Stochastic Newton methods with enhanced Hessian estimation

A THESIS

SUBMITTED FOR THE DEGREE OF

Master of Science (Engineering)

IN THE

Faculty of Engineering

BY

Danda Sai Koti Reddy

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

May, 2017

Declaration of Originality

I, **Danda Sai Koti Reddy**, with SR No. **04-04-00-10-21-14-1-11609** hereby declare that the material presented in the thesis titled

Stochastic Newton methods with enhanced Hessian estimation

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2014-2017**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Advisor Signature

DEDICATED TO

My parents and family members

Acknowledgements

I take this opportunity to express my profound gratitude to the people who have helped and supported me throughout my M.Sc(engg). I thank my guide, Prof. Shalabh Bhatnagar, for his constant and valuable suggestions during my research. Without his constant guidance and support, this thesis would not have been possible. Next I thank Prashanth L.A. for several interesting and useful discussions on our collaborative work. I thank Chandramouli K. for his help during the course work. I am grateful to all the professors of CSA Dept for sharing their knowledge and for their encouragement. I, also, thank the CSA Dept office staff for their invaluable support. Last but not the least, I express my heartfelt gratitude to my parents, sister, brother-in-law, relatives and friends for their love and blessings.

Abstract

Optimization problems involving uncertainties are common in a variety of engineering disciplines such as transportation systems, manufacturing, communication networks, healthcare and finance. The large number of input variables and the lack of a system model prohibit a precise analytical solution and a viable alternative is to employ simulation-based optimization. The idea here is to simulate a few times the stochastic system under consideration while updating the system parameters until a good enough solution is obtained.

Formally, given only noise-corrupted measurements of an objective function, we wish to find a parameter which minimises the objective function. Iterative algorithms using statistical methods search the feasible region to improve upon the candidate parameter. Stochastic approximation algorithms are best suited, most studied and applied algorithms for finding solutions when the feasible region is a continuously valued set. One can use information on the gradient/Hessian of the objective to aid the search process. However, due to lack of knowledge of the noise distribution, one needs to estimate the gradient/Hessian from noisy samples of the cost function obtained from simulation. Simple gradient search schemes take many iterations to converge to a local minimum and are heavily dependent on the choice of step-sizes. Stochastic Newton methods, on the other hand, can counter the ill-conditioning of the objective function as they incorporate second-order information into the stochastic updates. Stochastic Newton methods are often more accurate than simple gradient search schemes.

We propose enhancements to the Hessian estimation scheme used in two recently proposed stochastic Newton methods, based on the ideas of random directions stochastic approximation (2RDSA) [21] and simultaneous perturbation stochastic approximation (2SPSA-3[1]) [6], respectively. The proposed scheme, inspired by [29], reduces the error in the Hessian estimate by

[1]The 3 in the abbereviation of the algorithm is used to indicate that the algorithm in [6] requires 3 function evaluations per iteration.

(i) incorporating a zero-mean feedback term; and **(ii)** optimizing the step-sizes used in the Hessian recursion. We prove that both 2RDSA and 2SPSA-3 with our Hessian improvement scheme converges asymptotically to the true Hessian. The key advantage with 2RDSA and 2SPSA-3 is that they require only 75% of the simulation cost per-iteration for 2SPSA with improved Hessian estimation (2SPSA-IH) [29]. Numerical experiments show that 2RDSA-IH outperforms both 2SPSA-IH and 2RDSA without the improved Hessian estimation scheme.

Publications based on this Thesis

1. D. Sai Koti Reddy, Prashanth, L.A., and Bhatnagar, S. (2016), "Improved Hessian estimation for adaptive random directions stochastic approximation", *55th IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 3682-3687.*

2. D. Sai Koti Reddy, Prashanth, L.A., and Bhatnagar, S. (2017), "Stochastic Newton methods with enhanced Hessian estimation", *Under preparation, 2017.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimization problems can be classified into various categories depending upon the nature of the problem. Consider the problem of finding a θ^* that minimizes the objective function $f(\theta)$:

$$\min_{\theta \in \Theta} f(\theta), \tag{1.1}$$

where $f\colon \mathbb{R}^N \to \mathbb{R}$ is called the objective function, θ is a tunable N-dimensional parameter and $\Theta \subseteq \mathbb{R}^N$ is the constraint set in which θ takes values.

1.1 Classification of optimization problems based on objective function

If we have complete information about f and its derivatives etc., and about the set Θ then (1.1) would be a *deterministic optimization* problem. In the real world unfortunately, many problems do not fall in this class. However, optimization problems involving uncertainties are very common in a variety of engineering disciplines such as transportation systems, manufacturing, networks, healthcare and finance.

The setting in *stochastic optimization*, however, presumes that we have little knowledge on the structure of f and moreover f cannot be obtained directly, but rather is an expectation of another quantity $h(\theta, \xi)$, to which we have access, i.e.,

$$f(\theta) \equiv E_\xi[h(\theta, \xi)], \tag{1.2}$$

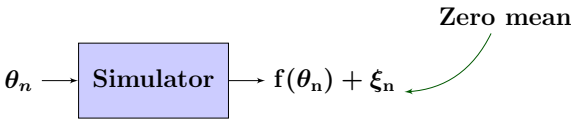


Figure 1.1: Simulation optimization

where ξ comprises the randomness in the system and one is allowed to observe only the $h(\theta, \xi)$ samples, largely because one does not have access to the distribution of noise ξ. These kinds of optimization problems are more challenging because of the added complexity of not knowing f and to find θ^* only on the basis of the aforementioned noisy samples. The large number of input variables and the lack of a precise system model may prohibit analytical solution approaches and a viable alternative is to employ a simulation-based optimization approach. As illustrated in Figure 1.1, the idea here is to simulate a few times the stochastic system under consideration until a good enough solution is obtained. A natural solution approach is to devise an algorithm that incrementally updates the parameter, say θ_n, in the descent direction using the gradient and/or Hessian of the objective f. However, in practice, one can only obtain estimates of the function f through black-box simulation and the challenge is to estimate the gradient and/or Hessian of f from function samples.

Suppose the objective function f has the form $f(\theta) = \sum_{i=1}^{n} E[h_i(X_i)]$, where n denotes the number of stages, X_i is the state of the underlying process in stage i and h_i denotes a stage and state dependent cost function. Then such optimization problems are called *multi-stage problems*. Let $\theta = (\theta_1, \ldots, \theta_n)^T$, with each θ_j being a scalar and let X_i depend on $\theta_1, \ldots, \theta_i$. The idea is that optimization can be done one stage at a time over n stages after observing the state X_i in each stage i. The value θ_i in stage i has a bearing on the cost of all subsequent stages $i + 1, ..., n$. This is a problem of dynamic optimization. Approaches such as dynamic programming can be used to solve this optimization problem.

One may also consider a sub-class of multi-stage problems having an infinite number of stages and with the *long-run average cost function* as objective. The objective function here is of the form:

$$f(\theta) = \lim_{n \to \infty} \frac{1}{n} E\left[\sum_{i=1}^{n} h_i(X_i)\right], \tag{1.3}$$

where X_i is defined as before and is a function of the parameter θ. The objective function (1.3) in most cases is not known analytically. In such cases it becomes difficult for the search procedure to update the search parameter without estimating the cost over an infinitly long trajectory. Such objective functions are common in reinforcement learning applications. Another sub-class of optimization problems are those based on the feasible region Θ, see (1.1).

1.2 Classification of optimization problems based on constrained feasible region

Consider the basic optimization problem in (1.1). The problem depends critically on the structure of the feasible region Θ, where Θ can be of the following types:

(i) *Discrete* **(ii)** *Continuous* **(iii)** *Hybrid* (continuous in some dimensions and discrete in others). The corresponding optimization problems are called discrete, continuous or hybrid optimization problems. These classifications are common to both deterministic and stochastic optimization problems presented in section 1.1.

1.3 Optimization methods via simulation

Optimization via simulation is a fast developing area for both researchers and practioners. As mentioned before, for stochastic optimization problems, simulation is a viable approach. Multiple simulation replications must be performed in order to get a good estimate of $E_\xi[h(\theta_i, \xi)]$, which is the function value at parameter θ_i. The standard approach to estimate this is via the sample mean

$$\overline{f_n}(\theta_i) = \frac{1}{n} \sum_{j=1}^{n} h(\theta, \xi_j), \tag{1.4}$$

where n is the number of simulation replications and ξ_j is the jth sample of randomness. As $n \to \infty$, $\overline{f_n}(\theta_i) \to E_\xi[h(\theta_i, \xi)]$. The primary concerns in stochastic optimization via simulation are **(i)** n must be large enough to get a good estimate of $E_\xi[h(\theta_i, \xi)]$, which has an impact on the final solution that the optimization procedure needs to find; and **(ii)** The above procedure must be applied for many different parameters, i.e., θ's in order to find the best θ. Hence one of the key criteria for comparing different simulation-based optimization approaches is through the

total number of simulations performed by them to attain a specific function value. Note that different approaches may require different number of simulations per iteration. In this section we present methods only for Continuous and Discrete optimization via simulation problems which often occur in practise. However, hybrid optimization problems are not frequently encountered. From now on, we consider the setting of optimization problems mentioned in this section as the stochastic optimization setting presented in section 1.1, i.e., when objective function is an expected value over certain noisy cost measurements.

1.3.1 Discrete optimization via simulation

Optimization problems for which Θ is a discrete set are known as discrete optimization via simulation problems. In the discrete case θ, can take only countable (finite or infinite) number of values. Discrete optimization problems are some times take the form of *combinatorial optimization* problems and they have applications in resource allocation, network routing, policy planning etc. In this section we consider the case of Θ being finite i.e., $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ where $k \in \mathbb{N}$. Discrete optimization problems are divided into two sub-classes when the feasible region is either small (often less than 100) or is large. We present optimization methods in the literature separately for each class.

1.3.1.1 Θ is small

In this case we can simulate all possible solutions and select the best among them. However, unlike in deterministic optimization, simulating the system only once for each parameter is not enough since the objective function is noisy in the stochastic optimization setting. Hence the main question is how to conduct multiple simulations effectively for each parameter in order to determine the best parameter θ. Two popular methods in literature for this case are Ranking and Selection (R&S)[2] and Multiple comparison procedures (MCP) [11]. These methods are well suited for simulation since the underlying assumptions such as normality and independence of observations can be met through careful selection procedures. Traditional methods in this setting are applicable for a maximum set size of 20 while some recent techniques allow more.

1. **Ranking and selection** R&S procedures are statistical procedures specially developed to select the best parameter or subset that contains the best parameter. Majority of

the work in R&S is classified into two categories. The first is called *Indifferent-zone ranking*[2, 20] which is aimed to select the best parameter and the other one is *Subset selection*[14] where the procedures are designed to find a subset of parameters consisting of the best parameter. Again there are two approaches for selecting the best parameter : *the frequentist approach* and *Bayesian approach*. One of the popular approaches in the Bayesian stream is the optimal computing budget allocation (OCBA) procedure [8, 9] where the simulation budget is allocated in a manner that maximizes the posterior probability of correct selection. This method is widely considered the best procedure for small scale discrete optimization.

2. **MCP** Like R&S, MCP attempts to identify the best parameter. But MCPs approach the optimization problem as a statistical inference problem and do not guarantee a decision. Three main classes of MCP that are used in practise are *Multiple Comparison approach* (MCA) , *Multiple Comparison with the Best* (MCB) and *Multiple Comparisons with Control* (MCC). The most popular among these is MCB. In particular, MCB looks at $f(\theta_j) - opt_{i \neq j} f(\theta_i)$ for $j = 1, 2, \ldots, k$ to determine j_* such that $f(\theta_{j_*}) - opt_{i \neq j_*} f(\theta_i) > 0$. Simultaneous confidence intervals $f(\theta_j) - opt_{i \neq j} f(\theta_i)$ for $j = 1, 2, \ldots, k$ can be used to determine j_* by looking for the confidence interval with the lower confidence limit of zero.

For a comprehensive treatment of these methods, see [31, 30].

1.3.1.2 Θ is large

When Θ is large, simulating for each parameter value is expensive. Some sort of search techniques like the ones for deterministic optimization must be applied to avoid simulating for all the parameters while ensuring a high chance of finding the best or good parameters. Some approaches in this category include the following :

1. *Model-based approaches:* Iterative algorithms using statistical methods search the feasible region to improve upon the candidate parameter. One can use *gradient/Hessian information* with respect to the parameter to help the search similar to deterministic case. But due to lack of knowledge of the function in analytical form and distribution of noise one needs to estimate the gradient/Hessian from samples of the function f. However, gradient/Hessian estimation could be quite noisy due to the stochastic nature involved in

these problems. Various approaches that are used in the continuous case can also be used in this setting to estimate the gradient/Hessian when the simulation model is treated as a black box. We will present detailed literature survey of these approaches in sections 1.4 and 1.5.

2. *Metaheuristic:* These are gradient free approaches. They include approaches such as genetic algorithms, evolutionary algorithms, simulated annealing [15], tabu search [13] and cross entropy [23]. Most of these iterative algorithms start with an initial population of parameters and in each iteration elite parameters are selected from the previous population and a better parameter population is generated as search progresses.

1.3.2 Continuous optimization via simulation

Optimization problems for which $\theta \in \Theta$ is a vector of continuous decision variables and Θ is a convex subset of $\mathcal{R}^N$ are called continuous optimization via simulation problems. In these problems θ takes an uncountable number of values. These kinds of problems frequently occur in applications such as model fitting, adaptive control, neural network training, signal processing etc. A natural solution approach is to devise an algorithm that incrementally updates the parameter, say θ_n, in the descent direction using the gradient and/or Hessian of the objective f. Stochastic approximation algorithms are best suited and most studied and used algorithms for solving continuous optimization problems via simulation.

The **stochastic approximation algorithm** (SA) takes the following iterative form:

$$\theta_{n+1} = \Gamma_\Theta \left[\theta_n - a_n \widehat{\nabla} f(\theta_n) \right], \tag{1.5}$$

where θ_n is the solution found at iteration n. Also, $\widehat{\nabla} f(\theta_n)$ is an estimate of the gradient $\nabla f(\theta_n)$ and $\{a_n\}$ is a sequence of positive reals satisfying the following properties: $\sum_{n=1}^{\infty} a_n = \infty$, $\sum_{n=1}^{\infty} a_n^2 < \infty$, and Γ_Θ denotes the projection operator. Under appropriate conditions, as the number of iterations goes to infinity one can guarantee convergence to the local minimum with probability one. The SA algorithm can be considered as a stochastic analog of the gradient decent method which seeks the next solution along the negative gradient direction. The main difference between the SA algorithm and steepest descent algorithm is even though the gradient estimate in the former algorithm is a noisy estimate it requires only weak assumptions (like

$E\left[\widehat{\nabla} f(\theta_n)\right] - \nabla f(\theta_n) \to 0$ at a certain rate) for convergence to local minimum. In practice, the performance of the SA algorithm is quite sensitive to the sequence $\{a_n\}$. For example in the case of $a_n = a/n$, the convergence is highly dependent on the choice of a. However, in practice, one can only obtain estimates of the function f through black-box simulation and the challenge is to estimate the gradient and/or Hessian of f through function samples. In the following sections we will present a brief survey of the existing methods to estimate the gradient and Hessian from function samples.

