Monte Carlo Methods and Stochastic Approximation : Theory and Applications to Machine Learning

Rémi Leluc

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT007

Thèse de doctorat

Monte Carlo Methods and Stochastic Approximation: Theory and Applications to Machine Learning

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°574 Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques aux interfaces

Thèse présentée et soutenue à Palaiseau, le 21 Mars 2023, par

RÉMI LELUC

Composition du Jury :

Nicolas CHOPIN Professor, ENSAE-CREST	Président
Sébastien GADAT Professor, Toulouse School of Economics	Rapporteur
Christian P. ROBERT Professor, Université Paris-Dauphine (CEREMADE)	Rapporteur
Francis BACH Professor, INRIA et ENS	Examinateur
Alexandra CARPENTIER Professor, Université de Potsdam	Examinatrice
Panayotis MERTIKOPOULOS Tenured Researcher, CNRS (POLARIS)	Examinateur
François PORTIER Associate Professor, ENSAI-CREST	Directeur de thèse
Pascal BIANCHI Professor, Télécom Paris (LTCI)	Co-directeur de thèse
Johan SEGERS Professor, UCLouvain (LIDAM/ISBA)	Invité

À mes grands-parents,
Louise et François,
Geneviève et Maurice,

Remerciements

"La raison la plus motivante de travailler, à l'école ou dans la vie, se trouve dans le plaisir que l'on y trouve, dans le plaisir du résultat atteint et dans la connaissance de la valeur de ce résultat pour la communauté." Albert Einstein

Je suis honoré et reconnaissant d'avoir achevé mon parcours de doctorant, et je tiens à saisir cette occasion pour exprimer mes sincères remerciements à toutes les personnes qui m'ont aidé et soutenu tout au long de ce parcours.

Tout d'abord, je tiens à exprimer ma profonde gratitude à mon directeur de thèse François Portier, dont l'expertise scientifique et l'esprit brillant ont été pour moi une source constante d'inspiration et de motivation tout au long de mon parcours. Je tiens à saluer son engagement, son enthousiasme et son dévouement à la recherche. Sa capacité à me motiver et la confiance qu'il m'a accordée ont joué un rôle déterminant dans la formation de mes compétences et mon approche de la recherche. Je remercie François pour sa disponibilité, ses conseils, sa bonne humeur, pour sa nature bienveillante et son intérêt sincère dans le bon déroulement de la thèse. Toutes ces attentions ont rendu ce voyage doctoral d'autant plus agréable et j'apprécie infiniment le temps et les efforts consacrés à me fournir un retour d'information réfléchi et constructif. Je suis également reconnaissant envers mon codirecteur de thèse Pascal Bianchi pour sa disponibilité, sa bienveillance, ses précieux conseils et commentaires qui m'ont aidé à affiner mes recherches et mon approche de la résolution de problèmes.

Je souhaite ensuite remercier les rapporteurs Sébastien Gadat et Christian Robert qui ont généreusement consacré leur temps à lire et à évaluer attentivement le manuscrit de ma thèse, et qui ont fourni des commentaires constructifs visant à améliorer la qualité de mon travail. Vos commentaires et suggestions perspicaces ont été d'une valeur inestimable, et je vous suis vivement reconnaissant.

À tous les membres du jury qui étaient présents lors de la soutenance de mon doctorat – Francis Bach, Alexandra Carpentier, Nicolas Chopin, Sébastien Gadat, Panayotis Mertikopoulos et Christian Robert – j'exprime ma sincère gratitude pour leur présence et l'intérêt qu'ils ont porté à ma recherche. Leurs questions stimulantes et leurs remarques pertinentes ont été très instructives, et je suis reconnaissant des efforts qu'ils ont déployés pour faire de cette soutenance une expérience précieuse d'apprentissage.

Je tiens à remercier les différents co-auteurs associés aux travaux de cette thèse pour leurs contributions à nos publications conjointes. Leur dévouement, leur travail acharné et leurs analyses ont été d'une valeur inestimable pour le succès de nos collaborations. Ce fut un honneur de travailler avec chacun d'entre vous et je vous suis profondément reconnaissant pour les connaissances et les compétences que vous avez partagées avec moi. Je voudrais en particulier remercier Johan Segers. Son mentorat, ses conseils et son amitié ont joué un rôle certain dans mon développement en tant que mathématicien. Merci de m'avoir donné l'occasion d'apprendre à tes côtés lors d'un séjour en Belgique et d'avoir cru en mon potentiel.

Je remercie les chercheurs du laboratoire S2A qui m'ont permis de découvrir l'activité d'enseignement au cours de mon parcours doctoral: François Roueff, Anne Sabourin, Olivier Fercoq et Florence d'Alché-Buc.

Je souhaite également remercier tous les doctorants et post-doctorants de l'équipe S2A du laboratoire LTCI, ainsi que l'ensemble des membres de l'école doctorale de mathématiques Hadamard et de Télécom Paris dans son ensemble. Votre présence, votre soutien et votre amitié ont fait de mon doctorat une expérience plus agréable et plus enrichissante.

Merci aux doctorants de l'ancienne génération – Emile, Mastane, Kévin, Kamélia, Nidham, Robin, Hamid, Pierre, Amaury, Kimia, Anas, Pierre et Lucien – pour leurs précieux conseils scientifiques et administratifs. Merci aux camarades de la même génération – Luc, Dimitri et Vincent – avec qui tout a commencé dans les anciens locaux rue Barrault. Bon courage enfin à la génération future: Elie, Joël, Anas, Khalid, Victor, Iyad, Emilia, Yazid, Junjie, Tamim et Arturo.

À mes amis proches, qui ont toujours été là pour moi, m'offrant leur soutien et leurs encouragements sans faille, merci. Votre amitié et votre gentillesse m'ont aidé à relever les défis de ce voyage. Je salue mes amis du lycée Condorcet et de la bande des machines avec Thibault, Agathe, Vincent, Alexane, Xavier, Valentine, Juliette, Ulysse, Jason et Félix, ainsi que les camarades du Master MVA Gauthier et Leello.

Enfin, je tiens à exprimer ma profonde gratitude aux membres de ma famille, à commencer par mes parents, pour leur soutien et leurs encouragements indéfectibles. Votre amour et votre soutien ont été une source de force constante tout au long de ce parcours. À mon frère et mes soeurs, merci d'avoir toujours été là pour moi, de m'avoir encouragé et motivé lorsque j'en avais le plus besoin.

Merci enfin à ma chère et tendre Pauline, future médecin anesthésiste, pour la confiance, le soutien, la joie et l'amour que tu m'apportes quotidiennement. Tous ces éléments ont été déterminants pour décrocher cette distinction académique et j'attends avec impatience le moment où tu obtiendras également le titre de Docteur. Ces années de thèse seront toujours indissociables de ces premières années ensemble.

Publications

- **R. Leluc**, F. Portier and J. Segers. Control Variate Selection for Monte Carlo Integration. In *Statistics and Computing 31, 50*, pages1-27, 2021.
- H. Jalalzai and **R. Leluc**. Feature Clustering for Support Identification in Extreme Regions. In *International Conference on Machine Learning (ICML)*, pages 4733-4743, 2021.
- **R. Leluc**, F. Portier, J. Segers and A. Zhuman. A Quadrature Rule combining Control Variates and Adaptive Importance Sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11842-11853, 2022.
- **R. Leluc** and F. Portier. SGD with Coordinate Sampling: Theory and Practice. In *Journal of Machine Learning Research 23 (JMLR)*, (342):1–47, 2022.
- **R. Leluc** and F. Portier. Asymptotic Analysis of Conditioned Stochastic Gradient Descent. *arXiv preprint 2006.02745*, 2022. (submitted)
- **R. Leluc**, E. Kadoche, A. Bertoncello and S. Gourvénec. MARLIM: Multi-Agent Reinforcement Learning for Inventory Management. In *NeurIPS Workshop on Reinforcement Learning for Real Life*, 2022.
- H. Jalalzai, E. Kadoche, **R. Leluc** and V. Plassier Membership Inference Attacks via Adversarial Examples. In *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning*, 2022.
- **R. Leluc**, F. Portier, J. Segers and A. Zhuman. Speeding up Monte Carlo Integration: Nearest Neighbors as Control Variates *arXiv preprint*, 2022. (submitted)

Abstract

Across a breadth of research areas, whether in Bayesian inference, reinforcement learning or variational inference, the need for accurate and efficient computation of integrals and parameters minimizing risk functions arises, making stochastic optimization and Monte Carlo methods one of the fundamental problems of statistical and machine learning research. This thesis focuses on Monte Carlo integration and stochastic optimization methods, both from a theoretical and practical perspectives, where the core idea is to use randomness to solve deterministic numerical problems. From a technical standpoint, the study is mainly based on two standard concepts: variance reduction and adaptive sampling techniques.

The first part of the thesis focuses on various control variates techniques for Monte Carlo integration. The study is based on mathematical tools coming from probability theory and statistics aiming to understand the behavior of certain existing algorithms and to design new ones with thorough analysis of the integration error. First, we present a LASSO-type procedure to allow the use of high-dimensional control variates. Then, a weighted least-squares estimate, called AISCV, is proposed to incorporate control variates within the adaptive importance sampling framework. Finally, a Monte Carlo method with control variates based on nearest neighbors estimates, called Control Neighbors, is provided.

The second part of the thesis deals with stochastic optimization algorithms. First, we investigate a general class of stochastic gradient descent (SGD) algorithms, called conditioned SGD, based on a preconditioning of the gradient direction. Using a discrete-time approach with martingale tools, we establish the weak convergence of the rescaled sequence of iterates for a broad class of conditioning matrices including stochastic first-order and second-order methods. Then we present a general framework to perform coordinate sampling for SGD algorithms. While classical forms of SGD algorithms treat the different coordinates in the same way, a framework allowing for adaptive (non uniform) coordinate sampling is developed to leverage structure in data. In a non-convex setting and including zeroth order gradient estimate, almost sure convergence as well as non-asymptotic bounds are provided. Within this framework, we develop an algorithm, MUSKETEER, based on a reinforcement strategy: after collecting information on the noisy gradients, it samples the most promising coordinate (all for one); then it moves along the one direction yielding an important decrease of the objective (one for all).

To emphazise the practical applications of the proposed methods, all algorithms are implemented and tested against state-of-the-art procedures and extensive numerical experiments are provided to allow reproducibility. All algorithms developed in this thesis are open-sourced and available online.

Résumé

Dans de nombreux domaines de recherche, que ce soit l'inférence variationnelle, l'inférence Bayésienne ou l'apprentissage par renforcement, le besoin d'un calcul précis et efficace d'intégrales et de paramètres minimisant des fonctions de risque apparaît, faisant des méthodes d'optimisation stochastiques et de Monte Carlo l'un des problèmes fondamentaux de la recherche en statistique et en apprentissage automatique. Cette thèse se concentre sur des méthodes d'intégration par Monte Carlo et d'optimisation stochastique, tant d'un point de vue théorique que pratique, où l'idée centrale est d'utiliser l'aléatoire pour résoudre des problèmes numériques déterministes. D'un point de vue technique, l'étude se concentre sur la réduction de la variance et des techniques d'échantillonnage adaptatif.

La première partie de la thèse se concentre sur diverses techniques de variables de contrôle pour l'intégration de Monte Carlo. L'étude est basée sur des outils mathématiques issus de la théorie des probabilités et des statistiques visant à comprendre le comportement de certains algorithmes existants et à en concevoir de nouveaux avec une analyse approfondie de l'erreur d'intégration. Tout d'abord, nous présentons une procédure de type LASSO pour permettre l'utilisation de variables de contrôle en grande dimension. Ensuite, une estimation pondérée des moindres carrés, appelée AISCV, est proposée pour incorporer les variables de contrôle dans le cadre de l'échantillonnage adaptatif par importance. Enfin, une méthode de Monte Carlo avec des variables de contrôle basée sur des estimateurs des plus proches voisins, appelée Control Neighbors, est proposée.

La deuxième partie de la thèse traite des algorithmes d'optimisation stochastique. Tout d'abord, nous étudions une classe générale d'algorithmes de descente de gradient stochastique (SGD), appelée SGD conditionnée, basée sur un préconditionnement de la direction du gradient. En utilisant une approche en temps discret avec des outils de martingale, nous établissons la convergence faible de la séquence rééchelonnée des itérés pour une large classe de matrices de conditionnement, y compris les méthodes stochastiques du premier et du second ordre. Nous présentons ensuite un cadre général pour effectuer l'échantillonnage des coordonnées pour les algorithmes SGD. Alors que les formes classiques d'algorithmes SGD traitent les différentes coordonnées de la même manière, un cadre permettant l'échantillonnage adaptatif (non uniforme) des coordonnées est développé pour exploiter la structure des données. Dans un cadre non convexe et en incluant une estimation du gradient d'ordre zéro, une convergence presque certaine ainsi que des limites non asymptotiques sont fournies. Dans ce cadre, nous développons un algorithme, MUSKETEER, basé sur une stratégie de renforcement : après avoir collecté des informations sur les gradients bruités, il échantillonne la coordonnée la plus prometteuse (tous pour un) ; puis il se déplace dans la direction qui entraîne une diminution importante de l'objectif (un pour tous).

Pour souligner les applications pratiques des méthodes proposées, tous les algorithmes sont implémentés et testés par rapport aux méthodes de l'état de l'art et des expériences numériques approfondies sont fournies pour permettre la reproductibilité. Tous les algorithmes développés dans cette thèse sont libres de droits et disponibles en ligne.

Thesis outline and reading guide

Outline

This thesis contains an introductory part (Part I) and is then divided into two main parts. The first main part (Part II) is composed of three chapters and tackles different variance reduction techniques based on control variates in the framework of Monte Carlo methods. The second main part (Part III) is composed of two chapters and investigates the use of conditioning matrices for stochastic approximation algorithms.

Part I contains one introductory Chapter.

- Chapter 1 is a general introduction about the theory and applications of Monte Carlo methods and stochastic approximation. It introduces the high level context of these research topics needed to read this thesis. It presents the main results of each chapter and provides a detailed outline of the rest of the thesis.

Part II focuses on control variates techniques for Monte Carlo integration.

- Chapter 2 deals with the use of high-dimensional control variates with the help of a LASSO-type procedure. Monte Carlo integration with variance reduction by means of control variates can be implemented by the ordinary least squares estimator for the intercept in a multiple linear regression model with the integrand as response and the control variates as covariates. Regularizing the ordinary least squares estimator by preselecting appropriate control variates via the LASSO turns out to increase the accuracy without additional computational cost. The findings in the numerical experiment are confirmed by concentration inequalities for the integration error.
  This Chapter is based on the journal paper Leluc et al. (2021).

- Chapter 3 combines control variates with adaptive importance sampling. Standard control variates methods do not allow the distribution of the particles to evolve during the algorithm, as is the case in sequential simulation methods. Within the standard adaptive importance sampling framework, a simple weighted least squares approach is proposed to improve the procedure with control variates. The procedure takes the form of a quadrature rule with adapted quadrature weights to reflect the information brought in by the control variates. The quadrature points and weights do not depend on the integrand, a computational advantage in case of multiple integrands. Our main result is a non-asymptotic bound on the probabilistic error of the procedure. The bound proves that for improving the estimate's accuracy, the benefits from adaptive importance sampling and control variates can be combined. The good behavior of the method is illustrated empirically on synthetic examples and real-world data for Bayesian linear regression. This Chapter is based on the conference paper Leluc et al. (2022)

- Chapter 4 has a more theoretical flavor by focusing on optimal convergence rates for the iteration error. Monte Carlo integration is a widespread technique to solve numerical integration problems with applications ranging from computational biology and engineering to finance and machine learning. While the standard Monte

Carlo estimate is easy and fast to compute, its $O(n^{-1/2})$ error rate may not be optimal for particular applications. This chapter provides a novel integration rule called *control neighbors* based on nearest neighbor estimates acting as control variates to speed up the convergence rate of the Monte Carlo procedure. The main result is the $O(n^{-1/2}n^{-1/d})$ convergence rate of this new estimate for Lipchitz functions, which is, in some sense, the best rate possible. Several numerical experiments validate the complexity bound and highlight the good performance of the proposed estimator.
This Chapter is related to a preprint version at the time of submission.

Part III deals with stochastic optimization algorithms.

- Chapter 5 investigates a general class of stochastic gradient descent (SGD) algorithms, called *conditioned SGD*, based on a preconditioning of the gradient direction. Using a discrete-time approach with martingale tools, we establish the weak convergence of the rescaled sequence of iterates for a broad class of conditioning matrices including stochastic first-order and second-order methods. Almost sure convergence results, which may be of independent interest, are also presented. When the conditioning matrix is an estimate of the inverse Hessian, the algorithm is proved to be asymptotically optimal. For the sake of completeness, a practical procedure to achieve this minimum variance is provided.
  This Chapter is based on the preprint version Leluc and Portier (2020).

- Chapter 6 presents the framework of stochastic gradient descent with coordinate sampling. While classical forms of stochastic gradient descent algorithm treat the different coordinates in the same way, a framework allowing for adaptive (*non uniform*) coordinate sampling is developed to leverage structure in data. In a non-convex setting and including zeroth order gradient estimate, almost sure convergence as well as non-asymptotic bounds are established. Within the proposed framework, we develop an algorithm, MUSKETEER, based on a reinforcement strategy: after collecting information on the noisy gradients, it samples the most promising coordinate (*all for one*); then it moves along the one direction yielding an important decrease of the objective (*one for all*). Numerical experiments on both synthetic and real data examples confirm the effectiveness of MUSKETEER in large scale problems.
  This Chapter is based on the journal paper Leluc and Portier (2022).

The final Chapter 7 is a conclusion and highlights the different research directions opened up to us by this thesis.

Reading guide

Each chapter of the main parts contains a small introduction which describes the necessary elements of context. It is then followed by a *verbatim* of the article related to the chapter, where all the precise results and proofs can be found. Note that full articles and appendices are gathered for this thesis to be self-contained. For each chapter, the verbatim articles are divided into main sections, which give context and results, and auxiliary sections, where most of the technical proofs may be found. Note that each chapter can be read independently.

For a quick overview of the different contributions presented in this thesis, the reader is invited to focus on the summary of contributions in Section 1.4 of Chapter 1.

