Distributed nonparametric regression in multi-agent systems

Damiano Varagnolo
(joint work with Gianluigi Pillonetto and Luca Schenato)

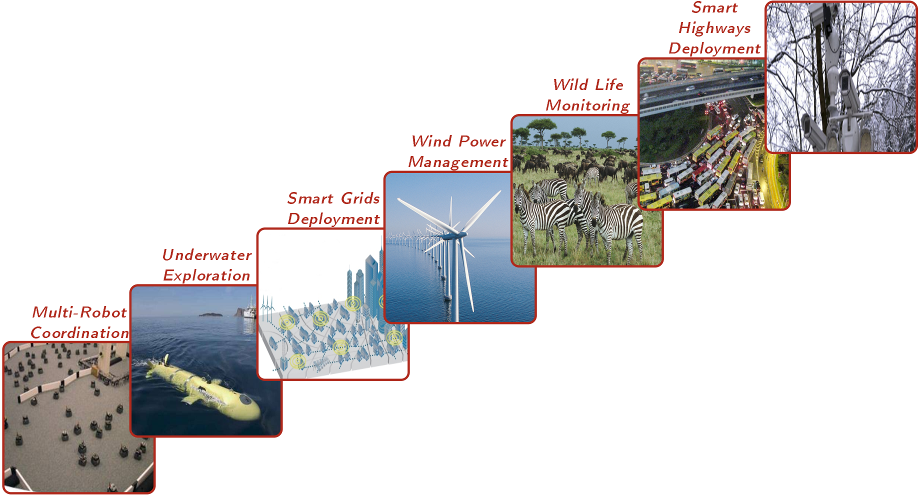
Department of Information Engineering - University of Padova

March, 25th 2011

DEPARTMENT OF
INFORMATION
ENGINEERING
UNIVERSITY OF PADOVA

entirely written in LaTeX2ε using Beamer and TikZ

Multi-agent systems: examples of applications



Multi-Robot Coordination

Underwater Exploration

Smart Grids Deployment

Wind Power Management

Wild Life Monitoring

Smart Highways Deployment

Smart Surveillance Implementation

Generic and important problem

Assumption

noisy measurements of

$$f(x, t) \; : \; \mathbb{R}^3 \times \mathbb{R} \mapsto \mathbb{R}$$

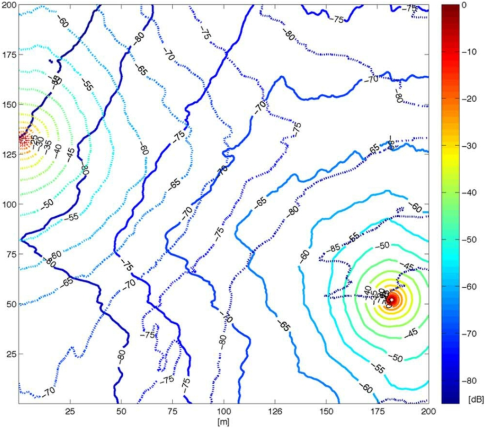
that are

- non uniformly sampled in space x
- non uniformly sampled in time t
- taken by different agents

Objective

smoothing in space (x) and forecast in time (t) the quantity $f(x, t)$

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

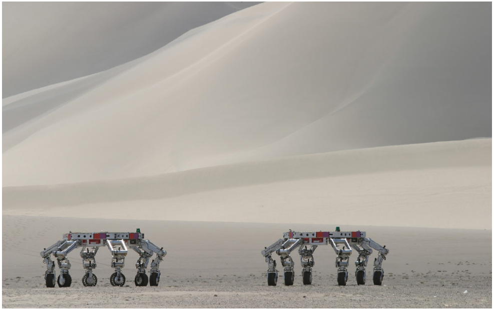
source: www.graysharboroceanenergy.com

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

Problems and difficulties

Information-related problems

- non-uniform samplings both in time and in space
- unknown dynamics of f
- unknown or extremely complex correlations in time and space

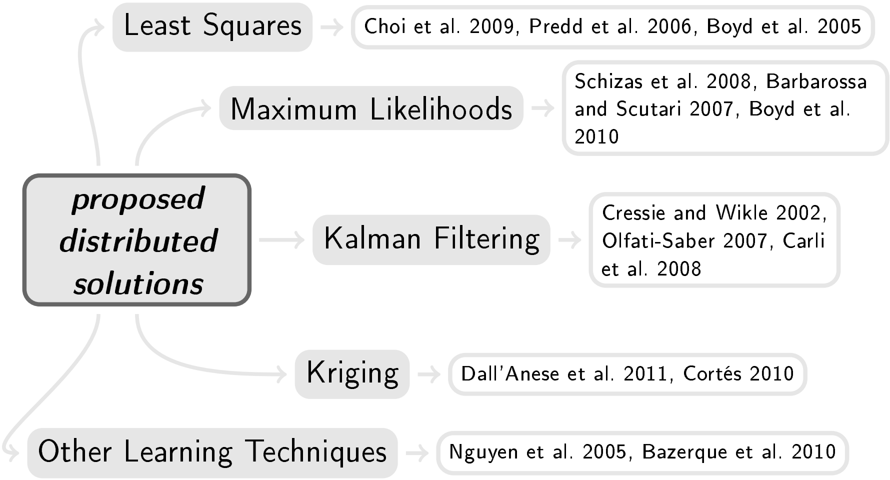
Hardware-related problems

- energy & computational & memory & bandwidth limitations

Framework-related problems

- mobile and time varying network

State of the art



proposed distributed solutions

Least Squares → Choi et al. 2009, Predd et al. 2006, Boyd et al. 2005

Maximum Likelihoods → Schizas et al. 2008, Barbarossa and Scutari 2007, Boyd et al. 2010

