Application of learning algorithms to nonlinear filtering and MCMC

Amazon Interview Presentation

May 16, 2021

Anand Radhakrishnan

COGNITION & CONTROL
IN COMPLEX SYSTEMS

Work done as part of PhD Research conducted at Department of ECE — University of Florida

Supervisor: Prof. Sean Meyn

List of Publications

Feedback Particle Filter:

- A. Radhakrishnan, A. M. Devraj, S. P. Meyn, **"Learning Techniques for Feedback Particle Filter Design"** *IEEE Conference on Decision and Control, Dec 2016.*
- A. Radhakrishnan, and S. P. Meyn, **"Feedback Particle Filter Design Using a Differential-Loss Reproducing Kernel Hilbert Space"** *American Control Conference, June 2018.*
- A. Radhakrishnan, and S. P. Meyn, **"Gain Function Tracking in the Feedback Particle Filter"** *American Control Conference, July 2019.*

Markov chain Monte Carlo methods:

- N. Brosse, A. Durmus, S. P. Meyn, E. Moulines, and A. Radhakrishnan, **"Diffusion Approximation and Control Variates for MCMC"** *Annals of Applied Probability, July 2019.*

Code on Github:

- FPF Matlab and Python code - **https://github.com/a4anandr/FPC-code**
- MCMC Matlab code - **https://github.com/a4anandr/MCMC-code**
- PhD Dissertation available at - **https://github.com/a4anandr/PhD-Dissertation**

Outline

1. Motivational applications
2. Poisson's Equation
3. Differential LSTD Learning
4. Applications to Nonlinear Filtering
5. Applications to Markov Chain Monte Carlo
6. Conclusions and Future Work

Motivational applications

- Reinforcement learning - AlphaGo, Atari games, Self-driving cars etc.

Motivational applications

- Reinforcement learning - AlphaGo, Atari games, Self-driving cars etc.
- Nonlinear state estimation - Fitness tracker, smart watch, weather prediction models.

Motivational applications

- Reinforcement learning - AlphaGo, Atari games, Self-driving cars etc.
- Nonlinear state estimation - Fitness tracker, smart watch, weather prediction models.
- Markov chain Monte Carlo - Computer vision, Decision theory etc.

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

$$\phi^{n+1}(x) = \operatorname*{arg\,min}_u \big\{ c(x,u) + \mathcal{D}_u h^n(x) \big\}$$

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

- Second order partial differential equation with applications in various fields.
- General form in stochastic systems

$$\mathcal{D}h = -f$$

$\mathcal{D}$ - differential operator
f - forcing function, usually centered, i.e. $\mathsf{E}[f] = 0$.

Poisson's Equation

- Second order partial differential equation with applications in various fields.
- General form in stochastic systems

$$\mathcal{D}h = -f$$

$\mathcal{D}$ - differential operator
f - forcing function, usually centered, i.e. $\mathsf{E}[f] = 0$.
h - solution to Poisson's equation

Poisson's Equation

Stochastic optimal control

Example: In average cost optimal control problems,

$$c(x,u) + P_u h(x) = h(x) + \eta$$

$c(x,u)$ - Cost function associated with state x and action u.

Poisson's Equation

Stochastic optimal control

Example: In average cost optimal control problems,

$$c(x,u) + P_u h(x) = h(x) + \eta$$

$c(x,u)$ - Cost function associated with state x and action u.
P_u - Transition kernel of the controlled Markov chain,

Poisson's Equation

Stochastic optimal control

Example: In average cost optimal control problems,

$$c(x,u) + P_u h(x) = h(x) + \eta$$

$c(x,u)$ - Cost function associated with state x and action u.

P_u - Transition kernel of the controlled Markov chain,
i.e. $P_u h(x) := \mathsf{E}[h(X_{t+1}) | X_t = x, U_t = u]$.

Poisson's Equation

Stochastic optimal control

Example: In average cost optimal control problems,

$$c(x,u) + P_u h(x) = h(x) + \eta$$

$c(x,u)$ - Cost function associated with state x and action u.
P_u - Transition kernel of the controlled Markov chain,
η - Average cost.
h - Relative value function

Poisson's Equation

Stochastic optimal control

Example: In average cost optimal control problems,

$$\min_{u \in \mathsf{U}} \big[c(x,u) + P_u h^*(x) \big] = h^*(x) + \eta^*$$

$c(x,u)$ - Cost function associated with state x and action u.
P_u - Transition kernel of the controlled Markov chain.
η^* - Optimal average cost.
h^* - Optimal relative value function

Poisson's equation is the average-cost dynamic programming equation for a fixed policy u.

Poisson's Equation

Stochastic optimal control

Example: In discounted cost optimal control problems,

$$\min_{u \in \mathsf{U}} \big[c(x,u) + P_u h_\gamma^*(x) \big] = (1+\gamma) h_\gamma^*(x)$$

$c(x,u)$ - Cost function associated with state x and action u.

P_u - Transition kernel of the controlled Markov chain.

γ - Discount rate, $\gamma \geq 0$ (usual notation $\beta = \frac{1}{1+\gamma}$, $\beta \leq 1$).

h_γ^* - Optimal discounted-cost value function

Poisson's equation is the average-cost dynamic programming equation for a fixed policy u.

Poisson's Equation

Langevin Diffusion

Langevin diffusion is given by the SDE,

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

$U \in C^1$ is called the *potential function*.

$\boldsymbol{W} = \{W_t : t \geq 0\}$ is a standard Brownian motion on $\mathbb{R}^d$.

Poisson's Equation

Langevin Diffusion

Langevin diffusion is given by the SDE,

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

$U \in C^1$	is called the *potential function*.
$\boldsymbol{W} = \{W_t : t \geq 0\}$	is a standard Brownian motion on $\mathbb{R}^d$.

- May be regarded as a d-dimensional gradient flow with "noise".

Poisson's Equation

Langevin Diffusion

Langevin diffusion is given by the SDE,

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

$U \in C^1$ is called the *potential function*.

$\boldsymbol{W} = \{W_t : t \geq 0\}$ is a standard Brownian motion on $\mathbb{R}^d$.

- May be regarded as a d-dimensional gradient flow with "noise".
- Diffusion is "reversible", with unique *invariant density* $\rho = e^{-U+\Lambda}$, where Λ is a normalizing constant.

Poisson's Equation

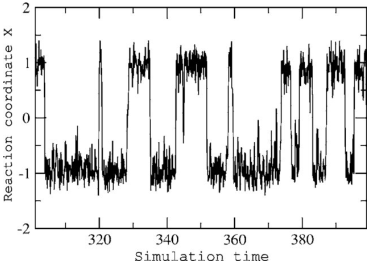
Langevin Diffusion

Langevin diffusion is given by the SDE,

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

$U \in C^1$ is called the *potential function*.

$\boldsymbol{W} = \{W_t : t \geq 0\}$ is a standard Brownian motion on $\mathbb{R}^d$.



A typical Langevin diffusion process

Poisson's Equation

Langevin Diffusion

Differential generator:

$$\mathcal{D}f := \lim_{t \to 0} \frac{\mathsf{E}[f(\Phi_t) | \Phi_0 = x] - f(x)}{t}$$

Poisson's Equation

Langevin Diffusion

Differential generator:

$$\begin{aligned}\mathcal{D}f &:= \lim_{t\to 0} \frac{\mathsf{E}[f(\Phi_t)|\Phi_0 = x] - f(x)}{t} \\ &= -\nabla U \cdot \nabla f + \Delta f, \qquad f \in C^2,\end{aligned}$$

where ∇ is the gradient and Δ is the Laplacian.

