Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

Basim Khajwal Luke Ong **Dominik Wagner**

UNIVERSITY OF OXFORD

32th European Symposium on Programming
26 April 2023

Probabilistic Programming

Probabilistic Programming

= programming paradigm to pose *Bayesian Inference* problems

Probabilistic Programming

= programming paradigm to pose *Bayesian Inference* problems

- separate modelling from inference

Variational Inference

Variational Inference:

frame posterior inference as (deterministic) optimisation problem

Variational Inference:

frame posterior inference as (deterministic) optimisation problem

Posit: *variational family* of “simpler” *guide* distributions

Variational Inference:

frame posterior inference as (deterministic) optimisation problem

Posit: *variational family* of "simpler" *guide* distributions

Aim: find guide that is *"closest"* to (true) posterior

Variational Inference:

frame posterior inference as (deterministic) optimisation problem

Posit: *variational family* of “simpler” *guide* distributions

Aim: find guide that is *“closest”* to (true) posterior

KL divergence

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s} \sim \mathcal{D}_{\boldsymbol{\theta}}} \left[f(\boldsymbol{\theta}, \mathsf{s}) \right]}$$

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\boldsymbol{\theta}}}\left[f(\boldsymbol{\theta}, \mathbf{s})\right]}$$

use ***Stochastic Gradient Descent***

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}_{\boldsymbol{\theta}}} \left[f(\boldsymbol{\theta}, \mathbf{s}) \right]}$$

use ***Stochastic Gradient Descent***

Key ingredient: estimation of gradient of expectation

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s} \sim \mathcal{D}_{\boldsymbol{\theta}}} \left[f(\boldsymbol{\theta}, \mathsf{s}) \right]}$$

use ***Stochastic Gradient Descent***

Key ingredient: estimation of gradient of expectation

- Score Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}_{\boldsymbol{\theta}}} \left[f(\boldsymbol{\theta}, \mathbf{s}) \right]}$$

use ***Stochastic Gradient Descent***

Key ingredient: estimation of gradient of expectation

- **Score Estimator:**
  widely applicable but *high variance*

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\boldsymbol{\theta}}}\left[f(\boldsymbol{\theta}, \mathbf{s})\right]}$$

use ***Stochastic Gradient Descent***

Key ingredient: estimation of gradient of expectation

- **Score Estimator:**
  widely applicable but *high variance*
- **Reparameterisation Estimator**

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\boldsymbol{\theta}}}\left[f(\boldsymbol{\theta}, \mathbf{s})\right]}$$

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\cancel{\boldsymbol{\theta}}}}[f(\boldsymbol{\theta}, \mathbf{s})]}$$

eliminate dependence on $\boldsymbol{\theta}$

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\cancel{\boldsymbol{\theta}}}}[f(\boldsymbol{\theta}, \mathbf{s})]}$$

eliminate dependence on $\boldsymbol{\theta}$

estimate: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[f(\boldsymbol{\theta}, \mathbf{s})] \approx \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \widehat{\mathbf{s}})$, where $\widehat{\mathbf{s}} \sim \mathcal{D}$

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}_{\cancel{\boldsymbol{\theta}}}}[f(\boldsymbol{\theta}, \mathsf{s})]}$$

eliminate dependence on $\boldsymbol{\theta}$

estimate: $\nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}\, [f(\boldsymbol{\theta}, \mathsf{s})] \approx \nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \widehat{\mathsf{s}})$, where $\widehat{\mathsf{s}} \sim \mathcal{D}$

(Unbiasedness) $\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, \mathsf{s})] \overset{?}{=} \nabla_{\boldsymbol{\theta}}\, \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[f(\boldsymbol{\theta}, \mathsf{s})]$

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}_{\cancel{\boldsymbol{\theta}}}}[f(\boldsymbol{\theta}, \mathsf{s})]}$$

eliminate dependence on $\boldsymbol{\theta}$

expressed in PL with conditionals,
may *not* be *differentiable/continuous*

estimate: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}} [f(\boldsymbol{\theta}, \mathsf{s})] \approx \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \widehat{\mathsf{s}})$, where $\widehat{\mathsf{s}} \sim \mathcal{D}$

(Unbiasedness) $\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \mathsf{s})] \overset{?}{=} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[f(\boldsymbol{\theta}, \mathsf{s})]$

Reparameterisation Gradient Estimator

$$\boxed{\operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}_{\cancel{\boldsymbol{\theta}}}}[f(\boldsymbol{\theta},\mathsf{s})]}$$

eliminate dependence on $\boldsymbol{\theta}$

expressed in PL with conditionals,
may *not* be *differentiable/continuous*

estimate: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[f(\boldsymbol{\theta},\mathsf{s})] \approx \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\widehat{\mathsf{s}})$, where $\widehat{\mathsf{s}} \sim \mathcal{D}$

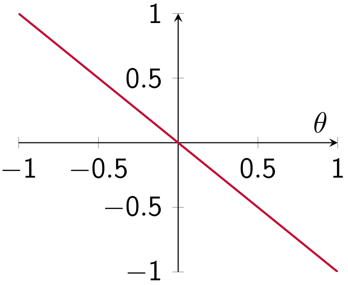
(Unbiasedness) $\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta},\mathsf{s})] \overset{?}{=} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[f(\boldsymbol{\theta},\mathsf{s})]$

may be compromised! [Lee et al., NeurIPS 2018]