1.4 Gradient estimation

1.4.1 Finite-difference stochastic approximation (FDSA)

When the simulation model is treated as a block box, the traditional means of forming the estimate of gradient is by using the finite-difference stochastic approximation (FDSA) method. There are mainly two variations in the FDSA scheme. First one is two-sided gradient approximation which involves function measurements $f(\theta_n + \delta_n e_i)$ and $f(\theta_n - \delta_n e_i)$, where e_i denotes ith column of the identity matrix of size $N \times N$. Denote these respective values by y_{ni}^+ and y_{ni}^-, i.e.,

$$y_{ni}^+ = f(\theta_n + \delta_n e_i) + \xi_{ni}^+, \quad y_{ni}^- = f(\theta_n - \delta_n e_i) + \xi_{ni}^-.$$

In the above, we assume the noise vector $(\xi_{ni}^+ - \xi_{ni}^-, i = 1, 2, \ldots, N)^\intercal$ is a martingale difference sequence for every $n \geq 0$, the sequence of the perturbation constants $\{\delta_n, n \geq 0\}$ is a positive and asymptotically vanishing sequence. The gradient estimate of the objective function using this scheme is given by

$$\widehat{\nabla} f(\theta_n) = \begin{pmatrix} \frac{y_{n1}^+ - y_{n1}^-}{2\delta_n} \\ \vdots \\ \frac{y_{nN}^+ - y_{nN}^-}{2\delta_n} \end{pmatrix}. \tag{1.6}$$

The second approach is of one-sided gradient approximation involves function measurements $f(\theta_n + \delta_n e_i)$ and $f(\theta_n)$. Let us denote these values by y_{ni}^+, y_n respectively, i.e., $y_{ni}^+ = f(\theta_n + \delta_n e_i) + \xi_{ni}^+$, $y_n = f(\theta_n) + \xi_n$, where we assume the noise terms $(\xi_{ni}^+ - \xi_n, i = 1, 2, \ldots, N)^\intercal$ satisfy the martingale difference sequence property for $n \geq 0$. The gradient estimate of the objective

function using this scheme is given by

$$\widehat{\nabla} f(\theta_n) = \begin{pmatrix} \frac{y_{n1}^+ - y_n}{\delta_n} \\ \vdots \\ \frac{y_{nN}^+ - y_n}{\delta_n} \end{pmatrix}. \tag{1.7}$$

It is important to determine conditions under which θ_n converges to θ^*, when one uses estimates shown in (1.6) or (1.7). The convergence theory for FDSA algorithm is similar to the convergence theory of Robbins and Monro root finding SA algorithm [22]. However, difficulties arise due to a bias in gradient approximation, i.e, $\mathbb{E}\left[\widehat{\nabla} f(\theta_n)|\mathcal{F}_n\right] - \nabla f(\theta_n)$, where $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denotes the underlying sigma-field. The conditions required for showing convergence of this algorithm are as follows:

(**A1**) $f : \mathbb{R}^N \to \mathbb{R}$ is three-times continuously differentiable[1] with $\left|\nabla^3_{i_1 i_2 i_3} f(\theta)\right| < \alpha_0 < \infty$, for $i_1, i_2, i_3 = 1, \ldots, N$ and for all $\theta \in \mathbb{R}^N$.

(**A2**) $\{\xi_n^+, \xi_n^-, n = 1, 2, \ldots\}$ satisfy $\mathbb{E}\left[\xi_n^+ - \xi_n^- | \mathcal{F}_n\right] = 0$.

(**A3**) For some $\alpha_1, \alpha_2, \zeta > 0$ and for all n, i, $\mathbb{E}\left|\xi_n^{\pm}\right|^{2+\zeta} \leq \alpha_1$, $\mathbb{E}\left|f(\theta_n \pm \delta_n e_i)\right|^{2+\zeta} \leq \alpha_2$.

(**A4**) The step-sizes a_n and perturbation constants δ_n are positive, for all n and satisfy

$$a_n, \delta_n \to 0 \text{ as } n \to \infty, \sum_n a_n = \infty \text{ and } \sum_n \left(\frac{a_n}{\delta_n}\right)^2 < \infty.$$

(**A5**) $\sup_n \|\theta_n\| < \infty$ w.p. 1.

Note that the two-sided gradient estimation scheme requires $2N$ simulations of the objective function in order to obtain a single gradient estimate, whereas the one-sided gradient estimation scheme requires only $N+1$ simulations, where N is the dimension of the vector θ. A requirement of number of simulations to be proportional to N in order to get one gradient estimate makes this procedure highly computationally expensive when N is large. Hence one requires a gradient estimation procedure that is independent of the dimension of θ.

[1]Here $\nabla^3 f(\theta) = \dfrac{\partial^3 f(\theta)}{\partial\theta^\intercal \partial\theta^\intercal \partial\theta^\intercal}$ denotes the third derivate of f at θ and $\nabla^3_{i_1 i_2 i_3} f(\theta)$ denotes the (i_1, i_2, i_3)th entry of $\nabla^3 f(\theta)$, for $i_1, i_2, i_3 = 1, \ldots, N$.

1.4.2 Simultaneous perturbation stochastic approximation (SPSA)

Simultaneous perturbation (SP) methods are a popular and efficient approach for estimating gradient/Hessian from function samples, especially in high dimensional problems - see [5] for a comprehensive treatment of this subject matter. Simultaneous perturbation stochastic approximation (SPSA) is a popular SP method. The first-order SPSA algorithm, henceforth referred to as 1SPSA, was proposed in [28]. The 1SPSA scheme for approximating gradient requires only two simulations or function measurements irrespective of the parameter dimension. The idea in this scheme is to use two function measurements by perturbing all components of the parameter randomly. The function measurements required for this scheme correspond to $\theta_n + \delta_n\Delta_n$ and $\theta_n - \delta_n\Delta_n$, respectively, where $\Delta_n = (\Delta_{n1}, \ldots, \Delta_{nN})^\mathsf{T}$ is any vector consisting of i.i.d, mean-zero, symmetric random variables whose inverse moments are bounded. The simplest and most commonly used perturbation distribution of random variables being used are the symmetric Bernoulli random variables $\Delta_{ni} = \pm 1$ *w.p.* $1/2$, $i = 1, \ldots, N$, $n \geq 0$. The gradient estimate of objective function using this scheme is as follows:

$$\widehat{\nabla} f(\theta_n) = \begin{pmatrix} \frac{f(\theta_n+\delta_n\Delta_n)-f(\theta_n-\delta_n\Delta_n)}{2\delta_n\Delta_{n1}} + \frac{\xi_n^+-\xi_n^-}{2\delta_n\Delta_{n1}} \\ \vdots \\ \frac{f(\theta_n+\delta_n\Delta_n)-f(\theta_n-\delta_n\Delta_n)}{2\delta_n\Delta_{nN}} + \frac{\xi_n^+-\xi_n^-}{2\delta_n\Delta_{nN}} \end{pmatrix}. \tag{1.8}$$

The reason why 1SPSA algorithm is a valid gradient estimation scheme can be seen easily from Taylor expansions of $f(\theta_n + \delta_n\Delta_n$ and $f(\theta_n - \delta_n\Delta_n)$ as under:

$$\begin{aligned} f(\theta_n + \delta_n\Delta_n) &= f(\theta_n) + \delta_n\Delta_n^\mathsf{T}\nabla f(\theta_n) + \frac{\delta_n^2}{2}\Delta_n^\mathsf{T}\nabla^2 f(\theta_n)\Delta_n + o(\delta_n^2), \\ f(\theta_n - \delta_n\Delta_n) &= f(\theta_n) - \delta_n\Delta_n^\mathsf{T}\nabla f(\theta_n) + \frac{\delta_n^2}{2}\Delta_n^\mathsf{T}\nabla^2 f(\theta_n)\Delta_n + o(\delta_n^2). \end{aligned}$$

From the above two equations, the ith component of the gradient approximation (1.8) is given by

$$\frac{f(\theta_n + \delta_n\Delta_n) - f(\theta_n - \delta_n\Delta_n)}{2\delta_n\Delta_{ni}} = \nabla_i f(\theta_n) + \sum_{j=1, j\neq i}^{N} \frac{\Delta_{nj}}{\Delta_{ni}}\nabla_j^2 f(\theta_n) + o(\delta_n). \tag{1.9}$$

Now the conditional expectation of the 1SPSA gradient estimate is given by

$$\mathbb{E}\left[\frac{f(\theta_n+\delta_n\Delta_n)-f(\theta_n-\delta_n\Delta_n)}{2\delta_n\Delta_{ni}}|\mathcal{F}_n\right]=\nabla_i f(\theta_n)+o(\delta_n). \tag{1.10}$$

The asymptotic convergence of 1SPSA algorithm requires one extra assumption other than the conditions specified in the FDSA scheme above. The extra condition for convergence of θ_n to θ^* is as follows:

(**A6**) $\{\Delta_{ni}, i=1,\ldots,N, n=1,2,\ldots\}$ are i.i.d., independent of $\mathcal{F}_n$ and for some $\alpha_3>0$ and $\forall n$, $\mathbb{E}(\Delta_{ni})^{-2}\leq\alpha_3$, for $i=1,2,\ldots,N$.

Since 1SPSA randomly perturbs the parameter vector θ, the number of function measurements required is two, irrespective of the dimension of θ, see that the numerator of (1.8) is the same in all N components of $\widehat{\nabla}f(\theta_n)$. The efficiency of 1SPSA depends on the shape of the objective function f, $\{a_n\}$, $\{\delta_n\}$ and the distribution of perturbation vectors, i.e., Δ_n $\forall n$. In [25], in addition to establishing formal convergence of $\theta_n\to\theta^*$, it was shown that probability distribution of appropriately scaled θ_n is approximately normal (with some mean and covariance matrix) for larger n. The general metric which is used for evaluating these classes of algorithms is asymptotic mean squared error (AMSE), defined as $\mathbb{E}\,\|\theta_n-\theta^*\|^2$. The relative efficiency of 1SPSA when compared to FDSA can be shown when we compare the similar asymptotic normality results of FDSA. That is,

$$\frac{\text{AMSE of 1SPSA}}{\text{AMSE of FDSA}}. \tag{1.11}$$

In [25], it was shown that under reasonably general conditions, and when $\{a_n\}$, $\{\delta_n\}$ are chosen according to the guidelines in [26], 1SPSA and FDSA gives the same AMSE when both the algorithms are run for the same number of iterations. Instead of looking at (1.11), an equivalent way of looking at the relative AMSE is to compare the number of function measurements required by FDSA to achieve the same AMSE as 1SPSA. One can compute it by equating the (1.11) to one. Then it will result in the following:

$$\frac{\text{No.of function measurements for 1SPSA}}{\text{No.of function measurements for FDSA}}=\frac{1}{N}. \tag{1.12}$$

1.4.3 Random directions stochastic approximation (RDSA)

A closely related algorithm to SPSA is the random directions stochastic approximation (RDSA) [16, pp. 58-60]. The gradient estimate in RDSA differs from that in SPSA, both in the construction as well as in the choice of random perturbations. In [16], the random perturbations for 1RDSA were generated by picking samples uniformly on the surface of a sphere and the resulting 1RDSA scheme was found to be inferior to 1SPSA from an asymptotic convergence rate viewpoint - see [10]. The RDSA estimate of the gradient is given by

$$\widehat{\nabla} f(\theta_n) = d_n \left[\frac{y_n^+ - y_n^-}{2\delta_n} \right], \tag{1.13}$$

where $y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+, \quad y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$, $d_n = (d_n^i, \ldots, d_n^N)^\intercal$, and d_n^i, $i = 1, \ldots, N$ are i.i.d random perturbations distributed uniformly on the N-dimensional sphere with radius 1. Like 1SPSA, 1RDSA also requires only two function measurements irrespective of the dimension of parameter θ. The asymptotic convergence of 1RDSA algorithm also requires one extra assumption other than the conditions specified in the FDSA scheme.

(**A7**) $\{d_n^i, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are i.i.d., independent of $\mathcal{F}_n$, and $\forall n$, $\mathbb{E}(d_n d_n^\intercal) = I_{N \times N}$, and for some $\alpha_4 > 0$, $\mathbb{E} {d_n^i}^2 f(\theta_n \pm \delta_n d_n) \leq \alpha_4$ for $i = 1, 2, \ldots, N$.

Recent work in [21] attempts to bridge the gap between 1RDSA and 1SPSA in terms of improving the performance by incorporating random perturbations based on a certain asymmetric Bernoulli distribution as well as another with the i.i.d uniform distribution. However, 1SPSA was found to be still marginally better than 1RDSA.

1.5 Hessian estimation

Stochastic Newton methods can counter the ill-conditioning problem of the objective f as they incorporate second-order information into the update iteration given by

$$\theta_{n+1} = \theta_n - a_n (\overline{H}_n)^{-1} \widehat{\nabla} f(\theta_n), \tag{1.14}$$

where a_n is the step-size that satisfies standard stochastic approximation conditions (see (A12) in Section 2.6), and $\widehat{\nabla} f(\theta_n)$ and $\overline{H}_n$ are estimates of the gradient and Hessian, respectively.

Thus, (1.14) can be considered as the stochastic version of the well-known Newton method for optimization. Stochastic Newton methods are often more accurate than simple gradient search schemes. A key early work in stochastic Newton methods is given in [1]. Also, an alternative analysis could be based on the 'ODE' approach using the results of [24].

In [12], an estimation scheme for $\overline{H}_n$ that uses $O(N^2)$ function samples per-iteration of (1.14) was proposed.

1.5.1 Second-order SPSA (2SPSA)

The number of samples per-iteration for estimating Hessian was brought down to four, regardless of the dimension N, by using the second-order SPSA algorithm (henceforth referred to as 2SPSA). The Hessian estimator is projected to the space of positive definite and symmetric matrices at each iterate for the algorithm to progress along a descent direction. In this scheme two independent perturbation sequences $\Delta_n = (\Delta_{n1}, \ldots, \Delta_{nN})^\mathsf{T}$, $\widehat{\Delta}_n = (\widehat{\Delta}_{n1}, \ldots, \widehat{\Delta}_{nN})^\mathsf{T}$, each consisting of random variables satisfying conditions specified in the 1SPSA scheme are used. The 2SPSA algorithm obtains four function samples y_n^{++}, y_n^{+}, y_n^{-+} and y_n^{-} at $\theta_n + \delta_n\Delta_n + \widehat{\delta}_n\widehat{\Delta}_n$, $\theta_n + \delta_n\Delta_n$, $\theta_n - \delta_n\Delta_n + \widehat{\delta}_n\widehat{\Delta}_n$ and $\theta_n - \delta_n\Delta_n$, where the sequences of the perturbation constants $\{\delta_n, n \geq 0\}, \{\widehat{\delta}_n, n \geq 0\}$ are individually a positive and asymptotically vanishing sequences and the random perturbations $\Delta_n, \widehat{\Delta}_n$ are such that $\{\Delta_{ni}, \widehat{\Delta}_{ni}, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are i.i.d. and independent of the noise sequence. Further, $y_n^{++} = f(\theta_n + \delta_n\Delta_n + \widehat{\delta}_n\widehat{\Delta}_n) + \xi_n^{++}$, $y_n^{+} = f(\theta_n + \delta_n\Delta_n) + \xi_n^{+}$, $y_n^{-+} = f(\theta_n - \delta_n\Delta_n + \widehat{\delta}_n\widehat{\Delta}_n) + \xi_n^{+-}$ and $y_n^{-} = f(\theta_n - \delta_n\Delta_n) + \xi_n^{-}$, where the noise terms $\xi_n^{++}, \xi_n^{+}, \xi_n^{-+}, \xi_n^{-}$ satisfy $\mathbb{E}\left[\xi_n^{++} - \xi_n^{+} - \xi_n^{-+} + \xi_n^{-} \,|\, \mathcal{F}_n\right] = 0$ with $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denoting the underlying sigma-field. The (i, j)th entry of the Hessian estimate $\widehat{H}_n$ in this case is given by

$$\left(\widehat{H}_n\right)_{ij} = \left[\frac{1}{\Delta_{ni}\widehat{\Delta}_{nj}} + \frac{1}{\Delta_{nj}\widehat{\Delta}_{ni}}\right]\left[\frac{y_n^{++} - y_n^{+} - y_n^{-+} + y_n^{-}}{4\delta_n\widehat{\delta}_n}\right]. \tag{1.15}$$

The second-order SPSA algorithm performs an update iteration as follows:

$$\theta_{n+1} = \theta_n - a_n\Upsilon(\overline{H}_n)^{-1}\widehat{\nabla}f(\theta_n), \tag{1.16}$$

$$\overline{H}_n = \frac{n}{n+1}\overline{H}_{n-1} + \frac{1}{n+1}\widehat{H}_n. \tag{1.17}$$

In the above,

- $\widehat{\nabla} f(\theta_n)$ as in (1.8) is the estimate of $\nabla f(\theta_n)$ and this corresponds to (1.8)
- $\widehat{H}_n$ is an estimate of the true Hessian $\nabla^2 f(\cdot)$ at θ_n.
- $\overline{H}_n$ is a smoothed version of $\widehat{H}_n$, which is crucial to ensure convergence.
- Υ is an operator that projects a matrix onto the set of positive definite matrices. Update (1.17) does not necessarily ensure that $\overline{H}_n$ is invertible and without Υ, the parameter update (1.16) may not move along a descent direction - see conditions (A14) in Section 2.6 for the precise requirements on the matrix projection operator.

It is important to determine conditions under which θ_n converges to θ^* and $\overline{H}_n \to H(\theta^*)$. Convergence theory for second-order methods varies significantly to that of first-order methods. One can see [27] for detailed convergence results.

1.5.2 Second-order RDSA (2RDSA)

The basic algorithm in (1.16)–(1.17) is similar to the adaptive scheme analyzed by [21]. However, they have used RDSA by incorporating random perturbations based on an asymmetric Bernoulli distribution as well as those with the i.i.d uniform distribution for the gradient and Hessian estimates (see (2.4) and (2.5), (2.19) and (2.20) for their respective estimates), while [27] employs SPSA. Though 1SPSA was found to be still marginally better than 1RDSA, results in [21] show that a second order RDSA approach (referred to as 2RDSA hereafter) can considerably outperform the corresponding second order SPSA algorithm [27], while requiring only three simulations per iteration of (1.14).