Poisson's Equation

Langevin Diffusion

Differential generator:

$$\begin{aligned}\mathcal{D}f &:= \lim_{t\to 0} \frac{\mathsf{E}[f(\Phi_t)|\Phi_0 = x] - f(x)}{t} \\ &= -\nabla U \cdot \nabla f + \Delta f, \qquad f \in C^2,\end{aligned}$$

where ∇ is the gradient and Δ is the Laplacian.
Let $c\colon \mathbb{R}^d \to \mathbb{R}$ be a function of interest, and

$$\eta = \int c(x)\rho(x)dx = \langle c, 1\rangle_{L^2}.$$

Poisson's Equation

Langevin Diffusion

Differential generator:

$$\begin{aligned}\mathcal{D}f &:= \lim_{t\to 0} \frac{\mathsf{E}[f(\Phi_t)|\Phi_0 = x] - f(x)}{t} \\ &= -\nabla U \cdot \nabla f + \Delta f, \qquad f \in C^2,\end{aligned}$$

where ∇ is the gradient and Δ is the Laplacian.
Let $c\colon \mathbb{R}^d \to \mathbb{R}$ be a function of interest, and

$$\eta = \int c(x)\rho(x)dx = \langle c, 1\rangle_{L^2}.$$

Function $h \in C^2$ solves Poisson's equation with forcing function c if

$$\mathcal{D}h := -\tilde{c}, \qquad \tilde{c} = c - \eta.$$

$$h := \int_0^\infty \mathsf{E}[\tilde{c}(\Phi_t)]\mathrm{d}t$$

Poisson's Equation

Existence of a solution

- A solution exists under weak assumptions on U and c [Glynn & Meyn 96, Kontoyiannis et al. 12].
- Representations for the gradient of h and bounds are obtained in [Laugesen et al. 15, Devraj et al. 18].
- A smooth solution $h \in C^2$ exists under stronger conditions in [Pardoux et al. 01], subject to growth conditions on c similar to [Glynn & Meyn 96].

Poisson's Equation

Approximate solution to Poisson's equation

Obtaining an analytical solution for h is difficult outside special cases. Hence approximation.

Poisson's Equation

Approximate solution to Poisson's equation

Obtaining an analytical solution for h is difficult outside special cases. Hence approximation.

Goal: For a given function class $\mathcal{H}$, find the minimizer of

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

Such minimum norm optimization problems can be solved using TD learning [Tsitsikilis 99].

Poisson's Equation

Approximate solution to Poisson's equation

Obtaining an analytical solution for h is difficult outside special cases. Hence approximation.

Goal: For a given function class $\mathcal{H}$, find the minimizer of

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

Such minimum norm optimization problems can be solved using TD learning [Tsitsikilis 99].

Challenge - No algorithm exists to solve $(\star)$ if the process does not regenerate (for diffusions, dim > 1 ruled out).

LSTD Learning

Discounted cost case

Discounted-cost value function:

$$h_\gamma(x) := \int_0^\infty e^{-\gamma t} \mathsf{E}_x[c(\Phi_t)] \mathrm{d}t, \qquad \gamma > 0 : \text{ discount rate}$$

Discounted-cost optimality equation:

$$\gamma h_\gamma = c + \mathcal{D} h_\gamma$$

LSTD Learning
Discounted cost case

Discounted-cost value function:

$$h_\gamma(x) := \int_0^\infty e^{-\gamma t} \mathsf{E}_x[c(\Phi_t)] \mathrm{d}t, \qquad \gamma > 0 : \text{ discount rate}$$

Discounted-cost optimality equation:

$$\gamma h_\gamma = c + \mathcal{D} h_\gamma$$

LSTD goal: $g^* := \operatorname*{arg\,min}_{g \in \mathcal{H}} \| h_\gamma - g \|_{L^2}^2$

LSTD Learning

Discounted cost case

LSTD goal: $g^* := \arg\min_{g \in \mathcal{H}} \|h_\gamma - g\|_{L^2}^2$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle \psi_i, h_\gamma \rangle_{L^2}$$

LSTD Learning

Discounted cost case

LSTD goal: $g^* := \arg\min_{g \in \mathcal{H}} \|h_\gamma - g\|^2_{L^2}$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle \psi_i, h_\gamma \rangle_{L^2}$$

LSTD Learning

Discounted cost case

LSTD goal: $g^* := \arg\min_{g \in \mathcal{H}} \|h_\gamma - g\|^2_{L^2}$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle \psi_i, h_\gamma \rangle_{L^2}$$
$$= \langle \psi_i,\ R_\gamma c \rangle_{L^2}$$

Resolvent kernel: $R_\gamma c\,(x) := \int_0^\infty \mathsf{E}_x\Big[e^{-\gamma t} c(\Phi_t)\Big] \mathrm{d}t$

$$R_\gamma c = (I\gamma - \mathcal{D})^{-1} c$$

LSTD Learning

Discounted cost case

LSTD goal: $g^* := \mathop{\arg\min}_{g \in \mathcal{H}} \|h_\gamma - g\|^2_{L^2}$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle \psi_i, R_\gamma c \rangle_{L^2}$$
$$= \langle R_\gamma^\dagger \psi_i, \, c \rangle_{L^2}$$

Using an adjoint operation and applying the stationarity of Φ.

LSTD Learning

Discounted cost case

$$\text{LSTD goal: } g^* := \arg\min_{g\in\mathcal{H}} \|h_\gamma - g\|^2_{L^2}$$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle R_\gamma^\dagger \psi_i,\, c\rangle_{L^2}$$

$$\text{Eligibility vector: } \varphi(t) := \int_0^\infty e^{-\gamma r} \psi(\Phi_{t-r}) \mathrm{d}r$$

$$R_\gamma^\dagger \psi_i(x) = \mathsf{E}[\varphi_i(t) | \Phi_t = x]$$

LSTD Learning

Discounted cost case

LSTD goal: $g^* := \displaystyle\arg\min_{g\in\mathcal{H}} \|h_\gamma - g\|^2_{L^2}$

For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i$$

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

$$M_{ij} = \langle \psi_i, \psi_j \rangle_{L^2}, \quad b_i = \langle R_\gamma^\dagger \psi_i, c \rangle_{L^2}$$

$$= \mathsf{E}[\varphi_i(t) c(\Phi_t)]$$

LSTD Learning

Discounted cost case

ODE formulation of the LSTD algorithm:

$$
\begin{aligned}
\frac{d}{dt}M(t) &= \psi(\Phi_t)\psi^{\intercal}(\Phi_t) \\
\frac{d}{dt}\varphi(t) &= -\gamma\,\varphi(t) + \psi(\Phi_t) \\
\frac{d}{dt}b(t) &= \varphi(t)c(\Phi_t) \\
\theta(t) &:= M(t)^{-1}b(t)
\end{aligned}
$$

LSTD Learning

Discounted cost case

ODE formulation of the LSTD algorithm:

$$\begin{aligned}
\frac{d}{dt}M(t) &= \psi(\Phi_t)\psi^{\intercal}(\Phi_t) \\
\frac{d}{dt}\varphi(t) &= -\gamma\,\varphi(t) + \psi(\Phi_t) \\
\frac{d}{dt}b(t) &= \varphi(t)c(\Phi_t) \\
\theta(t) &:= M(t)^{-1}b(t)
\end{aligned}$$

By law of large numbers,

$$\lim_{t\to\infty}\theta(t) = \theta^*$$

LSTD Learning

Discounted cost case

ODE formulation of the LSTD algorithm:

$$
\begin{aligned}
\frac{d}{dt} M(t) &= \psi(\Phi_t)\psi^{\intercal}(\Phi_t) \\
\frac{d}{dt} \varphi(t) &= -\gamma\, \varphi(t) + \psi(\Phi_t) \\
\frac{d}{dt} b(t) &= \varphi(t) c(\Phi_t) \\
\theta(t) &:= M(t)^{-1} b(t)
\end{aligned}
$$

By law of large numbers,

$$
\lim_{t\to\infty} \theta(t) = \theta^*
$$

For average-cost ($\gamma = 0$) LSTD requires the existence of a regenerating state.

Differential LSTD Learning (∇-LSTD)

Poisson's equation

Idea: Approximate the gradient of h directly [R et al. 16, Devraj et al. 16]:

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

Differential LSTD Learning (∇-LSTD)

Poisson's equation

Idea: Approximate the gradient of h directly [R et al. 16, Devraj et al. 16]:

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

Need to choose a function class $\mathcal{H}$ for g (or ∇g)

- A finitely parameterized family of functions.
- A reproducing kernel Hilbert space (RKHS).

Differential LSTD Learning (∇-LSTD)

Poisson's equation

Idea: Approximate the gradient of h directly [R et al. 16, Devraj et al. 16]:

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

Need to choose a function class $\mathcal{H}$ for g (or ∇g)

- A finitely parameterized family of functions.
- A reproducing kernel Hilbert space (RKHS).
  Choice of basis is not an easy task
  $\Longrightarrow$ RKHS framework is far easier to implement.