$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

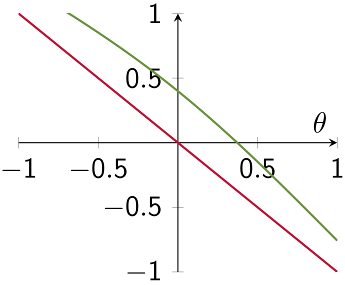
$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbb{E}_{\mathbf{s} \sim \mathcal{N}(\mathbf{0},\mathbf{1})} \left[\nabla_\theta \, \mathbf{f}(\theta, \mathbf{s}) \right] = -\theta$$



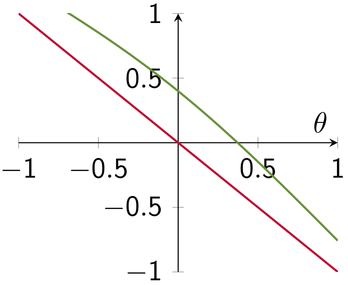
$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbb{E}_{\mathbf{s} \sim \mathcal{N}(\mathbf{0},\mathbf{1})} \left[\nabla_\theta \, \mathbf{f}(\theta, \mathbf{s})\right] = -\theta \neq -\theta + \mathcal{N}(-\theta \mid 0, 1) = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{s \sim \mathcal{N}(0,1)} \left[f(\theta, s)\right]$$



$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbb{E}_{\mathbf{s}\sim\mathcal{N}(\mathbf{0},\mathbf{1})}\left[\nabla_\theta \, \mathbf{f}(\theta, \mathbf{s})\right] = -\theta \neq -\theta + \mathcal{N}(-\theta \mid 0, 1) = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{s\sim\mathcal{N}(0,1)}\left[f(\theta, s)\right]$$



Stochastic Gradient Descent is incorrect!

Contributions

Fast yet correct **Stochastic Gradient Descent**
with Reparameterisation Gradient ***via Smoothing***

Contributions

Fast yet correct **Stochastic Gradient Descent with Reparameterisation Gradient** ***via Smoothing***

- Smoothed Denotational (Value) Semantics

Contributions

Fast yet correct **Stochastic Gradient Descent**
with Reparameterisation Gradient ***via Smoothing***

- Smoothed Denotational (Value) Semantics
- Correctness of Stochastic Gradient Descent via Type System

Contributions

Fast yet correct **Stochastic Gradient Descent**
with Reparameterisation Gradient ***via Smoothing***

- Smoothed Denotational (Value) Semantics
- Correctness of Stochastic Gradient Descent via Type System
- Convergence of Smooth Approximations

Contributions

Fast yet correct* Stochastic Gradient Descent with Reparameterisation Gradient *via Smoothing

- Smoothed Denotational (Value) Semantics
- Correctness of Stochastic Gradient Descent via Type System
- Convergence of Smooth Approximations
- Empirical Evaluation

Part I:
Problem Setup

simply typed λ-calculus with $\mathbb{R}$, primitive operations, parameters θ_i

$$M ::= x \mid \lambda x.\, M \mid M\, M \mid f(M, \ldots, M) \mid \theta_i$$

simply typed λ-calculus with $\mathbb{R}$, primitive operations, parameters θ_i
+ sample

$$\begin{aligned} M ::= {} & x \mid \lambda x.\, M \mid M\, M \mid f(M, \ldots, M) \mid \theta_i \\ & \mid \textbf{sample}_{\mathcal{D}} \end{aligned}$$

simply typed λ-calculus with $\mathbb{R}$, primitive operations, parameters θ_i

$+$ sample

$+$ *branching*

$$\begin{aligned} M ::= {} & x \mid \lambda x.\, M \mid M\, M \mid f(M, \ldots, M) \mid \theta_i \\ & \mid \mathbf{sample}_{\mathcal{D}} \\ & \mid \mathbf{if}\ M < 0\ \mathbf{then}\ M\ \mathbf{else}\ M \end{aligned}$$

Denotational Value Semantics:

Denotational Value Semantics: *deterministic* function from samples to value

Denotational Value Semantics: *deterministic* function from samples to value

$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

Denotational Value Semantics: *deterministic* function from samples to value

$$f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$M \equiv (\lambda s.\, -0.5 \cdot \theta^2 + (\textbf{if}\; s + \theta < 0 \;\textbf{then}\; 0 \;\textbf{else}\; 1))\;\textbf{sample}_{\mathcal{N}}$$

Denotational Value Semantics: *deterministic* function from samples to value

$$[\![M]\!]\,(\theta, s) = f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$M \equiv (\lambda s.\, -0.5 \cdot \theta^2 + (\mathbf{if}\ s + \theta < 0\ \mathbf{then}\ 0\ \mathbf{else}\ 1))\ \mathbf{sample}_{\mathcal{N}}$$

Denotational Value Semantics: *deterministic* function from samples to value

$$\llbracket M \rrbracket (\theta, s) = f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$M \equiv (\lambda s.\, -0.5 \cdot \theta^2 + (\textbf{if}\ s + \theta < 0\ \textbf{then}\ 0\ \textbf{else}\ 1))\ \textbf{sample}_{\mathcal{N}}$$

