Computationally Efficient Implementation of the Feedback Particle Filter Algorithm in High Dimensions

CSE Annual Meeting: 2017 Fellows Symposium

Amirhossein Taghvaei
Joint work with P. G. Mehta, R. S. Laugesen, and S. P. Meyn

Support from CSE fellowship award is gratefully acknowledged

Apr 26, 2017

ILLINOIS

Outline

Definition:

Classical Poisson equation: $-\Delta\phi = h, \quad \text{on} \quad \mathbb{R}^d$

Laplacian: $\Delta\psi := \nabla \cdot (\nabla\psi)$

- $\rho : \mathbb{R}^d \to \mathbb{R}^+$ (prob. density)
- $h : \mathbb{R}^d \to \mathbb{R}$ (given),
- $\phi : \mathbb{R}^d \to \mathbb{R}$ (unknown)

Problem: Design a computational algorithm

R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. *SIAM*, 2015

Problem Statement
Weighted Poisson equation

Definition:

Classical Poisson equation: $-\Delta\phi = h, \quad \text{on} \quad \mathbb{R}^d$

Laplacian: $\Delta\psi := \nabla\cdot(\nabla\psi)$

- $\rho : \mathbb{R}^d \to \mathbb{R}^+$ (prob. density)
- $h : \mathbb{R}^d \to \mathbb{R}$ (given),
- $\phi : \mathbb{R}^d \to \mathbb{R}$ (unknown)

Problem: Design a computational algorithm

R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. *SIAM*, 2015

Definition:

Classical Poisson equation: $-\Delta\phi = h, \quad \text{on} \quad \mathbb{R}^d$

Weighted Laplacian: $\Delta_\rho \psi := \frac{1}{\rho}\nabla\cdot(\rho\nabla\psi)$

- $\rho : \mathbb{R}^d \to \mathbb{R}^+$ (prob. density)
- $h : \mathbb{R}^d \to \mathbb{R}$ (given),
- $\phi : \mathbb{R}^d \to \mathbb{R}$ (unknown)

Problem: Design a computational algorithm

R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. *SIAM*, 2015

Problem Statement

Weighted Poisson equation

Definition:

Weighted Poisson equation: $-\Delta_\rho \phi = h - \hat{h}, \quad \text{on} \quad \mathbb{R}^d$

Weighted Laplacian: $\Delta_\rho \psi := \frac{1}{\rho} \nabla \cdot (\rho \nabla \psi)$

- $\rho : \mathbb{R}^d \to \mathbb{R}^+$ (prob. density)
- $h : \mathbb{R}^d \to \mathbb{R}$ (given), $\hat{h} := \int h(x)\rho(x)\,\mathrm{d}x$
- $\phi : \mathbb{R}^d \to \mathbb{R}$ (unknown)

Problem: Design a computational algorithm

R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. *SIAM*, 2015

Problem Statement
Weighted Poisson equation

Definition:

$$\textbf{Weighted Poisson equation:} \quad -\Delta_\rho \phi = h - \hat{h}, \quad \text{on} \quad \mathbb{R}^d$$

$$\textbf{Weighted Laplacian:} \quad \Delta_\rho \psi := \frac{1}{\rho} \nabla \cdot (\rho \nabla \psi)$$

- $\rho : \mathbb{R}^d \to \mathbb{R}^+$ (prob. density)
- $h : \mathbb{R}^d \to \mathbb{R}$ (given), $\hat{h} := \int h(x)\rho(x)\,\mathrm{d}x$
- $\phi : \mathbb{R}^d \to \mathbb{R}$ (unknown)

Problem: Design a computational algorithm

$$\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \phi^N$$

Such that $\phi^N \to \phi$ as $N \to \infty$

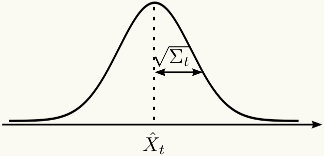
R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. *SIAM*, 2015