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

Challenge: the function h is not known,

and hence the objective function is not observable

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

Challenge: the function h is not known,

and hence the objective function is not observable

∇-LSTD: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

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

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

Challenge: the function h is not known,

and hence the objective function is not observable

∇-LSTD: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

$$M_{ij} = \langle \nabla\psi_i, \nabla\psi_j \rangle_{L^2}, \quad b_i = \langle \nabla\psi_i, R_{U''}\nabla c \rangle_{L^2}$$

Gen.resolvent kernel: $R_{U''}\nabla c\,(x) := \int_0^\infty \mathsf{E}_x\Big[\exp\Big(-\int_0^t U''(\Phi_s)\,\mathsf{d}s\Big)\nabla c(\Phi_t)\Big]\mathsf{d}t$

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

Challenge: the function h is not known,

and hence the objective function is not observable

∇-LSTD: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

$$M_{ij} = \langle \nabla\psi_i, \nabla\psi_j \rangle_{L^2}, \quad b_i = \langle \nabla\psi_i,\, R_{U''} \nabla c \rangle_{L^2}$$

$$= \langle R^\dagger_{U''} \nabla\psi_i,\, \nabla c \rangle_{L^2}$$

Differential LSTD Learning (∇-LSTD)

Poisson's equation

∇-LSTD goal: $g^* := \underset{g \in \mathcal{H}}{\arg\min} \|\nabla h - \nabla g\|_{L^2}^2$

Challenge: the function h is not known,

and hence the objective function is not observable

∇-LSTD-L: For Langevin diffusion, if $f, g \in L^2(\rho)$

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

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

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

∇-LSTD-L: For Langevin diffusion, if $f, g \in L^2(\rho)$

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

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

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

Differential LSTD Learning (∇-LSTD)

Poisson's equation

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

Challenge: the function h is not known,

and hence the objective function is not observable

∇-LSTD-L: For Langevin diffusion, if $f, g \in L^2(\rho)$

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

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

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

Differential LSTD Learning for Langevin (∇-LSTD-L)

Poisson's equation

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

∇-LSTD-L: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

Differential LSTD Learning for Langevin (∇-LSTD-L)

Poisson's equation

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

∇-LSTD-L: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

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

Differential LSTD Learning for Langevin (∇-LSTD-L)

Poisson's equation

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

∇-LSTD-L: For a linear parameterization

$$g = h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i \implies \nabla g = \sum_{i=1}^{\ell} \theta_i \nabla \psi_i$$

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

$$M_{ij} = \langle \nabla\psi_i, \nabla\psi_j \rangle_{L^2} \approx \frac{1}{t}\int_0^t \nabla\psi(\Phi_s)\, \nabla\psi^{\intercal}(\Phi_s)\, \mathrm{d}s$$

$$b_i = \langle \nabla\psi_i, \nabla h \rangle_{L^2} = \langle \psi_i, \tilde{c} \rangle_{L^2} \approx \frac{1}{t}\int_0^t \psi_i(\Phi_s)\, \tilde{c}(\Phi_s)\, \mathrm{d}s$$

Differential LSTD Learning on RKHS (∇-LSTD-RKHS)

Basics of RKHS

Choose a kernel function $K(x, y)$ that is

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

K defines a unique RKHS $\mathcal{H}$ [Moore-Aronszajn theorem].

Inner product: If $g_\alpha = \sum_i \alpha_i K(x^i, \cdot\,)$ and $g_\beta = \sum_j \beta_j K(y^j, \cdot\,)$,

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

Reproducing property: $g_\alpha(x) = \langle g_\alpha, K(x, \cdot\,) \rangle_{\mathcal{H}}\,, \quad x \in \mathbb{R}^d.$

Differential LSTD learning on RKHS

Empirical risk minimization (ERM)

Recall ∇-LSTD-L goal:

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

Approximation via empirical risk minimization (ERM):

$$\operatorname*{arg\,min}_{g \in \mathcal{H}} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \Big[\|\nabla g(x^i)\|^2 - 2\tilde{c}_N(x^i) g(x^i) \Big]}_{\text{Empirical risk}} + \underbrace{\lambda \|g\|_{\mathcal{H}}^2}_{\text{Regularization}}$$

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

Differential LSTD Learning on RKHS (∇-LSTD-RKHS)

∇-LSTD-RKHS-Opt

Empirical risk minimization (ERM):

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

Classical representer theorem [Wahba 70] is a remarkable result for ERMs in RKHS.

Differential LSTD Learning on RKHS (∇-LSTD-RKHS)

∇-LSTD-RKHS-Opt

Empirical risk minimization (ERM):

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

Classical representer theorem [Wahba 70] is a remarkable result for ERMs in RKHS.

Not applicable to our loss function due to gradient term.

Differential LSTD Learning on RKHS (∇-LSTD-RKHS)

∇-LSTD-RKHS-Opt

Empirical risk minimization (ERM):

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

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:

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

Differential LSTD Learning on RKHS (∇-LSTD-RKHS)

∇-LSTD-RKHS-N

Drawback : Complexity grows linearly with d, since $\beta^* \in \mathbb{R}^{(d+1)\times N}$

Simplified solution : By considering the finite-dimensional function class - $\mathcal{H}_N :=$ span$\{K_{x^i} : 1 \le i \le N\}$

$$g^*(y) = \sum_{j=1}^{N} \beta_i^* K(x^i, y)$$

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

Surprising empirical observation : Simplified solution does as good as the optimal solution for $d \le 5$.

Applications to Nonlinear Filtering

Feedback Particle Filter

Goal: To obtain estimates of the state of a stochastic dynamical system based on noisy partial observations.

Applications to Nonlinear Filtering

Feedback Particle Filter

Goal: To obtain estimates of the state of a stochastic dynamical system based on noisy partial observations.

Kalman filter is optimal for a linear Gaussian system.

Applications to Nonlinear Filtering

Feedback Particle Filter

Goal: To obtain estimates of the state of a stochastic dynamical system based on noisy partial observations.

Kalman filter is optimal for a linear Gaussian system.

For nonlinear systems, conditional distribution fails to be Gaussian, cannot be captured by a finite set of parameters.

Applications to Nonlinear Filtering

Feedback Particle Filter

Goal: To obtain estimates of the state of a stochastic dynamical system based on noisy partial observations.

Kalman filter is optimal for a linear Gaussian system.

For nonlinear systems, conditional distribution fails to be Gaussian, cannot be captured by a finite set of parameters.

Particle filters are popular Monte-Carlo approximations of the nonlinear filter.

Applications to Nonlinear Filtering

Feedback Particle Filter

Problem:

Signal: $$\mathrm{d}X_t = a(X_t)\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \rho_0^*,$$

Observation: $$\mathrm{d}Z_t = c(X_t)\mathrm{d}t + \mathrm{d}W_t,$$

- $X_t \in \mathbb{R}^d$ is the state at time t.
- $\{Z_t : t \geq 0\}$ is the observation process.
- $a(.)$,$c(.)$ are C^1 functions.
- $\{B_t\}$,$\{W_t\}$ are mutually independent Wiener processes.

Applications to Nonlinear Filtering

Feedback Particle Filter

Problem:

Signal: $\mathrm{d}X_t = a(X_t)\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \rho_0^*,$

Observation: $\mathrm{d}Z_t = c(X_t)\mathrm{d}t + \mathrm{d}W_t,$

- $X_t \in \mathbb{R}^d$ is the state at time t.
- $\{Z_t : t \geq 0\}$ is the observation process.
- $a(.)$,$c(.)$ are C^1 functions.
- $\{B_t\}$,$\{W_t\}$ are mutually independent Wiener processes.
- $\rho_t^* := P(X_t | Z_s : s \leq t)$ is the posterior distribution.

Applications to Nonlinear Filtering

Feedback Particle Filter

Feedback particle filter (FPF) [Yang et al. 13] is motivated by techniques from mean-field optimal control.

Applications to Nonlinear Filtering

Feedback Particle Filter

Feedback particle filter (FPF) [Yang et al. 13] is motivated by techniques from mean-field optimal control.

N *particles* are propagated in the form of a controlled system.

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

- $X_t^i \in \mathbb{R}$ is the state of the i^{th} particle at time t
- U_t^i is the control input applied to i^{th} particle
- $\{B_t^i\}$ are mutually independent standard Wiener processes.

Applications to Nonlinear Filtering

Feedback Particle Filter

Feedback particle filter (FPF) [Yang et al. 13] is motivated by techniques from mean-field optimal control.

N *particles* are propagated in the form of a controlled system.

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

- $X_t^i \in \mathbb{R}$ is the state of the i^{th} particle at time t
- U_t^i is the control input applied to i^{th} particle
- $\{B_t^i\}$ are mutually independent standard Wiener processes.

