Feedback Particle Filter and the Poisson Equation

Controlled Interacting Particle Systems for Nonlinear Filtering

SIAM Conference on Uncertainty Quantification

April 16–19, 2018

Sean Meyn

COGNITION & CONTROL IN COMPLEX SYSTEMS

Department of Electrical and Computer Engineering — University of Florida

Based in part on joint research with
Anand Radhakrishnan, Amirhossein Taghvaei, and Prashant G. Mehta

Thanks to the National Science Foundation

Outline

Poisson's Equation

$$0 = \tilde{c} + \mathcal{D}h$$

$$h(x) = \mathsf{E}\Big[\int_0^\tau \tilde{c}(X(t))\,dt\Big]$$

with $X(0) = x$

Optimal FPF Gain

$$\mathsf{K} = \nabla h$$

Optimal Control

$$\phi_{n+1}(x) = \underset{u}{\operatorname{argmin}}\big\{c(x,u) + \mathcal{D}_u h_n(x)\big\}$$

Optimal MCMC CV

$$\gamma^2_{\text{CLT}} = \langle 2\nabla h, \tilde{c}\rangle$$

$$\gamma^2_\theta = 2\langle h_\theta, \tilde{c}_\theta\rangle$$

$$= 2\|\nabla h_\theta\|^2$$

Poisson's Equation

Poisson's Equation

What is it?

All that is required here is the Langevin Diffusion with potential U:

$$\mathrm{d}\Phi_t = -\nabla U(\Phi_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t, \qquad \Phi \in \mathbb{R}^d$$

invariant density $\rho \propto e^{-U}$.

Poisson's Equation

What is it?

All that is required here is the Langevin Diffusion with potential U:

$$d\Phi_t = -\nabla U(\Phi_t)\,dt + \sqrt{2}\,dW_t, \qquad \Phi \in \mathbb{R}^d$$

invariant density $\rho \propto e^{-U}$.

Function $h \in C^2$ solves Poisson's equation:

$$\mathcal{D}h = -\tilde{c}$$

where

- $c\colon \mathbb{R}^d \to \mathbb{R}$ is the *forcing function*.
- normalized forcing function: $\tilde{c} = c - \eta, \quad \eta = \int c(x)\rho(x)dx$.
- Differential generator:

$$\mathcal{D}f = -\nabla U \cdot \nabla f + \Delta f, \qquad f \in C^2$$

Feedback Particle Filter

$$\begin{aligned} &\text{Signal:} \qquad && \mathrm{d}X_t = a(X_t)\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \rho_0^* \\ &\text{Observation:} \qquad && \mathrm{d}Z_t = c(X_t)\mathrm{d}t + \mathrm{d}W_t \end{aligned}$$

- $\boldsymbol{X} := \{X_t : t \geq 0\}$ is the state process.
- $\boldsymbol{Z} := \{Z_t : t \geq 0\}$ is the observation process.
- $a(\,\cdot\,)$, $c(\,\cdot\,)$ are C^1 functions.
- $\{B_t\}$,$\{W_t\}$ are mutually independent Wiener processes.
- ρ_t^* posterior distribution: $P(X_t \mid Z_s : s \leq t)$

Feedback Particle Filter

$$\begin{aligned} &\text{Signal:} && \mathrm{d}X_t = a(X_t)\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \rho_0^* \\ &\text{Observation:} && \mathrm{d}Z_t = c(X_t)\mathrm{d}t + \mathrm{d}W_t \end{aligned}$$

- $\boldsymbol{X} := \{X_t : t \geq 0\}$ is the state process.
- $\boldsymbol{Z} := \{Z_t : t \geq 0\}$ is the observation process.
- $a(\cdot)$, $c(\cdot)$ are C^1 functions.
- $\{B_t\}$,$\{W_t\}$ are mutually independent Wiener processes.
- ρ_t^* posterior distribution: $P(X_t \mid Z_s : s \leq t)$

Nonlinear filter: PDE to compute ρ_t^*

Feedback Particle Filter

Approximation of posterior :

$$\rho_t^*(A) \approx \rho_t^{(N)}(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{X_t^i \in A\}, \quad A \in \mathcal{B}(\mathbb{R}^d).$$

