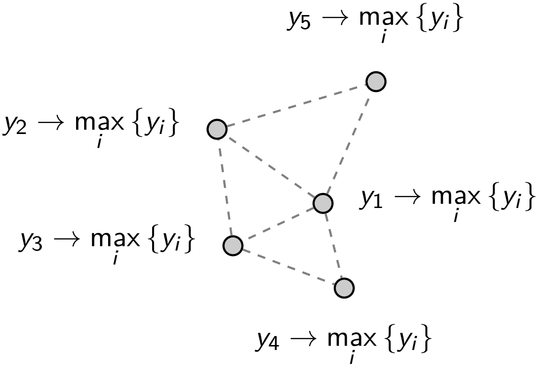


A building block: size estimation with max-consensus

i.i.d. local generation

↓

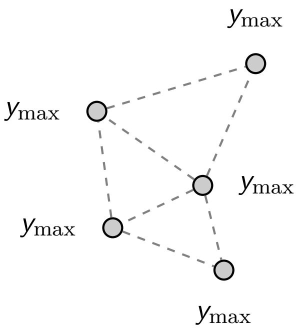
max consensus



A building block: size estimation with max-consensus

i.i.d. local generation

max consensus



A building block: size estimation with max-consensus

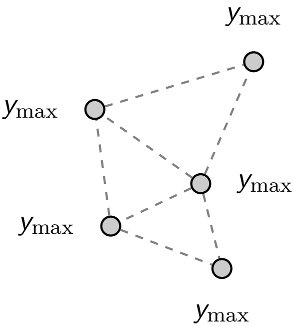
i.i.d. local generation

↓

max consensus

↓

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



A building block: size estimation with max-consensus

i.i.d. local generation

↓

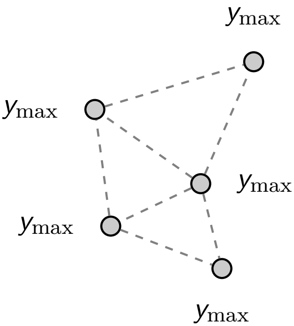
max consensus

↓

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

↓

estimate S with statistical inference



Performance characterization

(under no-quantization issues)

Generalizations:

- perform M independent trials in parallel
- $y_{i,m} \sim F(\cdot)$ (absolutely continuous distribution)

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

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

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

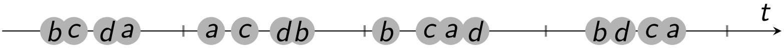
$$\Rightarrow \operatorname{var}\left(\frac{\widehat{S} - S}{S}\right) \approx \frac{1}{M}$$

$$\Rightarrow \left(\widehat{S}\right)^{-1} = \widehat{S^{-1}} \quad \textit{and} \quad \widehat{S^{-1}} \text{ is MVUE for } S^{-1}$$

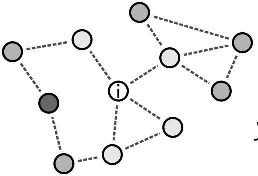
Extension 1 – neighborhoods size estimation

Assumption: synchronous communications

- time is divided in *epochs*
- everybody communicates once per epoch



⇒ induces well-defined *k-steps neighborhoods*:



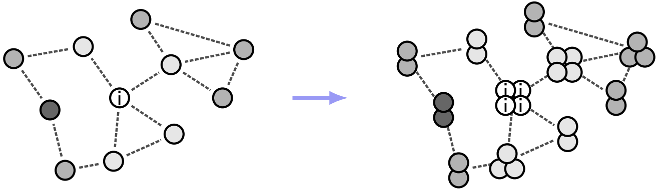
$t =$	0	1	2	3
$y_i(t)$:	0.3	0.7	0.7	0.7
	0.6	0.6	0.6	0.9
	0.1	0.8	0.8	0.8
	0.4	0.4	0.4	0.4

Extension 2 – number of links estimation

Assumption:

- every agent knows its in- and out-degrees

$\Rightarrow$ agents can *pretend* the behavior of an equivalent number of agents:

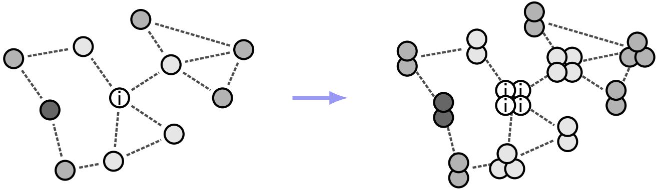


Extension 2 – number of links estimation

Assumption:

- every agent knows its in- and out-degrees

$\Rightarrow$ agents can *pretend* the behavior of an equivalent number of agents:



Remarks

- estimates *twice* the number of links
- can estimate the number of links *between* k-steps neighbors

Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \operatorname{dist}(i, j)$$

(i.e., longest shortest path starting from i)

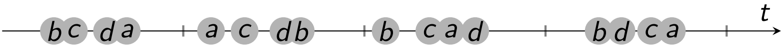
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$t = 0$

$y_i(t)$: $\begin{bmatrix} 0.3 \\ 0.6 \\ 0.1 \\ 0.4 \end{bmatrix}$

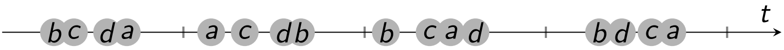
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



	$t =$ 0	1
$y_i(t)$:	0.3	0.7
	0.6	0.6
	0.1	0.8
	0.4	0.4

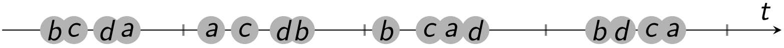
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$y_i(t)$: $t =$	0	1	2
	0.3	0.7	0.7
	0.6	0.6	0.6
	0.1	0.8	0.8
	0.4	0.4	0.4

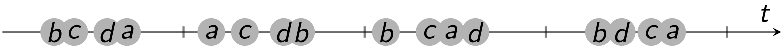
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \operatorname{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$t =$	0	1	2	3
$y_i(t)$:	0.3	0.7	0.7	0.7
	0.6	0.6	0.6	0.9
	0.1	0.8	0.8	0.8
	0.4	0.4	0.4	0.4

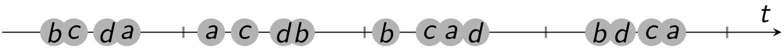
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



	$t =$ 0	1	2	3	4
$y_i(t)$:	0.3	0.7	0.7	0.7	0.7
	0.6	0.6	0.6	0.9	0.9
	0.1	0.8	0.8	0.8	0.8
	0.4	0.4	0.4	0.4	0.4

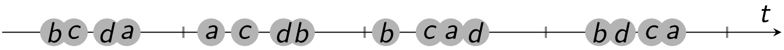
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



	$t =$ 0	1	2	3	4	5
$y_i(t)$:	0.3	0.7	0.7	0.7	0.7	0.7
	0.6	0.6	0.6	0.9	0.9	0.9
	0.1	0.8	0.8	0.8	0.8	0.8
	0.4	0.4	0.4	0.4	0.4	0.4

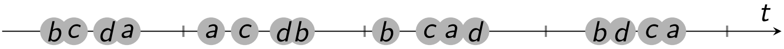
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \operatorname{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$y_i(t)$:

$t =$	0	1	2	3	4	5	6
	0.3	0.7	0.7	0.7	0.7	0.7	0.7
	0.6	0.6	0.6	0.9	0.9	0.9	0.9
	0.1	0.8	0.8	0.8	0.8	0.8	0.8
	0.4	0.4	0.4	0.4	0.4	0.4	0.4

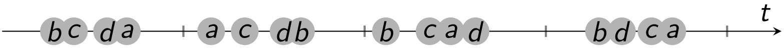
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \operatorname{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$t =$	0	1	2	3	4	5	6
$y_i(t)$:	0.3	0.7	0.7	0.7	0.7	0.7	0.7
	0.6	0.6	0.6	0.9	0.9	0.9	0.9
	0.1	0.8	0.8	0.8	0.8	0.8	0.8
	0.4	0.4	0.4	0.4	0.4	0.4	0.4

