Auto-tuning procedures for distributed nonparametric regression algorithms

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

ECC 2015 - Linz

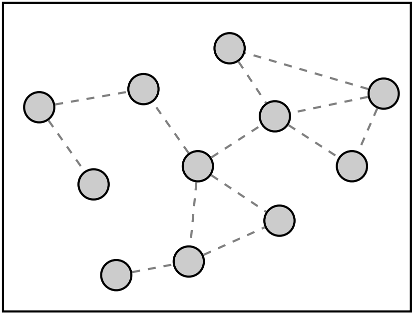
July 15, 2015

Thanks to. . .

Gianluigi Pillonetto
Univ. of Padova

Luca Schenato
Univ. of Padova

Framework



Agents:

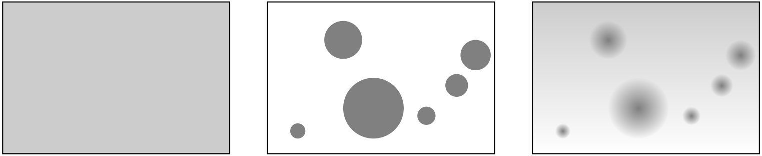
- noisily sample the same f
- limited computational & communication capabilities
- 1 measurement $\times$ agent (ease of notation)
- M measurements in total

Measurement model

$$y_m = f(x_m) + \nu_m \tag{1}$$

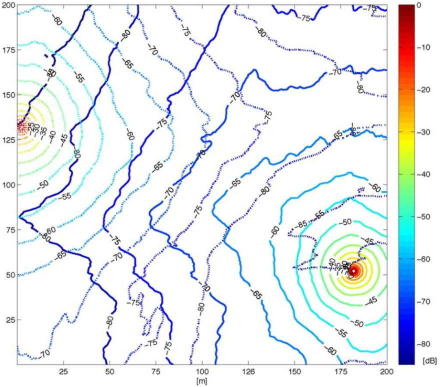
- $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ unknown ($\mathcal{X}$ compact)
- $\nu_m \perp x_m$, zero mean and variance σ^2
- $x_m \sim \mu$ i.i.d. (*agents know μ!!*)

examples of μ:



uniform   jitter   generic

Example 1 - channel gains in geographical areas



source: Dall'Anese et al., 2011

$x \in \mathbb{R}^2$: position

t : time

$f(x, t)$: channel gain

Example 2 - waves power extraction

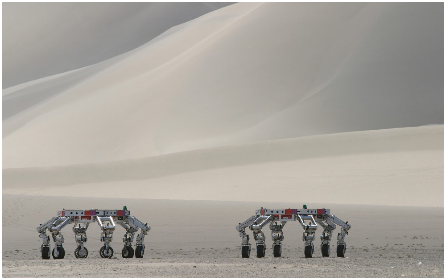
source: www.graysharboroceanenergy.com

$x \in \mathbb{R}^2$: position

t : time

$f(x, t)$: sea level

Example 3 - multi robot exploration



source: http://www-robotics.jpl.nasa.gov

$x \in \mathbb{R}^2$: position

$f(x)$: ground level

Considered cost function

$$Q(f) = \sum_{m=1}^{M} \left(y_m - f(x_m)\right)^2 + \gamma \|f\|_K^2$$

Considered cost function

$$Q(f) = \sum_{m=1}^{M} (y_m - f(x_m))^2 + \gamma \|f\|_K^2$$

lives in an infinite dimensional space

regularization factor, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Mercer kernel

Considered cost function

$$Q(f) = \sum_{m=1}^{M} (y_m - f(x_m))^2 + \gamma \|f\|_K^2$$

lives in an infinite dimensional space

regularization factor, $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ Mercer kernel

Centralized optimal solution as a Regularization Network

$$f_c = \sum_{m=1}^{M} c_m K(x_m, \cdot) \qquad \begin{bmatrix} c_1 \\ \vdots \\ c_M \end{bmatrix} = \left(\begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_M) \\ \vdots & & \vdots \\ K(x_M, x_1) & \cdots & K(x_M, x_M) \end{bmatrix} + \gamma I \right)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$$

