Distributed size estimation in anonymous networks

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

Department of Information Engineering, University of Padova

February 9th, 2012

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Table of Contents

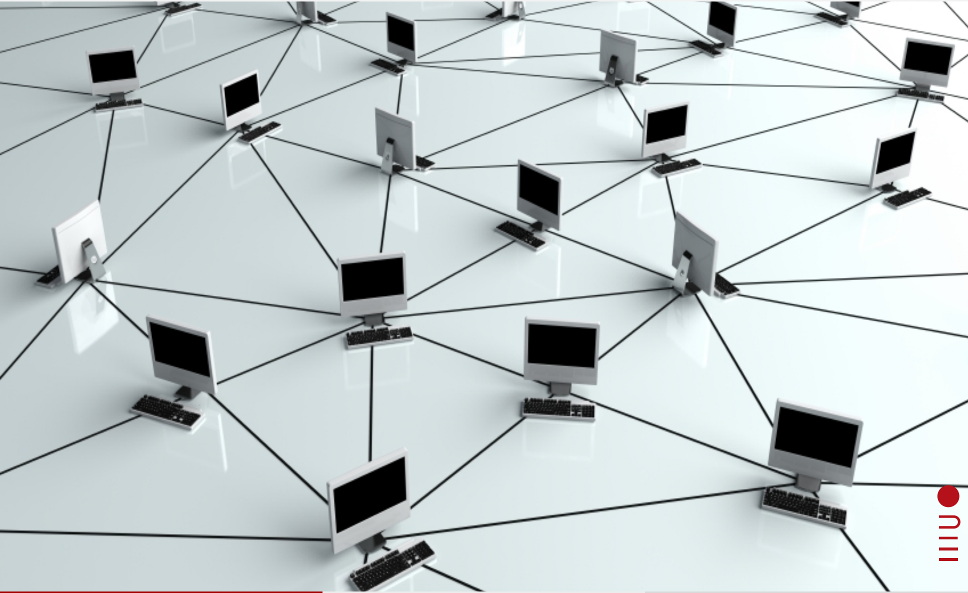
Table of Contents

Focus of this talk:

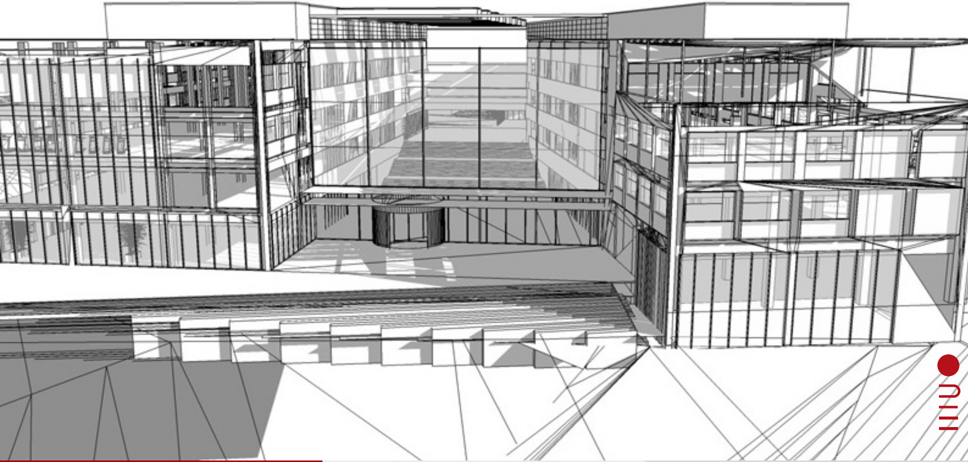
distributed estimation
of the size S of a network

$\rightarrow$ *i.e. let the agents know how many they are*

Motivations (1/3): network maintenance purposes

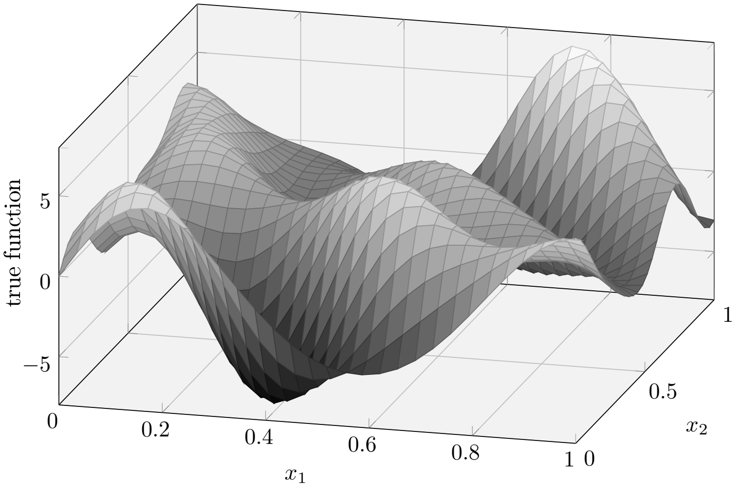


Motivations (2/3): smart buildings management



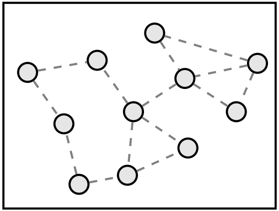
Motivations (3/3): estimation purposes

(also S^{-1} may be interesting!!)



Problem definition

hypotheses

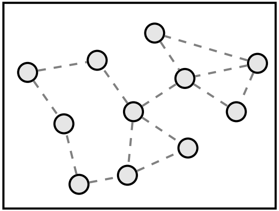


- $S :=$ network size
- S deterministic and constant in time
- agents have *limited computational / memory / communication capabilities*
- network is *anonymous*

  (no IDs or IDs not assured to be unique)

Problem definition

hypotheses



- $S :=$ network size
- S deterministic and constant in time
- agents have *limited computational / memory / communication capabilities*
- network is *anonymous*

  (no IDs or IDs not assured to be unique)

Goal: develop a distributed estimator $\widehat{S}$ of S satisfying the constraints

Literature review

network size estimation = not a new problem!!

Literature review

network size estimation = not a new problem!!

Deterministic scenario: theoretical limit for anonymous networks

$\nexists$ algorithm (with bounded average bit complexity) guaranteed to return the correct answer for every (finite) execution

Cidon, Shavitt (1995), Information Processing Letters

Literature review

network size estimation = not a new problem!!

Deterministic scenario: theoretical limit for anonymous networks

$\nexists$ algorithm (with bounded average bit complexity) guaranteed to return the correct answer for every (finite) execution

Cidon, Shavitt (1995), Information Processing Letters

Stochastic scenario: some existing approaches

Literature review

network size estimation = not a new problem!!

Deterministic scenario: theoretical limit for anonymous networks

$\nexists$ algorithm (with bounded average bit complexity) guaranteed to return the correct answer for every (finite) execution

Cidon, Shavitt (1995), Information Processing Letters

Stochastic scenario: some existing approaches

- random walk strategies

Literature review

network size estimation = not a new problem!!

Deterministic scenario: theoretical limit for anonymous networks

$\nexists$ algorithm (with bounded average bit complexity) guaranteed to return the correct answer for every (finite) execution

Cidon, Shavitt (1995), Information Processing Letters

Stochastic scenario: some existing approaches

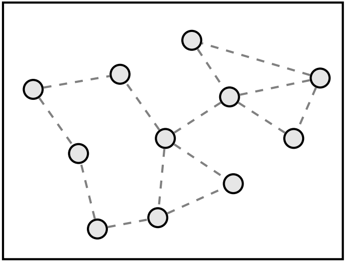
- random walk strategies
- capture-recapture strategies

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



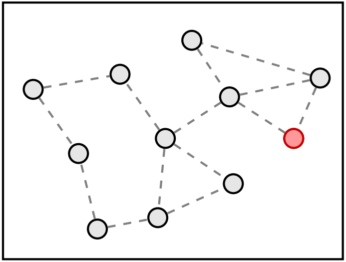
Algorithm

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

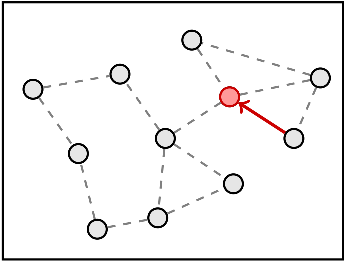
1. generate a "seed"

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

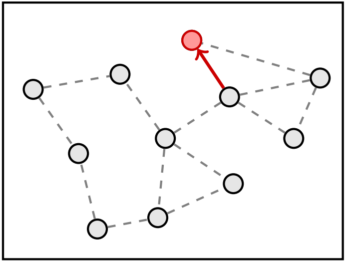
1. generate a "seed"
2. randomly propagate it

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

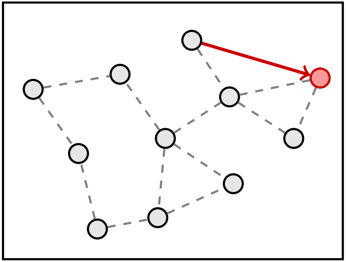
1. generate a "seed"
2. randomly propagate it

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

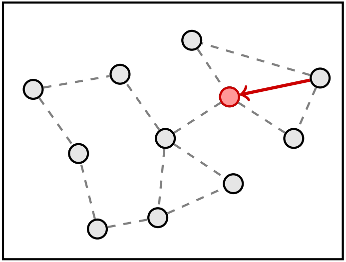
1. generate a "seed"
2. randomly propagate it

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

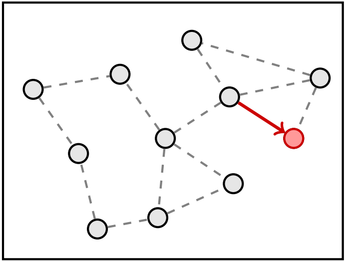
1. generate a “seed”
2. randomly propagate it

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing



Algorithm

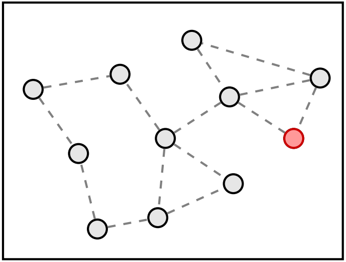
1. generate a "seed"
2. randomly propagate it

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)

Peer counting and sampling in overlay networks: random walk methods

ACM symposium on Principles of distributed computing

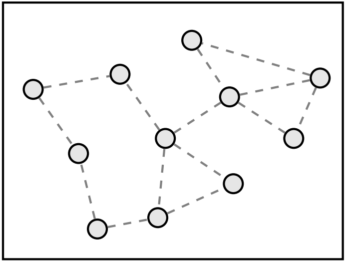


Algorithm

1. generate a "seed"
2. randomly propagate it
3. # of jumps → statistically dependent on S

Random walks

Massoulié, Le Merrer, Kermarrec, Ganesh (2006)
Peer counting and sampling in overlay networks: random walk methods
ACM symposium on Principles of distributed computing



Algorithm

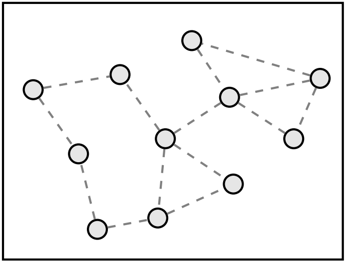
1. generate a "seed"
2. randomly propagate it
3. # of jumps $\longrightarrow$ statistically dependent on S
4. variance of the error: $\propto$ (# of generated seeds)$^{-1}$