Feedback Particle Filter

Approximation of posterior :

$$\rho_t^*(A) \approx \rho_t^{(N)}(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{X_t^i \in A\}, \quad A \in \mathcal{B}(\mathbb{R}^d).$$

Particle dynamics

$$\mathsf{d}X_t^{(i)} = a(X_t^i)dt + \mathsf{d}B_t^i + \mathsf{d}U_t^i, \qquad i = 1\ldots, N$$

- $X_t^i \in \mathbb{R}$ is the state of the i^{th} particle at time t
- U_t^i is the "control input"
- $\{B_t^i\}$ are mutually independent Wiener processes – statistically identical to state disturbance

Feedback Particle Filter

Particle dynamics

$$\mathrm{d}X_t^{(i)} = a(X_t^i)dt + \mathrm{d}B_t^i + \mathrm{d}U_t^i, \quad i = 1 \text{ to } N$$

$$\mathrm{d}U_t^i = \mathsf{K}_t(X_t^i) \circ \big(\overbrace{\mathrm{d}Z_t - \tfrac{1}{2}[c(X_t^i) + \hat{c}_t]\mathrm{d}t}^{\mathrm{d}I_t^i}\big)\,,$$

I_t^i : Innovations process

K_t : FPF gain, similar in nature to the Kalman gain.

Feedback Particle Filter

Particle dynamics

$$\mathrm{d}X_t^{(i)} = a(X_t^i)dt + \mathrm{d}B_t^i + \mathrm{d}U_t^i, \quad i = 1 \text{ to } N$$

$$\mathrm{d}U_t^i = \mathsf{K}_t(X_t^i) \circ \overbrace{\left(\mathrm{d}Z_t - \tfrac{1}{2}[c(X_t^i) + \hat{c}_t]\mathrm{d}t\right)}^{\mathrm{d}I_t^i},$$

I_t^i : Innovations process
K_t : FPF gain, similar in nature to the Kalman gain.

Representation: $\mathsf{K}_t = \nabla h$

h solves **Poisson's equation:** $-\tilde{c} = \mathcal{D}h = -\nabla U \cdot \nabla h + \Delta h$.

- Forcing function c is the observation function, $\mathrm{d}Z_t = c(X_t)\mathrm{d}t + \mathrm{d}W_t$.
- Potential $U_t = -\log(\rho_t)$

$$\widehat{\mathsf{K}} = \sum_{i=1}^{N} \Big[\beta_i^{0*} S(x^i, \cdot) + \sum_{k=1}^{d} \beta_i^{k*} \frac{\partial}{\partial x_k} S(x^i, \cdot) \Big]$$

Monte-Carlo Techniques for Approximation

Monte-Carlo Approximation Methods

Goal of TD-Learning (in this context): for a given function class $\mathcal{H}$, find best approximation to Poisson's equation in $L_2(\rho)$:

$$g := \arg\min_{g \in \mathcal{H}} \|h - g\|_{L^2}^2$$

One of many challenges:

Monte-Carlo Approximation Methods

Goal of TD-Learning (in this context): for a given function class $\mathcal{H}$, find best approximation to Poisson's equation in $L_2(\rho)$:

$$g := \arg\min_{g\in\mathcal{H}} \|h - g\|^2_{L^2}$$

One of many challenges:

no algorithm exists for state spaces of dimension > 1 [12, 7]

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Two approaches for $\mathcal{H}$ have been considered:

- Finitely parameterized family: [3] "*differential TD Learning*"

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|^2_{L^2}$$

Two approaches for $\mathcal{H}$ have been considered:

- Finitely parameterized family: [3] "*differential TD Learning*"
- Reproducing kernel Hilbert space (RKHS) [4]

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Two approaches for $\mathcal{H}$ have been considered:

- Finitely parameterized family: [3] "*differential TD Learning*"
- Reproducing kernel Hilbert space (RKHS) [4]

Choice of basis is not an easy task

$\Longrightarrow$ RKHS framework is far easier to implement.