Track samples (and distributions) in type system

Denotational Value Semantics: *deterministic* function from samples to value

$$[\![M]\!](\theta, s) = f(\theta, s) = -0.5 \cdot \theta^2 + \begin{cases} 0 & \text{if } s + \theta < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$M \equiv (\lambda s.\, -0.5 \cdot \theta^2 + (\textbf{if } s + \theta < 0 \textbf{ then } 0 \textbf{ else } 1))\, \textbf{sample}_{\mathcal{N}}$$

Track samples (and distributions) in type system

$$\theta : R \mid [\mathcal{N}] \vdash M : R$$

Problem Statement

Given: term-in-context, $\theta_1 : R, \ldots, \theta_m : R \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash M : R$

Problem Statement

Given: term-in-context, $\theta_1 : R, \ldots, \theta_m : R \mid [\mathcal{D}_1, \ldots, \mathcal{D}_n] \vdash M : R$

Find: $\mathrm{argmin}_{\boldsymbol{\theta}}\ \mathbb{E}_{s_1 \sim \mathcal{D}_1, \ldots, s_n \sim \mathcal{D}_n} \left[[\![M]\!] (\boldsymbol{\theta}, \mathsf{s}) \right]$

$$\boxed{\mathbb{E}_{s\sim\mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$\boxed{\mathbb{E}_{s\sim\mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$(\lambda x.\, \exp\,(x \cdot x))\,\mathbf{sample}_{\mathcal{N}}$$

$$\boxed{\mathbb{E}_{s\sim\mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$(\lambda x.\, \exp\,(x \cdot x))\, \mathbf{sample}_{\mathcal{N}}$$

Simplified assumption:

1. distributions have finite moments

$$\boxed{\mathbb{E}_{s \sim \mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$(\lambda x. \exp(x \cdot x))\,\textbf{sample}_{\mathcal{N}}$$

Simplified assumption:

1. distributions have finite moments $\mathbb{E}_{s \sim \mathcal{D}}[|s^p|] < \infty$

$$\boxed{\mathbb{E}_{s\sim\mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$(\lambda x.\, \exp\,(x \cdot x))\, \mathbf{sample}_{\mathcal{N}}$$

Simplified assumption:

1. distributions have finite moments $\mathbb{E}_{s\sim\mathcal{D}}[|s^p|] < \infty$
2. primitives are bounded by polynomials

$$\boxed{\mathbb{E}_{s\sim\mathcal{N}}\left[\exp(s^2)\right] = \infty}$$

$$(\lambda x.\exp(x\cdot x))\,\mathbf{sample}_{\mathcal{N}}$$

Simplified assumption:

1. distributions have finite moments $\mathbb{E}_{s\sim\mathcal{D}}[|s^p|] < \infty$
2. primitives are bounded by polynomials

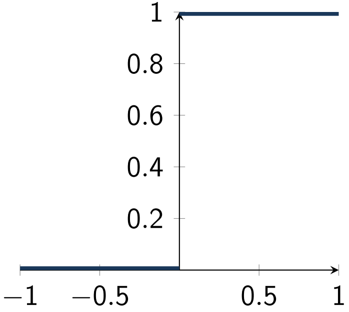
In paper: relax assumption, control use of $\log, \exp, {}^{-1}$ via type system

Part II:

Smoothed Value Semantics

$$[\![\textbf{if } z < 0 \textbf{ then } 0 \textbf{ else } M]\!]\,(z) = \begin{cases} 0 & \text{if } z < 0 \\ [\![M]\!]\,(z) & \text{otherwise} \end{cases}$$

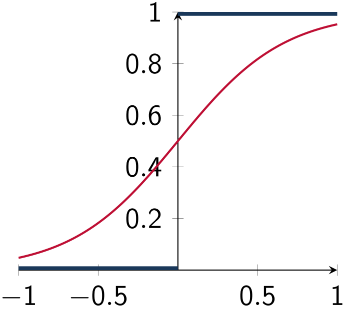
$$[\![\textbf{if}\ z < 0\ \textbf{then}\ 0\ \textbf{else}\ M]\!]\,(z) = [z \geq 0] \cdot [\![M]\!]\,(z)$$



$$[\![\textbf{if } z < 0 \textbf{ then } 0 \textbf{ else } M]\!]\,(z) = [z \geq 0] \cdot [\![M]\!]\,(z)$$

$$[\![\textbf{if } z < 0 \textbf{ then } 0 \textbf{ else } M]\!]_\eta\,(z) = \sigma_\eta(z) \cdot [\![M]\!]_\eta\,(z)$$

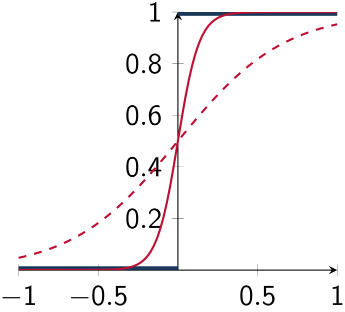
sigmoid function



$$[\![\mathbf{if}\ z < 0\ \mathbf{then}\ 0\ \mathbf{else}\ M]\!]\,(z) = [z \geq 0] \cdot [\![M]\!]\,(z)$$

$$[\![\mathbf{if}\ z < 0\ \mathbf{then}\ 0\ \mathbf{else}\ M]\!]_\eta\,(z) = \sigma_\eta(z) \cdot [\![M]\!]_\eta\,(z)$$

sigmoid function (parameterised by *accuracy* coefficient $\eta > 0$)



Categorical Model?