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



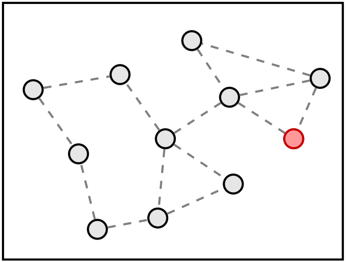
Algorithm

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

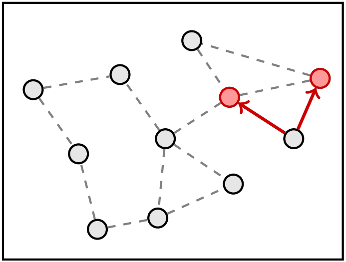
1. generate N seeds

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

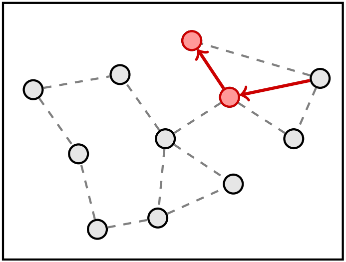
1. generate N seeds
2. propagate them

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

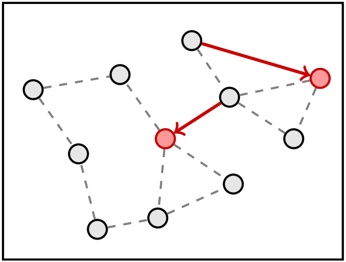
1. generate N seeds
2. propagate them

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

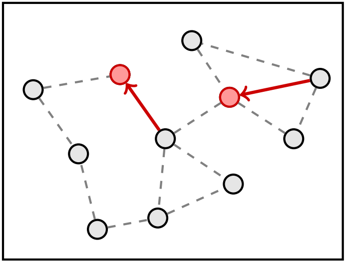
1. generate N seeds
2. propagate them

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

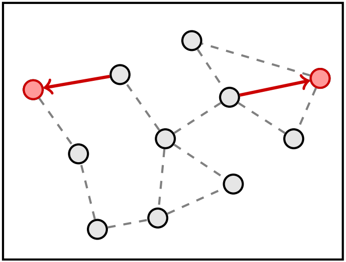
1. generate N seeds
2. propagate them

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

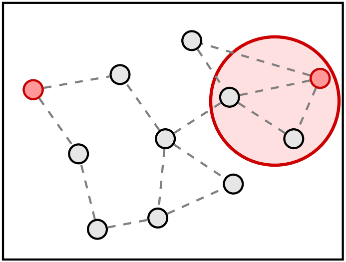
1. generate N seeds
2. propagate them

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

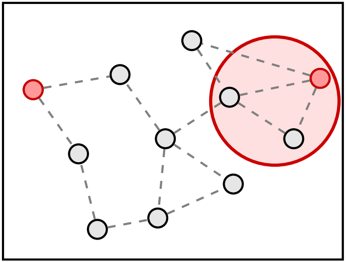
1. generate N seeds
2. propagate them
3. capture and infer

Capture-recapture

Seber (1982)

The estimation of animal abundance and related parameters

London: Charles Griffin & Co.



Algorithm

1. generate N seeds
2. propagate them
3. capture and infer
4. variance of the error:

   $\propto$ # of captured seeds

   (polynomially)

Our algorithm

several peculiarities
w.r.t. existing literature

Our algorithm

several peculiarities
w.r.t. existing literature

- full parallelism $\longrightarrow$ *every agent will have an estimate at the same time*

Our algorithm

several peculiarities
w.r.t. existing literature

- full parallelism $\longrightarrow$ *every agent will have an estimate at the same time*
- easily implementable in anonymous networks

Our algorithm

several peculiarities
w.r.t. existing literature

- full parallelism $\longrightarrow$ *every agent will have an estimate at the same time*
- easily implementable in anonymous networks
- nice mathematical properties

Our algorithm

several peculiarities
w.r.t. existing literature

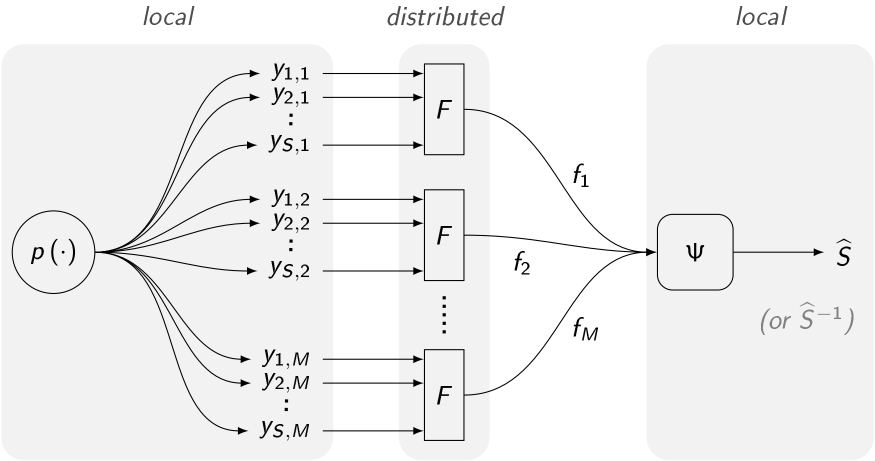
- full parallelism → *every agent will have an estimate at the same time*
- easily implementable in anonymous networks
- nice mathematical properties

the idea: generate random numbers → combine them with consensus → exploit statistical inference

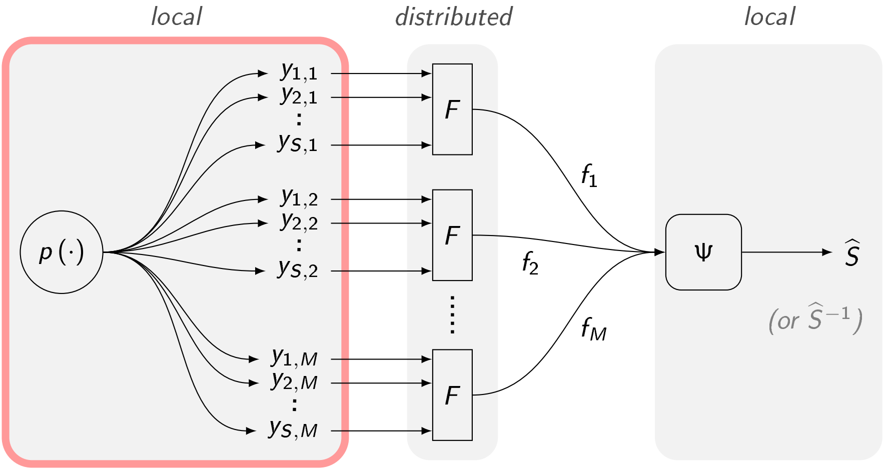
Cohen (1997), Journal of Computer and System Sciences,
Size-estimation framework with applications to transitive closure and reachability

Table of Contents

Block representation of our strategy

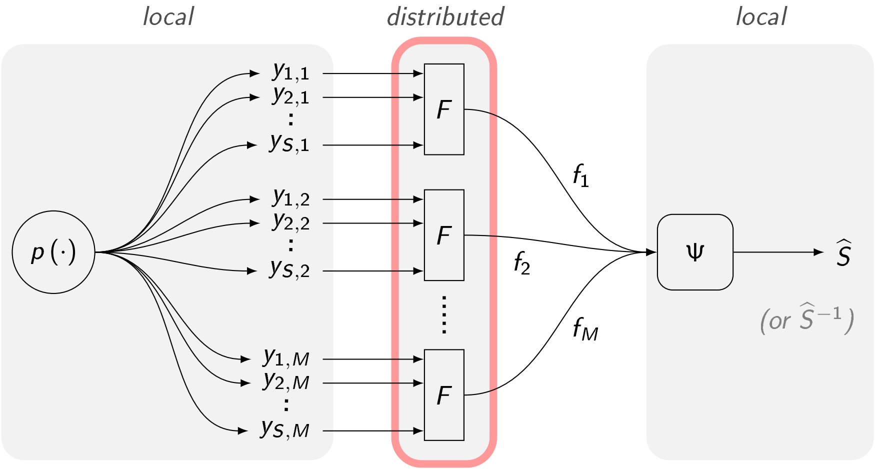


Block representation of our strategy



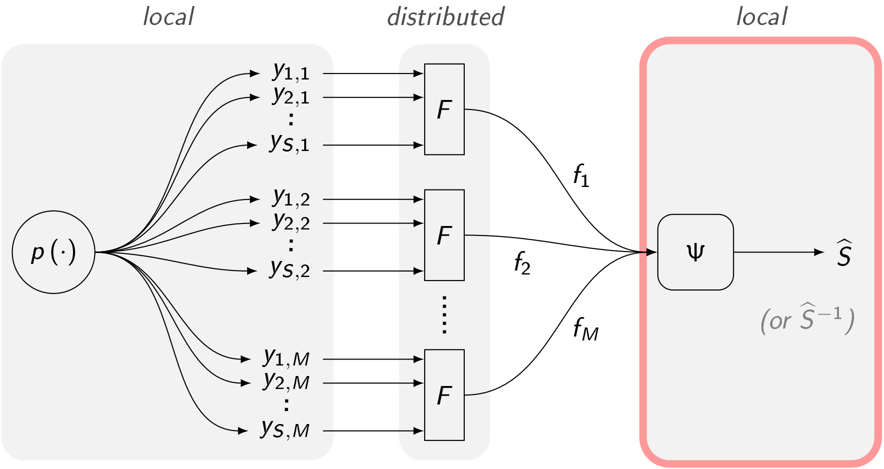
every agent i generates a M-tuple $\{y_{i,1}, \ldots, y_{i,M}\}$, $\quad y_{i,m} \sim p(\cdot)$

Block representation of our strategy



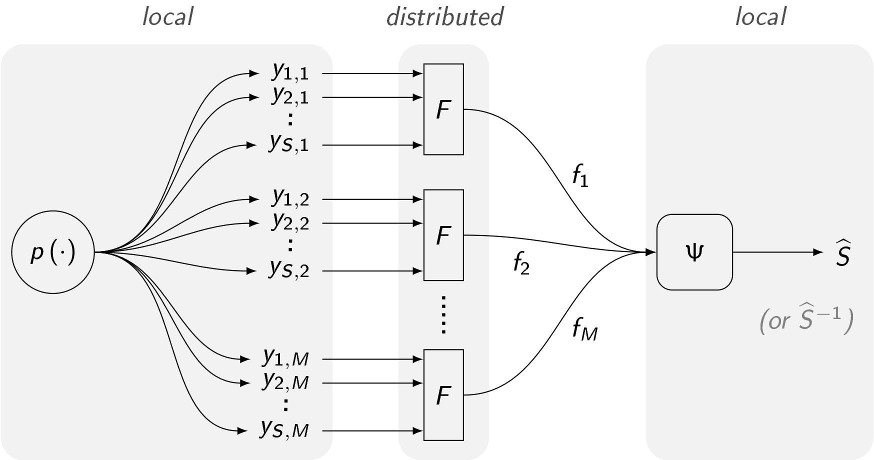
the S-tuples $\{y_{1,m}, \ldots, y_{S,m}\}$ are converted into a scalar f_m through F
(e.g. F = average, F = max)

Block representation of our strategy



the M-tuple $\{f_1, \ldots, f_M\}$ is converted into an estimate $\widehat{S}$ through Ψ
(e.g. Ψ = Maximum Likelihood)