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Two approaches for $\mathcal{H}$ have been considered:

- Finitely parameterized family: [3] "*differential TD Learning*"
- Reproducing kernel Hilbert space (RKHS) [4]

Choice of basis is not an easy task

$\Longrightarrow$ RKHS framework is far easier to implement.

See also the remarkable kernel approach of Taghvaei & Mehta [1].

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|^2_{L^2}$$

Challenge: the function h is not known,

and hence the objective function is not observable

Resolution: if $h, g \in L^2(\rho)$

$$\langle \nabla h, \nabla g \rangle_{L^2} = -\langle h, \mathcal{D} g \rangle_{L^2} = -\langle \mathcal{D} h, g \rangle_{L^2} .$$

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \operatorname*{arg\,min}_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Challenge: the function h is not known,
and hence the objective function is not observable

Resolution: if $h, g \in L^2(\rho)$

$$\langle \nabla h, \nabla g \rangle_{L^2} = -\langle h, \mathcal{D} g \rangle_{L^2} = -\langle \mathcal{D} h, g \rangle_{L^2}.$$

Applying this and Poisson's equation $\mathcal{D} h = -\tilde{c}$:

$$\begin{aligned} \|\nabla h - \nabla g\|_{L^2}^2 &= \|\nabla h\|_{L^2}^2 + \|\nabla g\|_{L^2}^2 - 2\langle \nabla h, \nabla g \rangle_{L^2} \\ &= \|\nabla h\|_{L^2}^2 + \|\nabla g\|_{L^2}^2 - 2\langle \tilde{c}, g \rangle_{L^2} \end{aligned}$$

Monte-Carlo Approximation Methods

Revisit TD-learning with our goal in mind:

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Observable objective function:

$$g^* = \arg\min_{g \in \mathcal{H}} \Big\{ \|\nabla g\|_{L^2}^2 - 2\langle \tilde{c}, g \rangle_{L^2} \Big\}$$

Monte-Carlo Approximation Methods

$$g^* := \arg\min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|^2_{L^2} = \arg\min_{g \in \mathcal{H}} \Big\{ \|\nabla g\|^2_{L^2} - 2\langle \tilde{c}, g \rangle_{L^2} \Big\}$$

Finite dimensional function class, $\mathcal{H} = \{\theta^{\intercal}\psi : \theta \in \mathbb{R}^{\ell}\}$

Monte-Carlo Approximation Methods

$$g^* := \arg\min_{g\in\mathcal{H}} \|\nabla h - \nabla g\|^2_{L^2} = \arg\min_{g\in\mathcal{H}} \Big\{ \|\nabla g\|^2_{L^2} - 2\langle \tilde{c}, g\rangle_{L^2} \Big\}$$

Finite dimensional function class, $\mathcal{H} = \{\theta^{\intercal}\psi : \theta \in \mathbb{R}^\ell\}$:

$$\theta^* = M^{-1} b,$$

$$M_{ij} = \langle \nabla\psi_i, \nabla\psi_j\rangle_{L^2} \qquad\qquad b_i = \langle \nabla\psi_i, \nabla h\rangle_{L^2} = \langle \psi_i, \tilde{c}\rangle_{L^2}$$

Monte-Carlo Approximation Methods

$$g^* := \arg\min_{g\in\mathcal{H}} \|\nabla h - \nabla g\|^2_{L^2} = \arg\min_{g\in\mathcal{H}} \Big\{ \|\nabla g\|^2_{L^2} - 2\langle \tilde{c}, g\rangle_{L^2} \Big\}$$

Finite dimensional function class, $\mathcal{H} = \{\theta^{\intercal}\psi : \theta \in \mathbb{R}^{\ell}\}$:

$$\theta^* = M^{-1} b,$$

$$\begin{aligned} M_{ij} &= \langle \nabla\psi_i, \nabla\psi_j\rangle_{L^2} \\ &\approx \frac{1}{t}\int_0^t \nabla\psi(\Phi_s)\,\nabla\psi^{\intercal}(\Phi_s)ds \end{aligned}$$

$$\begin{aligned} b_i &= \langle \nabla\psi_i, \nabla h\rangle_{L^2} = \langle \psi_i, \tilde{c}\rangle_{L^2} \\ &\approx \frac{1}{t}\int_0^t \psi(\Phi_s)\,\tilde{c}(\Phi_s)\,ds \end{aligned}$$

Monte-Carlo Approximation Methods

RKHS provides a basis independent approach to function approximation within a (potentially) richer function class.