Drawbacks

$$f_c = \sum_{m=1}^{M} c_m K(x_m, \cdot) \qquad \begin{bmatrix} c_1 \\ \vdots \\ c_M \end{bmatrix} = \left(\begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_M) \\ \vdots & & \vdots \\ K(x_M, x_1) & \cdots & K(x_M, x_M) \end{bmatrix} + \gamma I \right)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$$

- computational cost: $O(M^3)$ (inversion of $M \times M$ matrix)
- transmission cost: $O(M)$ (knowledge of whole $\{x_m, y_m\}_{m=1}^{M}$)

$$\Downarrow$$

need to find alternative solutions

Alternative centralized optimal solution (1^{st} on 2)

Structure of K implies

- $K(x_1, x_2) = \sum_{e=1}^{+\infty} \lambda_e \phi_e(x_1) \phi_e(x_2)$
- $f(x) = \sum_{e=1}^{+\infty} b_e \phi_e(x)$

$\lambda_e =$ eigenvalue
$\phi_e =$ eigenfunction

$\Rightarrow$ measurement model can be rewritten as

$$y_m = \overbrace{\left[\phi_1\left(x_m\right), \phi_2\left(x_m\right), \ldots\right]}^{C_m :=} \overbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix}}^{b :=} + \nu_m \tag{2}$$

Alternative centralized optimal solution (2^{nd} on 2)

$$b_c = \left(\frac{1}{M} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{M} \sum_{m=1}^{M} C_m^T C_m \right)^{-1} \left(\frac{1}{M} \sum_{m=1}^{M} C_m^T y_m \right) \quad (3)$$

involves infinite dimensional objects:

$$b_c = \begin{bmatrix} \bullet & \cdots & \cdots \\ \vdots & \ddots & \\ \vdots & & \ddots \end{bmatrix}^{-1} \begin{bmatrix} \bullet \\ \vdots \\ \vdots \end{bmatrix}$$

$\Rightarrow$ ***cannot be computed exactly***

Suboptimal finite dimensional solution

New estimator

$$b_r = \left(\frac{1}{M} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{M} \sum_{m=1}^{M} \left(C_m^E \right)^T C_m^E \right)^{-1} \left(\frac{1}{M} \sum_{m=1}^{M} \left(C_m^E \right)^T y_m \right)$$

- computable (involves $E \times E$ matrices and E-dimensional vectors)
- minimizes $Q^E(b) := \sum_{m=1}^{M} \left(y_m - C_m^E b \right)^2 + \gamma \sum_{e=1}^{E} \frac{b_e^2}{\lambda_e}$

Suboptimal finite dimensional solution

New estimator

$$b_r = \left(\frac{1}{M} \operatorname{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{M} \sum_{m=1}^{M} \left(C_m^E \right)^T C_m^E \right)^{-1} \left(\frac{1}{M} \sum_{m=1}^{M} \left(C_m^E \right)^T y_m \right)$$

- computable (involves $E \times E$ matrices and E-dimensional vectors)
- minimizes $Q^E(b) := \sum_{m=1}^{M} \left(y_m - C_m^E b \right)^2 + \gamma \sum_{e=1}^{E} \frac{b_e^2}{\lambda_e}$

Drawbacks

1. $O\left(E^3\right)$ computational effort
2. $O\left(E^2\right)$ transmission effort
3. must know M

Derivation of the distributed estimator

$$b_r = \left(\frac{1}{M} \mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{M} \sum_{m=1}^{M} \left(C_m^E\right)^T C_m^E \right)^{-1} \left(\frac{1}{M} \sum_{m=1}^{M} \left(C_m^E\right)^T y_m \right)$$

Consider the approximations

- $M \to M_g$ (guess)
- $\frac{1}{M} \sum_{m=1}^{M} \left(C_m^E\right)^T C_m^E \to \mathbb{E}_\mu \left[\left(C_m^E\right)^T C_m^E \right] = I$

Derivation of the distributed estimator

obtain:

$$b_d = \left(\frac{1}{M_g} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + I \right)^{-1} \left(\frac{1}{M} \sum_{m=1}^{M} \left(C_m^E \right)^T y_m \right)$$