Approximation of ρ_t^*:

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

Applications to Nonlinear Filtering

Feedback Particle Filter

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

I_t^i : Innovations process.

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

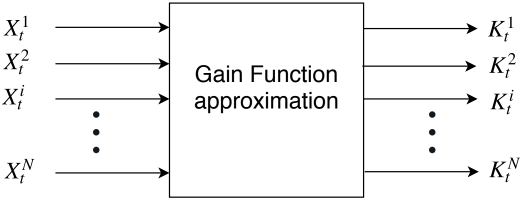
Applications to Nonlinear Filtering

Feedback Particle Filter

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

I_t^i : Innovations process.

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

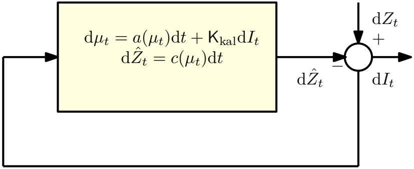


Finite-N implementation

Applications to Nonlinear Filtering

Feedback Particle Filter

KF: $$\mathrm{d}\mu_t = a(\mu_t)\mathrm{d}t + \underbrace{\mathsf{K}_{\mathsf{kal}}(\mathrm{d}Z_t - c(\mu_t)\mathrm{d}t)}_{\text{update}}$$



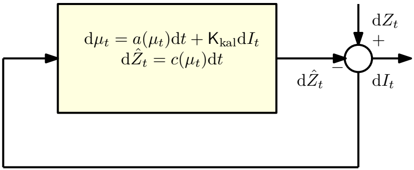
Kalman filter

Applications to Nonlinear Filtering

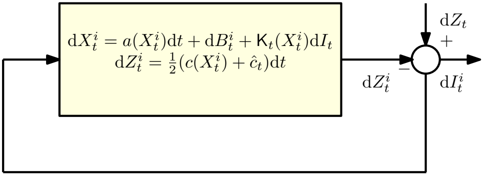
Feedback Particle Filter

KF: $$\mathsf{d}\mu_t = a(\mu_t)\mathsf{d}t + \underbrace{\mathsf{K}_{\mathsf{kal}}(\mathsf{d}Z_t - c(\mu_t)\mathsf{d}t)}_{\text{update}}$$

FPF: $$\mathsf{d}X_t^i = a(X_t^i)\mathsf{d}t + \mathsf{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ (\mathsf{d}Z_t - \tfrac{1}{2}[c(X_t^i) + \hat{c}_t]\mathsf{d}t)}_{\text{update}}$$



Kalman filter



Feedback particle filter (FPF)

Applications to Nonlinear Filtering

FPF gain function

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

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

Applications to Nonlinear Filtering

FPF gain function

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

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

Approximations to K can be obtained by

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

Applications to Nonlinear Filtering

FPF gain function

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

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

Approximations to K can be obtained by

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

Can be solved using ∇-LSTD learning

Applications to Nonlinear Filtering

FPF gain function

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

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

Approximations to K can be obtained by

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

FPF implementation requires online gain estimation for each t.

- ∇-LSTD-RKHS with optimal mean
- ∇-LSTD-RKHS with memory

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Constant gain approximation for K is the minimizer obtained over all deterministic vectors:

$$\widehat{\mathsf{K}}^* := \arg\min_{\widehat{\mathsf{K}} \in \mathbb{R}^d} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Constant gain approximation for K is the minimizer obtained over all deterministic vectors:

$$\widehat{\mathsf{K}}^* := \arg\min_{\widehat{\mathsf{K}} \in \mathbb{R}^d} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$$

Solution is evidently the mean, $\widehat{\mathsf{K}}^* = \mathsf{E}[\mathsf{K}]$.

$$\widehat{\mathsf{K}}_k^* = \langle \mathsf{K}, e_k \rangle_{L^2}$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Constant gain approximation for K is the minimizer obtained over all deterministic vectors:

$$\widehat{\mathsf{K}}^* := \arg\min_{\widehat{\mathsf{K}} \in \mathbb{R}^d} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$$

Solution is evidently the mean, $\widehat{\mathsf{K}}^* = \mathsf{E}[\mathsf{K}]$.

$$\begin{aligned} \widehat{\mathsf{K}}_k^* &= \langle \mathsf{K},\, e_k \rangle_{L^2} \\ &= \langle \nabla h,\, \nabla x_k \rangle_{L^2} \end{aligned}$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Constant gain approximation for K is the minimizer obtained over all deterministic vectors:

$$\widehat{\mathsf{K}}^* := \underset{\widehat{\mathsf{K}} \in \mathbb{R}^d}{\arg\min} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$$

Solution is evidently the mean, $\widehat{\mathsf{K}}^* = \mathsf{E}[\mathsf{K}]$.

$$\begin{aligned} \widehat{\mathsf{K}}_k^* &= \langle \mathsf{K},\, e_k \rangle_{L^2} \\ &= \langle \nabla h,\, \nabla x_k \rangle_{L^2} \\ &= -\langle \mathcal{D} h,\, x_k \rangle_{L^2} = \langle \tilde{c},\, x_k \rangle_{L^2} \end{aligned}$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Constant gain approximation for K is the minimizer obtained over all deterministic vectors:

$$\widehat{\mathsf{K}}^* := \arg\min_{\widehat{\mathsf{K}} \in \mathbb{R}^d} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$$

Solution is evidently the mean, $\widehat{\mathsf{K}}^* = \mathsf{E}[\mathsf{K}]$.

$$\begin{aligned}
\widehat{\mathsf{K}}_k^* &= \langle \mathsf{K}, e_k \rangle_{L^2} \\
&= \langle \nabla h, \nabla x_k \rangle_{L^2} \\
&= -\langle \mathcal{D} h, x_k \rangle_{L^2} = \langle \tilde{c}, x_k \rangle_{L^2}
\end{aligned}$$

Empirical approximation:

$$\widehat{\mathsf{K}}_k^* \approx \frac{1}{N} \sum_{i=1}^{N} [c(x^i) - \hat{c}] x_k^i$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Redefine the approximation to K as,

$$\nabla g = \widehat{\mathsf{K}}^* + \nabla \tilde{g}$$

Modified ERM with constaints is:

$$\tilde{g}^* := \arg\min_{\tilde{g} \in \mathcal{H}} \ \|\nabla h - \widehat{\mathsf{K}}^* - \nabla \tilde{g}\|_{L_2}^2$$

$$\text{s.t.} \quad \langle \partial_{x_k} \tilde{g}, 1 \rangle_{L_2} = 0, \quad 1 \leq k \leq d$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Redefine the approximation to K as,

$$\nabla g = \widehat{\mathsf{K}}^* + \nabla \tilde{g}$$

Modified ERM with constaints is:

$$\tilde{g}^* := \arg\min_{\tilde{g} \in \mathcal{H}} \ \|\nabla h - \widehat{\mathsf{K}}^* - \nabla \tilde{g}\|_{L_2}^2$$

$$\text{s.t.} \quad \langle \partial_{x_k} \tilde{g}, 1 \rangle_{L_2} = 0, \quad 1 \le k \le d$$

Solution obtained by finding a saddle point for the Lagrangian

$$L(\tilde{g}, \mu) := \|\nabla h - \widehat{\mathsf{K}}^* - \nabla \tilde{g}\|_{L_2}^2 + \langle \mu, \nabla \tilde{g} \rangle_{L_2}$$

where $\mu \in \mathbb{R}^d$ are the Lagrange multipliers.

Applications to Nonlinear filtering

∇-LSTD-RKHS-OM

Redefine the approximation to K as,

$$\nabla g = \widehat{\mathsf{K}}^* + \nabla \tilde{g}$$

Modified ERM with constaints is:

$$\tilde{g}^* := \operatorname*{arg\,min}_{\tilde{g} \in \mathcal{H}} \ \|\nabla h - \widehat{\mathsf{K}}^* - \nabla \tilde{g}\|_{L_2}^2$$

$$\text{s.t.} \quad \langle \partial_{x_k} \tilde{g}, 1 \rangle_{L_2} = 0, \quad 1 \le k \le d$$

Using $\mathcal{H}_N := \text{span}\{K_{x^j} : 1 \le j \le N\}$, β and μ can be obtained by solving $N + d$ linear equations

$$\mathsf{K} = \widehat{\mathsf{K}}^* + \nabla \tilde{g}^*$$

Applications to Nonlinear filtering

∇-LSTD-RKHS-memory

Gain updates are done at $t = n\delta$, where δ is the inter-sampling time.
Continuity: $\mathsf{K}_n = \mathsf{K}_{t_n} \approx \mathsf{K}_{t_{n-1}}$ if $\delta \approx 0$.

