Improved Hessian estimation for adaptive random directions stochastic approximation

Danda Sai Koti Reddy*

Joint work with Prashanth L.A.† and Shalabh Bhatnagar*

* Indian Institute of Science, Bangalore
† University of Maryland, College Park

Overview

Simulation Optimization

Random directions stochastic approximation (RDSA) + improved Hessian estimation

Numerical Results

Simulation Optimization

Energy Demand Management

- Consumer demand, energy generation are uncertain
- Objective is to minimize the absolute difference

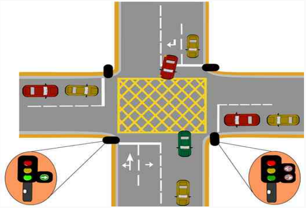
Optimization under uncertainity

Energy Demand Management

- Consumer demand, energy generation are uncertain
- Objective is to minimize the absolute difference

Traffic Signal Control

- Optimal order to switch traffic lights
- Objective is to minimize waiting time



Basic optimization problem

To find θ^* that minimizes the objective function $\mathrm{f}(\theta)$:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathrm{f}(\theta) \tag{1}$$

- $\mathrm{f}\colon \mathbb{R}^{\mathrm{N}} \to \mathbb{R}$ is called the objective function
- θ is tunable N-dimensional parameter
- $\Theta \subseteq \mathbb{R}^{\mathrm{N}}$ is the feasible region in which θ takes values

Classification of optimization problems

Deterministic optimization problem

- Complete information about objective function f
- First and higher order derivatives
- Feasible region

Classification of optimization problems

Deterministic optimization problem

- Complete information about objective function f
- First and higher order derivatives
- Feasible region

Stochastic optimization problem

- We have little knowledge on the structure of f
- f cannot be obtained directly
- $f(\theta) \equiv E_{\xi}[h(\theta, \xi)]$, where ξ comprises the randomness in the system

Difficult to find θ^* only on the basis of noisy samples

Stochastic optimization via simulation

Stochastic optimization deals with highly nonlinear and high dimensional systems. The challenges with these problems are:

- Too complex to solve analytically
- Many simplifying assumptions are required

Stochastic optimization deals with highly nonlinear and high dimensional systems. The challenges with these problems are:

- Too complex to solve analytically
- Many simplifying assumptions are required

A good alternative of modelling and analysis is "Simulation"

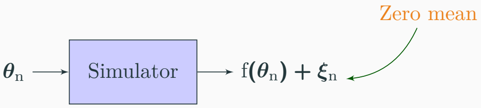


Figure 1: Simulation optimization

Stochastic approximation + Gradient descent

Stochastic analog of gradient descent

$$\theta_{n+1} = \Gamma_\Theta \left[\theta_n - a_n \widehat{\nabla} f(\theta_n)\right] \tag{2}$$

- $\widehat{\nabla} f(\theta_n)$ is a noisy estimate of the gradient $\nabla f(\theta_n)$, and it should satisfy $E\left[\widehat{\nabla} f(\theta_n)\right] - \nabla f(\theta_n) \to 0$
- $\{a_n\}$ are pre-determined step-sizes satisfying:
  $$\sum_{n=1}^{\infty} a_n = \infty, \quad \sum_{n=1}^{\infty} a_n^2 < \infty$$
- Γ_Θ denotes the projection of a point onto Θ

1-slide summary

Related second-order methods

(Spall 2000)[1]	Second-order SPSA (2SPSA)	4 simulations/iteration
(Spall 2009)[2]	2SPSA + feedback	4 simulations/iteration
(Prashanth L.A. et al 2016)[3]	Second-order RDSA (2RDSA)	3 simulations/iteration

Our work

We propose feedback and weighting mechanisms for improving Hessian estimate for 2RDSA algorithm

[1] J. C. Spall (2000), "Adaptive stochastic approximation by the simultaneous perturbation method," IEEE TAC.

[2] J. C. Spall (2009), "Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm," IEEE TAC.

[3] Prashanth L. A. et al. (2016) "Adaptive system optimization using random directions stochastic approximation," IEEE TAC.