1.6 Contributions of this thesis

The following is a brief description of the contributions of this thesis:

1. Our work in chapter 2 is centred on improving the 2RDSA scheme of [21] by

   I reducing the error in the Hessian estimate through a feedback term; and

   II optimizing the step-sizes used in the Hessian estimation recursion, again with the objective of improving the quality of the Hessian estimate.

Items (I) and (II) are inspired by the corresponding improvements to the Hessian estimation recursion in the enhanced 2SPSA from [29]. We shall refer to the latter algorithm as 2SPSA-IH. While item (II) above is a relatively straightforward migration to the 2RDSA setting, item (I) is a non-trivial contribution, primarily because the Hessian estimate in 2RDSA is entirely different from that in 2SPSA and the feedback term that we incorporate in 2RDSA to improve the Hessian estimate neither correlates with that in 2SPSA nor follows from the analysis in [29]. The advantage with 2RDSA scheme along with proposed improvement to Hessian estimation (henceforth referred to as 2RDSA-IH) is that it requires only 75% of the simulation cost per-iteration for 2SPSA-IH.

We establish that the proposed improvements to Hessian estimation in 2RDSA are such that the resulting 2RDSA-IH algorithm is provably convergent, in particular, the Hessian estimate $\overline{H}_n$ of 2RDSA-IH converges almost surely to the true Hessian. Further, we show empirically that 2RDSA-IH outperforms both 2SPSA-IH of [29] and regular 2RDSA of [21]. Our contribution is important because 2RDSA-IH, like 2RDSA, has lower simulation cost per iteration than 2SPSA and unlike 2RDSA, has an improved Hessian estimation scheme.

2. Our work in chapter 3 is centred on generalising the 2RDSA scheme of [21]. In [21], 2RDSA scheme involving only two perturbation distributions was proposed, those are uniform and asymmetric Bernoulli distributions. In chapter 3, we have proposed gradient and Hessian estimation schemes which are independent of the perturbation distributions. However, perturbations have to satisfy the i.i.d, mean-zero assumptions. Apart from these common assumptions one additional assumption, which is the difference between the squared second moment and fourth moment should be non-zero for Hessian estimation is also required. This generalisation of distributions of random variables for perturbations is important because it enhances the scope of improving the performance of the 1RDSA and 2RDSA schemes by incorporating several other distributions. Some well known distribution that can be used for these schemes as a result of the generalization are Normal (0,1) and truncated Cauchy distributions.

3. Our work in chapter 4 is centred on improving the second order SPSA scheme with three simulations (referred to as 2SPSA-3 hereafter), which was proposed in [6]. These improve-

ments arise as a result of incorporating the zero-mean feedback term in Hessian estimate and optimizing the step-sizes used in the Hessian estimate recursion. Though these ideas are same as the ideas presented in this section for 2RDSA scheme, the contribution is non-trivial because the derivation of feedback term and Hessian estimate is entirely different from 2SPSA, 2RDSA improvements. Moreover, we show, for the special case of a quadratic objective and when there is no noise, that 2SPSA-3 scheme along with proposed improvement to Hessian estimation (henceforth referred to as 2SPSA-3-IH) results in a convergence rate that is on par with the corresponding rate for 2SPSA with Hessian estimation improvements (2SPSA-IH) [29]. The advantage with 2SPSA-3-IH is that it requires only 75% of the simulation cost per-iteration for 2SPSA-IH.

The rest of the thesis is organised as follows: In Chapter 2, we describe the improved Hessian estimation scheme, which is incorporated into the 2RDSA algorithm from [21]. We also present the theoretical results for the 2RDSA algorithm with improved Hessian estimation and the results from numerical experiments. In Chapter 3, we describe the generalised RDSA scheme for both the gradient and Hessian estimates and also present the theoretical convergence results as well. The work related to improving the Hessian estimation scheme 2SPSA-3 is presented in Chapter 4. In Chapter 5, we provide simulation experiments for all the methods proposed in the earlier chapters and finally, in Chapter 6, the concluding remarks and future directions are provided.

Chapter 2

Second-order RDSA with improved hessian estimation (2RDSA-IH)

The second-order RDSA with improved Hessian estimate performs an update iteration as follows:

$$\theta_{n+1} = \theta_n - a_n \Upsilon(\overline{H}_n)^{-1} \widehat{\nabla} f(\theta_n), \tag{2.1}$$

$$\overline{H}_n = (1 - b_n)\overline{H}_{n-1} + b_n(\widehat{H}_n - \widehat{\Psi}_n), \tag{2.2}$$

where $\widehat{\nabla} f(\theta_n)$ is the estimate of $\nabla f(\theta_n)$, $\overline{H}_n$ is an estimate of the true Hessian $\nabla^2 f(\cdot)$, $\Upsilon(\cdot)$ projects any matrix onto the set of positive definite matrices and $\{a_n, n \geq 0\}$ is a step-size sequence that satisfies standard stochastic approximation conditions. There are standard procedures such as Cholesky factorization, see [3], for projecting a given square matrix to the set of positive definite matrices. Moreover, in the vicinity of a local minimum, one expects the Hessian to be positive definite. In such a case, Υ will represent the identity operator.

The recursion (2.1) is identical to that in 2RDSA, while the Hessian estimation recursion (2.2) differs as follows:

(i) $\widehat{\Psi}_n$ is a zero-mean feedback term that reduces the error in Hessian estimate; and

(ii) b_n is a general step-size that we optimize to improve the Hessian estimate.

On the other hand, $\widehat{H}_n$ is identical to that in 2RDSA, i.e., it estimates the true Hessian in each iteration using three function evaluations. For the sake of completeness, we first provide

below the construction for $\widehat{\nabla} f(\theta_n)$ and $\widehat{H}_n$ using asymmetric Bernoulli perturbations, and subsequently we present the feedback term that reduces the error in $\widehat{H}_n$.

Algorithm 1: Structure of 2RDSA-IH algorithm.

Input: initial parameter $\theta_0 \in \mathbb{R}^N$, perturbation constants $\delta_n > 0$, step-sizes $\{a_n, b_n\}$, operator Υ.

1. **Execution**:
   for $n \leftarrow 0, 1, 2, \ldots,$ **do**

   - Generate $\{d_n^i, i = 1, \ldots, N\}$, independent of $\{d_m, m = 0, 1, \ldots, n-1\}$.
   - For any $i = 1, \ldots, N$, d_n^i is distributed according to either asymmetric Bernoulli (see (2.3)) or Uniform $U[-\eta, \eta]$ distributions for some $\eta > 0$ (see Remark 2.1).
     - ***Function evaluation 1***
       Obtain $y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+$.
     - ***Function evaluation 2***
       Obtain $y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$.
     - ***Function evaluation 3***
       Obtain $y_n = f(\theta_n) + \xi_n$.
     - ***Newton step***
       Update the parameter and Hessian as follows:

$$\begin{aligned} \theta_{n+1} &= \theta_n - a_n \Upsilon(\overline{H}_n)^{-1} \widehat{\nabla} f(\theta_n), \\ \overline{H}_n &= (1 - b_n)\overline{H}_{n-1} + b_n(\widehat{H}_n - \widehat{\Psi}_n), \end{aligned}$$

     where $\widehat{H}_n$ and $\widehat{\Psi}_n$ are chosen according to (2.5) and (2.14), respectively.

   end

   return x_n.

Algorithm 1 presents the pseudocode and we describe the individual components of 2RDSA-IH below.

2.1 Function evaluations

Let $\delta_n, n \geq 0$ denote a sequence of diminishing positive real numbers and $d_n = (d_n^1, \ldots, d_n^N)^\intercal$ denote a random perturbation vector at instant n, where the perturbations $\{d_n^i, i = 1, \ldots, N, n =$

$1, 2, \ldots\}$ are i.i.d. and distributed as follows:

$$d_n^i = \begin{cases} -1 & \text{w.p. } \dfrac{(1+\epsilon)}{(2+\epsilon)}, \\ 1+\epsilon & \text{w.p. } \dfrac{1}{(2+\epsilon)}, \end{cases} \tag{2.3}$$

with $\epsilon > 0$ being a constant that can be chosen to be arbitrarily small.

The 2RDSA-IH algorithm obtains three function samples y_n, y_n^+ and y_n^- at θ_n, $\theta_n + \delta_n d_n$ and $\theta_n - \delta_n d_n$, respectively, i.e., $y_n = f(\theta_n) + \xi_n$, $y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+$ and $y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$, where the noise terms ξ_n, ξ_n^+, ξ_n^- satisfy $\mathbb{E}\left[\xi_n^+ + \xi_n^- - 2\xi_n \middle| \mathcal{F}_n\right] = 0$ with $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denoting the underlying sigma-field.

2.2 Gradient estimation

The RDSA estimate of the gradient $\nabla f(\theta_n)$ is given by

$$\widehat{\nabla} f(\theta_n) = \frac{1}{1+\epsilon} d_n \left[\frac{y_n^+ - y_n^-}{2\delta_n}\right], \tag{2.4}$$

2.3 Hessian estimation

$$\widehat{H}_n = M_n \left(\frac{y_n^+ + y_n^- - 2y_n}{\delta_n^2}\right), \text{ where} \tag{2.5}$$

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

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

2.4 Improved Hessian estimation

The Hessian estimate $\widehat{H}_n$ can be simplified as follows:

$$\begin{aligned}\widehat{H}_n &= M_n \left(\frac{y_n^+ + y_n^- - 2y_n}{\delta_n^2} \right) \\ &= M_n \left[\left(\frac{f(\theta_n + \delta_n d_n) + f(\theta_n - \delta_n d_n) - 2f(\theta_n)}{\delta_n^2} \right) + \left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2} \right) \right] \\ &= M_n \left(d_n^\intercal \nabla^2 f(\theta_n) d_n + O(\delta_n^2) + \left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2} \right) \right). \end{aligned} \tag{2.6}$$

For the first term on the RHS above, note that

$$\mathbb{E}[M_n \left(d_n^\intercal \nabla^2 f(\theta_n) d_n \right) \mid \mathcal{F}_n] = \mathbb{E}\left[M_n \times \left(\sum_{i=1}^{N-1} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) \right) \middle| \mathcal{F}_n \right]. \tag{2.7}$$

In analyzing the lth diagonal term in the above expression, the following zero-mean term appears (see the proof of Lemma 4 in [21]):

$$\mathbb{E}\left[([M_n]_D)_{l,l} \left(2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) \right) \middle| \mathcal{F}_n \right] = 0, \tag{2.8}$$

where for any matrix M, $[M]_D$ refers to a matrix that retains only the diagonal entries of M and replaces all the remaining entries with zero, and $([M]_D)_{i,j}$ refers to the (i,j)th entry in $[M]_D$. We shall also use $[M]_N$ to refer to a matrix that retains only the off-diagonal entries of M, while replaces all the diagonal entries with zero.

The term on the LHS in (2.8), denoted by $\Psi_n^1(\nabla^2 f(\theta_n))$, can be written in matrix form as follows:

$$\Psi_n^1(\nabla^2 f(\theta_n)) = [M_n]_D \left(d_n^\intercal \left[\nabla^2 f(\theta_n) \right]_N d_n \right). \tag{2.9}$$

In analyzing the off-diagonal term ((k,l) where $k < l$) of (2.7), the following zero-mean

term appears:

$$\mathbb{E}\left[([M_n]_N)_{k,l}\left(\sum_{i=1}^{N}(d_n^i)^2\nabla_{ii}^2 f(\theta_n)\right)\middle|\mathcal{F}_n\right]=0. \tag{2.10}$$

The term on the LHS above, denoted by $\Psi_n^2(\nabla^2 f(\theta_n))$, can be written in matrix form as follows:

$$\Psi_n^2(\nabla^2 f(\theta_n)) = [M_n]_N\left(d_n^{\mathrm{T}}\left[\nabla^2 f(\theta_n)\right]_D d_n\right). \tag{2.11}$$

From the foregoing, the per-iteration Hessian estimate $\widehat{H}_n$ can be re-written as follows:

$$\mathbb{E}\left[\widehat{H}_n\middle|\mathcal{F}_n\right]=\nabla^2 f(\theta_n)+\mathbb{E}\left[\Psi_n(\nabla^2 f(\theta_n))\middle|\mathcal{F}_n\right]+O(\delta_n^2)+\mathbb{E}\left[\left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{\delta_n^2}\right)\middle|\mathcal{F}_n\right], \tag{2.12}$$

where, for any matrix H,

$$\begin{aligned}\Psi_n(H) &= \Psi_n^1(H)+\Psi_n^2(H)\\ &= [M_n]_D\left(d_n^{\mathrm{T}}[H]_N d_n\right)+[M_n]_N\left(d_n^{\mathrm{T}}[H]_D d_n\right).\end{aligned} \tag{2.13}$$

In the RHS of (2.12), it is easy to see that the second term involving Ψ_n and the last term involving the noise are zero-mean. Moreover, since the noise is bounded by assumption, the last term in (2.12) vanishes asymptotically at the rate $O(\delta_n^{-2})$. So, the error in estimating the Hessian is due to the second term, which involves the perturbations d_n. This motivates the term $\widehat{\Psi}_n$ in the update rule (2.1).

Given that we operate in a simulation optimization setting, which implies $\nabla^2 f$ is not known, we construct the feedback term $\widehat{\Psi}_n$ in (2.1) by using $\overline{H}_{n-1}$ as a proxy for $\nabla^2 f$, i.e.,

$$\widehat{\Psi}_n = \Psi_n(\overline{H}_{n-1}). \tag{2.14}$$

2.5 Step-size optimization

Unlike the feedback term, adapting the idea of optimizing the step-sizes for 2RDSA is relatively straightforward from the corresponding approach for 2SPSA in [29]. The difference here is that there exists only one N-dimensional perturbation vector d_n in our setting, while 2SPSA

required two such vectors. This in turn implies that only the perturbation constant δ_n is needed in optimizing b_n.

The optimal choice for b_n in (2.2) is the following:

$$b_i = \delta_i^4 / \sum_{j=0}^{i} \delta_j^4. \tag{2.15}$$

The main idea behind the above choice is provided below. From (2.12), we can infer that

$$\mathbb{E} \parallel \widehat{H}_n \parallel^2 \leq \frac{C}{\delta_n^4} \text{ for some } C < \infty.$$

This is because the third term in (2.12) vanishes asymptotically, while the fourth term there dominates asymptotically. Moreover, the noise factors in the fourth term in (2.12) are bounded above due to (A16) and independent of n, leaving the δ_n^2 term in the denominator there.

So, the optimization problem to be solved at instant n is as follows:

$$\min_{\{\tilde{b}_k\}} \sum_{i=0}^{n} (\tilde{b}_k)^2 \delta_i^{-4}, \text{ subject to} \tag{2.16}$$

$$\tilde{b}_i \geq 0 \ \forall i \text{ and } \sum_{i=0}^{n} \tilde{b}_i = 1. \tag{2.17}$$

The optimization variable $\tilde{b}_i$ from the above is related to the Hessian recursion (2.2) as follows:

$$\overline{H}_n = \sum_{i=0}^{n} \tilde{b}_k (\widehat{H}_i - \widehat{\Psi}_i). \tag{2.18}$$

The solution to (2.16) is achieved for $\tilde{b}_i^* = \delta_i^4 / \sum_{j=0}^{n} \delta_j^4, i = 1, \ldots, n$. The optimal choice $\tilde{b}_i^*$ can be translated to the step-sizes b_i, leading to (2.15).

Remark 2.1. ***(Uniform perturbations)*** *In [21], the authors suggest two alternatives for the distribution of random perturbations d_n: the asymmetric Bernoulli, which we described earlier and uniform that we outline next.*

Choose d_n^i, $i = 1, \ldots, N$ to be i.i.d. $U[-\eta, \eta]$ for some $\eta > 0$, where $U[-\eta, \eta]$ denotes the uniform distribution on the interval $[-\eta, \eta]$. Then, the RDSA estimate of the gradient is given

by

$$\widehat{\nabla} f(\theta_n) = \frac{3}{\eta^2} d_n \left[\frac{y_n^+ - y_n^-}{2\delta_n} \right]. \tag{2.19}$$

The Hessian estimate in this case is given by

$$\widehat{H}_n = M_n \left(\frac{y_n^+ + y_n^- - 2y_n}{\delta_n^2} \right), \text{ where} \tag{2.20}$$

$$M_n = \frac{9}{2\eta^4} \begin{bmatrix} \frac{5}{2}\left((d_n^1)^2 - \frac{\eta^2}{3}\right) & \cdots & d_n^1 d_n^N \\ d_n^2 d_n^1 & \cdots & d_n^2 d_n^N \\ \cdots & \cdots & \cdots \\ d_n^N d_n^1 & \cdots & \frac{5}{2}\left((d_n^N)^2 - \frac{\eta^2}{3}\right) \end{bmatrix}.$$

The feedback term in (2.14) can be easily extended to the case of uniform perturbations by using the M_n as defined above instead of that for the asymmetric Bernoulli case.

