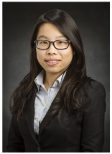
Learning Positive-Valued Functions with Pseudo Mirror Descent

Yingxiang Yang*, Haoxiang Wang*, Negar Kiyavash†, Niao He*

* University of Illinois at Urbana-Champaign (UIUC)
† Ecole Polytechnique Fédérale de Lausanne (EPFL)



NeurIPS 2019, Vancouver

Learning Positive-Valued Functions

Motivation: positive-valued functions appear ubiquitously in machine learning.

- Inference: learning probability density functions.
- Point process prediction: learning intensity-related functions.
- Ensemble learning: learning ensemble weight functions.

Learning Positive-Valued Functions

formulation

A general formulation:

$$\min_{x \in [\mathcal{H}]_+} f(x).$$

- f: objective functional.
- $\mathcal{H}$: a Hilbert space with norm $\|\cdot\|$ and inner product $\langle \cdot, \cdot \rangle$.
- **The positivity constraint**:

$$[\mathcal{H}]_+ := \{x \in \mathcal{H} : x(t) \geq 0, \ \forall t \in \text{support}(x)\}.$$

Learning Positive-Valued Functions

challenges and our contributions

Existing recipes for handling the positivity constraint:

- When f is convex, do projected gradient descent in reproducing kernel Hilbert spaces (RKHSs).
  - Theoretically guaranteed, computationally expensive on large datasets.
- Link function approach: set $x = y^2$ and optimize over y.
  - Computationally more efficient, compromises theoretical guarantees.

Learning Positive-Valued Functions

challenges and our contributions

Can we have theoretical guarantees and computational efficiency at the same time?

- Our approach: start from mirror descent algorithm.

A Mirror-Descent-Oriented Algorithm

Classical mirror descent iterate (Nemirovski & Yudin, 1983):

$$x^{(k+1)} = \operatorname*{argmin}_{x \in \mathcal{H}} \{ f(x^{(k)}) + \eta_k \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \Delta_\Phi(x, x^{(k)}) \}. \quad (1)$$

- Φ: a strongly convex function.
- η_k: the step size.
- $\Delta_\Phi(x, y)$ is the Bregman divergence:

$$\Delta_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla \Phi(y), x - y \rangle. \quad (2)$$

A Mirror-Descent-Oriented Algorithm

Certain Δ_Φ would lead to *positivity-preserving* **updates:**

$$x^{(k+1)}(t) = x^{(k)}(t) \exp(-\eta_k [\nabla f(x^{(k)})](t)).$$

- $\Delta_\Phi(x, y) = \langle x, \log x - \log y \rangle$.
- $\mathcal{H}$ chosen to be $\mathcal{L}_2$ Hilbert space.

Challenge: gradient not always available in practice.

A Mirror-Descent-Oriented Algorithm

Poisson maximum log-likelihood estimation:

$$\min_{x \in [\mathcal{L}_2[0,1]]_+} \quad f(x) := \int_0^1 x(t) - x^*(t) \log x(t) \mathrm{d}t. \tag{3}$$

- x^*: ground truth intensity function.

The gradient

$$[\nabla f(x)](t) = 1 - \frac{x^*(t)}{x(t)} \tag{4}$$

requires value of x^* (unknown in practice!)

Pseudo-Gradients

Pseudo-gradients (Polyak, 1973):

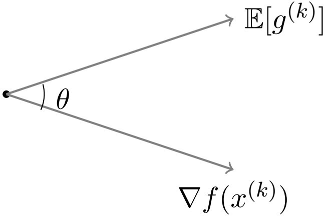


Figure: Pseudo-gradient for gradient descent.

- $g^{(k)}$ is a pseudo-gradient when $\theta < 90^\circ$:

$$\langle \mathbb{E}[g^{(k)}], \nabla f(x^{(k)}) \rangle \geq 0.$$

Pseudo-gradients for mirror descent (this work):

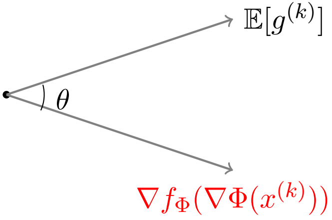


Figure: Pseudo-gradient for mirror descent.

- $\nabla f_\Phi(\nabla \Phi(x^{(k)}))$: gradient adapted to the Bregman divergence.

$$\langle \mathbb{E}[g^{(k)}], \nabla f_\Phi(\nabla \Phi(x^{(k)})) \rangle \geq 0.$$

$f_\Phi(z) = f(\nabla \Phi^*(z))$ where Φ^* is the Fenchel dual of Φ.

Pseudo Mirror Descent

algorithm and theory

The pseudo mirror descent (PMD) algorithm:

$$\textbf{PMD} = \textbf{classical mirror descent} + \textbf{pseudo-gradients}.$$

Theoretical guarantees?

- Yes! Under standard assumptions, converges in
  - gradient norm at rate $\mathcal{O}(1/\sqrt{k})$.
  - objective value at rate $\mathcal{O}(1/k)$ (with Polyak-Łojasiewicz condition).

Can pseudo-gradients be efficiently constructed?

- Yes! For example, use the kernel embedding of $\nabla f_\Phi(\nabla \Phi(x))$.

Pseudo Mirror Descent in Action

learning intensity functions for Poisson processes

For the Poisson example: $\nabla f_\Phi(\nabla\Phi(x)) = x - x^*$, and

$$g^{(k)}(t) = \sum_{j=1}^{N_1} K(\tau_j, t) - \sum_{i=1}^{N_2} K(\tau_i, t)$$

for a positive definite kernel $K(\cdot,\cdot)$.

- τ_j's: sampled from $x^{(k)}$.
- τ_i's: sampled from x^*.

Pseudo Mirror Descent in Action

simulation

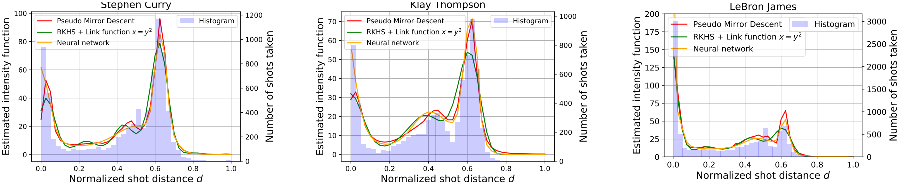


Figure: Basketball shot distance dataset: recovery of the intensities using pseudo mirror descent (red curve), the link function approach), and neural networks (yellow curve).

Poster 55

East Exhibition Hall B+C

Tuesday, Dec.10th, 5:30 - 7:30 p.m.