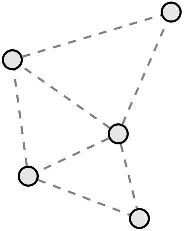
Block representation of our strategy



cost function: $J(p, F, \Psi) := \mathbb{E}\left[\left(S - \widehat{S}\right)^2\right]$

An example

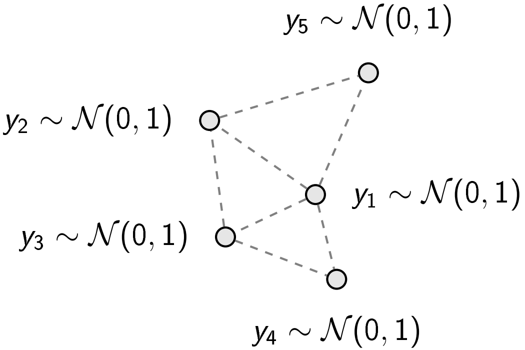
Algorithm ($M = 1$):



An example

Algorithm ($M = 1$):

local generation with $p = \mathcal{N}(0, 1)$



An example

Algorithm ($M = 1$):

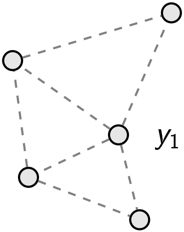
local generation with $p = \mathcal{N}(0,1)$

↓

$F =$ average consensus

$$y_5 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_2 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$



$$y_1 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_3 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_4 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

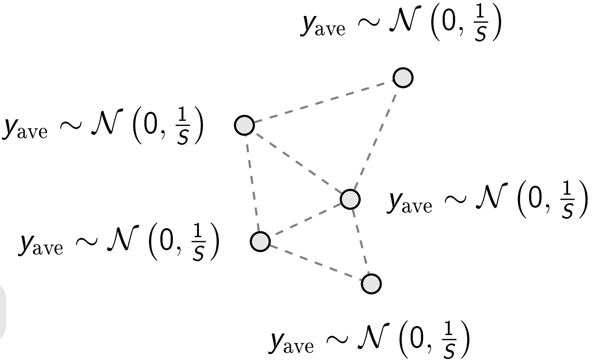
An example

Algorithm ($M = 1$):

local generation with $p = \mathcal{N}(0, 1)$

↓

$F =$ average consensus



An example

Algorithm ($M = 1$):

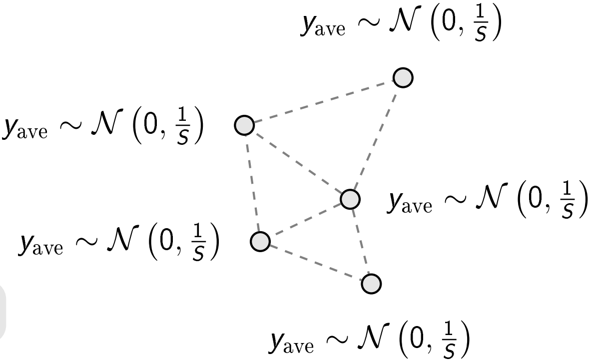
local generation with $p = \mathcal{N}(0, 1)$

↓

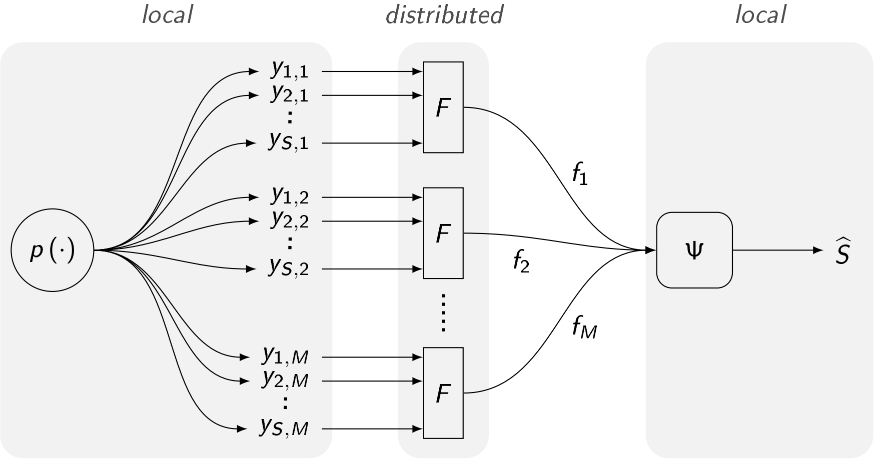
F = average consensus

↓

Ψ = Maximum Likelihood → $\widehat{S} = y_{\text{ave}}^{-2}$



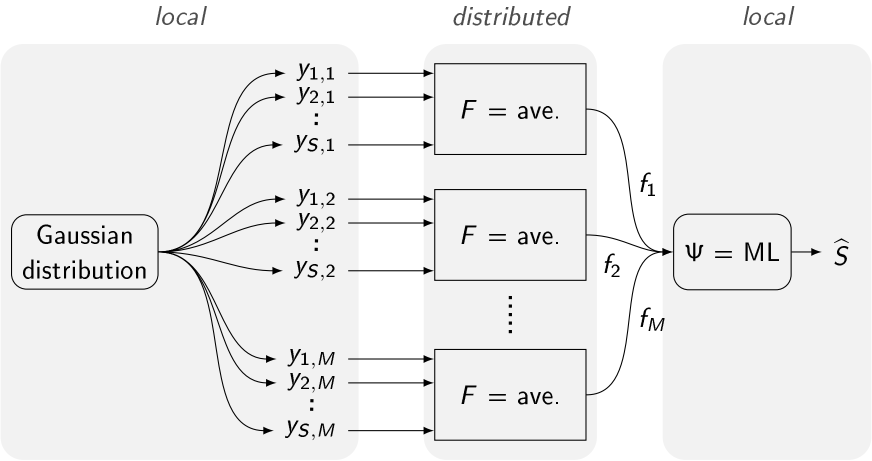
A formidable infinite-dimensional problem



$$\arg\min_{p,F,\Psi} J(p, F, \Psi) = ?? \qquad J(p, F, \Psi) := \mathbb{E}\left[\left(S - \widehat{S}\right)^2\right]$$

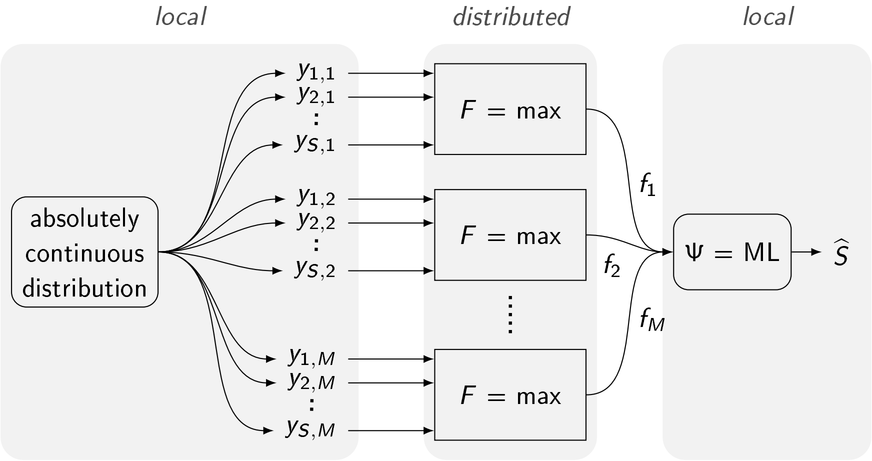
Our case studies

Case 1:



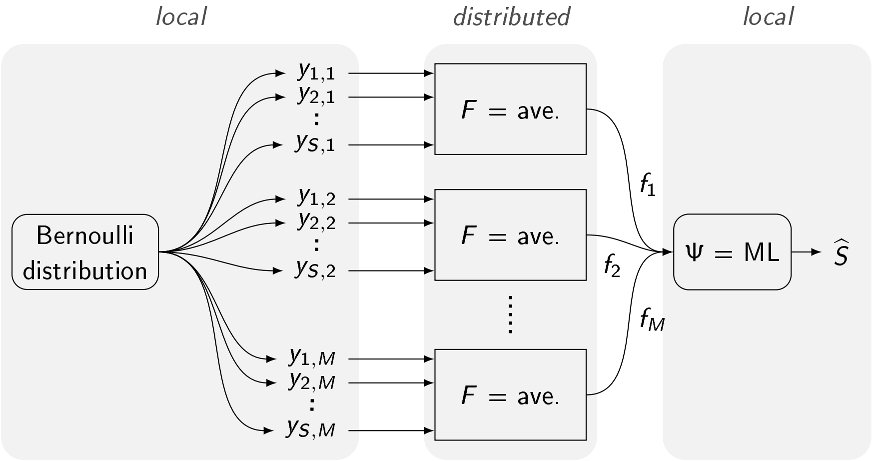
Our case studies

Case 2:



Our case studies

Case 3:



An historical case study

The German Tank problem

infer tanks production from serial numbers analysis

(June 1940 → September 1942)

intelligence	statisticians	actual
1400	256	

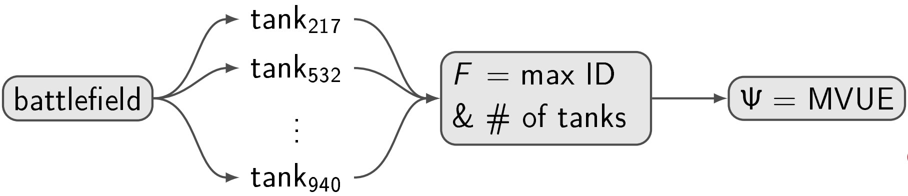
An historical case study

The German Tank problem

infer tanks production from serial numbers analysis

(June 1940 $\rightarrow$ September 1942)

intelligence	statisticians	actual
1400	256	



An historical case study

The German Tank problem

infer tanks production from serial numbers analysis

(June 1940 → September 1942)

intelligence	statisticians	actual
1400	256	**255**

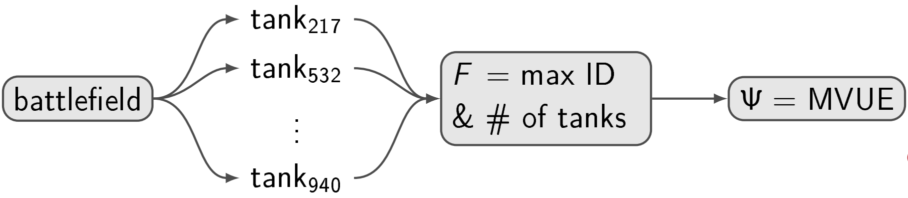
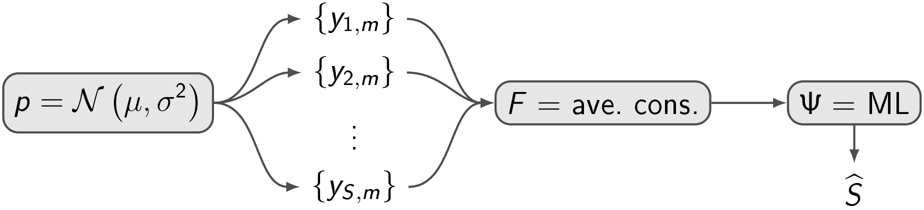
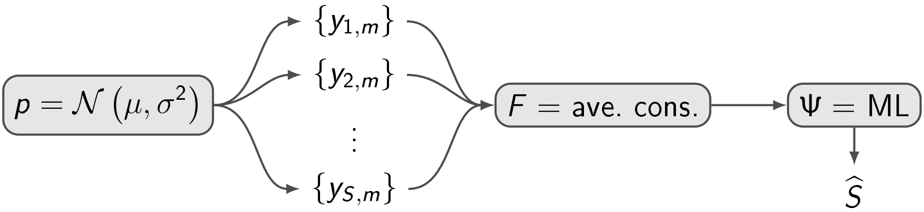