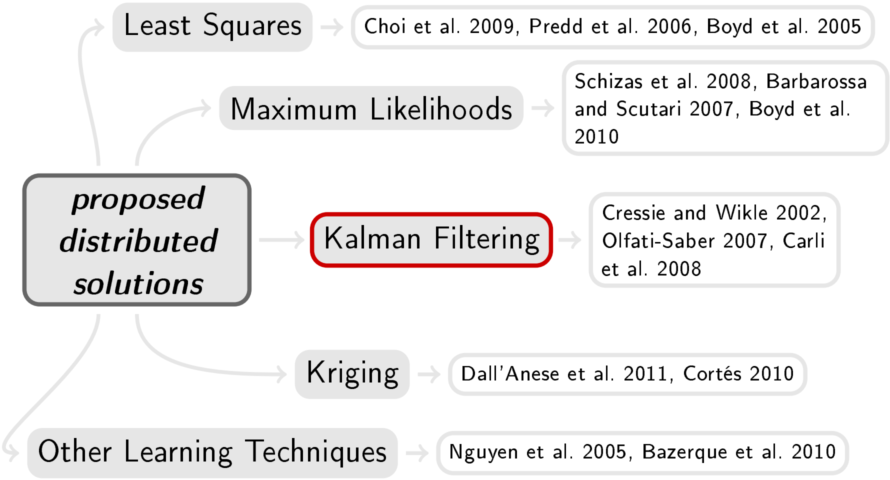
Kalman Filtering → Cressie and Wikle 2002, Olfati-Saber 2007, Carli et al. 2008

Kriging → Dall'Anese et al. 2011, Cortés 2010

Other Learning Techniques → Nguyen et al. 2005, Bazerque et al. 2010

State of the art

dynamic scenarios



proposed distributed solutions

Least Squares → Choi et al. 2009, Predd et al. 2006, Boyd et al. 2005

Maximum Likelihoods → Schizas et al. 2008, Barbarossa and Scutari 2007, Boyd et al. 2010

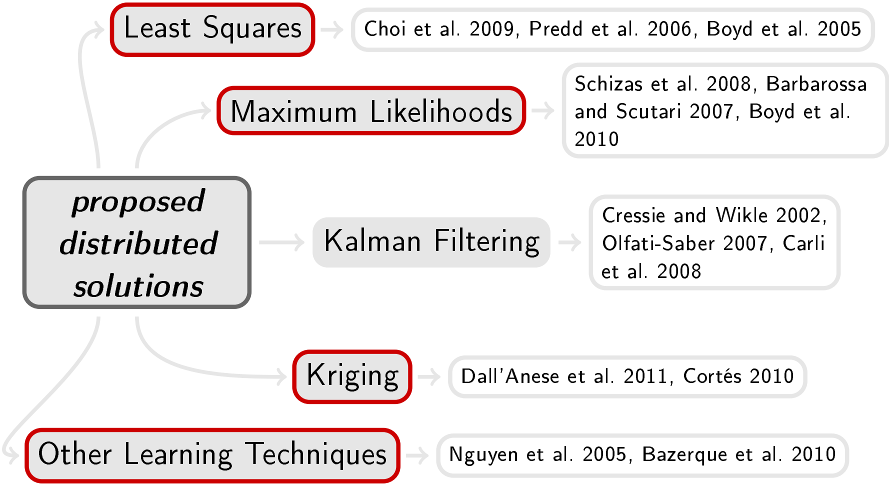
Kalman Filtering → Cressie and Wikle 2002, Olfati-Saber 2007, Carli et al. 2008

Kriging → Dall'Anese et al. 2011, Cortés 2010

Other Learning Techniques → Nguyen et al. 2005, Bazerque et al. 2010

State of the art

static scenarios



proposed distributed solutions

Least Squares → Choi et al. 2009, Predd et al. 2006, Boyd et al. 2005

Maximum Likelihoods → Schizas et al. 2008, Barbarossa and Scutari 2007, Boyd et al. 2010

Kalman Filtering → Cressie and Wikle 2002, Olfati-Saber 2007, Carli et al. 2008

Kriging → Dall'Anese et al. 2011, Cortés 2010

Other Learning Techniques → Nguyen et al. 2005, Bazerque et al. 2010

State of the art - Vision

obtain

$$\widehat{f}(x, t) = \Psi\left(\text{past measurements}, \theta\right)$$

being

- distributed
- capable of both smoothing and prediction

State of the art - Vision

obtain

$$\widehat{f}(x,t) = \Psi\left(\text{past measurements}\right)$$

being

- distributed
- capable of both smoothing and prediction

Our approach

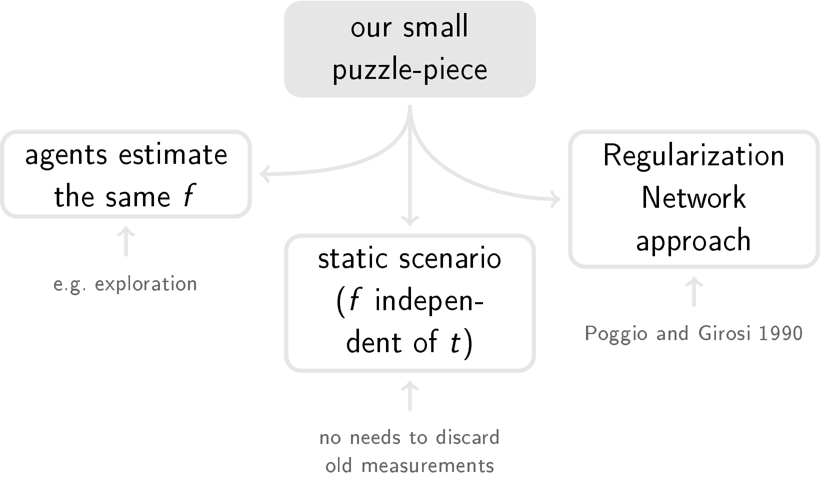
nonparametric: $\Psi(\cdot)$ lives in an ***infinite dimensional space***

Why should we use a nonparametric approach?

Motivations

- it could be difficult or even impossible to define a parametric model (e.g. when only regularity assumptions are available)
- parametric models could involve a large number of parameters (could require nonlinear optimization techniques)
- lead to convex optimization problems
- consistent, i.e. $\widehat{f} \to f$ when # measurements $\to \infty$ (De Nicolao, Ferrari-Trecate, 1999)

State of the art - where we actually contribute



our small puzzle-piece

agents estimate the same f

e.g. exploration

static scenario (f independent of t)

no needs to discard old measurements

Regularization Network approach

Poggio and Girosi 1990

Goal of the current presentation

completely self-organizing multi-agent regression strategy

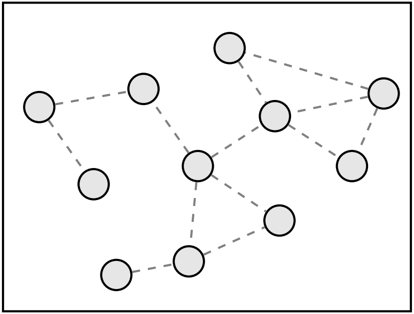
Benefits of self-organization

- widest applicability
- robustness w.r.t. human error

Requirements

- simple strategy
- quantifiable performances
- automatic tuning of the parameters

