Distributed Parametric-Nonparametric Estimation in Networked Control Systems

Damiano Varagnolo

Department of Information Engineering - University of Padova

April, 18th 2011
Research topics

- Multi-agent systems
- Smart grids
Research topics

- distributed optimization
- application of consensus
- parametric regression
- nonparametric regression
- estimation of # of agents
- multi-agent systems
- smart grids

Damiano Varagnolo (DEI) varagnolo@dei.unipd.it April, 18th, 2011
Research topics

- Distributed optimization
- Parametric regression
- Nonparametric regression
- Application of consensus
- Multi-agent systems
- Smart grids
- Estimation of # of agents
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
First problem considered in this speech

Assumption

noisy measurements of

\[ f(x, t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R} \]

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

\[ x \in \mathbb{R}^2 : \text{position} \]

\[ t : \text{time} \]

\[ f(x, t) : \text{channel gain} \]

source: Dall’Anese et al., 2011
Example 2 - waves power extraction

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

$t$: time

$f(x, t)$: sea level

source: www.graysharboroceanenergy.com
Example 3 - multi robot exploration

\[ x \in \mathbb{R}^2 : \text{position} \]
\[ f(x) : \text{ground level} \]

source: http://wwwrobotics.jpl.nasa.gov
## Difficulties related to this problem

### Information-related difficulties
- non-uniform samplings both in time and in space
- unknown dynamics of $f$
- unknown or extremely complex correlations in time and space

### Hardware-related difficulties
- energy & computational & memory & bandwidth limitations

### Framework-related difficulties
- mobile and time varying network
State of the art

<table>
<thead>
<tr>
<th>proposed distributed solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Likelihoods</td>
</tr>
<tr>
<td>Kriging</td>
</tr>
<tr>
<td>Other Learning Techniques</td>
</tr>
</tbody>
</table>
State of the art

**dynamic scenarios**

- **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

**proposed distributed solutions**
State of the art

**static scenarios**

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

**proposed distributed solutions**

- 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

\[ \hat{f}(x, t) = \Psi \text{ (past measurements)} \]

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. \( \hat{f} \to f \) when \( \# \) measurements \( \to \infty \) (De Nicolao, Ferrari-Trecate, 1999)
State of the art - where we actually contributed

agents estimate the same $f$

e.g. exploration

our small puzzle-piece

static scenario ($f$ independent of $t$)

no needs to discard old measurements

Regularization Network approach

Poggio and Girosi 1990
Our goal

obtain a *simple, self-evaluating* and *auto-tuning* multi-agent regression strategy
Framework

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

\[ y_m = f (x_m) + \nu_m \] (1)

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

examples of \( \mu \): uniform, jitter, generic
Considered cost function

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

Centralized optimal solution as a Regularization Network

\[ f_c = \sum_{m=1}^{M} c_m K(x_m, \cdot) \]

\[
\begin{bmatrix}
    c_1 \\
    \vdots \\
    c_M
\end{bmatrix}
= \left( \begin{bmatrix}
    K(x_1, x_1) & \cdots & K(x_1, x_M) \\
    \vdots & \ddots & \vdots \\
    K(x_M, x_1) & \cdots & K(x_M, x_M)
\end{bmatrix} + \gamma I \right)^{-1}
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_M
\end{bmatrix}
\]
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

Centralized optimal solution as a Regularization Network

\[
f_c = \sum_{m=1}^{M} c_m K(x_m, \cdot)
\]

\[
\begin{bmatrix}
  c_1 \\
  \vdots \\
  c_M
\end{bmatrix} = \left( \begin{bmatrix}
  K(x_1, x_1) & \cdots & K(x_1, x_M) \\
  \vdots & \ddots & \vdots \\
  K(x_M, x_1) & \cdots & K(x_M, x_M)
\end{bmatrix} + \gamma I \right)^{-1} \begin{bmatrix}
  y_1 \\
  \vdots \\
  y_M
\end{bmatrix}
\]
Drawbacks

\[ f_c = \sum_{m=1}^{M} c_m K(x_m, \cdot) \]

\[
\begin{bmatrix}
    c_1 \\
    \vdots \\
    c_M
\end{bmatrix} = 
\left( 
\begin{bmatrix}
    K(x_1, x_1) & \cdots & K(x_1, x_M) \\
    \vdots & \ddots & \vdots \\
    K(x_M, x_1) & \cdots & K(x_M, x_M)
\end{bmatrix} + \gamma I \right)^{-1} 
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_M
\end{bmatrix}
\]