Table of Contents

Case 1: (p Gaussian) + (F = average) + (Ψ = ML)



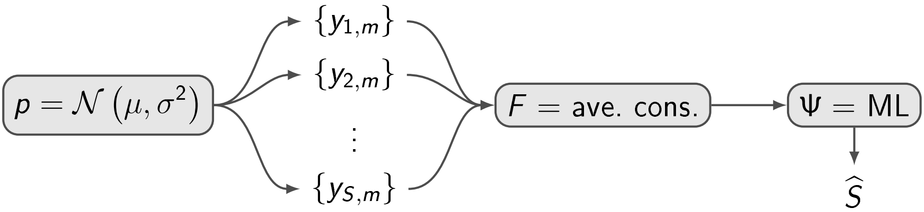
Case 1: (p Gaussian) + (F = average) + (Ψ = ML)



Results: (1 / 2) (independent of μ and σ^2)

- $\widehat{S} = \left(\frac{1}{M} \sum_{m=1}^{M} y_{\mathrm{ave},m}^2 \right)^{-1}$ $\qquad (MS)^{-1}\widehat{S} \sim \mathrm{Inv} - \chi^2(M)$
- $\mathbb{E}\left[\frac{\widehat{S}}{S} \right] = \frac{M}{M-2}$ $\qquad \mathrm{var}\left(\frac{\widehat{S} - S}{S} \right) \approx \frac{2}{M}$

Case 1: (p Gaussian) + (F = average) + (Ψ = ML)

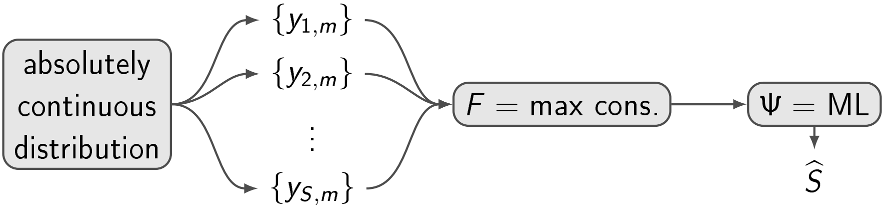


Results: (2 / 2)

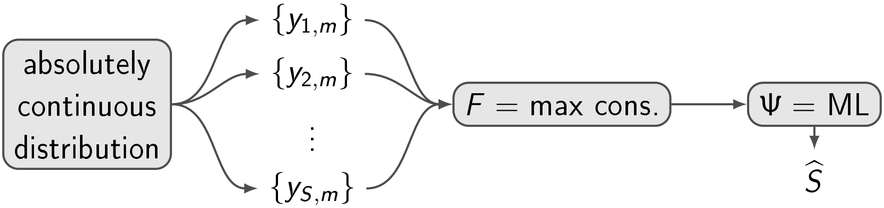
- $\left(\widehat{S}\right)^{-1} = \widehat{S^{-1}}$ *and* $\widehat{S^{-1}}$ is MVUE for S^{-1}
- for generic regular $p(\cdot)$, $S \uparrow \ \Rightarrow \ \frac{1}{S} \sum y_i \xrightarrow{\text{dist.}} \mathcal{N}\left(0, \frac{1}{S}\right)$

  implication: performances tend to become independent of $p(\cdot)$

Case 2: (p continuous) + (F = max) + (Ψ = ML)



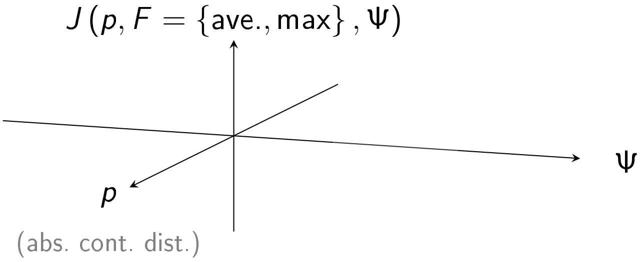
Case 2: (p continuous) + (F = max) + (Ψ = ML)



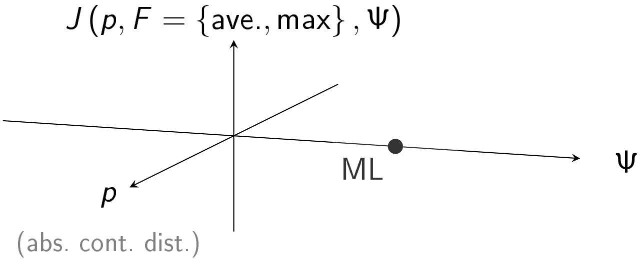
Results: ***independent of*** $p(\cdot)$

- $\widehat{S} = \left(\frac{1}{M} \sum_{m=1}^{M} - \log \left(\mathbb{P} \left[y_{\mathrm{ave},m} \right] \right) \right)^{-1}$ $\quad (MS)^{-1} \widehat{S} \sim \mathrm{Inv} - \Gamma(M, 1)$
- $\mathbb{E}\left[\frac{\widehat{S}}{S} \right] = \frac{M}{M-1} \quad \mathrm{var}\left(\frac{\widehat{S} - S}{S} \right) \approx \frac{1}{M}$ ***($\times \frac{1}{2}$ w.r.t. average)***
- $\left(\widehat{S} \right)^{-1} = \widehat{S^{-1}}$ *and* $\widehat{S^{-1}}$ is MVUE for S^{-1}

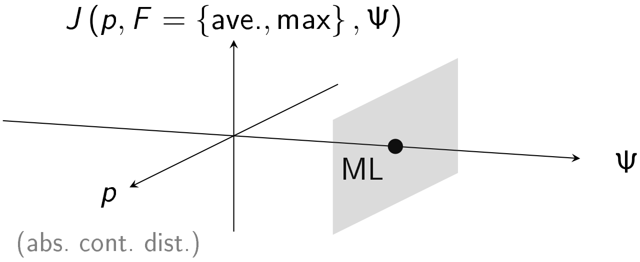
A graphical summary



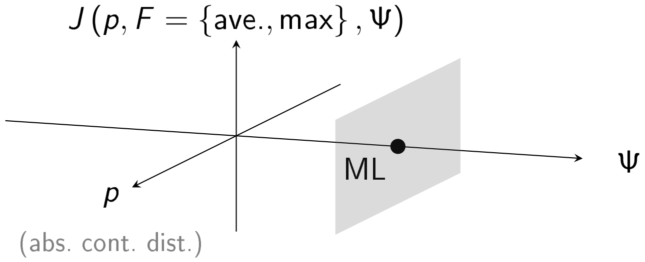
A graphical summary

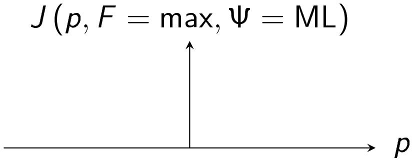


A graphical summary

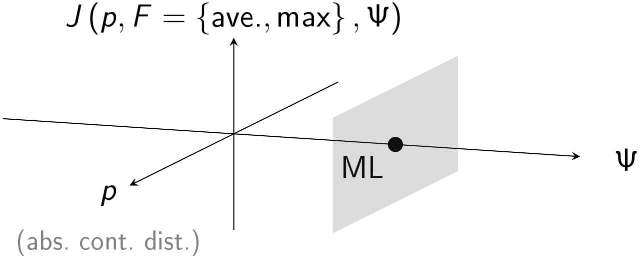


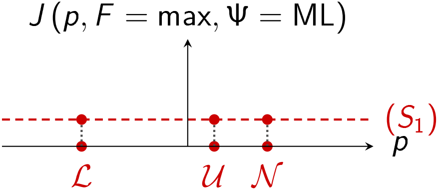
A graphical summary





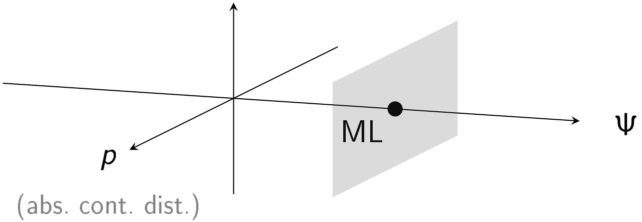
A graphical summary



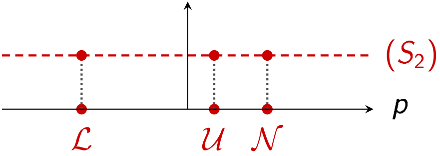


A graphical summary

$$J\left(p, F = \{\text{ave.}, \max\}, \Psi\right)$$

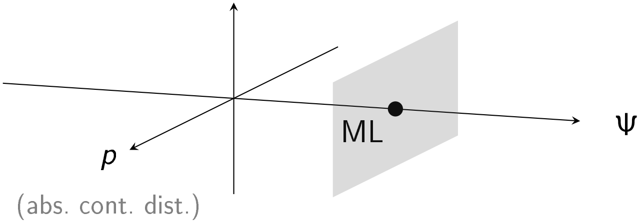


$$J\left(p, F = \max, \Psi = \text{ML}\right)$$

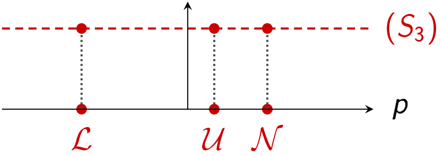


A graphical summary

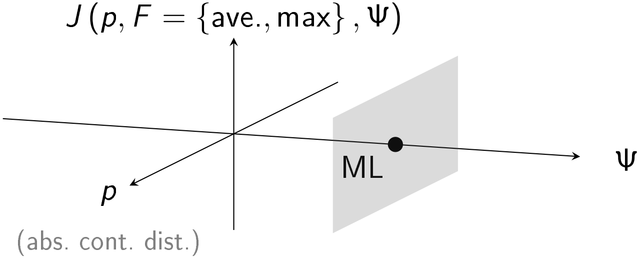
$J\left(p, F = \{\text{ave.}, \max\}, \Psi\right)$

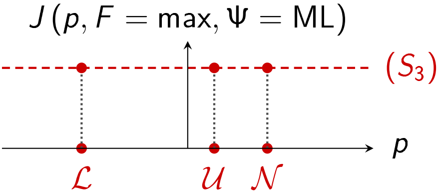


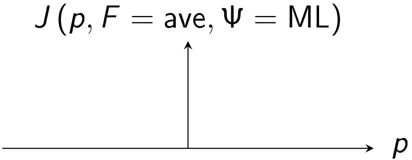
$J\left(p, F = \max, \Psi = \text{ML}\right)$