Contents

Notation

$:=$	Equal by definition
$\mathbb{N}, \mathbb{R}$	Sets of natural and real numbers
$\mathbb{R}^d$	Set of d-dimensional real-valued vectors
$\langle x, y \rangle$	Inner product of vectors $x, y \in \mathbb{R}^d$
$\|x\|_p$	ℓ_p-norm of vector $x \in \mathbb{R}^d$
$\|A\|$	Matrix norm induced $\|A\| = \sup\{\|Au\| : u \in \mathbb{R}^p, \|u\| = 1\}$
$\mathbb{R}^{n \times d}$	Set of real matrices of size $n \times d$
$\mathcal{S}_d(\mathbb{R})$	Set of real symmetric matrices of size $d \times d$
$\mathcal{S}_d^+(\mathbb{R}), \mathcal{S}_d^{++}(\mathbb{R})$	Set of real symmetric positive (semi)-definite matrices of size $d \times d$
I_d	Identity matrix of size $d \times d$
$A^\top$	Transpose of matrix A
$\mathrm{Tr}(A), det(A)$	Trace and Determinant of matrix A
$\lambda_{\min}(A), \lambda_{\max}(A)$	Smallest and Largest eigenvalue of matrix A
$A \otimes B$	Kronecker product of A and B
$vec(A)$	Vectorization of matrix A by stacking its columns
$\mathrm{supp}(\cdot)$	Support of a function or a vector
$\mathcal{B}(\mathcal{X})$	Borel σ-field on $\mathcal{X}$
$\mathbb{1}_E$	Characteristic function of set E
A^c	Complementary set of set A
$\mathbb{P}(\cdot)$	Probability of an event
$\mathbb{E}[\cdot]$	Expectation of a random variable
$\overset{\text{i.i.d.}}{\sim}$	Independent and Identically Distributed
$L_2(\pi)$	Set of square integrable functions with respect to measure π
$X \sim \pi$	Random variable X has distribution π
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance matrix Σ
∇f	Gradient function of $f : \mathbb{R}^d \to \mathbb{R}$
$\nabla^2 f$	Hessian matrix of $f : \mathbb{R}^d \to \mathbb{R}$

Part I

Introduction & Preliminaries

"Good and evil, reward and punishment, are the only motives to a rational creature: these are the spur and reins whereby all mankind are set on work, and guided."

(**John Locke**, Some Thoughts Concerning Education, 1693)

Chapter 1

General Introduction, Motivations and Contributions

Contents

This Chapter provides a high-level exposition of the main tools of this thesis, namely Monte Carlo methods and stochastic optimization. First, Section 1.1 motivates the use of random methods in statistical and machine learning applications. Then, the key concepts of Monte Carlo methods and stochastic optimization algorithms are presented in Sections 1.2 and 1.3 with a focus on research questions. Finally, Section 1.4 gives a summary of the main contributions of this thesis.

1.1 Numerical Integration and Gradient Estimation

1.1.1 Motivations for a stochastic approach

For the last fifty years, the computations of integrals and gradients of an expectation have been a key factor in the development of the computational sciences. This calculation lies at the heart of modern machine learning algorithms and can be found in a wide variety of applications ranging from object detection (Carion et al., 2020) and natural language processing (Hirschberg and Manning, 2015) to pricing of financial derivatives (Glasserman, 2004) and complex biological tasks (Jumper et al., 2021). However, solving a numerical integration problem or computing a gradient is not without complexity as it can involve *(i) theoretical problems*, when one faces analytical intractability and *(ii) practical difficulties*, when one encounters computational issues.
In this context, the need for accurate and efficient computation of integrals appears, making the numerical integration problem one of the fundamental problems of statistical and machine learning research. This main question may be written through the lens of a generic probabilistic function $\mathcal{F}$ of the following form

$$\mathcal{F}(\theta) = \mathbb{E}_{\pi_\theta(x)}[f(x)] = \int_{\mathcal{X}} f(x)\pi_\theta(x)\mathrm{d}x. \tag{1.1}$$

This objective function consists in evaluating the expectation of a cost function f with respect to an input distribution $\pi_\theta(x)$ parameterized by a distributional parameter θ. The underlying numerical integration problem in Eq.(1.1) naturally appears in many machine learning applications such as Bayesian inference where one is interested in integrating particular cost functions f against the *posterior distribution* to measure the uncertainty of a model parameter or in variational inference where the goal is to

approximate complex unknown distributions (see details below). Furthermore, observe that in the precise framework of variational inference, one is interested in optimizing the objective $\mathcal{F}$ with respect to the distributional parameter θ. Thus, in the perspective of sequential algorithms, many integrals – one for each new value of the parameter θ – are actually needed for particular applications. This calls for the construction of efficient estimators that can handle many integrands with potentially complex target densities.

In order to learn the optimal distributional parameter θ defined as the arg min of $\mathcal{F}$, one may compute the gradient of Eq.(1.1). When the measure π_θ is differentiable with respect to θ then the gradient is equal to

$$\mathcal{G} = \nabla_\theta \mathcal{F}(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta(x)}[f(x)]. \tag{1.2}$$

This last equation is the *sensitivity analysis* of $\mathcal{F}$ (Mohamed et al., 2020) and refers to the impact of changes in expected performance upon changes of some of the input parameter θ. The computation of this gradient is of great importance not only to study how various sources of uncertainty contribute to the model's overall uncertainty but also for optimization purposes. Indeed, efficient gradient estimators combined with fast optimization methods are the key ingredients for training today's machine learning models. However, the gradient estimation problem in Eq.(1.2) can be difficult to solve in general. The main issues come from *(i) theoretical intractability* since there is not necessarily a closed expression of the gradient and from *(ii) computational expense* as the integrals over x can be high-dimensional.

All these challenges may be tackled by using both Monte Carlo estimates[1] of the gradients and efficient stochastic optimization procedures. These methods have become more and more popular as their inherent randomness provides several advantages compared to deterministic methods:

(a) Easy and Practical. Monte Carlo standard approach requires only three steps – sampling, evaluating, averaging – or equivalently three lines of code from an algorithmic point of view, making it one of the most spread technique to approximate unknown quantities. This ease of implementation makes Monte Carlo methods simple and practical to solve intractable problems, especially for black-box models.

(b) Randomness as a Strength. The inherent randomness of Monte Carlo methods is of great benefit for deterministic numerical computation. For example, when employed for optimization, the randomness permits stochastic algorithms to naturally escape local optima (Gadat et al., 2018). When computing an integral, a fine tuning of the sampling mechanism enables a complete exploration of the search space, a feature which is not usually shared by their deterministic counterparts.

(c) Scalable. Monte Carlo algorithms tend to be simple, flexible, and scalable. Monte Carlo algorithms are eminently parallelizable, in particular when various parts can be run independently. This allows the parts to be run on different computers or processors, therefore significantly reducing the computation time. Similarly in stochastic optimization, the use of gradient estimates is the key to treat large-scale learning problems with a very large number of training samples. For instance, in *supervised learning* with n samples in dimension d, the computation of a deterministic gradient scales as $O(nd)$ while its stochastic version reduces this cost to $O(d)$ operations.

[1] Monte Carlo methods are a large class of computational algorithms that rely on repeated random sampling to obtain numerical results. The core idea is to *use randomness to solve problems that are deterministic in principle.* For a detailed introduction and overview of Monte Carlo methods, one may refer to the textbook of Robert and Casella (1999).

(d) Theoretical justifications. There is a vast body of mathematical and statistical knowledge underpinning Monte Carlo techniques, *e.g.* unbiasedness and consistency, allowing precise statements on the accuracy of a given Monte Carlo estimator or the efficiency of Monte Carlo algorithms. As detailed in Novak (2016), stochastic integration rules offer some advantages in terms of complexity rates. Consider a d-dimensional integrand with bounded s first derivatives and an integration procedure based on n particles. Compared to deterministic methods with complexity rates in $\mathrm{O}(n^{-s/d})$, the optimal convergence rate of a random procedure is $\mathrm{O}(n^{-1/2}n^{-s/d})$. This complexity rate is informative as it advocates the use of random methods over deterministic integration rules since random methods have some $\mathrm{O}(n^{-1/2})$ gain compared to deterministic counterparts.

1.1.2 Key examples

To highlight the question of numerical integration and gradient computation, we provide details in four key examples: reinforcement learning, bayesian inference, variational inference and computational finance. For each one of them, the explicit form of the objective function $\mathcal{F}$ is given.

Reinforcement Learning. In model-free reinforcement learning (Sutton and Barto, 2018), one learns a policy π – a distribution over actions – which maximises, on average, the accumulation of long-term rewards. Consider a Markov Decision Process with finite horizon T. Denote by $\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1})$ a trajectory, *i.e.*, a sequence of states and actions of length T such that $s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_t \sim p(\cdot|s_{t-1}, a_{t-1})$ for $t = 1, \ldots, T-1$ and some policy π. The probability density $\pi_\theta(\tau)$ of the trajectory τ that is generated by following policy π_θ, is given by

$$\pi_\theta(\tau) = \mu(s_0)\pi_\theta(a_0|s_0) \prod_{t=1}^{T-1} p(s_t|s_{t-1}, a_{t-1})\pi_\theta(a_t|s_t).$$

Introduce $\mathcal{R}(\tau) = \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$ the discounted cumulative return of the path τ. It is a random variable both because the path τ itself is a random variable and because even for a given path, each of the rewards sampled in it may be stochastic. Denote by $\mathcal{T}_\theta$ the set of all trajectories than can be generated using a policy π_θ. In this context, the performance of the policy π_θ can be written as

$$\mathcal{F}(\theta) = \mathbb{E}_{\pi_\theta(\tau)}[\mathcal{R}(\tau)] = \int_{\mathcal{T}_\theta} \mathcal{R}(\tau)\pi_\theta(\tau)\mathrm{d}\tau.$$

The computation of the gradient relies on the well-known policy gradient theorem. Combined with gradient-based optimization methods, it has been the root of many successful applications such as playing board games (Mnih et al., 2015; Silver et al., 2018; Vinyals et al., 2019), robotics (Kober et al., 2013), autonomous driving (Okuda et al., 2014; Sallab et al., 2017) or biological tasks (Jumper et al., 2021). The gradient is given by

$$\mathcal{G} = \nabla_\theta \mathcal{F}(\theta) = \mathbb{E}_{p_\theta(\tau)}[\mathcal{R}(\tau)\nabla_\theta \log p_\theta(\tau)].$$

For the gradient estimate, the frequentist approach is to use a Monte Carlo approximation to compute the expectation, which leads to the algorithm REINFORCE (Williams, 1992). After collecting many trajectories $\tau_1, \ldots, \tau_n \sim p_\theta$ according to the current distribution p_θ, the Monte Carlo gradient estimate is simply an average over the evaluations $\mathcal{R}(\tau_i)\nabla_\theta \log p_\theta(\tau_i)$.

Bayesian Inference. Statistical inference is the process of modelling a phenomenon given some data. Bayesian inference is a type of statistical inference that takes into account *prior knowledge* about the model parameters when fitting a probability model to observed data. Assume that we have access to some observed variables $\mathcal{D}$ generated from a dominated probabilistic model with density $p(\mathcal{D}|\theta)$ parameterized by a hidden random variable $\theta \in \Theta$ that is drawn from a certain prior with density p_0. The cornerstone of Bayesian inference is Bayes'rule which gives the *posterior density* of the latent variable θ given the data $\mathcal{D}$

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p_0(\theta)}{p(\mathcal{D})},$$

where $p(\mathcal{D}) = \int_\Theta p(\mathcal{D}|\theta)p_0(\theta)\mathrm{d}\theta$ is the marginal likelihood or *model evidence*. The posterior density allows to quantify the uncertainty of the parameter θ after observing the data $\mathcal{D}$ through quantities of interest such as the *posterior mean* $\int_\Theta \theta p(\theta|\mathcal{D})\mathrm{d}\theta$. More generally, given a function f defined on Θ, the succes of Bayesian inference methods relies on the ability to compute integrals of the form

$$\int_\Theta f(\theta)p(\theta|\mathcal{D})\mathrm{d}\theta.$$

Typically, this integral is analytically intractable. It is also difficult to approximate numerically, especially when the dimension d of the parameter space Θ is large or when the model is complex. Therefore, it is essential to discover approaches that make Bayesian inference computationally efficient and able to handle large amounts of data.

Since exact Bayesian inference is often impossible, one may rely on approximate Bayesian Inference methods, which mainly fall into two broad categories: *(i)* Monte Carlo methods (*e.g.* Adaptive Importance Sampling (Oh and Berger, 1992), Markov Chain Monte Carlo (Neal, 1993), Sequential Monte Carlo (Del Moral et al., 2006)), that are *sampling* methods; *(ii)* Variational Inference methods (*e.g.* Variational Bayes (Jordan et al., 1999), Expectation Propagation (Minka, 2001)), that rely on optimization techniques. In Bayesian inference, we find yet another thriving area of research where numerical integration and gradient estimation play a fundamental role.

Variational Inference. Variational inference methods (Jordan et al., 1999) are a set of techniques for approximating a complex posterior distribution by a simpler variational density q belonging to some tractable density family $\mathcal{Q}$. These methods can be used in various problems arising from Bayesian inference and machine learning situations where there is a need to approximate a difficult distribution.

In variational inference, one has access to some *observed variables* $x = (x_1, \ldots, x_n)$ which depend on a set of *unobserved* or *latent variables* $z = (z_1, \ldots, z_m)$. The underlying generative process is $p(x, z) = p(x|z)p(z)$ and involves the data distribution $p(x|z)$ and a prior distribution $p(z)$. The associated posterior distribution $p(z|x)$ is typically unknown, and is approximated by a variational distribution $q_\theta(z|x) \in \mathcal{Q}$ over the latent space. Here $\mathcal{Q}$ denotes a parameterized family of distributions with variational parameters θ. For instance, one may think of $\theta = (\mu, \Sigma)$ corresponding to the mean μ and covariance Σ of a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$.

Variational inference methods consider the approximation problem as an optimization problem involving a measure of dissimilarity D between the target posterior distribution $p(z|x)$ and the variational distribution $q_\theta(z|x)$

$$\inf_{q\in\mathcal{Q}} \mathsf{D}(q_\theta(z|x)||p(z|x)).$$

A classical choice is to take D equal to the Kullback-Leibler divergence (Kullback and Leibler, 1951) between the target and candidate distributions

$$\mathrm{KL}(q_\theta(z|x)||p(z|x)) = \int q_\theta(z|x) \log \frac{q_\theta(z|x)}{p(z|x)} \mathrm{d}x = \mathbb{E}_{q_\theta(z|x)}\Big[\log \frac{q_\theta(z|x)}{p(z|x)}\Big].$$

This particular choice combined with Bayes'rule yields an objective $\mathcal{F}$, called the *variational free-energy*. This function $\mathcal{F}$ optimises the log-likelihood $\log p(x|z)$ under a regularization constraint which promotes closeness between the density q and the prior distribution $p(z)$ (Blei et al., 2017). Instead of minimizing the objective $\mathcal{F}$, one may equivalently maximise its opposite, known as the *Evidence Lower BOund* (ELBO) and defined by

$$\mathrm{ELBO} = -\mathcal{F}(\theta) = \mathbb{E}_{q_\theta(z|x)}[\log p(x|z)] - \mathrm{KL}(q_\theta(z|x)||p(z)).$$

The first term describes the probability $p(x|z)$ of the data given the latent variable. When one maximizes the ELBO then it translates in picking those models $q_\theta(z|x)$ in the variational family $\mathcal{Q}$ that better predict the data x. The second term, is the negative KL divergence between our variational model $q_\theta(z|x)$ and the prior over the latent variables $p(z)$. When one maximizes the ELBO this term is pushed towards zero meaning that the two distributions are forced to be close. The optimization procedure requires the gradient of the free energy with respect to the variational parameters θ :

$$\mathcal{G} = \nabla_\theta \mathbb{E}_{q_\theta(z|x)}\Big[\log p(x|z) - \log \frac{q_\theta(z|x)}{p(z)}\Big].$$

Computational Finance. Financial engineering (Glasserman, 2004) is a branch of applied mathematics and computer science where expectations and gradient estimation problems are commonly faced. Whether it be for the pricing of derivatives or a risk analysis, the goal is to evaluate the various potential future outcomes of different investments based on various pricing and return assumptions, in order to select the strategy which offers the highest potential yield. In the standard setting of Black-Scholes option pricing model (Black and Scholes, 1973), the price of an option may be expressed as the expectation $\mathbb{E}_Q$, under the so-called risk-neutral measure, of the payoff discounted to the present value. Consider a contract of European type, which specifies a payoff $f(S_T)$, depending on the level of the underlying asset S_t at maturity $t = T$ with discount factor γ. The value $\mathcal{F}$ of the contract at time $t = 0$, conditional on an underlying value S_0 is given by

$$\mathcal{F} = \mathbb{E}_Q[e^{-\gamma T} f(S_T)].$$

Following the sensitivity analysis of Eq.(1.2), one may look for insights on the gradient value. This gives information to comprehend how future yields could be affected by different pricing suppositions, and creates a precise measure of the financial hazard that an investment strategy will have to face. The gradient with respect to S_0 is the Black-Scholes Delta (Chriss and Chriss, 1997)

$$\mathcal{G} = \nabla_{S_0} \mathbb{E}_Q[e^{-\gamma T} f(S_T)].$$

Note that, in the Black-Scholes model, the gradient above can be computed in closed form. However, in more complex settings, *e.g.* when the payoff function is path-dependent or when the measure is not log-normal, one faces theoretical intractability and there is a need for accurate integral estimation (see Chapter 4 for more details).

1.2 Monte Carlo Integration and Variance Reduction

Motivated by the central question of computing integrals in the form of Eq.(1.1), this section presents Monte Carlo methods[2] for numerical integration with a focus on the *control variates* technique. This flagship problem is at the heart of Part II of this thesis where the goal is to understand the behavior of certain existing algorithms and to design new ones with thorough analysis of the integration error.

1.2.1 Mathematical background

Let $(\mathcal{X}, \mathcal{A}, \pi)$ be a probability space and let X be a random variable with distribution π. Let $f \in L_2(\pi)$ be a square integrable, real-valued function on $\mathcal{X}$ of which one would like to calculate the integral

$$\pi(f) := \int_{\mathcal{X}} f(x)\pi(\mathrm{d}x) = \mathbb{E}_\pi[f(X)].$$

When the function f is unknown or no approximation is sufficiently accurate, one may rely on the following Monte Carlo types of procedures:

- Choose *random* points, called *nodes* or *particles*, $X_1, \ldots, X_n$ in $\mathcal{X}, n \in \mathbb{N}^\star$.
- Evaluate the function at nodes $f(X_1), \ldots, f(X_n)$.
- Compute an approximation of $\pi(f)$ based on $((X_1, f(X_1)), \ldots, (X_n, f(X_n)))$.

Here the integrand f is evaluated exactly, *i.e.*, without any noise. Moreover, the methods are concerned by stochastic integration methods where the particles are random, in contrast to deterministic methods where point grids are fixed by the user.

Running any Monte Carlo algorithm is associated to some computational time. In some cases, for each $x \in \mathcal{X}$, the evaluation $f(x)$ can be given by a single elementary operation. In some other cases, the evaluation of f is heavy. The same can be stated concerning the generation of random variables. Therefore, for any Monte Carlo method, one shall have a particular interest in the following aspects: *(i)* the analysis of the integration error regarding the number of nodes and *(ii)* the computation time of the Monte Carlo estimators.

Let $X_1, ..., X_n \overset{\text{i.i.d.}}{\sim} \pi$ be an independent and identically distributed (*i.i.d.*) random sample from π. The naive Monte Carlo estimator $\hat{\alpha}_n^{\mathrm{mc}}(f)$ of $\pi(f)$ is given by the empirical mean

$$\hat{\alpha}_n^{\mathrm{mc}}(f) = \pi_n(f) := \frac{1}{n}\sum_{i=1}^{n} f(X_i). \tag{1.3}$$

Using the strong law of large numbers and relying on the central limit theorem, the asymptotics of the standard Monte Carlo estimate can be easily stated.