Advantages

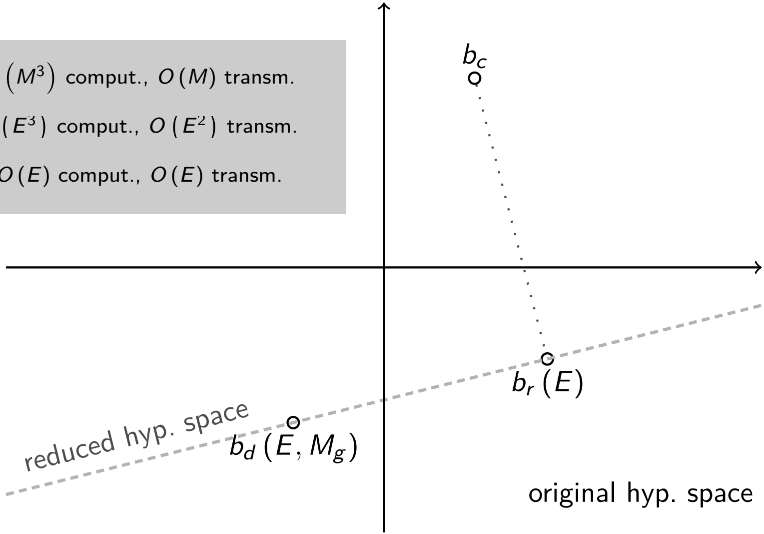
1. $O(E)$ computational effort
2. $O(E)$ transmission effort

Summary of proposed estimation schemes

b_c: $O\left(M^3\right)$ comput., $O\left(M\right)$ transsm.

b_r: $O\left(E^3\right)$ comput., $O\left(E^2\right)$ transm.

b_d: $O\left(E\right)$ comput., $O\left(E\right)$ transm.

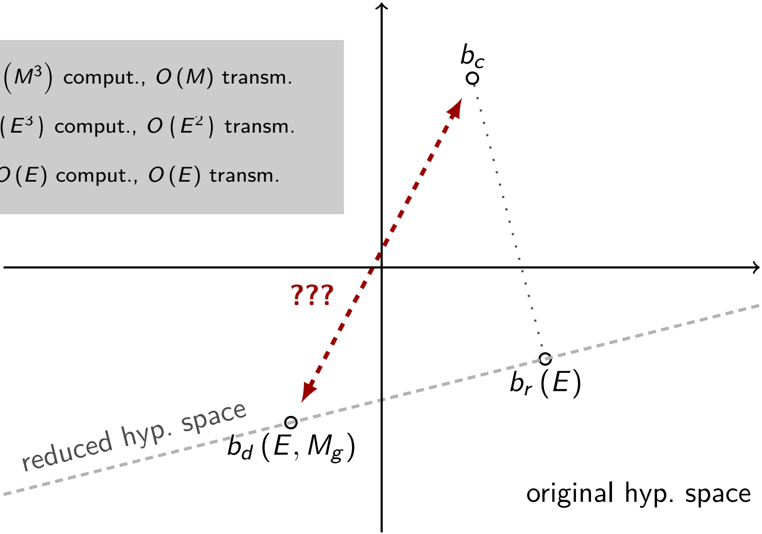


Summary of proposed estimation schemes

b_c: $O\left(M^3\right)$ comput., $O\left(M\right)$ transm.

b_r: $O\left(E^3\right)$ comput., $O\left(E^2\right)$ transm.

b_d: $O\left(E\right)$ comput., $O\left(E\right)$ transm.

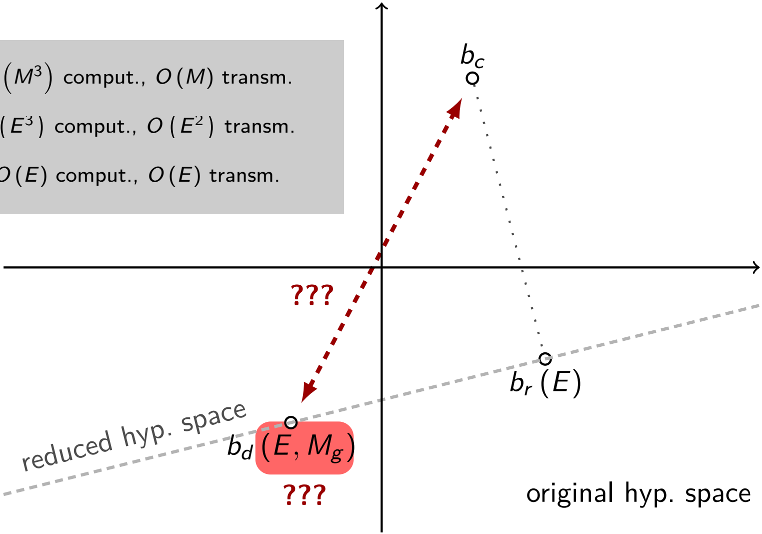


Summary of proposed estimation schemes

b_c: $O\left(M^3\right)$ comput., $O(M)$ transm.