Assumptions:

- *Symmetric*: $S(x, y) = S(y, x)$ for any $x, y \in \mathbb{R}^d$
- *Positive definite*: For any finite subset $\{x^i\} \subset \mathbb{R}^d$, the matrix $\{M_{ij} := S(x^i, x^j)\}$ is positive definite.
- *Smooth*: S is C^2.

Monte-Carlo Approximation Methods

Vector space $\mathcal{H}^\circ$: all finite linear combinations

$$g_\alpha(y) = \sum_{i=1}^m \alpha_i S(x^i, y), \quad y \in \mathbb{R}^d\,,$$

scalars $\{\alpha_i\} \subset \mathbb{R}$ and $\{x^i\} \subset \mathbb{R}^d$ arbitrary.

Inner product: for $g_\alpha, g_\beta \in \mathcal{H}^\circ$,

$$\langle g_\alpha, g_\beta \rangle_{\mathcal{H}} := \sum_{i,j} \alpha_i \beta_j S(x^i, z^j)$$

Reproducing property: $g_\alpha(x) = \langle g_\alpha, S(x,\,\cdot\,) \rangle\,, \quad x \in \mathbb{R}^d.$

Assume $\mathcal{H}^\circ$ admits a completion $\mathcal{H}$

Monte-Carlo Approximation Methods

Recall goal:

$$g^* = \arg\min_{g \in \mathcal{H}} \Big\{ \|\nabla g\|_{L^2}^2 - 2\langle \tilde{c}, g\rangle_{L^2} \Big\}$$

Approximation via *empirical risk minimization* (ERM):

$$\arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Big[\|\nabla g(x^i)\|^2 - 2\tilde{c}_N(x^i) g(x^i) \Big] + \lambda \|g\|_{\mathcal{H}}^2$$

where $\tilde{c}$ is also approximated:

$$\tilde{c}_N(x) = c(x) - \frac{1}{N} \sum_{i=1}^{N} c(x^i)\,, \quad x \in \mathbb{R}^d\,.$$

Regularization parameter $\lambda > 0$ introduced to avoid overfitting.

Monte-Carlo Approximation Methods

Extended Representer Theorem [Zhou 08]

If loss function $L(x, \cdot, \cdot)$ is convex on $\mathbb{R}^{d+1}$ for each $x \in \mathbb{R}^d$, then the optimizer g^* over $g \in \mathcal{H}$ exists, is unique and has the form

$$g^*(\cdot) = \sum_{i=1}^{N} \Big[\beta_i^{0*} S(x^i, \cdot) + \sum_{k=1}^{d} \beta_i^{k*} \frac{\partial}{\partial x_k} S(x^i, \cdot) \Big],$$

where $\{\beta_i^{k*} \colon i = 1, \cdots, N,\, k = 0, \cdots, d\}$ are real numbers.

Monte-Carlo Approximation Methods

Extended Representer Theorem [Zhou 08]

If loss function $L(x, \cdot\,, \cdot\,)$ is convex on $\mathbb{R}^{d+1}$ for each $x \in \mathbb{R}^d$, then the optimizer g^* over $g \in \mathcal{H}$ exists, is unique and has the form

$$g^*(\,\cdot\,) = \sum_{i=1}^{N} \Big[\beta_i^{0*} S(x^i, \,\cdot\,) + \sum_{k=1}^{d} \beta_i^{k*} \frac{\partial}{\partial x_k} S(x^i, \,\cdot\,) \Big],$$

where $\{\beta_i^{k*} \colon i = 1, \cdots, N,\ k = 0, \cdots, d\}$ are real numbers.