[2] From a historical point of view, an early variant of the Monte Carlo method can be seen in the Buffon's needle experiment (1733), in which the mathematical constant π can be estimated by dropping needles on a floor made of parallel and equidistant strips. Later on, in the 1930s, Enrico Fermi experimented with the Monte Carlo method while studying neutron diffusion but did not publish anything on it. The modern version of the Monte Carlo method was first introduced in Metropolis and Ulam (1949) by John von Neuman, Nicholas Metropolis and Stanislaw Ulam while working on nuclear weapons projects at the Los Alamos National Laboratory.

Proposition 1.1. *Assume* $\pi(|f|) < \infty, \pi(|f|^2) < \infty$ *and define* $\sigma^2(f) = \pi[(f - \pi(f))^2]$. *The Monte Carlo estimator* $\hat{\alpha}_n^{\mathrm{mc}}(f)$ *of* $\pi(f)$ *is unbiased, strongly consistent and has variance* $\sigma^2(f)/n$. *By the central limit theorem, we have the convergence in distribution:*

$$\sqrt{n}(\hat{\alpha}_n^{\mathrm{mc}}(f) - \pi(f)) \xrightarrow[n\to+\infty]{d} \mathcal{N}(0, \sigma^2(f)).$$

The classical estimator $\hat{\sigma}_n^2(f)$ of $\sigma^2(f)$ is $\hat{\sigma}_n^2(f) = n^{-1}\sum_{i=1}^n (f(X_i) - \pi_n(f))^2$. Using Slutsky's Lemma, one can extend the previous proposition to the analysis of $\hat{\sigma}_n^2(f)$ and obtain the convergence in distribution of $(\sqrt{n}/\hat{\sigma}_n(f))(\hat{\alpha}_n^{\mathrm{mc}}(f) - \pi(f))$ towards the standard normal law. This last point if useful to build asymptotically consistent confidence intervals.

Monte Carlo integration typically has an error variance of the form σ^2/n. One way to reduce the error is by sampling with a larger value of n, but the computing time grows with n. Sometimes it is possible to find a way to reduce σ instead. To do this, a new Monte Carlo problem is constructed with the same expected value as the original one but with a lower σ. Methods to do this are known as *variance reduction* techniques and are developed in Chapters 2 to 4 where the focus is on *control variates* and *adaptive importance sampling*. The next section presents the general concepts behind these two variance reduction techniques. First, the method and key questions of *control variates* are provided. Then the framework of (adaptive) importance sampling is presented. Finally, some remarks about the complexity rates of *control variates* are provided as these methods not only allow to perform variance reduction but also to accelerate the convergence speed of standard Monte Carlo estimates.

1.2.2 Variance reduction techniques

Variance reduction with Control Variates. Control variates is based on the following one-sentence principle: *"if you wish to evaluate the (unknown) integral of a certain function you better use functions of which you know the integral"*. The control variates method consists in incorporating this new piece of information, the known integral value of some control functions, in the basic Monte Carlo framework. The aim is to perform *variance reduction*. The basic ideas of control variates are now introduced. These techniques are developed with more details in the next Chapters 2 to 4.

Let $((X_1, Z_1), \ldots, (X_n, Z_n))$ be an independent and identically distributed sequence of random variables in $\mathcal{X} \times \mathbb{R}$ and assume that $f : \mathcal{X} \to \mathbb{R}$ is such that $\mathbb{E}[|f(X_1)|] < \infty$ and that $\mathbb{E}[Z_1]$ is known. The aim of the control variate method is to estimate $\pi(f) = \mathbb{E}[f(X_1)]$ using the knowledge of $\mathbb{E}[Z_1]$. Since the latter is known, one can assume without any loss of generality that $\mathbb{E}[Z_1] = 0$. The control variates class of estimator is

$$\hat{\alpha}_n^{(\mathrm{cv})}(f) = \frac{1}{n}\sum_{i=1}^n (f(X_i) - Z_i). \tag{1.4}$$

The control variates is an extension of Monte Carlo, as taking $Z_1 = 0$ recovers the Monte Carlo estimate. It also includes antithetic variates methods, when taking $Z_1 = (f(X_1) - (f \circ \varphi)(X_1))/2$ where $\varphi : \mathcal{X} \to \mathcal{X}$ is such that $\varphi(X)$ has the same distribution as X. So far one cannot be sure that the introduction of control variates (Z_i) reduces the variance over Monte Carlo as it is not guaranteed that $\mathrm{Var}(f(X_1) - Z_1) \leq \mathrm{Var}(f(X_1))$. Hence it makes sense to parameterize the control variate estimate in order to play

on the influence of the control variates of the estimation. In many examples, one should deal with the observation of several control variates $Z_1, \dots, Z_n$ where for each $i = 1, \dots, n$, $Z_i \in \mathbb{R}^m$. This leads to the following control variates estimate

$$\hat{\alpha}_n^{(\mathrm{cv})}(f, \beta) = \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - \beta^\top Z_i) \tag{1.5}$$

where $\beta \in \mathbb{R}^m$. Note that the special choice $\beta = 0$ recovers the standard Monte Carlo estimate. According to the variance, the best possible choice of β is the one associated to the variance term $\sigma_m^2(f) = \arg\min_{\beta \in \mathbb{R}^m} \mathrm{var}(f(X) - \beta^\top Z)$.

By Hilbert projection theorem, the optimal coefficient $\beta^\star$ is the solution of the normal equation $\mathbb{E}[Z_1 Z_1^T]\beta^\star = \mathbb{E}[Z_1 f(X_1)]$. Intuitively, one shall see that the integration error depends on the accuracy of the approximation in $L_2(\pi)$ of $(f - \pi(f))$ by elements of the form $\beta^\top Z$. For visual interpretation, this is illustrated in Figure 1.1 below which depicts the orthogonal projection of $(f - \pi(f))$ onto the linear space of control variates.

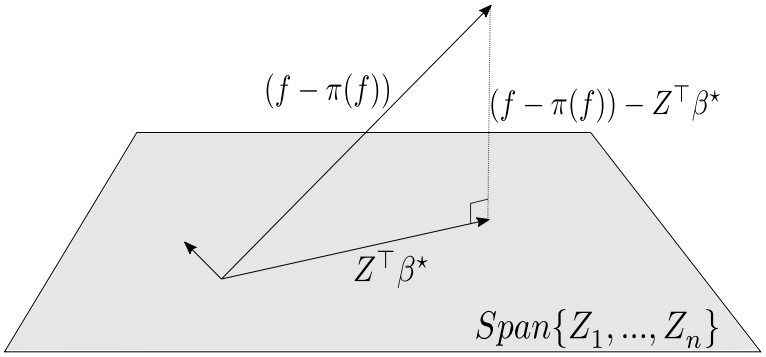


Figure 1.1 – Visualization of L_2-orthogonal projection.

In practice, one may define $\hat{\beta}_n$ as the solution of an *Ordinary Least Squares* (OLS) problem through the empirical normal equations

$$(Z_{n,m} Z_{n,m}^\top)\hat{\beta}_n = Z_{n,m}^\top f_n,$$

where $Z_{n,m} = (Z_1 - \bar{Z}, \dots, Z_n - \bar{Z})^\top$, $f_n = (f(X_1), \dots, f(X_n))^\top$ and $\bar{Z} = n^{-1} \sum_{i=1}^n Z_i$. Among the solutions of the previous equations, we define $\hat{\beta}_n$ as

$$\hat{\beta}_n = (Z_{n,m}^\top Z_{n,m})^+ Z_{n,m}^\top f_n. \tag{1.6}$$

The resulting control variate estimate is obtained by injecting $\hat{\beta}_n$ in Eq.(1.5). The asymptotics of this Monte Carlo estimate are given in the next Proposition. Interestingly, the estimation of $\hat{\beta}_n$ has no effect on the asymptotics.

Proposition 1.2. *Suppose that* $\mathbb{E}[|f(X_1)|] < \infty$, $\mathbb{E}[|f(X_1) Z_{k,1}|] < \infty$ *for* $k = 1, \dots, m$ *and* $\mathbb{E}[Z_1 Z_1^\top]$ *is invertible. The Monte Carlo estimator* $\hat{\alpha}_n^{(\mathrm{cv})}(f, \hat{\beta}_n)$ *of* $\pi(f)$ *is biased and strongly consistent. If moreover* $\mathbb{E}[|f(X_1)|^2] < \infty$ *then we have the convergence in distribution*

$$\sqrt{n}(\hat{\alpha}_n^{(\mathrm{cv})}(f, \hat{\beta}_n) - \pi(f)) \xrightarrow[n \to +\infty]{d} \mathcal{N}(0, \sigma_m^2(f)).$$

The associated variance estimate is $\hat{\sigma}_n^2(f) = n^{-1} \sum_{i=1}^n (f(X_i) - \hat{\beta}_n^\top Z_i - \hat{\alpha}_n^{(\mathrm{cv})}(f, \hat{\beta}_n))^2$. Similarly to the standard Monte Carlo estimate, one may apply the strong law of large numbers to obtain the almost sure convergence of $\hat{\sigma}_n^2(f)$ towards $\sigma_m^2(f)$ and derive

asymptotically consistent confidence intervals using the asymptotic normality of the rescaled process $(\sqrt{n}/\hat{\sigma}_n)(\hat{\alpha}_n^{(\mathrm{cv})}(f, \hat{\beta}_n) - \pi(f))$.

Interestingly, when using only the first ℓ out of n control variates, where $\ell \in \{0, 1, \ldots, n\}$, it holds $\sigma_n^2(f) \leq \sigma_\ell^2(f)$. In terms of asymptotic variance, it therefore never harms to add more control variates. However, in practice, the coefficient β of Eq.(1.6) may become numerically unstable as a growing number of control variates is used.

Asymptotically, the OLS error is bounded by the MC error and is proportional to the L_2 approximation error of the integrand in the linear span of control variates (Glynn and Szechtman, 2002). In combination with well-known approximation results in L_p-spaces (Rudin, 2006), this representation of the OLS error suggests to use an increasing number of control variates. Indeed, in Portier and Segers (2019) it is shown that when m grows with n, the OLS error rate can be faster than $1/\sqrt{n}$.

However, when based on a large number of control variates, the OLS suffers from two classical problems common for least squares methods: *(i)* numerical instabilities when the control variates are nearly collinear, and *(ii)* a computational complexity in $m^3 + nm^2$, which might be prohibitive. These difficulties raise the following research questions.

Research Question #1

How to solve the numerical instability and computational complexity problems when using OLS-based Monte Carlo methods with a large number of *control variates* ? To what extent can one quantify regularization techniques to address the underlying ill-conditioned regression problems ?

Importance sampling. Importance sampling (IS) refers to a collection of Monte Carlo methods where a mathematical expectation with respect to a target distribution is approximated by a weighted average of random draws from another distribution. Recall that the problem is to find the expectation $\pi(f) = \mathbb{E}_\pi[f(X)]$ where X is drawn from a probability density function π. Then for any probability density q that satisfies $q(x) > 0$ whenever $f(x)\pi(x) \neq 0$, one can make a multiplicative adjustement to compensate sampling from q instead of π,

$$\mathbb{E}_\pi[f(X)] = \int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x = \int_{\mathcal{X}} \frac{\pi(x)}{q(x)} f(x)q(x)\mathrm{d}x = \mathbb{E}_q[w(X)f(X)].$$

The distribution q is called the *importance distribution* and the adjustement factor $w(x) = \pi(x)/q(x)$ is called the *likelihood ratio*. A particular interest should be dedicated to the optimal choice of q.

Let $X_1, ..., X_n \overset{\text{i.i.d}}{\sim} q$, then the importance sampling estimate is given by

$$\hat{\alpha}_n^{(\mathrm{is})}(f) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)f(X_i). \tag{1.7}$$

In many applications, the density π is known only up to a normalizing constant. In that case, one may rely on the normalized importance sampling estimate given by

$$\tilde{\alpha}_n^{(\mathrm{is})}(f) = \frac{\sum_{i=1}^n w(X_i)f(X_i)}{\sum_{i=1}^n w(X_i)}. \tag{1.8}$$

Define -whenever these quantities are finite - the variances $v^2(f,\pi)$ and $\tilde{v}^2(f,\pi)$ associated to the importance sampling estimates $\hat{\alpha}_n^{(\mathrm{is})}(f)$ and $\tilde{\alpha}_n^{(\mathrm{is})}(f)$ respectively, *i.e.*

$$v^2(f,\pi) = \mathbb{E}_q[(w(X)f(X) - \pi_f)^2] \quad \text{and} \quad \tilde{v}^2(f,\pi) = \mathbb{E}_q[w(X)^2(f(X) - \pi_f)^2].$$

Similarly to the previous Monte Carlo estimates, the asymptotics of the importance sampling estimates are described by the strong law of large numbers and a central limit theorem.

Proposition 1.3. *Suppose that $\pi(|f|) < \infty$. The Monte Carlo estimator $\hat{\alpha}_n^{(\mathrm{is})}(f)$ of $\pi(f)$ is unbiased and strongly consistent. The Monte Carlo estimator $\tilde{\alpha}_n^{(\mathrm{is})}(f)$ of $\pi(f)$ is biased and strongly consistent. If $\pi(|f|^2) < \infty$ then we have*

$$\sqrt{n}(\hat{\alpha}_n^{(\mathrm{is})}(f) - \pi(f)) \xrightarrow[n\to+\infty]{d} \mathcal{N}(0, v^2(f,\pi)),$$
$$\sqrt{n}(\tilde{\alpha}_n^{(\mathrm{is})}(f) - \pi(f)) \xrightarrow[n\to+\infty]{d} \mathcal{N}(0, \tilde{v}^2(f,\pi)).$$

Adaptive Importance Sampling (AIS). In adaptive importance sampling, $\mathbb{E}_\pi[f]$ is again estimated by a weighted mean over a sample of random particles $X_1, \ldots, X_n$ in $\mathbb{R}^d$. Since appropriate sampling densities naturally depend on f and π, one generally cannot simulate from them. They are then approximated in an adaptive manner by a family of tractable densities $(q_i)_{i\geq 0}$ that often evolve towards a density q_{opt} that optimizes some criterion. The adaptive choice of a sampling policy lies at the heart of many fields of machine learning where former Monte Carlo experiments guide the forthcoming ones. A classical approach is to look for sampling densities that converge towards the target density π. This is illustrated in the one dimensional example of Figure 1.2 below which shows the evolution of the samplers $q_1, q_2, \ldots, q_T$ and a target density π.

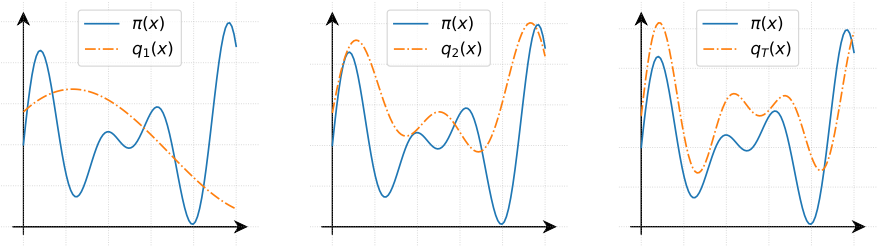


Figure 1.2 – Evolution of sampling policy in Adaptive Importance Sampling.

While the starting density q_0 is fixed, the density q_i for $i \geq 1$ is determined in function of the particles $X_1, \ldots, X_i$ already sampled; think for instance of a parametric family, where the parameter of q_i is a function of $X_1, \ldots, X_i$. Given the particles $X_1, \ldots, X_i$, the next particle, X_{i+1}, is then drawn from q_i. Formally, let $(X_i)_{i\geq 1}$ be a sequence of random vectors on $\mathbb{R}^d$ defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The distribution of the sequence $(X_i)_{i\geq 1}$ is specified by its policy as defined below.

Definition 1.4 (Policy)**.** *A policy is a random sequence of probability density functions $(q_i)_{i\geq 0}$ on $\mathbb{R}^d$ adapted to the σ-field $(\mathcal{F}_i)_{i\geq 0}$ defined by $\mathcal{F}_0 = \{\varnothing, \Omega\}$ and $\mathcal{F}_i = \sigma(X_1, \ldots, X_i)$ for $i \geq 1$. The sequence $(q_i)_{i\geq 0}$ is the policy of $(X_i)_{i\geq 1}$ whenever X_i has density q_{i-1} conditionally on $\mathcal{F}_{i-1}$.*

The (normalized) adaptive importance sampling estimate of $\mathbb{E}_\pi[f]$ is then defined as

$$\hat{\alpha}_n^{(\text{ais})}(f) = \frac{\sum_{i=1}^n w_i f(X_i)}{\sum_{i=1}^n w_i} \quad \text{where} \quad w_i = \frac{\pi(X_i)}{q_{i-1}(X_i)} \quad \text{for } i = 1, \dots, n. \tag{1.9}$$

The sampling weights w_i reflect the fact that X_i has been sampled from q_{i-1} rather than from π. The division by $\sum_{i=1}^n w_i$ rather than by n has two benefits: first, the integration is exact for constant integrands, and second, π needs to be known only up to a proportionality constant, an advantage for Bayesian inference.

Since updating the density q_i at each iteration may be computationally expensive, it is customary to hold it fixed over a pre-determined number of iterations. Writing $n = n_1 + \cdots + n_T$ in terms of positive integers $(n_t)_{t=1}^T$ called the *allocation policy*, the AIS estimate then becomes

$$\hat{\alpha}_T^{(\text{ais})}(f) = \frac{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} f(X_{t,i})}{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i}} \quad \text{where} \quad w_{t,i} = \frac{\pi(X_{t,i})}{q_t(X_{t,i})} \tag{1.10}$$

for $t = 1, \dots, T$ and $i = 1, \dots, n_t$. At stage t, the particles $X_{t,1}, \dots, X_{t,n_t}$ are sampled independently from q_{t-1}, while all particles sampled up to and including stage t are used to determine the sampling density q_t for stage $t+1$. It is easy to see that the two formulations of the AIS estimate are equivalent: (1.9) arises from (1.10) by setting $n_t = 1$ for all t, while (1.10) can be obtained from (1.9) by constructing the policy in such a way that the densities q_i do not change within integer intervals of the form $\{0, \dots, n_1 - 1\}$, $\{n_1, \dots, n_1 + n_2 - 1\}$, and so on. While the shorter representation (1.9) is more convenient for theoretical purposes, formulation (1.10) is the one used in practice (see Section 3.6 in Chapter 3).

Interestingly, the AIS estimate (1.9) may be seen as a weighted least-squares estimate minimizing the loss function $a \mapsto \sum_{i=1}^n w_i (f(X_i) - a)^2$. This property is key to understand the links between *control variates* and *adaptive importance sampling*. To the best of found knowledge, the existing control variates methods do not account for sequential changes in the particle distribution as is the case in adaptive importance sampling.

Research Question #2

While the design of algorithms with adaptive policies has been of major interest recently, only a few studies have focused on using control variates to reduce the variance. How can the benefits of *control variates* technique and *adaptive importance sampling* be combined ?