Framework



Sensors:

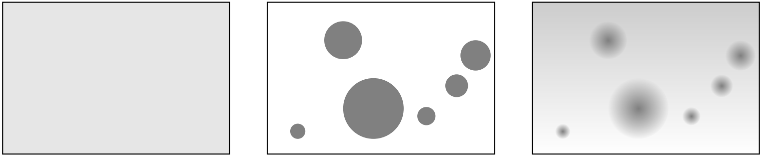
- $\# = S$, identical
- sample the same noisy function
- limited computational & communication capabilities
- want to collaborate
- 1 measurement × sensor (ease of notation)

Measurement model

$$y_i = f\left(x_i\right) + \nu_i \tag{1}$$

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

examples of μ:



uniform jitter generic

Regularization Networks

$$Q(f) = \sum_{i=1}^{S} (y_i - f(x_i))^2 + \gamma \|f\|_K^2$$

Centralized optimal solution

$$f_c = \sum_{i=1}^{S} c_i K(x_i, \cdot) \qquad \begin{bmatrix} c_1 \\ \vdots \\ c_S \end{bmatrix} = \left(\begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_S) \\ \vdots & & \vdots \\ K(x_S, x_1) & \cdots & K(x_S, x_S) \end{bmatrix} + \gamma I \right)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_S \end{bmatrix}$$

Regularization Networks

$$Q\left(f\right)=\sum_{i=1}^{S}\left(y_i-f\left(x_i\right)\right)^2+\gamma\left\|f\right\|_K^2$$

lives in an infinite
dimensional space

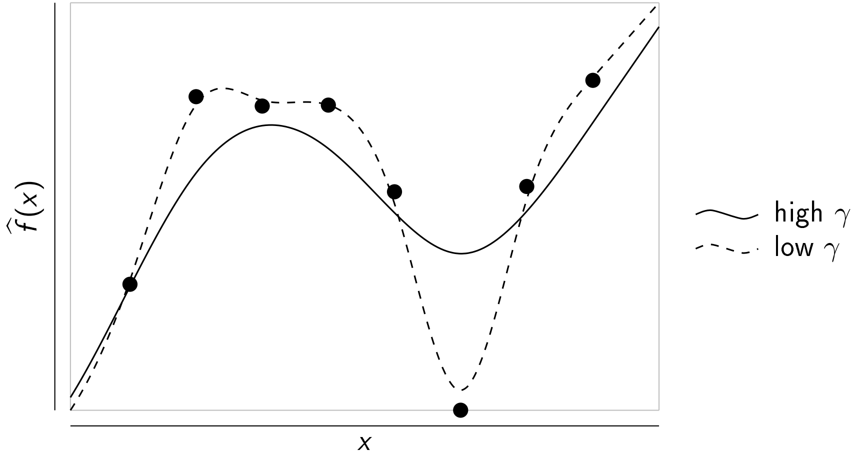
regularization factor,
$K:\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}$
Mercer kernel

Centralized optimal solution

$$f_c=\sum_{i=1}^{S}c_iK\left(x_i,\cdot\right)\qquad\begin{bmatrix}c_1\\ \vdots\\ c_S\end{bmatrix}=\left(\begin{bmatrix}K\left(x_1,x_1\right)&\cdots&K\left(x_1,x_S\right)\\ \vdots&&\vdots\\ K\left(x_S,x_1\right)&\cdots&K\left(x_S,x_S\right)\end{bmatrix}+\gamma I\right)^{-1}\begin{bmatrix}y_1\\ \vdots\\ y_S\end{bmatrix}$$

Example - Cubic splines smoothing

Effects of variation of γ for a given K



Drawbacks

$$f_c = \sum_{i=1}^{S} c_i K(x_i, \cdot) \qquad \begin{bmatrix} c_1 \\ \vdots \\ c_S \end{bmatrix} = \left(\begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_S) \\ \vdots & & \vdots \\ K(x_S, x_1) & \cdots & K(x_S, x_S) \end{bmatrix} + \gamma I \right)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_S \end{bmatrix}$$

- computational cost: $O(S^3)$ (inversion of $S \times S$ matrix)
- transmission cost: $O(S)$ (knowledge of whole $\{x_i, y_i\}_{i=1}^{S}$)

$\Downarrow$

need to find alternative solutions

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

① consider that

$$K(x_1, x_2) = \sum_{e=1}^{+\infty} \lambda_e \phi_e(x_1) \phi_e(x_2) \qquad \begin{aligned} \lambda_e &= \text{eigenvalue} \\ \phi_e &= \text{eigenfunction} \end{aligned} \tag{2}$$

② rewrite the measurement model:

$$y_i = \overbrace{\left[\phi_1\left(x_i\right), \phi_2\left(x_i\right), \ldots\right]}^{C_i :=} \overbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix}}^{b :=} + \nu_i \tag{3}$$

③ rewrite the cost function:

$$Q\left(b\right) = \sum_{i=1}^{S} \left(y_i - C_i b\right)^2 + \gamma \sum_{e=1}^{+\infty} \frac{b_e^2}{\lambda_e} \tag{4}$$

Alternative centralized optimal solution (2nd on 2)

$$b_c = \left(\frac{1}{S} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} C_i^T C_i \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} C_i^T y_i \right) \qquad (5)$$

involves infinite dimensional objects:

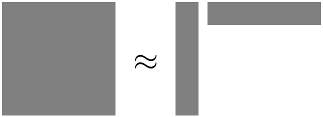
$$b_c = \begin{bmatrix} \bullet & \cdots & \cdots \\ \vdots & \ddots & \\ \vdots & & \ddots \end{bmatrix}^{-1} \begin{bmatrix} \bullet \\ \vdots \\ \vdots \end{bmatrix}$$

$\Rightarrow$ *cannot be computed exactly*

Hypothesis space reduction

PCA

$$P \geq 0 \qquad\qquad P =$$



Connection with PCA

$\mathrm{span}\left\langle \phi_1, \dots, \phi_E \right\rangle$ is the E-dimensional subspace that captures the biggest part of the energy of the estimate ($E \ll S$)

Implication: new measurement model

$$y_i = C_i^E b + \nu_i$$

with $C_i^E := \left[\phi_1(x_i), \dots, \phi_E(x_i)\right]$ $\qquad b := \left[b_1, \dots, b_E\right]^T$