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

\( \downarrow \)

\textit{need to find alternative solutions}
Alternative centralized optimal solution (1\textsuperscript{st} on 2)

Structure of $K$ implies

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

$\Rightarrow$ measurement model can be rewritten as

$$y_m = \begin{bmatrix} \phi_1(x_m) \\ \phi_2(x_m) \\ \vdots \end{bmatrix} b + \nu_m$$

(2)
Alternative centralized optimal solution (2\textsuperscript{nd} on 2)

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

involves infinite dimensional objects:

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

\[\Rightarrow \text{cannot be computed exactly}\]
Suboptimal finite dimensional solution

New estimator

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

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

New estimator

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

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

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

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

Consider the approximations

- \( M \rightarrow M_g \) (guess)
- \( \frac{1}{M} \sum_{m=1}^{M} (C_m^E)^T C_m^E \rightarrow \mathbb{E}_\mu \left[ (C_m^E)^T C_m^E \right] = I \)
Derivation of the distributed estimator

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

Advantages

1. \( O(E) \) computational effort
2. \( O(E) \) transmission effort
Summary of proposed estimation schemes

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

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

\( b_d: O(E) \) comput., \( O(E) \) transm.
Summary of proposed estimation schemes

\[ b_c: O(M^3) \text{ comput.}, O(M) \text{ transm.} \]
\[ b_r: O(E^3) \text{ comput.}, O(E^2) \text{ transm.} \]
\[ b_d: O(E) \text{ comput.}, O(E) \text{ transm.} \]
Summary of proposed estimation schemes

- \( b_c: O(M^3) \) comput., \( O(M) \) transm.
- \( b_r: O(E^3) \) comput., \( O(E^2) \) transm.
- \( b_d: O(E) \) comput., \( O(E) \) transm.

Reduced hyp. space
\[ b_d(E, M_g) \]

Original hyp. space
Quantification of performances

Assumption: $E, M_g$ already chosen, $b_d$ already computed

$$\|b_c - b_d\|_2 \leq \frac{1}{M} \sum_{m=1}^{M} |r_m| + \|U_M b_d\|_2 + \|U_C b_d\|_2$$
Quantification of performances
Assumption: \( E, M_g \) already chosen, \( b_d \) already computed

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

local residuals
Quantification of performances

Assumption: $E, M_g$ already chosen, $b_d$ already computed

\[
\|b_c - b_d\|_2 \leq \frac{1}{M} \sum_{m=1}^{M} |r_m| + \|U_M b_d\|_2 + \|U_C b_d\|_2
\]
Quantification of performances

Assumption: $E, M_g$ already chosen, $b_d$ already computed

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

Local residuals

\[
\propto \frac{1}{M_{\text{min}}} - \frac{1}{M_{\text{max}}}
\]

\[
\propto I - \frac{1}{M} \sum_{m=1}^{M} (C_m^E)^T C_m^E
\]
Quantification of performances

Assumption: $E, M_g$ already chosen, $b_d$ already computed

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

local residuals

computable through distributed MC

\[ \propto \frac{1}{M_{\min}} - \frac{1}{M_{\max}} \]

\[ \propto I - \frac{1}{M} \sum_{m=1}^{M} (C_m^E)^T C_m^E \]
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 $E$
    - assure $\| b_c - b_r (E) \|$ to be sufficiently small
  - number of measurements $M_g$
    - minimize the bound on $\| b_c - b_d (E, M_g) \|$
Regression strategy effectiveness example