2.6 Convergence analysis

We make the same assumptions as those used in the analysis of [21], with a few minor alterations. The assumptions are listed below:

(**A8**) The function f is four-times differentiable[1] with $\left|\nabla^4_{i_1 i_2 i_3 i_4} f(\theta)\right| < \infty$, for $i_1, i_2, i_3, i_4 = 1, \ldots, N$ and for all $\theta \in \mathbb{R}^N$.

(**A9**) For each n and all θ, there exists a $\rho > 0$ not dependent on n and θ, such that $(\theta - \theta^*)^\intercal \bar{f}_n(\theta) \geq \rho \left\|\theta_n - \theta\right\|$, where $\bar{f}_n(\theta) = \Upsilon(\overline{H}_n)^{-1} \nabla f(\theta)$.

(**A10**) $\{\xi_n, \xi_n^+, \xi_n^-, n = 1, 2, \ldots\}$ are such that, for all n, $\mathbb{E}\left[\xi_n^+ + \xi_n^- - 2\xi_n \middle| \mathcal{F}_n\right] = 0$, where $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denotes the underlying sigma-field..

(**A11**) $\{d_n^i, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are i.i.d. and independent of $\mathcal{F}_n$.

[1]Here $\nabla^4 f(\theta) = \dfrac{\partial^4 f(\theta)}{\partial\theta^\intercal \partial\theta^\intercal \partial\theta^\intercal \partial\theta^\intercal}$ denotes the fourth derivate of f at θ and $\nabla^4_{i_1 i_2 i_3 i_4} f(\theta)$ denotes the (i_1, i_2, i_3, i_4)th entry of $\nabla^4 f(\theta)$, for $i_1, i_2, i_3, i_4 = 1, \ldots, N$.

(**A12**) The step-sizes a_n and perturbation constants δ_n are positive, for all n and satisfy

$$a_n, \delta_n \to 0 \text{ as } n \to \infty, \sum_n a_n = \infty \text{ and } \sum_n \left(\frac{a_n}{\delta_n}\right)^2 < \infty.$$

(**A13**) For each $i = 1, \ldots, N$ and any $\rho > 0$, $P(\{\bar{f}_{ni}(\theta_n) \geq 0 \text{ i.o}\} \cap \{\bar{f}_{ni}(\theta_n) < 0 \text{ i.o}\} \mid \{|\theta_{ni} - \theta_i^*| \geq \rho \quad \forall n\}) = 0$.

(**A14**) The operator Υ satisfies $\delta_n^2 \Upsilon(H_n)^{-1} \to 0$ a.s. and $E(\|\Upsilon(H_n)^{-1}\|^{2+\zeta}) \leq \rho$ for some $\zeta, \rho > 0$.

(**A15**) For any $\tau > 0$ and nonempty $S \subseteq \{1, \ldots, N\}$, there exists a $\rho'(\tau, S) > \tau$ such that

$$\limsup_{n \to \infty} \left| \frac{\sum_{i \notin S} (\theta - \theta^*)_i \bar{f}_{ni}(\theta)}{\sum_{i \in S} (\theta - \theta^*)_i \bar{f}_{ni}(\theta)} \right| < 1 \text{ a.s.}$$

for all $|(\theta - \theta^*)_i| < \tau$ when $i \notin S$ and $|(\theta - \theta^*)_i| \geq \rho'(\tau, S)$ when $i \in S$.

(**A16**) For some $\alpha_5, \alpha_6 > 0$ and for all n, $\mathbb{E}{\xi_n}^2 \leq \alpha_5$, $\mathbb{E}\xi_n^{\pm 2} \leq \alpha_5$, $\mathbb{E}f(\theta_n)^2 \leq \alpha_6$, $\mathbb{E}f(\theta_n \pm \delta_n d_n)^2 \leq \alpha_6$ and $\mathbb{E}\left(\left\|\Upsilon(\overline{H}_n)\right\|^2 \mid \mathcal{F}_n\right) \leq \alpha_6$.

(**A17**) $\delta_n = \frac{\delta_0}{(n+1)^\varsigma}$, where $\delta_0 > 0$ and $0 < \varsigma \leq 1/8$.

The reader is referred to Section II-B of [21] for a detailed discussion of the above assumptions. We remark here that (A8)-(A15) are identical to assumptions in [21], while (A16) and (A17) introduce minor additional requirements on $\left\|\Upsilon(\overline{H}_n)\right\|^2$ and δ_n, respectively and these are inspired from [29].

Lemma 2.1. *(**Bias in Hessian estimate**) Under (A8)-(A17), with $\widehat{H}_n$ defined according to (2.5), we have a.s. that*[1]*, for $i, j = 1, \ldots, N$,*

$$\left| \mathbb{E}\left[\widehat{H}_n(i,j) \middle| \mathcal{F}_n \right] - \nabla^2_{ij} f(\theta_n) \right| = O(\delta_n^2). \tag{2.21}$$

Proof. See Lemma 4 in [21]. □

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

Theorem 2.1. *(**Strong Convergence of Hessian**) Under (A8)-(A17), we have that*

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

In the above, θ_n and $\overline{H}_n$ are updated according to (2.1) and (2.2), respectively, $\widehat{H}_n$ is defined according to (2.5) and the step-sizes b_n are chosen as suggested in (2.15).

Proof. The first part of the claim regarding θ_n follows in exactly the same fashion as the proof of Theorem 5 in [21]. For proving the claim regarding $\overline{H}_n$, we closely follow the approach used to prove a corresponding result for 2SPSA (see Theorem 1 in [29]). The first step is to prove the following:

$$\sum_{k=0}^{n} \frac{\delta_k^4 \left(\widehat{H}_k - \widehat{\Psi}_k - \mathbb{E}(\widehat{H}_k \mid \mathcal{F}_k) \right)}{\sum_{i=0}^{n} \delta_i^4} \to 0. \tag{2.22}$$

By a completely similar argument to that used in the proof of Theorem 1 in [29], we obtain: For any $i, j = 1, \ldots, N$,

$$\mathbb{E}\left[\left((\widehat{H}_k)_{i,j} - (\widehat{\Psi}_k)_{i,j} - \mathbb{E}((\widehat{H}_k)_{i,j} \mid \mathcal{F}_k) \right)^2 \right] = O(\delta_k^{-4}).$$

Now (2.22) follows by an application of Kronecker's Lemma along with the martingale convergence theorem (see Theorem 6.2.1 of [17]).

From Lemma 2.1, we have

$$\mathbb{E}[\widehat{H}_k \mid \mathcal{F}_k] = \nabla^2 f(\theta_k) + O(\delta_k^2) \text{ a.s.}$$

Since the Hessian is continuous near θ_n and θ_n converges almost surely to θ^*, we have

$$\begin{aligned}
\sum_{k=0}^{n} \frac{\delta_k^4 \left(\mathbb{E}(\widehat{H}_k \mid \mathcal{F}_k) \right)}{\sum_{i=0}^{n} \delta_i^4} &= \sum_{k=0}^{n} \frac{\delta_k^4 \left(\nabla^2 f(\theta_k) + O(\delta_k^2) \right)}{\sum_{i=0}^{n} \delta_i^4} \\
&= \sum_{k=0}^{n} \frac{\delta_k^4 \left(\nabla^2 f(\theta^*) + o(1) \right)}{\sum_{i=0}^{n} \delta_i^4} \\
&\to \nabla^2 f(\theta^*) \text{ a.s. as } n \to \infty.
\end{aligned}$$

The last step above follows from Toeplitz Lemma (see p. 89 of [17]) after observing that $\sum_{i=0}^{n} \delta_i^4 \to \infty$ due to (A17). The main claim now follows since

$$\overline{H}_n = \sum_{k=0}^{n} \frac{\delta_k^4 \left(\widehat{H}_k - \Psi_k \right)}{\sum_{i=0}^{n} \delta_i^4}.$$

□

Chapter 3

Generalised RDSA

Our work in this chapter is centred on generalising the 2RDSA scheme of [21]. In [21], the RDSA scheme involving only two perturbation distributions was proposed, those are uniform and asymmetric Bernoulli distributions. We propose gradient and Hessian estimation schemes which are independent of the perturbation distributions. However, perturbations have to satisfy the i.i.d, mean-zero assumptions. Apart from these common assumptions, an additional assumption required is that the difference between the squared second moment and fourth moment should be non zero for the Hessian estimation. This generalisation of distributions of random variables for perturbations is important because it enhances the scope of improving the performance of the 1RDSA and 2RDSA schemes by incorporating several other distributions. Some well known distribution that can be used for these schemes as a result of the generalisation are Normal (0,1) and truncated Cauchy distributions. We will prove that the gradient and Hessian estimates as a result of the generalisation are indeed asymptotically unbiased. The improved Hessian estimation scheme proposed in chapter 2 for the 2RDSA algorithm with uniform and asymmetric Bernoulli distributions can be easily extended to the generalised RDSA scheme.

3.1 Function evaluations

Let $\delta_n, n \geq 0$ denote a sequence of diminishing positive real numbers and $d_n = (d_n^1, \ldots, d_n^N)^\intercal$ denote a random perturbation vector at instant n, where the perturbations $\{d_n^i, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are any i.i.d., mean zero random variables.

The generalised 2RDSA algorithm obtains three function samples y_n, y_n^+ and y_n^- at θ_n,

$\theta_n + \delta_n d_n$ and $\theta_n - \delta_n d_n$, respectively, i.e., $y_n = f(\theta_n) + \xi_n$, $y_n^+ = f(\theta_n + \delta_n d_n) + \xi_n^+$ and $y_n^- = f(\theta_n - \delta_n d_n) + \xi_n^-$, where the noise terms ξ_n, ξ_n^+, ξ_n^- satisfy $\mathbb{E}\left[\xi_n^+ + \xi_n^- - 2\xi_n \middle| \mathcal{F}_n\right] = 0$ with $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denoting the underlying sigma-field.

3.2 Generalised RDSA gradient estimate

The generalised RDSA estimate of the gradient $\nabla f(\theta_n)$ is given by

$$\widehat{\nabla} f(\theta_n) = \frac{1}{\lambda} d_n \left[\frac{y_n^+ - y_n^-}{2\delta_n}\right], \tag{3.1}$$

where $\lambda = \mathbb{E}(d_n^i)^2$ and the perturbations d_n^i, $i = 1, \ldots, N$ are i.i.d and zero-mean random variables.

3.3 Generalised RDSA Hessian estimate

$$\widehat{H}_n = M_n \left(\frac{y_n^+ + y_n^- - 2y_n}{\delta_n^2}\right), \text{ where} \tag{3.2}$$

$$M_n = \begin{bmatrix} \frac{1}{\kappa}\left((d_n^1)^2 - \lambda\right) & \cdots & \frac{1}{2\lambda^2} d_n^1 d_n^N \\ \frac{1}{2\lambda^2} d_n^2 d_n^1 & \cdots & \frac{1}{2\lambda^2} d_n^2 d_n^N \\ \cdots & \cdots & \cdots \\ \frac{1}{2\lambda^2} d_n^N d_n^1 & \cdots & \frac{1}{\kappa}\left((d_n^N)^2 - \lambda\right) \end{bmatrix},$$

where $\lambda = \mathbb{E}(d_n^i)^2$, $\tau = E(d_n^i)^4$, and $\kappa = (\tau - \lambda^2)$ for any $i = 1, \ldots, N$. From the above Hessian estimation scheme it is easy to see that perturbations d_n^i, $i = 1, \ldots, N$ should satisfy $\kappa \neq 0$. However, many distributions of the random variables satisfy this requirement.

3.4 Convergence analysis

(**A18**) For some $\alpha_1, \alpha_2, \zeta > 0$ and for all n, $\mathbb{E}\left|\xi_n^{\pm}\right|^{2+\zeta} \leq \alpha_1$, $\mathbb{E}\left|f(\theta_n \pm \delta_n d_n)\right|^{2+\zeta} \leq \alpha_2$.

(**A19**) $\{d_n^i, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are i.i.d., zero-mean and independent of $\mathcal{F}_n$. For some $\eta_1, \eta_2, \zeta > 0$ and for all n, $\mathbb{E}(d_n^i)^{2+\zeta} \leq \eta_1, \mathbb{E}(d_n^i)^{4+\zeta} \leq \eta_2$.

Lemma 3.1. *(**Bias in the gradient estimate**) Under (A1),(A4),(A5),(A18),(A19) for*

$\widehat{\nabla} f(\theta_n)$ *defined according to* (3.1) *, we have a.s. that*[1]

$$\left|\mathbb{E}\left[\widehat{\nabla}_i f(\theta_n)\middle|\mathcal{F}_n\right] - \nabla_i f(\theta_n)\right| = O(\delta_n^2), \quad \text{for } i = 1, \ldots, N. \tag{3.3}$$

Proof. We use the proof techniques of [25],[21] (see Lemma 1) in order to prove the main claim here.

Notice that

$$\mathbb{E}\left[d_n\left(\frac{y_n^+ - y_n^-}{2\delta_n}\right)\middle|\mathcal{F}_n\right] = \mathbb{E}\left[d_n\left(\frac{f(\theta_n + \delta_n d_n) - f(\theta_n - \delta_n d_n)}{2\delta_n}\right)\middle|\mathcal{F}_n\right].$$

The last equality above follows from the fact that $\mathbb{E}\left[d_n\left(\frac{\xi_n^+ - \xi_n^-}{2\delta_n}\right)\middle|\mathcal{F}_n\right] = 0$ from (A2) and (A18). We now analyse the term on the RHS above. Let $\nabla^2 f(\cdot)$ denote the Hessian of f. By Taylor's series expansions, we obtain, a.s.,

$$f(\theta_n \pm \delta_n d_n) = f(\theta_n) \pm \delta_n d_n^{\mathrm{T}} \nabla f(\theta_n) + \frac{\delta_n^2}{2} d_n^{\mathrm{T}} \nabla^2 f(\theta_n) d_n \pm \frac{\delta_n^3}{6} \nabla^3 f(\tilde{\theta}_n^+)(d_n \otimes d_n \otimes d_n),$$

where $\otimes$ denotes the Kronecker product and $\tilde{\theta}_n^+$ (resp. $\tilde{\theta}_n^-$) are on the line segment between θ_n and $(\theta_n + \delta_n d_n)$ (resp. $(\theta_n - \delta_n d_n)$). Hence,

$$\begin{aligned}\mathbb{E}\left[d_n\left(\frac{f(\theta_n + \delta_n d_n) - f(\theta_n - \delta_n d_n)}{2\delta_n}\right)\middle|\mathcal{F}_n\right] &= \mathbb{E}\left[d_n d_n^{\mathrm{T}} \nabla f(\theta_n)\middle|\mathcal{F}_n\right] \\ &\quad + \mathbb{E}\left[\frac{\delta_n^2}{12} d_n (\nabla^3 f(\tilde{\theta}_n^+) + \nabla^3 f(\tilde{\theta}_n^-))(d_n \otimes d_n \otimes d_n)\middle|\mathcal{F}_n\right]. \end{aligned} \tag{3.4}$$

The first term on the RHS above can be simplified as follows:

$$\begin{aligned}\mathbb{E}\left[d_n d_n^{\mathrm{T}} \nabla f(\theta_n)\middle|\mathcal{F}_n\right] &= \mathbb{E}\left[d_n d_n^{\mathrm{T}}\right] \nabla f(\theta_n) \\ &= \lambda \nabla f(\theta_n). \end{aligned} \tag{3.5}$$

In the above, the first equality follows from (A19) and the last equality in (3.5) follows from $\mathbb{E}[(d_n^i)^2] = \lambda$ and $\mathbb{E}[d_n^i d_n^j] = \mathbb{E}[d_n^i]\mathbb{E}[d_n^j] = 0$ for $i \neq j$.