Suboptimal finite dimensional solution

New estimator

$$b_r = \left(\frac{1}{S} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

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

Suboptimal finite dimensional solution

New estimator

$$b_r = \left(\frac{1}{S} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

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

Drawbacks

① $O(E^3)$ computational effort

② $O(E^2)$ transmission effort

③ must know S

Derivation of the distributed estimator

$$b_r = \left(\frac{1}{S} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

Consider the approximations

- $S \to S_g$ (guess, will be tuned later on)
- $\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \to \mathbb{E} \left[\left(C_i^E \right)^T C_i^E \right] = I$

Derivation of the distributed estimator

obtain:

$$\boxed{b_d = \left(\frac{1}{S_g} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + I \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)}$$

Advantages

① $O\left(E\right)$ computational effort

② $O\left(E\right)$ transmission effort

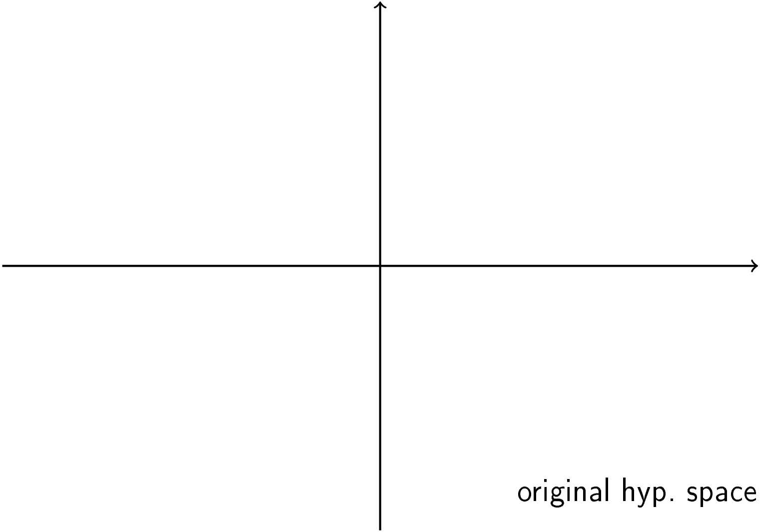
Summary of proposed estimation schemes

$$b_c = \left(\frac{1}{S} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} C_i^T C_i \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} C_i^T y_i \right)$$

$$b_r = \left(\frac{1}{S} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

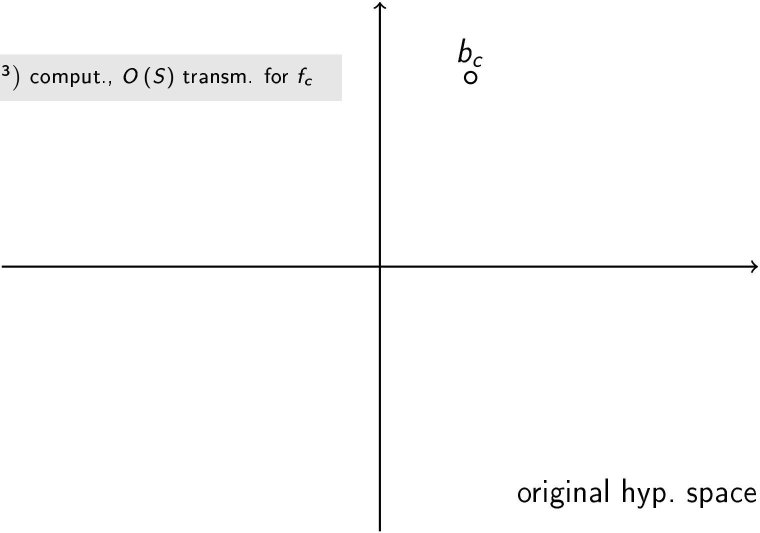
$$b_d = \left(\frac{1}{S_g} \mathrm{diag} \left(\frac{\gamma}{\lambda_e} \right) + I \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

Summary of proposed estimation schemes



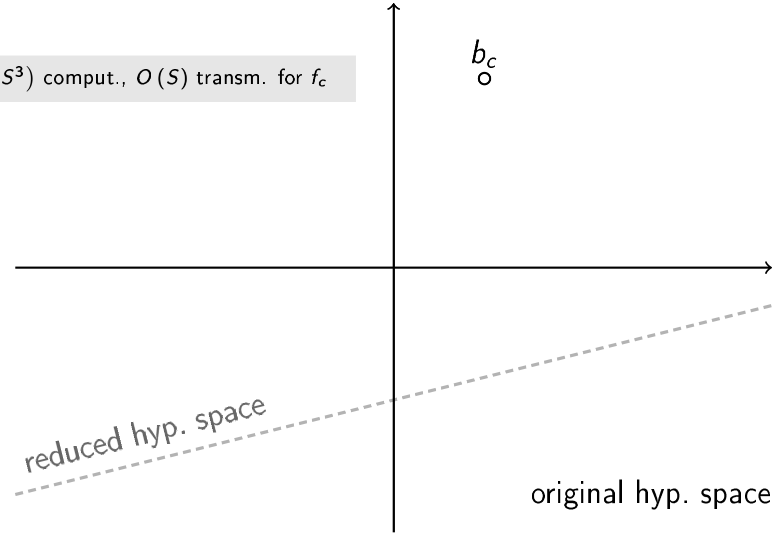
Summary of proposed estimation schemes

b_c: $O\left(S^3\right)$ comput., $O\left(S\right)$ transm. for f_c



Summary of proposed estimation schemes

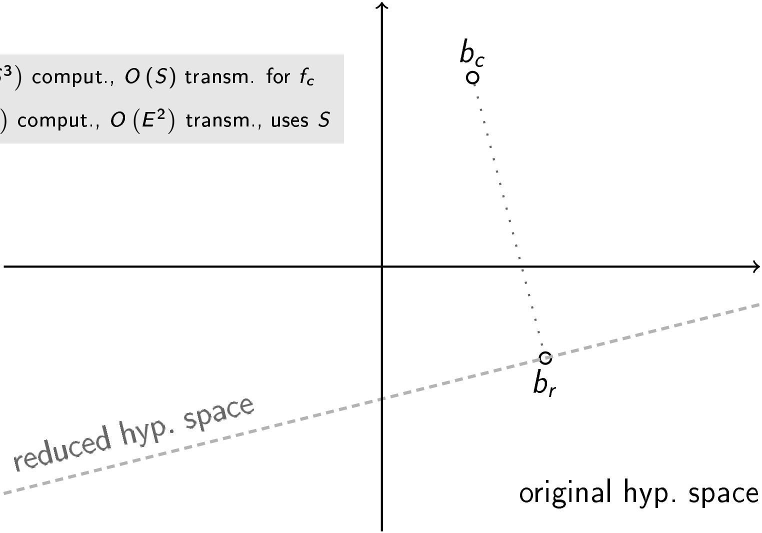
b_c: $O\left(S^3\right)$ comput., $O\left(S\right)$ transm. for f_c