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta \coloneqq (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta \coloneqq (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

morphism?

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta \coloneqq (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

morphism?

Adapt construction of Frölicher spaces

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta := (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

morphism?

Adapt construction of Frölicher spaces

+ vector space structure for underlying set

+ condition for “curves”

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta \coloneqq (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

morphism?

Adapt construction of Frölicher spaces

\+ vector space structure for underlying set

\+ condition for “curves”

CCC **VectFr** of *Vector Frölicher Spaces*

Categorical Model?

Smoothness: CCC of *Frölicher* spaces

$$[\![\textbf{if } L < 0 \textbf{ then } M \textbf{ else } N]\!]_\eta \coloneqq (\sigma_\eta \circ (-[\![L]\!]_\eta)) \cdot [\![M]\!]_\eta + (\sigma_\eta \circ [\![L]\!]_\eta) \cdot [\![N]\!]_\eta$$

morphism?

Adapt construction of Frölicher spaces

+ vector space structure for underlying set
+ condition for “curves”

CCC **VectFr** of *Vector Frölicher Spaces*

If $\phi_1, \phi_2 \in \mathbf{VectFr}(X, Y)$ and $\alpha \in \mathbf{Vect}(X, \mathbb{R})$ then $\alpha \cdot \phi_1 + \phi_2 \in \mathbf{VectFr}(X, Y)$.

Part III:

Applying Stochastic Gradient Descent

$$\boldsymbol{\theta}_{k+1} \coloneqq \boldsymbol{\theta}_k - \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \left[\!\left[M \right]\!\right]_\eta (\boldsymbol{\theta}_k, \mathsf{s}_k) \qquad\qquad \mathsf{s}_k \sim \mathcal{D}$$

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta (\boldsymbol{\theta}_k, \mathsf{s}_k)}_{\text{gradient estimation}} \qquad\qquad \mathsf{s}_k \sim \mathcal{D}$$

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} [\![M]\!]_{\eta} (\boldsymbol{\theta}_k, \mathsf{s}_k)}_{\text{gradient estimation}} \qquad \mathsf{s}_k \sim \mathcal{D}$$

step size

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} \llbracket M \rrbracket_\eta (\boldsymbol{\theta}_k, \mathsf{s}_k)}_{\text{gradient estimation}} \qquad \mathsf{s}_k \sim \mathcal{D}$$

step size

(Unbiasedness) $$\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} \llbracket M \rrbracket_\eta (\boldsymbol{\theta}, \mathsf{s})] = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\llbracket M \rrbracket_\eta (\boldsymbol{\theta}, \mathsf{s})]$$

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}_k, \mathsf{s}_k)}_{\text{gradient estimation}} \qquad \mathsf{s}_k \sim \mathcal{D}$$

step size

(Unbiasedness) $$\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})] = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})]$$

partial derivatives of $[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})$ *are bounded by polynomial*

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} [\![M]\!]_\eta(\boldsymbol{\theta}_k, \mathsf{s}_k)}_{\text{gradient estimation}} \qquad\qquad \mathsf{s}_k \sim \mathcal{D}$$

step size

(Unbiasedness) $$\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} [\![M]\!]_\eta(\boldsymbol{\theta}, \mathsf{s})] = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[[\![M]\!]_\eta(\boldsymbol{\theta}, \mathsf{s})]$$

partial derivatives of $[\![M]\!]_\eta(\boldsymbol{\theta}, \mathsf{s})$ *are bounded by polynomial*

Correctness of SGD for Smoothing

If M is typable, $\boldsymbol{\Theta}$ is compact and the step size scheme is “suitable”

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}_k, \mathbf{s}_k)}_{\text{gradient estimation}} \qquad\qquad \mathbf{s}_k \sim \mathcal{D}$$

step size

(Unbiasedness) $$\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})] = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})]$$

partial derivatives of $[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})$ *are bounded by polynomial*

Correctness of SGD for Smoothing

If M is typable, $\mathbf{\Theta}$ is compact and the step size scheme is “suitable” then

$$\inf_{i\in\mathbb{N}} \mathbb{E}[\nabla g(\boldsymbol{\theta}_i)] = 0$$

where $g(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathsf{s})]$.

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \gamma_k \cdot \underbrace{\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}_k, \mathbf{s}_k)}_{\text{gradient estimation}} \qquad\qquad \mathbf{s}_k \sim \mathcal{D}$$

step size

(Unbiasedness) $\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}} \, [\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathbf{s})] = \nabla_{\boldsymbol{\theta}} \, \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathbf{s})]$

partial derivatives of $[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathbf{s})$ *are bounded by polynomial*

Correctness of SGD for Smoothing

If M is typable, $\boldsymbol{\Theta}$ is compact and the step size scheme is “suitable” then

$$\inf_{i\in\mathbb{N}} \mathbb{E}[\nabla g(\boldsymbol{\theta}_i)] = 0$$

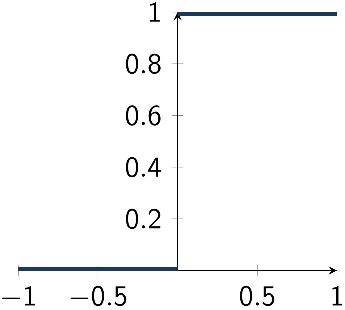
where $g(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[[\![M]\!]_\eta \, (\boldsymbol{\theta}, \mathbf{s})]$.