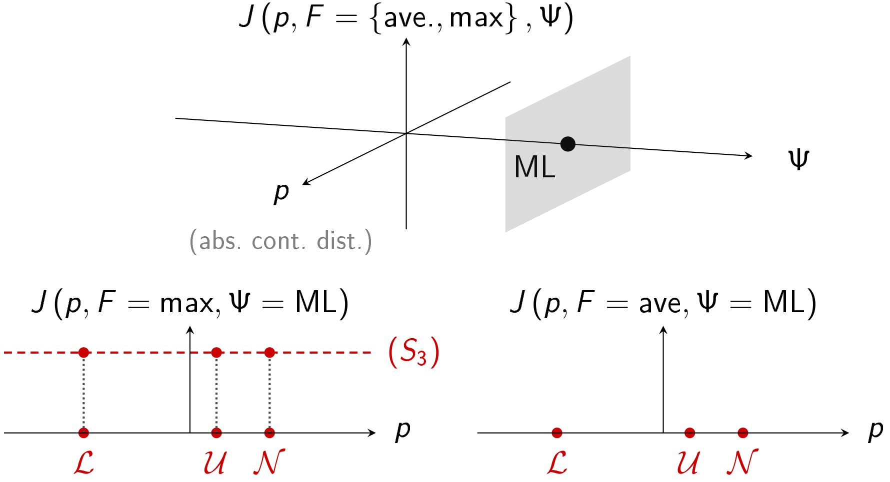
A graphical summary





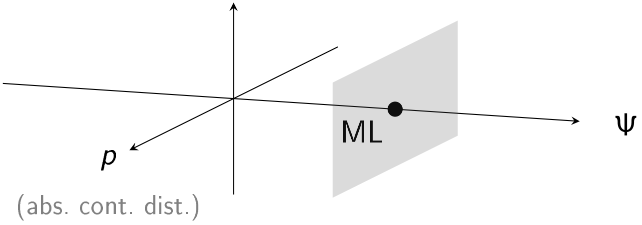


A graphical summary

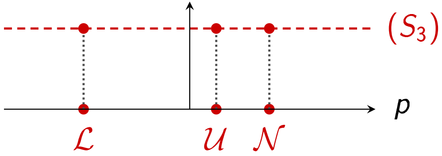


A graphical summary

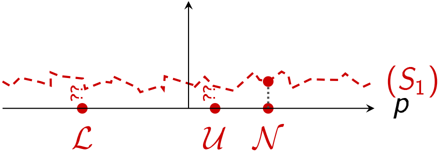
$$J\left(p, F = \left\{\text{ave.}, \max\right\}, \Psi\right)$$



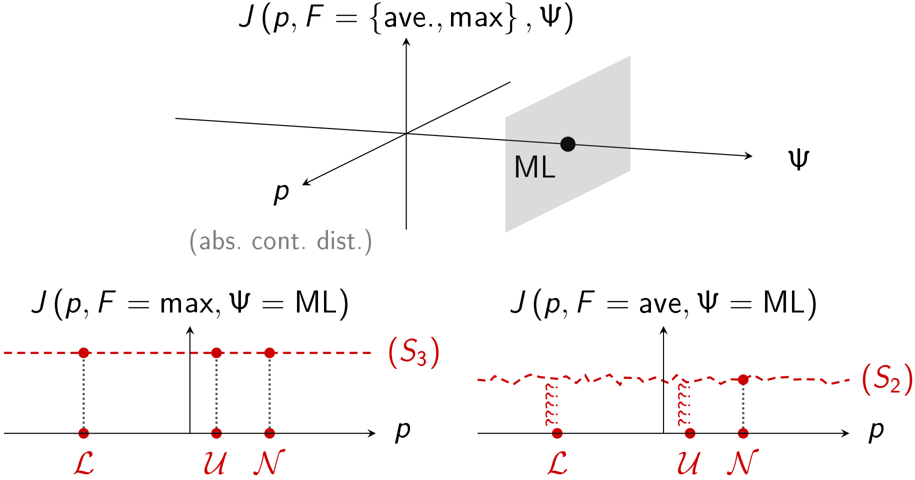
$$J\left(p, F = \max, \Psi = \text{ML}\right)$$



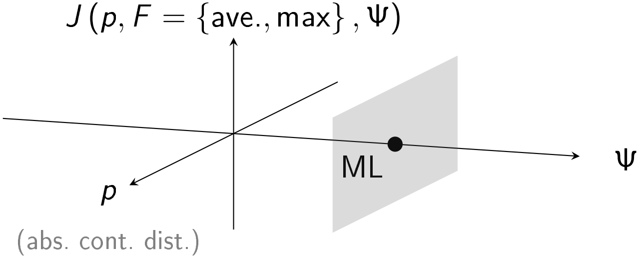
$$J\left(p, F = \text{ave}, \Psi = \text{ML}\right)$$

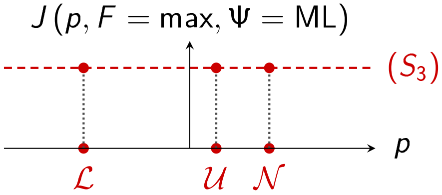


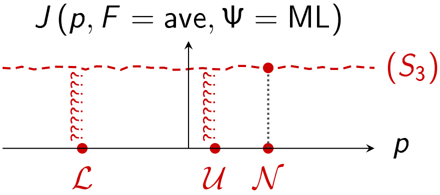
A graphical summary



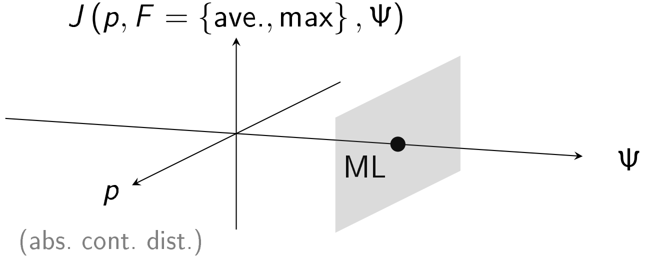
A graphical summary

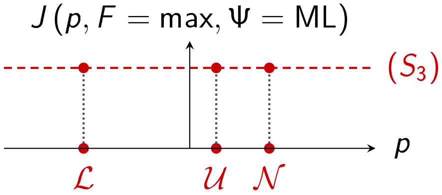


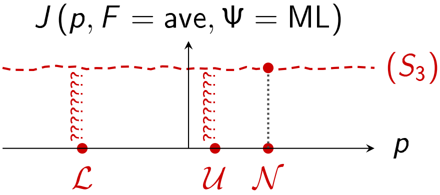




A graphical summary







is it possible to do better using discrete distributions?

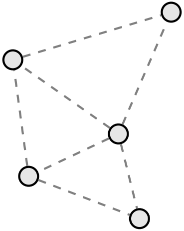
Table of Contents

Example with Bernoulli trials

disclaimer: finite precision will be handled later

Example with Bernoulli trials

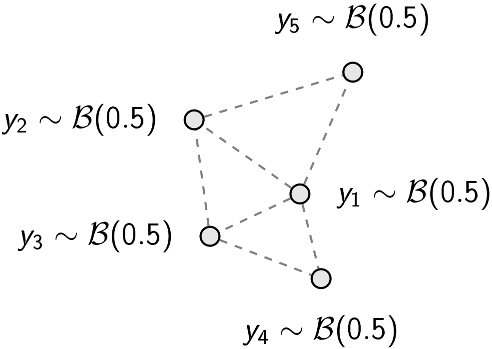
Algorithm ($M = 1$):



Example with Bernoulli trials

Algorithm ($M = 1$):

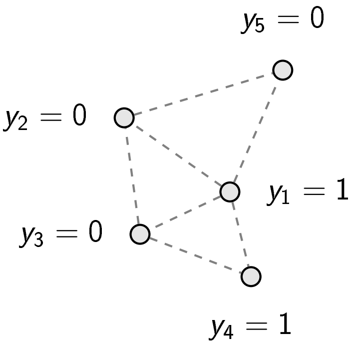
local generation with $p = \mathcal{B}(0.5)$



Example with Bernoulli trials

Algorithm ($M = 1$):

local generation with $p = \mathcal{B}(0.5)$



Example with Bernoulli trials

Algorithm ($M = 1$):

local generation with $p = \mathcal{B}(0.5)$

↓

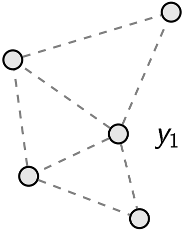
F = average consensus

$$y_5 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_2 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_1 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

$$y_3 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$



$$y_4 \to \frac{1}{S}\sum_{i=1}^{S} y_i$$

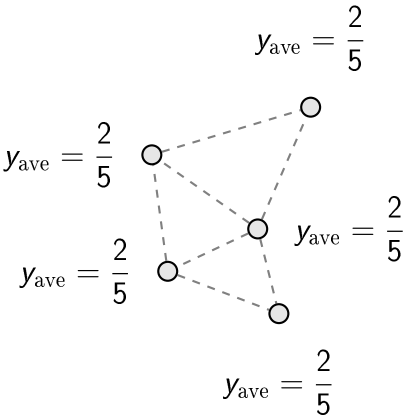
Example with Bernoulli trials

Algorithm ($M = 1$):

local generation with $p = \mathcal{B}(0.5)$

↓

F = average consensus



Example with Bernoulli trials

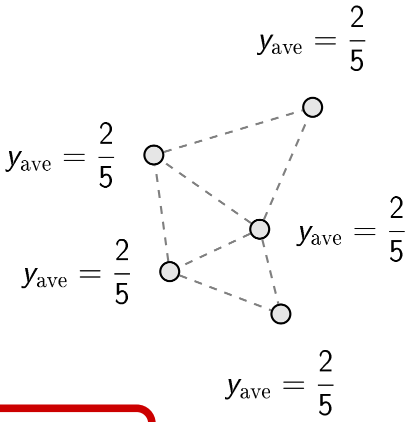
Algorithm ($M = 1$):

local generation with $p = \mathcal{B}(0.5)$

↓

F = average consensus

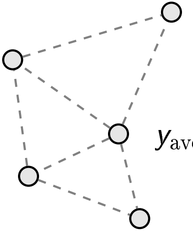
idea: estimator $\widehat{S}$ = denominator!



Example with Bernoulli trials - insights

$$y_{\text{ave}} = \frac{2}{5}$$

$$y_{\text{ave}} = \frac{2}{5}$$

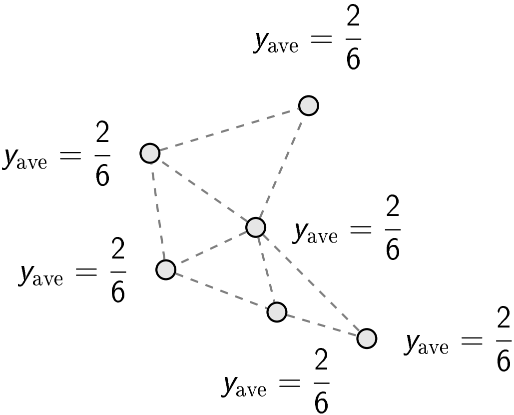


$$y_{\text{ave}} = \frac{2}{5}$$

$$y_{\text{ave}} = \frac{2}{5}$$

$$y_{\text{ave}} = \frac{2}{5}$$

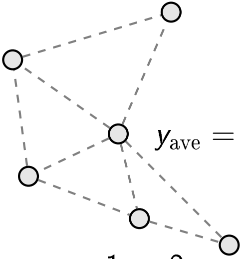
Example with Bernoulli trials - insights



Example with Bernoulli trials - insights

$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$

$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$



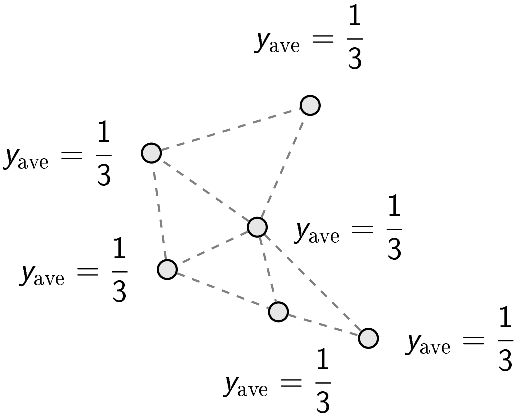
$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$

