Numerical Methods to Solve the Weighted Poisson Equation

Amirhossein Taghvaei, Prashant Mehta

CSL Student Conference, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, February 2017

Classical Poisson equation

Physics

$$-\Delta\phi = h$$

ϕ is the electric/gravitational potential

Stochastic Optimal Control

$$\phi = P\phi + h$$

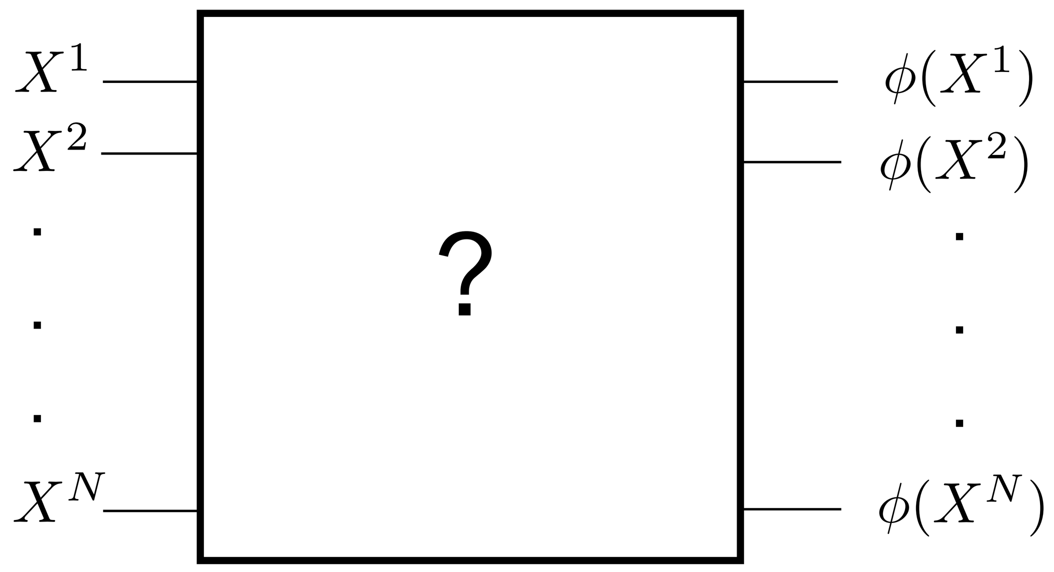
ϕ is the relative value function

Weighted Poisson equation and Problem statement

Weighted Poisson equation

$$-\frac{1}{\rho(x)}\nabla\cdot(\rho(x)\nabla\phi(x)) = h(x) - \hat{h}$$

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



Input: $\{X^1,\ldots,X^N\} \overset{\text{i.i.d}}{\sim} \rho$

Output: $\{\phi(X^1),\ldots,\phi(X^N)\}$

[R. S. Laugesen, et. al. SICON, (2015)]