$\Rightarrow \widehat{e}(i) = 3$

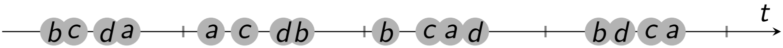
Extension 3 – estimation of eccentricities

Definition

$$e(i) := \max_{j \in \mathcal{V}} \mathrm{dist}(i, j)$$

(i.e., longest shortest path starting from i)

Assumption: synchronous communications



$t =$	0	1	2	3	4	5	6
$y_i(t)$:	0.3	0.7	0.7	0.7	0.7	0.7	0.7
	0.6	0.6	0.6	0.9	0.9	0.9	0.9
	0.1	0.8	0.8	0.8	0.8	0.8	0.8
	0.4	0.4	0.4	0.4	0.4	0.4	0.4

$\Rightarrow \widehat{e}(i) = 3$

remark: statistical properties may depend on the actual graph

Extension 4 – estimation of radii and diameters

Definitions

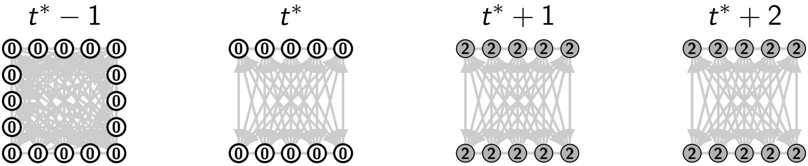
$$r := \min_{i \in \mathcal{V}} e(i) \qquad\qquad d := \max_{i \in \mathcal{V}} e(i)$$

$\Rightarrow$ *under synchronous communications assumptions* one can distributedly estimate r, d through

$$\widehat{r} = \min_{i \in \mathcal{V}} \widehat{e}(i) \qquad\qquad \widehat{d} = \max_{i \in \mathcal{V}} \widehat{e}(i)$$

Application 1 - change detection

Implemented `INRIA SensLab Strasbourg`

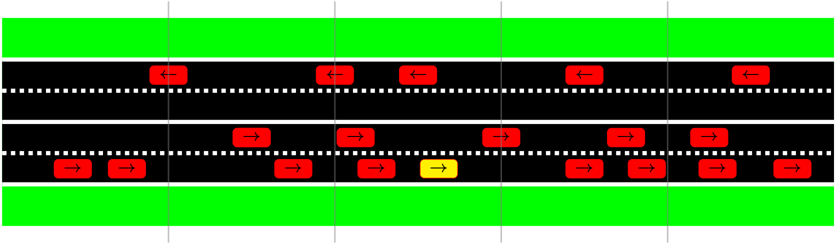


GLR approach

$$\begin{cases} \mathcal{H}_0 : & S(i) \geq \overline{S} \text{ for all } i \in \{t - T, \ldots, t\} \\ \mathcal{H}_1 : & \text{exists } i \in \{t - T, \ldots, t\} \text{ s.t. } S(i) < \overline{S} \end{cases}$$

$\Rightarrow$ accept $\mathcal{H}_0$ or $\mathcal{H}_1$ depending on the relative likelihood

Application 2 - transportation systems



Idea: estimate

- how many cars per meter per road's segment & lane
- traffic behavior per segment

  (average speed / acceleration, variances, ...)

Uses:

- speed control
- early warning
- ...

Big question:

what can we estimate using max consensus?