Random directions stochastic approximation (RDSA) + improved Hessian estimation

Our algorithm

- Matrix projection
- Gradient estimate

$$\theta_{n+1} = \theta_n - a_n \Upsilon (\overline{H}_n)^{-1} \widehat{\nabla} f(\theta_n) \tag{3}$$

Our algorithm

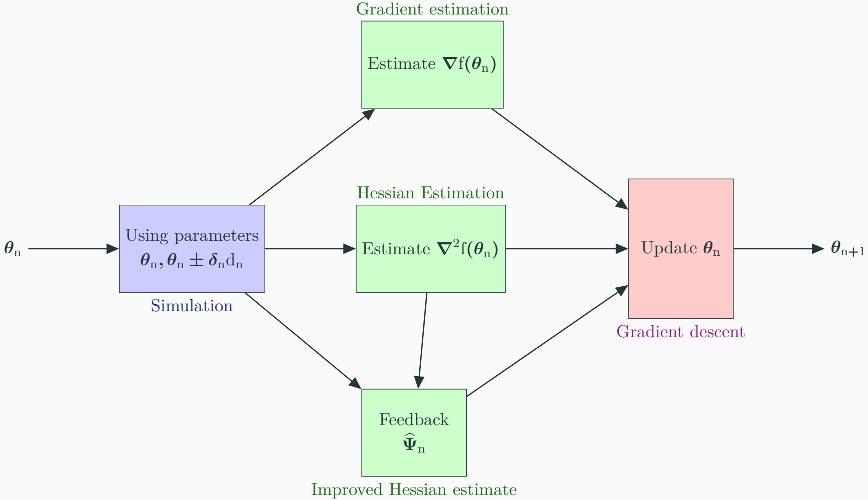
- Matrix projection
- Gradient estimate

$$\theta_{n+1} = \theta_n - a_n \Upsilon(\overline{H}_n)^{-1} \widehat{\nabla} f(\theta_n) \tag{3}$$

$$\overline{H}_n = (1 - b_n)\overline{H}_{n-1} + b_n(\widehat{H}_n - \widehat{\Psi}_n) \tag{4}$$

- Optimal step-sizes
- Hessian estimate
- Feedback term

Overall flow of 2RDSA-IH



RDSA gradient estimate

Function measurements

$$y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+, \quad y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$$

RDSA gradient estimate

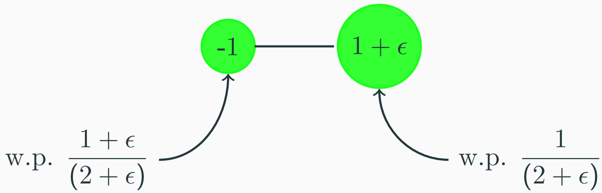
Function measurements

$$y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+, \quad y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$$

Gradient estimate

$$\widehat{\nabla} f(\theta_n) = \frac{1}{1+\epsilon} d_n \left[\frac{y_n^+ - y_n^-}{2\delta_n} \right] \tag{5}$$

Asymmetric Bernoulli distribution for $d_n^i, i = 1, \ldots, N$:



2RDSA Hessian estimate

Function measurements

$$y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+, \;\; y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-, \;\; y_n = f(\theta_n) + \xi_n$$

2RDSA Hessian estimate

Function measurements

$$y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+, \;\; y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-, \;\; y_n = f(\theta_n) + \xi_n$$

Hessian estimate $\widehat{H}_n$

$$\begin{aligned}\widehat{H}_n &= M_n\left(\frac{y_n^+ + y_n^- - 2y_n}{\delta_n^2}\right)\\ &= M_n\left[\left(\frac{f(\theta_n + \delta_n d_n) + f(\theta_n - \delta_n d_n) - 2f(\theta_n)}{\delta_n^2}\right) + \left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2}\right)\right]\\ &= M_n\left(d_n^T \nabla^2 f(\theta_n) d_n + O(\delta_n^2) + \left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2}\right)\right) \qquad (6)\end{aligned}$$

Want to recover $\nabla^2 f(\theta_n)$ from this

Zero-mean

How to choose M_n?