Summary of proposed estimation schemes

b_c: $O\left(S^3\right)$ comput., $O\left(S\right)$ transm. for f_c

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

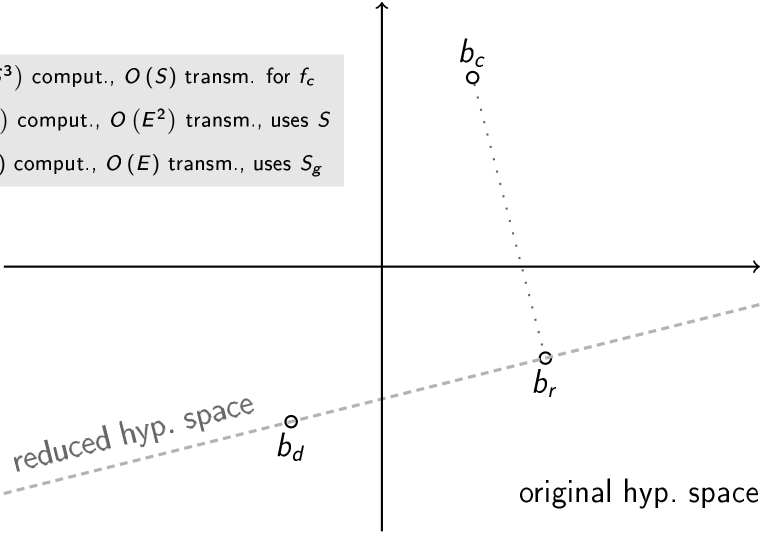


Summary of proposed estimation schemes

b_c: $O\left(S^3\right)$ comput., $O\left(S\right)$ transm. for f_c

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

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

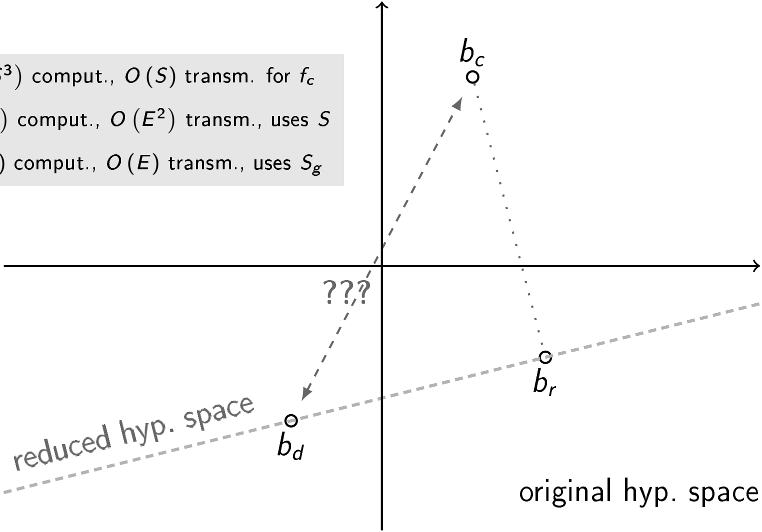


Summary of proposed estimation schemes

b_c: $O\left(S^3\right)$ comput., $O\left(S\right)$ transm. for f_c

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

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

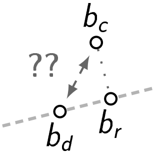


Open problems

quantification of performance

how to tune E *and* S_g

Quantification of performances: first results



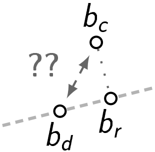
$$b_c = \left(\frac{1}{S} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} C_i^T C_i \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} C_i^T y_i \right)$$

$$b_r = \left(\frac{1}{S} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

$$b_d = \left(\frac{1}{S_g} \mathrm{diag}\left(\frac{\gamma}{\lambda_e} \right) + I \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E \right)^T y_i \right)$$

$$\left\| b_c - b_d \right\|_2 \ \leq \ \alpha \left\| b_r - b_d \right\|_2 \ + \ \frac{1}{S} \sum_{i=1}^{S} \left| r_i \right|$$

Quantification of performances: first results



$$b_c = \left(\frac{1}{S} \mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{S} \sum_{i=1}^{S} C_i^T C_i \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} C_i^T y_i \right)$$

$$b_r = \left(\frac{1}{S} \mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E\right)^T C_i^E \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E\right)^T y_i \right)$$

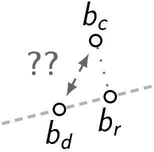
$$b_d = \left(\frac{1}{S_g} \mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + I \right)^{-1} \left(\frac{1}{S} \sum_{i=1}^{S} \left(C_i^E\right)^T y_i \right)$$

$$\|b_c - b_d\|_2 \leq \alpha \|b_r - b_d\|_2 + \frac{1}{S} \sum_{i=1}^{S} |r_i|$$

computed offline

average of local residuals

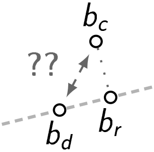
Quantification of performances



$$\|b_c - b_d\|_2 \leq \alpha \|b_r - b_d\|_2 + |r_{\mathrm{ave}}|$$

$$\|b_r - b_d\|_2 \;\leq\; \|U_S b_d\|_2 \;+\; \|U_C b_d\|_2$$

Quantification of performances

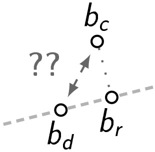


$$\left\| b_c - b_d \right\|_2 \leq \alpha \left\| b_r - b_d \right\|_2 + \left| r_{\text{ave}} \right|$$

$$\left\| b_r - b_d \right\|_2 \ \leq \ \left\| U_S b_d \right\|_2 \ + \ \left\| U_C b_d \right\|_2$$

$$\propto \frac{1}{S_{\min}} - \frac{1}{S_{\max}}$$

Quantification of performances



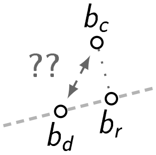
$$\|b_c - b_d\|_2 \leq \alpha \|b_r - b_d\|_2 + |r_{\mathrm{ave}}|$$

$$\|b_r - b_d\|_2 \leq \|U_S b_d\|_2 + \|U_C b_d\|_2$$

$$\propto \frac{1}{S_{\min}} - \frac{1}{S_{\max}}$$

$$\propto I - \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E\right)^T C_i^E$$