Complexity Rates. The *control variates* technique not only allows to perform variance reduction but also to accelerate the convergence speed of standard Monte Carlo estimators. As detailed in Novak (2016), the complexity of integration algorithms may be analyzed through the convergence rate of the error. Any randomized procedure based on n particles yields an estimate $\hat{\alpha}_n(f)$ of the integral $\pi(f)$. In this context, the error of the procedure is defined as $\mathbb{E}[|\hat{\alpha}_n(f) - \pi(f)|^2]^{1/2}$. For the specific problem of integration with respect to the uniform measure over the unit cube $[0,1]^d$ with $d \geq 1$, the complexity rate of randomized methods for Lipschitz integrands is known to be $\mathrm{O}(n^{-1/2} n^{-1/d})$ (see Novak (2016)). Furthermore, when the integrand has bounded s first derivatives, the convergence rate becomes $\mathrm{O}(n^{-1/2} n^{-s/d})$.

In Portier and Segers (2019), when using m control variates, the convergence rate is $\mathrm{O}(n^{-1/2}m^{-s/d})$ where s is the regularity of f. The associated computation of optimal control variates relies on ordinary least squares regression. To avoid ill-conditioning and for numerical stability, it requires that m should be of a smaller order than n and thus, it prevents from achieving the optimal rate. Relying on some control function constructed in a reproducing kernel Hilbert space, Oates et al. (2017) derived an acceleration compared to the naive $\sqrt{n}$-convergence rate and obtained $\mathrm{O}(n^{-7/12})$ for a specific class of functions.

Another reliable technique to improve the rate of convergence of standard Monte Carlo is stratification. This technique consists in partitioning the space and sampling over each element of the partition. It has allowed to improve the convergence rate of Monte Carlo estimates (Haber, 1966, 1967) and to derive a general framework called stochastic quadrature rules (Haber, 1969). Recently, Haber's work has been extended to take advantage of higher smoothness in the integrand (Chopin and Gerber, 2022). To the best of found knowledge, the works of Haber (1966) and Chopin and Gerber (2022) are the only ones achieving the best rate of convergence for Lipschitz function and for general regularity space. Observe that the methods in Haber (1966) and in Chopin and Gerber (2022), even though they achieve the optimal convergence rate, are only valid for integration over the unit cube. In addition they involve a geometric number (ℓ^d) of evaluations of the integrand f which is problematic in practice for applications with small computational budget as in complex bayesian models. All these remarks motivate the following research question.

Research Question #3

Relying on *control variates* techniques, how to build an efficient Monte Carlo estimate that reaches the optimal complexity rate of randomized methods for Lipschitz integrands ?

All the Monte Carlo estimates presented in this section are now summarized in the Table below. Note that the sampling process of *control variates* methods is the same as the one of standard Monte Carlo whereas importance sampling estimates heavily rely on particular sampling densities.

Method	Particles	Estimate
Standard Monte Carlo	$X_i \sim \pi$	$\hat{\alpha}_n^{(\mathrm{mc})}(f) = \frac{1}{n}\sum_i f(X_i)$
Control Variate Monte Carlo	$X_i \sim \pi$	$\hat{\alpha}_n^{(\mathrm{cv})}(f) = \frac{1}{n}\sum_i (f(X_i) - \hat{\beta}^\top Z_i)$
Importance Sampling	$X_i \sim q$	$\hat{\alpha}_n^{(\mathrm{is})}(f) = \sum_i w_i f(X_i) / \sum_i w_i$
Adaptive Importance Sampling	$X_{t,i} \sim q_{t-1}$	$\hat{\alpha}_n^{(\mathrm{ais})}(f) = \sum_{t,i} w_{t,i} f(X_{t,i}) / \sum_{t,i} w_{t,i}$

Table 1.1 – Summary of Monte Carlo estimates $\hat{\alpha}_n(f)$.

1.3 Machine Learning and Stochastic Optimization

In order to learn the optimal parameter of the objective $\mathcal{F}$ defined in Eq.(1.1), one needs to rely on powerful optimization algorithms. The success of certain optimization methods for machine learning has inspired great numbers in various research communities to design new methods that are more widely applicable. In that perspective, it is the goal of Part III to study general stochastic optimization methods (Chapter 5) and to provide new frameworks to leverage structure in data (Chapter 6).

1.3.1 Mathematical foundations

From a general perspective, the goal of machine learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. The specificity of machine learning is that the learning comes from **data**: one has access to a finite set of samples $(z_i)_{1 \leq i \leq n} \in \mathcal{Z}^n$ which are used to learn the function f. Roughly speaking, the complexity of machine learning problems is summarized through two main quantities:
• **dimension** d: the dimension of the input space $\mathcal{X}$.
• **sample size** n: the number of available data points to learn the function f.

In modern machine learning tasks, the scale is large, meaning that the number n of samples is large, and the dimension d of the data points is also large. One may think for instance at a dataset of images composed of thousands or millions of images ($n \sim 10^6$), each one of them being represented as a vector of pixels which are all RGB-encoded in dimension $d = 256 \times 256 \times 256 = 2^{24}$.

For simplicity, the focus is on problems that arise in the context of *supervised learning* (Hastie et al., 2009); *i.e.* where the data takes the form of input-output pairs $z_i = (x_i, y_i)$ and the goal is to predict an output $y = f(x) \in \mathcal{Y}$ from an observation $x \in \mathcal{X}$. For example, one may want to predict whether a patient will survive (binary classification with $\mathcal{Y} = \{-1; +1\}$) given its medical record and treatment or predict the price of an asset given customer data (regression with $\mathcal{Y} = \mathbb{R}$). The complexity of the underlying problem is encapsulated into a probability distribution π to which we have limited access through the data: this is the statistical learning framework.

In this setting, the data samples $z_1, \dots, z_n$ are assumed to be realisations of a random variable Z on $\mathcal{Z}$ with the associated probability measure π on $\mathcal{Z}$. A very standard assumption – which may not be satisfied in practice – is that the data points $z_1, \dots, z_n$ are independent and identically distributed from Z. In the case of supervised learning, the samples (x_i, y_i) are assumed to be drawn from a joint distribution $Z = (X, Y)$.

Generalization error. The goal is to find a good predictor $f : \mathcal{X} \to \mathcal{Y}$ such that $f(X)$ is a good approximation of Y. For that matter, denote by $\mathcal{M}(\mathcal{X}, \mathcal{Y})$ the set of measureable functions from $\mathcal{X}$ to $\mathcal{Y}$. The quality of the approximation of a prediction $f(x)$ compared to $y \in \mathcal{Y}$ is defined through the notion of *loss function*. A loss function is a map $\ell : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{M}(\mathcal{X}, \mathcal{Y}) \to \mathbb{R}_+$ such that $\ell((x, y), f)$ quantifies the error of approximating y by $f(x)$. The risk or generalization error of a predictor f is then

$$\forall f \in \mathcal{M}(\mathcal{X}, \mathcal{Y}), \quad \mathcal{R}(f) := \mathbb{E}_\pi[\ell((X, Y), f)], \tag{1.11}$$

where π is the joint distribution of $Z = (X, Y)$. With this quantity of risk, one is interested in finding the optimal predictor $f^\star \in \mathcal{M}(\mathcal{X}, \mathcal{Y})$ which gives the smallest possible risk

$$\mathcal{R}(f^\star) = \inf_{f \in \mathcal{M}(\mathcal{X}, \mathcal{Y})} \mathcal{R}(f). \tag{1.12}$$

Losses. The choice of the loss ℓ in Eq.(1.11) defines what one considers to be a good inference from data and depends on the problem we are considering. Two classical supervised learning problems and their associated losses are now highlighted. Both are considered in numerical experiments of Part III:

(i) Regression and square loss. In this case, the output Y takes value in $\mathcal{Y} = \mathbb{R}$ or more generally a subset of $\mathbb{R}^k$ and the loss is $\ell_{x,y}(f) = \|y - f(x)\|^2$. This is the most standard loss for regression and solving Eq.(1.12) is called the least squares regression problem. Observe that as soon as $Y \in L_2$ there exists a solution given by the projection of Y onto $L_2(\mathcal{X}, \pi_{\mathcal{X}})$ seen as a closed linear subspace of $L_2(\mathcal{X} \times \mathcal{Y}, \pi)$, also called the conditional expectation of Y given X, *i.e.*, $f^\star(X) = \mathbb{E}[Y|X]$.

(ii) Binary classification. In this case, Y takes value in $\mathcal{Y} = \{-1, +1\}$ and the natural loss is the 0/1 penalization $\ell_{x,y}(f) = \mathbb{1}_{f(x)\neq y} = \mathbb{1}_{yf(x)<0}$. However this loss is neither smooth nor convex and can be hard to optimize. Instead, it is easier to allow f to be real-valued and predict the output according to the sign of $f(x)$. If f is linearly parameterized, the predictor f is called a separating hyperplane as $\{f(x) = 0\}$ defines the boundary between the two classes. More generally, classical losses are of the form $\varphi(yf(x))$ where φ is a surrogate for $\mathbb{1}_{u<0}$. This include the Hinge loss defined by $\varphi(u) = \max\{1-u; 0\}$ which is convex but not smooth and the logistic loss $\varphi(u) = \log(1+e^{-u})$ which is smooth and convex.

Empirical Risk. The expression of the expected risk in Eq.(1.11) relies on an expectation which is in general analytically intractable. Thus, in practice, one seeks the solution of a problem that involves an estimate $\hat{\mathcal{R}}$ of the true risk $\mathcal{R}$. One has access to the distribution of Z only through the samples $z_1, \ldots, z_n$. Therefore, one may replace the data distribution π by its empirical counterpart $\hat{\pi} := (1/n)\sum_{i=1}^n \delta_{z_i}$ and define the so-called *empirical risk* as

$$\hat{\mathcal{R}}_n(f) := \mathbb{E}_{\hat{\pi}}[\ell_Z(f)] = \frac{1}{n}\sum_{i=1}^n \ell_{z_i}(f). \tag{1.13}$$

Starting from the ideal problem of expected risk minimization (1.12), we have come to the empirical risk problem (1.13) which is still to be solved. This is the role of optimization algorithms. We adopt the standard notation from optimization where f is no longer the predictor but denotes the objective function to optimize and d is the dimension of the set on which we perform the optimization procedures.

1.3.2 (Stochastic) Optimization Methods

From a general standpoint, an optimization algorithm aims at solving $\inf_{\theta\in\Theta} f(\theta)$ where $\Theta \subset \mathbb{R}^d$. In view of the empirical risk minimization paradigm of Eq.(1.13), the objective function usually takes the form of a finite-sum $f(\theta) = \sum_{i=1}^n f_i(\theta)$ where the cost in time and memory of computing a gradient of f_i is $O(d)$. Optimization algorithms may vary according to the way we access the function f, the structure, the regularity and convexity properties, the time and space complexity and the means of computations. We first present (stochastic) first-order methods and their adaptive variants which are the optimization workhorse in machine learning. Then we discuss the use of second order methods in machine learning.

First-order methods. The most well-known method in optimization is *gradient descent* (GD). Starting from $\theta_0 \in \Theta$, the idea is simply, at each iteration t, to evaluate the gradient $\nabla f(\theta_t)$ and to go in the direction $-\gamma_{t+1}\nabla f(\theta_t)$ with a stepsize $\gamma_t > 0$. This is illustrated in Figure 1.3 below which depicts the trajectory followed by a gradient-based optimization algorithm. The update rule is

$$\theta_{t+1} = \theta_t - \gamma_{t+1}\nabla f(\theta_t). \tag{1.14}$$

The cost of each iteration here is a priori $O(nd)$ in time and $O(d)$ in memory. Since in large scale learning both n and d may be large, the computation of the full gradient may be prohibitive. Whereas traditional gradient-based methods may be effective for solving small-scale learning problems in which a batch approach may be used, in the context of large-scale machine learning it has been a stochastic algorithm—namely, the *stochastic gradient descent* (SGD) method proposed by Robbins and Monro (1951) — that has been the core strategy of interest. This algorithm uses a single stochastic gradient at each iteration and is defined by the update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1}\nabla f_j(\theta_t), \tag{1.15}$$

where j is selected uniformly at random in $\{1, \ldots, n\}$ at each iteration. The cost of an iteration is thus reduced to only $O(d)$ compared to previous $O(nd)$ since we access only one gradient of the f_i. Another stochastic approach, referred to as mini-batching (Gower et al., 2019), consists in generating uniformly a set of k indices $B = \{i_1, \ldots, i_k\}$ and computing the gradient as the average over this batch $\sum_{i\in B} \nabla f_i(\theta_t)/|B|$.

In view of performing noise reduction, one may consider *gradient aggregation* by storing gradient estimates corresponding to samples employed in previous iterations, updating one (or some) of these estimates in each iteration, and defining the search direction as a weighted average of these estimates. Other noise reduction methods include *dynamic sampling* which gradually increases the minibatch size used in the gradient computation, and *iterate averaging* which maintains an average not of the gradient but of the iterates computed during the optimization process.

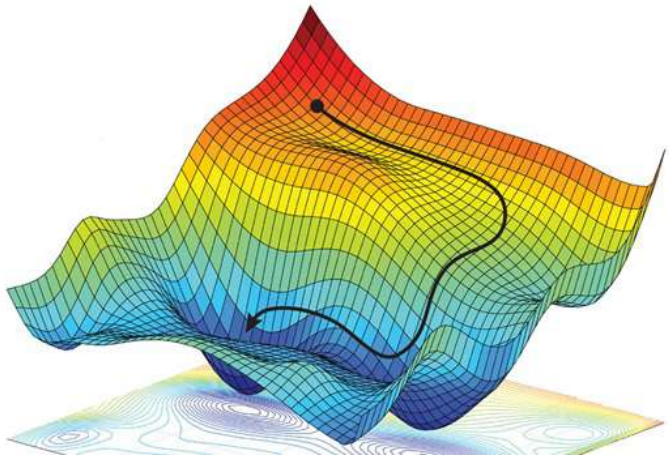


Figure 1.3 – A visualization of the 'route' followed by a gradient optimization algorithm across a loss surface as it is trained (Amini et al., 2018). At each iteration t, one evaluates $\nabla f(\theta_t)$ and follows the direction $-\gamma_{t+1}\nabla f(\theta_t)$ with a stepsize $\gamma_t > 0$.

The paper by Robbins and Monro represents a landmark in the history of numerical optimization methods. Together with the invention of back propagation (Rumelhart et al., 1986), it also represents one of the most notable developments in the field of machine learning. Although widely used in practice, the standard SGD algorithm has at least two limitations:

(i) The choice of the learning rate is generally difficult; large learning rates result in large fluctuations of the estimate, whereas small learning rates induce slow convergence. *(ii)* A common learning rate is used for every coordinate despite the possible discrepancies in the values of the gradient vector's coordinates.

To address these limitations, one may look at *second-order methods* which make use of the information brought in by the curvature of the objective function f and can solve ill-conditioning. Furtheremore, the learning rate sequence may be replaced by *diagonal rescaling methods* which adjust the learning rate coordinate-wise, as functions of the past values of the gradient evaluations.

Second-order methods. The canonical second order algorithm is the Newton method and follows the update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1}\Delta_t, \quad \Delta_t = \nabla^2 f(\theta_t)^{-1}\nabla f(\theta_t), \tag{1.16}$$

where $\gamma_t > 0$ is still the learning rate and Δ_t is a renormalized gradient step called Newton step. Intuitively, when the gradient g is stable, $g^\top \nabla^2 f(\theta) g$ is small and the newton method renormalizes the direction as if multiplying by large stepsize. On the contrary, in directions where the gradient changes quickly, the Newton method renormalizes the direction as if multiplying by a small stepsize. Computing one Newton step is computationally expensive : computing the Hessian at a given point usually takes time of order $O(nd^2)$, and computing its inverse takes time $O(d^3)$. This is prohibitive both in terms of storage capacity and of time complexity. To overcome this issue, there exists a wide variety of stochastic second-order methods that are Hessian-free or that attempt to mimic the behavior of a Newton algorithm through first-order information computed over sequences of iterates. These include natural gradient, Quasi-Newton, Gauss-Newton, Hessian-free Newton and related algorithms that employ only diagonal rescalings. A schematic overview of all the mentioned optimization methods is presented in Figure 1.4 and further details can be found in Bottou et al. (2018).

Diagonal rescaling methods. This modification can be seen as a diagonal preconditioning of the stochastic gradient in SGD based on past observed gradients. The independent works of Duchi et al. (2011) and McMahan and Streeter (2010) in the context of online convex optimization led the way to a new class of algorithms that are referred to as *adaptive gradient methods*. As proposed by Duchi et al. (2011), AdaGrad consists of dividing the learning rate by the square root of the sum of previous gradients squared componentwise. The idea is to give larger learning rates to highly informative but infrequent features instead of using a fixed predetermined schedule. This is particularly relevant in applications such as click through rate prediction for online advertising and text classification where many features only occur rarely with only a few number of non-zero features while few occur very often.

Both (stochastic) second-order methods and adaptive methods can be written in a general form as *Conditioned* SGD, which consists in multiplying the gradient estimate by some random conditioning matrix C_t at each iteration leading to the update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1} C_t \nabla f_j(\theta_t), \quad C_t \in \mathbb{R}^{d\times d}. \tag{1.17}$$

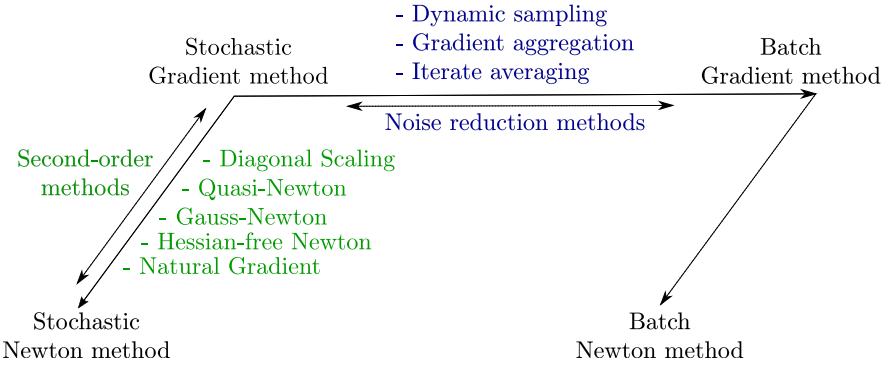


Figure 1.4 – Schematic of a two-dimensional spectrum of optimization methods for machine learning. The horizontal axis represents methods designed to control stochastic noise; the second axis, methods that deal with ill conditioning (Bottou et al., 2018).

Observe that randomness is introduced by both the gradient estimate and the conditioning matrix. This general framework can lead to better performance as shown in several recent studies ranging from natural gradient (Amari, 1998; Kakade, 2002) and stochastic second-order methods with quasi-Newton (Byrd et al., 2016) and (L)-BFGS methods (Liu and Nocedal, 1989) to diagonal scalings and adaptive methods such as AdaGrad (Duchi et al., 2011), RMSProp (Tieleman et al., 2012), Adam (Kingma and Ba, 2014) and AMSGrad (Reddi et al., 2018).

Interestingly, the optimal choice according to the asymptotic variance is the inverse of the Hessian matrix at optimal point, i.e., $C_k = \nabla^2 f(\theta^\star)^{-1}$; see (Benveniste et al., 2012, Chapter 3) or Section 5.2.3 in Chapter 5. With this matrix, the rate of convergence remains the same and only the asymptotic variance can be reduced; e.g., Agarwal et al. (2009). Important questions which are still open to the best of found knowledge, are the following.