Now, the lth coordinate of the second term in the RHS of (3.4) can be upper-bounded as

[1]Here $\widehat{\nabla}_i f(\theta_n)$ and $\nabla_i f(\theta_n)$ denote the ith coordinates in the gradient estimate $\widehat{\nabla} f(\theta_n)$ and true gradient $\nabla f(\theta_n)$, respectively.

follows:

$$\left|\mathbb{E}\left[\frac{\delta_n^2}{12}d_n^l(\nabla^3 f(\tilde{\theta}_n^+)+\nabla^3 f(\tilde{\theta}_n^-))(d_n\otimes d_n\otimes d_n)\middle|\mathcal{F}_n\right]\right| \leq \frac{\alpha_0\delta_n^2}{6}\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\mathbb{E}\left|\left(d_n^l d_n^{i_1} d_n^{i_2} d_n^{i_3}\right)\right| \leq \frac{\alpha_0\delta_n^2\eta N^3}{6}. \tag{3.6}$$

The first inequality above follows from (A1), while the second inequality follows from (A19) and considering $\eta = \max\{\eta_1^2, \eta_2\}$. The claim follows by plugging (3.5) and (3.6) into (3.4) .

□

Lemma 3.2. *(**Bias in Hessian estimate**) Under (A8)-(A10),(A12)-(A17),(A19) with $\widehat{H}_n$ defined according to (3.2) , we have a.s. that*[1]*, for $i, j = 1, \ldots, N$,*

$$\left|\mathbb{E}\left[\widehat{H}_n(i,j)\middle|\mathcal{F}_n\right] - \nabla_{ij}^2 f(\theta_n)\right| = O(\delta_n^2). \tag{3.7}$$

Proof. We use the proof techniques of [21] (see Lemma 4) in order to prove the main claim here. However, we provide the proof for a general RDSA scheme in which perturbations need not be asymmetric Bernoulli or uniform as in the case of [21].

By a Taylor's series expansion, we obtain

$$f(\theta_n \pm \delta_n d_n) = f(\theta_n) \pm \delta_n d_n^{\mathrm{T}}\nabla f(\theta_n) + \frac{\delta_n^2}{2}d_n^{\mathrm{T}}\nabla^2 f(\theta_n)d_n \pm \frac{\delta_n^3}{6}\nabla^3 f(\theta_n)(d_n\otimes d_n\otimes d_n) + \frac{\delta_n^4}{24}\nabla^4 f(\tilde{\theta}_n^+)(d_n\otimes d_n\otimes d_n\otimes d_n).$$

The fourth-order term in each of the expansions above can be shown to be of order $O(\delta_n^4)$ using (A8) and arguments similar to those in Lemma 3.1. Hence,

$$\frac{f(\theta_n+\delta_n d_n)+f(\theta_n-\delta_n d_n)-2f(\theta_n)}{\delta_n^2} = d_n^{\mathrm{T}}\nabla^2 f(\theta_n)d_n + O(\delta_n^2) = \sum_{i=1}^{N}\sum_{j=1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) + O(\delta_n^2)$$

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

$$= \sum_{i=1}^{N} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) + O(\delta_n^2).$$

Now, taking the conditional expectation of the Hessian estimate $\widehat{H_n}$ and observing that $\mathbb{E}[\xi_n^+ + \xi_n^- - 2\xi_n \mid \mathcal{F}_n, d_n] = 0$ by (A10), we obtain the following:

$$\mathbb{E}[\widehat{H}_n \mid \mathcal{F}_n] = \mathbb{E}\left[M_n \left(\sum_{i=1}^{N-1} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) \quad + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) + O(\delta_n^2) \right) \middle| \mathcal{F}_n \right]. \tag{3.8}$$

Note that the $O(\delta_n^2)$ term inside the conditional expectation above remains $O(\delta_n^2)$ even after the multiplication with M_n. We analyse the diagonal and off-diagonal terms in the multiplication of the matrix M_n with the scalar above, ignoring the $O(\delta_n^2)$ term.

Diagonal terms in (3.8):

Consider the lth diagonal term inside the conditional expectation in (3.8):

$$\frac{1}{\kappa}\left((d_n^l)^2 - \lambda\right) \left(\sum_{i=1}^{N} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) \right) = \frac{1}{\kappa} (d_n^l)^2 \sum_{i=1}^{N} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n)$$
$$+ \frac{2}{\kappa} (d_n^l)^2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) - \frac{\lambda}{\kappa} \sum_{i=1}^{N} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) - \frac{2\lambda}{\kappa} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n). \tag{3.9}$$

From the distributions of d_n^i, d_n^j and the fact that d_n^i is independent of d_n^j for $i < j$, it is easy to see that $\mathbb{E}\left((d_n^l)^2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) \middle| \mathcal{F}_n \right) = 0$ and $\mathbb{E}\left(\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_n^i d_n^j \nabla_{ij}^2 f(\theta_n) \middle| \mathcal{F}_n \right) = 0$. Thus, the conditional expectations of the second and fourth terms on the RHS of (3.9) are both zero.

The first term on the RHS of (3.9) with the conditional expectation can be simplified as follows:

$$\frac{1}{\kappa}\mathbb{E}\left((d_n^l)^2 \sum_{i=1}^{N} (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) \middle| \mathcal{F}_n \right) = \frac{1}{\kappa}\mathbb{E}\left((d_n^l)^4 \nabla_{ll}^2 f(\theta_n) | \mathcal{F}_n \right) + \frac{1}{\kappa}\mathbb{E}\left(\sum_{i=1, i\neq l}^{N} (d_n^l)^2 (d_n^i)^2 \nabla_{ii}^2 f(\theta_n) \middle| \mathcal{F}_n \right)$$
$$= \frac{1}{\kappa}\mathbb{E}\left((d_n^l)^4 | \mathcal{F}_n \right) \nabla_{ll}^2 f(\theta_n) + \frac{1}{\kappa} \sum_{i=1, i\neq l}^{N} \mathbb{E}\left((d_n^l)^2 (d_n^i)^2 \middle| \mathcal{F}_n \right) \nabla_{ii}^2 f(\theta_n)$$

$$
\begin{aligned}
&= \frac{1}{\kappa}\mathbb{E}\left((d_n^l)^4\right)\nabla^2_{ll}f(\theta_n) + \frac{1}{\kappa}\sum_{i=1,i\neq l}^{N}\mathbb{E}\left((d_n^l)^2(d_n^i)^2\right)\nabla^2_{ii}f(\theta_n) \\
&= \frac{1}{\kappa}\left(\tau\nabla^2_{ll}f(\theta_n) + \lambda^2\sum_{i=1,i\neq l}^{N}\nabla^2_{ii}f(\theta_n)\right), \text{ a.s.} \qquad (3.10)
\end{aligned}
$$

For the second equality above, we have used the fact that $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$. For the third equality above, we have used the fact that d_n is independent of $\mathcal{F}_n$, for all n. And for the last equality above, we have used the fact that $\mathbb{E}[(d_n^l)^4] = \tau$ and $\mathbb{E}[(d_n^l)^2(d_n^i)^2] = \mathbb{E}[(d_n^l)^2]\mathbb{E}[(d_n^i)^2] = \lambda^2$, $\forall l \neq i$.

The third term in (3.9) with the conditional expectation and without the negative sign can be simplified as follows:

$$
\begin{aligned}
\frac{\lambda}{\kappa}\mathbb{E}\left(\sum_{i=1}^{N}(d_n^i)^2\nabla^2_{ii}f(\theta_n)\middle|\mathcal{F}_n\right) &= \frac{\lambda}{\kappa}\sum_{i=1}^{N}\mathbb{E}\left[(d_n^i)^2\right]\nabla^2_{ii}f(\theta_n) \\
&= \frac{\lambda^2}{\kappa}\sum_{i=1}^{N}\nabla^2_{ii}f(\theta_n), \text{ a.s.} \qquad (3.11)
\end{aligned}
$$

Combining the above followed by some algebra, i.e., (3.10) − (3.11), we obtain

$$
\frac{1}{\kappa}(\tau - \lambda^2)\nabla^2_{ll}f(\theta_n).
$$

By using the fact that $\kappa = \tau - \lambda^2$, we obtain

$$
\frac{1}{\kappa}\mathbb{E}\left[\left((d_n^l)^2 - \lambda\right)\left(\sum_{i=1}^{N}(d_n^i)^2\nabla^2_{ii}f(\theta_n) + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}d_n^i d_n^j\nabla^2_{ij}f(\theta_n)\right)\middle|\mathcal{F}_n\right] = \nabla^2_{ll}f(\theta_n), \text{ a.s.}
$$

Off-diagonal terms in (3.8):

We now consider the (k,l)th term in (3.8): Assume w.l.o.g that $k < l$. Then,

$$
\begin{aligned}
&\frac{1}{2\lambda^2}\mathbb{E}\left[d_n^k d_n^l\left(\sum_{i=1}^{N}(d_n^i)^2\nabla^2_{ii}f(\theta_n) + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}d_n^i d_n^j\nabla^2_{ij}f(\theta_n)\right)\middle|\mathcal{F}_n\right] \\
&= \frac{1}{2\lambda^2}\sum_{i=1}^{N}\mathbb{E}\left(d_n^k d_n^l (d_n^i)^2\right)\nabla^2_{ii}f(\theta_n) + \frac{1}{\lambda^2}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\mathbb{E}\left(d_n^k d_n^l d_n^i d_n^j\right)\nabla^2_{ij}f(\theta_n) \qquad (3.12)
\end{aligned}
$$

$$= \nabla^2_{kl} f(\theta_n).$$

The last equality follows from the fact that the first term in (3.12) is 0 since $k \neq l$, while the second term in (3.12) can be seen to be equal to $\frac{1}{\lambda^2}\mathbb{E}\left((d_n^k)^2(d_n^l)^2\right)\nabla^2_{kl} f(\theta_n) = \nabla^2_{kl} f(\theta_n)$.

□

Chapter 4

Second-order SPSA-3 with improved hessian estimation (2SPSA-3-IH)

The second-order SPSA-3 with improved Hessian estimate performs an update iteration similar to that of 2RDSA-IH proposed in chapter 2 (See (2.1) and (2.2)). The main difference in the iterates is that 2SPSA-3-IH uses gradient and Hessian estimates according to (4.3) and (4.4). In this chapter we present the improvements to 2SPSA-3 Hessian recursion (4.2) by incorporating a zero-mean feedback term and a general step-size. Note that though these ideas of improving the Hessian recursion are similar to the ideas presented in chapter 2 for 2RDSA-IH, the derivation of the feedback term does not directly follow. We show that the proposed improvements to Hessian estimation in 2SPSA-3 are such that the resulting 2SPSA-3-IH algorithm is provably convergent, in particular, the Hessian estimate $\overline{H}_n$ of 2SPSA-3-IH converges almost surely to the true Hessian. Moreover, we show, for the special case of a quadratic objective in noise-free setting, that the 2SPSA-3 scheme along with proposed improvement to Hessian estimation (henceforth referred to as 2SPSA-3-IH) results in a convergence rate that is on par with the corresponding rate for 2SPSA with Hessian estimation improvements (2SPSA-IH) [29]. The advantage with 2SPSA-3-IH is that it requires only 75% of the simulation cost per-iteration for 2SPSA-IH.

Algorithm 2 presents the pseudocode and we describe the individual components of 2SPSA-3-IH below.

Algorithm 2: Structure of 2SPSA-3-IH algorithm.

Input: initial parameter $\theta_0 \in \mathbb{R}^N$, perturbation constants $\delta_n > 0$, step-sizes $\{a_n, b_n\}$, operator Υ.

1. **Execution**:
   for $n \leftarrow 0, 1, 2, \ldots,$ **do**

   - Generate $\{\Delta_n^i, i = 1, \ldots, N\}$,$\{\widehat{\Delta}_n^i, i = 1, \ldots, N\}$ independent of $\{\Delta_m, \widehat{\Delta}_m, m = 0, 1, \ldots, n-1\}$.
   - For any $i = 1, \ldots, N$, Δ_n^i,$\widehat{\Delta}_n^i$ satisfy condition (A23) of chapter 4.5,(most popular distribution being used is symmetric Bernoulli distribution, ± 1 w.p. 1/2).
     - ***Function evaluation 1***
       Obtain $y_n^+ = f(\theta_n + \delta_n\Delta_n + \delta_n\widehat{\Delta}_n) + \xi_n^+$.
     - ***Function evaluation 2***
       Obtain $y_n^- = f(\theta_n - \delta_n\Delta_n - \delta_n\widehat{\Delta}_n) + \xi_n^-$.
     - ***Function evaluation 3***
       Obtain $y_n = f(\theta_n) + \xi_n$.
     - ***Newton step***
       Update the parameter and Hessian as follows:

$$\theta_{n+1} = \theta_n - a_n\Upsilon(\overline{H}_n)^{-1}\widehat{\nabla}f(\theta_n), \tag{4.1}$$

$$\overline{H}_n = (1 - b_n)\overline{H}_{n-1} + b_n(\widehat{H}_n - \widehat{\Psi}_n), \tag{4.2}$$

       where $\widehat{H}_n$ and $\widehat{\Psi}_n$ are chosen according to (4.4) and (4.14), respectively.

   end

   return θ_n.

4.1 Function evaluations

Let y_n, y_n^+ and y_n^- denote the function evaluations at θ_n, $\theta_n+\delta_n\Delta_n+\delta_n\widehat{\Delta}_n$ and $\theta_n-\delta_n\Delta_n-\delta_n\widehat{\Delta}_n$ respectively, i.e., $y_n = f(\theta_n)+\xi_n$, $y_n^+ = f(\theta_n+\delta_n\Delta_n+\delta_n\widehat{\Delta}_n)+\xi_n^+$ and $y_n^- = f(\theta_n-\delta_n\Delta_n-\delta_n\widehat{\Delta}_n)+\xi_n^-$, where the noise terms ξ_n, ξ_n^+, ξ_n^- satisfy $\mathbb{E}\left[\xi_n^+ + \xi_n^- - 2\xi_n \middle| \mathcal{F}_n, \Delta_n, \widehat{\Delta}_n\right] = 0$ with $\mathcal{F}_n = \sigma(\theta_m, m < n)$ denoting the underlying sigma-field.

Further, $\delta_n, n \geq 0$ is a sequence of diminishing positive real numbers and $\Delta_n = (\Delta_n^1, \ldots, \Delta_n^N)^\intercal$, $\widehat{\Delta}_n = (\widehat{\Delta}_n^1, \ldots, \widehat{\Delta}_n^N)^\intercal$ are the perturbation random vectors at instant n with $\Delta_n^i, i = 1, \ldots, N$, and $\widehat{\Delta}_n^i, i = 1, \ldots, N$ being independent identically distributed (i.i.d), mean-zero random variables having finite inverse moments of order greater than 2 and satisfying the condition (A23) of section 4.5.

4.2 Gradient estimate

The SPSA estimate of the gradient $\nabla f(\theta_n)$ is given by

$$\widehat{\nabla}_{(i)} f(\theta_n) = \left[\frac{y_n^+ - y_n^-}{2\delta_n \Delta_n^{(i)}}\right], \tag{4.3}$$

where the perturbations Δ_n^i, $i = 1, \ldots, N$ are i.i.d. and as mentioned above.