Quantification of performances



$$\|b_c - b_d\|_2 \le \alpha \|b_r - b_d\|_2 + |r_{\text{ave}}|$$

$$\|b_r - b_d\|_2 \le \|U_S b_d\|_2 + \|U_C b_d\|_2$$

$$\propto \frac{1}{S_{\min}} - \frac{1}{S_{\max}}$$

$$\boxed{\propto I - \frac{1}{S}\sum_{i=1}^{S} \left(C_i^E\right)^T C_i^E}$$

not distributedly computable

Distributed Monte Carlo Estimation

(assumption: b_d and $|r_{\mathrm{ave}}|$ already computed)

$$\left\| b_c - b_d \right\|_2 \leq |r_{\mathrm{ave}}| + \alpha \left\| U_S b_d \right\|_2 + \alpha \left\| U_C b_d \right\|_2$$

Distributed Monte Carlo Estimation

(assumption: b_d and $|r_{\mathrm{ave}}|$ already computed)

$$\left\| b_c - b_d \right\|_2 \leq |r_{\mathrm{ave}}| + \alpha \left\| U_S b_d \right\|_2 + \alpha \left\| U_C b_d \right\|_2$$

① (locally) generate a sample of $\left\| U_C b_d \right\|_2$

Distributed Monte Carlo Estimation

(assumption: b_d and $|r_{\mathrm{ave}}|$ already computed)

$$\|b_c - b_d\|_2 \leq |r_{\mathrm{ave}}| + \alpha \|U_S b_d\|_2 + \alpha \|U_C b_d\|_2$$

① (locally) generate a sample of $\|U_C b_d\|_2$

② (distributedly) estimate $\mathbb{E}\left[\|U_C b_d\|_2\right]$, $\mathrm{var}\left(\|U_C b_d\|_2\right)$

Distributed Monte Carlo Estimation

(assumption: b_d and $|r_{\mathrm{ave}}|$ already computed)

$$\left\| b_c - b_d \right\|_2 \leq |r_{\mathrm{ave}}| + \alpha \left\| U_S b_d \right\|_2 + \alpha \left\| U_C b_d \right\|_2$$

① (locally) generate a sample of $\left\| U_C b_d \right\|_2$

② (distributedly) estimate $\mathbb{E}\left[\left\| U_C b_d \right\|_2 \right]$, $\mathrm{var}\left(\left\| U_C b_d \right\|_2 \right)$

③ use Cantelli's inequality:

$$\mathbb{P}\left[\star \leq \mathbb{E}\left[\star\right] + k \cdot \mathrm{stdev}\left(\star\right) \right] \geq \frac{1}{1 + k^2}$$

Distributed Monte Carlo Estimation

(assumption: b_d and $|r_{\mathrm{ave}}|$ already computed)

$$\|b_c - b_d\|_2 \leq |r_{\mathrm{ave}}| + \alpha \|U_S b_d\|_2 + \alpha \|U_C b_d\|_2$$

① (locally) generate a sample of $\|U_C b_d\|_2$

② (distributedly) estimate $\mathbb{E}\left[\|U_C b_d\|_2\right]$, $\mathrm{var}\left(\|U_C b_d\|_2\right)$

③ use Cantelli's inequality:

$$\mathbb{P}\left[\star \leq \mathbb{E}\left[\star\right] + k \cdot \mathrm{stdev}\left(\star\right)\right] \geq \frac{1}{1+k^2}$$

obtain:

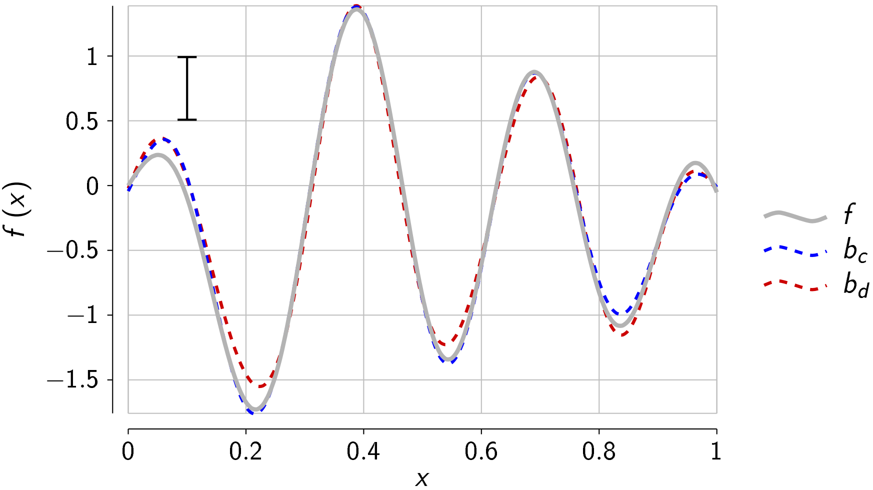
$$\boxed{\|b_c - b_d\|_2 \leq |r_{\mathrm{ave}}| + \alpha \|U_S b_d\|_2 + \alpha \left(\widehat{\mathbb{E}}\left[\|U_C b_d\|_2\right] + k \cdot \widehat{\mathrm{stdev}}\left(\|U_C b_d\|_2\right)\right)}$$

Description of the simulations

- $f : \mathbb{R} \mapsto \mathbb{R}$
- $\mu = \mathcal{U}[0,1]$
- composition of f: sum of sinusoidals
- K: Gaussian kernel
- measurement noise st. dev.: 0.25 (average SNR: ≈ 10)

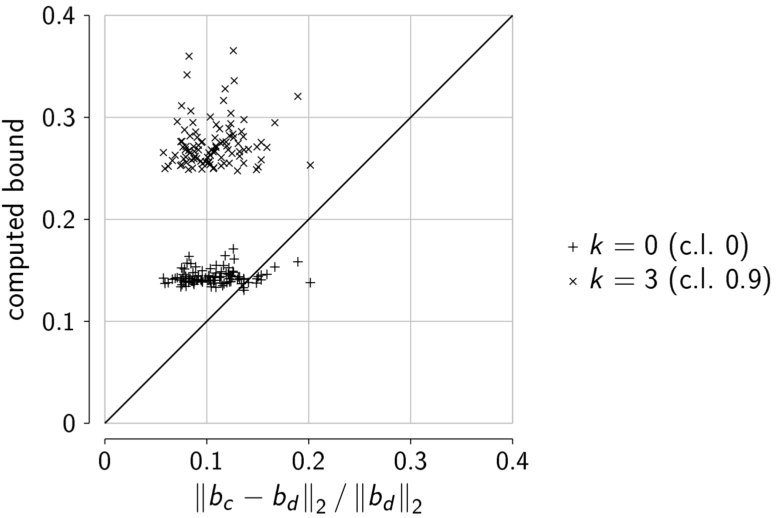
Regression strategy effectiveness example