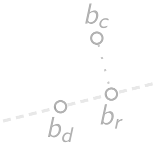
b_r: $O\left(E^3\right)$ comput., $O\left(E^2\right)$ transm.

b_d: $O(E)$ comput., $O(E)$ transm.



Tuning of the parameters - key ideas

Assumption: have some information on the energy of f

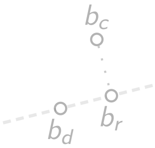


parameters to be estimated

- number of eigenfunctions E
- number of measurements M_g

Tuning of the parameters - key ideas

Assumption: have some information on the energy of f



parameters to be estimated

number of eigenfunctions E

number of measurements M_g

assure $\|b_c - b_r(E)\|$ to be sufficiently small

Tuning of the parameters - key ideas

Assumption: have some information on the energy of f

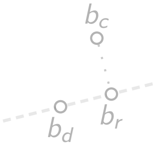
parameters to be estimated

number of eigenfunctions E

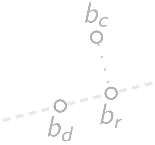
number of measurements M_g

assure $\|b_c - b_r(E)\|$ to be sufficiently small

minimize the bound on $\|b_c - b_d(E, M_g)\|$

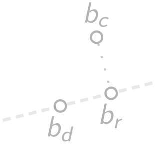


Tuning of the parameters - in practice



$$\|b_c - b_d\|_2 \;\leq\; \frac{1}{M}\sum_{m=1}^{M} |r_m| \;+\; \|U_M b_d\|_2 \;+\; \|U_C b_d\|_2$$

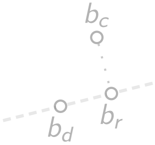
Tuning of the parameters - in practice



$$\|b_c - b_d\|_2 \;\leq\; \frac{1}{M}\sum_{m=1}^{M} |r_m| \;+\; \|U_M b_d\|_2 \;+\; \|U_C b_d\|_2$$

local residuals, func. of M_g

Tuning of the parameters - in practice

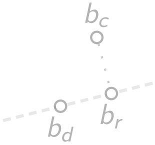


func. of M_g and $\propto \dfrac{1}{M_{\min}} - \dfrac{1}{M_{\max}}$

$$\|b_c - b_d\|_2 \leq \frac{1}{M}\sum_{m=1}^{M} |r_m| + \|U_M b_d\|_2 + \|U_C b_d\|_2$$

local residuals, func. of M_g

Tuning of the parameters - in practice



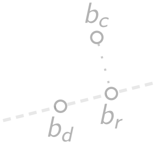
func. of M_g and $\propto \frac{1}{M_{\min}} - \frac{1}{M_{\max}}$

$$\|b_c - b_d\|_2 \leq \frac{1}{M}\sum_{m=1}^{M} |r_m| + \|U_M b_d\|_2 + \|U_C b_d\|_2$$

local residuals, func. of M_g

func. of M_g and $\propto I - \frac{1}{M}\sum_{m=1}^{M}\left(C_m^E\right)^T C_m^E$

Tuning of the parameters - in practice



func. of M_g and $\propto \frac{1}{M_{\min}} - \frac{1}{M_{\max}}$

$$\|b_c - b_d\|_2 \;\leq\; \frac{1}{M}\sum_{m=1}^{M} |r_m| \;+\; \|U_M b_d\|_2 \;+\; \|U_C b_d\|_2$$