Asymmetric Bernoulli Perturbation

$$M_n = \begin{bmatrix} \frac{1}{\kappa}\left((d_n^1)^2 - (1+\epsilon)\right) & \cdots & \frac{1}{2(1+\epsilon)^2} d_n^1 d_n^N \\ \frac{1}{2(1+\epsilon)^2} d_n^2 d_n^1 & \cdots & \frac{1}{2(1+\epsilon)^2} d_n^2 d_n^N \\ \cdots & \cdots & \cdots \\ \frac{1}{2(1+\epsilon)^2} d_n^N d_n^1 & \cdots & \frac{1}{\kappa}\left((d_n^N)^2 - (1+\epsilon)\right) \end{bmatrix} \tag{7}$$

where $\kappa = \tau \left(1 - \frac{(1+\epsilon)^2}{\tau}\right)$ and $\tau = E(d_n^i)^4 = \frac{(1+\epsilon)(1+(1+\epsilon)^3)}{(2+\epsilon)}$, for any $i = 1, \ldots, N$

Zero-mean feedback term

Zero-mean term

Mean of the Hessian estimate

$$\mathbb{E}\left[\widehat{H}_n \middle| \mathcal{F}_n\right] = \nabla^2 f(\theta_n) + \mathbb{E}\left[\Psi_n(\nabla^2 f(\theta_n)) \middle| \mathcal{F}_n\right] + O(\delta_n^2) + \mathbb{E}\left[\left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2}\right) \middle| \mathcal{F}_n\right] \quad (8)$$

Zero-mean

[1] For any matrix P, $[P]_D$ refers to a matrix that retains only the diagonal entries of P and replaces all the remaining entries with zero

[2] $[P]_N$ to refer to a matrix that retains only the off-diagonal entries of P, while replaces all the diagonal entries with zero

Zero-mean feedback term

Zero-mean term

Mean of the Hessian estimate

$$\mathbb{E}\left[\widehat{H}_n \middle| \mathcal{F}_n\right] = \nabla^2 f(\theta_n) + \mathbb{E}\left[\Psi_n(\nabla^2 f(\theta_n)) \middle| \mathcal{F}_n\right] + O(\delta_n^2) + \mathbb{E}\left[\left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2}\right) \middle| \mathcal{F}_n\right] \tag{8}$$

Zero-mean

Feedback term

$$\Psi_n(H) = [M_n]_D \left(d_n^T [H]_N d_n\right) + [M_n]_N \left(d_n^T [H]_D d_n\right) \tag{9}$$

[1] For any matrix P, $[P]_D$ refers to a matrix that retains only the diagonal entries of P and replaces all the remaining entries with zero

[2] $[P]_N$ to refer to a matrix that retains only the off-diagonal entries of P, while replaces all the diagonal entries with zero

Zero-mean feedback term

Problem

Feedback term is function of current Hessian $\nabla^2 f$

Zero-mean feedback term

Problem

Feedback term is function of current Hessian $\nabla^2 f$

Solution

Use $\overline{H}_{n-1}$ as a proxy for $\nabla^2 f$

$$\widehat{\Psi}_n = \Psi_n(\overline{H}_{n-1}) \tag{10}$$

Step-size optimization

Recall the Hessian recursion, $\overline{\mathrm{H}}_{\mathrm{n}} = (1 - \mathrm{b}_{\mathrm{n}})\overline{\mathrm{H}}_{\mathrm{n}-1} + \mathrm{b}_{\mathrm{n}}(\widehat{\mathrm{H}}_{\mathrm{n}} - \widehat{\Psi}_{\mathrm{n}})$

Step-size optimization

Recall the Hessian recursion, $\overline{H}_n = (1 - b_n)\overline{H}_{n-1} + b_n(\widehat{H}_n - \widehat{\Psi}_n)$

Rewriting the Hessian recursion

$$\overline{H}_n = \sum_{i=0}^{n} \tilde{b}_i(\widehat{H}_i - \widehat{\Psi}_i) \tag{11}$$

Recall the Hessian recursion, $\overline{\mathrm{H}}_n = (1 - b_n)\overline{\mathrm{H}}_{n-1} + b_n(\widehat{\mathrm{H}}_n - \widehat{\Psi}_n)$

Rewriting the Hessian recursion

$$\overline{\mathrm{H}}_n = \sum_{i=0}^{n} \tilde{b}_i(\widehat{\mathrm{H}}_i - \widehat{\Psi}_i) \tag{11}$$

Optimization problem for weights

$$\min_{\{\tilde{b}_i\}} \sum_{i=0}^{n} (\tilde{b}_i)^2 \delta_i^{-4}, \text{ subject to} \tag{12}$$

$$\tilde{b}_i \geq 0 \ \forall i \text{ and } \sum_{i=0}^{n} \tilde{b}_i = 1 \tag{13}$$