4.3 Hessian estimate

The (i, j)th entry of the Hessian estimate in this case is given by

$$\left(\widehat{H}_n\right)_{ij} = \left(\frac{y_n^+ + y_n^- - 2y_n}{2\delta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}}\right). \tag{4.4}$$

4.4 Feedback term $\widehat{\Psi}_n$

The (i, j)th term of the Hessian estimate $\widehat{H}_n$ can be simplified as follows:

$$\begin{aligned}\left(\widehat{H}_n\right)_{ij} &= \left(\frac{y_n^+ + y_n^- - 2y_n}{2\delta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}}\right) \\ &= \left(\frac{(\Delta_n + \widehat{\Delta}_n)^\intercal \nabla^2 f(\theta_n)(\Delta_n + \widehat{\Delta}_n)}{2\Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} + O(\delta_n^2) + \left(\frac{\xi_n^+ + \xi_n^- - 2\xi_n}{2\delta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}}\right)\right).\end{aligned} \tag{4.5}$$

For the first term on the RHS above, note that

$$\mathbb{E}\left[\frac{(\Delta_n+\widehat{\Delta}_n)^\intercal\nabla^2 f(\theta_n)(\Delta_n+\widehat{\Delta}_n)}{2\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\middle|\mathcal{F}_n\right]=\mathbb{E}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\Delta_n^{(m)}}{2\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\right.$$
$$\left.+\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}+\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\widehat{\Delta}_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}}{2\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\middle|\mathcal{F}_n\right]. \tag{4.6}$$

In analysing the RHS of the above expression, the first and the third terms are mean-zero (See Proposition 4.2 in [6]):

$$\mathbb{E}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\Delta_n^{(m)}}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\middle|\mathcal{F}_n\right]=0\ \textit{and}\ \mathbb{E}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\widehat{\Delta}_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\middle|\mathcal{F}_n\right]=0. \tag{4.7}$$

From the above equations the terms that are in expectation, denoted by $\Psi_n^1(\nabla^2 f(\theta_n))$, can be written in matrix form as follows:

$$\Psi_n^1(\nabla^2 f(\theta_n))=\frac{1}{2}M_n\left[\Delta_n^\intercal\nabla^2 f(\theta_n)\Delta_n+\widehat{\Delta}_n^\intercal\nabla^2 f(\theta_n)\widehat{\Delta}_n\right], \tag{4.8}$$

where $M_n=[1/\Delta_n^{(1)},\ldots,1/\Delta_n^{(N)}]^\intercal[1/\Delta_n^{(1)},\ldots,1/\Delta_n^{(N)}]$. Now consider the second term in the RHS of (4.6). It can be re-written as follows:

$$\mathbb{E}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\frac{\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\middle|\mathcal{F}_n\right]=\mathbb{E}\left[\nabla^2_{ij}f(\theta_n)+\frac{1}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\sum_{\substack{l=1\\lm\neq ij}}^{N}\sum_{m=1}^{N}\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}\middle|\mathcal{F}_n\right] \tag{4.9}$$

In analysing the RHS of the above expression, the second term in the expectation is mean-zero term:

$$\mathbb{E}\left[\frac{1}{\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}}\sum_{\substack{l=1\\lm\neq ij}}^{N}\sum_{m=1}^{N}\Delta_n^{(l)}\nabla^2_{lm}f(\theta_n)\widehat{\Delta}_n^{(m)}\middle|\mathcal{F}_n\right]=0. \tag{4.10}$$

The term on the LHS above, can be denoted by $\Psi_n^2(\nabla^2 f(\theta_n))$, and can be written in matrix form as follows:

$$\Psi_n^2(\nabla^2 f(\theta_n)) = \widehat{N}_n^{\intercal}\nabla^2 f(\theta_n)N_n + \widehat{N}_n^{\intercal}\nabla^2 f(\theta_n) + \nabla^2 f(\theta_n)N_n. \tag{4.11}$$

where $N_n, \widehat{N}_n$ are defined as follows: $N_n = \Delta_n \left[\frac{1}{\Delta_n^{(1)}}, \ldots, \frac{1}{\Delta_n^{(N)}}\right] - I_N$ and $\widehat{N}_n = \widehat{\Delta}_n \left[\frac{1}{\widehat{\Delta}_n^{(1)}}, \ldots, \frac{1}{\widehat{\Delta}_n^{(N)}}\right] - I_N$ and I_N is identity matrix of size $N \times N$. From the foregoing, the per-iteration Hessian estimate $\widehat{H}_n$ can be re-written as follows:

$$\widehat{H}_n = \nabla^2 f(\theta_n) + \Psi_n(\nabla^2 f(\theta_n)) + O(\delta_n^2) + O(\delta_n^{-2}) \tag{4.12}$$

where, for any matrix H,

$$\begin{aligned}\Psi_n(H) &= \Psi_n^1(H) + \Psi_n^2(H) \\ &= \frac{1}{2}M_n\left[\Delta_n^{\intercal}H\Delta_n + \widehat{\Delta}_n^{\intercal}H\widehat{\Delta}_n\right] + \widehat{N}_n^{\intercal}HN_n + \widehat{N}_n^{\intercal}H + HN_n.\end{aligned} \tag{4.13}$$

Given that we operate in a simulation optimization setting, which implies $\nabla^2 f$ is not known, we construct the feedback term $\widehat{\Psi}_n$ in (4.2) by using $\overline{H}_{n-1}$ as a proxy for $\nabla^2 f$, i.e.,

$$\widehat{\Psi}_n = \Psi_n(\overline{H}_{n-1}). \tag{4.14}$$

Optimizing the step-sizes b_n The optimal choice for b_n in (4.2) is the following:

$$b_i = \delta_i^4 / \sum_{j=0}^{i} \delta_j^4. \tag{4.15}$$

The main idea behind the above choice is provided below. From (4.12), we can infer that

$$\mathbb{E} \parallel \widehat{H}_n \parallel^2 \leq \frac{C}{\delta_n^4} \text{ for some } C < \infty.$$

This is because the third term in (4.12) vanishes asymptotically, while the fourth term there dominates asymptotically. Moreover, the noise factors in the fourth term in (4.12) are bounded above due to (A28) and independent of n, leaving the δ_n^2 term in the denominator there.

So, the optimization problem to be solved at instant n is as follows:

$$\sum_{i=0}^{n}(\tilde{b}_i)^2\delta_i^{-4}, \text{ subject to} \tag{4.16}$$

$$\tilde{b}_i \geq 0 \ \forall i \text{ and } \sum_{i=0}^{n}\tilde{b}_i = 1. \tag{4.17}$$

The optimization variable $\tilde{b}_i$ from the above is related to the Hessian recursion (4.2) as follows:

$$\overline{H}_n = \sum_{i=0}^{n}\tilde{b}_i(\widehat{H}_i - \widehat{\Psi}_i). \tag{4.18}$$

The solution to (4.16) is achieved for $\tilde{b}_i^* = \delta_i^4 / \sum_{j=0}^{n}\delta_j^4, i = 1, \ldots, n$. The optimal choice $\tilde{b}_i^*$ can be translated to the step-sizes b_i, leading to (4.15).

4.5 Convergence analysis for 2SPSA-3-IH

We make the same assumptions as those used in the analysis of [21], with a few minor alterations. The assumptions are listed below:

(**A20**) The function f is four-times differentiable[1] with $\left|\nabla^4_{i_1 i_2 i_3 i_4} f(\theta)\right| < \infty$, for $i_1, i_2, i_3, i_4 = 1, \ldots, N$ and for all $\theta \in \mathbb{R}^N$.

(**A21**) For each n and all θ, there exists a $\rho > 0$ not dependent on n and θ, such that $(\theta - \theta^*)^\intercal \bar{f}_n(\theta) \geq \rho \left\|\theta_n - \theta\right\|$, where $\bar{f}_n(\theta) = \Upsilon(\overline{H}_n)^{-1}\nabla f(\theta)$.

(**A22**) $\{\xi_n, \xi_n^+, \xi_n^-, n = 1, 2, \ldots\}$ are such that, for all n, $\mathbb{E}\left[\xi_n^+ + \xi_n^- - 2\xi_n \middle| \mathcal{F}_n, \Delta_n, \widehat{\Delta}_n\right] = 0$, where $\mathcal{F}_n = \sigma(\theta_m, m < n)$ denotes the underlying sigma-field.

(**A23**) $\{\Delta_n^i, \widehat{\Delta}_n^i, i = 1, \ldots, N, n = 1, 2, \ldots\}$ are i.i.d independent of $\mathcal{F}_n$ and for some $\alpha_3 > 0$ and for all n, l $|\Delta_{nl}| \leq \alpha_3$, Δ_{nl} is symmetrically distributed about 0, Δ_{nl} are mutually independent across n and l and they satisfy $\mathbb{E}(\Delta_{nl}^{-2})$, $\mathbb{E}(\widehat{\Delta}_{nl}^{-2}) \leq \alpha_3$.

[1]Here $\nabla^4 f(\theta) = \dfrac{\partial^4 f(\theta)}{\partial\theta^\intercal\partial\theta^\intercal\partial\theta^\intercal\partial\theta^\intercal}$ denotes the fourth derivate of f at θ and $\nabla^4_{i_1,i_2,i_3,i_4} f(\theta)$ denotes the (i_1, i_2, i_3, i_4)th entry of $\nabla^4 f(\theta)$, for $i_1, i_2, i_3, i_4 = 1, \ldots, N$.

(**A24**) The step-sizes a_n and perturbation constants δ_n are positive, for all n and satisfy

$$a_n, \delta_n \to 0 \text{ as } n \to \infty, \sum_n a_n = \infty \text{ and } \sum_n \left(\frac{a_n}{\delta_n}\right)^2 < \infty.$$

(**A25**) For each $i = 1, \ldots, N$ and any $\rho > 0$, $P(\{\bar{f}_{ni}(\theta_n) \geq 0 \text{ i.o}\} \cap \{\bar{f}_{ni}(\theta_n) < 0 \text{ i.o}\} \mid \{|\theta_{ni} - \theta_i^*| \geq \rho \quad \forall n\}) = 0$.

(**A26**) The operator Υ satisfies $\delta_n^2 \Upsilon(H_n)^{-1} \to 0$ a.s. and $E(\|\Upsilon(H_n)^{-1}\|^{2+\zeta}) \leq \rho$ for some $\zeta, \rho > 0$.

(**A27**) For any $\tau > 0$ and nonempty $S \subseteq \{1, \ldots, N\}$, there exists a $\rho'(\tau, S) > \tau$ such that

$$\limsup_{n \to \infty} \left| \frac{\sum_{i \notin S} (\theta - \theta^*)_i \bar{f}_{ni}(\theta)}{\sum_{i \in S} (\theta - \theta^*)_i \bar{f}_{ni}(\theta)} \right| < 1 \text{ a.s.}$$

for all $|(\theta - \theta^*)_i| < \tau$ when $i \notin S$ and $|(\theta - \theta^*)_i| \geq \rho'(\tau, S)$ when $i \in S$.

(**A28**) For some $\alpha_7, \alpha_8, \alpha_9 > 0$ and for all n, l, m, $\mathbb{E}{\xi_n}^2 \leq \alpha_7$, $\mathbb{E}\xi_n^{\pm 2} \leq \alpha_7$, $\mathbb{E}f(\theta_n)^2 \leq \alpha_8$, $\mathbb{E}f(\theta_n + \delta_n\Delta_n + \delta_n\widehat{\Delta}_n)^2, \mathbb{E}f(\theta_n - \delta_n\Delta_n - \delta_n\widehat{\Delta}_n)^2 \leq \alpha_8$, $\mathbb{E}\left[|f(\theta_n + \delta_n\Delta_n + \delta_n\widehat{\Delta}_n)/(\Delta_{nl}\widehat{\Delta}_{nm})|^{2+\alpha_9} \mid \mathcal{F}_n\right]$, $\mathbb{E}\left[|f(\theta_n - \delta_n\Delta_n - \delta_n\widehat{\Delta}_n)/(\Delta_{nl}\widehat{\Delta}_{nm})|^{2+\alpha_9} \mid \mathcal{F}_n\right]$, $\mathbb{E}\left[(\xi_n^+ + \xi_n^- - 2\xi_n)^2/(\Delta_{nl}\widehat{\Delta}_{nm})^2 \middle| \mathcal{F}_n\right] \leq \alpha_8$ and $\mathbb{E}\left(\left\|\Upsilon(\overline{H}_n)\right\|^2 \mid \mathcal{F}_n\right) \leq \alpha_8$.

(**A29**) $\delta_n = \frac{\delta_0}{(n+1)^\varsigma}$, where $\delta_0 > 0$ and $0 < \varsigma \leq 1/8$.

The reader is referred to Section II-B of [21] for a detailed discussion of the above assumptions. We remark here that (A20)-(A27) are identical to those in [21], while (A28) and (A29) introduce minor additional requirements on $\left\|\Upsilon(\overline{H}_n)\right\|^2$ and δ_n, respectively and these are inspired from [29].

Lemma 4.1. *(**2SPSA-3-IH Bias in Hessian estimate**) Under (A20)-(A29), with $\widehat{H}_n$ defined according to (4.4) , we have a.s. that*[1]*, for $i, j = 1, \ldots, N$,*

$$\left| \mathbb{E}\left[\widehat{H}_n(i,j) \middle| \mathcal{F}_n\right] - \nabla_{ij}^2 f(\theta_n) \right| = O(\delta_n^2). \tag{4.19}$$

Proof. See Proposition 4.2 in [6]. □

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

Theorem 4.1. *(**2SPSA-3-IH Strong Convergence of Hessian**) Under (A20)-(A29), we have that*

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

In the above, θ_n and $\overline{H}_n$ are updated according to (4.1) and (4.2), respectively, $\widehat{H}_n$ defined according to (4.4) and the step-sizes b_n are chosen as suggested in (4.15).

Proof. The proof is similar to the proof of the Theorem 2.1 and is therefore omitted. □

We next present a convergence rate result for the special case of a quadratic objective function under the following additional assumptions:

(**A30**) f is quadratic and $\nabla^2 f(\theta^*) > 0$.

(**A31**) The operator Υ is chosen such that $\mathbb{E}(\| \Upsilon(\overline{H}_n) - \overline{H}_n \|^2) = o(e^{-2wn^{1-r}/(1-r)})$ and $\| \Upsilon(H) - H \|^2 / (1 + \| H \|^2)$ is uniformly bounded.

Theorem 4.2. *(**2SPSA-3-IH Quadratic case - Convergence rate**) Assume (A23), (A29), (A30) and (A31) and also that the setting is noise-free. Let $b_n = b_0/n^r$, $n = 1, 2, \ldots, k$, where $1/2 < r < 1$ and $0 < b_0 \leq 1$. For notational simplicity, let $H^* = \nabla^2 f(\theta^*)$. Letting $\Lambda_k = \overline{H}_k - H^*$, we have*

$$trace[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)] = O(e^{-2b_0 n^{1-r}/1-r}). \tag{4.20}$$

Proof. Since the setting is noise-free with a quadratic objective, we can rewrite (4.12) as follows:

$$\begin{aligned} \widehat{H}_n &= \nabla^2 f(\theta_n) + \Psi_n(\nabla^2 f(\theta_n)) \\ &= \nabla^2 f(\theta^*) + \Psi_n(\nabla^2 f(\theta^*)) \\ &= H^* + \Psi_n(H^*), \end{aligned} \tag{4.21}$$

where $\Psi_n(H)$ is defined in (4.14). The proof involves the following steps:

Step 1: Here we prove the MSE convergence of $\overline{H}_k$, i.e., $\mathbb{E}[\Lambda_n^\intercal \Lambda_n] \to 0$ a.s. as $n \to \infty$.

Step 2: We unroll the recursion (4.2) and then derive a convenient representation for the trace$[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)]$.

Step 3: We derive the main result in (4.20) using a proof by contradiction.

Step 1: MSE convergence of $\overline{H}_n$

This part of the proof follows along similar lines as part (i) in Theorem 3 of [29].

By rewriting the (4.2) in the following form

$$\begin{aligned}\overline{H}_n &= \overline{H}_{n-1} - b_n(\overline{H}_{n-1} - \widehat{H}_n + \widehat{\Psi}_n) \\ &= \overline{H}_{n-1} - b_n(\overline{H}_{n-1} - H^* + \widehat{\Psi}_n - \Psi_n(H^*)).\end{aligned} \tag{4.22}$$

The above equation can be thought as a stochastic approximation analogue of (4.2). The above stochastic approximation algorithm is aimed at finding the roots of the equation $H - H^* = 0$. In (4.22), $\mathbb{E}(\widehat{\Psi}_n - \Psi_n(H^*)|\mathcal{F}_n) = 0 \;\; a.s.$ Given a Lyapunov function, one can get the conditions required to show the mean square convergence of the (4.22) from [19], pp.92-94. It is easy to see that $V(H) = \frac{1}{2}(H - H^*)^\intercal(H - H^*)$ indeed serves as Lyapunov function. From part (i) in Theorem 3 of [29], one can check that our algorithm indeed satisfies those conditions, which in turn implies that $\mathbb{E}[\Lambda_n^\intercal \Lambda_n] \to 0 \;\; a.s.$ as $n \to \infty$.