$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$

$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$

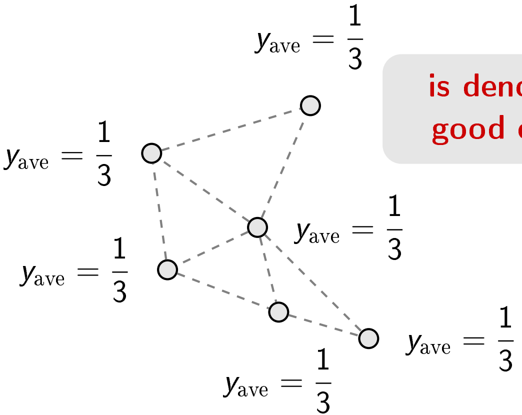
$$y_{\text{ave}} = \frac{1}{3} = \frac{2}{6} = \dots$$

Example with Bernoulli trials - insights



assumption: agents compute only coprime representations

Example with Bernoulli trials - insights



is denominator a good estimator?

assumption: agents compute only coprime representations

Statistical characterization of the estimator

Proposition

Hypotheses:

- $y_i \sim \mathcal{B}(p)$
- $y_{\text{ave}} = \frac{1}{S} \sum_{i=1}^{S} y_i = \frac{\widehat{k}}{\widehat{S}}$ ***coprime***

Statistical characterization of the estimator

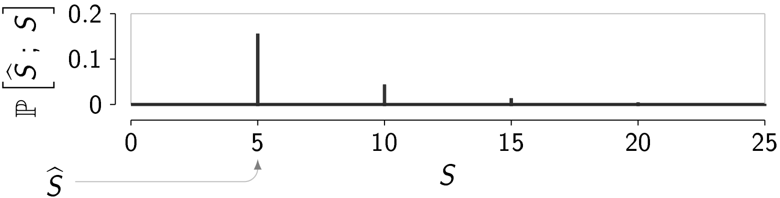
Proposition

Hypotheses:

- $y_i \sim \mathcal{B}(p)$
- $y_{\text{ave}} = \frac{1}{S} \sum_{i=1}^{S} y_i = \frac{\widehat{k}}{\widehat{S}}$ ***coprime***

Thesis:

$\widehat{S}$ = ML estimate of S for every p



Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

$$y_{\text{ave}} = \frac{\widehat{k}}{\widehat{S}} = \frac{2\widehat{k}}{2\widehat{S}} = \frac{3\widehat{k}}{3\widehat{S}} = \cdots$$

Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

$$y_{\text{ave}} = \frac{\widehat{k}}{\widehat{S}} = \frac{2\widehat{k}}{2\widehat{S}} = \frac{3\widehat{k}}{3\widehat{S}} = \ldots$$

$\widehat{S}$ agents, $\widehat{k}$ generated "1"

Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

$$y_{\text{ave}} = \frac{\widehat{k}}{\widehat{S}} = \frac{2\widehat{k}}{2\widehat{S}} = \frac{3\widehat{k}}{3\widehat{S}} = \dots$$

$2\widehat{S}$ agents, $2\widehat{k}$ generated "1"

Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

$$y_{\text{ave}} = \frac{\widehat{k}}{\widehat{S}} = \frac{2\widehat{k}}{2\widehat{S}} = \frac{3\widehat{k}}{3\widehat{S}} = \dots$$

$3\widehat{S}$ agents, $3\widehat{k}$ generated "1"

Intuition behind the ML property

Ockham's razor (William of Ockham, c. 1288 - c. 1348)

"select from among competing hypotheses the one that makes the fewest new assumptions"

$$y_{\text{ave}} = \frac{\widehat{k}}{\widehat{S}} = \frac{2\widehat{k}}{2\widehat{S}} = \frac{3\widehat{k}}{3\widehat{S}} = \cdots$$

the simplest network / hypothesis

An historical and related question

The Newton-Pepys problem (Isaac Newton, 1643 - 1727; Samuel Pepys, 1633 - 1703)

Which one is the most likely event?

1. have at least 1 six when rolling 6 dice
2. have at least 2 sixes when rolling 12 dice
3. have at least 3 sixes when rolling 18 dice

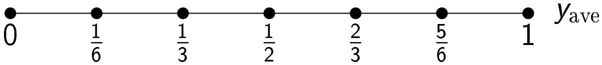
Our result:

$$\mathbb{P}\left[\text{have exactly } k \text{ sixes when rolling } kN \text{ dice}\right]$$

decreases when increasing k

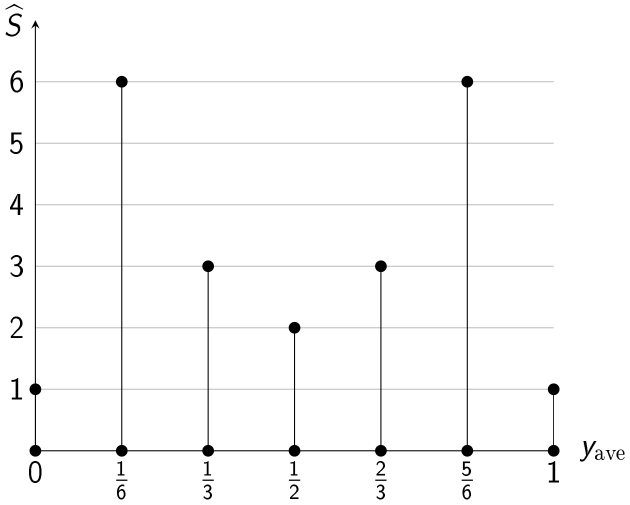
The nonlinear behavior of the estimator

assumption:
S known,
$S = 6$



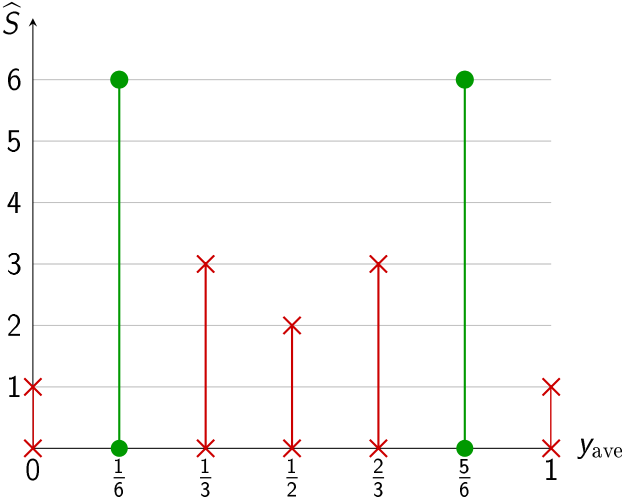
The nonlinear behavior of the estimator

assumption:
S known,
$S = 6$



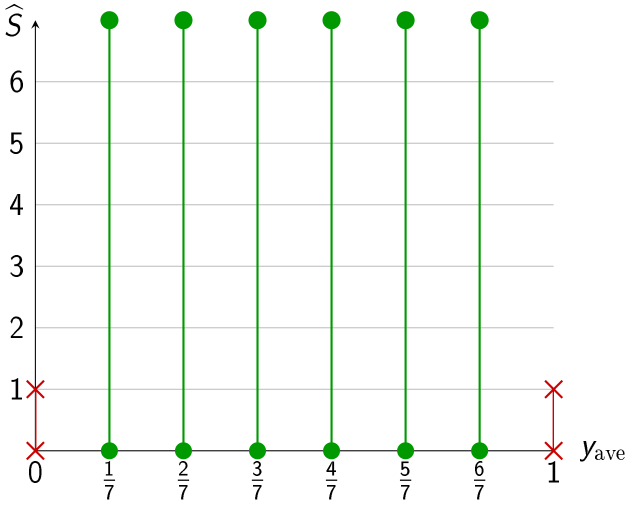
The nonlinear behavior of the estimator

assumption:
S known,
$S = 6$



The nonlinear behavior of the estimator

assumption:
S known,
$S = 7$



Connections with number theory

Definition: totative of an integer S

a positive integer $k \leq S$ which is also relatively prime to S

Connections with number theory

Definition: totative of an integer S

a positive integer $k \leq S$ which is also relatively prime to S

Definition: Euler's ϕ-function

$$\phi(S) := \text{number of totatives of } S$$

Connections with number theory

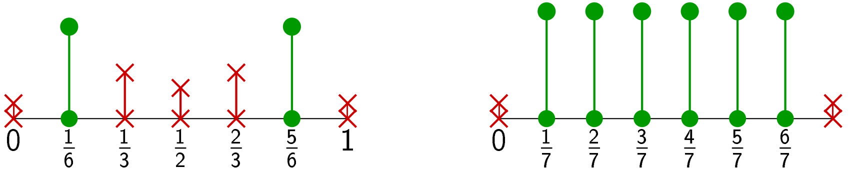
Definition: totative of an integer S

a positive integer $k \leq S$ which is also relatively prime to S

Definition: Euler's ϕ-function

$$\phi(S) := \text{number of totatives of } S$$

for our purposes, $\phi(S)$ = number of good values



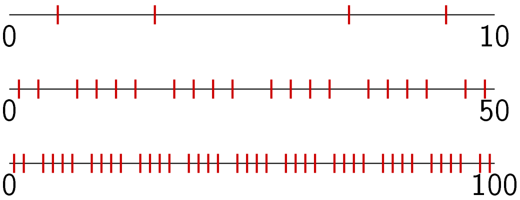
Totatives' characteristics (1/2)

Distribution: $\approx$ uniform on $\mathbb{N}$

$S = 10$: (40%)

$S = 50$: (40%)

$S = 100$: (40%)



Totatives' characteristics (1/2)

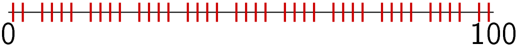
Distribution: $\approx$ uniform on $\mathbb{N}$

$S = 10$:


(40%)

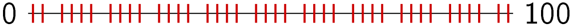
$S = 50$:


(40%)

$S = 100$:


(40%)

very important: Bernoulli's p **has not** key roles



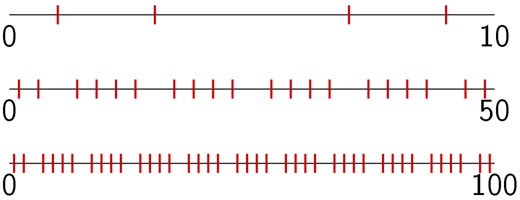
Totatives' characteristics (1/2)

Distribution: $\approx$ uniform on $\mathbb{N}$

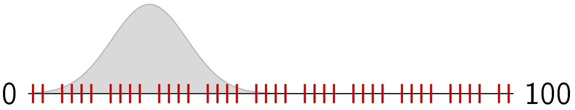
$S = 10$: (40%)

$S = 50$: (40%)

$S = 100$: (40%)



very important: Bernoulli's p **has not** key roles



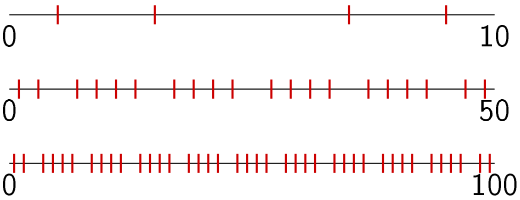
Totatives' characteristics (1/2)