Outline

Motivation: Nonlinear filtering
Feedback Particle Filter

Kalman Filter

Linear system
Posterior is Gaussian $N(\hat{X}_t, \Sigma_t)$



$$\mathrm{d}\hat{X}_t = \underbrace{\ldots}_{\text{Propagation}} + \underbrace{\mathsf{K}_t\,\mathrm{d}I_t}_{\text{Correction}}$$

K_t is the Kalman gain

Feedback Particle Filter

Nonlinear system
Posterior $\approx$ empirical dist. $\{X^1, \ldots, X^N\}$,

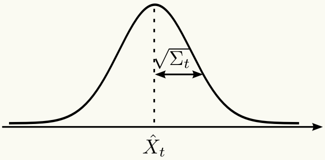
$$\mathrm{d}X_t^i = \underbrace{\ldots}_{\text{Propagation}} + \underbrace{\mathsf{K}_t(X_t^i) \circ \mathrm{d}I_t^i}_{\text{Correction}}$$

$\boxed{\mathsf{K}_t = \nabla\phi \text{ from Poisson eq.}}$

T. Yang, P. G. Mehta, and S. P. Meyn. feedback particle filter, *TAC*, 2013
T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn. Multivariable feedback particle filter, *Automatica*, 2016

Kalman Filter

Linear system
Posterior is Gaussian $N(\hat{X}_t, \Sigma_t)$

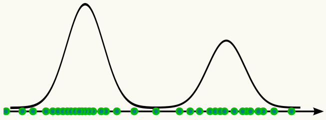


$$\mathrm{d}\hat{X}_t = \underbrace{\ldots}_{\text{Propagation}} + \underbrace{\mathsf{K}_t\,\mathrm{d}I_t}_{\text{Correction}}$$

K_t is the Kalman gain

Feedback Particle Filter

Nonlinear system
Posterior $\approx$ empirical dist. $\{X^1, \ldots, X^N\}$,



$$\mathrm{d}X_t^i = \underbrace{\ldots}_{\text{Propagation}} + \underbrace{\mathsf{K}_t(X_t^i) \circ \mathrm{d}I_t^i}_{\text{Correction}}$$

$\boxed{\mathsf{K}_t = \nabla\phi \text{ from Poisson eq.}}$

T. Yang, P. G. Mehta, and S. P. Meyn. feedback particle filter, *TAC*, 2013
T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn. Multivariable feedback particle filter, *Automatica*, 2016

Motivation: Other applications

Stochastic analysis:

- Simulation and optimization theory for Markov models [S. Meyn, R. Tweedie, 2012]

Statistical learning:

- Nonlinear dimensionality reduction [M. Belkin, 2003]
- Diffusion maps [R. Coifman, S. Lafon, 2006]
- Spectral clustering [M. Hein, et. al. 2006]

Transporting densities:

- Optimal transportation [Villani, 2003]

Global optmization:

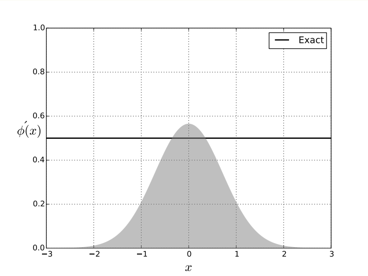
- A Controlled Particle Filter for Global Optimization [C. Zhang, et. al. 2017]

Outline

Problem Challenges

(Easy case)

Gaussian distribution
linear h



$\nabla\phi(x) = \text{constant}$ (Kalman gain)

Challenges:

- Multi scale
- Unknown underlying distribution

(Difficult case)

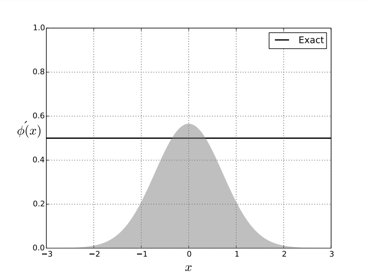
Bimodal distribution
linear h

$\nabla\phi(x) = \ldots$ (Nonlinear gain)

Problem Challenges

(Easy case)

