Multi-agent map-building:

Kalman Filtering meets Machine Learning

Luca Schenato
University of Padova
Stuttgart, April 2017

M.Ag.I.C.
Multi Agent Intelligent Control

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Joint work with

Colleagues at Univ. of Padova

Ruggero Carli

Gianluigi Pillonetto

Current Ph.D/post-docs:

Marco Todescato

Nicoletta Bof

Former Ph.D/post-docs:

Damiano Varagnolo
Lulea Univ., Sweden

Guido Cavraro
Virginia Tech, USA

Andrea Carron
ETH, Switzerland

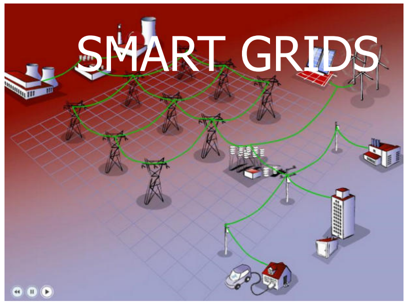
Outline

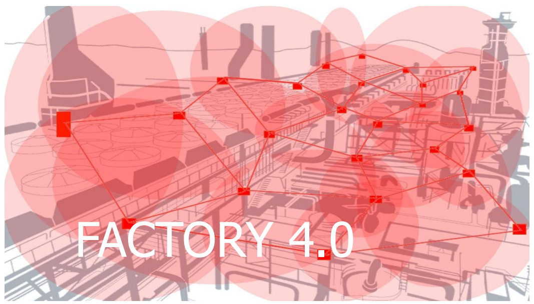
- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

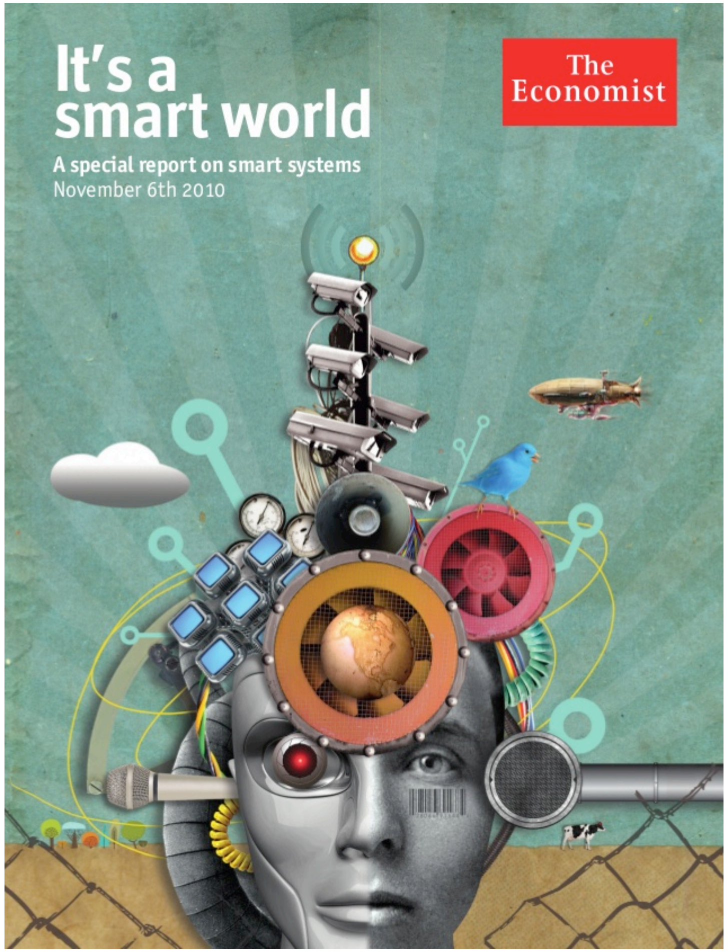
Outline

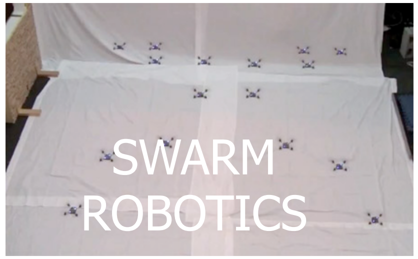
- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

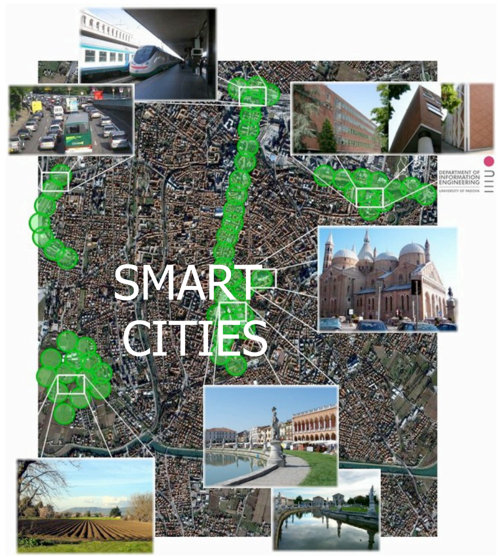
The XXI century: a Smart World





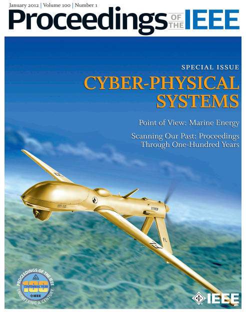


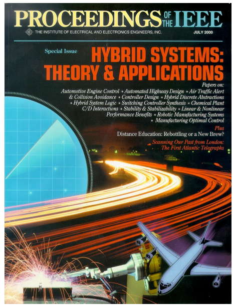


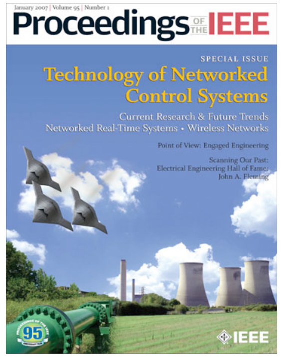


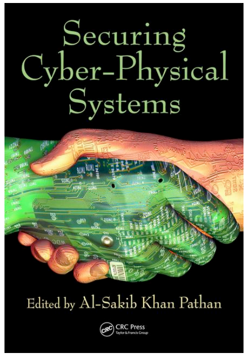
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

The ICT scientific army

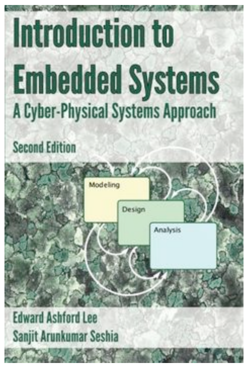


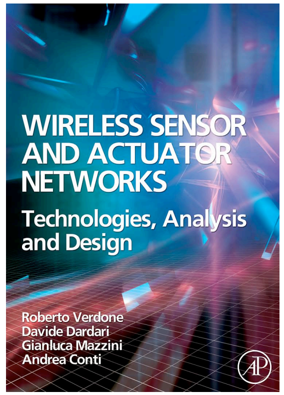


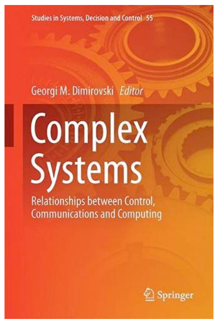


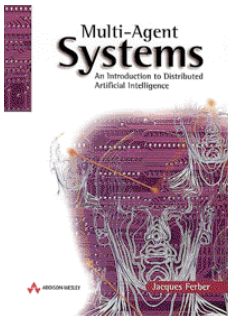


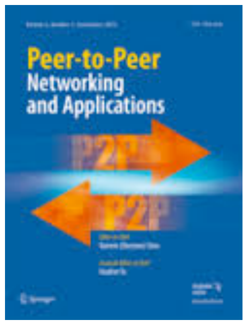
Before 2000's	Today		Tomorrow
Centralized Reliable comm.	Multi-agent Reliable comm.	Centralized Unrelia. Comm.	Multi-agent Unrelia. Comm.
✓	✓	✓	?

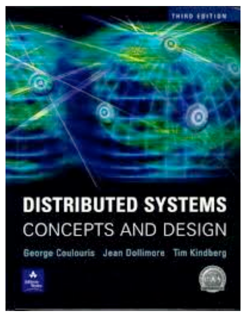












Target applications: MAgIC Lab. at University of Padova

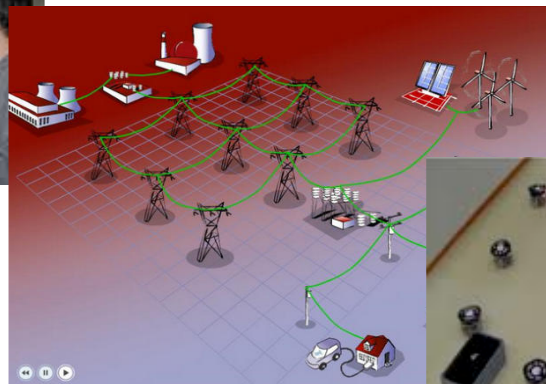
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