Step-size optimization

Above optimization problem solution

$$\tilde{b}_i^* = \delta_i^4 / \sum_{j=0}^{n} \delta_j^4, i = 1, \ldots, n \tag{14}$$

[1] Step-size optimization is a relatively straightforward migration from Spall 2009

Above optimization problem solution

$$\tilde{b}_i^* = \delta_i^4 / \sum_{j=0}^{n} \delta_j^4, i = 1, \dots, n \tag{14}$$

Optimal weights for original Hessian recursion

$$b_i = \delta_i^4 / \sum_{j=0}^{i} \delta_j^4 \tag{15}$$

[1] Step-size optimization is a relatively straightforward migration from Spall 2009

Lemma

(Bias in Hessian estimate) From Prashanth L. A. et al. (2016)[1], we have a.s. that[2], for $i, j = 1, \ldots, N$,

$$\left| \mathbb{E}\left[\widehat{H}_n(i,j) \middle| \mathcal{F}_n \right] - \nabla^2_{ij} f(\theta_n) \right| = O(\delta_n^2) \tag{16}$$

Theorem

(Strong Convergence of Hessian) Under assumptions similar to those for 2SPSA and 2RDSA, we have that

$$\theta_n \to \theta^*, \overline{H}_n \to \nabla^2 f(\theta^*) \text{ a.s. as } n \to \infty$$

[1] Prashanth L. A. et al. (2016) "Adaptive system optimization using random directions stochastic approximation," IEEE TAC.

[2] Here $\widehat{H}_n(i,j)$ and $\nabla^2_{ij} f(\cdot)$ denote the (i,j)th entry in the Hessian estimate $\widehat{H}_n$ and the true Hessian $\nabla^2 f(\cdot)$, respectively.

Numerical Results

Quadratic loss

$$f(\theta) = \theta^T A \theta + b^T \theta \tag{17}$$

Fourth-order loss

$$f(\theta) = \theta^T A^T A \theta + 0.1 \sum_{j=1}^{N} (A\theta)_j^3 + 0.01 \sum_{j=1}^{N} (A\theta)_j^4 \tag{18}$$

Additive Noise : $[\theta^T, 1]Z$, where $Z \approx \mathcal{N}(0, \sigma^2 I_{N+1 \times N+1})$

[1] The implementation is available at https://github.com/prashla/RDSA/archive/master.zip

Normalized MSE (NMSE)

$$\left\|\theta_{\mathrm{n_{end}}} - \theta^*\right\|^2 / \left\|\theta_0 - \theta^*\right\|^2 \tag{19}$$

Normalized loss

$$\mathrm{f}(\theta_{\mathrm{n_{end}}})/\mathrm{f}(\theta_0) \tag{20}$$

Table 1: Normalized loss values for fourth-order objective (18) with noise: simulation budget = 10,000 and standard error from 500 replications shown after ±

Noise parameter $\sigma = 0.1$		
	Regular	Improved Hessian estimation
2SPSA	0.132 ± 0.0267	0.104 ± 0.0355
2RDSA-Unif[1]	0.115 ± 0.0214	0.0271 ± 0.0538
2RDSA-AsymBer	0.0471 ± 0.021	0.0099 ± 0.0014

[1] 2RDSA-Unif uses Unif$[-1, 1]$ with a different M_n

[2] Observation 1: Schemes with improved Hessian estimation performs better than their respective regular schemes

[3] Observation 2: 2RDSA-IH-AsymBer is performing the best overall

Table 2: NMSE values for quadratic objective (17) with noise: simulation budget = 10,000 and standard error from 500 replications shown after ±

Noise parameter $\sigma = 0.1$		
	Regular	Improved Hessian estimation
2SPSA	0.9491 ± 0.0131	0.5495 ± 0.0217
2RDSA-Unif	1.0073 ± 0.0140	0.1953 ± 0.0095
2RDSA-AsymBer	0.1667 ± 0.0095	0.0324 ± 0.0007

[1] Observation 1: Schemes with improved Hessian estimation performs better than their respective regular schemes

[2] Observation 2: 2RDSA-IH-AsymBer is performing the best overall

Conclusions and Future work

Conclusions

- Improved Hessian estimation scheme for the 2RDSA algorithm
- 2RDSA-IH requires only 75% of the simulation cost per-iteration for 2SPSA, 2SPSA-IH

Future work

To derive finite time bounds for 2RDSA-IH

Thank You