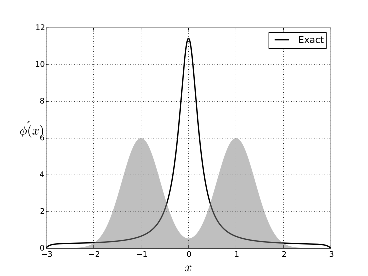
Gaussian distribution
linear h



$\nabla\phi(x) = \text{constant}$ (Kalman gain)

(Difficult case)

Bimodal distribution
linear h



$\nabla\phi(x) = \ldots$ (Nonlinear gain)

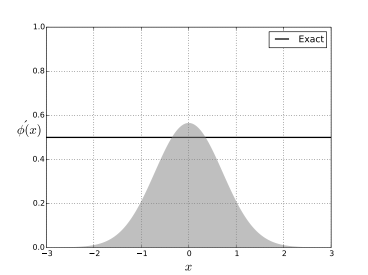
Challenges:

- Multi scale
- Unknown underlying distribution

Problem Challenges

(Easy case)

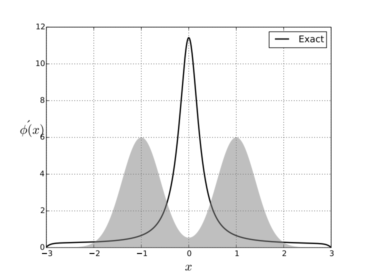
Gaussian distribution
linear h



$\nabla\phi(x) = \text{constant}$ (Kalman gain)

(Difficult case)

Bimodal distribution
linear h



$\nabla\phi(x) = \ldots$ (Nonlinear gain)

Challenges:

- Multi scale
- Unknown underlying distribution

Outline

Problem summary:

$$-\Delta_\rho \phi = h - \hat{h}$$

$$\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \phi^N$$

Two solution approaches:

(I) PDE

- Theory of elliptic operators
- Weak formulation
- Approximation via projection
- Solve a system of linear equations

(Galerkin algorithm)

(II) Stochastic

- Generator of Markov process
- Fixed pt formulation using semigroup
- Approximation via kernel
- Solve the fixed pt problem iteratively

(kernel-based algorithm)

Solution Approach
Two viewpoints

Problem summary:

$$-\Delta_\rho \phi = h - \hat{h}$$

$$\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \phi^N$$

Two solution approaches:

(I) PDE

- Theory of elliptic operators
- Weak formulation
- Approximation via projection
- Solve a system of linear equations

(Galerkin algorithm)

(II) Stochastic

- Generator of Markov process
- Fixed pt formulation using semigroup
- Approximation via kernel
- Solve the fixed pt problem iteratively

(kernel-based algorithm)

Solution Approach
Two viewpoints

Problem summary:

$$-\Delta_\rho \phi = h - \hat{h}$$

$$\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \phi^N$$

Two solution approaches:

(I) PDE

- Theory of elliptic operators
- Weak formulation
- Approximation via projection
- Solve a system of linear equations

(Galerkin algorithm)

(II) Stochastic

- Generator of Markov process
- Fixed pt formulation using semigroup
- Approximation via kernel
- Solve the fixed pt problem iteratively

(kernel-based algorithm)

Solution Approach
Two viewpoints

Problem summary:

$$-\Delta_\rho \phi = h - \hat{h}$$

$$\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \phi^N$$

Two solution approaches:

(I) PDE

- Theory of elliptic operators
- Weak formulation
- Approximation via projection
- Solve a system of linear equations

(Galerkin algorithm)

(II) Stochastic

- Generator of Markov process
- Fixed pt formulation using semigroup
- Approximation via kernel
- Solve the fixed pt problem iteratively

(kernel-based algorithm)