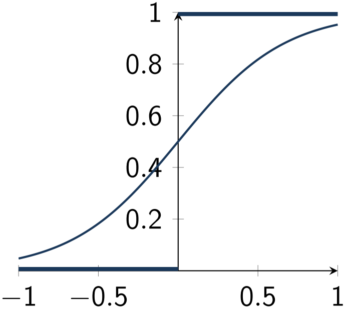
exploit Lipschitz smoothness and bounded variance

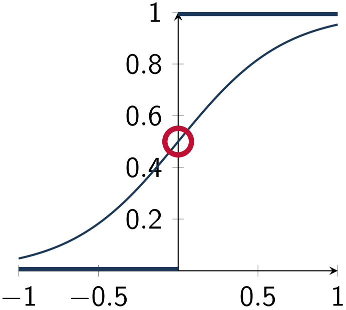
How does solving the smoothed problem help solve the original problem?

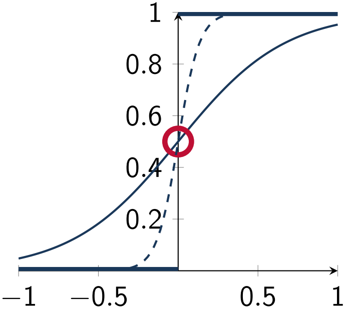
Part IV:

Convergence of Smoothings


1
0.8
0.6
0.4
0.2
−1
−0.5
0.5
1

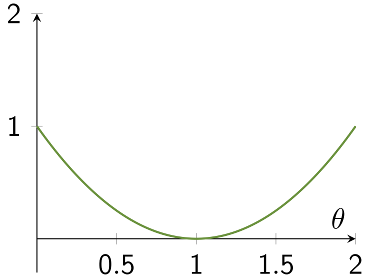

1
0.8
0.6
0.4
0.2
−1
−0.5
0.5
1


1
0.8
0.6
0.4
0.2
−1
−0.5
0.5
1


1
0.8
0.6
0.4
0.2
−1
−0.5
0.5
1

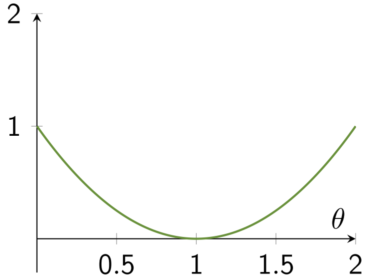
$$\boxed{M \equiv \textbf{if}\ 0 < 0\ \textbf{then}\ \theta^2 + 1\ \textbf{else}\ (\theta - 1)^2}$$

$$\boxed{M \equiv \textbf{if } 0 < 0 \textbf{ then } \theta^2 + 1 \textbf{ else } (\theta - 1)^2}$$



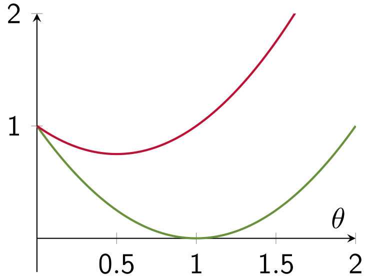
$$\boxed{M \equiv \textbf{if}\ 0 < 0\ \textbf{then}\ \theta^2 + 1\ \textbf{else}\ (\theta - 1)^2}$$

$$[\![M]\!]_\eta(\theta) = \frac{1}{2}(\theta^2 + 1) + \frac{1}{2}(\theta - 1)^2$$



$$\boxed{M \equiv \textbf{if } 0 < 0 \textbf{ then } \theta^2 + 1 \textbf{ else } (\theta - 1)^2}$$

$$[\![M]\!]_\eta(\theta) = \frac{1}{2}(\theta^2 + 1) + \frac{1}{2}(\theta - 1)^2$$



Ensure that guards are not 0 almost everywhere

$\textbf{if}\ x - x < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$$\textbf{if } x - x < 0 \textbf{ then } M \textbf{ else } N \quad ✗$$

$$(\lambda y, z.\, \textbf{if } y - z < 0 \textbf{ then } M \textbf{ else } N)\, x\, x \quad ✗$$

$\textbf{if}\ x - x < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$(\lambda y, z.\, \textbf{if}\ y - z < 0\ \textbf{then}\ M\ \textbf{else}\ N)\ x\ x$ ✗

$\textbf{if}\ \theta < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$\textbf{if}\ x - x < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$(\lambda y, z.\ \textbf{if}\ y - z < 0\ \textbf{then}\ M\ \textbf{else}\ N)\ x\ x$ ✗

$\textbf{if}\ \theta < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$\textbf{if}\ (\ \textbf{sample}_{\mathcal{N}} + \theta\ ) < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✓