Our loss function is convex: $L(x, g, \nabla g) = \|\nabla g(x)\|^2 - 2\tilde{c}(x) g(x)$

Monte-Carlo Approximation Methods

Solution in one dimension:

$$g^* = \arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Big\{ (g'(x^i))^2 - 2\tilde{c}_N(x^i) g(x^i) \Big\} + \lambda \|g\|_{\mathcal{H}}^2$$

$$g^*(y) = \sum_{i=1}^{N} \Big\{ \beta_i^{0*} S(x^i, y) + \beta_i^{1*} S_x(x^i, y) \Big\}, \quad y \in \mathbb{R}$$

Monte-Carlo Approximation Methods

Solution in one dimension:

$$g^* = \arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Big\{ (g'(x^i))^2 - 2\tilde{c}_N(x^i) g(x^i) \Big\} + \lambda \|g\|^2_{\mathcal{H}}$$

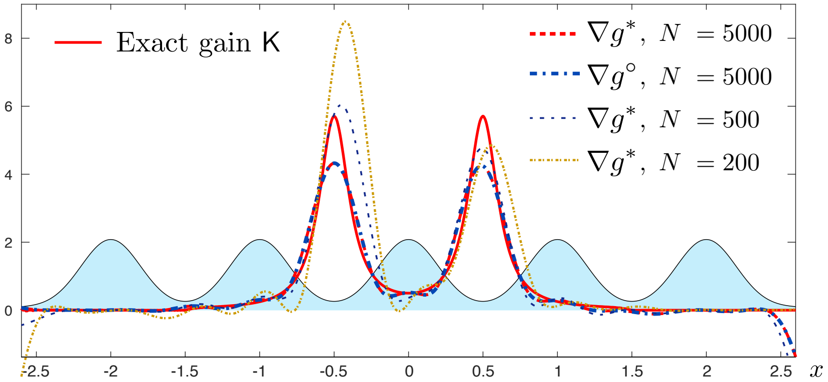
$$g^*(y) = \sum_{i=1}^{N} \Big\{ \beta_i^{0*} S(x^i, y) + \beta_i^{1*} S_x(x^i, y) \Big\}, \quad y \in \mathbb{R}$$

Computation: $\beta^* = M^{-1} b$

$$M = \frac{1}{N} \left[\frac{S_y}{S_{xy}} \right] [S_x \, | S_{xy}] + \lambda \left[\begin{array}{c|c} S & S_y \\ \hline S_x & S_{xy} \end{array} \right]$$

$$b = \frac{1}{N} \left[\frac{S}{S_x} \right] \varsigma, \qquad \varsigma^T = [\tilde{c}_N(x^1), \ldots, \tilde{c}_N(x^N)]$$

$$\beta^T = [\beta_1^0, \ldots, \beta_N^0, \, \beta_1^1, \ldots, \beta_N^1]$$



Numerical Examples

Application to Nonlinear filtering

Test the gain approximation:

$$\min_{\hat{\mathsf{K}} \in \mathcal{K}} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2 = \min_{g \in \mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Using differential TD learning:

- Finite dimensional function space
- RKHS

Application to Nonlinear filtering

Test the gain approximation:

$$\min_{\hat{\mathsf{K}}\in\mathcal{K}} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2 = \min_{g\in\mathcal{H}} \|\nabla h - \nabla g\|_{L^2}^2$$

Using differential TD learning:

- Finite dimensional function space
- RKHS

For comparison: $\mathsf{K}_{\mathsf{BE}^*} = \nabla h^\circ$,

$$h^\circ = \arg\min_{g\in\mathcal{H}} \|\tilde{c} + \mathcal{D}g\|_{L_2}^2$$

Application to Nonlinear filtering

Example: ρ mixture of two Gaussian densities

$c(x) \equiv x$

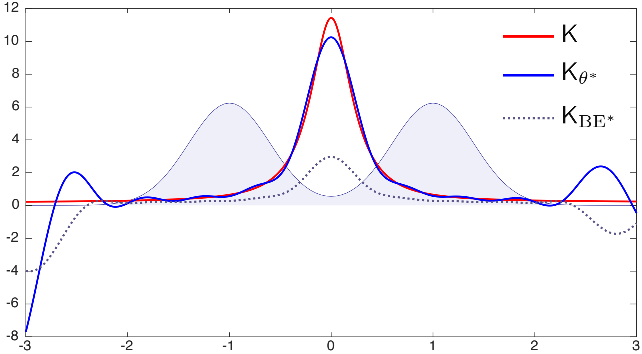
Basis: "Polynomial×Gauss densities" $\{\psi_{i,j}(x) = x^i p_j(x)\}$