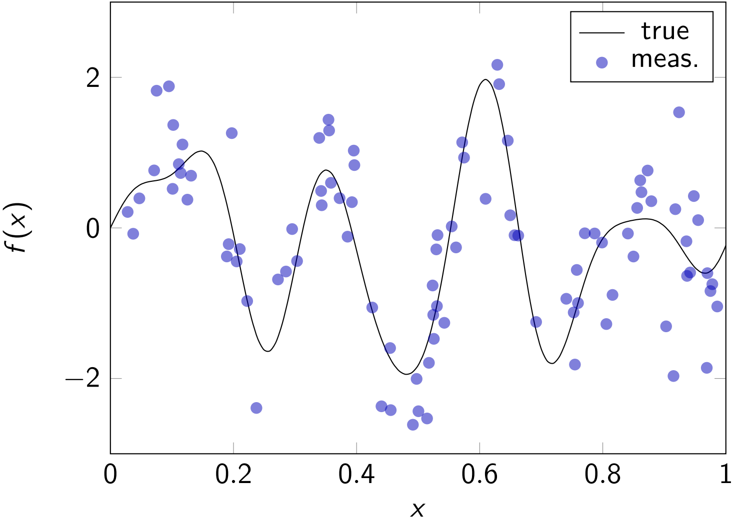
local residuals, func. of M_g

func. of M_g and $\propto I - \frac{1}{M}\sum_{m=1}^{M}\left(C_m^E\right)^T C_m^E$

computable through distributed MC

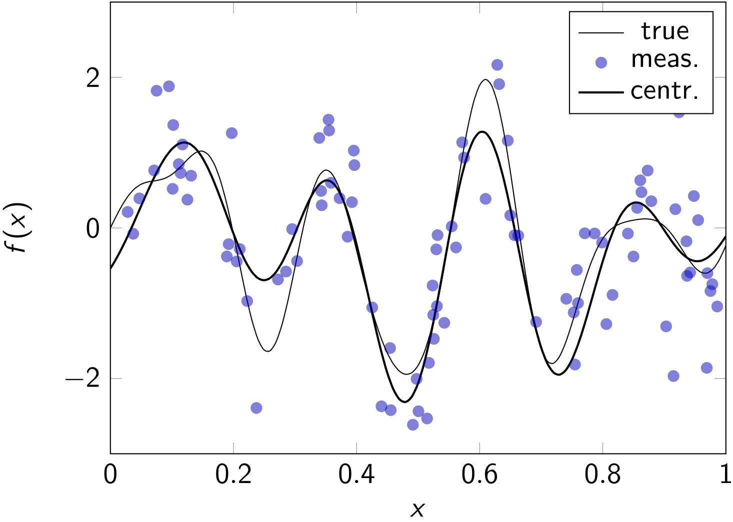
Regression strategy effectiveness example

$M = 100$, $E = 20$, $M_{\min} = 90$, $M_{\max} = 110$, SNR ≈ 2.5



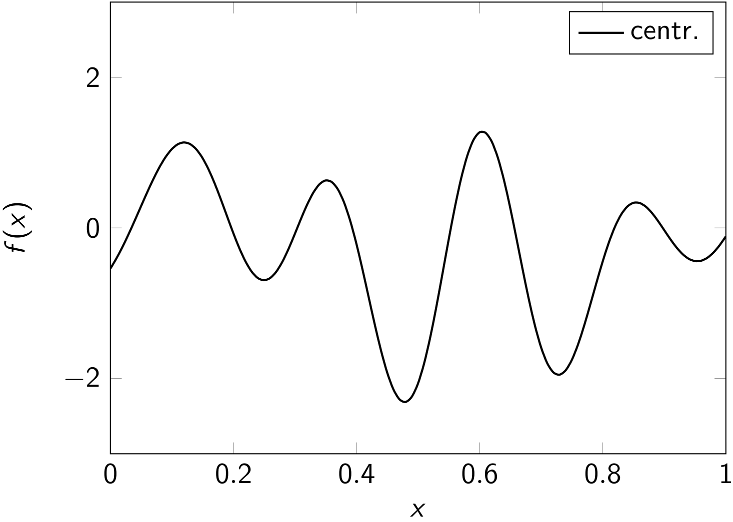
Regression strategy effectiveness example