Big question:

what can we estimate using max consensus?

statistical strategies require
statistical identifiability characterizations

Statistical identifiability

Notation

- local values: $x_1 \sim p_{x_1}(\cdot),\ \ldots,\ x_S \sim p_{x_S}(\cdot)$
- graph-dependent map: $y = f_{\mathcal{G}}(x_1, \ldots, x_S) = f_{\mathcal{G}}(\boldsymbol{x})$
- parameter of interest: θ

Definition

θ is said statistically identifiable if

$$\mathcal{G}_1, \mathcal{G}_2 \text{ s.t. } \theta_1 \neq \theta_2 \quad \Rightarrow \quad \mathbb{P}\left[\boldsymbol{x} \in f_{\mathcal{G}_1}^{-1}(y)\right] \neq \mathbb{P}\left[\boldsymbol{x} \in f_{\mathcal{G}_2}^{-1}(y)\right]$$

for some y

A first characterization

Theorem

Hypotheses:

- $p_{x_i} = p_x$ (i.e., agents are equal)
- θ statistically not identifiable from p_x
- $f(\boldsymbol{x})$ anonymously computable and independent on $\mathcal{G}$
- $f(\boldsymbol{x})$ is a *consensus*

Thesis:

- unique statistically identifiable θ is the network size

A first characterization

Theorem

Hypotheses:

- $p_{x_i} = p_x$ (i.e., agents are equal)
- θ statistically not identifiable from p_x
- $f(\boldsymbol{x})$ anonymously computable and independent on $\mathcal{G}$
- $f(\boldsymbol{x})$ is a *consensus*

Thesis:

- unique statistically identifiable θ is the network size

Implication:

- under theorem's assumptions and "at most d communications" one can infer just the network size

Induced (and unanswered) questions

Assumptions *(bounded memory / network size)*

- $\dfrac{\text{memory}}{\text{network size}}$ enough to have time counters
- $\dfrac{\text{memory}}{\text{network size}}$ **not** enough to share graph views

Induced (and unanswered) questions

Assumptions *(bounded memory / network size)*

- $\dfrac{\text{memory}}{\text{network size}}$ enough to have time counters
- $\dfrac{\text{memory}}{\text{network size}}$ **not** enough to share graph views

- what can be computed with "max-consensus + time-counters"?

Induced (and unanswered) questions

Assumptions *(bounded memory / network size)*

- $\dfrac{\text{memory}}{\text{network size}}$ enough to have time counters
- $\dfrac{\text{memory}}{\text{network size}}$ **not** enough to share graph views

- what can be computed with "max-consensus + time-counters"?
- can we prove that "max-consensus + time-counters" are the fastest?

Induced (and unanswered) questions

Assumptions *(bounded memory / network size)*

- $\dfrac{\text{memory}}{\text{network size}}$ enough to have time counters
- $\dfrac{\text{memory}}{\text{network size}}$ **not** enough to share graph views

- what can be computed with "max-consensus + time-counters"?
- can we prove that "max-consensus + time-counters" are the fastest?
- are they also "unique"?

An even more basic question

Why should we do max-consensus on reals?

I.e., is it better to use discrete or "continuous" r.v.s?

Two simple and open problems

Assumptions:

- memory $=$ 50 bits (example)
- $\exists$ upper bound on the number of agents
- metric: statistical estimation performance

Two simple and open problems

Assumptions:

- memory $=$ 50 bits (example)
- $\exists$ upper bound on the number of agents
- metric: statistical estimation performance

Scheme A: do as before *(quantize real values)*

- divide the 50 bits in M scalars *(how?)*
- quantize opportunely *(how?)*

Two simple and open problems

Assumptions:

- memory $=$ 50 bits (example)
- $\exists$ upper bound on the number of agents
- metric: statistical estimation performance

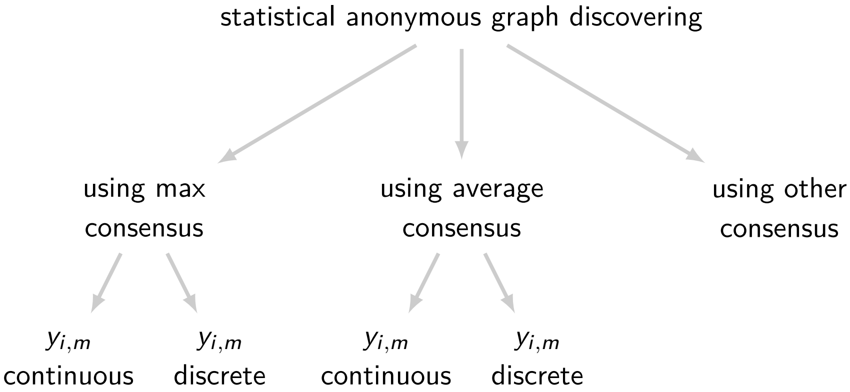
Scheme A: do as before *(quantize real values)*