Outline

Galerkin Algorithm
Concept

Strong form:

$$-\Delta_\rho \phi = h - \hat{h}$$

Weak form:

$$\langle \nabla\phi, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in H^1(\mathbb{R}^d, \rho)$$

where $\langle f, g \rangle := \int f(x)g(x)\rho(x)\,\mathrm{d}x$

Galerkin approximation:

$$\langle \nabla\phi^{(M)}, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in S$$

where $S = \text{span}\{\psi_1, \ldots, \psi_M\}$

Empirical approximation

$$\frac{1}{N}\sum_{i=1}^{N} \nabla\phi^{(M,N)}(X^i) \cdot \nabla\psi(X^i) = \frac{1}{N}\sum_{i=1}^{N}(h(X^i) - \hat{h})\psi(X^i), \quad \forall \psi \in S$$

where $X^i \sim \rho$

Strong form:

$$-\Delta_\rho \phi = h - \hat{h}$$

Weak form:

$$\langle \nabla\phi, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in H^1(\mathbb{R}^d, \rho)$$

where $\langle f, g \rangle := \int f(x) g(x) \rho(x) \, \mathrm{d}x$

Galerkin approximation:

$$\langle \nabla\phi^{(M)}, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in S$$

where $S = \text{span}\{\psi_1, \ldots, \psi_M\}$

Empirical approximation

$$\frac{1}{N}\sum_{i=1}^{N} \nabla\phi^{(M,N)}(X^i) \cdot \nabla\psi(X^i) = \frac{1}{N}\sum_{i=1}^{N} (h(X^i) - \hat{h})\psi(X^i), \quad \forall \psi \in S$$

where $X^i \sim \rho$

Strong form:

$$-\Delta_\rho \phi = h - \hat{h}$$

Weak form:

$$\langle \nabla\phi, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in H^1(\mathbb{R}^d, \rho)$$

where $\langle f, g \rangle := \int f(x)g(x)\rho(x)\,\mathrm{d}x$

Galerkin approximation:

$$\langle \nabla\phi^{(M)}, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in S$$

where $S = \text{span}\{\psi_1, \ldots, \psi_M\}$

Empirical approximation

$$\frac{1}{N}\sum_{i=1}^{N} \nabla\phi^{(M,N)}(X^i) \cdot \nabla\psi(X^i) = \frac{1}{N}\sum_{i=1}^{N} (h(X^i) - \hat{h})\psi(X^i), \quad \forall \psi \in S$$

where $X^i \sim \rho$

Strong form:

$$-\Delta_\rho \phi = h - \hat{h}$$

Weak form:

$$\langle \nabla\phi, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in H^1(\mathbb{R}^d, \rho)$$

where $\langle f, g \rangle := \int f(x) g(x) \rho(x) \, \mathrm{d}x$

Galerkin approximation:

$$\langle \nabla\phi^{(M)}, \nabla\psi \rangle = \langle h - \hat{h}, \psi \rangle, \quad \forall \psi \in S$$

where $S = \text{span}\{\psi_1, \ldots, \psi_M\}$

Empirical approximation

$$\frac{1}{N} \sum_{i=1}^{N} \nabla\phi^{(M,N)}(X^i) \cdot \nabla\psi(X^i) = \frac{1}{N} \sum_{i=1}^{N} (h(X^i) - \hat{h}) \psi(X^i), \quad \forall \psi \in S$$

where $X^i \sim \rho$

Input: $\underbrace{\{\psi_1, \ldots, \psi_M\}}_{\text{basis functions}}$, $\{X^1, \ldots, X^N\}$, $\{h(X^1), \ldots, h(X^N)\}$

Output: Approximate solution $\phi^{M,N}$

1. Compute the matrix $A \in \mathbb{R}^{M \times M}$ and $b \in \mathbb{R}^M$:

$$A_{ml} = \frac{1}{N} \sum_{i=1}^{N} \nabla \psi_m(X^i) \cdot \nabla \psi_l(X^i)$$

$$b_m = \frac{1}{N} \sum_{i=1}^{N} \psi_m(X^i) h(X^i) - \hat{h})$$

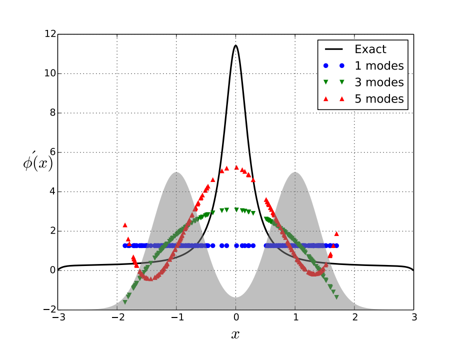
2. Solve for $c \in \mathbb{R}^M$:

$$c = A^{-1} b$$

3. Express the approximate solution as

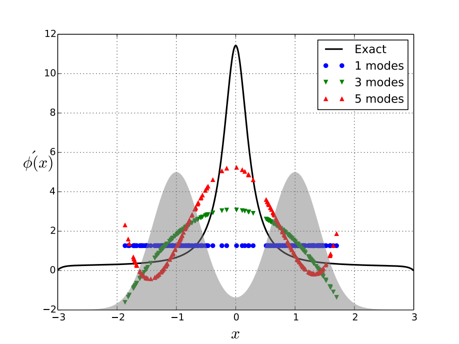
$$\phi^{(M,N)}(x) = \sum_{m=1}^{M} c_m \psi_m(x)$$

Galerkin Algorithm
Numerical result



Issues:

- Choice of basis functions
- Numerical instability (the matrix A may become singular)
- Does not scale well with dimension (inverting a $M \times M$ matrix)



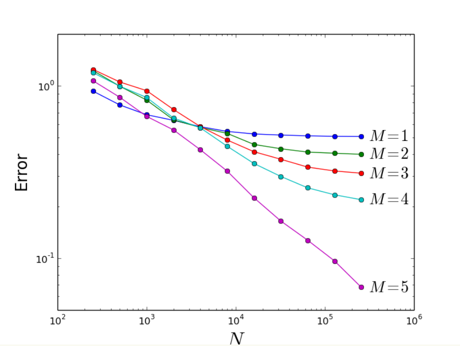
Issues:

- Choice of basis functions
- Numerical instability (the matrix A may become singular)
- Does not scale well with dimension (inverting a $M \times M$ matrix)

Galerkin Algorithm
Error analysis

Special case: The basis functions are eigenfunctions of Δ_ρ

$$\underbrace{\mathsf{E}\left[\|\nabla\phi - \nabla\phi^{(M,N)}\|_{L^2}\right]}_{\text{Total error}} \leq \underbrace{\frac{1}{\sqrt{\lambda_M}}\|h - \Pi_S h\|_{L^2}}_{\text{Bias}} + \underbrace{\frac{1}{\sqrt{N}}\|h\|_\infty \sqrt{\sum_{m=1}^{M}\frac{1}{\lambda_m}}}_{\text{Variance}}$$



Outline

Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

Semigroup identity: $e^{\epsilon \Delta_\rho} = I + \int_0^\epsilon e^{s\Delta_\rho} \Delta_\rho \, \mathrm{d}s$

Semigroup formulation:

$$\phi = e^{\epsilon \Delta_\rho} \phi + \tilde{h}$$

where $\tilde{h} := \int_0^\epsilon e^{s\Delta_\rho}(h - \hat{h}) \, \mathrm{d}s$

Kernel representation:

$$\phi(x) = \int \tilde{k}_\epsilon(x, y) \phi(y) \rho(y) \, \mathrm{d}y + \tilde{h}(x)$$

Empirical approximation:

$$\phi(x) = \frac{1}{N} \sum_{i=1}^{N} \tilde{k}_\epsilon(x, X^i) \phi(X^i) + \tilde{h}(x)$$

But $\tilde{k}_\epsilon(x, y) = ?$

Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

Semigroup identity: $e^{\epsilon \Delta_\rho} = I + \int_0^\epsilon e^{s\Delta_\rho} \Delta_\rho \, \mathrm{d}s$

Semigroup formulation:

$$\phi = e^{\epsilon \Delta_\rho} \phi + \tilde{h}$$

where $\tilde{h} := \int_0^\epsilon e^{s\Delta_\rho} (h - \hat{h}) \, \mathrm{d}s$

Kernel representation:

$$\phi(x) = \int \tilde{k}_\epsilon(x,y) \phi(y) \rho(y) \, \mathrm{d}y + \tilde{h}(x)$$

Empirical approximation:

$$\phi(x) = \frac{1}{N} \sum_{i=1}^N \tilde{k}_\epsilon(x, X^i) \phi(X^i) + \tilde{h}(x)$$