$M = 100$, $E = 20$, $M_{\min} = 90$, $M_{\max} = 110$, SNR ≈ 2.5



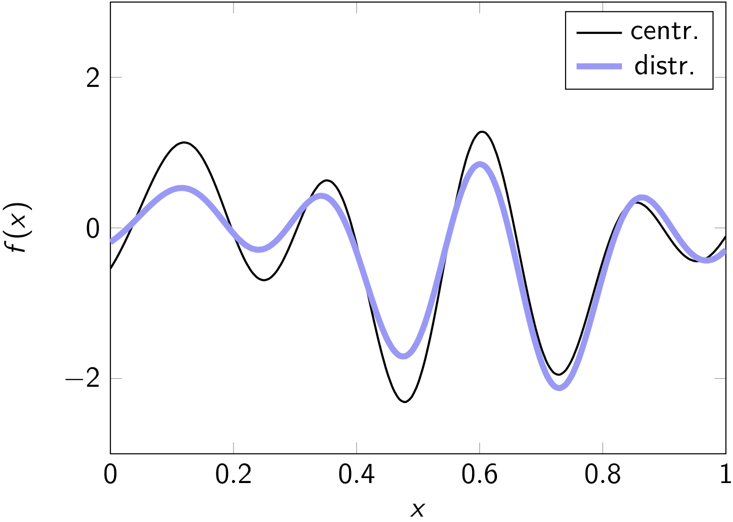
Regression strategy effectiveness example

$M = 100$, $E = 20$, $M_{\min} = 90$, $M_{\max} = 110$, SNR ≈ 2.5



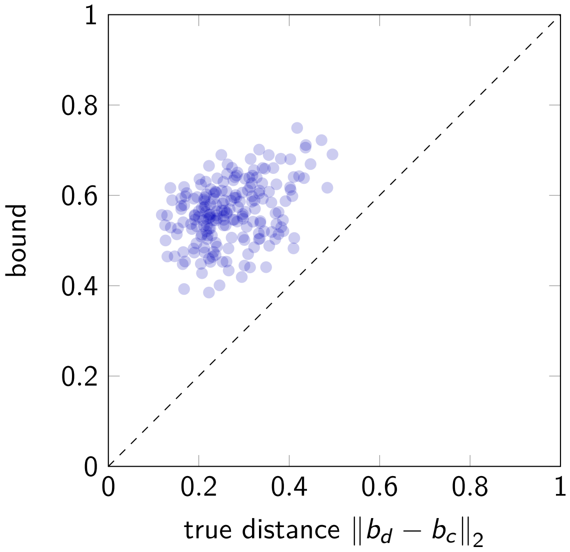
Regression strategy effectiveness example

$M = 100$, $E = 20$, $M_{\min} = 90$, $M_{\max} = 110$, SNR ≈ 2.5



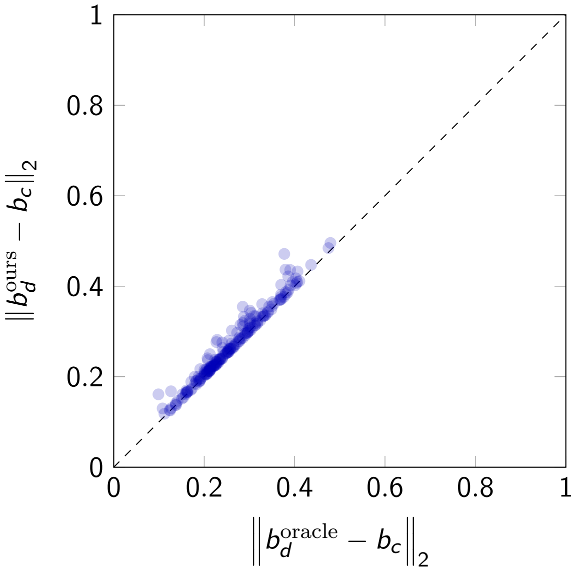
Accuracy of the computed bound

$M = 100,\ E = 20,\ M_{\min} = 90,\ M_{\max} = 110$



Comparison with oracle

$M = 100,\ E = 20,\ M_{\min} = 90,\ M_{\max} = 110$



Conclusions and future works

Conclusions

Strategy:

- is effective and easy to be implemented
- has self-evaluation capabilities
- has self-tuning capabilities

Future works

- exploit statistical knowledge about M
- incorporate effects of finite number of steps in consensus algorithms
- extend to dynamic scenarios (long term objective)

Auto-tuning procedures for distributed nonparametric regression algorithms

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

ECC 2015 - Linz

July 15, 2015

damiano.varagnolo@ltu.se