Application to Nonlinear filtering

Example: ρ mixture of two Gaussian densities

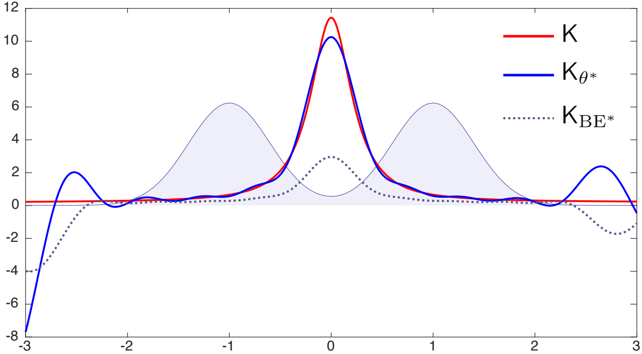
$c(x) \equiv x$

Basis: "Polynomial×Gauss densities" $\{\psi_{i,j}(x) = x^i p_j(x)\}$



Application to Nonlinear filtering

Example: ρ mixture of two Gaussian densities
$c(x) \equiv x$
Basis: "Polynomial×Gauss densities" $\{\psi_{i,j}(x) = x^i p_j(x)\}$

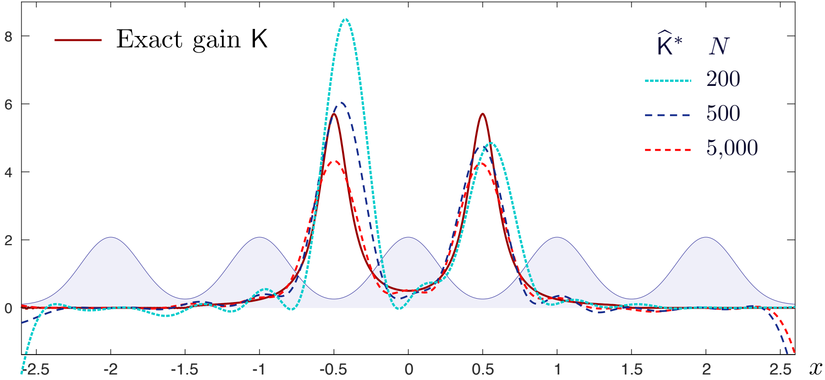


Bellman error optimal is very poor in this example

Application to Nonlinear filtering

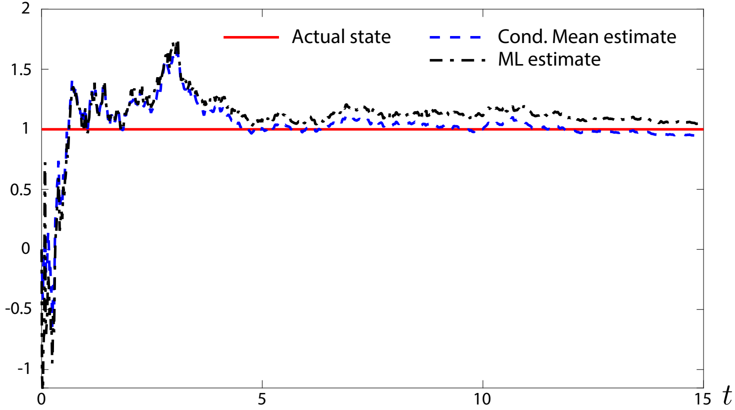
Example: ρ mixture of five Gaussians densities
c difference of indicator functions