\( M = 100, \ E = 20, \ M_{\text{min}} = 90, \ M_{\text{max}} = 110, \ \text{SNR} \approx 2.5 \)
Regression strategy effectiveness example

\[ M = 100, \ E = 20, \ M_{\text{min}} = 90, \ M_{\text{max}} = 110, \ SNR \approx 2.5 \]
Accuracy of the computed bound

\[ M = 100, \ E = 20, \ M_{\text{min}} = 90, \ M_{\text{max}} = 110 \]
Comparison with oracle

$M = 100$, $E = 20$, $M_{\text{min}} = 90$, $M_{\text{max}} = 110$
Conclusions and future works for this part

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)
Part Two
Privacy-aware number of agents estimation

Estimation of the number of agents ($A$) can be important in:

- distributed estimation
- analysis of connectivity

We assume *privacy concerns* $\rightarrow$ *do not use IDs!*

Our goal: obtain an easily implementable distributed estimator satisfying the constraints
The basic idea

Algorithm:

\[ \hat{A} - 1 = y \]
The basic idea

Algorithm:

\[ y_1 \sim \mathcal{N}(0, 1) \]
\[ y_2 \sim \mathcal{N}(0, 1) \]
\[ y_3 \sim \mathcal{N}(0, 1) \]
\[ y_4 \sim \mathcal{N}(0, 1) \]
\[ y_5 \sim \mathcal{N}(0, 1) \]
The basic idea

Algorithm:

\[ y_1 \rightarrow \frac{1}{A} \sum_{a=1}^{A} y_a \]
\[ y_2 \rightarrow \frac{1}{A} \sum_{a=1}^{A} y_a \]
\[ y_3 \rightarrow \frac{1}{A} \sum_{a=1}^{A} y_a \]
\[ y_4 \rightarrow \frac{1}{A} \sum_{a=1}^{A} y_a \]
\[ y_5 \rightarrow \frac{1}{A} \sum_{a=1}^{A} y_a \]
The basic idea

Algorithm:

1. **Local generation**
2. **Average consensus**
3. **Maximum Likelihood**

\[ \hat{A}^{-1} = y_{\text{ave}}^2 \]

\[ y_{\text{ave}} \sim \mathcal{N} \left(0, \frac{1}{A} \right) \]
A simple example

The basic idea

Algorithm:

1. local generation
2. average consensus
3. Maximum Likelihood

\[
y_{\text{ave}} \sim \mathcal{N} \left(0, \frac{1}{A} \right)
\]

\[
y_{\text{ave}} \sim \mathcal{N} \left(0, \frac{1}{A} \right)
\]

\[
y_{\text{ave}} \sim \mathcal{N} \left(0, \frac{1}{A} \right)
\]

\[
y_{\text{ave}} \sim \mathcal{N} \left(0, \frac{1}{A} \right)
\]

\[
\hat{A}^{-1} = y_{\text{ave}}^2
\]

does not require sending IDs
Reformulating the idea as a block scheme

\[ \mathcal{N} \left( 0, 1 \right) \rightarrow y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y_A \rightarrow \text{aver. cons.} \rightarrow \text{ML} \rightarrow \hat{A} \]
A simple example

Plausible ways to generalize the idea

\[ p(\cdot) \xrightarrow{local} y_1^1, \ldots, y_1^r \xrightarrow{distributed} F \xrightarrow{local} \Psi \xrightarrow{average} \hat{A} \]

- \( \mathcal{N}(\mu, \sigma^2) \)
- \( \mathcal{U}[\alpha, \beta] \)
- ??
- average
- max
- ??
- ML
- MMSE
- MAP
- ??

Damiano Varagnolo (DEI)
varagnolo@dei.unipd.it
April, 18th 2011 35 / 48
Theoretical results

Which cost function we consider

notice: we want to estimate $A^{-1}$ instead of $A$

\[ \hat{A}^{-1} \]

estimator $=: \hat{A}^{-1}$

Considered cost function

\[
\mathbb{E} \left[ \left( \hat{A}^{-1} - A^{-1} \right)^2 \right]
\]

(\equiv \text{variance if } \hat{A}^{-1} \text{ unbiased})

Why?