Step 2: Representation of trace$[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)]$

From (4.2) and (4.21), we have

$$\begin{aligned}\Lambda_n &= \Lambda_{n-1} - b_n(\overline{H}_{n-1} - \hat{H}_n + \hat{\Psi}_n) \\ &= (1 - b_n)\Lambda_{n-1} - b_n(H^* + \hat{\Psi}_n - \hat{H}_n) \\ &= (1 - b_n)\Lambda_{n-1} - b_n(\hat{\Psi}_n - \Psi_n(H^*)) \\ &= (1 - b_n)\Lambda_{n-1} - b_n\Psi_n(\Upsilon(\overline{H}_{n-1})) + b_n\Psi_n(H^*) \\ &= (1 - b_n)\Lambda_{n-1} - b_n\Psi_n(\Lambda'_{n-1}),\end{aligned} \tag{4.23}$$

where $\Lambda'_n = \Upsilon(\overline{H}_n) - H^*$. The equality in (4.23) follows from (2.13). Unrolling the recursion (4.23), we obtain

$$\Lambda_n = \left[\prod_{k=1}^{n}(1 - b_k)\right]\Lambda_0 - \sum_{k=1}^{n}\left[\prod_{j=k+1}^{n}(1 - b_j)\right] b_k\Psi_k(\Lambda'_{k-1}) \;\; a.s. \tag{4.24}$$

It follows from (4.24) that

$$\mathbb{E}(\Lambda_n^\intercal \Lambda_n) = \left[\prod_{k=1}^{n}(1-b_k)\right]^2 \mathbb{E}(\Lambda_0^\intercal \Lambda_0) + \sum_{k=1}^{n}\left[\prod_{j=k+1}^{n}(1-b_j)\right]^2 b_k^2 \mathbb{E}(\Psi_k(\Lambda'_{k-1})^\intercal \Psi_k(\Lambda'_{k-1})). \quad (4.25)$$

The equality above uses the fact that $\mathbb{E}(\Psi_k(\Lambda'_{k-1})) = 0$, which gets rid of the corresponding cross terms with the first term on RHS of (4.24). We now characterize trace$[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)]$ using (4.25) as follows:

$$\text{trace}\left[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)\right] = \left[\prod_{k=1}^{n}(1-b_k)\right]^2 \text{trace}\left[\mathbb{E}(\Lambda_0^\intercal \Lambda_0)\right] + \sum_{k=1}^{n}\left[\prod_{j=k+1}^{n}(1-b_j)\right]^2 b_k^2\, \tau\left(\mathbb{E}(\Lambda'_{k-1}\otimes\Lambda'_{k-1})\right). \quad (4.26)$$

whereas in the proof of Theorem 3 of [29], $\tau(\cdot)$ transforms the $\mathbb{E}(\Lambda'_{k-1}\otimes\Lambda'_{k-1})$ in a fashion and then returns the trace of the resulting $N\times N$ matrix. As in the proof of Theorem 3 of [29], observe that $1-b_k = e^{-b_k}(1-O(b_k^2))$ and since $0<b_k<1$, we have that the $O(b_k^2)$ term is strictly positive. Letting $\Gamma_{ij} = \sum_{k=i}^{j} b_k$ with $\Gamma_{nn}=1$ and $\beta_{kn} = \left[\prod_{i=k+1}^{n}(1-O(b_i^2))\right]^2$, we can simplify (4.26) as follows:

$$\text{trace}\left[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)\right] = e^{-2\Gamma_{1n}}\beta_{0n}\text{trace}\left[\mathbb{E}(\Lambda_0^\intercal \Lambda_0)\right] + e^{-2\Gamma_{1n}}\sum_{k=1}^{n} e^{2\Gamma_{1k}}\beta_{kn} b_k^2\, \tau\left(\mathbb{E}(\Lambda'_{k-1}\otimes\Lambda'_{k-1})\right). \quad (4.27)$$

Comparing the sum with integrals, we obtain

$$\Gamma_{ij} = \int_i^j \frac{b_0}{x^r}dx + O(1) = \left(\frac{b_0}{1-r}\right)(j^{1-r} - i^{1-r}) + O(1),$$

where we have used the facts that $0<b_k<1, \forall k\geq 2$ and $\sum_{k=i}^{j} b_k \to \infty$ as $j-i\to\infty$ since $b_k = b_0/k^r$ with $r>0.5$. Observing that β_{kn} are uniformly upper-bounded, say by $\bar{\beta}_n$, we have

$$\begin{aligned}\text{trace}\left[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)\right] &= e^{-2\Gamma_{1n}}\beta_{0n}\text{trace}\left[\mathbb{E}(\Lambda_0^\intercal \Lambda_0)\right] \\ &\quad + \bar{\beta}_n e^{-2b_0 n^{1-r}/(1-r)}\sum_{k=1}^{n} e^{2b_0 k^{1-r}/(1-r)} \times \frac{b_0^2}{k^{2r}} \times \tau\left(\mathbb{E}(\Lambda'_{k-1}\otimes\Lambda'_{k-1})\right). \quad (4.28)\end{aligned}$$

Step 3: The big-O result on trace$[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)]$ convergence

We will use similar argument in part (iii) in Theorem 3 of [29] to show the claim (4.20). The idea here is to show that the contradiction exists if RHS of (4.20) exhibits any slower rate of decay. Let us assume the following slower rate of decay for the LHS of (4.20).

$$\text{trace}[\mathbb{E}(\Lambda_n^{\intercal}\Lambda_n)] = d(n)e^{-2b_0 n^{1-r}/1-r}, \tag{4.29}$$

where $d(n) = o(e^{2b_0 n^{1-r}/1-r})$. Now we will to show that there is a contradiction. By using the series of arguments used in part (iii) in Theorem 3 of [29] and applying the condition (A31) we get the following:

$$\mathbb{E}(\Lambda_n' \otimes \Lambda_n') = \mathbb{E}(\Lambda_n \otimes \Lambda_n) + o(d(n)e^{-2b_0 n^{1-r}/1-r}). \tag{4.30}$$

Now from the above equation we can write

$$\mathbb{E}(\Lambda_{n-1}' \otimes \Lambda_{n-1}') = O(\text{trace}[\mathbb{E}(\Lambda_{n-1}^{\intercal}\Lambda_{n-1})]). \tag{4.31}$$

Now we can use the result in the RHS of equation (4.28) to show the contradiction. We consider two cases to show the contradiction. The first is where $\frac{d(n)}{n^{2r}} = O(1)$ and the other is $limsup_{n\to\infty}\frac{d(n)}{n^{2r}} = \infty$. We show contradiction for the both cases separately. Let us consider the first case, i.e., $\frac{d(n)}{n^{2r}} = O(1)$, Then

$$\begin{aligned}\text{RHS of (4.28)} &= O(1)e^{-2\Gamma_{1n}} + O(1)e^{-2b_0 n^{1-r}/(1-r)}\sum_{i=1}^{n}\frac{d(i)}{i^p}\frac{b_0^2}{i^{2r-p}} \\ &= e^{-2b_0 n^{1-r}/1-r}\left[O(1) + O(1)\sum_{i=1}^{n}\frac{1}{i^{2r-p}}\right],\end{aligned} \tag{4.32}$$

where there exist $0 < p \leq 2r$, $0 < \epsilon < 2r-1$ such that $\frac{d(n)}{i^p} = O(1)$ and $limsup_{n\to\infty}\frac{d(n)}{n^{p-\epsilon}} = \infty$. In (4.32), the first equality uses (4.29) and (4.31).

Now there exists a subsequence $\{i_1, \ldots, i_n\}$ such that $\frac{d(i_n)}{i_n^{p-\epsilon}} = \infty$ as $n \to \infty$. Now LHS of (4.28) satisfies $LHS(i_n) \geq C i_n^{p-\epsilon} e^{-2b_0 i_n^{1-r}/1-r}$ for some $C > 0$. Now we have $\frac{LHS(i_n)}{RHS(i_n)} \to 0$ because $0 < \epsilon < 2r-1$. This leads to a contradiction since $LHS(n) = RHS(n)$ for all n.

Now consider the second case where $limsup_{n\to\infty}\frac{d(n)}{n^{2r}} = \infty$. There exists a subsequence $\{i_1, \ldots, i_n\}$ such that $\frac{d(i_n)}{i_n^{2r}} \geq \frac{d(i)}{i^{2r}}$ as $i \leq i_n$. Then for some $C > 0$ the RHS of (4.28) satisfies

the following:

$$\begin{aligned}\text{RHS}(i_n) \text{ of } (4.28) &= O(1)e^{-2\Gamma_{1i_n}} + O(1)e^{-2b_0 i_n^{1-r}/(1-r)}\left[\sum_{i=1}^{i_n} e^{2b_0 i^{1-r}/(1-r)}\frac{b_0^2}{i^{2r}}\mathbb{E}(\Lambda_n \otimes \Lambda_n)\right] \\ &\leq Ce^{-2b_0 n^{1-r}/1-r}\left[1 + O(1)\sum_{i=1}^{i_n}\frac{d(i)}{i^{2r}}\right] \\ &\leq Ce^{-2b_0 n^{1-r}/1-r}\left[1 + \frac{d(i_n)}{{i_n}^{2r}}\right], \end{aligned} \tag{4.33}$$

where $\text{RHS}(i_n)$ denotes, RHS after the algorithm is iterated for i_n times. From the equation (4.29), the LHS of (4.28) after i_n iterations is $d(i_n)e^{-2b_0 i_n^{1-r}/1-r}$. By substituting (4.29) and (4.33), in $\frac{RHS(i_n)}{LHS(i_n)}$, it turns out that $\frac{RHS(i_n)}{LHS(i_n)} \to 0$ as $n \to \infty$. This results in a contradiction since $LHS(n) = RHS(n)$ for all n. Hence the claim $\text{trace}[\mathbb{E}(\Lambda_n^\intercal \Lambda_n)] = O(e^{-2b_0 n^{1-r}/1-r})$ follows. □

Chapter 5

Simulation experiments

5.1 Implementation

We test the performance of 2RDSA-Unif, 2RDSA-AsymBer, 2SPSA-3 and 2SPSA, with/without improved Hessian estimation. 2SPSA and 2SPSA-3 algorithms use Bernoulli ± 1-valued perturbations, while 2RDSA/2RDSA-IH come in two variants - one that uses $U[-1, 1]$ distributed perturbations (referred to as 2RDSA-Unif/2RDSA-IH-Unif) and the other that uses asymmetric Bernoulli perturbations (referred to as 2RDSA-AsymBer/2RDSA-IH-AsymBer)[1].

For the empirical evaluations, we use the following two loss functions in $N = 10$ dimensions:

Quadratic loss

$$f(x) = \theta^\intercal A\theta + b^\intercal \theta. \tag{5.1}$$

The optimum θ^* for the above f is such that each coordinate of θ^* is -0.9091, with $f(\theta^*) = -4.55$.

Fourth-order loss

$$f(x) = \theta^\intercal A^\intercal A\theta + 0.1\sum_{j=1}^{N}(A\theta)_j^3 + 0.01\sum_{j=1}^{N}(A\theta)_j^4. \tag{5.2}$$

The optimum θ^* for the above f is $\theta^* = 0$, with $f(\theta^*) = 0$.

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

In both functions, A is such that NA is an upper triangular matrix with each entry one, b is the N-dimensional vector of ones and the noise structure is similar to that used in [27]. For any θ, the noise is $[\theta^{\intercal}, 1]z$, where $z \approx \mathcal{N}(0, \sigma^2 I_{11\times 11})$. We perform experiments for noisy as well as noise-less settings, with $\sigma = 0.1$ for the noisy case.

For all algorithms, we set $\delta_n = 3.8/n^{0.101}$ and $a_n = 1/n^{0.6}$, while b_n are set according to (2.15). These choices have been used for 2SPSA implementations before (see [27]) and have demonstrated good finite-sample performance empirically, while satisfying the theoretical requirements needed for asymptotic convergence. For all the algorithms, the initial point θ_0 is the N-dimensional vector of ones. For 2SPSA, 2SPSA-3 and 2RDSA/2RDSA-IH, an initial 20% of the simulation budget was used up by 1SPSA/1RDSA and the resulting iterate was used to initialize 2SPSA, 2SPSA-3/2RDSA. The distribution parameter ϵ is set to 0.0001 for 2RDSA and to 0.01 for 1RDSA.

5.2 Results

We use normalized loss and normalized MSE (NMSE) as performance metrics for evaluating the algorithms. NMSE is the ratio $\|\theta_{n_{\text{end}}} - \theta^*\|^2 / \|\theta_0 - \theta^*\|^2$, while normalized loss is the ratio $f(\theta_{n_{\text{end}}})/f(\theta_0)$. Here n_{end} denotes the iteration number when the algorithm stopped updating its parameter. Note that n_{end} is a function of the simulation budget. 2RDSA/2RDSA-IH, 2SPSA-3/2SPSA-3-IH use only three simulations per-iteration and hence, n_{end} is 1/3rd of the simulation budget, while it is 1/4th of the simulation budget for 2SPSA, since the latter algorithm uses four simulations per-iteration.

Tables 5.1–5.2 present the normalized loss values observed for the four algorithms - 2SPSA, 2SPSA-3, 2RDSA-Unif and 2RDSA-AsymBer - with/without improved Hessian estimation scheme and for the fourth-order and quadratic loss functions, respectively. Table 5.3 presents the NMSE values obtained for the aforementioned algorithms with the quadratic loss. The results in Tables 5.1–5.3 are obtained after running all the algorithms with a budget of $10,000$ function evaluations.

Figures 5.1, 5.4 plot the normalized loss as a function of the simulation budget with the fourth-order loss objective with $\sigma = 0.1$ for algorithms 2RDSA and 2SPSA-3 respectively when compared with 2SPSA. Figures 5.2, 5.5 plot the normalized loss as a function of the simulation budget with the quadratic loss objective with $\sigma = 0$ for algorithms 2RDSA and 2SPSA-3

Table 5.1: Normalized loss values for fourth-order objective (5.2) with and without noise: standard error from 500 is replications shown after ±.

Noise parameter $\sigma = 0.1$		
	Regular	**Improved Hessian estimation**
2SPSA	0.132 ± 0.0267	0.104 ± 0.0355
2SPSA-3	0.0893 ± 0.001	0.0457 ± 0.0004
2RDSA-Unif	0.115 ± 0.0214	0.0271 ± 0.0538
2RDSA-AsymBer	0.0471 ± 0.021	$\mathbf{0.0099 \pm 0.0014}$
Noise parameter $\sigma = 0$		
	Regular	**Improved Hessian estimation**
2SPSA	0.0795 ± 0.0234	0.0628 ± 0.0234
2SPSA-3	0.0338 ± 0.0001	0.0315 ± 0.0007
2RDSA-Unif	0.0813 ± 0.0275	0.0214 ± 0.00376
2RDSA-AsymBer	0.0199 ± 0.0114	$\mathbf{0.0098 \pm 0.00147}$

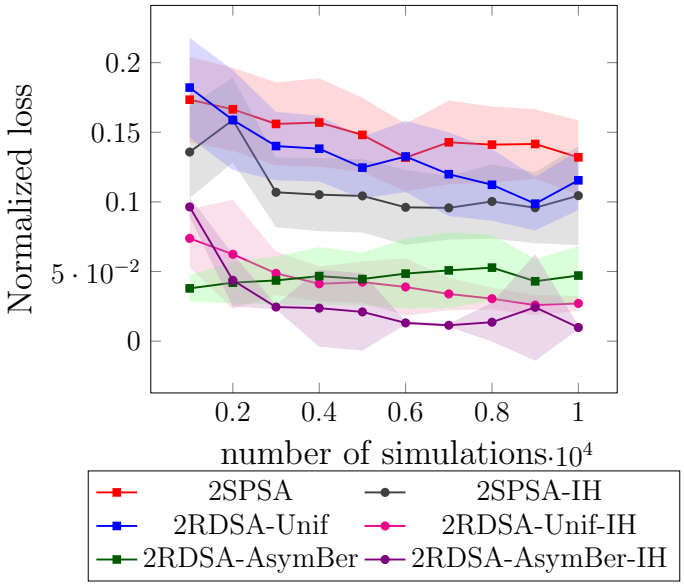


Figure 5.1: Normalized loss vs. number of simulations for fourth-order loss (5.2) with $\sigma = 0.1$ for 2SPSA, 2RDSA-Unif and 2RDSA-AsymBer algorithms with/without improved Hessian estimation: bands around the curves represent standard error from 500 replications.

Table 5.2: Normalized loss values for quadratic objective (5.1) with and without noise: standard error from 500 replications is shown after ±.

Noise parameter $\sigma = 0.1$		
	Regular	**Improved Hessian estimation**
2SPSA	-0.0062 ± 0.1164	-0.1229 ± 0.1374
2SPSA-3	-0.0161 ± 0.0955	-0.2748 ± 0.0629
2RDSA-Unif	0.0485 ± 0.1465	-0.259 ± 0.0398
2RDSA-AsymBer	-0.2564 ± 0.068	$\mathbf{-0.2877 \pm 0.0051}$
Noise parameter $\sigma = 0$		
	Regular	**Improved Hessian estimation**
2SPSA	-0.0785 ± 0.1178	-0.1716 ± 0.1339
2SPSA-3	-0.0667 ± 0.0602	-0.2793 ± 0.0625
2RDSA-Unif	0.0326 ± 0.1599	-0.2672 ± 0.0299
2RDSA-AsymBer	-0.2777 ± 0.0488	$\mathbf{-0.2881 \pm 0.0012}$

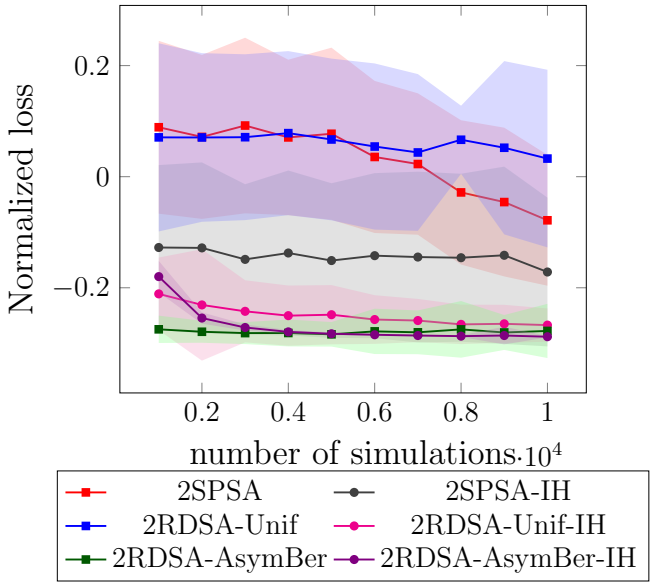


Figure 5.2: Normalized loss vs. number of simulations for quadratic loss (5.1) with $\sigma = 0$ for 2SPSA, 2RDSA-Unif and 2RDSA-AsymBer algorithms with/without improved Hessian estimation.