RKHS : standard Gaussian kernel



Applications to Nonlinear filtering

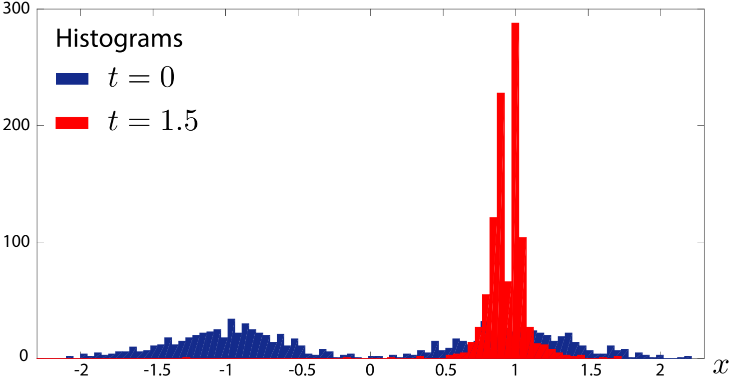
Example: Parameter Estimation with bimodal prior
Observations: parameter plus additive noise



State estimates (Maximum likelihood and conditional mean) from the FPF

Applications to Nonlinear filtering

Example: Parameter Estimation with bimodal prior
Observations: parameter plus additive noise



Histograms of the particles at $t = 0$ and $t = 1.5$

Poisson's Equation

$$0 = \tilde{c} + \mathcal{D}h$$

$$h(x) = \mathsf{E}\Big[\int_0^\tau \tilde{c}(X(t))\,dt\Big]$$

with $X(0) = x$

Optimal FPF Gain

$$\mathsf{K} = \nabla h$$

Optimal Control

$$\phi_{n+1}(x) = \operatorname*{argmin}_u \{c(x,u) + \mathcal{D}_u h_n(x)\}$$

Optimal MCMC CV

$$\begin{aligned}\gamma^2_{\mathrm{CLT}} &= \langle 2\nabla h, \tilde{c}\rangle \\ \gamma^2_\theta &= 2\langle h^\theta, \tilde{c}^\theta\rangle \\ &= 2\|\nabla h^\theta\|^2\end{aligned}$$

Conclusions

Conclusions

Every paper in this domain raises more questions than answers:

- The representation $\mathsf{K} = \nabla h$ remains a deep mathematical mystery.

Conclusions

Every paper in this domain raises more questions than answers:

- The representation $\mathsf{K} = \nabla h$ remains a deep mathematical mystery.
- Absent are mathematical techniques to understand filter robustness

Conclusions

Every paper in this domain raises more questions than answers:

- The representation $\mathsf{K} = \nabla h$ remains a deep mathematical mystery.
- Absent are mathematical techniques to understand filter robustness
- Myriad of algorithmic questions:
  - Kernel choices (see poster of Taghvaei last night)
  - Reduced complexity differential loss algorithms

all with the goal of a more “plug and play” architecture

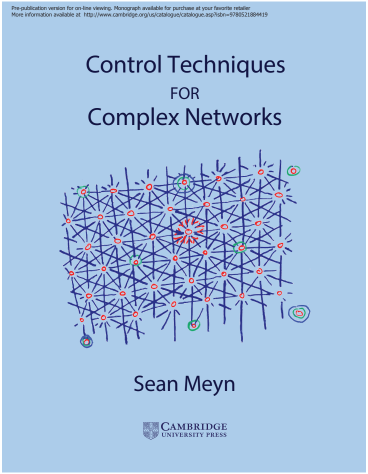
Conclusions

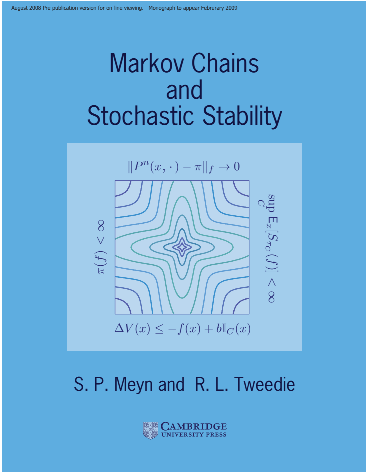
Every paper in this domain raises more questions than answers:

- The representation $\mathsf{K} = \nabla h$ remains a deep mathematical mystery.
- Absent are mathematical techniques to understand filter robustness
- Myriad of algorithmic questions:
  - Kernel choices (see poster of Taghvaei last night)
  - Reduced complexity differential loss algorithms

  all with the goal of a more "plug and play" architecture
- Applications beyond nonlinear filtering:
  - Variance reduction using control variates
  - Reinforcement learning / approximate dynamic programming