$S = 1000$, $E = 40$ (motivated later)



Examples of the effectiveness of the bounds

$S = 1000$, $E = 40$, $S_{\min} = 900$, $S_{\min} = 1100$, 100 Monte Carlo runs



Swept under the carpet

i.e. (some of) the things we should have proved

- how to compute the bound
- why estimates of $\mathbb{E}\left[\left\|U_C b_d\right\|_2\right]$ and $\mathrm{stdev}\left(\left\|U_C b_d\right\|_2\right)$ are good estimates
- why we can use Cantelli's inequality

The need for automatic tuning procedures

$$b_d\left(S_g, E\right) = \left(\frac{1}{S_g}\mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + I\right)^{-1}\left(\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T y_i\right)$$

The need for automatic tuning procedures

$$b_d\left(S_g, E\right) = \left(\frac{1}{S_g}\mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + I\right)^{-1}\left(\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T y_i\right)$$

to be distributedly computed

The need for automatic tuning procedures

can be considered fixed

$$b_d\left(S_g, E\right) = \left(\frac{1}{S_g}\mathrm{diag}\left(\frac{\gamma}{\lambda_e}\right) + I\right)^{-1}\left(\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T y_i\right)$$

to be distributedly computed

Tuning under centralized scenarios

General aim

maximize the generalization capabilities of the estimators

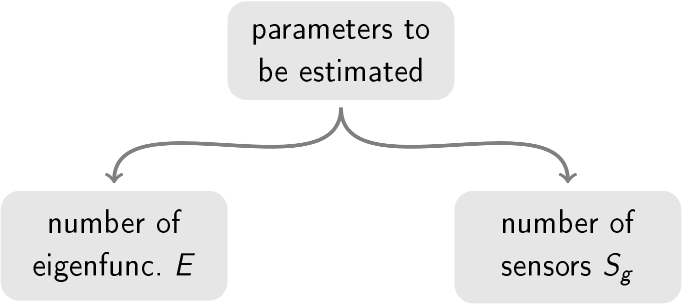
Classical approaches

- cross validation
- maximum (marginal) likelihood
- ...

Peculiarities of our distributed framework

- no test sets available
- involves unknown quantities

Proposed solution - key ideas

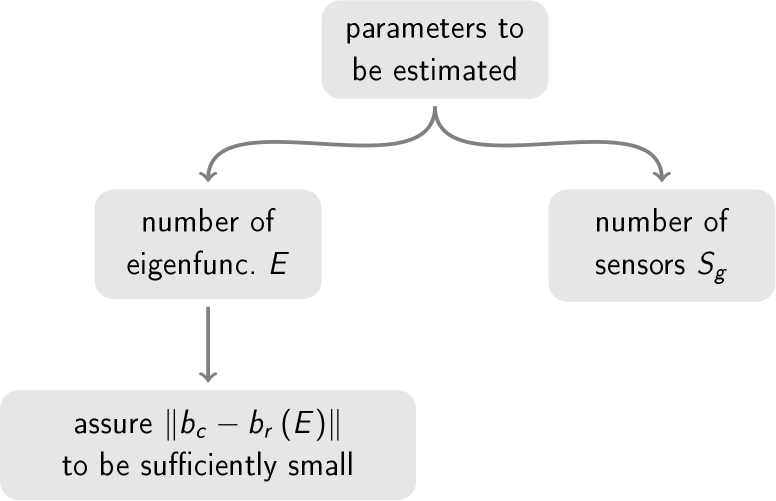


parameters to be estimated

number of eigenfunc. E

number of sensors S_g

Proposed solution - key ideas



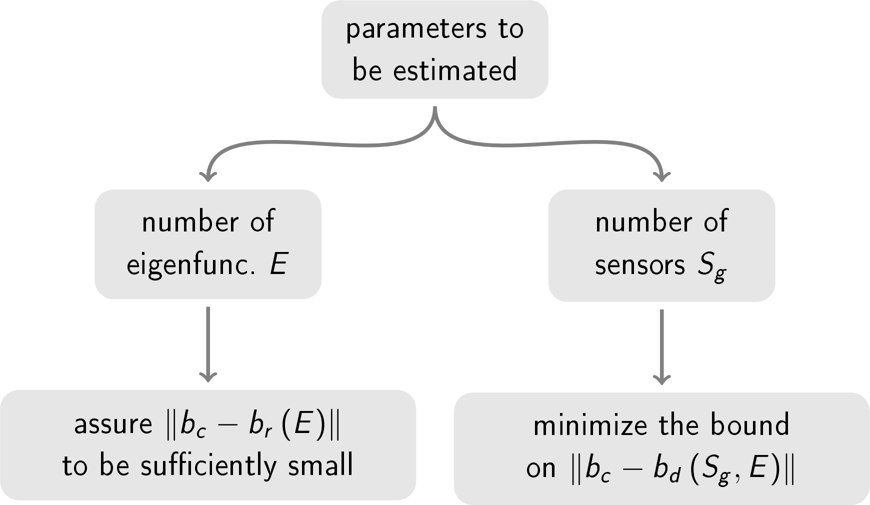
parameters to be estimated

number of eigenfunc. E

number of sensors S_g

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

Proposed solution - key ideas



parameters to be estimated

number of eigenfunc. E

number of sensors S_g

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

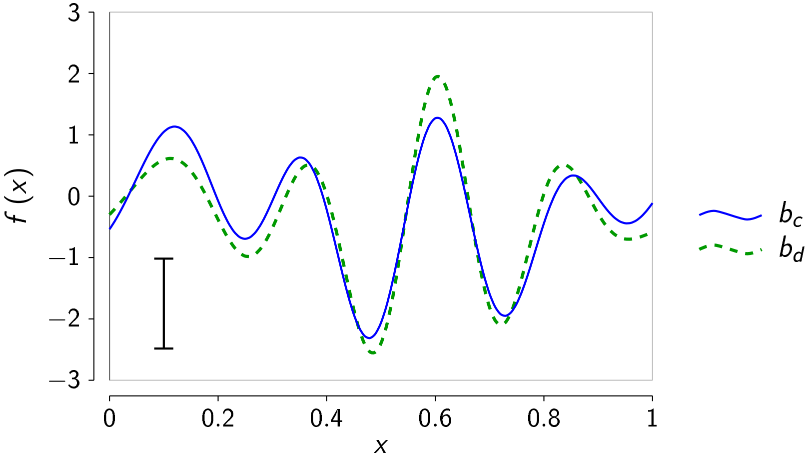
minimize the bound on $\|b_c - b_d(S_g, E)\|$