- convenient in order to obtain mathematical results
- in our cases, asymptotically in $r$:

\[
\lim_{r \to +\infty} \mathbb{E} \left[ \left( \frac{\hat{A}^{-1} - A^{-1}}{A^{-1}} \right)^2 \right] = \mathbb{E} \left[ \left( \frac{\hat{A} - A}{A} \right)^2 \right]
\]
Theoretical results: average-consensus + ML

Assumptions

- $y_a$ generated through Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$
- fusion of $y_a$ is through average-consensus
Theoretical results: average-consensus + ML

Assumptions
- $y_a$ generated through Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$
- fusion of $y_a$ is through average-consensus

Results: ML estimators:
- writable in closed form
Theoretical results: average-consensus + ML

Assumptions
- $y_a$ generated through Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$
- fusion of $y_a$ is through average-consensus

Results: ML estimators:
- writable in closed form
- are MVUE (Minimum Variance and Unbiased)
Theoretical results: average-consensus + ML

Assumptions

- $y_a$ generated through Gaussian distributions $\mathcal{N} (\mu, \sigma^2)$
- fusion of $y_a$ is through average-consensus

Results: ML estimators:

- writable in closed form
- are MVUE (Minimum Variance and Unbiased)
- performances: $\text{var} \left( \frac{\hat{A}^{-1} - A^{-1}}{A^{-1}} \right) = \frac{2}{r}$ (independent of $\mu$ and $\sigma^2$)
Theoretical results: average-consensus + ML

Assumptions
- $y_a$ generated through Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$
- fusion of $y_a$ is through average-consensus

Results: ML estimators:
- writable in closed form
- are MVUE (Minimum Variance and Unbiased)
- performances: $\text{var} \left( \frac{\hat{A}^{-1} - A^{-1}}{A^{-1}} \right) = \frac{2}{r}$ (independent of $\mu$ and $\sigma^2$)
- (conjecture: Law of Large Numbers) if $r \to +\infty$ then performances are independent of $p(\cdot)$
Theoretical results: max-consensus + ML

Assumptions

- cumulative distribution $P(\cdot)$ of $y_a$ is strictly monotonic and continuous
- fusion of $y_a$ is through max-consensus
Theoretical results: max-consensus + ML

Assumptions
- cumulative distribution $P(\cdot)$ of $y_a$ is strictly monotonic and continuous
- fusion of $y_a$ is through max-consensus

Results: ML estimators:
- writable in closed form
Theoretical results: max-consensus + ML

\[ p(\cdot) \rightarrow y_1, y_2, \ldots, y_s \rightarrow \text{max cons.} \rightarrow \text{ML} \rightarrow \hat{s} \]

Assumptions
- Cumulative distribution \( P(\cdot) \) of \( y_a \) is strictly monotonic and continuous
- Fusion of \( y_a \) is through max-consensus

Results: ML estimators:
- Writable in closed form
- Are MVUE (Minimum Variance and Unbiased)
Theoretical results: max-consensus + ML

Assumptions

- cumulative distribution $P(\cdot)$ of $y_\cdot$ is strictly monotonic and continuous
- fusion of $y_\cdot$ is through max-consensus

Results: ML estimators:

- writable in closed form
- are MVUE (Minimum Variance and Unbiased)
- performances: $\text{var} \left( \frac{\hat{A}^{-1} - A^{-1}}{A^{-1}} \right) = \frac{1}{r}$, independent of $P(\cdot)$
Theoretical results: max-consensus + ML

\[ p(\cdot) \xrightarrow{\cdot} y_1 \xrightarrow{\cdot} y_2 \xrightarrow{\cdot} \ldots \xrightarrow{\cdot} y_S \xrightarrow{\cdot} \text{max cons.} \xrightarrow{\cdot} \text{ML} \xrightarrow{\cdot} \hat{s} \]