Distribution: $\approx$ uniform on $\mathbb{N}$

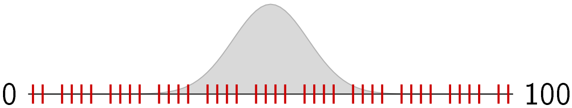
$S = 10$: (40%)

$S = 50$: (40%)

$S = 100$: (40%)



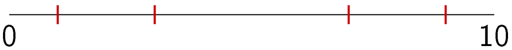
very important: Bernoulli's p **has not** key roles



Totatives' characteristics (1/2)

Distribution: $\approx$ uniform on $\mathbb{N}$

$S = 10$:



(40%)

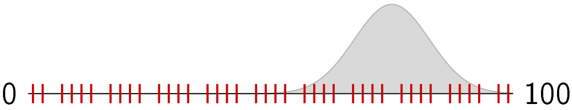
$S = 50$:

(40%)

$S = 100$:

(40%)

very important: Bernoulli's p **has not** key roles



Totatives' characteristics (2/2)

How many?

$$\phi(S) > \frac{S}{e^{\gamma} \log\log S + \frac{3}{\log\log S}} \qquad \text{i.e.} \qquad \frac{\phi(S)}{S} > 0.15 \quad \forall\, S \in [2,\ 10^{10}]$$

($\gamma \approx 0.577$, Euler-Mascheroni constant)

Totatives' characteristics (2/2)

How many?

$$\phi(S) > \frac{S}{e^{\gamma} \log \log S + \dfrac{3}{\log \log S}} \quad \text{i.e.} \quad \frac{\phi(S)}{S} > 0.15 \quad \forall\, S \in [2,\ 10^{10}]$$

($\gamma \approx 0.577$, Euler-Mascheroni constant)

an other important result:

at least 15% of the plausible y_{ave} are good ones

Totatives' characteristics (2/2)

How many?

$$\phi(S) > \frac{S}{e^{\gamma} \log\log S + \frac{3}{\log\log S}} \qquad \text{i.e.} \qquad \frac{\phi(S)}{S} > 0.15 \quad \forall\, S \in [2,\ 10^{10}]$$

($\gamma \approx 0.577$, Euler-Mascheroni constant)

an other important result:

at least 15% of the plausible y_{ave} are good ones

only 15%??

Extension to the multiple-generations case

y_1:	0	1	0	1	1	0	1	0	0	0
y_2:	1	1	1	0	0	1	1	1	0	1
y_3:	0	0	1	0	0	0	0	1	1	0
y_4:	1	1	0	0	1	1	1	1	1	1
y_5:	0	0	1	1	1	0	0	1	0	0

Extension to the multiple-generations case

y_1:	0	1	0	1	1	0	1	0	0	0
y_2:	1	1	1	0	0	1	1	1	0	1
y_3:	0	0	1	0	0	0	0	1	1	0
y_4:	1	1	0	0	1	1	1	1	1	1
y_5:	0	0	1	1	1	0	0	1	0	0

locally generated (size $= M$)

Extension to the multiple-generations case

y_1:	0	1	0	1	1	0	1	0	0	0
y_2:	1	1	1	0	0	1	1	1	0	1
y_3:	0	0	1	0	0	0	0	1	1	0
y_4:	1	1	0	0	1	1	1	1	1	1
y_5:	0	0	1	1	1	0	0	1	0	0

component-wise consensus

Extension to the multiple-generations case

y_1:	0	1	0	1	1	0	1	0	0	0
y_2:	1	1	1	0	0	1	1	1	0	1
y_3:	0	0	1	0	0	0	0	1	1	0
y_4:	1	1	0	0	1	1	1	1	1	1
y_5:	0	0	1	1	1	0	0	1	0	0
	$\widehat{S}_1$	$\widehat{S}_2$	$\widehat{S}_3$	$\widehat{S}_4$	$\widehat{S}_5$	$\widehat{S}_6$	$\widehat{S}_7$	$\widehat{S}_8$	$\widehat{S}_9$	$\widehat{S}_{10}$

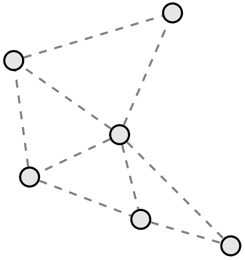
Extension to the multiple-generations case

y_1:	0	1	0	1	1	0	1	0	0	0
y_2:	1	1	1	0	0	1	1	1	0	1
y_3:	0	0	1	0	0	0	0	1	1	0
y_4:	1	1	0	0	1	1	1	1	1	1
y_5:	0	0	1	1	1	0	0	1	0	0
	$\widehat{S}_1$	$\widehat{S}_2$	$\widehat{S}_3$	$\widehat{S}_4$	$\widehat{S}_5$	$\widehat{S}_6$	$\widehat{S}_7$	$\widehat{S}_8$	$\widehat{S}_9$	$\widehat{S}_{10}$

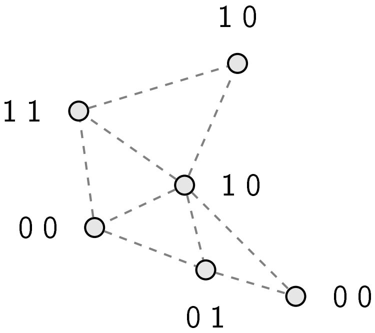
ML

$$\widehat{S} = \mathrm{LCM}\left(\left\{\widehat{S}_m\right\}\right)$$

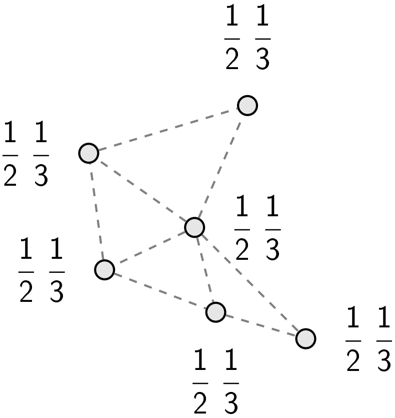
Intuition behind the $\mathrm{LCM}(\cdot)$ operation



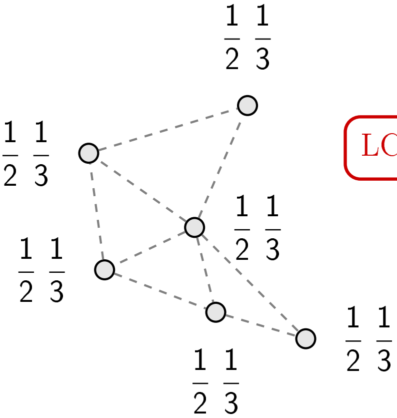
Intuition behind the $\mathrm{LCM}(\cdot)$ operation



Intuition behind the $\mathrm{LCM}(\cdot)$ operation



Intuition behind the LCM(·) operation



$\mathrm{LCM}(2,3) = 6$

Estimation performance

Main result

$$(0.5)^{S_{\max} M} \leq \mathbb{P}\left[\widehat{S} \neq S \; ; \; M\right] \leq (0.85)^{M}$$

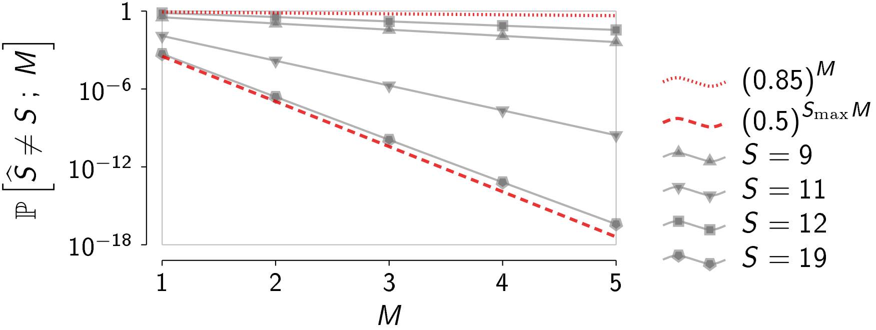


Table of Contents

Robustness issues

need to take into account several non-idealities

- quantization errors
- consensus errors

robustness properties of the various strategies are ***very different***

Robustness: Gaussian + average

Assumptions and definitions

- $y_{\text{ave}}^{\text{actual}} = (1+\delta) y_{\text{ave}}^{\text{ideal}} + \Delta$
- $\dfrac{\Delta \widehat{S}}{\widehat{S}}$:= relative error btw. ***ideal case*** and ***actual estimate***

Robustness: Gaussian + average

Assumptions and definitions

- $y_{\text{ave}}^{\text{actual}} = (1+\delta) y_{\text{ave}}^{\text{ideal}} + \Delta$
- $\dfrac{\Delta \widehat{S}}{\widehat{S}}$:= relative error btw. ***ideal case*** and ***actual estimate***

First-order approximation

$$\left| \frac{\Delta \widehat{S}}{\widehat{S}} \right| \lesssim 2\delta_{\max} + 2\sqrt{S}\Delta_{\max}$$

Robustness: Gaussian + average

Assumptions and definitions

- $y_{\text{ave}}^{\text{actual}} = (1+\delta) y_{\text{ave}}^{\text{ideal}} + \Delta$
- $\dfrac{\Delta \widehat{S}}{\widehat{S}}$:= relative error btw. ***ideal case*** and ***actual estimate***

First-order approximation

$$\left| \frac{\Delta \widehat{S}}{\widehat{S}} \right| \lesssim 2\delta_{\max} + 2\sqrt{S}\Delta_{\max}$$

well posed map

Robustness: absolutely continuous dist. + max

Assumptions and definitions

- $y_{\text{ave}}^{\text{actual}} = (1+\delta) y_{\text{ave}}^{\text{ideal}} + \Delta$
- $\dfrac{\Delta\widehat{S}}{\widehat{S}}$:= relative error btw. ***ideal case*** and ***actual estimate***

First-order approximation

$$\left|\frac{\Delta\widehat{S}}{\widehat{S}}\right| \lesssim S\delta_{\max} + S\Delta_{\max}$$

Robustness: absolutely continuous dist. + max

Assumptions and definitions

- $y_{\text{ave}}^{\text{actual}} = (1 + \delta) y_{\text{ave}}^{\text{ideal}} + \Delta$
- $\dfrac{\Delta \widehat{S}}{\widehat{S}} :=$ relative error btw. ***ideal case*** and ***actual estimate***

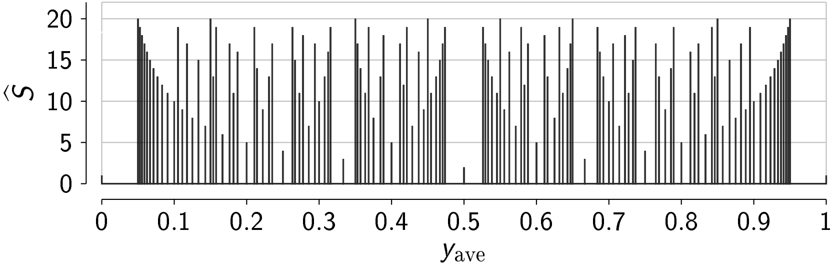
First-order approximation

$$\left| \frac{\Delta \widehat{S}}{\widehat{S}} \right| \lesssim S \delta_{\max} + S \Delta_{\max}$$

tradeoff robustness vs. performance

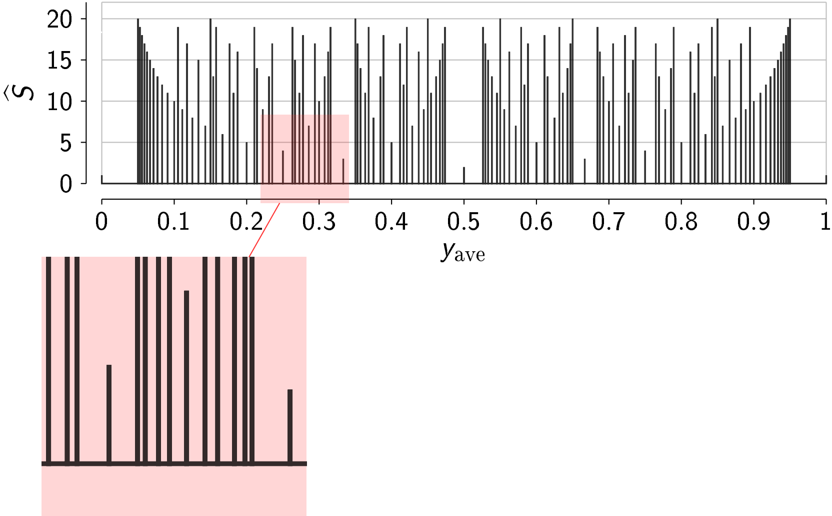
Robustness: Bernoulli + average

Extremely non-linear map (**requires $S_{\max}$**):