Application: Classification

Feature vector: $X \in \mathbb{R}^d$

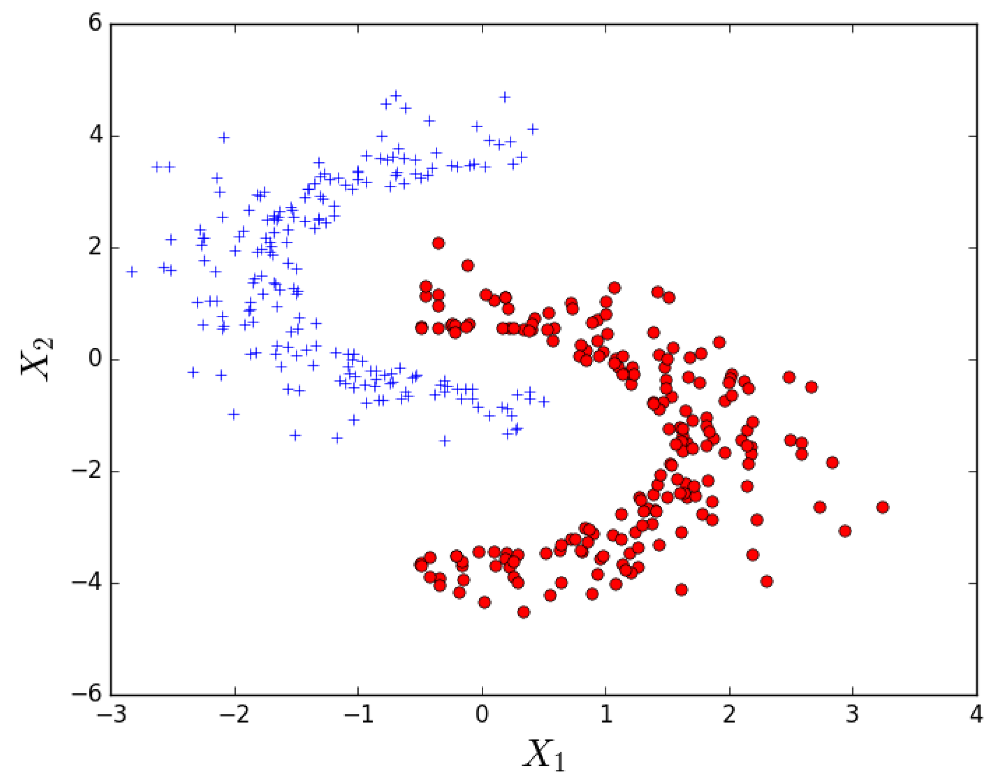
Label: $Y \in \{-1, 1\}$

Training data: $\{(X^1,Y^1),\ldots,(X^N,Y^N)\}$

Classifier: $\phi(x) = ?$

$$\min_{\phi\in\Phi} \mathrm{E}\Big[\underbrace{\frac{1}{2}|\nabla\phi(X)|^2}_{\text{Regularizer}} \underbrace{-(\phi(X)-\hat{\phi})Y}_{\text{Loss function}}\Big]$$

The minimizer solves the Poisson equation



Application: Transporting densities

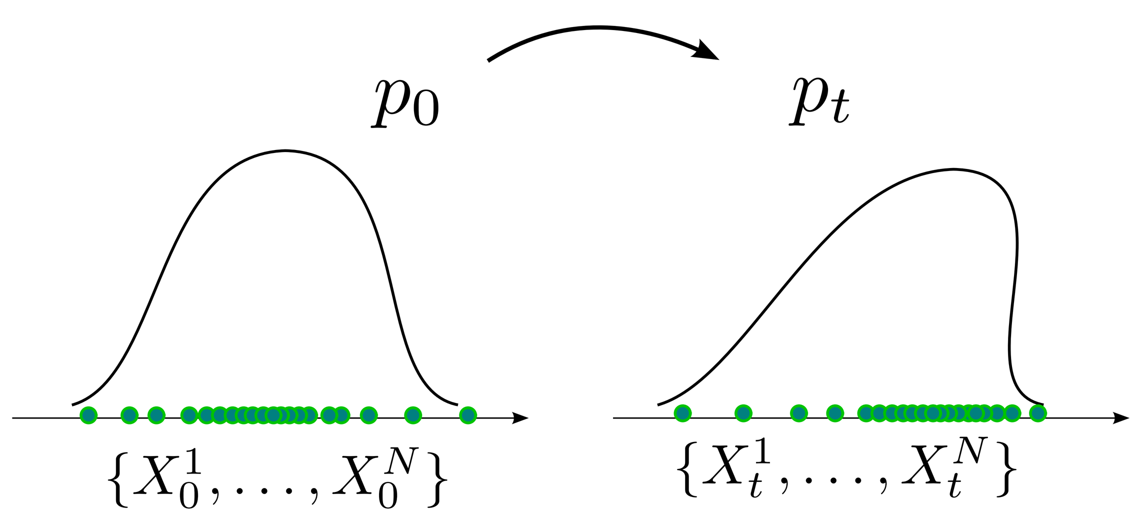
Input: $\{X_0^1,\ldots,X_0^N\} \overset{\text{i.i.d}}{\sim} p_0$ (prior)

Output: $\{X_t^1,\ldots,X_t^N\} \overset{\text{i.i.d}}{\sim} p_t$ (posterior)