Table 5.3: NMSE values for quadratic objective (5.1) with and without noise: standard error from 500 replications is shown after ±.

Noise parameter $\sigma = 0.1$		
	Regular	**Improved Hessian estimation**
2SPSA	0.9491 ± 0.0131	0.5495 ± 0.0217
2SPSA-3	0.8378 ± 0.0179	0.1045 ± 0.0005
2RDSA-Unif	1.0073 ± 0.0140	0.1953 ± 0.0095
2RDSA-AsymBer	0.1667 ± 0.0095	$\mathbf{0.0324 \pm 0.0007}$
Noise parameter $\sigma = 0$		
	Regular	**Improved Hessian estimation**
2SPSA	0.7325 ± 0.0180	0.3939 ± 0.0230
2SPSA-3	0.6661 ± 0.01554	0.0684 ± 0.0006
2RDSA-Unif	0.9834 ± 0.0170	0.1623 ± 0.0086
2RDSA-AsymBer	0.0686 ± 0.0078	$\mathbf{0.0316 \pm 0.0006}$

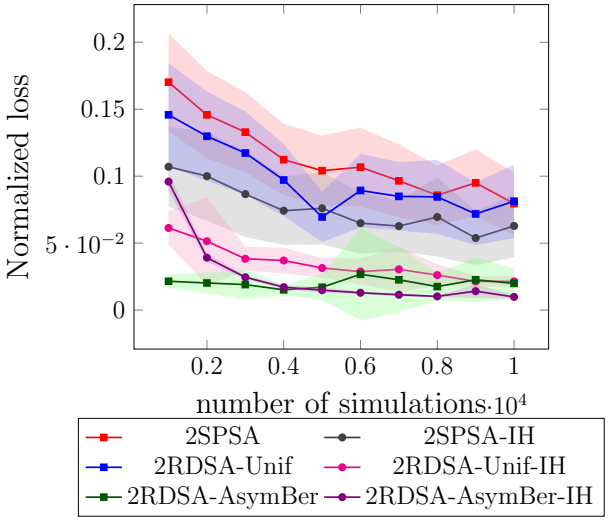
respectively when compared with 2SPSA. See Figures 5.3a– 5.3b and 5.6a– 5.6b for similar results with $\sigma = 0$ for fourth-order loss and $\sigma = 0.1$ for quadratic loss. From the results in Tables 5.1–5.3 and Figures, we make the following observations:

Observation 1: *Schemes with improved Hessian estimation perform better than their respective regular schemes.*

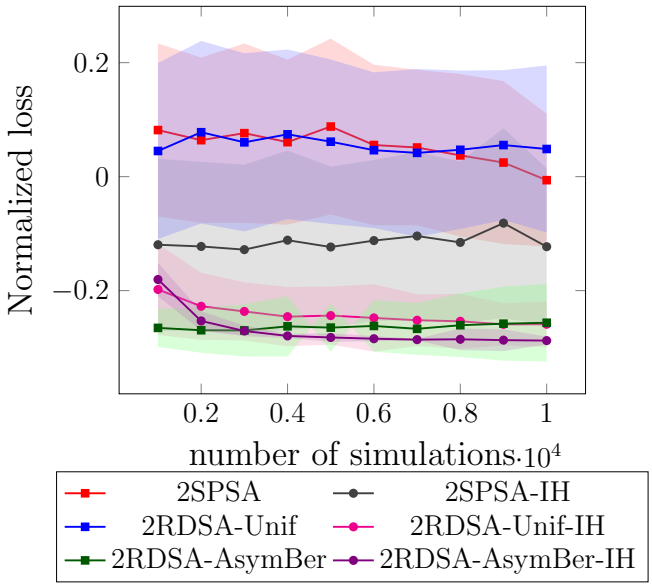
Observation 2: *2RDSA-IH variants outperform both 2SPSA and 2SPSA-IH.*

Observation 3: *2SPSA-3-IH variants outperform both 2SPSA and 2SPSA-IH.*

Observation 4: *2RDSA-IH-AsymBer perform the best overall.*



(a) Fourth-order loss (5.2) with $\sigma = 0$.



(b) Quadratic loss (5.1) with $\sigma = 0.1$.

Figure 5.3: Normalized loss vs. number of simulations in two different loss settings for all the algorithms.

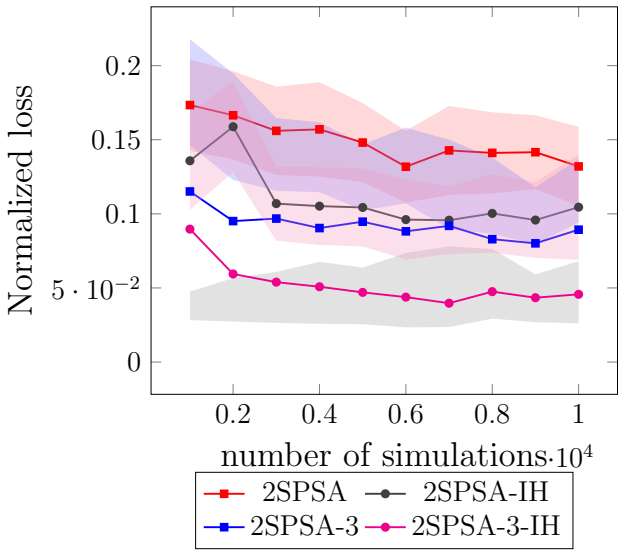


Figure 5.4: Normalized loss vs. number of simulations for fourth-order loss (5.2) with $\sigma = 0.1$ for 2SPSA, 2SPSA-3 algorithms with/without improved Hessian estimation: bands around the curves represent standard error from 500 replications.

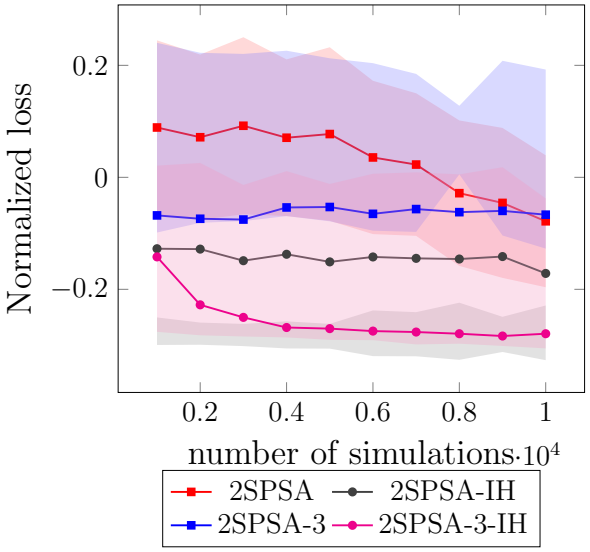
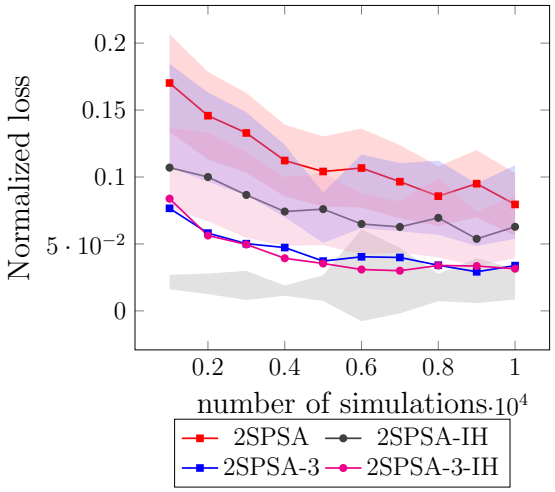
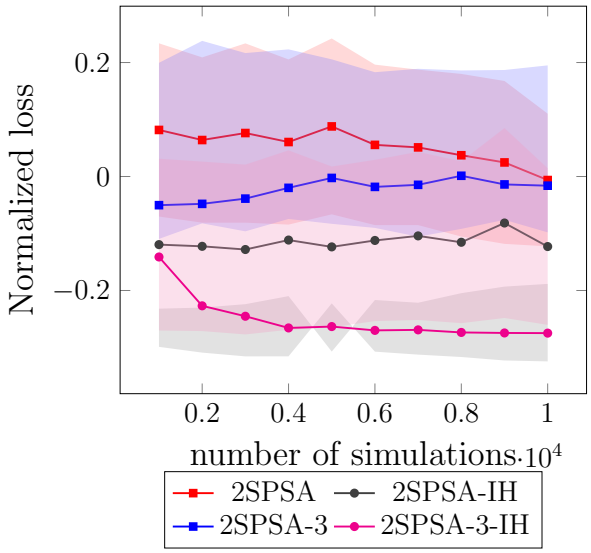


Figure 5.5: Normalized loss vs. number of simulations for quadratic loss (5.1) with $\sigma = 0$ for 2SPSA, 2SPSA-3 algorithms with/without improved Hessian estimation.



(a) Fourth-order loss (5.2) with $\sigma = 0$.



(b) Quadratic loss (5.1) with $\sigma = 0.1$.

Figure 5.6: Normalized loss vs. number of simulations in two different loss settings for 2SPSA, 2SPSA-3 algorithms.

Chapter 6

Conclusions and Future work

In this thesis, we presented an improved Hessian estimation scheme for the 2RDSA algorithm [21]. The proposed scheme was shown to be provably convergent to the true Hessian. We proposed generalisations of the RDSA algorithm proposed in [21]. As a result of the generalisation, we are able to use a much larger class of distributions for the perturbation random variables. The proposed generalisation inherits all the improved Hessian estimation schemes proposed earlier. We also presented an improved Hessian estimation scheme for the 2SPSA-3 algorithm [6] and for the special case of a quadratic objective, it resulted in a convergence rate that is on par with the corresponding rate for 2SPSA with Hessian estimation improvements (2SPSA-IH) [29]. The advantage with 2RDSA-IH, 2SPSA-3-IH is that these schemes it require only 75% of the simulation cost per-iteration for 2SPSA with Hessian estimation improvements (2SPSA-IH) [29]. Numerical experiments demonstrated that 2RDSA-IH, 2SPSA-3-IH outperform both 2SPSA-IH and 2SPSA without the improved Hessian estimation procesdure. They also indicate that schemes with improved Hessian estimation outperform the respective regular schemes without the improved Hessian estimation procedure. 2RDSA-IH with asymmetric Bernoulli distribution is seen to perform the best overall.

As future work, it would be interesting to look into the following directions:

1. Derive finite time bounds that show a lower Hessian estimation error for 2RDSA-IH when compared to 2RDSA and 2SPSA.

2. Stochastic Newton methods converge faster and are often more accurate than simple gradient search schemes. However they are computationally much costlier than first-order

methods. Since first-order methods exhibit slower convergence rate and heavily depend on the step-size selection one has to resort to second-order methods for improved performance. As a future work, it would be interesting to look at methods such as momentum descent, conjugate gradient as well as, Hessian free optimization approaches which result in faster convergence for the first-order methods and are known to be computationally cheaper than second-order methods in the deterministic optimization scenario. It would be interesting to extend those methods to the stochastic optimization setting and prove that they exhibit faster convergence than first-order methods and result in lower computational cost.

3. Newton scheme is not popular due to expensive matrix inversion at each parameter update step even in the deterministic case. Quasi-Newton schemes given by the Broyden family may be used instead. As a future work, it would be interesting to develop stochastic approximation versions of the quasi-Newton schemes. For some initial work in this direction, see [7],[18].

4. In the methods described in this thesis the perturbations used are random variables. In [4], for the first-order SPSA method, construction of two different deterministic perturbation schemes was proposed and under certain conditions on the perturbation vectors and noise sequences, the algorithm is shown to converge. The idea behind using deterministic perturbations is to reduce the bias in the gradient estimates. It would be interesting to extend this construction to the RDSA scheme as well. And moreover this kind of construction for second-order methods is not available for both SPSA and RDSA. One can use the above ideas and show significant improvements to the existing methods.

5. From an application point of view it would be interesting to apply these methods to some interesting real world applications. It would be interesting to explore applications of these methods to the design of reinforcement learning algorithms.

Bibliography

[1] Dan Anbar. A stochastic newton-raphson method. *Journal of Statistical Planning and Inference*, 2(2):153–163, 1978. 12

[2] R. E. Bechhofer. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *The Annals of Mathematical Statistics*, pages 16–39, 1954. 4, 5

[3] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999. 16

[4] S. Bhatnagar, M. C. Fu, S. I. Marcus, and I Wang. Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):180–209, 2003. 55

[5] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth. *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods (Lecture Notes in Control and Information Sciences)*, volume 434. Springer, 2013. 9

[6] S. Bhatnagar and L. A. Prashanth. Simultaneous perturbation newton algorithms for simulation optimization. *Journal of Optimization Theory and Applications*, 164(2):621–643, 2015. ii, 14, 36, 39, 54

[7] Antoine Bordes, Léon Bottou, and Patrick Gallinari. Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, 10(Jul):1737–1754, 2009. 55

[8] C-H. Chen. *Stochastic simulation optimization: an optimal computing budget allocation*, volume 1. World scientific, 2010. 5

[9] C-H. Chen, J. Lin, E. Yücesan, and S. E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 10(3):251–270, 2000. 5

[10] D. C. Chin. Comparative study of stochastic algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 27(2):244–249, 1997. 11

[11] C. W. Dunnett. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*, 50(272):1096–1121, 1955. 4

[12] V. Fabian. Stochastic approximation. In *Optimizing Methods in Statistics (ed. J.J.Rustagi)*, pages 439–470, New York, 1971. Academic Press. 12

[13] F. Glover. Tabu search and adaptive memory programming?advances, applications and challenges. In *Interfaces in computer science and operations research*, pages 1–75. Springer, 1997. 6

[14] S. S. Gupta. On some multiple decision (selection and ranking) rules. *Technometrics*, 7(2):225–245, 1965. 5

[15] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984. 6

[16] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Verlag, 1978. 11

[17] R. G. Laha and V. K. Rohatgi. *Probability Theory*. Wiley, New York, 1979. 24, 25

[18] K. Lakshmanan and S. Bhatnagar. Smoothed functional and quasi-newton algorithms for routing in multi-stage queueing network with constraints. In *International Conference on Distributed Computing and Internet Technology*, pages 175–186. Springer, 2011. 55

[19] Mikhail Borisovich Nevel'son and Rafail Zalmanovich Khas' minskii. *Stochastic approximation and recursive estimation*. American Mathematical Society Providence, 1973. 41

[20] E. Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, pages 174–180, 1964. 5

[21] L. A. Prashanth, S. Bhatnagar, M. C. Fu, and S. Marcus. Adaptive system optimization using random directions stochastic approximation. *IEEE Transactions on Automatic Control (To appear)*, 2017. ii, 11, 13, 14, 15, 19, 21, 22, 23, 24, 26, 28, 29, 38, 39, 54

[22] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951. 8

[23] R. Y. Rubinstein and D. P. Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013. 6

[24] Steve Smale. A convergent process of price adjustment and global newton methods. *Journal of Mathematical Economics*, 3(2):107–120, 1976. 12

[25] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Auto. Cont.*, 37(3):332–341, 1992. 10, 28

[26] J. C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems*, 34(3):817–823, 1998. 10

[27] J. C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Trans. Autom. Contr.*, 45:1839–1853, 2000. 13, 46

[28] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, volume 65. John Wiley & Sons, 2005. 9

[29] J. C. Spall. Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm. *IEEE Trans. Autom. Contr.*, 54(6):1216–1229, 2009. ii, iii, 14, 15, 20, 23, 24, 33, 39, 41, 42, 43, 54

[30] J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben. A survey of recent advances in discrete input parameter discrete-event simulation optimization. *IIE Transactions*, 36(6):591–600, 2004. 5

[31] J. R. Swisher, S. H. Jacobson, and E. Yücesan. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):134–154, 2003. 5