Assumptions
- cumulative distribution \( P(\cdot) \) of \( y_a \) is strictly monotonic and continuous
- fusion of \( y_a \) is through max-consensus

Results: ML estimators:
- writable in closed form
- are MVUE (Minimum Variance and Unbiased)
- performances: \( \var\left( \frac{A^{-1} - A^{-1}}{A^{-1}} \right) = \frac{1}{r} \) independent of \( P(\cdot) \)
- performances are twice as good as average-consensus
Results of various simulated systems (1)

<table>
<thead>
<tr>
<th>$\mathcal{N}(0,1)$</th>
<th>average consensus</th>
<th>ML</th>
<th>$A = 10$</th>
</tr>
</thead>
</table>

Simulative results

Density plots for $\hat{A}$ with $A = 10$, $r = 10$, $r = 40$, $r = 70$, and $r = 100$. The plots show the distribution of $\hat{A}$ for different values of $r$. The title of the slide is "Simulative results".
Results of various simulated systems (2)

<table>
<thead>
<tr>
<th>$\mathcal{U} [0, 1]$</th>
<th>max consensus</th>
<th>ML</th>
<th>$A = 10$</th>
</tr>
</thead>
</table>

$p(\hat{A}|A)$

$r = 10$

$r = 40$

$r = 70$

$r = 100$
Simulative results

Results of various simulated systems (3)

<table>
<thead>
<tr>
<th>$\mathcal{U} [0, 1]$</th>
<th>max consensus</th>
<th>ML</th>
<th>$A = 100$</th>
</tr>
</thead>
</table>

$$p(\hat{A} | A)$$

- $r = 10$
- $r = 40$
- $r = 70$
- $r = 100$
Conclusions

... and future extensions

Conclusions
- effective and robust algorithm
- quantifiable performances
- rely on statistical concepts → preserves privacy
- inherits good qualities of consensus strategies

Future extensions
- analyze optimal quantization strategies
- find optimal distributions for average consensus
- use the strategy for topological change detection purposes
Distributed Parametric-Nonparametric Estimation in Networked Control Systems

Damiano Varagnolo

Department of Information Engineering - University of Padova

April, 18th 2011

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

Damiano Varagnolo (DEI)

licensed under the Creative Commons BY-NC-SA 2.5 Italy License:
Distributed Optimization


Applications of Consensus


## Parametric Regression


Nonparametric Regression / Classification


**Number of Sensors Estimation**


**Smart Grids**

<table>
<thead>
<tr>
<th>Key/Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>left mouse / page down / space</td>
<td>next slide</td>
</tr>
<tr>
<td>right mouse / page up / backspace</td>
<td>previous slide</td>
</tr>
<tr>
<td>q / esc</td>
<td>exit</td>
</tr>
<tr>
<td>tab</td>
<td>overview toggling</td>
</tr>
<tr>
<td>t</td>
<td>clock toggling</td>
</tr>
<tr>
<td>home</td>
<td>go to first slide</td>
</tr>
<tr>
<td>end</td>
<td>go to last slide</td>
</tr>
<tr>
<td>l</td>
<td>go to the previously last seen slide</td>
</tr>
<tr>
<td>b</td>
<td>fade the screen to black</td>
</tr>
<tr>
<td>w</td>
<td>fade the screen to white</td>
</tr>
<tr>
<td>f</td>
<td>full screen toggling</td>
</tr>
<tr>
<td>enter</td>
<td>spotlight toggling</td>
</tr>
<tr>
<td>+ / -</td>
<td>adjust the spotlight size</td>
</tr>
<tr>
<td>mouse wheel</td>
<td>adjust the spotlight size</td>
</tr>
<tr>
<td>left mouse (dragging a box)</td>
<td>highlight a box</td>
</tr>
<tr>
<td>right mouse (on a highlighted box)</td>
<td>remove the highlight of that box</td>
</tr>
<tr>
<td>z</td>
<td>zoom toggling</td>
</tr>
<tr>
<td>right mouse (in zoom modality)</td>
<td>move on the image</td>
</tr>
</tbody>
</table>