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = \nabla\phi(X_t^i)$$

where ϕ solves the Poisson equation

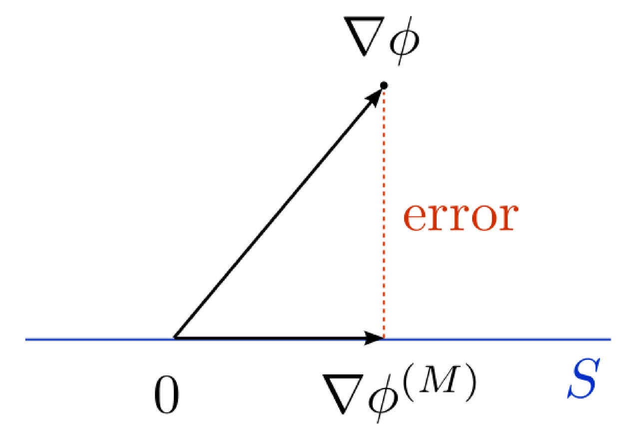
[T, Yang, et. al. Automatica, (2016)]



Two formulations, Two algorithms

1) Weak formulation: (PDE viewpoint)

$$\mathrm{E}[\nabla\phi(X)\cdot\nabla\psi(X)] = \mathrm{E}[\psi(X)(h(X)-\hat{h})] \quad \forall\psi\in H^1(\mathbb{R}^d,\rho)$$



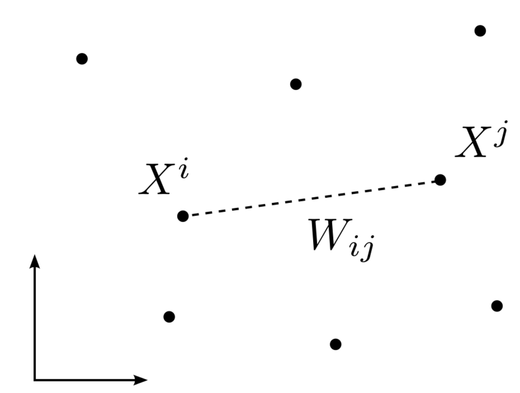
Algorithm: (Galerkin)

- Select basis functions $\{\psi_1,\ldots,\psi_M\}$
- Solve system of M linear equations

2) Semigroup formulation: (Stochastic viewpoint)

$$\phi = P\phi + \tilde{h}$$

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



Algorithm: (kernel-based)

- Approximate P with a finite rank Markov operator

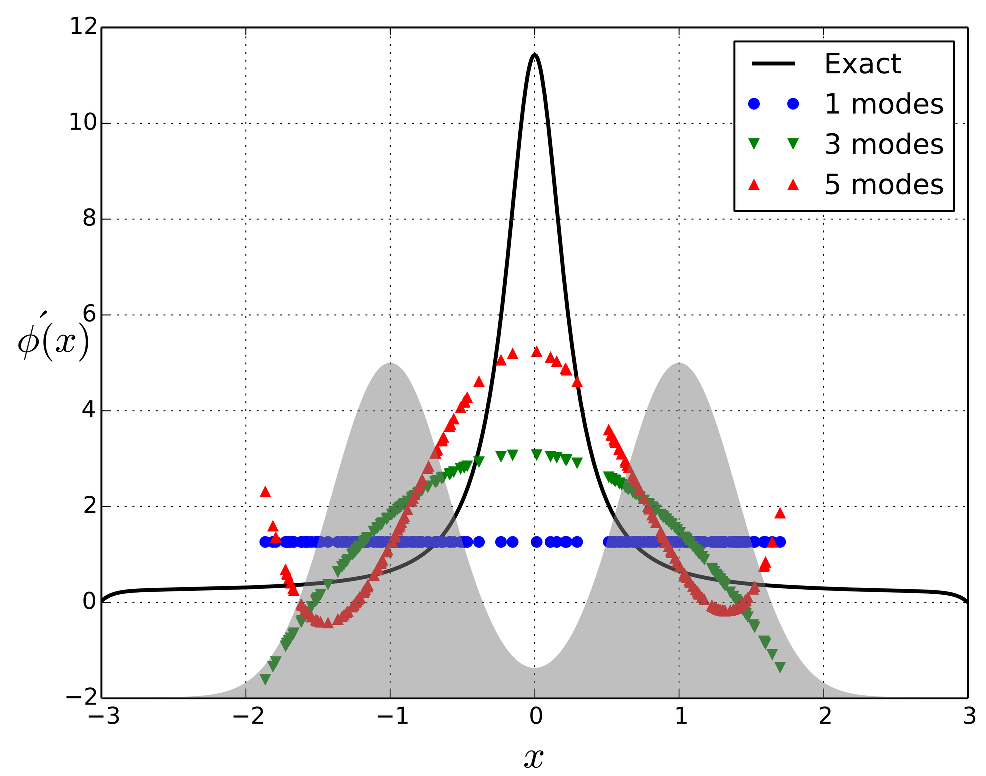
$$P\phi(x) \approx \sum_{i=1}^{N} k_\epsilon(x,X^i)\phi(X^i)$$