- divide the 50 bits in M scalars *(how?)*
- quantize opportunely *(how?)*

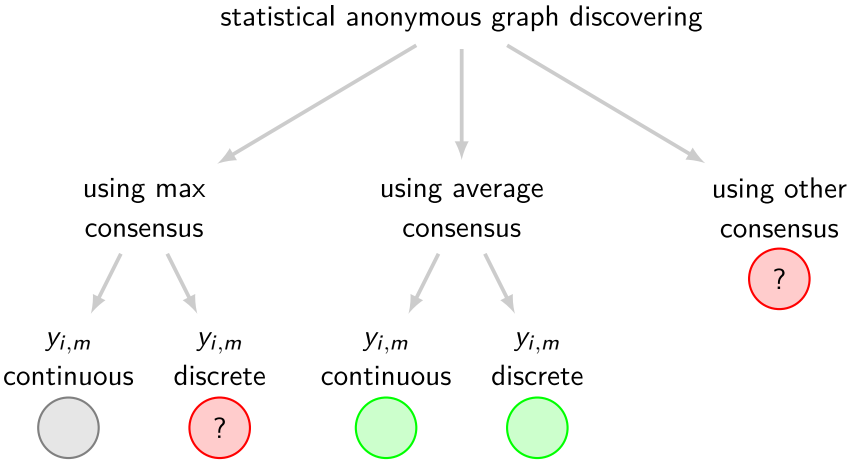
Scheme B: each bit $=$ Bernoulli

- compute the ML estimator *(how?)*
- set the success probability optimally *(how?)*

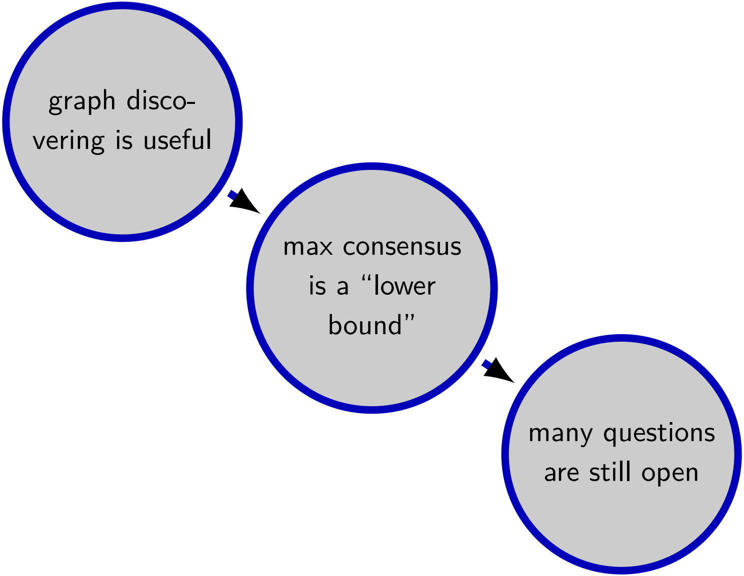
Summary: what do we know



Summary: what do we know



Summarizing . . .



Bibliography

Varagnolo, Pillonetto, Schenato (20??)
Distributed size estimation in anonymous networks
IEEE Transactions on Automatic Control

Garin, Varagnolo, Johansson (2012)
Distributed estimation of diameter, radius and eccentricities in anonymous networks
NecSys

Fast graph discovering in anonymous networks

Damiano Varagnolo

KTH Royal Institute of Technology

October 19, 2012

damiano@kth.se

entirely written in LaTeX 2ε using Beamer and Tik Z