Applications to Nonlinear filtering

∇-LSTD-RKHS-memory

Gain updates are done at $t = n\delta$, where δ is the inter-sampling time.
Continuity: $\mathsf{K}_n = \mathsf{K}_{t_n} \approx \mathsf{K}_{t_{n-1}}$ if $\delta \approx 0$.
Adding a regularization term to the loss function:

$$g_n^* := \arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{j=1}^{N} L_n(x_n^j, g, \nabla g) + \lambda \|g\|_{\mathcal{H}}^2$$

$$L_n(x, g, \nabla g) := \|\nabla g(x)\|^2 - 2\tilde{c}_N(x) g(x) + \underbrace{\lambda_{mem} \|\nabla g(x) - \nabla g_{n-1}(x)\|^2}_{\text{continuity penalty}}$$

$$\beta_n^* = M^{-1} b$$

Application to nonlinear filtering

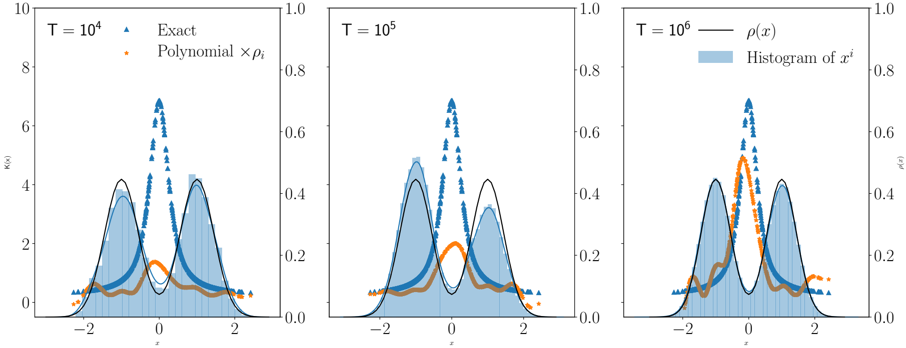
Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$T = 10^4, 10^5, 10^6$ with $\delta = 0.01$, $N = 1000$

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$T = 10^4, 10^5, 10^6$ with $\delta = 0.01$, $N = 1000$

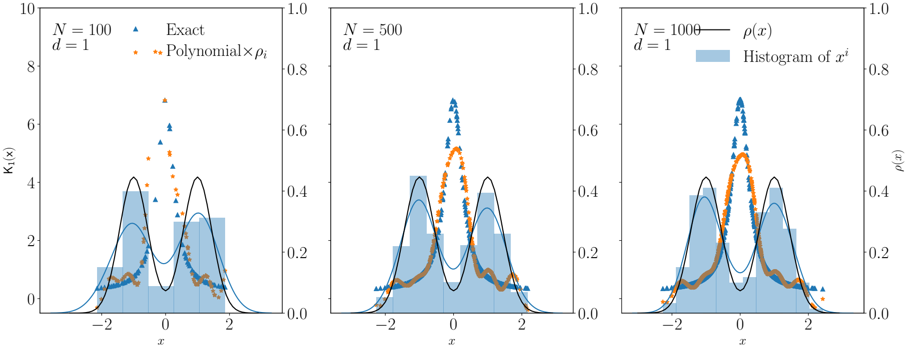


∇-LSTD with $\psi_i = x^i \rho_1(x)$, $\psi_{i+1} = x^i \rho_2(x)$ with $1 \leq i \leq 5$.

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

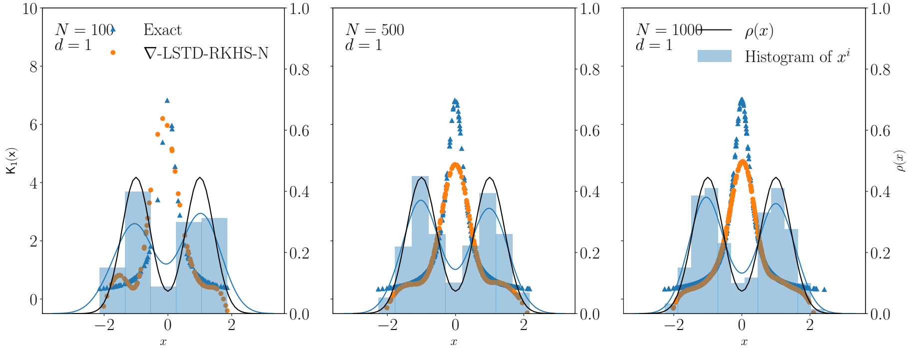


∇-LSTD-L with $\psi_i = x^i\rho_1(x)$, $\psi_{i+1} = x^i\rho_2(x)$ with $1 \le i \le 5$.

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

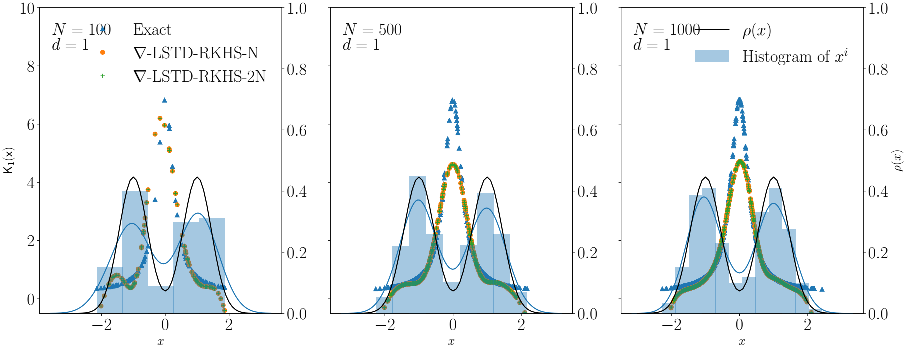


∇-LSTD-RKHS-N with Gaussian kernel, $\varepsilon = 0.1$ and $\lambda = 10^{-2}$ (best)

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

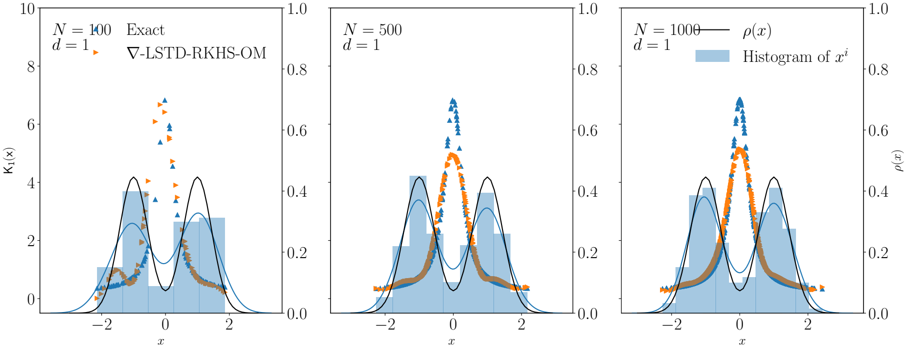


∇-LSTD-RKHS-2N with Gaussian kernel, $\varepsilon = 0.1$ and $\lambda = 10^{-2}$ (best)

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

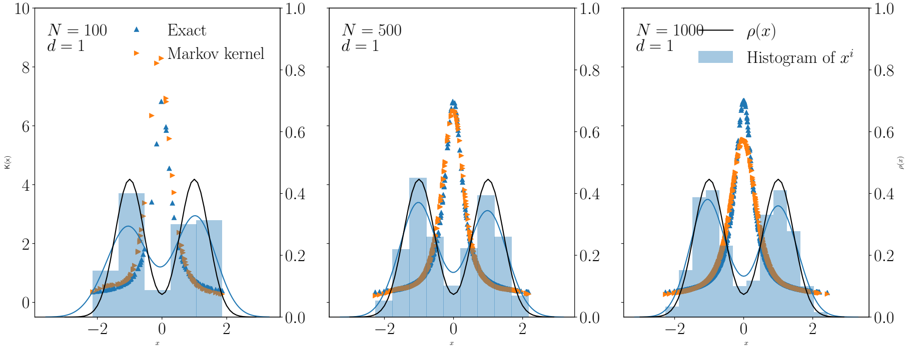


∇-LSTD-RKHS-OM with Gaussian kernel, $\varepsilon = 0.1$ and $\lambda = 10^{-2}$ (best)