- Solve the fixed point equation iteratively

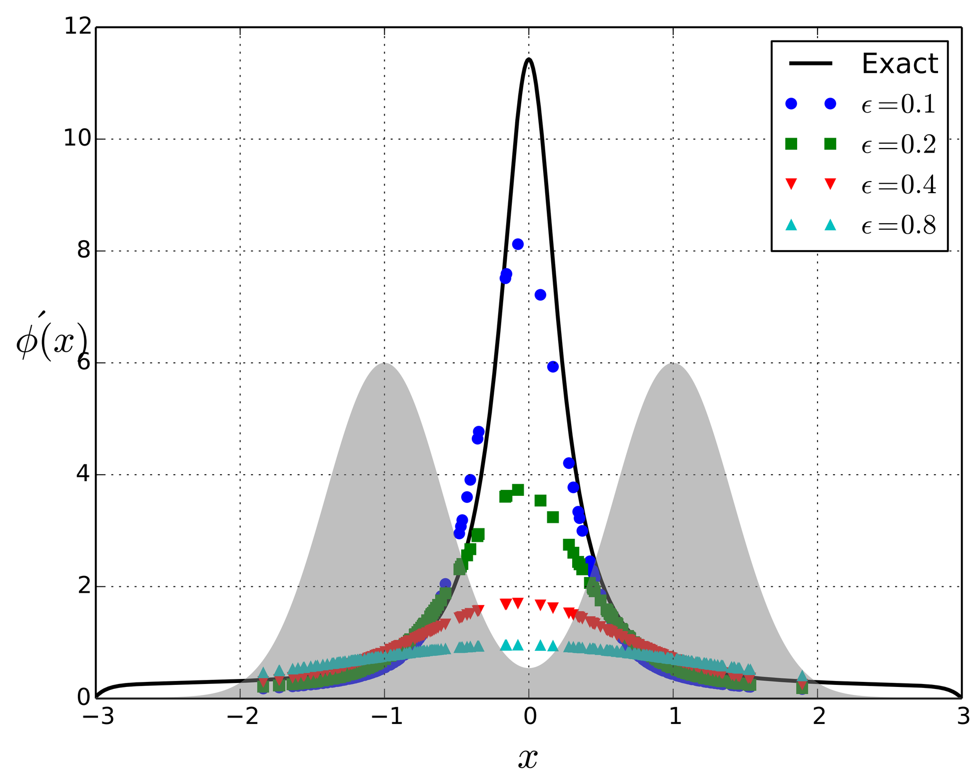
[Coifman, Lafon, 2006] [Hein, et. al. 2007]

Numerical result

Galerkin Algorithm



Kernel-based Algorithm



Error analysis

Galerkin Algorithm

$$\text{Total error} \le \underbrace{C\|h-\Pi_S h\|_{L^2}}_{\text{Bias}} + \underbrace{\frac{1}{\sqrt{N}}\|h\|_\infty\sqrt{\sum_{m=1}^{M}\frac{1}{\lambda_m}}}_{\text{Variance}}$$

Kernel-based Algorithm

$$\text{Total error} \le \underbrace{O(\epsilon)}_{\text{Bias}} + \underbrace{O\Big(\frac{1}{\epsilon^{1+d/4}\sqrt{N}}\Big)}_{\text{Variance}}$$

Acknowledgement

Research supported by Computational Science and Engineering Fellowship, UIUC, 2016-2017