M.A.g.I.C.
Multi Agent Intelligent Control

Wireless Sensor Actuator Networks

Smart Camera Networks

Smart Energy Grids



Robotic Networks

2002

2008

2012

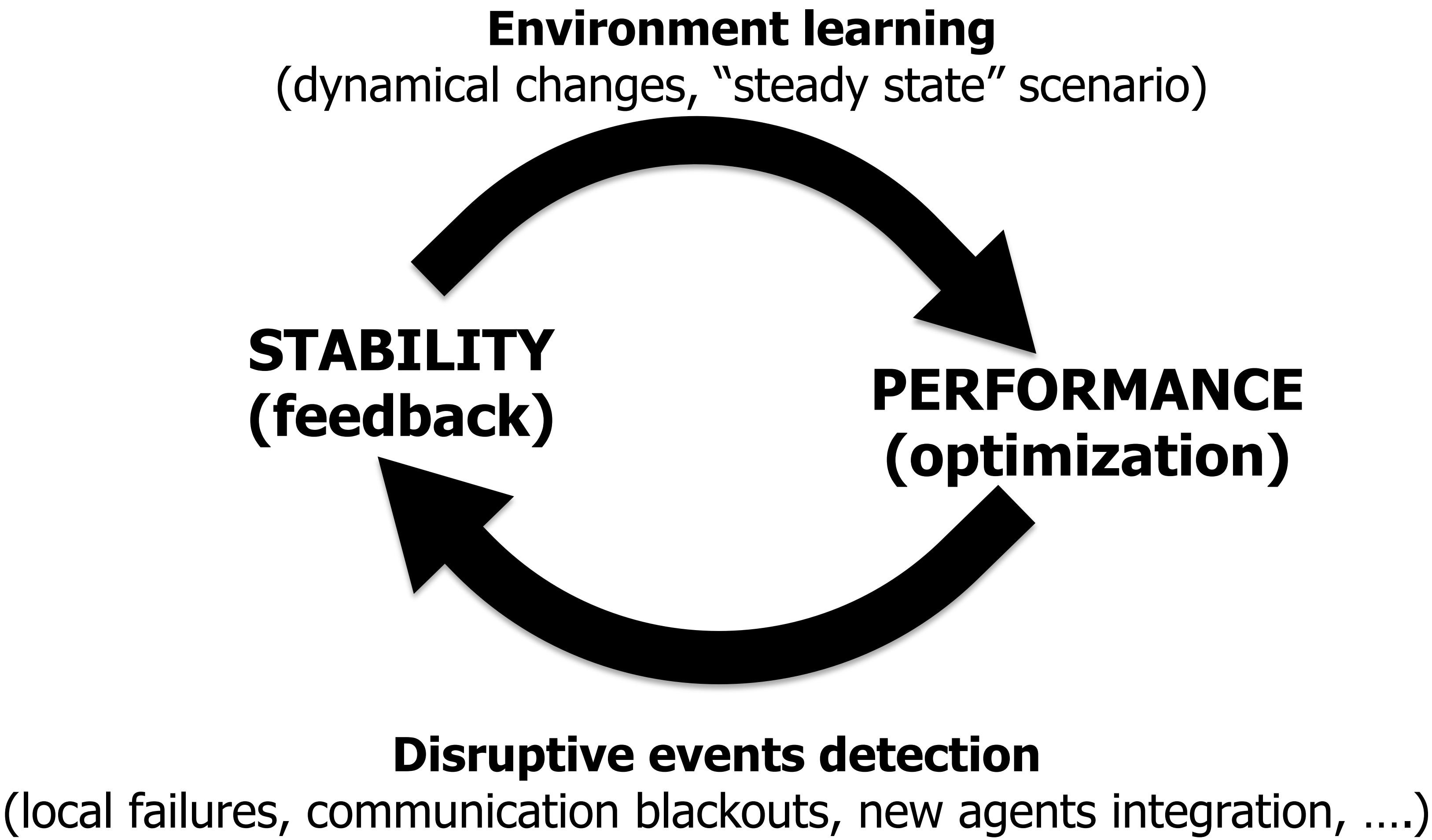
Challenges

- Unreliable (wireless) communication:
  - Random delay, packet loss, limited communication range
- Scalability:
  - Complexity (CPU, memory, communication) per agent must be constant
- Robustness/resilience and adaptiveness/learning:
  - Mild performance degradation when local failures
  - Continuous environmental learning
- Architecture:
  - Centralized vs hierarchical vs distributed vs decentralized
  - Cooperative vs competitive

Challenges

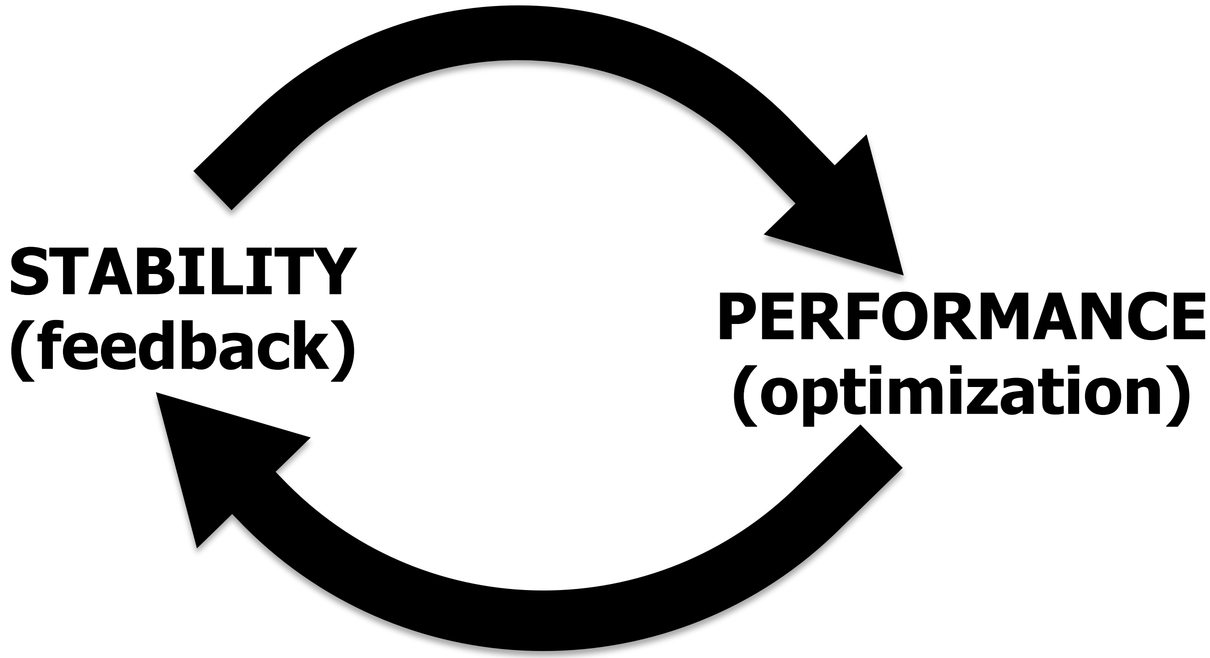
- Unreliable (wireless) communication:
  - Random delay, packet loss, limited communication range
- Scalability:
  - Complexity (CPU, memory, communication) per agent must be constant
- Robustness/resilience and adaptiveness/learning:
  - Mild performance degradation when local failures
  - Continuous environmental learning
- Architecture:
  - Centralized vs hierarchical vs distributed vs decentralized
  - Cooperative vs competitive

Dynamic learning and optimization



Dynamic learning and optimization

Environment learning
(dynamical changes, “steady state” scenario)

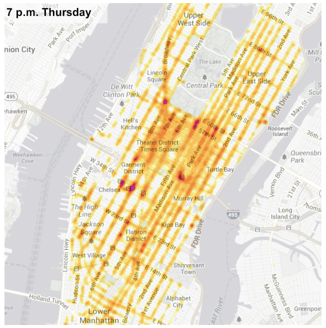


Disruptive events detection
(local failures, communication blackouts, new agents integration, ….)

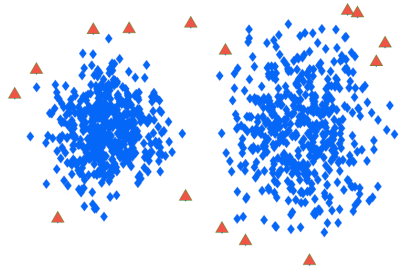
Learning problems:

Density estimation:

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$
$$f(x) \geq 0, \quad \int f(x) = 1$$
$$\mathcal{D} = \{x_1, x_2, ...\}: \text{ events}$$



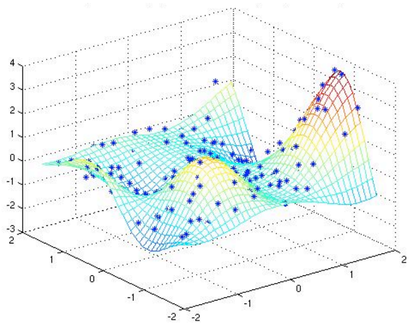
Taxi pick-up calls



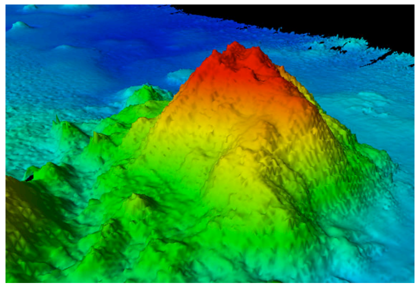
Anomaly detection

Regression:

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$
$$y_i = f(x_i) + v_i$$
$$v_i \text{ noise}$$



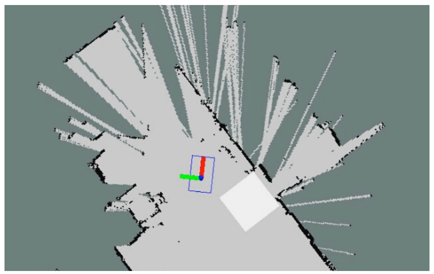
Pollution level profile



Seabed depth profile

Classification

$$f(x) : \mathbb{R}^n \to \{0, 1\}$$
$$y_i = f(x_i) + v_i$$
$$v_i \text{ noise}$$



Obstacles map

Oil-spill boundary

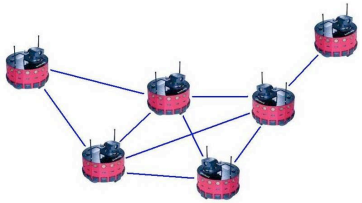
Multi-agent regression

- Parametric vs non-parametric

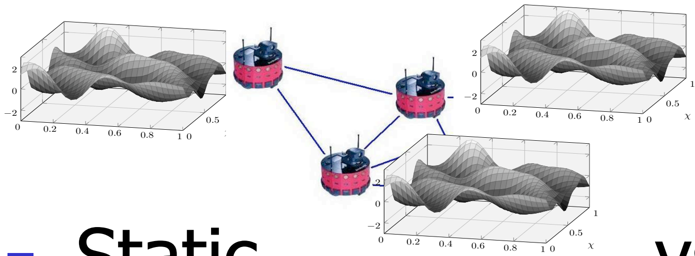
  $f(x) = \sum_{i=1}^{m} \theta_i g_i(x)$
  $\theta \in \mathbb{R}^m$, unknown
  $g_i(x)$ known

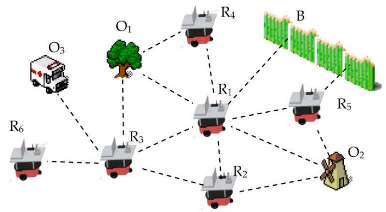
  $f(x) \in RKHS$, infinite dimensional
  $f(x)$ defined via Kernel $k(x,x')$
  $k(x,x')$ known

- Cloud-based vs peer-to-peer

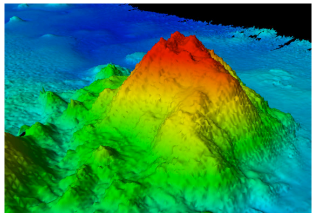


- Global vs Local estimation

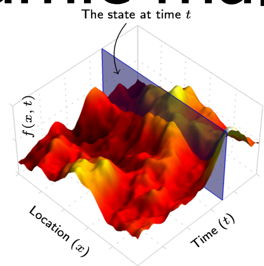




- Static vs dynamic maps:



$f(x)$



$f(x,t)$

Multi-agent regression

- Parametric vs non-parametric

$f(x) = \sum_{i=1}^{m} \theta_i g_i(x)$

$\theta \in \mathbb{R}^m$, unknown

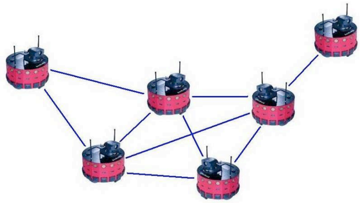
$g_i(x)$ known

$f(x) \in RKHS$, infinite dimensional

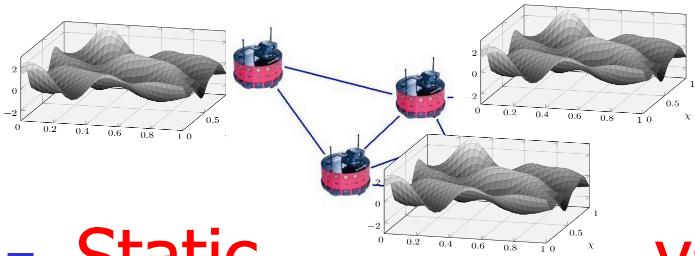
$f(x)$ defined via Kernel $k(x, x')$

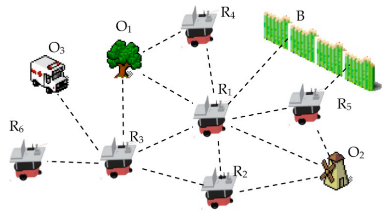
$k(x, x')$ known

- Cloud-based vs peer-to-peer

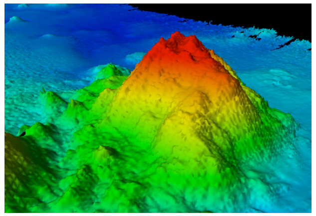


- Global vs Local estimation

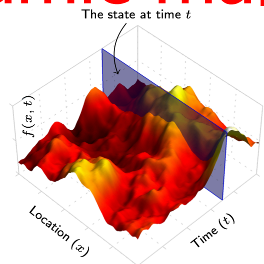




- Static vs dynamic maps:



$f(x)$

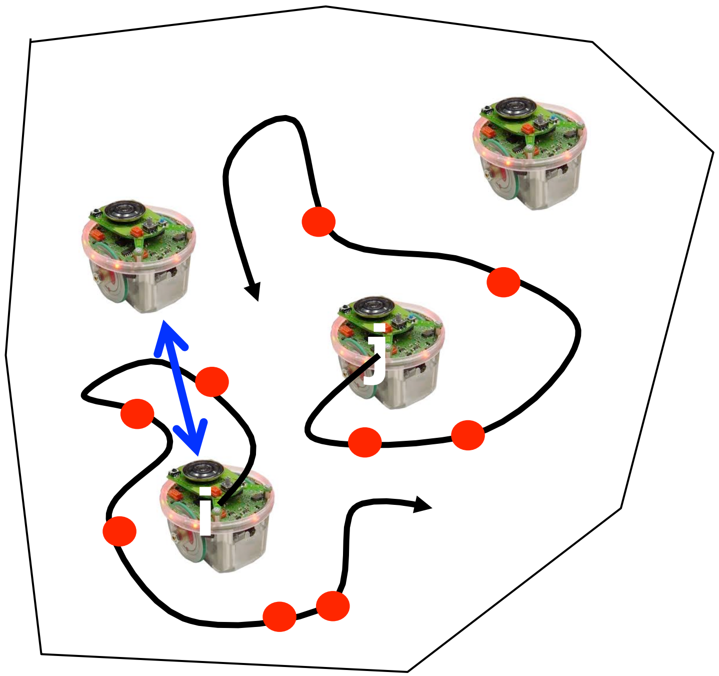


$f(x, t)$

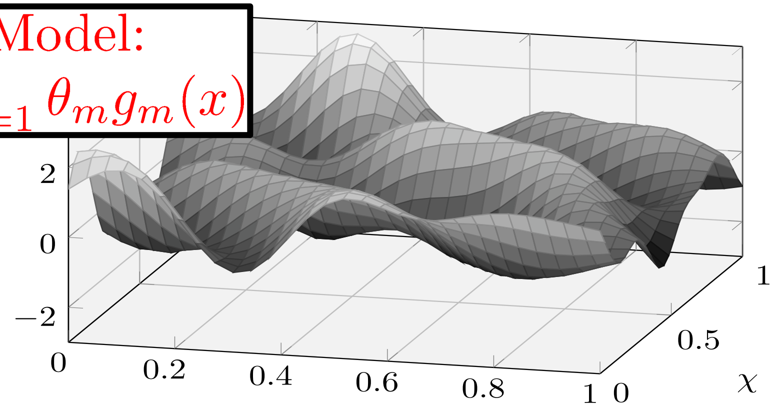
Outline

- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

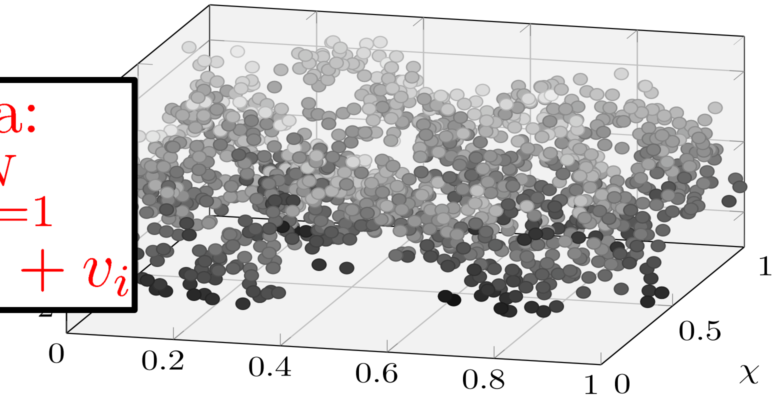
Example: Map-building in robotic networks



Parametric Model:
$f(x) = \sum_{m=1}^{M} \theta_m g_m(x)$



Noisy data:
$\{(x_i, y_i)\}_{i=1}^{N}$
$y_i = f(x_i) + v_i$



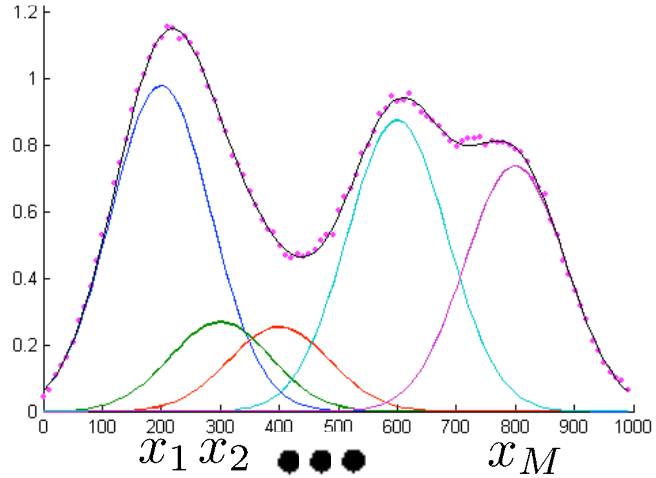
Goal:
$\min_\theta \sum_i v_i^2$

Parametric model: linear vs non-linear

Linear combination of radial basis functions

$$f(x) = \sum_{i=1}^{M} \theta_i g_i(x)$$

$$g_i(x) = e^{\frac{|x - x_i|^2}{2\sigma^2}}$$

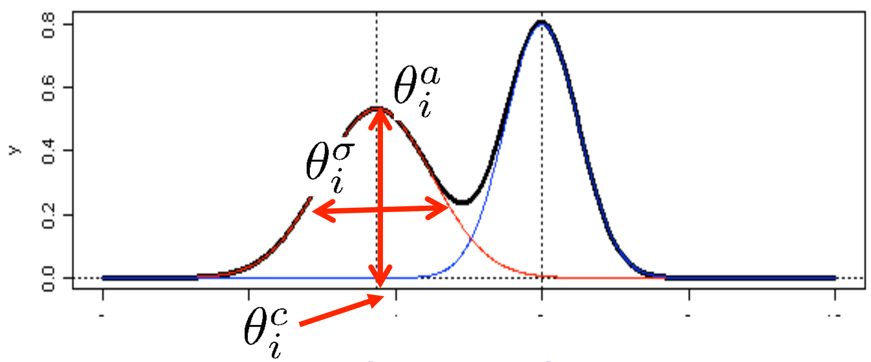


- Large number of basis
- Convex problem

Mixture of Gaussians

$$f(x) = \sum_{i=1}^{M} g_i(x, \theta_i)$$

$$g_i(x, \theta_i) = \theta_i^a e^{\frac{|x - \theta_i^c|^2}{2\theta_i^{\sigma\,2}}}$$

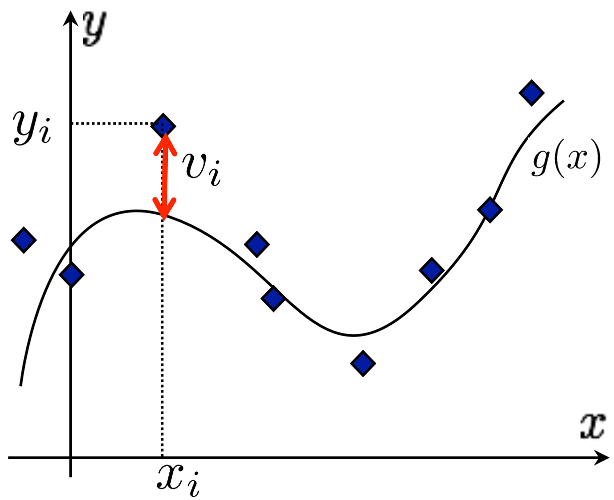


- Needs fewer functions
- Non-linear problem

Map-building as least-squares regression

- Model class:

$$f(x) = \sum_{m=1}^{M} \theta_m g_m(x)$$



- Noisy measurements:

$$y_i = \sum_{m=1}^{M} \theta_m g_m(x_i) + v_i, \quad i = 1, \ldots, N$$

G_i^T

$$\begin{bmatrix} y(x_1) \\ y(x_2) \\ \vdots \end{bmatrix} = \begin{bmatrix} g_1(x_1) & \ldots & g_M(x_1) \\ \vdots & \vdots & \vdots \\ g_1(x_N) & \ldots & g_M(x_N) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}$$

$$y = G\theta + v$$

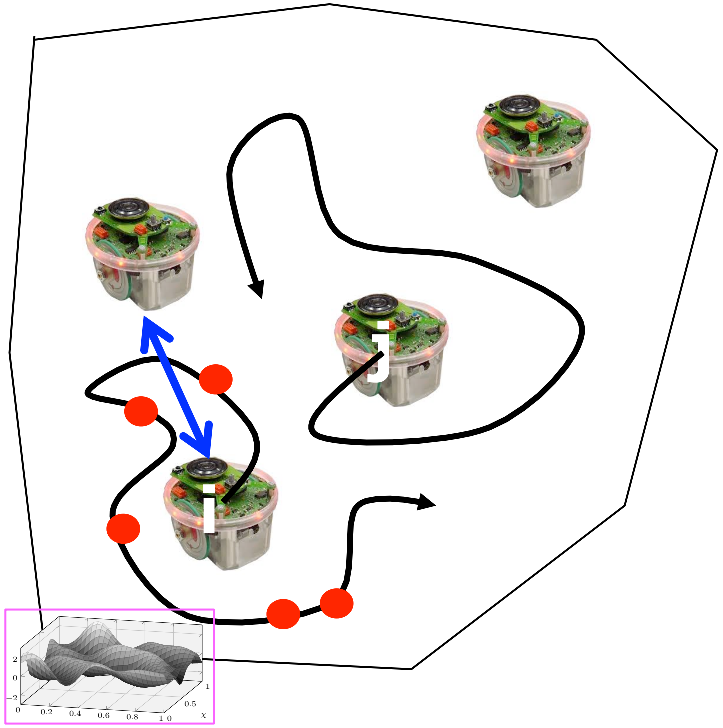
- Goal: minimize sum of squares of residues

$$\hat{\theta} \;=\; \text{argmin}_\theta \sum_{i=1}^{N} v_i^2$$

$$\boxed{\begin{aligned} \hat{\theta} &= \left(\textstyle\sum_{i=1}^{N} G_i G_i^T\right)^{-1}\left(\textstyle\sum_{i=1}^{N} G_i y_i\right) \\ &= \left(\tfrac{1}{N}\textstyle\sum_{i=1}^{N} G_i G_i^T\right)^{-1}\left(\tfrac{1}{N}\textstyle\sum_{i=1}^{N} G_i y_i\right) \end{aligned}}$$

- Xiao-Boyd-Lall, 2005
- Bolognani-Del Favero-Schenato-Varagnolo, 2010

Consensus-based Map-building: gossip communication



ALGORITHM:

1) Initialize statistics:

$$Z_0^i = 0 \in R^{M\times M}$$
$$z_0^i = 0 \in R^M$$

2) Collect data and build local statistics:

$$Z_{t+1}^i = Z_i^t + G_t^i {G_t^i}^T$$
$$z_{t+1}^i = z_i^t + G_t^i y_t^i$$

3) Choose neighbor j and do gossip consensus:

$$Z_{t+1}^j = Z_{t+1}^i = \tfrac{1}{2} Z_t^i + \tfrac{1}{2} Z_t^j$$
$$z_{t+1}^j = z_{t+1}^i = \tfrac{1}{2} z_i^t + \tfrac{1}{2} z_j^t$$

4) Estimate map:

$$\hat{\theta}_t^i = (Z_t^i)^{-1} z_t^i$$

5) Repeat steps 2,3,4 (non necessarely in order)

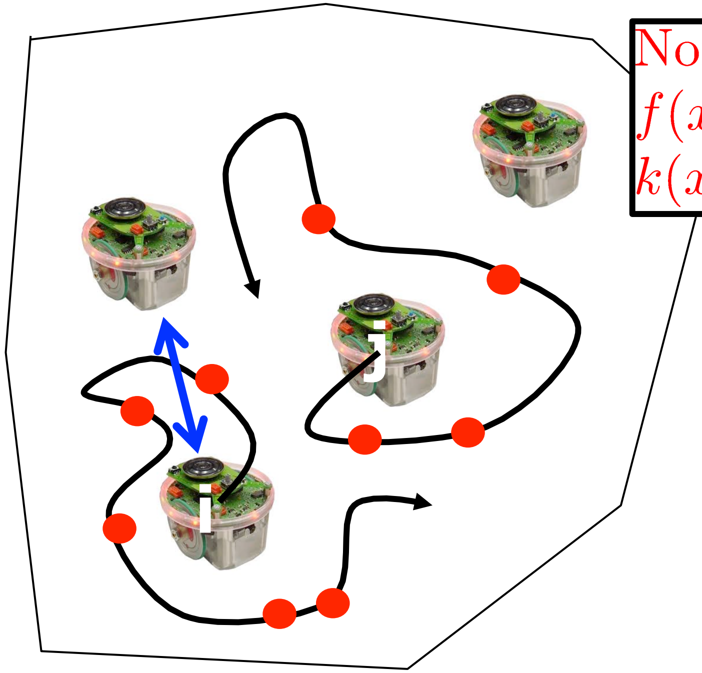
- PROS:
  - Can be distributed
  - Gradient-based implementation: ADMM, gradient-consensus,
  - Extension to robust costs, e.g. $|| \, ||_1$
- CONS:
  - How to select basis functions
  - No estimate unless at least M data
  - Gradient-based implementations require step-size design

Simulations: broadcast based map building

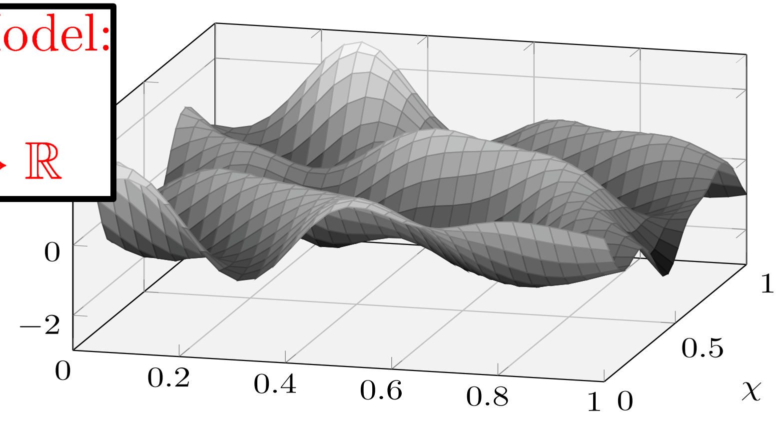
Outline

- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

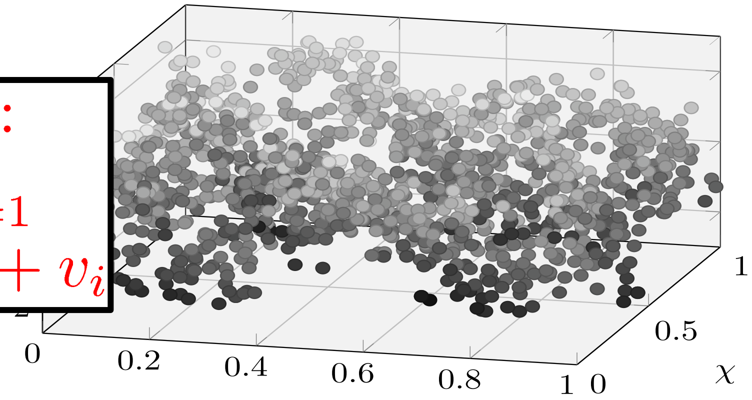
Gaussian regression (non parametric)



Non-parametric Model:
$f(x) \in \text{RKHS}$
$k(x, x') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$



Noisy data:
$\{(x_i, y_i)\}_{i=1}^{N}$
$y_i = f(x_i) + v_i$



Goal:
$\min_\theta \sum_i v_i^2$

Reproducing Kernel Hilbert Spaces (RKHS) (con't)

$k(x,x') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$: Mercel Kernel
1) $k(\cdot,\cdot)$ continous, $\mathcal{X}$: compact
2) symmetric: $k(x,x') = k(x',x)$
3) positive semidefinite: $K \in \mathbb{R}^{N\times N} \geq 0, \quad [K]_{i,j} = k(x_i,x_j), \forall x_i, \forall N,$

Bayesian Interpretation:

$\mathbb{E}[f(x)] = 0, \quad \mathbb{E}[f(x)f(x')] = k(x,x')$: zero-mean gaussian process

$\{(x_i,y_i)\}_{i=1}^{N}$: Noisy data:
$y_i = f(x_i) + v_i, \quad v_i \sim \mathcal{N}(0,\sigma^2)$

$\widehat{f}(x) = \mathbb{E}[f(x) \,|\, \{x_i,y_i\}_{i=1}^{M}] = \sum_{i=1}^{N} c_i k(x_i,x)$

$$\begin{bmatrix} c_1 \\ \vdots \\ c_M \end{bmatrix} = \left(K + \sigma^2 I\right)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}, \quad K := \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_M) \\ \vdots & & \vdots \\ k(x_M,x_1) & \cdots & k(x_M,x_M) \end{bmatrix}$$

Parametric vs non-parametric

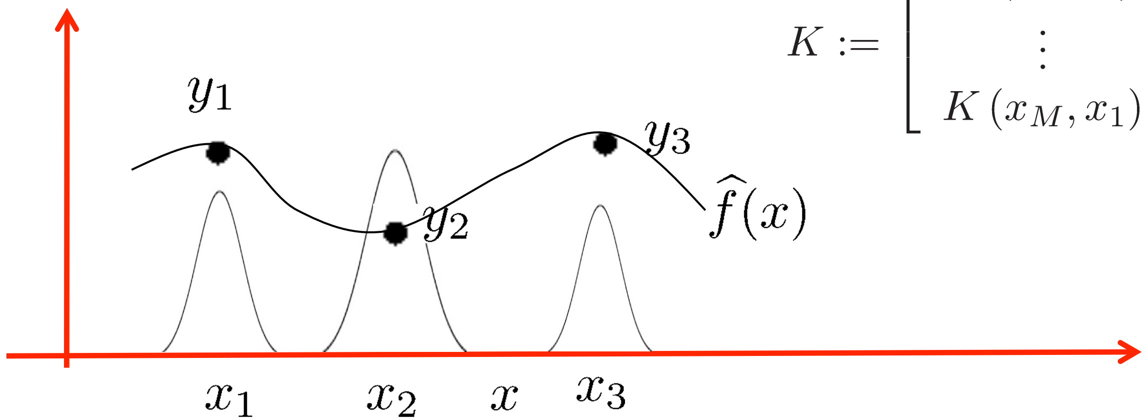
$$\{(x_i, y_i)\}_{i=1}^{N}: \text{ Noisy data:}$$
$$y_i = f(x_i) + v_i, \quad v_i \sim \mathcal{N}(0, \sigma^2)$$

$$f(x) = \sum_{i=1}^{M} \theta_i g_i(x)$$

$$g_i(x) = e^{\frac{|x - x_i|^2}{2\sigma^2}}$$

$$k(x, x') = e^{\frac{|x - x'|^2}{2\sigma^2}}$$

$$K := \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_M) \\ \vdots & & \vdots \\ K(x_M, x_1) & \cdots & K(x_M, x_M) \end{bmatrix}$$



$$\widehat{f}(x) = \sum_{i=1}^{M} \widehat{\theta}_i g_i(x)$$

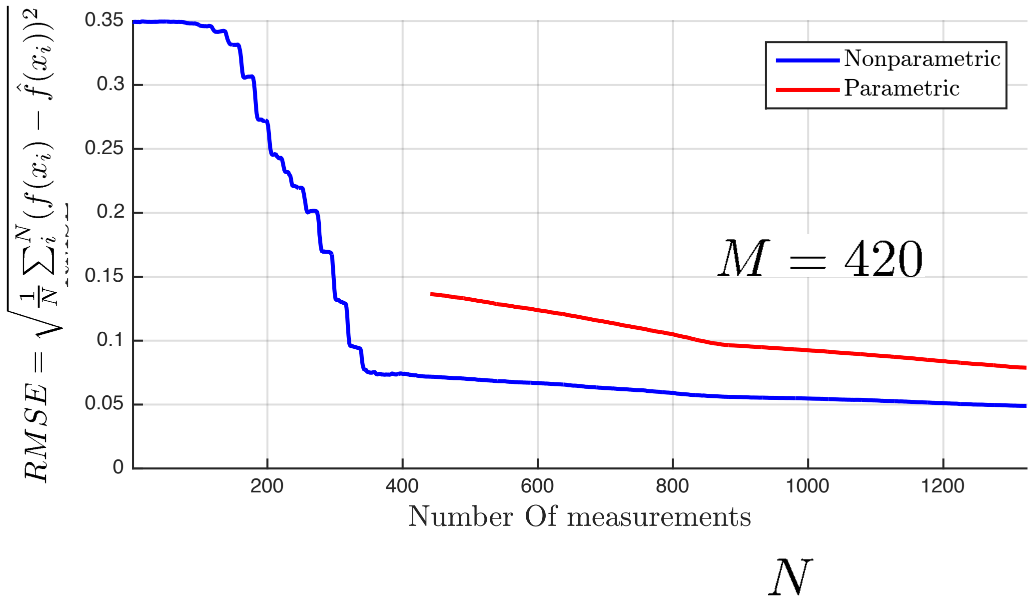
$$\widehat{f}(x) = \sum_{i=1}^{M} \hat{c}_i k(x_i, x) = \sum_{i=1}^{M} \hat{c}_i g_i(x)$$

$$\widehat{\theta} = K^{-1} y$$

$$\widehat{c} = (K + \sigma^2 I)^{-1} y$$

regularization term

Parametric vs non-parametric



	PARAMETRIC	NON-PARAMETRIC
PROS	• Distributed (consensus) • Bounded complexity $O(M^3)$	• Better performance • Adaptable resolution
CONS	• What $g_i(x)$? • Need N>M points • Over-fitting & ill-conditioned	• Regularization factor design • Data-limited complexity $O(N^3)$