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

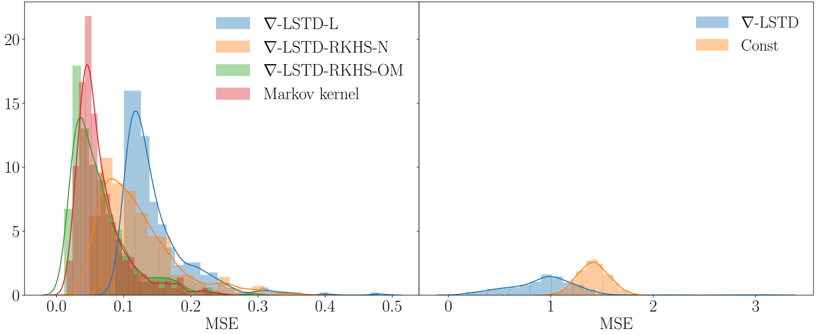


Markov kernel approximation [Taghvaei et al. 18] with $\epsilon = 0.1$ (best)

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: For a fixed t, ρ_t a Gaussian mixture
$c(x) \equiv x$, $d = 1$
$N = 100, 500, 1000$

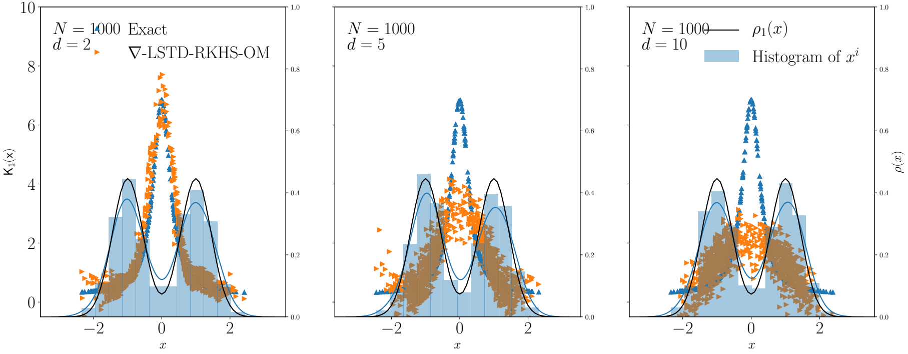


Histogram of MSEs obtained from 1000 independent trials

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: $\rho(x) = \prod_{k=1}^{d} \rho_k(x_k)$, each ρ_k a Gaussian mixture
$c(x) = C^T x$, where $C = \mathbb{I}_d$
$d = 2, 5, 10, \quad N = 1000$

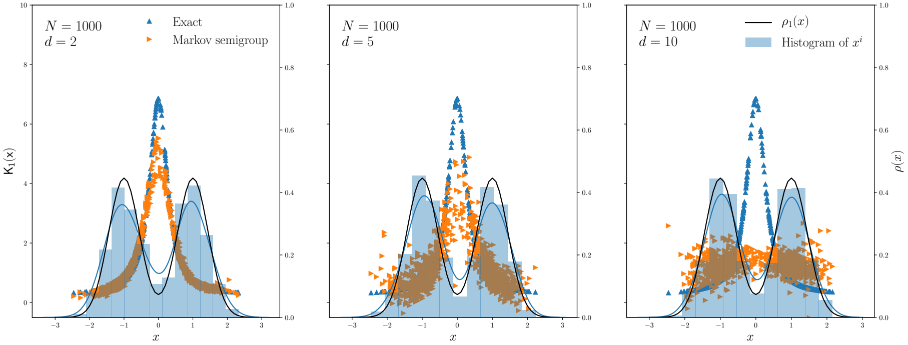


∇-LSTD-RKHS-OM

Application to nonlinear filtering

Numerical example - Gain approximation for a fixed t

Example: $\rho(x) = \prod_{k=1}^{d} \rho_k(x_k)$, each ρ_k a Gaussian mixture
$c(x) = C^T x$, where $C = \mathbb{I}_d$
$d = 2, 5, 10, \quad N = 1000$



Markov kernel approximation

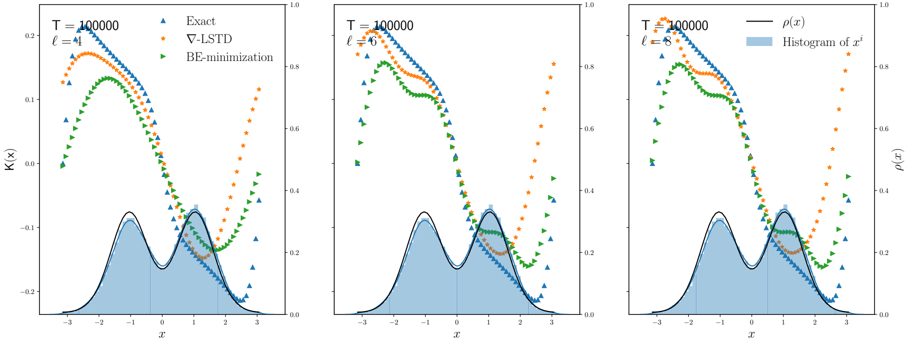
Application to nonlinear filtering

Numerical example - Gain for a nonlinear oscillator model

Example: $\rho(x)$ is a mixture of von Mises densities on a circle

$$\mathrm{d}\vartheta = \omega \mathrm{d}t + \sigma_B \mathrm{d}B_t \mod 2\pi,$$

$$\mathrm{d}Z_t = c(\vartheta)\mathrm{d}t + \sigma_W \mathrm{d}W_t, \quad c(\vartheta) = \frac{1}{2}[1+\cos(\vartheta)]$$

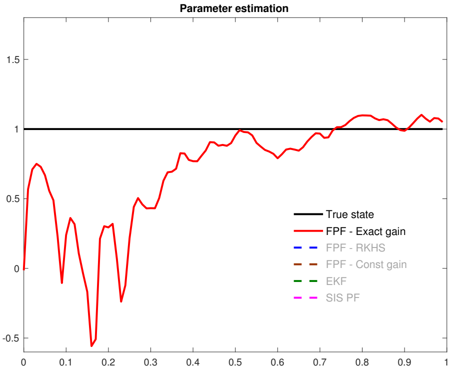


∇-LSTD-L with $\psi_i = \sin(ix), \psi_{i+1} = \cos(ix)$ with $1 \le i \le \ell/2$, $\ell = 4, 6, 8$.

Applications to Nonlinear filtering

Numerical example - Parameter estimation

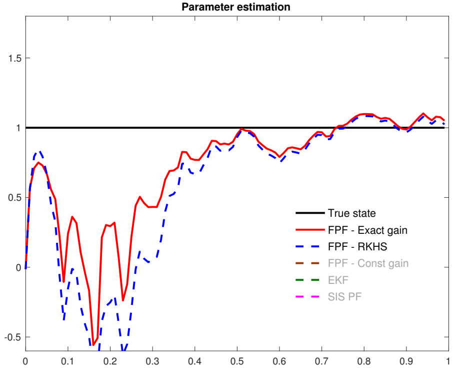
Example: Parameter estimation with bimodal prior
Observations: parameter plus additive noise with $\sigma_W = 1$.



Applications to Nonlinear filtering

Numerical example - Parameter estimation

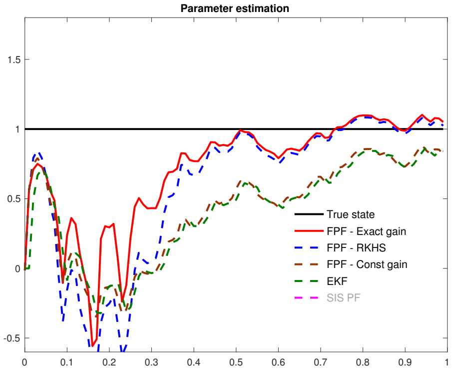
Example: Parameter estimation with bimodal prior
Observations: parameter plus additive noise with $\sigma_W = 1$.