$\textbf{if } x - x < 0 \textbf{ then } M \textbf{ else } N$ ✗

$(\lambda y, z.\, \textbf{if } y - z < 0 \textbf{ then } M \textbf{ else } N)\; x\; x$ ✗

$\textbf{if } \theta < 0 \textbf{ then } M \textbf{ else } N$ ✗

$\textbf{if } (\underbrace{\textsf{sample}_{\mathcal{N}} + \theta}_{\textbf{transform sample}_{\mathcal{N}} \textbf{ by } (\lambda x.x+\theta)}) < 0 \textbf{ then } M \textbf{ else } N$ ✓

$\textbf{if}\ x - x < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$(\lambda y, z.\, \textbf{if}\ y - z < 0\ \textbf{then}\ M\ \textbf{else}\ N)\ x\ x$ ✗

$\textbf{if}\ \theta < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✗

$\textbf{if}\ (\underbrace{\textbf{sample}_{\mathcal{N}} + \theta}_{\textbf{transform sample}_{\mathcal{N}}\ \textbf{by}\ (\lambda x. x+\theta)}) < 0\ \textbf{then}\ M\ \textbf{else}\ N$ ✓

$(\lambda y, z.\, \textbf{if}\ y - z < 0\ \textbf{then}\ M\ \textbf{else}\ N)\ \textbf{sample}_{\mathcal{N}}\ (\textbf{transform sample}_{\mathcal{N}}\ \textbf{by}\ T)$ ✓

$$\tau ::= R^{(g,\Delta)}$$

$$\tau ::= R^{(g,\Delta)}$$

guard-safe?

$$\tau ::= R^{(g,\Delta)}$$

guard-safe?

dependency on (transformed) samples

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \qquad \Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N : \sigma}$$

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe? dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \quad \Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash \textbf{if } L < 0 \textbf{ then } M \textbf{ else } N : \sigma}$$

$$\frac{}{\Gamma \vdash 0 : R^{(\mathbf{f},\Delta)}}$$

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \quad \Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash \textbf{if } L < 0 \textbf{ then } M \textbf{ else } N : \sigma}$$

$$\frac{}{\Gamma \vdash 0 : R^{(\mathbf{f},\Delta)}}$$

$$\frac{}{\Gamma \vdash \textbf{transform sample}_{\mathcal{N}} \textbf{ by } T : R^{(\mathbf{t},\{s_j\})}} \; T \text{ diffeomorphic}$$

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \qquad \Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N : \sigma}$$

$$\frac{}{\Gamma \vdash 0 : R^{(\mathbf{f},\Delta)}}$$

$$\frac{}{\Gamma \vdash \mathbf{transform\ sample}_{\mathcal{N}}\ \mathbf{by}\ T : R^{(\mathbf{t},\{s_j\})}}\ T \text{ diffeomorphic}$$

fresh

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \quad \Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash \textbf{if } L < 0 \textbf{ then } M \textbf{ else } N : \sigma}$$

$$\frac{}{\Gamma \vdash 0 : R^{(\mathbf{f},\Delta)}}$$

$$\frac{}{\Gamma \vdash \textbf{transform sample}_{\mathcal{N}} \textbf{ by } T : R^{(\mathbf{t},\{s_j\})}} \quad T \text{ diffeomorphic}$$

fresh

$$\frac{\Gamma \vdash M : R^{(\mathbf{t},\Delta_1)} \quad N : R^{(\mathbf{t},\Delta_2)}}{\Gamma \vdash M - N : R^{(\mathbf{t},\Delta_1 \cup \Delta_2)}} \quad \Delta_1 \cap \Delta_2 = \emptyset$$

$$\tau ::= R^{(g,\Delta)} \mid \tau \to \tau$$

guard-safe?

dependency on (transformed) samples

$$\frac{\Gamma \vdash L : R^{(\mathbf{t},\Delta)} \qquad \Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash \mathbf{if}\ L < 0\ \mathbf{then}\ M\ \mathbf{else}\ N : \sigma}$$

$$\frac{}{\Gamma \vdash 0 : R^{(\mathbf{f},\Delta)}}$$

$$\frac{}{\Gamma \vdash \mathbf{transform\ sample}_{\mathcal{N}}\ \mathbf{by}\ T : R^{(\mathbf{t},\{s_j\})}}\ T \text{ diffeomorphic}$$

fresh

$$\frac{\Gamma \vdash M : R^{(\mathbf{t},\Delta_1)} \qquad N : R^{(\mathbf{t},\Delta_2)}}{\Gamma \vdash M - N : R^{(\mathbf{t},\Delta_1 \cup \Delta_2)}}\ \Delta_1 \cap \Delta_2 = \emptyset$$

Establish correctness via logical relations

Uniform Convergence

If M is typable then

$$\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\llbracket M\rrbracket_\eta(\boldsymbol{\theta},\mathsf{s})] \xrightarrow{\text{unif.}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\llbracket M\rrbracket(\boldsymbol{\theta},\mathsf{s})] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta}\in\boldsymbol{\Theta}$$

Uniform Convergence

If M is typable then

$$\mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[[\![M]\!]_\eta(\boldsymbol{\theta},\mathsf{s})] \xrightarrow{\text{unif.}} \mathbb{E}_{\mathsf{s}\sim\mathcal{D}}[[\![M]\!](\boldsymbol{\theta},\mathsf{s})] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