Outline

- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

Representer theorem

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

$k(x,x'):\mathcal{X}\times\mathcal{X}\to\mathbb{R}$: Mercel Kernel
1) $k(\cdot,\cdot)$ continous, $\mathcal{X}$: compact
2) symmetric: $k(x,x')=k(x',x)$
3) positive semidefinite: $K\in\mathbb{R}^{N\times N}\geq 0,\quad [K]_{i,j}=k(x_i,x_j),\forall x_i,\forall N,$

$\mu:\mathcal{X}\to\mathbb{R}^+$: measure function (sampling density)

$h(x):=T_{k,\mu}[g](x):=\int_{\mathcal{X}} g(x')k(x,x')d\mu(x')$: Hilbert-Schmidt integral operator
$h(x),g(x)\in\mathcal{L}^2(\mu)$

Since T is a linear operator → eigenvalues and eigenfunctions

$$T_{k,\mu}[\phi(x)]=\lambda\,\phi(x),\lambda\geq 0$$

Representer Theorem: Let $k(\cdot,\cdot)$ be a Mercer kernel on $\mathcal{X}\times\mathcal{X}$, $\lambda_\ell>0\ \ \forall\ell$ and μ a non-degenerate measure. Then, $\{\phi_\ell\}_{\ell=1}^{+\infty}$ is an orthonormal basis in $\mathcal{L}^2(\mu)$ while the associated RKHS is

$$\mathcal{H}_K:=\left\{f(x)\in\mathcal{L}^2(\mu)\ \ \text{s.t.}\ \ f(x)=\sum_{\ell=1}^{\infty}\alpha_\ell\phi_\ell(x)\ \text{and}\ \sum_{e=1}^{\infty}\frac{\alpha_e^2}{\lambda_e}<+\infty\right\}$$

Map-building as least-squares regression

- Model class:

$$f(x) = \sum_{m=1}^{M} \theta_m g_m(x)$$

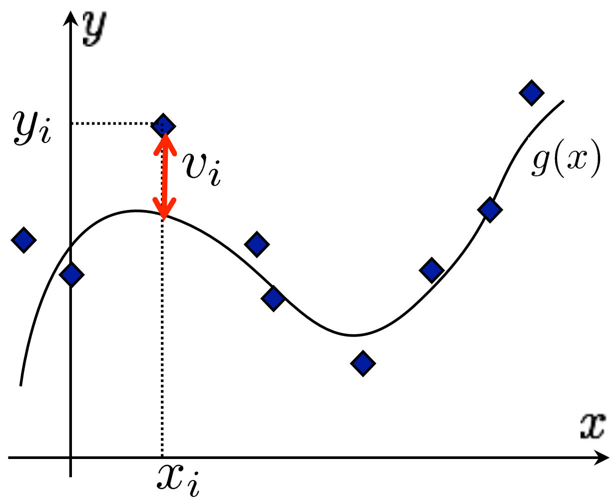
- Noisy measurements:

$$y_i = \sum_{m=1}^{M} \theta_m g_m(x_i) + v_i, \quad i = 1, \ldots, N$$

$$\begin{bmatrix} y(x_1) \\ y(x_2) \\ \vdots \end{bmatrix} = \begin{bmatrix} g_1(x_1) & \ldots & g_M(x_1) \\ \vdots & \vdots & \vdots \\ g_1(x_N) & \ldots & g_M(x_N) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}$$

G_i^T

$$y = G\theta + v$$



- Goal: minimize sum of squares of residues

$$\widehat{\theta} = \text{argmin}_\theta \sum_{i=1}^{N} v_i^2$$

$$\boxed{\begin{aligned} \widehat{\theta} &= (\textstyle\sum_{i=1}^{N} G_i G_i^T)^{-1}(\sum_{i=1}^{N} G_i y_i) \\ &= (\frac{1}{N}\textstyle\sum_{i=1}^{N} G_i G_i^T)^{-1}(\frac{1}{N}\sum_{i=1}^{N} G_i y_i) \end{aligned}}$$

- Xiao-Boyd-Lall, 2005
- Bolognani-Del Favero-Schenato-Varagnolo, 2010

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Semi-parametric estimation

1st IDEA: Use first eigenfunctions as basis function for parametric estimation

$$f(x) = \sum_{\ell=1}^{+\infty} \alpha_\ell \phi_\ell(x)$$

$$y_i = \sum_{\ell=1}^{+\infty} \alpha_\ell \phi_\ell(x_i) = \underbrace{[\phi_1(x_i) \ \ \phi_2(x_i) \ \ \ldots]}_{G_i^T} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \end{bmatrix} + v_i$$

$$\widehat{\alpha} = \left(\mathrm{diag}\left(\frac{\sigma^2}{\lambda_\ell}\right) + \sum_{i=1}^N G_i G_i^T\right)^{-1} \left(\sum_{i=1}^N G_i y_i\right)$$

$$\widehat{\alpha}^E = \left(\mathrm{diag}\left(\frac{\sigma^2}{\lambda_\ell}\right) + \sum_{i=1}^N G_i^E (G_i^E)^T\right)^{-1} \left(\sum_{i=1}^N G_i^E y_i\right)$$

$$G_i^E = [\phi_1(x_i) \cdots \phi_E(x_i)]$$

(intuition: $\alpha_i \approx 0$, for $i > E$, therefore $\widehat{f}(x) \approx \widehat{f}^E(x)$)

Semi-parametric estimation (cont'd)

2st IDEA: Use orthonormality of eigenfunctions ϕ_n and i.i.d. sampling of x_i

$$\widehat{\alpha}^E = \left(\mathrm{diag}\left(\frac{\sigma^2}{N\lambda_\ell}\right) + \frac{1}{N}\sum_{i=1}^N G_i^E (G_i^E)^T\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^N G_i^E y_i\right)$$

$$\left[\frac{1}{N}\sum_{i=1}^N G_i^E \left(G_i^E\right)^T\right]_{mn} = \frac{1}{N}\sum_{i=1}^N \phi_m\left(x_i\right)\phi_n\left(x_i\right)$$

$$\left[\frac{1}{N}\sum_{i=1}^N \phi_m(x_i)\phi_n(x_i)\right] \xrightarrow{N\to+\infty, x_i\sim\mu(x)} \int \phi_m(x)\phi_n(x)d\mu(x) = \delta_{mn}$$

$$\widehat{\alpha}^I(x) = \left(\mathrm{diag}\left(\frac{\sigma^2}{N\lambda_\ell}\right) + I\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^N G_i^E y_i\right)$$

Complexity of semi-parametric approaches

$$\widehat{f}(x) = \sum_{i=1}^{N} c_i k(x_i, x), \quad c = (K + I)^{-1} y, \quad [K]_{mn} = k(x_m, x_n)$$

$$\widehat{f}^E(x) = \sum_{i=1}^{E} \alpha_i^E \phi_i(x) \qquad \widehat{\alpha}^E = \left(\mathrm{diag}\left(\frac{\sigma^2}{N\lambda_\ell}\right) + \frac{1}{N}\sum_{i=1}^{N} G_i^E (G_i^E)^T\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} G_i^E y_i\right)$$

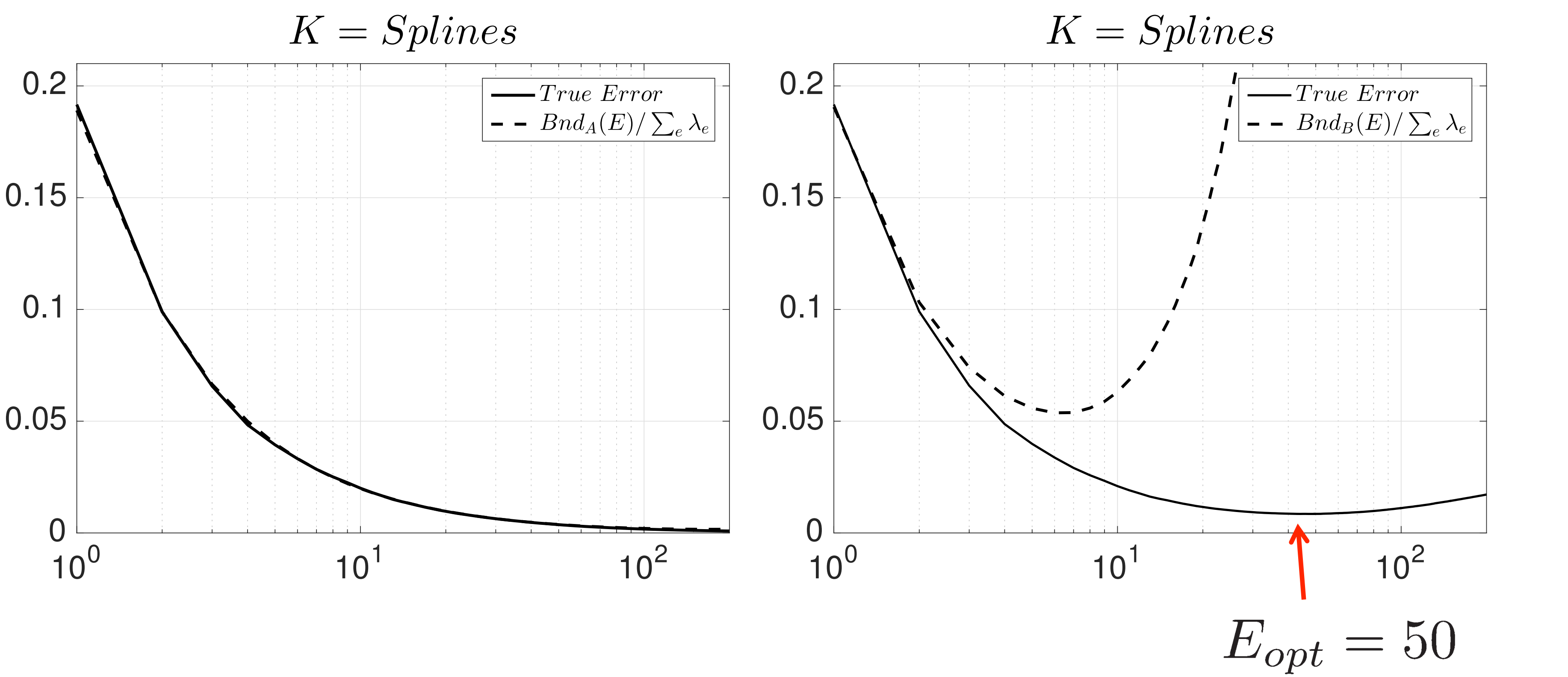
$$\widehat{f}^I(x) = \sum_{i=1}^{I} \alpha_i^E \phi_i(x) \qquad \widehat{\alpha}^I = \left(\mathrm{diag}\left(\frac{\sigma^2}{N\lambda_\ell}\right) + I\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} G_i^E y_i\right)$$

estimator	*comput. cost*	*commun. cost*	*memory cost*
$\widehat{f}(x)$	$O(N^3)$	$O(N)$	$O(N)$
$\widehat{f}^E(x)$	$O(E^3)$	$O(E^2)$	$O(E^2)$
$\widehat{f}^I(x)$	$O(E)$	$O(E)$	$O(E)$