Applications to Nonlinear filtering

Numerical example - Parameter estimation

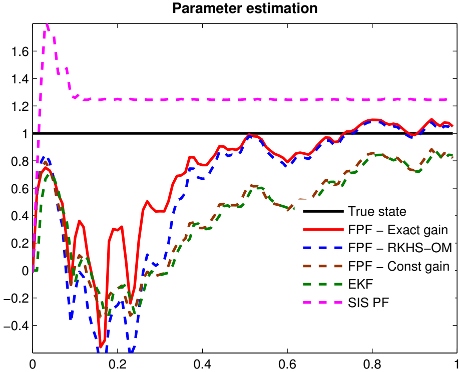
Example: Parameter estimation with bimodal prior
Observations: parameter plus additive noise with $\sigma_W = 1$.



Applications to Nonlinear filtering

Numerical example - Parameter estimation

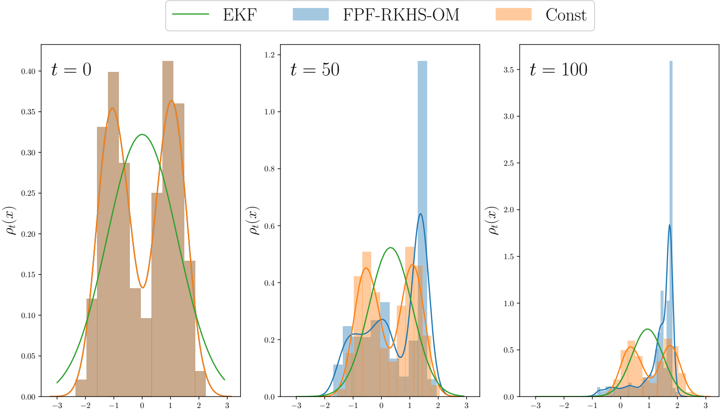
Example: Parameter estimation with bimodal prior
Observations: parameter plus additive noise with $\sigma_W = 1$.



Applications to Nonlinear filtering

Numerical example - Parameter estimation

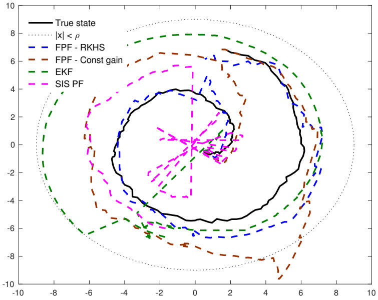
Example: Parameter estimation with bimodal prior
Observations: parameter plus additive noise with $\sigma_W = 1$.



Applications to Nonlinear filtering

Numerical example - Nonlinear $2d$ ship dynamics model

Example: Nonlinear ship dynamics model in $2d$.
Observations: $c(x) = \arctan(x_1/x_2)$ with std. deviation $\approx 18°$.



Filter	Σ_1	Lost track
RKHS-OM	0.90	4
RKHS-mem.	0.91	7
Const.	1.30	14
SIR PF	3.14	57
EKF	6.52	93

Applications to MCMC

Introduction to MCMC

In many applications, we need to compute

$$\eta = \int c(x)\rho(x)\,\mathrm{d}x$$

- $c\colon \mathbb{R}^\ell \to \mathbb{R}$ is a measurable function.
- ρ is a target probability density in $\mathbb{R}^\ell$.

Markov-Chain Monte Carlo (MCMC) methods provide numerical algorithms to obtain estimates:

$$\eta_t = \frac{1}{t}\int_0^t c(\boldsymbol{\Phi}(s))\,ds$$

$\boldsymbol{\Phi}$ is a Markov process with steady state distribution ρ.

Applications to MCMC

Asymptotic Variance

Estimates η_t obey a Central Limit Theorem,

$$\sqrt{t}(\eta_t - \eta) \xrightarrow{d} N(0, \gamma^2)$$

Representation in terms of h [see eg: Glynn & Meyn 96]:

$$\gamma^2 = 2\langle h, \tilde{c} \rangle$$

Applications to MCMC

Asymptotic Variance

Estimates η_t obey a Central Limit Theorem,

$$\sqrt{t}(\eta_t - \eta) \xrightarrow{d} N(0, \gamma^2)$$

Representation in terms of h [see eg: Glynn & Meyn 96]:

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

(For Langevin diffusion)

Applications to MCMC

Control Variates

Goal: To minimize asymptotic variance.

Idea: Modify the estimator using control variates [Henderson 97, CTCN[1]]

$$c_g = c + \underbrace{\mathcal{D}g}_{\text{Control variate}} \quad , \quad \textit{where} \quad g \in \mathcal{H}$$

$$\eta_t^g = \frac{1}{t}\int_0^t c_g(\Phi_s)\, ds$$

For any $g \in C^2 \cap L^2(\rho)$, $\langle \mathcal{D}g,\, 1\rangle_{L^2} = 0$.

[1] Control Techniques for Complex Networks, S.Meyn

Applications to MCMC

Optimal control variates

$$\mathcal{D}(h-g) = -c_g + \eta$$

$h - g$ is the solution to Poisson's equation with $\tilde{c}_g$ as the forcing function.

Applications to MCMC

Optimal control variates

$$\mathcal{D}(h-g) = -c_g + \eta$$

$h - g$ is the solution to Poisson's equation with $\tilde{c}_g$ as the forcing function.

Asymptotic variance of the new estimator:

$$\gamma_g^2 = 2\langle h-g, \tilde{c}_g\rangle_{L^2}, \quad \tilde{c}_g = c_g - \eta$$

Applications to MCMC

Optimal control variates

$$\mathcal{D}(h - g) = -c_g + \eta$$

$h - g$ is the solution to Poisson's equation with $\tilde{c}_g$ as the forcing function.

Asymptotic variance of the new estimator:

$$\begin{aligned} \gamma_g^2 &= 2\langle h - g, \tilde{c}_g \rangle_{L^2}, \quad \tilde{c}_g = c_g - \eta \\ &= 2\|\nabla h - \nabla g\|_{L^2}^2, \text{ (For Langevin diffusion)} \end{aligned}$$

Applications to MCMC

Optimal control variates

$$\mathcal{D}(h-g) = -c_g + \eta$$

$h-g$ is the solution to Poisson's equation with $\tilde{c}_g$ as the forcing function.

Asymptotic variance of the new estimator:

$$\begin{aligned}\gamma_g^2 &= 2\langle h-g, \tilde{c}_g\rangle_{L^2}, \quad \tilde{c}_g = c_g - \eta \\ &= 2\|\nabla h - \nabla g\|_{L^2}^2, \text{ (For Langevin diffusion)}\end{aligned}$$

Can be minimized using ∇-LSTD algorithms.

Applications to MCMC

Numerical Examples - Variance v Asymptotic variance

Variance v Asymptotic variance

$$\sigma^2 = \langle \tilde{c}, \tilde{c} \rangle_{L^2} = R(0)$$ - Variance

Applications to MCMC

Numerical Examples - Variance v Asymptotic variance

Variance v Asymptotic variance

$$\sigma^2 = \langle \tilde{c}, \tilde{c} \rangle_{L^2} = R(0) \qquad \text{- Variance}$$

$$\gamma^2 = 2\langle h, \tilde{c} \rangle_{L^2} = \int_{-\infty}^{\infty} R(s)ds \qquad \text{- Asymptotic variance}$$

Applications to MCMC

Numerical Examples - Variance v Asymptotic variance

Variance v Asymptotic variance

$$\sigma^2 = \langle \tilde{c}, \tilde{c} \rangle_{L^2} = R(0) \qquad \text{- Variance}$$

$$\gamma^2 = 2\langle h, \tilde{c} \rangle_{L^2} = \int_{-\infty}^{\infty} R(s)ds \qquad \text{- Asymptotic variance}$$

Minimizing σ^2 is easier than minimizing γ^2 (ZV method [Papamarkou 14])

But does minimizing $\sigma^2 \implies$ minimizing γ^2 ?

Applications to MCMC

Numerical Examples - Variance v Asymptotic variance

Variance v Asymptotic variance

$$\sigma^2 = \langle \tilde{c}, \tilde{c} \rangle_{L^2} = R(0) \qquad \text{- Variance}$$

$$\gamma^2 = 2\langle h, \tilde{c} \rangle_{L^2} = \int_{-\infty}^{\infty} R(s)ds \qquad \text{- Asymptotic variance}$$

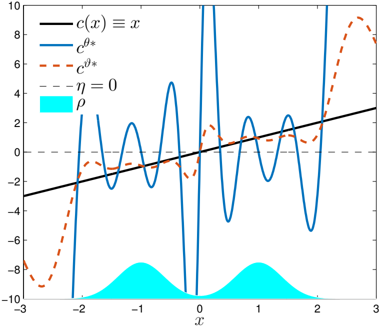
Minimizing σ^2 is easier than minimizing γ^2 (ZV method [Papamarkou 14])

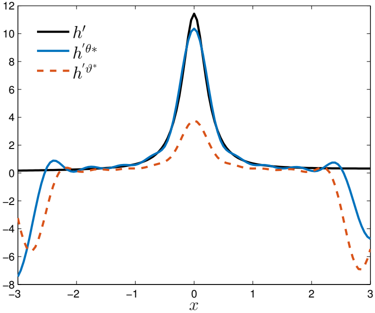
But does minimizing $\sigma^2 \implies$ minimizing γ^2 ? NO !