Research Question #4

What is the asymptotic behavior of general *Conditioned*-SGD methods and can the optimal variance be achieved by such an algorithm for non-convex f ?

Coordinate Descent methods. The idea of coordinate descent is to decompose a large optimization problem into a sequence of one-dimensional optimization problems. The algorithm was first described for the minimization of quadratic functions by Gauss and Seidel in Seidel (1873). At each iteration, the algorithm determines a coordinate or coordinate block via a coordinate selection rule, then exactly or inexactly minimizes over the corresponding coordinate hyperplane while fixing all other coordinates or coordinate blocks.

Coordinate Descent (CD) algorithms have become unavoidable in modern machine learning because they are tractable (Nesterov, 2012) and competitive to other methods when dealing with key problems such as support vector machines, logistic regression, LASSO regression and other ℓ_1-regularized learning problems (Wu et al., 2008; Friedman et al., 2010). Moreover, the decomposition into small subproblems means that only a small part of the data is processed at each iteration and this makes coordinate descent easily scalable to high dimensions. Starting from the *conditioned* SGD update rule in Eq.(1.17), one may look at particular instances of the conditioning matrix C_t

and restrict the study to sparse diagonal matrices. Such a choice will produce a coordinatewise version of the standard SGD algorithm and allows to select the coordinate of the gradient estimates in an adaptive manner.

On the one hand, efficient forms of CD methods rely on a deterministic procedure (Nutini et al., 2015) which adapts to the underlying structure in data at the expense of higher calculation and thus, may be costly. On the other hand, stochastic gradient descent (SGD) methods are computationally efficient but often treat all coordinates equally and thus, may be sub-optimal. In the spirit of adaptive schemes and by combining the best of both worlds, we are interested in the following research question.

Research Question #5

Can we derive, within a noisy gradient framework, a general stochastic coordinate descent method with a particular selection strategy ?

1.4 Summary of Contributions

Motivated by the different research questions (RQ) mentioned in the previous sections, we now provide a detailed overview of the contributions of this thesis where each chapter is dedicated to one of the research direction.

Part II: Monte Carlo methods and Variance Reduction

• Chapter 2: *Control Variate Selection for Monte Carlo Integration*(*RQ*#1)

To deal with the computational issues of using a large number of *control variates*, it has been proposed in South et al. (2022) to regularize the OLS estimate by adding a ℓ_1-penalty term in the minimization problem, just as in the LASSO (Tibshirani, 1996). Simulation results in South et al. (2022) show that this approach, referred to as LASSO, provides great improvements in practice. However, those practical findings are not supported by an asymptotic error rate nor by a non-asymptotic error bound. The main objective of this chapter is to provide a non-asymptotic theory for the use of control variates in Monte Carlo simulations.

Contributions. The main contributions are as follows.

(1) A *new method* called LSLASSO is proposed. In the spirit of the procedure of Belloni and Chernozhukov (2013), it consists in selecting the best control variates via the LASSO, using subsampling to decrease the computation time, and then to apply OLS with the selected controls.

(2) *Support recovery*: the LASSO procedure is shown to select the correct control variates with large probability.

(3) *Concentration inequalities* are derived for the OLS, LASSO and LSLASSO integration errors. The one for the OLS highlights a compromise between the approximation error of the integrand in the linear span of control variates and the multicollinearities between the control variates. The ones for (LS)LASSO show significant improvements regarding the effects of multicollinearity.

The approach for the proofs combines well known sub-Gaussian concentration inequalities (Boucheron et al., 2013a) along with a lower bound for the smallest eigenvalue of an empirical Gram matrix, based on a Chernoff inequality for matrices (Tropp, 2015, Theorem 5.1.1).

• Chapter 3: *A Quadrature Rule combining Control Variates and Adaptive Importance Sampling*(*RQ*#2)

The use of control variates is a well studied variance-reduction technique (Glynn and Szechtman, 2002; Owen and Zhou, 2000). The benefits can be established theoretically in terms of error bounds (see Oates et al. (2017) and chapter 2), weak convergence (Portier and Segers, 2019), the excess risk (Belomestny et al., 2022) and even uniform error bounds over large classes of integrands (Plassier et al., 2020). Importance sampling and control variates in case of a Gaussian target density is explored in Jourdain (2009). Recently, the procedure in Kawai (2020) incorporates control variates and is said to involve adaptive importance sampling, but in fact the particles are always sampled from the uniform distribution on the unit cube. To the best of found knowledge, the existing control variate methods do not account for sequential changes in the particle distribution as is the case in AIS. The main goal of this chapter is to develop a framework to combine control variates and adaptive importance sampling.

Contributions. The contributions may be summarized as follows:

(1) A simple weighted least squares approach is proposed to improve the procedure of sequential algorithms with control variates. The proposed AISCV estimate significantly improves the accuracy of the initial algorithm, both theoretically and in practice.

(2) Several theoretical properties of the AISCV estimate are provided. In particular, we derive a probabilistic, non-asymptotic bound on the integration error.

(3) Practical considerations and implementations of the control variates are presented along with convincing numerical experiments.

The proposed approach to use control variates within the sequential AIS framework relies on the ordinary least squares expression of control variates (see for instance Portier and Segers (2019)). To take care of the policy changes, some re-weighting must be applied. The AISCV estimate of the integral $\int f\pi \,\mathrm{d}\lambda$ is defined as the first coordinate of the solution to the weighted least squares problem

$$(\hat{\alpha}_n, \hat{\beta}_n) = \underset{a\in\mathbb{R}, b\in\mathbb{R}^m}{\arg\min} \sum_{i=1}^{n} w_i \left(f(X_i) - a - b^\top h(X_i) \right)^2,$$

with w_i the importance weights from before. The AISCV estimate has several interesting properties:

(a) Whenever g is of the form $\alpha + \beta^\top h$ for some $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, the error is zero, i.e., $\hat{\alpha}_n = \alpha = \int f\pi \,\mathrm{d}\lambda$.

(b) The estimate takes the form of a quadrature rule $\hat{\alpha}_n = \sum_{i=1}^n v_{n,i} f(X_i)$, for quadrature weights $v_{n,i}$ that do not depend on the function f and that can be computed by a single weighted least squares procedure.

(c) It can be computed even when π is known only up to a multiplicative constant.

Point (a) suggests that when the linear combinations of the functions h_k span a rich function class, the integration error is likely to be small. Point (b) implies that multiple integrals can be computed just as easily as a single one. Point (c) shows that the approach is applicable for Bayesian computations. In addition, the control variates can be brought into play in a *post-hoc* scheme, after generation of the particles and importance weights, and this for any AIS algorithm.

The main theoretical result of the chapter is a probabilistic, non-asymptotic bound on $\hat{\alpha}_n - \alpha$. Under appropriate conditions, the bound scales as $\tau/\sqrt{n}$, where τ^2 is the scale constant in a sub-Gaussian tail condition on the error variable $\varepsilon = f - \alpha - \beta^\top h$ for $(\alpha, \beta) = \arg\min_{a,b} \int (f - a - b^\top h)^2 \pi \, \mathrm{d}\lambda$. Note that ε has the smallest possible variance one could get using control variates h. As a consequence, when the space of control variates is well suited for approximating g, the AISCV estimate will be highly accurate. Also, our bound depends only on the linear function space spanned by the control variates $h_1, \ldots, h_m$, not on the particular basis chosen in that space.

The results rely on martingale theory, in particular on a concentration inequality for norm-subGaussian martingales in Jin et al. (2019). In the course of the proof, we develop a novel bound on the smallest eigenvalue of certain random matrices, extending an inequality from (Tropp, 2015) to the martingale case.

• Chapter 4: *Speeding up Monte Carlo Integration: Nearest Neighbors Estimates as Control Variates*(*RQ*#3)

This chapter deals with the use of *control variates* from complexity rates point of view. As mentioned in section 1.2, the methods in Haber (1966) and in Chopin and Gerber (2022), even though they achieve the optimal convergence rate, are only valid for integration over the unit cube. In addition they involve a geometric number (k^d) of evaluations of the integrand f which is problematic in practice for applications with small computational budget as in complex bayesian models. Interestingly, as mentioned in Chopin and Gerber (2022), their stratification method is related to a specific control variates construction relying on a piecewise constant control function which has a very low bias compared to traditional regression estimate.

This precise idea of using an estimate with small bias is the starting point of this chapter. It is relevant to the considered framework because the function f is accessible without noise. Note that this kind of estimates – with small bias – has also been successfully used in the related topic of adaptive rejection sampling (Achddou et al., 2019) allowing to reach optimal rate. The main goal of this chapter is to develop the framework of *control neighbors* which use nearest neighbors as control variates to achieve optimal convergence rate for the integration error.

Contributions. The contributions may be summarized as follows:

(1) A new Monte Carlo method called *control neighbors* is introduced. This method constructs an estimate $\hat{\alpha}_n(f)$ to approximate the integral $\pi(f)$ for general probability measure π and the core idea follows from using 1-Nearest Neighbor estimates as control variates.

(2) This estimate is shown to achieve the optimal convergence rate in $\mathrm{O}(n^{-1/2}n^{-1/d})$ for Lipschitz functions. To the best of found knowledge, obtaining the optimal convergence rate for general probability measure makes this method the first of its kind.

(3) Several practical considerations of the control neighbors are presented along with promising numerical experiments.

The most remarkable properties of the *control neighbors* estimate are:

(a) The *control neighbors* estimate can be obtained under the same framework as standard Monte Carlo, *i.e.*, as soon as one can both *(i)* draw random particles from π and *(ii)* evaluate the integrand f. Contrary to the classical control variates framework (Portier and Segers, 2019), the proposed estimate does not require the existence of control variates with known integrals.

(b) *control neighbors* takes the form of a linear integration rule $\sum_{i=1}^{n} w_{i,n} f(X_i)$ where the weights $w_{i,n}$ do not depend on the integrand f but only on the sampled particles $X_1, \ldots, X_n$. This key property allows computational benefits when several integrals are to be computed with respect to the same measure π.

(c) The convergence rate is shown to be optimal for Lipschitz functions, *i.e.*, the integration error decreases as $\mathrm{O}(n^{-1/2} n^{-1/d})$ whenever f is Lipschitz (Novak, 2016). Other approaches (for general measure μ) that have been developed recently, e.g., (Oates et al., 2017; Portier and Segers, 2019) do not achieve this rate.

(d) Since the weights $w_{n,i}$ are built using nearest neighbor estimates, complete practical tools are already available, including effective nearest neighbor search with k-dimensional tree (Bentley, 1975) and efficient compression and parallelization (Pedregosa et al., 2011; Johnson et al., 2019).

(e) The proposed approach is *post-hoc* in the sense that it can be run after sampling the particles and independently from the sampling mechanism. In particular, it can be implemented for other sampling design including MCMC or AIS.

Part III: Stochastic Approximation: Conditioning & Adaptive Sampling

• Chapter 5: *Asymptotic Analysis of Conditioned SGD* (*RQ#4*)

In light of research question (*RQ#4*), this chapter concerns optimization problems of the following form: $\min_{\theta \in \mathbb{R}^d} \{f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)]\}$, where f is a loss function and ξ is a random variable. *Conditioned SGD* generalizes *standard SGD* by adding a conditioning step to refine the descent direction. Starting from $\theta_0 \in \mathbb{R}^d$, the algorithm of interest is defined by the following iteration

$$\theta_{t+1} = \theta_t - \gamma_{t+1} C_t g(\theta_t, \xi_{t+1}), \qquad t \geq 0,$$

where $g(\theta_t, \xi_{t+1})$ is some unbiased gradient valued in $\mathbb{R}^d$, $C_t \in \mathbb{R}^{d \times d}$ is called *conditioning matrix* and $(\gamma_t)_{t \geq 1}$ is a decreasing learning rate sequence.

Related work. Seminal works around standard SGD ($C_t = I_d$) were initiated by Robbins and Monro (1951) and Kiefer et al. (1952). Since then, a large literature known as *stochastic approximation*, has developed. The almost sure convergence is studied in Robbins and Siegmund (1971) and Bertsekas and Tsitsiklis (2000); rates of

convergence are investigated in Kushner and Huang (1979) and Pelletier (1998a); non-asymptotic bounds are given in Moulines and Bach (2011). The asymptotic normality can be obtained using two different approaches: a diffusion-based method is employed in Pelletier (1998b) and Benaïm (1999) whereas martingale tools are used in Sacks (1958) and Kushner and Clark (1978). We refer to Nevelson and Khas'minskiĭ (1976); Delyon (1996); Benveniste et al. (2012); Duflo (2013) for general textbooks on *stochastic approximation*.

The aforementioned results do not apply directly to *conditioned* SGD because of the presence of the matrix sequence $(C_t)_{t\geq 0}$ involving an additional source of randomness in the algorithm. Seminal papers dealing with the weak convergence of *conditioned* SGD are Venter (1967) and Fabian (1968). Within a restrictive framework (univariate case $d = 1$ and strong assumptions on the function f), their results are encouraging because the limiting variance of the procedure is shown to be smaller than the limiting variance of standard SGD. Venter's and Fabian's results have then been extended to more general situations (Fabian, 1973; Nevelson and Khas'minskiĭ, 1976; Wei, 1987). In Wei (1987), the framework is still restrictive not only because the random errors are assumed to be independent and identically distributed but also because the objective f must satisfy their assumption (4.10) which hardly extends to objectives other than quadratic.

More recently, Bercu et al. (2020) have obtained the asymptotic normality as well as the efficiency of certain *conditioned* SGD estimates in the particular case of *logistic regression*. The previous approach has been generalized not long ago in Boyer and Godichon-Baggioni (2020) where the use of the Woodbury matrix identity is promoted to compute the Hessian inverse in the online setting. Several theoretical results, including the weak convergence of *conditioned* SGD, are obtained for convex objective functions. The main objective of this chapter is to derive an asymptotic theory for *conditioned* SGD for general non-convex objectives.

Contributions. The main results of this chapter are as follows:

(1) A high-level result dealing with the weak convergence of the rescaled sequence of iterates $(\theta_t - \theta^\star)/\sqrt{\gamma_t}$ is provided for general *conditioned* SGD methods.

(2) Another result of independent interest dealing with the almost sure convergence of the gradients $\nabla f(\theta_t) \to 0$ is also presented.

(3) For the sake of completeness, we present practical ways to compute the *conditioning* matrix C_t and show that the resulting procedure satisfies the high-level conditions of our main Theorem. This yields a feasible algorithm which achieves minimum variance.

Interestingly, our asymptotic normality result consists of the following continuity property: whenever the matrix sequence $(C_t)_{t\geq 0}$ converges to a matrix C and the iterates $(\theta_t)_{t\geq 0}$ converges to a minimizer $\theta^\star$, the algorithm behaves in the same way as an oracle version in which C would be used instead of C_t. We stress that contrary to Boyer and Godichon-Baggioni (2020), no convexity assumption is needed on the objective function and no rate of convergence is required on the sequence $(C_t)_{t\geq 0}$. This is important because, in most cases, deriving a convergence rate on $(C_t)_{t\geq 0}$ requires a specific convergence rate on the iterates $(\theta_t)_{t\geq 0}$ which, in general, is unknown at this stage of the analysis.

To obtain these results, instead of approximating the rescaled sequence of iterates by a continuous diffusion (as for instance in Pelletier (1998b)), we rely on a discrete-time approach where the recursion scheme is directly analyzed (as for instance in Delyon (1996)). More precisely, the sequence of iterates is studied with the help of an auxiliary linear algorithm whose limiting distribution can be deduced from the central limit theorem for martingale increments (Hall and Heyde, 1980). The limiting variance is derived from a discrete time matrix-valued dynamical system algorithm. It corresponds to the solution of a Lyapunov equation involving the matrix C. It allows a special choice for C which guarantees an optimal variance. Finally, in order to examine the remaining part, a particular recursion is identified. By studying it on a particular event, we show that this remaining part is negligible.

• Chapter 6: *SGD with Coordinate Sampling: Theory and Practice*(*RQ*#5)

Recall that the SGD algorithm is defined by the update rule

$$\forall t \geq 0, \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

where $g_t \in \mathbb{R}^d$ is a gradient estimate at θ_t (possibly biased) and $(\gamma_t)_{t\geq 1}$ is some learning rate sequence that should decrease throughout the algorithm. While the computation of g_t may be cheap, it still requires the computation of a vector of size d which may be a critical issue in high-dimensional problems. To address this difficulty, we rely on sampling well-chosen coordinates of the gradient estimate at each iteration.

In this chapter, we develop the framework of stochastic coordinate gradient descent (SCGD) which modifies standard stochastic gradient descent methods by adding a selection step to perform random coordinate descent. The SCGD algorithm is defined by the following iteration

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

where ζ_{t+1} is a random variable valued in $[\![1, d]\!]$ which selects a coordinate of the gradient estimate. The distribution of ζ_t is called the *coordinate sampling policy*. Note that the SCGD framework is very general as it contains as many methods as there are ways to generate both the gradient estimate g_t and the random variables ζ_t.

Related work. The authors of (Nutini et al., 2015) investigate the deterministic Gauss-Southwell rule which consists of picking the coordinate with maximum gradient value. In trusting large gradients, this rule looks like the one of our proposed algorithm MUSKETEER except that no stochastic noise -neither in the gradient evaluation nor in the coordinate selection- is present in their algorithm. In that aspect, our method differs from all the previous CD studies (Loshchilov et al., 2011; Richtárik and Takáč, 2016a; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017) which rely on ∇f. Among the SGD literature, compression and sparsification methods (Alistarh et al., 2017; Wangni et al., 2018) were developed for communication efficiency. The former use compression operators to select a few components of the gradient estimates at the cost of full gradient computation and coordinate sorting. The latter use a gradient estimate g which is sparsified using probability weights to reach an unbiased estimate of the gradient. In contrast, the SCGD framework allows the gradient to be biased as no importance re-weighting is performed. Note also that,

to cover zeroth-order methods, the gradient estimate itself g_t is allowed to be biased as for instance in the recent study of Ajalloeian and Stich (2020).

The objective of this chapter is twofold: from a theoretical point of view, the goal is to develop and study a general framework to enable coordinate sampling within the SGD framework; from a practical standpoint, the aim is to provide an efficient algorithm to perform stochastic optimization.

Contributions. The contributions are as follows:

(1) *(Theory)* We show the almost-sure convergence of the SCGD iterates $(\theta_t)_{t\in\mathbb{N}}$ towards stationary points in the sense that $\nabla f(\theta_t) \to 0$ almost surely as well as non-asymptotic bounds on the optimality gap $\mathbb{E}[f(\theta_t) - f^\star]$ where $f^\star$ is a lower bound of f. The working conditions are relatively weak as the function f is only required to be L-smooth (classical in non-convex problems) and the stochastic gradients are possibly biased with unbounded variance, using a growth condition related to *expected smoothness* (Gower et al., 2019).