Performance of semi-parametric approaches

$$N = 10000$$

$\widehat{f}^E(x)$ $\widehat{f}^I(x)$



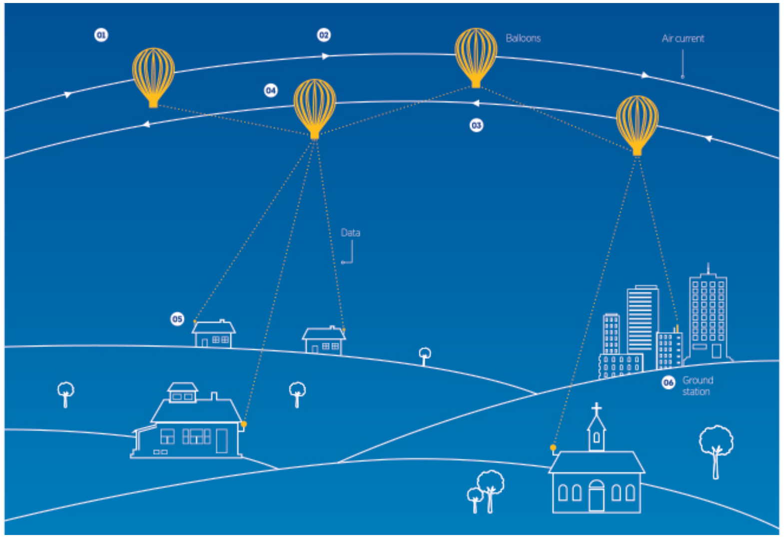
Outline

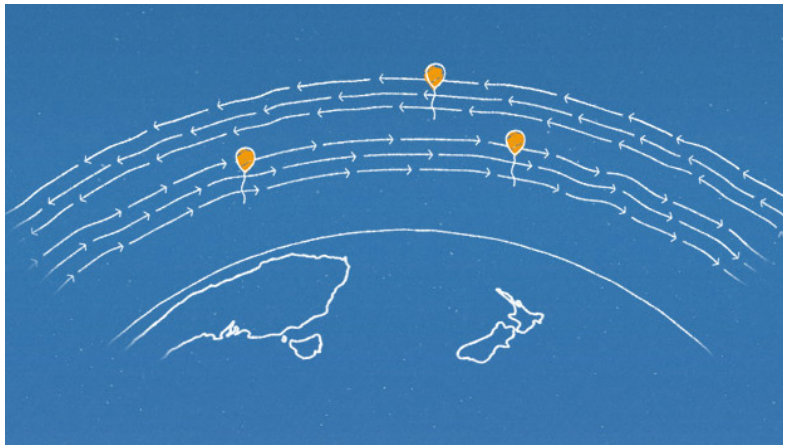
- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

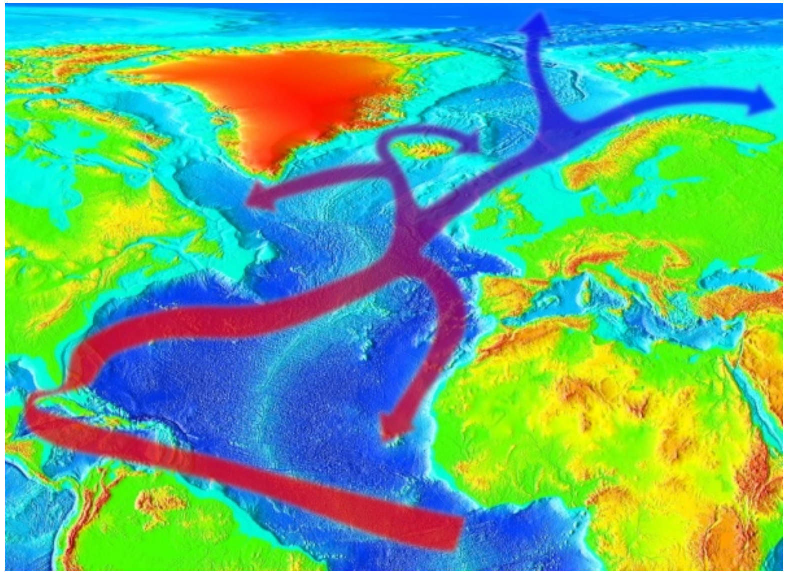
Non-parametric regression for dynamical systems

Project Loon[1]





Wind/Ocean Current 4 Energy/Air Vehicles



DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Time-varying regression f(x,t): parametric vs non-parametric

Single location: $f(\bar{x}, t) : \mathbb{R} \rightarrow \mathbb{R}$

$$\begin{aligned} s_{k+1} &= A s_k + w_k, \quad w_k \sim \mathcal{N}(0, Q) \\ y_k &= C s_k + v_k, \quad v_k \sim \mathcal{N}(0, R) \\ f_k &= C s_k, \qquad s_0 \sim \mathcal{N}(0, \Sigma_0) \end{aligned}$$

Data: $\{(t_k, y_k)\}_{i=1}^N, t_k = kT$

$$\begin{aligned} &\widehat{s}_{k+1|k} = A \widehat{s}_{k|k} \\ &\Sigma_{k+1|k} = A \Sigma_{k|k} A^T + Q \\ &\widehat{s}_{k+1|k+1} = \widehat{s}_{k+1|k} + L_{k+1} \left(y_{k+1} - C_k \widehat{s}_{k+1|k} \right) \\ &\Sigma_{k+1|k+1} = \Sigma_{k+1|k} - L_{k+1} C_k \Sigma_{k+1|k} \\ &L_{k+1} = \Sigma_{k+1|k} C_k^T \left(C_k \Sigma_{k+1|k} C_k^T + R \right)^{-1} \end{aligned}$$

$$\widehat{s}_{k|k} = \mathbb{E}[s_k \mid y_k, \ldots, y_0] \Rightarrow \widehat{f}(\bar{x}, kT) = C \widehat{s}_{k|k}$$

$$\widehat{s}_{k|k} = \mathbb{E}[s_k \mid y_k, \ldots, y_0] = \mathbb{E}[s_k \mid y_k, \widehat{s}_{k-1|k-1}]$$

Numerical efficient but requires to know the exact model (A,C,Q,R)

Single instant: $f(x, \bar{t}) : \mathbb{R}^p \rightarrow \mathbb{R}$

$$\begin{aligned} &y_i = f(x_i) + v_i, \quad v_i \sim \mathcal{N}(0, \sigma^2) \\ &f(x) \in \mathrm{RKHS} \leftrightarrow k(x, x') \end{aligned}$$

Data: $\{(x_i, y_i)\}_{i=1}^N$

$$\widehat{f}(x) = \sum_{i=1}^N c_i k(x_i, x)$$

$$c = (K + I)^{-1} y, \quad [K]_{mn} = k(x_m, x_n)$$

Model-free and best performance but high computational complexity $O(N^3)$

Time-varying regression f(x,t): parametric vs non-parametric

General case: $f(x,t) : \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}$

Stochastic PDEs:

space-time white Gaussian noise

$$\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{\partial^2 f(x,t)}{\partial t^2} - \lambda^2 f(x,t) = w(x,t),$$

Data: $\{(t_i, x_i, y_i)\}_{i=1}^N$

Discretize in time and space
and run finite-element methods

Approximated solutions and requires
a good physical model

Define $\xi = (x,t)$ and kernel $k(\xi, \xi')$:

$$y_i = f(\xi_i) + v_i, \quad v_i \sim \mathcal{N}(0, \sigma^2)$$
$$f(\xi) \in \mathrm{RKHS} \leftrightarrow k(\xi, \xi')$$

Data: $\{(\xi_i, y_i)\}_{i=1}^N$

$$\widehat{f}(\xi) = \sum_{i=1}^N c_i k(\xi_i, \xi)$$

$$c = (K + I)^{-1} y, \quad [K]_{mn} = k(\xi_m, \xi_n)$$

Time space treated equally and
unbounded complexity O(t^3)

S. Sarkka, A. Solin , J Hartikainen, Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. IEEE Sig. Proc. Mag., 30(4),2013

Combining parametric and non-parametric: Kalman filtering meets Machine Learning (1)

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

General case: $f(x,t) : \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}$

Two fundamental assumptions:

1. finite number of measurement locations $\mathcal{X}_{meas} = \{x_1, \ldots, x_M\}$
2. separable kernels: $k(x,t,x',t') = k_s(x,x')h(\tau), \;\; \tau = t - t'$

$$\mathbf{f}_t = [f(x_1,t) \;\; f(x_2,t) \cdots f(x_M,t)]^T \qquad \mathbf{f}_t : \mathcal{X}_{meas} \times \mathbb{R} \to \mathbb{R}^M$$

Fourier Transform $S(\omega) = \mathcal{F}[h(\tau)]$

Spectral Factorization $S(\omega) = W(i\omega)W(-i\omega)$

$$W(i\omega) = \frac{b_{r-1}(i\omega)^{r-1} + b_{r-2}(i\omega)^{r-2} + \cdots + b_0}{(i\omega)^r + a_{r-1}(i\omega)^{r-1} + \cdots + a}$$

State Space representation:

$$\dot{s}_t = F s_t + G w_t$$
$$z_t = H s_t$$

$$F = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{r-1} \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

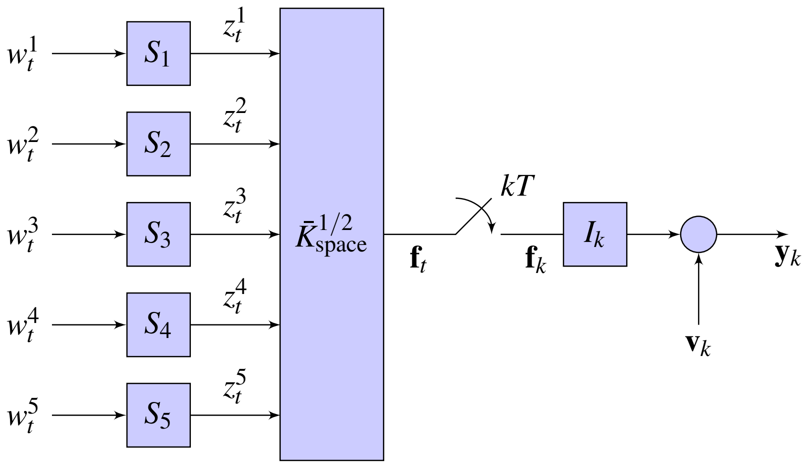
$$H = \begin{bmatrix} b_0 & b_1 & b_2 & \ldots & b_{r-1} \end{bmatrix},$$

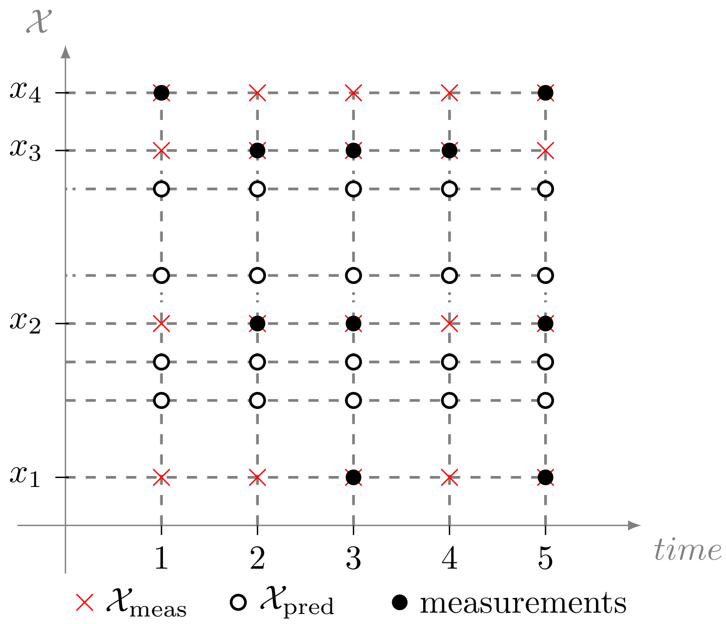
Combining parametric and non-parametric: Kalman filtering meets Machine Learning (2)

$$\mathbf{f}_t = [f(x_1,t)\ f(x_2,t)\cdots f(x_M,t)]^T$$

Dynamics of $\mathbf{f}_t$:

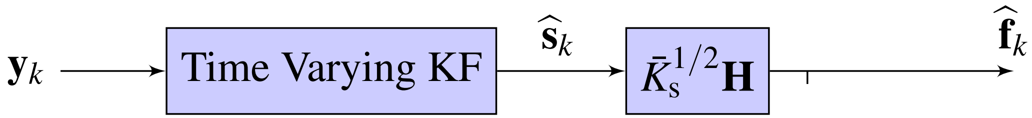
$$\dot{s}_t^j = F s_t^j + G w_t^j$$
$$z_t^j = H s_t^j, \qquad j = 1,\ldots,M$$
$$\mathbf{f}_t = K_s^{1/2}\mathbf{z}_t, \quad [K_s]_{mn} = k_s(x_m,x_n)$$
$$y_k^j = f_k(x_j) + v_k^j \qquad x_j \in M(k) \subset \mathcal{X}_{meas}\ , \ \ v_k^j \sim \mathcal{N}(0,\sigma^2)$$

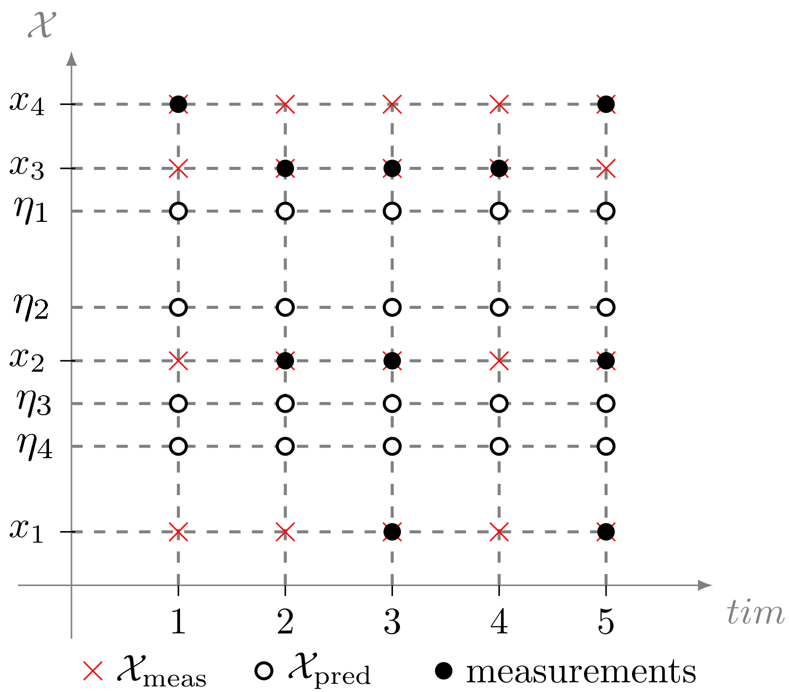




Combining parametric and non-parametric: Kalman filtering meets Machine Learning (3)

$$\widehat{\mathbf{f}}_k = \mathbb{E}[\mathbf{f}_{kT} \mid \{(y_i, x_i)\}, i \in M(k)]$$





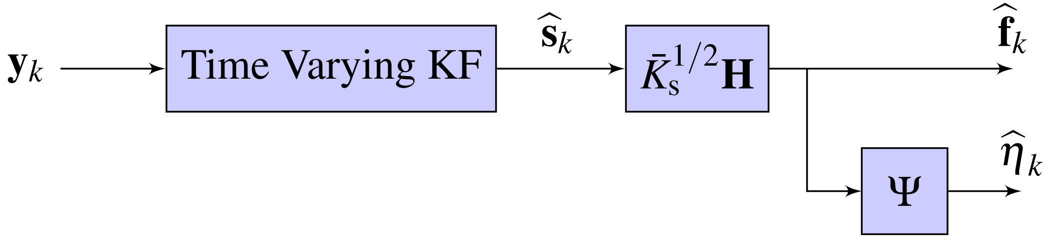
$$\begin{aligned}
\widehat{s}_{k+1|k} &= A\widehat{s}_{k|k} \\
\Sigma_{k+1|k} &= A\Sigma_{k|k}A^T + Q \\
\widehat{s}_{k+1|k+1} &= \widehat{s}_{k+1|k} + L_{k+1}\left(y_{k+1} - C_k\widehat{s}_{k+1|k}\right) \\
\Sigma_{k+1|k+1} &= \Sigma_{k+1|k} - L_{k+1}C_k\Sigma_{k+1|k} \\
L_{k+1} &= \Sigma_{k+1|k}C_k^T\left(C_k\Sigma_{k+1|k}C_k^T + R\right)^{-1}
\end{aligned}$$

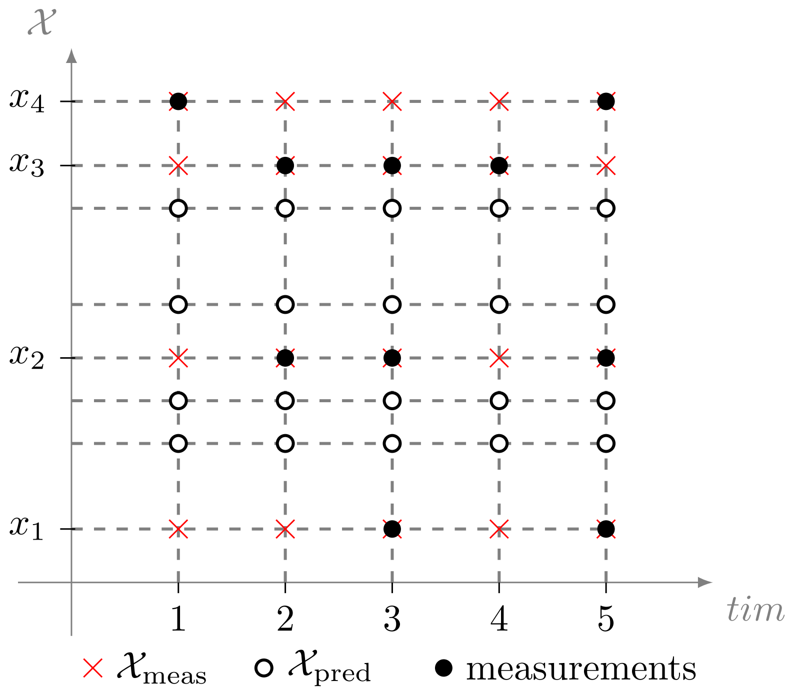
$$A = \mathrm{blkdiag}(\overline{F}, \dots, \overline{F}), \;\; Q = \mathrm{blkdiag}(\overline{Q}, \dots, \overline{Q})$$
$$\overline{F} = e^{FT}, Q = \int_0^T e^{F\tau}GG^T(e^{F\tau})^T d\tau, R = \sigma^2 I$$
$$C_k := I_k K_s^{1/2}\mathbf{H}, \;\; \mathbf{H} = \mathrm{blkdiag}(H, \dots, H), \quad I_k \in \{0,1\}^{M_k \times M}$$

$$\mathcal{X}_{pred} = \{x_{M+1}, \dots, x_{M+S}\}, \mathcal{X}_{pred} \cap \mathcal{X}_{meas} = \emptyset$$
$$\boldsymbol{\eta}_t = [f(x_{M+1}, t) \cdots f(x_{M+S}, t)]^T$$
$$\widehat{\boldsymbol{\eta}}_k = \Psi\widehat{\mathbf{f}}_k, \quad \Psi \in \mathbb{R}^{S \times M}, \Psi = \Psi(\mathcal{X}_{meas}, \mathcal{X}_{pred})$$

Combining parametric and non-parametric: Kalman filtering meets Machine Learning (4)

$$\widehat{\mathbf{f}}_k = \mathbb{E}[\mathbf{f}_{kT} \mid \{(y_i, x_i)\}, i \in M(k)]$$





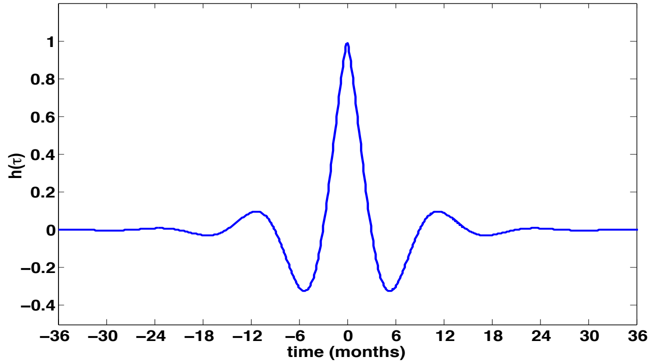
Truncated Gaussian regression vs Kalman-based Gaussian regression

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

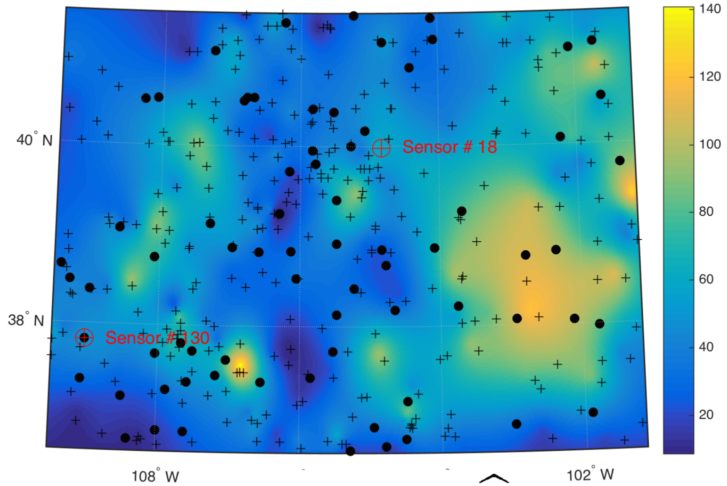
Colorado Weather Dataset: 365 stations, 100 years, monthly rain precipitation

$$K_s(x,x') = e^{-\sigma_s \|x-x'\|}\,, \qquad \sigma_s = 0.5\,,$$ Space kernel

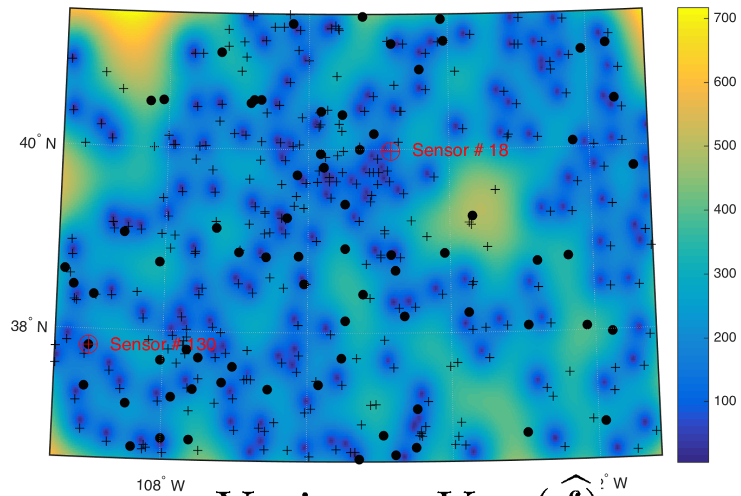
$$h(\tau) = \lambda \cos(2\pi\ f|\tau|)e^{-\sigma_t|\tau|}\,, \quad \lambda = 2\times 10^3\,,\ \sigma_t = 0.2\,,$$ Time-Kernel