Applications to MCMC

Numerical Examples - Variance vs Asymptotic variance

Example: Unadjusted Langevin algorithm (ULA)
$c(x) \equiv x$



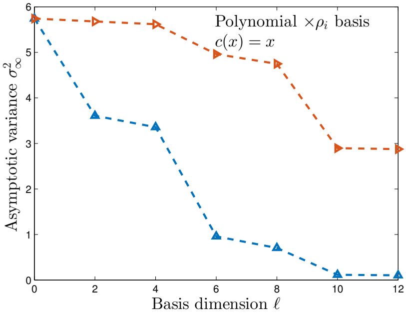


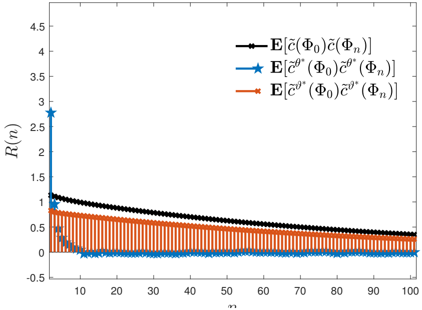
Applications to MCMC

Numerical Examples - Variance vs Asymptotic variance

Example: Unadjusted Langevin algorithm (ULA)

$c(x) \equiv x$

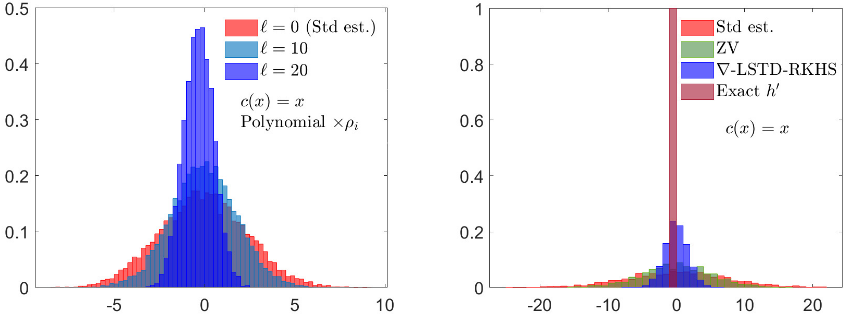




Applications to MCMC

Numerical Examples - ULA

Example: Unadjusted Langevin algorithm (ULA)
$c(x) \equiv x,\ \rho \sim 0.5\mathcal{N}(-1, 0.4472) + 0.5\mathcal{N}(1, 0.4472)$



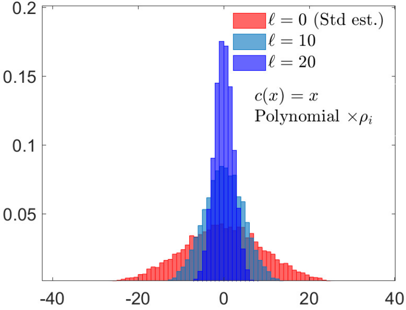
Applications to MCMC

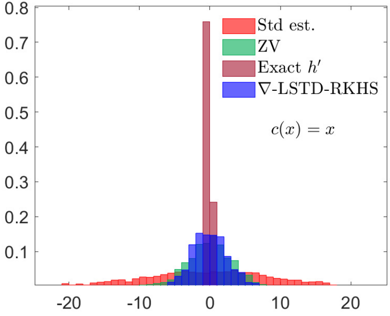
Numerical Examples - RWM

Example: Random walk Metropolis (RWM)

$c(x) \equiv x$, $\rho \sim 0.5\mathcal{N}(-1, 0.4472) + 0.5\mathcal{N}(1, 0.4472)$

Theoretical justification based on [Brosse et al. 19]

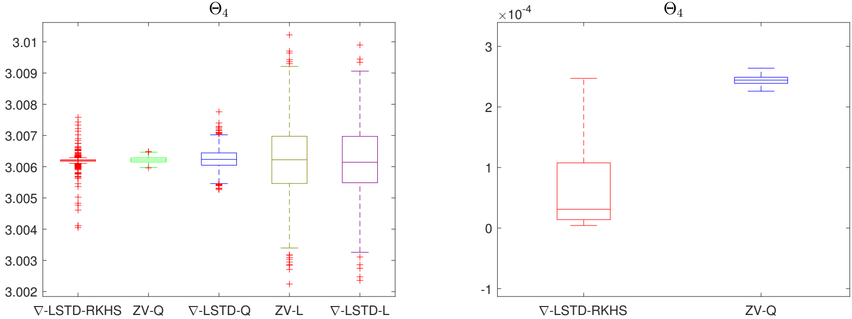




Applications to MCMC

Numerical Examples - Logistic Regression with RWM sampling

Example: Logistic Regression for Swiss bank notes
RWM sampling



Box plots of estimates of Θ_4.

Conclusions

- Differential LSTD learning algorithms to approximate solution to Poisson's equation for the Langevin diffusion.
  - RKHS based approaches solve the basis selection problem and enable easy extensions to higher dimensions.
  - Optimal mean gain is a useful design tool and makes the algorithm more robust to ε and λ.
- Two interesting applications
  - Gain function approximation in feedback particle filter.
  - Asymptotic variance reduction in MCMC algorithms.
- Recent research extended to include reversible Markov chains.

Future Work

Analysis:

- Error analysis of the ∇-LSTD-RKHS method, which could lead to proper choices of hyper parameters.
- A more thorough comparison of the various gain approximation algorithms.
- Nagging question: Why reduced complexity solution is as good as the optimal?

Future Work

Analysis:

- Error analysis of the ∇-LSTD-RKHS method, which could lead to proper choices of hyper parameters.
- A more thorough comparison of the various gain approximation algorithms.
- Nagging question: Why reduced complexity solution is as good as the optimal?

Algorithm:

- ∇-LSTD-RKHS algorithm with a differential regularizer. This is limited by the scope of representer theorem.
- An algorithm based on semiparametric representer theorem, that allows the use of additional parameterized functions.

Future Work

Analysis:

- Error analysis of the ∇-LSTD-RKHS method, which could lead to proper choices of hyper parameters.
- A more thorough comparison of the various gain approximation algorithms.
- Nagging question: Why reduced complexity solution is as good as the optimal?

Algorithm:

- ∇-LSTD-RKHS algorithm with a differential regularizer. This is limited by the scope of representer theorem.
- An algorithm based on semiparametric representer theorem, that allows the use of additional parameterized functions.

Applications:

- Real time filtering problem - Potential application to battery SOC estimation is being explored currently.
- Application of FPF to control in the context of POMDPs.

Future Work

- Create algorithms whose complexity does not grow with dimension.
- A tractable solution to the ERM with regularized gradient may also lead to more efficient algorithms.
- Solidarity with the prior work will require the extension of the RKHS theory to the case of self-adjoint kernels that are not symmetric.
- Research and large scale testing is required for nonlinear filtering and MCMC applications.

References

A. Radhakrishnan, A. Devraj and S. Meyn, "Learning techniques for feedback particle filter design," *2016 IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016.

A. Radhakrishnan, S. Meyn, "Feedback particle filter design using a differential-loss reproducing kernel Hilbert space," *2018 American Control Conference (ACC)*, Milwaukee, WI, 2018.

A. Radhakrishnan, S. Meyn, "Gain function tracking in the feedback particle filter," *2019 American Control Conference (ACC)*, Philadelphia, PA, 2019.

S.P.Meyn, "Control Techniques for Complex Networks", *Cambridge University Press*, Dec 2007.

S. Henderson. *Variance Reduction Via an Approximating Markov Process*. PhD thesis, Stanford University, Stanford, California, 1997.

T. Yang, P. G. Mehta and S. P. Meyn, "Feedback Particle Filter," in IEEE Transactions on Automatic Control, vol. 58, no. 10, pp. 2465-2480, Oct. 2013.

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

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

Thank You!

Questions?