Thank You


Pre-publication version for on-line viewing. Monograph available for purchase at your favorite retailer
More information available at http://www.cambridge.org/us/catalogue/catalogue.asp?isbn=9780521884419
Control Techniques
FOR
Complex Networks
Sean Meyn
CAMBRIDGE
UNIVERSITY PRESS


August 2008 Pre-publication version for on-line viewing. Monograph to appear Februrary 2009
Markov Chains
and
Stochastic Stability
$\|P^n(x,\cdot)-\pi\|_f \to 0$
$\pi(f) < \infty$
$\sup_C \mathsf{E}_x[S_{\tau_C}(f)] < \infty$
$\Delta V(x) \le -f(x) + b\mathbb{I}_C(x)$
S. P. Meyn and R. L. Tweedie
CAMBRIDGE
UNIVERSITY PRESS

References

Selected References I

More at www.meyn.ece.ufl.edu

[1] A. Taghvaei and P. G. Mehta. Gain function approximation in the feedback particle filter. In *IEEE Conference on Decision and Control*, pages 5446–5452, Dec 2016.

[2] A. Taghvaei, P. G. Mehta, and S. P. Meyn. Error estimates for the kernel gain function approximation in the feedback particle filter. In *Proc. of the American Control Conference and arXiv*, 2017.

[3] A. Radhakrishnan, A. Devraj and S. Meyn, Learning techniques for feedback particle filter design. *55th Conference on Decision and Control*, Las Vegas, NV, 2016.

[4] A. Radhakrishnan, S. Meyn, Feedback particle filter design using a differential-loss reproducing kernel Hilbert space. *2018 American Control Conference*, Milwaukee, WI, 2018.

[5] T. Yang, P. Mehta, and S. Meyn. Feedback particle filter. *IEEE Trans. Automat. Control*, 58(10):2465–2480, Oct 2013.

[6] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn. Multivariable feedback particle filter. *Automatica*, 71:10–23, 9 2016.

Selected References II

More at www.meyn.ece.ufl.edu

[7] S.P.Meyn, Control Techniques for Complex Networks. Cambridge University Press, Dec 2007.

[8] S. P. Meyn and R. L. Tweedie. Markov chains and stochastic stability. Cambridge University Press, Cambridge, second edition, 2009. Published in the Cambridge Mathematical Library. 1993 edition online.

[9] P. W. Glynn and S. P. Meyn. A Liapounov bound for solutions of the Poisson equation. *Ann. Probab.*, 24(2):916–931, 1996.

[10] A. Devraj, I. Kontoyiannis, and S. Meyn. Geometric Ergodicity in a Weighted Sobolev Space. *ArXiv e-prints, and Submitted for publication*, November, 2017.

[11] A. Devraj, I. Kontoyiannis, and S. Meyn. Geometric Ergodicity in a Weighted Sobolev Space: Part 2, Markovian diffusions. *In preparation*, 2018.

[12] J. N. Tsitsiklis and B. Van Roy. On average versus discounted reward Temporal-Difference Learning. *Machine Learning*, 49(2):179–191, 2002.

Selected References III

More at www.meyn.ece.ufl.edu

[13] S. Asmussen and P. W. Glynn. Stochastic Simulation: Algorithms and Analysis. Volume 57 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, New York, 2007.

[14] S. Henderson. Variance Reduction Via an Approximating Markov Process. PhD thesis, Stanford University, Stanford, California, USA, 1997.

[15] S. Kim and S. G. Henderson. Adaptive control variates for finite-horizon simulation. *Math. Oper. Res.*, 32(3):508–527, 2007.

[16] S. G. Henderson, S. P. Meyn, and V. B. Tadić. Performance evaluation and policy selection in multiclass networks. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):149–189, 2003. Special issue on learning, optimization and decision making (invited).

[17] A. M. Devraj and S. P. Meyn, Differential TD learning for value function approximation. *55th Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016.

[18] D.X. Zhou, Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, Vol. 220, Issues 1–2, 2008.