For any error tolerance $\epsilon > 0$,

exists accuracy coefficient $\eta > 0$ **s.t.** ***for all*** $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\mathbb{E}_{\mathsf{s}}[[\![M]\!](\boldsymbol{\theta},\mathsf{s})] < \mathbb{E}_{\mathsf{s}}[[\![M]\!]_\eta(\boldsymbol{\theta},\mathsf{s})] + \epsilon$$

Uniform Convergence

If M is typable then

$$\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\llbracket M \rrbracket_\eta (\boldsymbol{\theta}, \mathsf{s})] \xrightarrow{\text{unif.}} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}[\llbracket M \rrbracket (\boldsymbol{\theta}, \mathsf{s})] \qquad \text{as } \eta \searrow 0 \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta}$$

For any error tolerance $\epsilon > 0$,

exists accuracy coefficient $\eta > 0$ **s.t.** ***for all*** $\boldsymbol{\theta} \in \boldsymbol{\Theta}$

$$\mathbb{E}_{\mathbf{s}}[\llbracket M \rrbracket (\boldsymbol{\theta}, \mathsf{s})] < \mathbb{E}_{\mathbf{s}}[\llbracket M \rrbracket_\eta (\boldsymbol{\theta}, \mathsf{s})] + \epsilon$$

In particular for θ^* **obtained by**

SGD with Reparameterisation Gradient (*fast!*) for η**-smoothing**

Part V:

Empirical Evaluation

Score Estimator

- ✗ high variance

Standard Reparameterisation Estimator

- ✗ biased

Score Estimator

- ✗ high variance

Standard Reparameterisation Estimator

- ✗ biased

[Lee et al., NeurIPS 2018]:

Score Estimator

✗ high variance

Standard Reparameterisation Estimator

✗ biased

[Lee et al., NeurIPS 2018]:

- Fix bias with additional non-trivial *boundary* terms

✗ Only discuss efficient method for *affine* guards

Score Estimator

✗ high variance

Standard Reparameterisation Estimator

✗ biased

[Lee et al., NeurIPS 2018]:

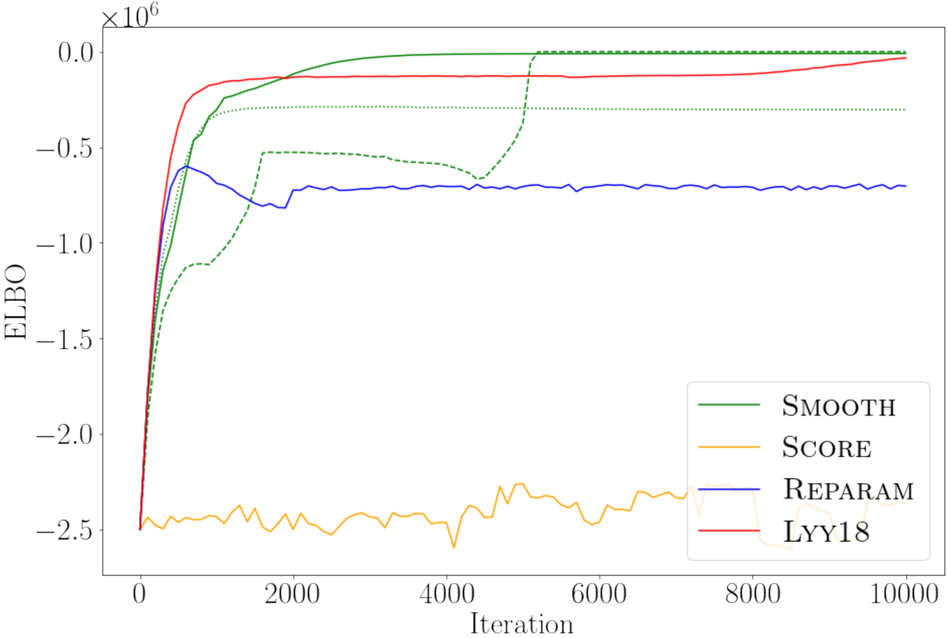
- Fix bias with additional non-trivial *boundary* terms