Description of the simulations

- $f \ : \ \mathbb{R} \mapsto \mathbb{R}$
- $\mu = \mathcal{U}\left[0, 1\right]$
- composition of f: sum of sinusoidals
- K: Gaussian kernel
- measurement noise st. dev.: 0.75 (implies an average SNR of ≈ 2.5)

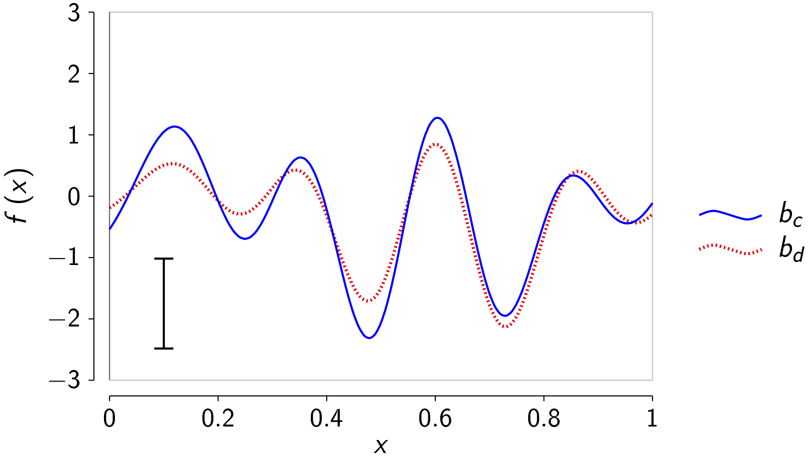
Regression strategy effectiveness example

$S = 100,\ E = 20,\ S_{\min} = 90,\ S_{\max} = 110$



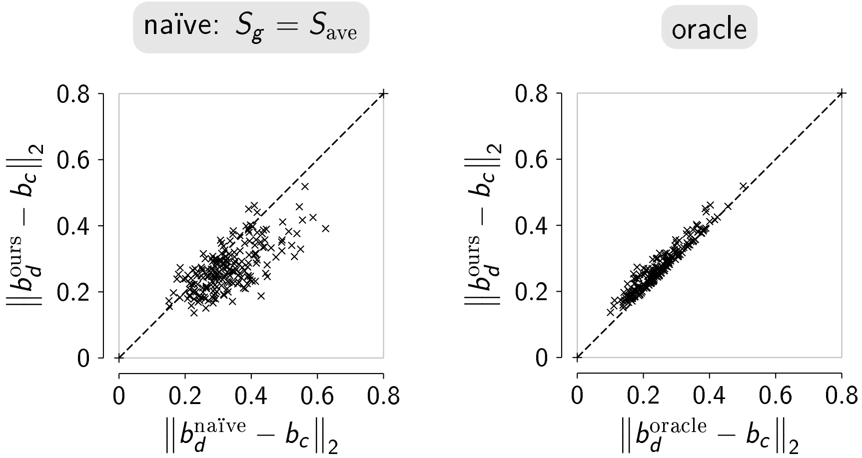
Regression strategy effectiveness example

$S = 100,\ E = 20,\ S_{\min} = 20,\ S_{\max} = 2000$



Comparisons with naïve and oracle strategies

$S = 100, \ E = 20, \ S_{\min} = 20, \ S_{\max} = 2000$



naïve: $S_g = S_{\text{ave}}$

oracle

Conclusions and future works

Conclusions

Strategy is:

- effective and easy to be implemented strategy
- tight performances bound
- self-tuning capabilities

Future works

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

Distributed nonparametric regression in multi-agent systems

Damiano Varagnolo
(joint work with Gianluigi Pillonetto and Luca Schenato)

Department of Information Engineering - University of Padova

March, 25th 2011

damiano.varagnolo@dei.unipd.it
www.dei.unipd.it/∼varagnolo/
google: damiano varagnolo

entirely written in LaTeX2ε using Beamer and TikZ

Barbarossa, S. and Scutari, G. (2007). Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems. *IEEE Transactions on Signal Processing*, 55(7):3456 – 3470.

Bazerque, J., Mateos, G., and Giannakis, G. (2010). Distributed lasso for in-network linear regression. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. Technical report, Stanford University.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26:622–633.

Choi, J., Oh, S., and Horowitz, R. (2009). Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802 – 2814.

Cortés, J. (2009). Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816 – 2827.

Cressie, N. and Wikle, C. K. (2002). Space - time Kalman filter. *Encyclopedia of Environmetrics*, 4:2045 – 2049.

Dall'Anese, E., Kim, S.-J., and Giannakis, G. B. (2011). Channel gain map tracking via distributed kriging. *IEEE Transactions on Vehicular Technology*, 2011.

De Nicolao, G. and Ferrari-Trecate, G. (1999). Consistent identification of NARX models via Regularization Networks. *IEEE Transactions on Automatic Control*, 44(11):2045 – 2049.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2005). Nonparametric decentralized detection using kernel methods. *IEEE Transactions on Signal Processing*, 53(11):4053 – 4066.

Olfati-Saber, R. (2007). Distributed kalman filtering for sensor networks. In *IEEE Conference on Decision and Control*.

Predd, J. B., Kulkarni, S. R., and Poor, H. V. (2006). Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56 – 69.

Schizas, I. D., Ribeiro, A., and Giannakis, G. B. (2008). Consensus in ad hoc WSNs with noisy links - part I: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350 – 364.

Varagnolo, D., Pillonetto, G., and Schenato, L. (2011a). Auto-tuning procedures for distributed nonparametric regression algorithms. *IEEE Conference on Decision and Control*, Submitted.

Varagnolo, D., Pillonetto, G., and Schenato, L. (2011b). Distributed parametric and nonparametric regression with on-line performance bounds computation. *Automatica*, Submitted.