	Memory [MB]	CPU time [sec.]
Kalman-based Alg.1	4	0.02
Classical GP (all data)	$1.5\ 10^6$	NA
Truncated GP (1 year data)	150	15
Truncated GP (2 years data)	600	120
Truncated GP (3 years data)	1300	410



Estimate $\widehat{f}$

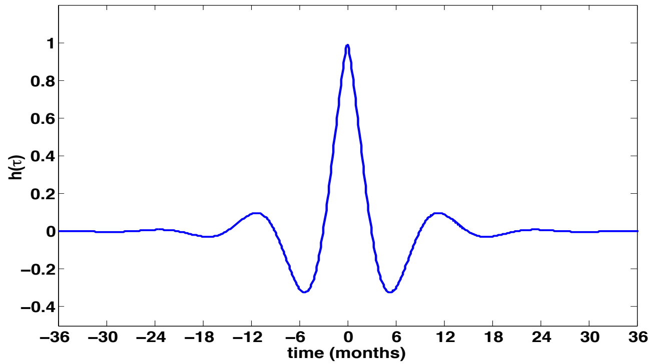


Variance $Var(\widehat{f})$

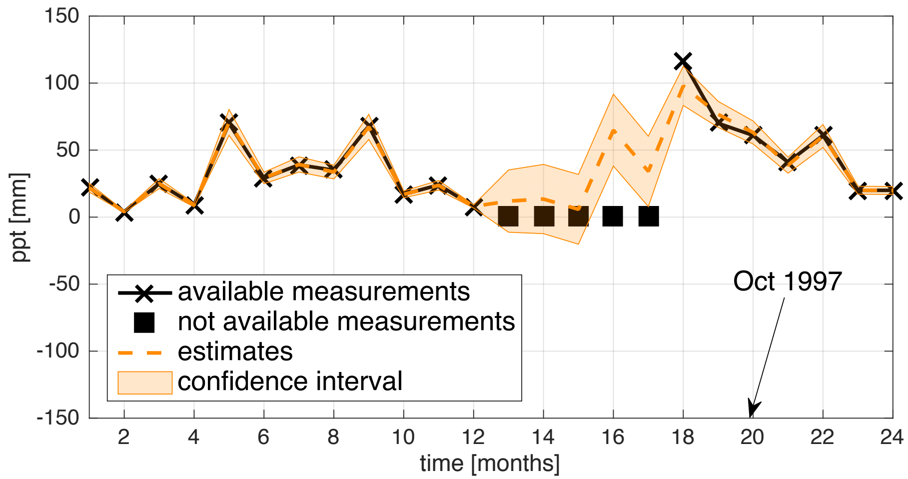
Truncated Gaussian regression vs Kalman-based Gaussian regression

$$K_s(x,x') = e^{-\sigma_s \|x-x'\|}\,, \qquad \sigma_s = 0.5\,,$$ Space kernel

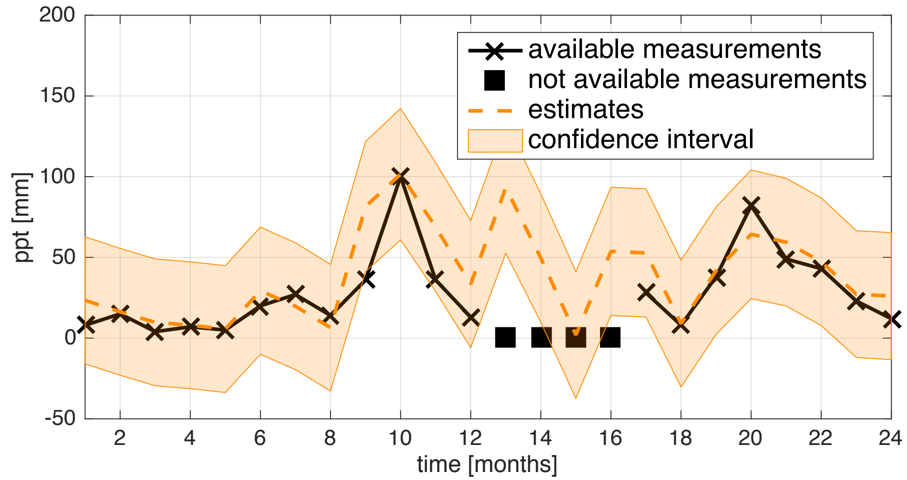
$$h(\tau) = \lambda \cos(2\pi\ f|\tau|)e^{-\sigma_t|\tau|}\,, \quad \lambda = 2\times 10^3\,,\ \sigma_t = 0.2\,,$$ Time-Kernel



	Memory [MB]	CPU time [sec.]
Kalman-based Alg.1	4	0.02
Classical GP (all data)	1.5 10^6	NA
Truncated GP (1 year data)	150	15
Truncated GP (2 years data)	600	120
Truncated GP (3 years data)	1300	410

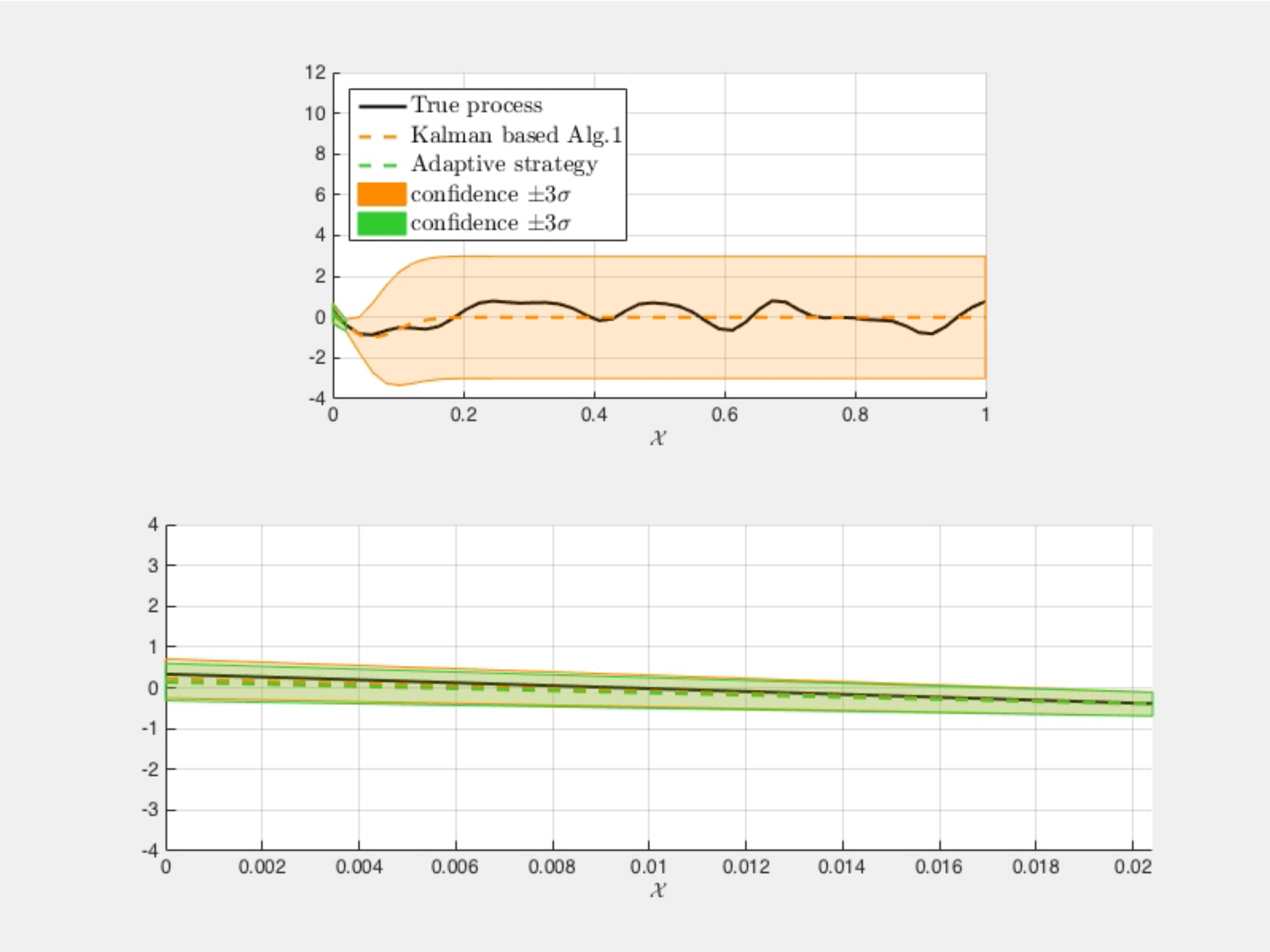


Estimate on measured location



Estimate on non-measured location

Dynamic Grid (suboptimal solution)



Outline

- Motivations, target applications & challenges
- Parametric regression
- Non-parametric regression
- Semi non-parametric regression
- Non-parametric regression for dynamical systems
- Conclusion and open problems

Conclusions & open problems

- Non-parametric approach has great potential but it is unclear how to
  - make it distributed
  - incorporate time
  - adaptively design the sampling density, i.e. $\mu(x) \propto \widehat{f}(x)$
- Many details swept under the carpet:
  - Real-time and distributed design of regularization parameter for non-parametric approaches
  - Packet loss & asynchronous computation
  - Computation of eigenfunctions
- Integration of learning with control & optimization

References

- Parametric vs non-parametric
  - D. Varagnolo, G. Pillonetto, L. Schenato. **Distributed parametric and nonparametric regression with on-line performance bounds computation.** *Automatica*, vol. 48(10), pp. 2468 -- 2481, 2012
- Cloud-based vs peer-to-peer
  - M. Todescato, A. Carron, R. Carli, G. Pillonetto, L. Schenato. **Multi-Robots Gaussian Estimation and Coverage Control: from Server-based to Distributed Architecture.** *Automatica [to appear]*
- Global vs Local estimation
  - D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, L. Schenato. **Newton-Raphson Consensus for Distributed Convex Optimization.** *IEEE Transactions on Automatic Control*, vol. 61(4), pp. 994--1009, 2016
  - A. Carron, M. Todescato, R. Carli, L. Schenato. **An asynchronous consensus-based algorithm for estimation from noisy relative measurements.** *IEEE Transactions on Control of Network Systems*, vol. 1(3), pp. 283 - 295, 2014
  - N. Bof, M. Todescato, R. Carli, L. Schenato. **Robust Distributed Estimation for Localization in Lossy Sensor Networks.** *6th IFAC Workshop on Distributed Estimation and control in Networked Systems (NecSys16)*, 2016
- Static vs dynamic maps:
  - M. Todescato, A. Carron, R. Carli, L. Schenato, G. Pillonetto. **Machine Learning meets Kalman Filtering.** *55th IEEE Conference on Decision and Control (CDC16)*

Q&A

Thank you

WC IFAC ’17 (Toulouse)

Open Invited Track

Multi-agent distributed learning and optimization of dynamical systems

Proponents: Ruggero Carli (Univ. Padova), Jongeun Choi (Yonsei University Seoul), Hideaki Ishii (Tokyo Institute of Technology), Jerome Le Ny (Polytechnique Montreal), Luca Schenato (Univ. Padova)