(2) *(Practice)* We develop a new algorithm, called MUSKETEER, for *MUltivariate Stochastic Knowledge Extraction Through Exploration Exploitation Reinforcement*. In the image of the motto 'all for one and one for all', this procedure belongs to the SCGD framework with a particular design for the *coordinate sampling policy*. It compares the value of all past gradient estimates g_t to select a descent direction (*all for one*) and then moves the current iterate according to the chosen direction (*one for all*). The heuristic is the one of reinforcement learning in the sense that large gradient coordinates represent large decrease of the objective and can be seen as high rewards. The resulting directions should be favored compared to the path associated to small gradient coordinates. By updating the *coordinate sampling policy*, the algorithm is able to detect when a direction becomes rewarding and when another one stops being engaging.

The proofs of the asymptotic convergence results are based on ideas from Bertsekas and Tsitsiklis (2000) with particular extensions in the framework of biased gradient estimates. Finally, the non-asymptotic bounds are inspired from Moulines and Bach (2011) where the authors provide a non-asymptotic analysis for standard SGD.

Part II

Monte Carlo Methods & Variance Reduction

"Mathematics has a threefold purpose. They must provide an instrument for the study of nature. But that is not all: they have a philosophical purpose and, I dare say, an aesthetic purpose."

(**Henri Poincaré**, La valeur de la Science, 1908)

Chapter 2

Control Variate Selection for Monte Carlo Integration

Contents

Monte Carlo integration with variance reduction by means of control variates can be implemented by the ordinary least squares estimator for the intercept in a multiple linear regression model with the integrand as response and the control variates as covariates. Even without special knowledge on the integrand, significant efficiency gains can be obtained if the control variate space is sufficiently large. Incorporating a large number of control variates in the ordinary least squares procedure may however result in (i) a certain instability of the ordinary least squares estimator and (ii) a possibly prohibitive computation time. Regularizing the ordinary least squares estimator by preselecting appropriate control variates via the Lasso turns out to increase the accuracy without additional computational cost. The findings in the numerical experiment are confirmed by concentration inequalities for the integration error.

2.1 Introduction

Whereas the basic Monte Carlo (MC) estimate of an integral or expectation is given by $(1/n)\sum_i f_i$, for independent and identically distributed random variables f_i, the control variates method is based on $(1/n)\sum_i(f_i + h_i)$, where the variables h_i, called control variates, are constructed to have zero expectation. When the controls h_i have been selected or estimated properly (based on the samples f_i), the use of control variates might reduce the variance of the basic MC estimate significantly. The method of control variates, already used frequently to compute prices of financial derivatives (Glasserman, 2004; Gobet and Labart, 2010), has been employed recently in many different fields of Machine Learning and Statistics. Examples include (i) *reinforcement learning* and more particularly *policy gradient* methods (Jie and Abbeel, 2010; Liu et al., 2018) where the score function permits to define many control variates; (ii) inference in complex probabilistic models (Ranganath et al., 2014) where the Stein method allows to define accurate control variates (see e.g., (Oates et al., 2017; Brosse et al., 2018; Belomestny et al., 2020) and the references therein); (iii) gradient based *optimization* (Wang et al., 2013; Gower

et al., 2018), (iv) *time series analysis* when approximating the characteristic function (Davis et al., 2021), and (v) semi-supervised inference (Zhang et al., 2019).

Suppose that $m \geq 1$ control variates are available and $n \geq 1$ samples have been generated. Any linear combination of control variates can be used as a particular control variate. In terms of the variance of the estimation error, the optimal linear combination can be estimated based on the empirical risk minimization principle applied to an ordinary least squares (OLS) regression problem [see Eq. (2.3) below]. This approach, referred to as OLS, is the most common implementation of the control variates method as detailed for instance in (Owen, 2013, Section 8.3) or (Portier and Segers, 2019; South et al., 2022), although other implementations are possible, see Remark 2.2 below.

Asymptotically, the OLS error is bounded by the MC error and is proportional to the L_2 approximation error of the integrand in the linear span of control variates (Glynn and Szechtman, 2002). In combination with well-known approximation results in L_p-spaces (Rudin, 2006), this representation of the OLS error suggests to use an increasing number of control variates. Indeed, in Portier and Segers (2019) it is shown that when m grows with n, the OLS error rate can be faster than $1/\sqrt{n}$.

However, when based on a large number of control variates, the OLS suffers from two classical problems common for least squares methods: (i) numerical instabilities when the control variates are nearly collinear, and (ii) a computational complexity in $m^3 + nm^2$, which might be prohibitive.

To deal with these two issues, it has been proposed in South et al. (2022) to regularize the OLS estimate by adding a ℓ_1-penalty term in the minimization problem, just as in the LASSO (Tibshirani, 1996). Simulation results in South et al. (2022) show that this approach, referred to as LASSO, provides great improvements in practice. However, those practical findings are not supported by an asymptotic error rate nor by a non-asymptotic error bound.

The main objective of the chapter is to provide a non-asymptotic theory for the use of control variates in Monte Carlo simulations. The contributions are as follows.

1. A *new method* called LSLASSO is proposed. In the spirit of (Belloni and Chernozhukov, 2013), it consists in selecting the best control variates via the LASSO, using subsampling to decrease the computation time, and then to apply OLS with the selected controls.

2. *Support recovery*: the LASSO is shown to select the correct control variates with large probability.

3. *Concentration inequalities* are derived for the OLS, LASSO and LSLASSO integration errors. The one for the OLS highlights a compromise between the approximation error of the integrand in the linear span of control variates and the multicollinearities between the control variates. The ones for (LS)LASSO show significant improvements regarding the effects of multicollinearity.

The approach for the proofs combines well known sub-Gaussian concentration inequalities (Boucheron et al., 2013a) along with a lower bound for the smallest eigenvalue of an empirical Gram matrix, based on a Chernoff inequality for matrices (Tropp, 2015, Theorem 5.1.1).

The outline of the chapter is as follows. Section 2.2 introduces the theoretical background and the different MC estimates and provides some comments about their practical implementation and some possible alternative approaches. Section 2.3 contains the statements of the theoretical results. Sections 2.4 and 2.5 describe numerical experiments on artificial and real data to illustrate the finite-sample behavior of the methods. Section 2.6 concludes the main part of the chapter with a discussion of avenues for further research. Section 2.A.1 contains some auxiliary results, whereas the proofs of the four theorems stated in Section 2.3 are given in Sections 2.A.2 to 2.A.5.

2.2 Monte Carlo integration and control variates

Background. Let $f \in L_2(\pi)$ be a square integrable, real-valued function on a probability space $(\mathcal{X}, \mathcal{A}, \pi)$ of which we would like to calculate the integral

$$\pi(f) = \int_{\mathcal{X}} f(x)\, \pi(\mathrm{d}x).$$

The MC estimator of $\pi(f)$ based on independent random variables $X_1, \ldots, X_n$ taking values in $\mathcal{X}$ and with common distribution π is

$$\hat{\alpha}_n^{\mathrm{mc}}(f) = \pi_n(f) = \frac{1}{n} \sum_{i=1}^{n} f(X_i).$$

This estimator is unbiased and has variance $n^{-1}\sigma_0^2(f)$, where $\sigma_0^2(f) = \pi[(f - \pi(f))^2]$.

The control variates are functions $h_1, \ldots, h_m \in L_2(\pi)$ with known expectations. Without loss of generality, assume that $\pi(h_k) = 0$ for all $k \in \{1, \ldots, m\}$. Let $h = (h_1, \ldots, h_m)^\top$ denote the $\mathbb{R}^m$-valued function with the m control variates as elements. Let $\mathcal{F}_m = \mathrm{Span}\{h_1, \ldots, h_m\} = \{\beta^\top h : \beta \in \mathbb{R}^m\}$ denote the closed linear subspace of $L_2(\pi)$ generated by the control variates.

For any coefficient vector $\beta = (\beta_1, \ldots, \beta_m)^\top \in \mathbb{R}^m$, we have $\pi(f - \beta^\top h) = \pi(f)$, so that $\pi_n(f - \beta^\top h)$ is an unbiased estimator of $\pi(f)$, with variance $n^{-1}\pi[(f - \pi(f) - \beta^\top h)^2]$. Any oracle coefficient

$$\beta^\star(f) \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^m} \pi[(f - \pi(f) - \beta^\top h)^2]$$

minimizes the variance. If such a $\beta^\star(f)$ would be known, the resulting oracle estimator would be

$$\hat{\alpha}_n^{\mathrm{or}}(f) = \pi_n[f - \beta^\star(f)^\top h]. \tag{2.1}$$

By definition, the oracle estimator achieves the minimal variance $n^{-1}\sigma_m^2(f)$ where $\sigma_m^2(f)$ is the minimum value of the variance term $\pi[(f - \pi(f) - \beta^\top h)^2]$ with respect to β. For any $m' \in \{0, 1, \ldots, m\}$, if we use only the first m' control variates $h_1, \ldots, h_{m'}$, or even none at all in case $m' = 0$, we have $\sigma_m^2(f) \leq \sigma_{m'}^2(f)$. In particular, if $\beta^\star(f)$ would be known, the use of control variates would always reduce the variance of the basic Monte Carlo estimator.

As $\beta^\star(f)^\top h$ is the $L_2(\pi)$-projection of $f - \pi(f)$ on the linear vector space $\mathcal{F}_m$ and since the control variates are centered, $\beta^\star(f)$ satisfies the normal equations $\pi(hh^\top)\beta^\star(f) = \pi(hf)$. The integral $\pi(f)$ thus appears as the intercept of a linear regression model with

response f and explanatory variables $h_1, \dots, h_m$, and it can be expressed as

$$(\pi(f), \beta^\star(f)) \in \underset{(\alpha,\beta)\in\mathbb{R}\times\mathbb{R}^m}{\arg\min}\ \pi[(f-\alpha-\beta^T h)^2]. \tag{2.2}$$

The empirical risk minimization paradigm applied to the risk function on the right-hand side of (2.2) will lead to the OLS and LASSO estimates, to be defined further in this section. The same paradigm suggests the use of other regression methods for MC integration such as Principal Component Regression (PCR) or Ridge Regression, which will not be considered in this chapter.

Remark 2.1 (Choice of control variates)**.** *Which control variates work well depends on the problem. In the Black–Scholes model, for instance, an effective control variate for the price of an option is the geometric average of the price series (Glasserman, 2004, Example 4.1.2)). Two generic ways to construct control variates are to be noted. Whenever $\pi(\mathrm{d}x) = w(x)\,Q(\mathrm{d}x)$, where $w : \mathcal{X} \to [0,\infty)$ and Q is a probability measure on $(\mathcal{X}, \mathcal{A})$, the quantity of interest is $\pi(f) = Q(wf)$, so that we can use control variates for wf with respect to Q. This trick can be useful in combination with importance sampling (Owen and Zhou, 2000). If π has density p with respect to the Lebesgue measure and if we have access to the derivatives of p, Stein's method might be used to build infinitely many control functions (Oates et al., 2017).*

Ordinary Least Squares Monte Carlo. Replacing the distribution π by the sample measure π_n in (2.2), we obtain the OLS estimator $\hat{\alpha}_n^{\mathrm{ols}}(f)$ of $P(f)$ as a minimizer of the empirical risk

$$\left(\hat{\alpha}_n^{\mathrm{ols}}(f), \hat{\beta}_n^{\mathrm{ols}}(f)\right) \in \underset{(\alpha,\beta)\in\mathbb{R}\times\mathbb{R}^m}{\arg\min} \left\{ \mathcal{R}_n(\alpha,\beta) = \left\| f^{(n)} - \alpha \mathbb{1}_n - H\beta \right\|_2^2 \right\} \tag{2.3}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, $\mathbb{1}_n = (1,\dots,1)^\top \in \mathbb{R}^n$ is a vector of ones, $f^{(n)} = (f(X_1),\dots,f(X_n))^\top \in \mathbb{R}^n$ is the vector of evaluations and H is the random $n \times m$ matrix defined by

$$H = \left(h_j(X_i)\right)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} = \begin{pmatrix} h_1(X_1) & \dots & h_m(X_1) \\ \vdots & \ddots & \vdots \\ h_1(X_n) & \dots & h_m(X_n) \end{pmatrix}.$$

The minimization problem in (2.3) can be expressed using an OLS estimate with centered variables as

$$\begin{cases} \hat{\alpha}_n^{\mathrm{ols}}(f) = \pi_n[f - \hat{\beta}_n^{\mathrm{ols}}(f)^\top h], \\ \hat{\beta}_n^{\mathrm{ols}}(f) \in \underset{\beta\in\mathbb{R}^m}{\arg\min}\, ||f_c^{(n)} - H_c\beta||_2^2, \end{cases} \tag{2.4}$$

where $f_c^{(n)} = f^{(n)} - \mathbb{1}_n(\mathbb{1}_n^\top f^{(n)})/n$ and $H_c = H - \mathbb{1}_n(\mathbb{1}_n^\top H)/n$. Indeed, for fixed $\beta \in \mathbb{R}^m$, the minimizer over $\alpha \in \mathbb{R}$ of the objective function in (2.3) is just $\pi_n(f - \beta^\top h) = \pi_n(f) - \beta^\top \pi_n(h)$, and since $\pi_n(f) = (\mathbb{1}_n^\top f^{(n)})/n$ and $\pi_n(h) = (\mathbb{1}_n^\top H)/n$, the equivalence of (2.3) and (2.4) follows.

Remark 2.2 (Variations)**.** *The solution of the linear regression problem (2.4) involves the empirical covariance matrix defined by* $n^{-1}H_c^\top H_c = \pi_n(hh^\top) - \pi_n(h)\pi_n(h^\top)$*. Using different estimates of the Gram matrix* $\pi(hh^\top)$ *leads to alternative control variate MC estimates for* $\pi(f)$ *(Glynn and Szechtman, 2002; Portier and Segers, 2019). For fixed* m *and as* $n \to \infty$*, all these estimators are consistent and asymptotically normal. The OLS estimator, however, is the only one that can integrate both the constant functions and the control functions without error.*

Remark 2.3 (Invariance)**.** *The OLS estimator does not change if we replace the control variate vector* h *by* Ah*, where* A *is an arbitrary invertible* $m \times m$ *matrix. Provided the control functions are linearly independent, the property of isotropy, i.e.,* $\pi(hh^\top) = I_m$*, can therefore always be enforced by an appropriate linear transformation of the vector of control variates.*

Remark 2.4 (Computation time)**.** *The computation time of the OLS method is of the order* $nm^2 + m^3 + nt$*, where* nm^2 *and* m^3 *operations are needed for computing and inverting* $H_c^\top H_c$ *respectively and where* t *stands for the time needed to evaluate* f*. Computational benefits occur when there are multiple integrands, since the OLS estimate can be represented as* $w^\top f^{(n)}$*, where the weight vector* $w \in \mathbb{R}^n$ *does not depend on the integrands Portier and Segers (2019). If* q *integrals need to be evaluated, the computing time becomes* $nm^2 + m^3 + qnt$*, since the matrix* $H_c^\top H_c$ *only depends on the control variates but not on the integrand.*

Remark 2.5 (Variance reduction)**.** *The advantage of using a given set of* m *control variates over standard MC can be assessed through the value of the residual standard deviation* $\sigma_m(f)$*. In Portier and Segers (2019), bounds for* $\sigma_m(f)$ *are computed in specific examples. For instance, if* $\mathcal{X} = [-1, 1]^d$ *and the* h_k *are tensor products of Legendre polynomials, then for any* k*-times continuously differentiable function* f *it holds that* $\sigma_m(f) = O(m^{-k/d})$ *as* $m \to \infty$*. This bound emphasizes the benefits of using polynomials when the integrand is regular.*

LASSO Monte Carlo. The LASSO, introduced in Tibshirani (1996), is a regression technique that consists in minimizing the usual least squares loss plus an ℓ_1-penalty term on the vector of regression coefficients. In contrast with OLS, the LASSO usually produces a vector with many zero coefficients, meaning that the corresponding variables are no longer included in the predictive model. The LASSO thus achieves estimation and variable selection at the same time. As the use of control variates in MC integration is linked with regression, the LASSO can take advantage from situations where many control variates are present but not all of them are useful.

The LASSO estimator $\hat{\alpha}_n^{\text{lasso}}(f)$ of $\pi(f)$ follows from adding a ℓ_1-penalization to the objective function in (2.3). It is formally defined as

$$\left(\hat{\alpha}_n^{\text{lasso}}(f), \hat{\beta}_n^{\text{lasso}}(f)\right) \in \underset{(\alpha,\beta)\in\mathbb{R}\times\mathbb{R}^m}{\arg\min} \frac{1}{2n}\mathcal{R}_n(\alpha, \beta) + \lambda \left\|\beta\right\|_1$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm on Euclidean space. By the same argument used to justify the equivalence of (2.3) and (2.4), the LASSO can be based on centered variables via

$$\begin{cases} \hat{\alpha}_n^{\text{lasso}}(f) = \pi_n[f - \hat{\beta}_n^{\text{lasso}}(f)^\top h], \\ \hat{\beta}_n^{\text{lasso}}(f) \in \underset{\beta \in \mathbb{R}^m}{\arg\min} \dfrac{1}{2n} ||f_c^{(n)} - H_c\beta||_2^2 + \lambda \left\|\beta\right\|_1 . \end{cases} \tag{2.5}$$

Remark 2.6 (Computation)**.** *For the practical implementation of the LASSO, it is commonly recommended to first center and rescale the explanatory variables empirically (Tibshirani et al., 2015, section 2.2). The centering by the sample mean is taken care of in (2.5). However, for ease of presentation, no empirical rescaling of the control variates is considered in the theoretical analysis. This is in line with the approach proposed in (Tibshirani et al., 2015, Chapter 11). Still, such rescaling is done in the simulation experiments reported in Section 2.4.*

Remark 2.7 (Computation time)**.** *The LASSO solution is usually computed approximately by* cyclical coordinate descent*. At each iteration, this algorithm minimizes (2.5) with respect to a single coordinate, say β_k, while considering other coordinates, $\beta_{(-k)} \in \mathbb{R}^{m-1}$, as constant. This one-dimensional optimization problem has an explicit argmin. Let $H_{c,k}$ be the k-th column of H_c that has been normalized such that $\|H_{c,k}\|_2 = 1$ (as indicated in the previous remark). The argmin is then simply given by $\eta_\lambda(\langle z_k, H_{c,k}\rangle)$ where $z_k = f_c^{(n)} - H_{c,(-k)}\beta_{(-k)}$, $H_{c,(-k)}$ is obtained by removing $H_{c,k}$ from H_c and η is the* soft-thresholding function *(Tibshirani et al., 2015, Section 2.4, Eq. (2.14)). Since n operations are needed to update z_k and the same number is needed to compute the scalar product, the LASSO requires only $nD + nt$ operations, where D stands for the number of iterations conducted in the cyclical coordinate descent and t represents the time needed to evaluate f. The value of D is often imposed by a stopping rule within the algorithm but it could also be fixed by the user in order to control the computing time. The selection of the next coordinate k to update can be done cyclically or at random.*

LSLASSO Monte Carlo. The application of ordinary least squares after model selection by the LASSO has been recently studied in Belloni and Chernozhukov (2013). They show, in the setting of nonparametric regression, that OLS post-LASSO, which is also known under the name LSLASSO, performs better than the LASSO in terms of rate of convergence. Motivated by this result we propose to first use the LASSO to select the active variables among a large number of control variates and then to compute the OLS estimate using only the variables selected at the previous stage. We refer to this approach as the LSLASSO. To decrease the computation time when the dimensions involved in the problem, either n or m, are large, we recommend to use sub-sampling of size N smaller than n when conducting the first step.