But $\tilde{k}_\epsilon(x,y) = ?$

Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

Semigroup identity: $e^{\epsilon \Delta_\rho} = I + \int_0^\epsilon e^{s\Delta_\rho} \Delta_\rho \, \mathrm{d}s$

Semigroup formulation:

$$\phi = e^{\epsilon \Delta_\rho} \phi + \tilde{h}$$

where $\tilde{h} := \int_0^\epsilon e^{s\Delta_\rho} (h - \hat{h}) \, \mathrm{d}s$

Kernel representation:

$$\phi(x) = \int \tilde{k}_\epsilon(x,y) \phi(y) \rho(y) \, \mathrm{d}y + \tilde{h}(x)$$

Empirical approximation:

$$\phi(x) = \frac{1}{N} \sum_{i=1}^{N} \tilde{k}_\epsilon(x, X^i) \phi(X^i) + \tilde{h}(x)$$

But $\tilde{k}_\epsilon(x,y) = ?$

Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

Semigroup identity: $e^{\epsilon \Delta_\rho} = I + \int_0^\epsilon e^{s\Delta_\rho} \Delta_\rho \, \mathrm{d}s$

Semigroup formulation:

$$\phi = e^{\epsilon \Delta_\rho} \phi + \tilde{h}$$

where $\tilde{h} := \int_0^\epsilon e^{s\Delta_\rho} (h - \hat{h}) \, \mathrm{d}s$

Kernel representation:

$$\phi(x) = \int \tilde{k}_\epsilon(x,y) \phi(y) \rho(y) \, \mathrm{d}y + \tilde{h}(x)$$

Empirical approximation:

$$\phi(x) = \frac{1}{N} \sum_{i=1}^{N} \tilde{k}_\epsilon(x, X^i) \phi(X^i) + \tilde{h}(x)$$

But $\tilde{k}_\epsilon(x,y) = ?$

Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

Semigroup identity: $e^{\epsilon\Delta_\rho} = I + \int_0^\epsilon e^{s\Delta_\rho}\Delta_\rho \, \mathrm{d}s$

Semigroup formulation:

$$\phi = e^{\epsilon\Delta_\rho}\phi + \tilde{h}$$

where $\tilde{h} := \int_0^\epsilon e^{s\Delta_\rho}(h - \hat{h})\, \mathrm{d}s$

Kernel representation:

$$\phi(x) = \int \tilde{k}_\epsilon(x,y)\phi(y)\rho(y)\, \mathrm{d}y + \tilde{h}(x)$$

Empirical approximation:

$$\phi(x) = \frac{1}{N}\sum_{i=1}^{N} \tilde{k}_\epsilon(x, X^i)\phi(X^i) + \tilde{h}(x)$$

But $\tilde{k}_\epsilon(x,y) = ?$

Kernel-based Algorihtm

Special case: $\rho = 1$

$$e^{\epsilon\Delta} f(x) = \int g_\epsilon(x,y) f(y)\,\mathrm{d}y. \quad \text{(for all } \epsilon > 0\text{)}$$

where g_ϵ is the Gaussian kernel.

In general:

$$e^{\epsilon\Delta_\rho} f(x) \approx \int \frac{1}{n_\epsilon(x)} \frac{g_\epsilon(x,y)}{\sqrt{\int g_\epsilon(y,z)\rho(z)\,\mathrm{d}z}} f(y)\rho(y)\,\mathrm{d}y \quad \text{(for } \epsilon \downarrow 0\text{)}$$

where n_ϵ is normalizing constant.

Empirical apprximation:

$$e^{\epsilon\Delta_\rho} f(x) \approx \sum_{j=1}^{N} \frac{1}{n_\epsilon^{(N)}(x)} \frac{g_\epsilon(x,X^j)}{\sqrt{\frac{1}{N}\sum_{l=1}^{N} g_\epsilon(X^j,X^l)}} f(X^j) \quad \text{(for } N \uparrow \infty\text{)}$$

where $n_\epsilon^{(N)}$ is normalizing constant.