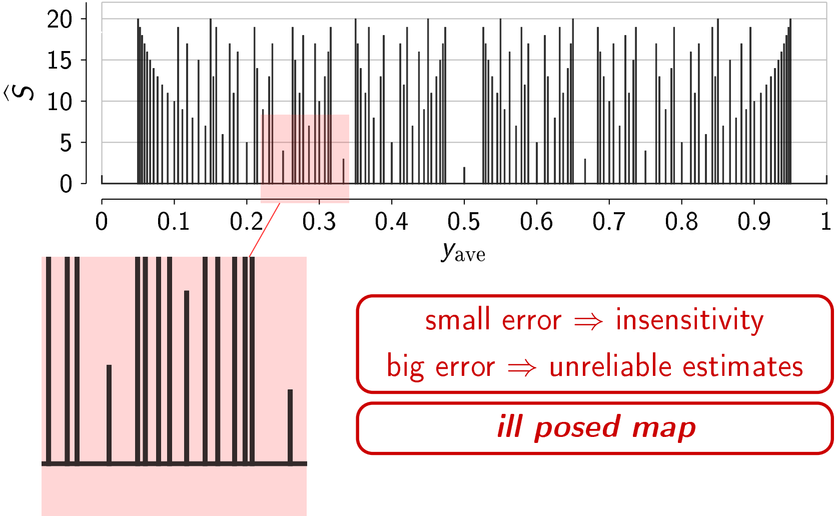
Robustness: Bernoulli + average

Extremely non-linear map (**requires $S_{\max}$**):



Robustness: Bernoulli + average

Extremely non-linear map (**requires $S_{\max}$**):

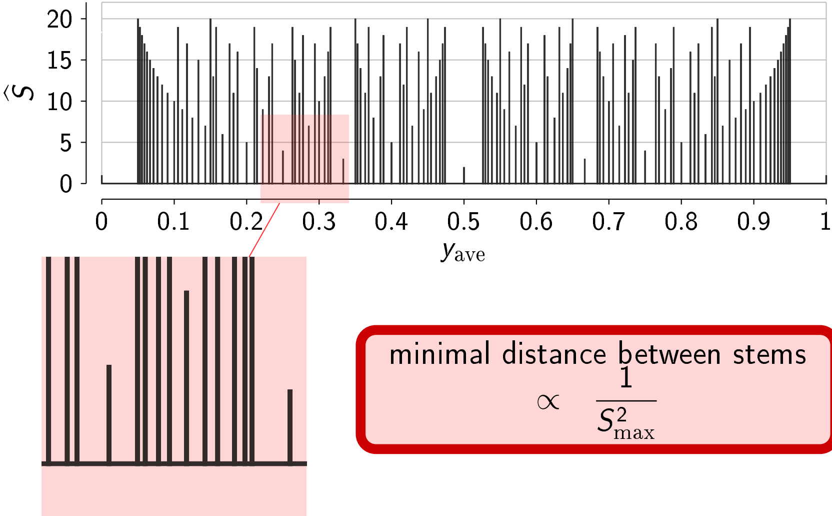


small error ⇒ insensitivity
big error ⇒ unreliable estimates

ill posed map

Robustness: Bernoulli + average

Extremely non-linear map (**requires $S_{\max}$**):



minimal distance between stems $\propto \dfrac{1}{S_{\max}^2}$

Table of Contents

Two main directions:

1. dynamic case

   (continuously run the previous algorithms and tie the results – forthcoming at 51st CDC)

Two main directions:

1. dynamic case

   (continuously run the previous algorithms and tie the results – forthcoming at 51st CDC)

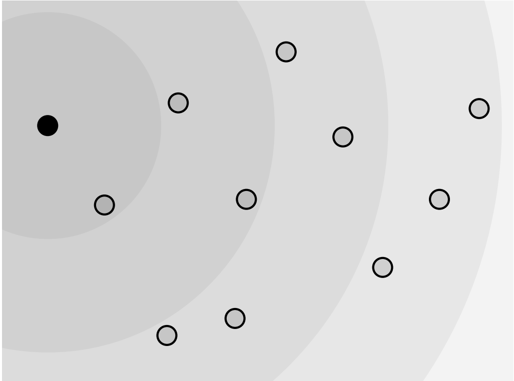
2. max-consensus based networks structure identification

structure identification with max-consensus

protocol: each agent communicates once per epoch

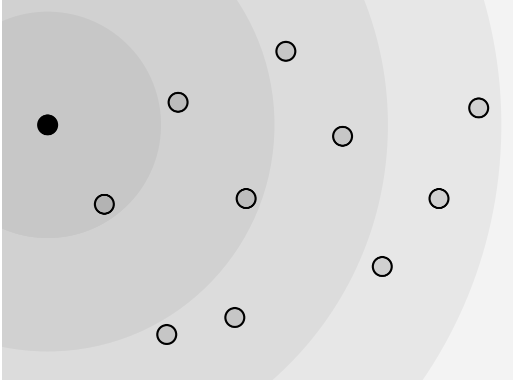
structure identification with max-consensus

protocol: each agent communicates once per epoch



structure identification with max-consensus

protocol: each agent communicates once per epoch



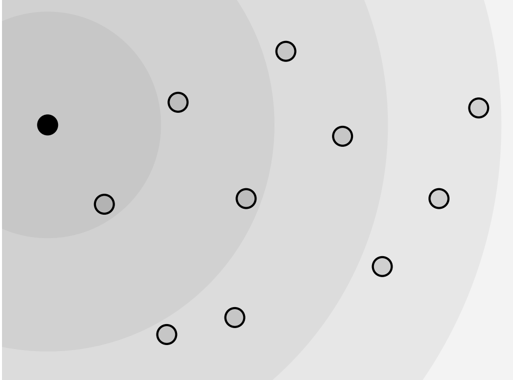
$\{y_m(t)\}$

×	×	×	×	×

$t = 0$

structure identification with max-consensus

protocol: each agent communicates once per epoch

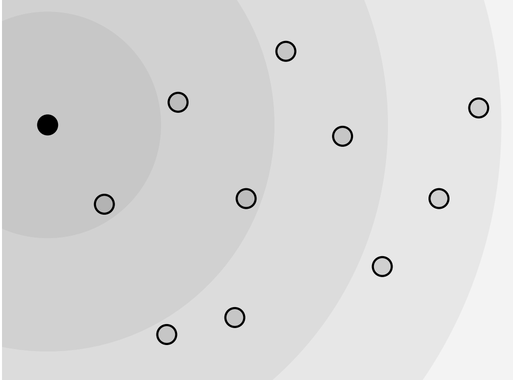


$\{y_m(t)\}$

×	×	×	×	×	$t = 0$
×	×			×	$t = 1$

structure identification with max-consensus

protocol: each agent communicates once per epoch

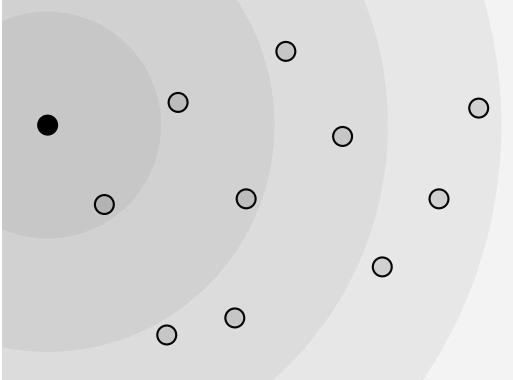


$\{y_m(t)\}$

×	×	×	×	×	$t = 0$
×	×			×	$t = 1$
	×	×			$t = 2$

structure identification with max-consensus

protocol: each agent communicates once per epoch

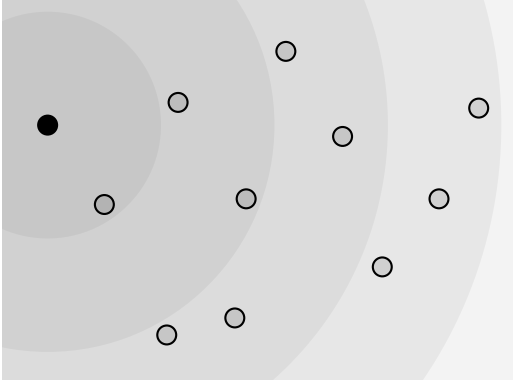


$\{y_m(t)\}$

×	×	×	×	×	$t = 0$
×	×			×	$t = 1$
	×	×			$t = 2$
	×			×	$t = 3$

structure identification with max-consensus

protocol: each agent communicates once per epoch

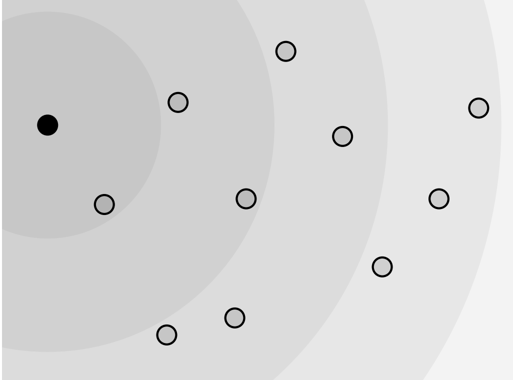


$\{y_m(t)\}$

×	×	×	×	×	$t = 0$
×	×			×	$t = 1$
	×	×			$t = 2$
	×			×	$t = 3$
			×		$t = 4$

structure identification with max-consensus

protocol: each agent communicates once per epoch



$\{y_m(t)\}$

×	×	×	×	×	$t = 0$
×	×			×	$t = 1$
	×	×			$t = 2$
	×			×	$t = 3$
			×		$t = 4$
				×	$t = 5$

Vision

develop algorithms able to detect

network faults

and give indications

for self-reconfiguration purposes

Bibliography

Varagnolo, Pillonetto, Schenato (2010)
Distributed statistical estimation of the number of nodes in Sensor Networks
IEEE Conference on Decision and Control

Varagnolo, Pillonetto, Schenato (2012)
Consensus based estim. of anonymous networks size using Bernoulli trials
American Control Conference

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

Terelius, Varagnolo (2012)
Distributed size estimation in dynamic anonymous networks
IEEE Conference on Decision and Control (in preparation)

Distributed size estimation in anonymous networks

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

Department of Information Engineering, University of Padova

February 9th, 2012

varagnolo@dei.unipd.it
http://automatica.dei.unipd.it/people/varagnolo.html