The *active set* associated to the coefficient $\beta \in \mathbb{R}^m$ is $\text{supp}(\beta) = \{j = 1, \dots, m : \beta_j \neq 0\}$. Let $\hat{S}_N = \text{supp}(\hat{\beta}_N^{\text{lasso}}(f))$ denote the active set of control variates based on the LASSO coefficient vector defined as in (2.5) but using only the first N random variables $X_1, \dots, X_N$ generated. The LSLASSO estimate $\hat{\alpha}_n^{\text{lslasso}}(f)$ of $\pi(f)$ is then defined as the OLS estimate in (2.3) based on the full sample $X_1, \dots, X_n$ but using only the control variates h_j restricted to $j \in \hat{S}_N$, that is,

$$\left(\hat{\alpha}_n^{\text{lslasso}}(f), \hat{\beta}_n^{\text{lslasso}}(f)\right) \in \underset{(\alpha,\beta) \in \mathbb{R} \times \mathbb{R}^{\hat{\ell}}}{\arg\min} \left\| f^{(n)} - \alpha \mathbb{1}_n - H_{\hat{S}_N}^{(n)} \beta \right\|_2^2$$

where $H_{\hat{S}_N}^{(n)}$ is the $n \times \hat{\ell}$ matrix $(h_j(X_i))_{i=1,\dots,n,\, j\in\hat{S}_N}$ and $\hat{\ell}$ is the cardinality of $\hat{S}_N$.

Remark 2.8 (Computation time)**.** *The number of operations needed for the LSLASSO is of the order $ND + n\hat{\ell}^2 + \hat{\ell}^3 + nt$, combining the cost of selecting the control variates on the subsample of size N via cyclical coordinate descent as in Remark 2.7 and running the OLS estimate based on the selected control variates for the full sample of size n as in Remark 2.4.*

2.3 Non-asymptotic bounds

To derive concentration inequalities for the errors of the estimators proposed in Section 2.2, we use the notion of sub-Gaussianity as defined for instance in (Boucheron et al., 2013a, Section 2.3). Recall that the moment generating function of a centered Gaussian random variable with variance σ^2 is equal to $\lambda \mapsto \exp(\lambda^2\sigma^2/2)$.

Definition 2.9. *A centered random variable Y is* sub-Gaussian *with variance factor $\tau^2 > 0$, notation $Y \in \mathcal{G}(\tau^2)$, if and only if $\log \mathbb{E}[\exp(\lambda Y)] \leq \lambda^2\tau^2/2$ for all $\lambda \in \mathbb{R}$.*

If $Y \in \mathcal{G}(\tau^2)$, then necessarily $\mathrm{Var}(Y) \leq \tau^2$ (Boucheron et al., 2013a, Exercise 2.16). Chernoff's inequality provides exponential bounds on the tails of sub-Gaussian random variables. Moreover, the sum of independent sub-Gaussian variables is again sub-Gaussian. Centered, bounded random variables taking values in an interval $[a, b]$ are sub-Gaussian with variance factor at most $(b-a)^2/4$ (Boucheron et al., 2013a, Lemma 2.2).

The concentration inequalities for the various Monte Carlo methods with control variates will be largely due to the following assumption that requires the residuals to be sub-Gaussian.

Assumption 2.10 (Sub-Gaussian residuals)**.** *The residual function $\epsilon = f - \pi(f) - \beta^\star(f)^\top h$ satisfies $\epsilon \in \mathcal{G}(\tau^2)$ for some $\tau > 0$, that is, $\int_{\mathcal{X}} \exp\{\lambda\epsilon(x)\}\,\pi(\mathrm{d}x) \leq \exp(\lambda^2\tau^2/2)$ for all $\lambda \in \mathbb{R}$.*

The estimation error of the oracle estimator in (2.1) is just $\hat{\alpha}_n^{\mathrm{or}}(f) - \pi(f) = \pi_n(\epsilon) = n^{-1}\sum_{i=1}^n \epsilon(X_i)$. Under Assumption 2.10, this is a sub-Gaussian variable with variance factor τ^2/n. Chernoff's inequality (Boucheron et al., 2013a, p. 25) then implies that for all $\delta \in (0,1)$ and all integer $n \geq 1$, with probability at least $1-\delta$,

$$\left|\hat{\alpha}_n^{\mathrm{or}}(f) - \pi(f)\right| \leq \sqrt{2\log(2/\delta)}\frac{\tau}{\sqrt{n}}. \tag{2.6}$$

This concentration inequality provides a baseline when the best possible control variate in the space $\mathcal{F}_m$ is selected. The case $m = 0$ also covers the basic MC method: in that case, τ^2 is the variance factor of the sub-Gaussian variable $f - \pi(f)$ on $(\mathcal{X}, \mathcal{A}, P)$.

Assumption 2.11 (Bounded control variates)**.** *The control variates $h_1, \dots, h_m \in L_2(\pi)$ are uniformly bounded. Put $U_h := \max_{j=1,\dots,m} \sup_{x\in\mathcal{X}} \left|h_j(x)\right|$.*

For a symmetric real matrix A, let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote its smallest and largest eigenvalues, respectively.

Assumption 2.12 (Linear independence of control variates)**.** *The control variates h_1, …, $h_m \in L_2(\pi)$ are linearly independent. As a consequence, the $m \times m$ Gram matrix $G := \pi(hh^\top)$ is positive definite and its smallest eigenvalue $\gamma := \lambda_{\min}(G)$ is positive.*

Consider the ortho-normalized vector of control variates $\hbar = (\hbar_1, \ldots, \hbar_m)^\top = G^{-1/2}h$ and put

$$B = \sup_{x \in \mathcal{X}} h(x)^\top G^{-1} h(x) = \sup_{x \in \mathcal{X}} \hbar(x)^\top \hbar(x), \tag{2.7}$$

a finite quantity by Assumptions 2.11 and 2.12. The error OLS estimation error is subject to the following concentration bound.

Theorem 2.13 (Concentration inequality for OLS)**.** *Suppose Assumptions 2.10, 2.11 and 2.12 hold. Then for all $\delta \in (0,1)$ and all integer n such that*

$$n \geq \max\Big(18B\log(4m/\delta),\ 75m\log(4/\delta)\Big)$$

we have, with probability at least $1-\delta$,

$$\left|\hat{\alpha}_n^{\mathrm{ols}}(f) - \pi(f)\right| \leq \sqrt{2\log(8/\delta)}\frac{\tau}{\sqrt{n}} + 58\sqrt{Bm\log(8m/\delta)\log(4/\delta)}\frac{\tau}{n}. \tag{2.8}$$

Compared to the bound (2.6) for the oracle estimator, the bound (2.8) for the OLS estimator has an additional term. This term is due to the additional learning step that is needed to estimate the optimal control variate.

Remark 2.14 (On the factor B)**.** *Defined as the supremum of the* leverage function *q_n in (Portier and Segers, 2019, Eq. (14)), the quantity B plays an important role in our analysis as well as in other regression studies (Hsu et al., 2012; Newey, 1997). Just as the OLS estimate (see Remark 2.3), the quantity B remains invariant by invertible linear transformation of the control variates. We have*

$$m \leq B \leq \sup_{x \in \mathcal{X}} h^\top(x) h(x)/\gamma \leq mU_h^2/\gamma.$$

Remark 2.15 (On the parameters τ and γ)**.** *The parameter τ in Assumption 2.10 is by definition an upper bound of the residual variance $\sigma_m^2(f)$. In many situations, its value is not too far from $\sigma_m^2(f)$. Hence, τ should capture the adequacy between the control variate space and the integrand f and should decrease with m. The full rank condition expressed in Assumption 2.12 is not crucial as one could work with the Moore–Penrose inverse when solving (2.4). More importantly, a large value of the minimal eigenvalue γ of the Gram matrix G reflects that the OLS problem is well-conditioned, enhancing numerical stability. As control functions are added, rows and columns are added to G and so γ cannot increase. For the Fourier basis in Example 2.28, we have $\gamma = 1$, while for the Legendre polynomials in Example 2.29, we have $\gamma \simeq 1/m$.*

Remark 2.16 (Link with OLS prediction risk analysis)**.** *The approach taken in the proof of Theorem 2.13 requires to bound what is called the prediction risk, defined as $\|G^{1/2}(\hat{\beta}_n^{\mathrm{ols}}(f) - \beta^\star(f))\|_2$. With probability greater than $1-\delta$, we obtain an upper bound of order $\sqrt{B\tau^2\log(m/\delta)/n}$ on the prediction risk. This makes our approach comparable to the one of the recent study (Hsu et al., 2012) where concentration bounds for the OLS prediction risk (and ridge) with random design are established. In contrast to their*

bound, our bound involves the quantity B which shares the same invariant property as the OLS estimate and we don't require the noise to be sub-Gaussian conditionally on the covariate but just sub-Gaussian which is weaker.

Remark 2.17 (Rates)**.** *Consider an asymptotic set-up where the number of control variates m tends to infinity with the Monte Carlo sample size n. The OLS method improves upon the basic MC method ($m = 0$), which has rate $1/\sqrt{n}$, as soon as $\tau + \tau\sqrt{mB\log(m)/n} \to 0$. To recover the same order as the one of the oracle estimator $\hat{\alpha}_n^{\mathrm{or}}(f)$, which has rate $\tau/\sqrt{n}$, one must have $mB\log(m) = O(n)$ as $n \to \infty$, that is, m must not be too large compared to n.*

Remark 2.18 (Leverage condition)**.** *Theorem 2.13 may be seen as a non-asymptotic version of the asymptotic results provided in Portier and Segers (2019) in which the leverage condition, $\sup\{h(x)^\top G^{-1} h(x) : x \in \mathcal{X}\} = o(n/m)$, is required to obtain a similar (asymptotic) bound (see Theorem 1 therein) as the one of Theorem 2.13. In the present non-asymptotic version, the leverage condition is expressed through mB when requiring that $18B\log(4m/\delta) \leq n$.*

LASSO takes advantage of *sparse* regression models. A regression model is sparse whenever many of the coefficients of the parameter vector β are equal to zero, i.e., many of the covariates are useless to predict the output in the presence of the other covariates. The number of elements in the active set of the vector of regression coefficients $\beta^\star(f)$,

$$S^\star := \mathrm{supp}(\beta^\star(f)),$$

is denoted by $\ell^\star := \left|S^\star\right|$ and quantifies the level of sparsity associated to the regression model. To avoid trivialities, we tacitly assume that $S^\star$ is non-empty, so $\ell^\star \geq 1$. The factor $\ell^\star$ represents the level of sparsity of f with respect to the control functions and plays an important role in describing the benefits of the LASSO over the OLS. No assumption is made on $\ell^\star$, which could be any integer in $\{1, \ldots, m\}$.

We follow the approach presented in (Tibshirani et al., 2015, Section 11.4.1) (see also Bickel et al. (2009); van de Geer and Bühlmann (2009)), in which the analysis of the LASSO is carried out using a *restricted eigenvalue condition*. For a vector $\beta \in \mathbb{R}^m$ and for a non-empty set $S \subset \{1, \ldots, m\}$, write $\beta_S = (\beta_k)_{k \in S}$, seen as a (column) vector in $\mathbb{R}^{|S|}$. Define a collection of cones of interest. For $\alpha > 0$ and $S \subset \{1, \ldots, m\}$, we set $\overline{S} = \{1, \ldots, m\} \setminus S$ and

$$\mathcal{C}(S; \alpha) = \{u \in \mathbb{R}^m \, : \, \left\|u_{\overline{S}}\right\|_1 \leq \alpha \left\|u_S\right\|_1\}.$$

Assumption 2.19 (Restricted eigenvalue condition)**.** *There exists $\gamma^\star > 0$ such that $u^\top G u \geq \gamma^\star \|u\|_2^2$ for all $u \in \mathcal{C}(S^\star; 3)$.*

In practice, we do not know the active set $S^\star$, so the only way to ensure Assumption 2.19 is to make sure all control variates $h_1, \ldots, h_m$ are linearly independent. The practical value of the assumption is that $\gamma^\star \geq \gamma$, yielding sharper bounds below.

Recall that the ℓ_1-penalty of the LASSO is weighted by a regularization parameter $\lambda > 0$.

Theorem 2.20 (Concentration inequality for LASSO)**.** *Suppose Assumptions 2.10, 2.11 and 2.19 hold. Introduce* $\xi = \ell^\star(U_h^2/\gamma^\star)$*. Then for all* $\delta \in (0,1)$ *and all integer* n *such that*

$$n \geq \max\left(8\xi^2 \log(8m^2/\delta); 128\xi \log(8m/\delta)\right),$$
$$\lambda \geq 7U_h\sqrt{\log(8m/\delta)}\tau/\sqrt{n}$$

we have, with probability at least $1-\delta$*,*

$$\left|\hat{\alpha}_n^{\text{lasso}}(f) - \pi(f)\right| \leq \sqrt{2\log(8/\delta)}\frac{\tau}{\sqrt{n}} + 68\lambda\ell^\star\sqrt{\log(8m/\delta)}\frac{U_h/\gamma^\star}{\sqrt{n}}. \tag{2.9}$$

For λ *equal to the lower bound, we have on the same event*

$$\left|\hat{\alpha}_n^{\text{lasso}}(f) - \pi(f)\right| \leq \sqrt{2\log(8/\delta)}\frac{\tau}{\sqrt{n}} + 476\ell^\star\log(8m/\delta)(U_h^2/\gamma^\star)\frac{\tau}{n}. \tag{2.10}$$

Remark 2.21 (LASSO vs OLS)**.** *The benefits of LASSO over OLS can be observed by comparing the bounds in (2.8) and (2.10). The total number* m*, of control functions has been replaced by the active number* $\ell^\star$ *of such functions. Further, because* $\Gamma_{S^\star} = \{u \in \mathbb{R}^p : \|u\|_2 = 1, u \in \mathcal{C}(S^\star; 3)\}$ *is included in the unit sphere,* $\gamma^\star = \inf_{u\in\Gamma_{S^\star}} u^\top G u$ *in Assumption 2.12 is at least as large as the smallest eigenvalue of* G*,* $\gamma = \inf_{\|u\|_2=1} u^\top G u$ *in Assumption 2.19.*

The theoretical analysis of the LSLASSO estimator depends on the success of the LASSO-based model selection, i.e., the LASSO needs to correctly recover all the components of the true model. To ensure this selection step, the restricted eigenvalue condition is replaced by the two following ones.

Assumption 2.22 (Linear independence of active functions)**.** *The active control variates* h_k*,* $k \in S^\star$*, are linearly independent. As a consequence, the* $\ell^\star \times \ell^\star$ *Gram matrix* $G_{S^\star} = P(h_{S^\star} h_{S^\star}^\top)$ *is positive definite and its smallest eigenvalue* $\gamma^{\star\star} := \lambda_{\min}(G_{S^\star})$ *is strictly positive.*

Note that because $\{u \in \mathbb{R}^p : \|u\|_2 = 1, \forall k \notin S, u_k = 0\} \subset \Gamma_S$ (introduced in remark 2.21), we have that $\gamma^{\star\star} \geq \gamma^\star$. Finally, it is required that that the active control functions are orthogonal, in $L_2(\pi)$, to the inactive ones.

Assumption 2.23 (Orthogonality)**.** *We have* $\pi(h_j h_k) = 0$ *for all* $j \in \{1,\dots,m\} \setminus S^\star$ *and all* $k \in S^\star$*.*

Since we do not know $S^\star$ in practice, the way to ensure Assumption 2.23 is by making all control variates orthogonal: $\pi(h_j h_k) = 0$ for all $j, k \in \{1,\dots,m\}$. The Gram matrices G and $G^\star$ are then diagonal. In the absence of zero control variates, Assumptions 2.12 and 2.19 are then satisfied as well, with $\gamma^{\star\star} = \min_{k\in S^\star} \pi(h_k^2) \geq \min_{k=1,\dots,m} \pi(h_k^2) = \gamma > 0$.

Theorem 2.24 (Support recovery of LASSO)**.** *Suppose Assumptions 2.10, 2.11, 2.22 and 2.23 hold. Then for all* $\delta \in (0,1)$*, all integer* n *such that*

$$n \geq 70(\ell^\star U_h^2/\gamma^{\star\star})^2 \log(10\ell^\star m/\delta),$$

and all λ *such that*

$$13U_h\sqrt{\log(10m/\delta)}\frac{\tau}{\sqrt{n}} \leq \lambda \leq \frac{\gamma^{\star\star}}{3\sqrt{\ell^\star}}\min_{k\in S^\star}|\beta_k^\star(f)|, \tag{2.11}$$

it holds that, with probability at least $1-\delta$*, the LASSO based solution* $\hat{\beta}_n^{\text{lasso}}(f)$ *is unique and the true active set is recovered,* $\text{supp}(\hat{\beta}_n^{\text{lasso}}(f)) = S^\star$.

The upper and lower bounds on λ in (2.11) must not contradict each other, and this effectively implies an additional lower bound on n. Define $B^\star = \sup_{x\in\mathcal{X}} h_{S^\star}^\top(x)G_{S^\star}^{-1}h_{S^\star}(x)$ and note that

$$B^\star \leq \lambda_{\max}(G_{S^\star}^{-1})\sup_{x\in\mathcal{X}} h_{S^\star}^\top(x)h_{S^\star}(x) \leq \ell^\star U_h^2/\gamma^{\star\star}. \tag{2.12}$$

Theorem 2.25 (Concentration inequality for LSLASSO)**.** *Suppose Assumptions 2.10, 2.11, 2.22 and 2.23 hold. Write* $\xi^\star = \ell^\star(U_h^2/\gamma^{\star\star})$*. Then for all* $\delta \in (0,1)$ *and all integer* $N \in \{1,\dots,n\}$ *such that*

$$N \geq 75\xi^{\star 2}\log(20\ell^\star m/\delta),$$

and all λ *such that*

$$13U_h\sqrt{\log(20m/\delta)}\frac{\tau}{\sqrt{N}} \leq \lambda \leq \frac{\gamma^{\star\star}}{3\sqrt{\ell^\star}}\min_{k\in S^\star}|\beta_k^\star(f)|,$$

we have, with probability at least $1-\delta$*,*

$$\left|\hat{\alpha}_n^{\text{lslasso}}(f) - \pi(f)\right| \leq \sqrt{2\log(16/\delta)}\frac{\tau}{\sqrt{n}} + 58\sqrt{B^\star\ell^\star\log(16\ell^\star/\delta)\log(8/\delta)}\frac{\tau}{n}. \tag{2.13}$$

The logic behind Theorem 2.25 is that, by Theorem 2.24, the active set $\hat{S}_N = \text{supp}(\hat{\beta}_N^{\text{lasso}}(f))$ identified by means of the subsample of size N is equal to the true active set $S^\star = \text{supp}(\beta^\star(f))$ with large probability. On the event that the two sets coincide, the LSLASSO estimator is then the same as the OLS estimator based on the active control variates only, and the error bound follows from Theorem 2.13. In practice, it turns out that LSLASSO works well even when the true active set is not identified perfectly. However, to show this formally remains an open problem.