✗ Only discuss efficient method for *affine* guards

✗ No discussion of PL aspects

✗ Only concerned with unbiasedness, not with overall *correctness* of SGD

temperature



`temperature:` Variance and Cost

Estimator	Cost	Variance
Score	1	1
Reparam		
Smooth (ours)		
Lyy18		

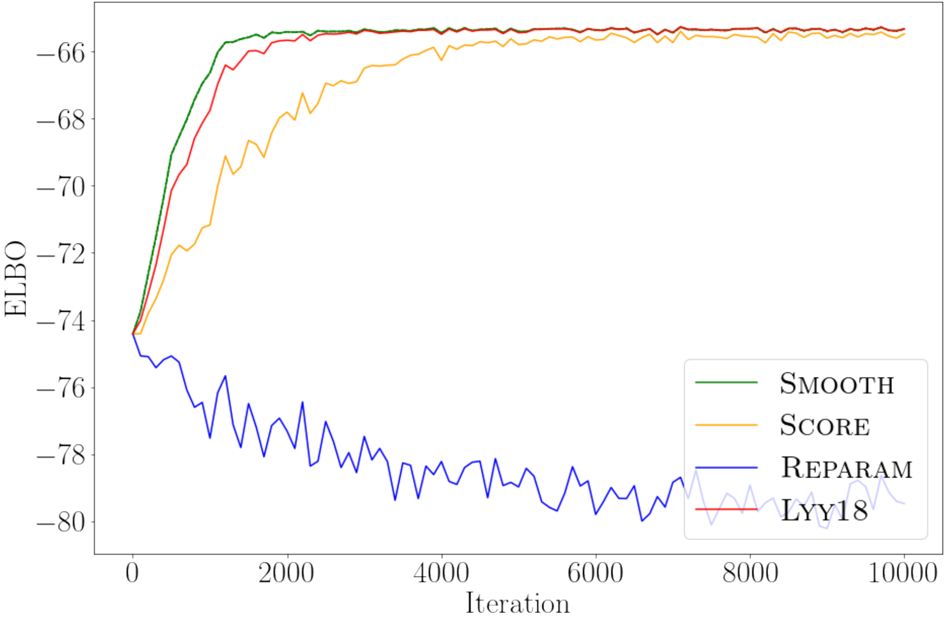
`temperature`: Variance and Cost

Estimator	Cost	Variance
Score	1	1
Reparam	1.28	
Smooth (ours)	1.62	
Lyy18	*9.12*	

`temperature`: Variance and Cost

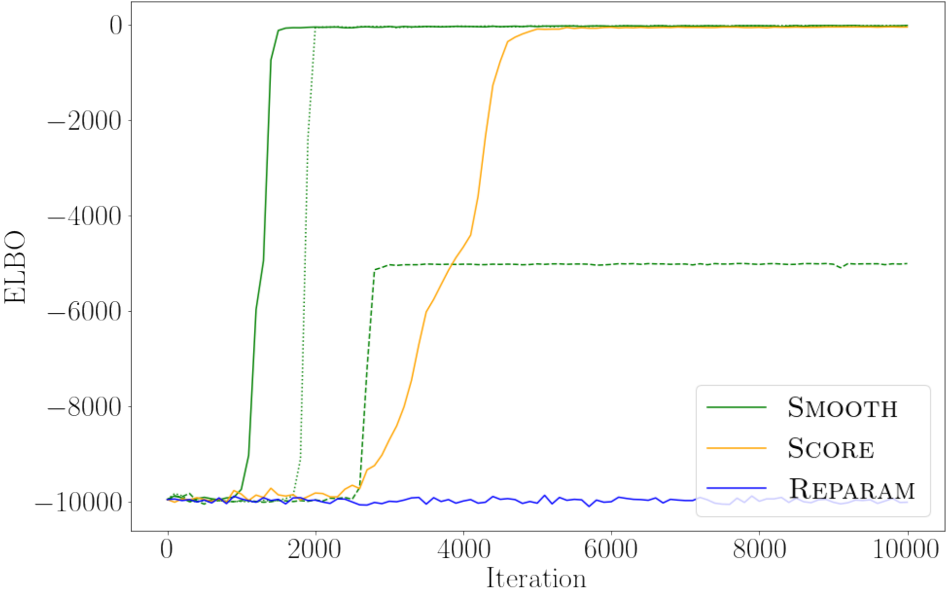
Estimator	Cost	Variance
Score	1	1
Reparam	1.28	1.48e-08
Smooth (ours)	1.62	3.17e-10
Lyy18	*9.12*	1.22e-06

cheating



ELBO
−66
−68
−70
−72
−74
−76
−78
−80
0
2000
4000
6000
8000
10000
Iteration
Smooth
Score
Reparam
Lyy18

xornet


0
−2000
−4000
−6000
−8000
−10000
ELBO
0
2000
4000
6000
8000
10000
Iteration
Smooth
Score
Reparam

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

dominik.wagner@cs.ox.ac.uk

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

- Smoothed semantics avoids bias (caused by branching)
  - categorical model based on Frölicher spaces

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

- Smoothed semantics avoids bias (caused by branching)
    - categorical model based on Frölicher spaces
- Type systems enforce restrictions

dominik.wagner@cs.ox.ac.uk

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

- Smoothed semantics avoids bias (caused by branching)
  - categorical model based on Frölicher spaces
- Type systems enforce restrictions
- Stochastic Gradient Descent is provably *correct*
- Approximations *converge* uniformly

dominik.wagner@cs.ox.ac.uk

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

- Smoothed semantics avoids bias (caused by branching)
    - categorical model based on Frölicher spaces
- Type systems enforce restrictions
- Stochastic Gradient Descent is provably *correct*
- Approximations *converge* uniformly
- *Competitive* on benchmarks

`dominik.wagner@cs.ox.ac.uk`

Fast and Correct Gradient-Based Optimisation for Probabilistic Programming via Smoothing

- Smoothed semantics avoids bias (caused by branching)
    - categorical model based on Frölicher spaces
- Type systems enforce restrictions
- Stochastic Gradient Descent is provably *correct*
- Approximations *converge* uniformly
- *Competitive* on benchmarks

Ongoing Work

- Choice of accuracy coefficient

`dominik.wagner@cs.ox.ac.uk`