R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, J. Audibert, U. Von Luxburg, Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

Kernel-based Algorihtm

Special case: $\rho = 1$

$$e^{\epsilon\Delta} f(x) = \int g_\epsilon(x,y) f(y)\,\mathrm{d}y. \quad \text{(for all } \epsilon > 0\text{)}$$

where g_ϵ is the Gaussian kernel.

In general:

$$e^{\epsilon\Delta_\rho} f(x) \approx \int \frac{1}{n_\epsilon(x)} \frac{g_\epsilon(x,y)}{\sqrt{\int g_\epsilon(y,z)\rho(z)\,\mathrm{d}z}} f(y)\rho(y)\,\mathrm{d}y \quad \text{(for } \epsilon \downarrow 0\text{)}$$

where n_ϵ is normalizing constant.

Empirical apprximation:

$$e^{\epsilon\Delta_\rho} f(x) \approx \sum_{j=1}^{N} \frac{1}{n_\epsilon^{(N)}(x)} \frac{g_\epsilon(x,X^j)}{\sqrt{\frac{1}{N}\sum_{l=1}^{N} g_\epsilon(X^j,X^l)}} f(X^j) \quad \text{(for } N \uparrow \infty\text{)}$$

where $n_\epsilon^{(N)}$ is normalizing constant.

R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, J. Audibert, U. Von Luxburg, Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

Kernel-based Algorihtm

Special case: $\rho = 1$

$$e^{\epsilon\Delta} f(x) = \int g_\epsilon(x,y) f(y)\,\mathrm{d}y. \quad \text{(for all } \epsilon > 0)$$

where g_ϵ is the Gaussian kernel.

In general:

$$e^{\epsilon\Delta_\rho} f(x) \approx \int \frac{1}{n_\epsilon(x)} \frac{g_\epsilon(x,y)}{\sqrt{\int g_\epsilon(y,z)\rho(z)\,\mathrm{d}z}} f(y)\rho(y)\,\mathrm{d}y \quad \text{(for } \epsilon \downarrow 0)$$

where n_ϵ is normalizing constant.

Empirical apprximation:

$$e^{\epsilon\Delta_\rho} f(x) \approx \sum_{j=1}^{N} \frac{1}{n_\epsilon^{(N)}(x)} \frac{g_\epsilon(x,X^j)}{\sqrt{\frac{1}{N}\sum_{l=1}^{N} g_\epsilon(X^j,X^l)}} f(X^j) \quad \text{(for } N \uparrow \infty)$$

where $n_\epsilon^{(N)}$ is normalizing constant.

R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, J. Audibert, U. Von Luxburg, Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

Input: $\underbrace{\epsilon}_{\text{kernel bandwidth}}$, $\{X^1, \dots, X^N\}$, $\{h(X^1), \dots, h(X^N)\}$

Output: Approximate solution $\phi^{\epsilon,N}$

1. Compute the (Markov) matrix $\mathbf{T} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$:

$$\mathbf{T}_{ij} = \frac{1}{n_\epsilon(X^i)} \frac{g_\epsilon(X^i, X^j)}{\sqrt{\frac{1}{N}\sum_{l=1}^N g_\epsilon(X^i, X^l)}}$$

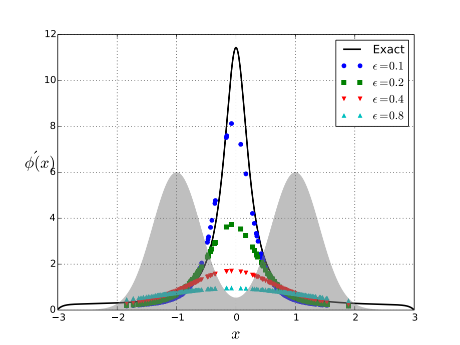
2. Compute $\Phi \in \mathbb{R}^N$ iteratively:

$$\Phi = \mathbf{T}\Phi + \epsilon(\mathbf{h} - \hat{h})$$

3. Express the approximate solution:

$$\phi^{(\epsilon,N)}(x) := \sum_{i=1}^N k_\epsilon^{(N)}(x, X^i)\Phi_i + \epsilon(h(x) - \hat{h})$$

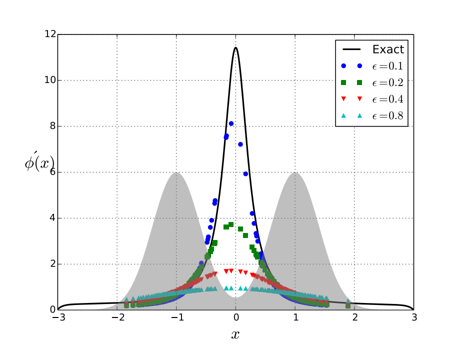
Kernel-based algorithm
Numerical result



Properties

1. Numerical stability
2. Easy extension to Manifolds [C. Zhang, et. al. CDC 2015]
3. Better error bounds
4. Computational cost $O(N^2)$ (good in high dimensions)

Kernel-based algorithm
Numerical result

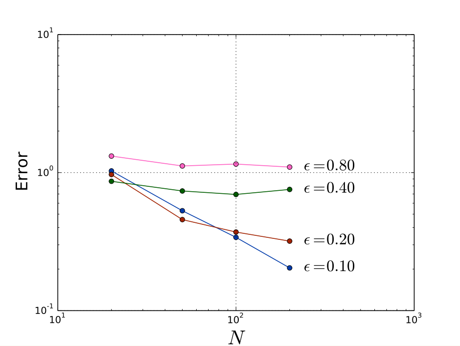


Properties

1. Numerical stability
2. Easy extension to Manifolds [C. Zhang, et. al. CDC 2015]
3. Better error bounds
4. Computational cost $O(N^2)$ (good in high dimensions)

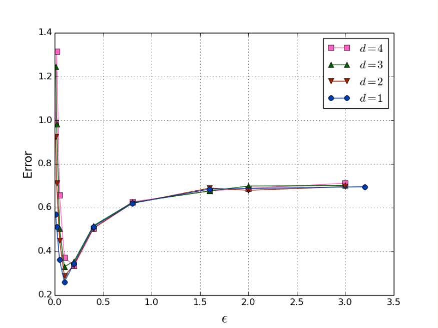
Special case: Bounded domain

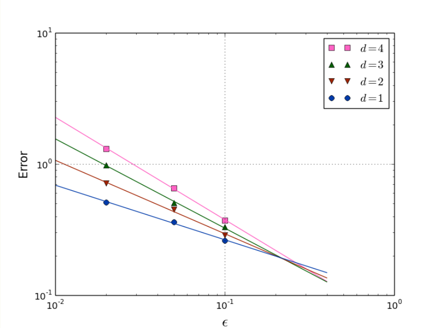
$$\underbrace{\mathsf{E}\left[\|\nabla\phi - \nabla\phi_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \leq \underbrace{O(\epsilon)}_{\text{Bias}} + \underbrace{O(\frac{1}{\epsilon^{1+d/4}\sqrt{N}})}_{\text{Variance}}$$



Special case: Bounded domain

$$\underbrace{\mathsf{E}\left[\|\nabla\phi - \nabla\phi_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \leq \underbrace{O(\epsilon)}_{\text{Bias}} + \underbrace{O(\frac{1}{\epsilon^{1+d/4}\sqrt{N}})}_{\text{Variance}}$$





Conclusion

References:

1. A. Taghvaei, P. G. Mehta, *Gain Function Approximation in the Feedback Particle Filter* IEEE Conference on Decision and Control (CDC), Las Vegas, December, 2016.
2. A. Taghvaei, P. G. Mehta. S. P. Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, IEEE American Control Conference (ACC), Seattle, May, 2017.
3. C. Zhang, A. Taghvaei, P. G. Mehta. Attitude Estimation with Feedback Particle Filter, IEE Conference on Decision and Control (CDC), Las Vegas, December, 2016.
4. C. Zhang, A. Taghvaei, P. G. Mehta. Attitude Estimation of a Wearable Motion Sensor, IEEE American Control Conference (ACC), Seattle, May, 2017.

Future work:

1. Error analysis of the overall filtering algorithm
2. Improve the computational efficiency
3. Distributed implementation

Support from CSE fellowship award is gratefully acknowledged

Thank you for your attention!