The assumptions and concentration inequalities in our theorems feature explicit rather than generic constants. Although we have worked hard to keep these constants under control [see in particular the proof of Lemma 2.34 as well as Step 6(ii) in the proof of Theorem 2.24], it is likely that, at the cost of lengthier computations, sharper constants can still be found.

Remark 2.26 (Bounded control variates)**.** *In Assumption 2.11, the control variates were assumed to be bounded. Even if this assumption is valid for the two classic families in Examples 2.28 and 2.29 below, it might fail when control variates are produced with the Stein's method as suggested in Remark 2.1. The boundedness assumption is needed to keep the same variance factor* τ^2 *in the sub-Gaussian property of both variables* $\epsilon(X_1)$ *and* $\epsilon(X_1)h(X_1)$*; see, e.g., Step 3.2 in the proof of Theorem 2.13 or Equation (2.35) in the proof of Theorem 2.20. Avoiding this assumption is thus possible at the price of more specific assumptions on the sub-Gaussianity of* $\epsilon(X_1)h(X_1)$*. Note finally that (different) asymptotic results are valid for unbounded control variates (Portier and Segers, 2019).*

Remark 2.27 (Overfitting)**.** *Theorems 2.20 and 2.25 advocate the use of the LASSO in favor of the OLS in scenarios where ℓ^* is smaller than m or in the presence of collinearities in the design matrix making the parameter γ close to zero; see also Remark 2.21. Another notable advantage of the (LS)LASSO and more generally of penalization methods, is the ability to prevent over-fitting. This occurs when the number of control variates m is large compared to the Monte Carlo sample size n or, more generally, when the approximation space is large compared to the sample size. While the theory developed here is unable to address such phenomena, one of the objectives of the numerical experiments conducted in the next section is to empirically demonstrate the superior performance of the LASSO-based methods even in the absence of sparsity.*

To illustrate the application of our results in a standard framework, we consider two classic families of control functions, the Fourier basis and the Legendre polynomials.

Example 2.28 (Fourier basis)**.** *On $\mathcal{X} = [0,1]$ equipped with the uniform distribution P, let $h_j(x)$ be equal to $\sqrt{2}\cos((j+1)\pi x)$ is j is odd and to $\sqrt{2}\sin(j\pi x)$ is j is even. The Fourier basis is orthonormal so that the Gram matrix is the identity, $G = I_m$, and $\gamma = \gamma^\star = \gamma^{\star\star} = 1$. The cosine and sine functions being bounded by 1, a uniform bound is $U_h = \sqrt{2}$, which implies $B \leq 2m, B^\star \leq 2\ell^\star$. Under the proper assumptions, we get from Theorems 2.13 and 2.25 that with probability at least $1-\delta$, since $58\sqrt{2} < 83$,*

$$\left|\hat{\alpha}_n^{\text{ols}}(f) - \pi(f)\right| \leq \sqrt{2\log(8/\delta)}\frac{\tau}{\sqrt{n}} + 83m\sqrt{\log(8m/\delta)\log(4/\delta)}\frac{\tau}{n}$$

and

$$\left|\hat{\alpha}_n^{\text{lslasso}}(f) - \pi(f)\right| \leq \sqrt{2\log(16/\delta)}\frac{\tau}{\sqrt{n}} + 83\ell^\star\sqrt{\log(16\ell^\star/\delta)\log(8/\delta)}\frac{\tau}{n}.$$

Example 2.29 (Legendre polynomials)**.** *Suppose that $h_j = L_j$ is the Legendre polynomial of degree $j \in \{1,\dots,m\}$. The Legendre polynomials are orthogonal on $\mathcal{X} = [-1,1]$ with respect to the uniform distribution π and satisfy $|L_j(x)| \leq 1$ for $x \in [-1,1]$ with $L_j(1) = 1$ and*

$$\int_{-1}^{1} L_i(x)L_j(x)\,\mathrm{d}x = \frac{2}{2j+1}\delta_{ij}.$$

The Gram matrix $G = \pi(hh^\top)$ is diagonal with entries $1/(2j+1)$, so the minimum eigenvalue is $\gamma = 1/(2m+1)$ and a uniform bound is $U_h = 1$. Consequently, $B \leq 2m+1$. Similarly, considering only active control variates, we have $U_h^\star = 1$, while the smallest eigenvalue, $\gamma^{\star\star}$, of $G_{S^\star}$ satisfies $1/(2m+1) \leq \gamma^{\star\star} \leq 1/(2\ell^\star+1)$. Under suitable assumptions, we get from Theorems 2.13 and 2.25 that with probability at least $1-\delta$,

$$\left|\hat{\alpha}_n^{\text{ols}}(f) - \pi(f)\right| \leq \sqrt{2\log(8/\delta)}\frac{\tau}{\sqrt{n}} + 58\sqrt{(2m+1)m\log(8m/\delta)\log(4/\delta)}\frac{\tau}{n},$$

$$\left|\hat{\alpha}_n^{\text{lslasso}}(f) - \pi(f)\right| \leq \sqrt{2\log(16/\delta)}\frac{\tau}{\sqrt{n}} + 58\sqrt{(2\ell^\star+1)\ell^\star\log(16\ell^\star/\delta)\log(8/\delta)}\frac{\tau}{n}.$$

Compared to the Fourier basis, the improvement of LSLASSO over the OLS estimator is not only related to the number of active varables ℓ^* compared to m but also to the place of the active variables within the set of Legendre polynomials.

2.4 Numerical illustration

To compare the finite-sample performance of the various control variate methods, we consider synthetic data examples involving the standard integration problem over the unit cube $[0,1]^d$. The goal is to compute $\int_{[0,1]^d} f(x)\,\mathrm{d}x$. We shall consider various dimensions $d \geq 1$, different integrands $f : [0,1]^d \to \mathbb{R}$, and several choices for the Monte Carlo sample size, n, and the number of control variates, m. We shall focus on difficult situations where d is relatively large compared to n. In Section 2.5, we turn to real data examples in the context of Bayesian inference. For the sake of reproducibility, the data and Python code are available online[1].

Methods in competition. We consider all the methods presented in Section 2.2 with two different strategies regarding the sub-sample size used to compute the active set in LSLASSO. The methods in competition are OLS, LASSO, LSLASSO (sub-sample size $N = n$) and LSLASSOX (sub-sample size $N = \lfloor 15\sqrt{n} \rfloor$). The latter choice accelerates the computation in a substantial manner without deteriorating too much the support recovery property of the LASSO. For synthetic data, because the integration domain is the unit cube $[0,1]^d$, Quasi-Monte Carlo (QMC) methods (Caflisch, 1998) are suitable for comparison. We run such methods in the experiments with two classical low-discrepancy sets of particles, namely Halton and Sobol sequences.

On the choice of λ. In the LASSO-step of LSLASSO(X), the choice of the regularization parameter λ is essential since it controls the number of active variables. It is common to tune this parameter using K-fold cross-validation at the price of additional computations. This method, presented in general form in Algorithm 2.1, uses the prediction error of the underlying regression problem as a proxy to calibrate the control variates estimate. In Algorithm 2.1, the "data" X correspond to the matrix H of observed control variables and the "labels" y to the vector $f^{(n)}$ of observed function values. The method is computationally expensive, partitioning the training set in several folds and solving many regression problems for every value of λ in a given grid.

Algorithm 2.1 K-fold cross-validation

Require: data X, labels y, grid search λ_{grid}, n, K.

1. Divide $\{1, \ldots, n\}$ into K folds $F_1, \ldots, F_K$.
2. **For** $k = 1, \ldots, K$
3. Set training folds $F_{-k} = \{F_1, \ldots, F_{k-1}, F_{k+1}, \ldots, F_K\}$.
4. **For** $\lambda \in \lambda_{\text{grid}}$
5. Compute estimate $\hat{\beta}_\lambda^{-k}$ on training set.
6. Compute test error $e_k(\lambda) = \sum_{i \in F_k} (y_i - x_i^\top \hat{\beta}_\lambda^{-k})^2$.
7. **For** $\lambda \in \lambda_{\text{grid}}$
8. Compute average error $CV(\lambda) = \frac{1}{n} \sum_{k=1}^{K} e_k(\lambda)$.
9. **Return** $\hat{\beta}_{\lambda^\star}$ with $\lambda^\star \in \arg\min_{\lambda \in \lambda_{\text{grid}}} CV(\lambda)$.

[1] https://github.com/RemiLELUC/ControlVariateSelection.git

To accelerate the computations, we suggest a new method based on a dichotomic search. Motivated by Eq. (2.8) and Remark 2.17, the value of λ is tuned such that the number of selected control variates is of the order $\sqrt{n}$, which is the order obtained for m when equating the two terms in (2.8) with $B = m$. Specifically, we enforce the number of activated control functions to lie in the range $[c_1\sqrt{n}, c_2\sqrt{n}]$ for constants $0 < c_1 < c_2$ to be chosen (see below). This choice offers two advantages. On the one hand, the upper bound $c_2\sqrt{n}$ ensures that the number of selected control variates is relatively small compared to the sample size n, promoting stability and fast computation in the final OLS step. On the other hand, the lower bound $c_1\sqrt{n}$ reduces the risk of excluding relevant control variates.

The full procedure for the selection of the regularization parameter using a dichotomic search is described below in Algorithm 2.2. In all experiments, we set $c_1 = 3$ and $c_2 = 12$. We initialize $\lambda = \lambda_\infty$ to be the smallest value of λ for which $\hat{\beta}^{\text{lasso}} = 0$, that is, $\lambda_\infty = \max_{k=1,\dots,m} |H_{c,k}^{(N)T} f_c^{(N)}|/N$, where $H_{c,k}^{(N)}$ stands for the k-th column of $H_c^{(N)}$, which is the same as the matrix H_c but then based on the first N Monte Carlo draws (Tibshirani et al., 2015, Exercise 2.1). Next, we decrease the value of λ, e.g., by dividing it by two, such as to incorporate more and more control variates. If too many control functions are selected, i.e., more than $c_2\sqrt{n}$, we increase the value of λ again, e.g., by multiplying it by two, to finally reach the desired range for the number of active variables. In the end, this procedure ensures a straightforward computation of the LSLASSO(X) because the size of the associated linear system remains reasonable. Contrary to K-fold cross-validation, it is not necessary to split the data into multiple folds, leading to a reduced computation time.

Algorithm 2.2 Dichotomic Search

Require: $f_c^{(n)}$, H_c, n, $N \leq n$, (c_1, c_2).

1. Initialize $\lambda = \lambda_\infty$ and $\hat{\ell} = 0$.
2. **While** $\hat{\ell} \notin [c_1\sqrt{n}, c_2\sqrt{n}]$
3. $\quad \hat{\beta}_N^\lambda(f) \in \arg\min_{\beta\in\mathbb{R}^m} \frac{1}{2N}\|f_c^{(N)} - H_c^{(N)}\beta\|_2^2 + \lambda \left\|\beta\right\|_1$.
4. $\quad \hat{S}_N = \text{supp}(\hat{\beta}_N^\lambda(f))$ and $\hat{\ell} = \left|\hat{S}_N\right|$.
5. $\quad$ **if** $\hat{\ell} < c_1\sqrt{n}$ **then** decrease λ.
6. $\quad$ **if** $\hat{\ell} > c_2\sqrt{n}$ **then** increase λ.
7. **Return** $\hat{\beta}_N^\lambda(f)$.

The pseudo-code of the corresponding LSLASSO(X) method is provided in Algorithm 2.3. The regression coefficients $\hat{\beta}_n^{\text{ols}}$ and $\hat{\beta}_n^{\text{lasso}}$ for OLS and LASSO are computed using the Scikit-Learn library (Pedregosa et al., 2011), employing coordinate descent to solve the LASSO problem.

Algorithm 2.3 Least-Squares Lasso Monte-Carlo (LSLASSO)

Require: $f : \mathcal{X} \to \mathbb{R}$, $h_j : \mathcal{X} \to \mathbb{R}, 1 \le j \le m$, π, n, $N \le n$.

1. Generate $(X_i)_{i=1,\dots,n}$ independently according to π.
2. $f^{(n)} = (f(X_1), \dots, f(X_n))$ and $H = \Big(h_j(X_i)\Big)_{i=1,\dots,n}^{j=1,\dots,m}$.
3. $f_c^{(n)} = f^{(n)} - \mathbb{1}_n(\mathbb{1}_n^\top f^{(n)})/n$ and $H_c = H - \mathbb{1}_n(\mathbb{1}_n^\top H)/n$.
4. Solve $\hat{\beta}_N^\lambda(f)$ by cross-validation or dichotomic search.
5. $\hat{S}_N = \operatorname{supp}(\hat{\beta}_N^\lambda(f))$ and $\hat{\ell} = \left|\hat{S}_N\right|$.
6. Slice $n \times \hat{\ell}$ matrix $H_{c,\hat{S}_N}^{(n)} = (H_{cij}^{(n)})_{i=1,\dots,n,\, j\in\hat{S}_N}$
7. $\hat{\beta}^{\text{lslasso}}(f) \in \arg\min_{\beta\in\mathbb{R}^m} \|f_c^{(n)} - H_{c,\hat{S}_N}^{(n)}\beta\|_2^2$.
8. MC estimate $\hat{\alpha}_{n,N}^{\text{lslasso}}(f) = P_n[f - \hat{\beta}^{\text{lslasso}}(f)^\top h]$.

Integrands. We consider several integrands f on $[0,1]^d$:

$$\varphi(x_1, \dots, x_d) = 1 + \sin\left(\pi\left(\frac{2}{d}\sum_{i=1}^d x_i - 1\right)\right), \tag{2.14}$$

and for all $j = 1, \dots, d$,

$$f_j(x_1, \dots, x_d) = \prod_{i=1}^j (2/\pi)^{1/2} x_i^{-1} \mathrm{e}^{-\log(x_i)^2/2}, \tag{2.15}$$

$$g_j(x_1, \dots, x_d) = \prod_{i=1}^j \frac{\log(2)}{2^{x_i - 1}} = \log(2)^j 2^{\sum_{i=1}^j (1-x_i)}. \tag{2.16}$$

All these functions integrate to 1 on $[0,1]^d$. The functions f_j and g_j are built using tensor products of log-normal and exponential density functions, respectively, and depend on the first j coordinates only. This construction ensures that for small j, the integrands f_j and g_j lend themselves to Monte Carlo integration based on selected control variates. In contrast, the functions φ, f_d and g_d represent more difficult situations where all the coordinates are involved and the symmetry of their role makes it harder to select some meaningful control functions. None of the integrands belongs to the linear span of the control variates constructed in the next paragraph.

Control variates. Multidimensional control functions with respect to the uniform distribution over $[0,1]^d$ are easy to construct based on univariate ones. Let $(h_1, \dots, h_k)$ be a vector of one-dimensional control functions, i.e., $\int_0^1 h_j(x)\,\mathrm{d}x = 0$ for each $j = 1, \dots, k$. Let $h_0 = 1$ denote the constant function equal to one. Without further information on the integrand, the usual way to construct multivariate controls is by forming tensor products of the form

$$h_\ell(x_1, \dots, x_d) = \prod_{j=1}^d h_{\ell_j}(x_j)$$

for a multi-index $\ell = (\ell_1, \dots, \ell_d)$ in $\{0, \dots, k\}^d \setminus \{(0, \dots, 0)\}$, yielding a total number of $(k+1)^d - 1$ control functions.

A drawback of such a construction is that the number of control functions grows quickly with k. Alternative approaches yielding smaller control spaces consist of imposing $\ell_j = 0$ for all but a small number (one or two, say) of coordinates $j = 1, \dots, d$ or simply picking at random a desired number, say m, of indices $\ell = (\ell_1, \dots, \ell_d)$.

In this study, the set of control variates at our disposal is constructed as follows. We consider different settings of dimension d with k univariate control functions in each dimension. For $j \in \{1, \dots, k\}$, let $h_j(x) = L_j(2x - 1)$ for $x \in [0, 1]$, with L_j the univariate Legendre polynomial (Legendre function of the first kind) of degree j; see Example 2.29. We have $\int_0^1 h_j(x)\,\mathrm{d}x = 0$ for all $j = 1, \dots, m$. Because the Legendre polynomials are orthogonal, they provide some numerical stability when inverting the Gram matrix. The multivariate control functions are sorted in ascending order according to the total degree $\sum_{j=1}^{d} \ell_j \in \{1, \dots, , kd\}$ of the polynomial. In the experiments, the number of control functions m is increased by progressively including all polynomials whose total degree is lower than or equal to a fixed threshold *deg*.

Settings. For the triple (d, k, n) we consider the dimension $d \in \{3, 5, 8\}$ with $k \in \{12, 10, 3\}$ and $n \in \{2\,000,\ 5\,000,\ 10\,000\}$. For each choice of (d, k), the number of control variates m with a total degree lower than or equal to a fixed threshold *deg* are given in Table 2.1. The case $d = 8$ represents a difficult situation as the number of points n is relatively small compared to the dimension. For instance, a grid made of only four points in each direction would already comprise 65 536 points.

d	k	Degree threshold (*deg*)				
		1	3	5	10	12
3	12	3	19	55	285	454
5	10	5	55	251	3 001	6 157
8	3	8	164	1 214	20 993	36 813

Table 2.1 – Number of control variates m by degree threshold *deg* in dimension d constructed out of tensor products of k univariate polynomials.

The sub-sample sizes N along with the bounds $c_1\sqrt{n}$ and $c_2\sqrt{n}$ are given in Table 2.2.

n	N	$\lfloor 3\sqrt{n} \rfloor$	$\lfloor 12\sqrt{n} \rfloor$
2 000	700	134	536
5 000	1 000	212	848
10 000	2 000	300	1 200

Table 2.2 – Sample sizes n and sub-sample sizes N together with the range $[c_1\sqrt{n}, c_2\sqrt{n}]$ corresponding to the imposed number of selected control variates in LSLASSO.

Results. The different Monte Carlo estimates are compared on the basis of their mean squared error (MSE). Figure 2.1 presents the boxplots obtained over 100 replications of the values returned by each of the methods. In Tables 2.3 to 2.6, we provide the ratio $MSE(\text{vanilla})/MSE(\cdot)$, the MSE of the vanilla Monte Carlo estimate divided by the MSE for the current method, as a measure of statistical efficiency of the method relative

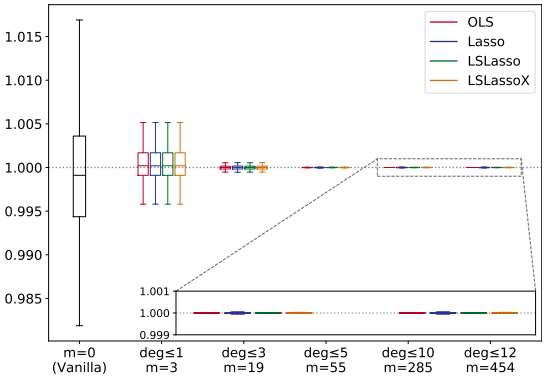
to naive Monte Carlo integration. The four tables correspond to the four panels (a) to (d) in Figure 2.1. For a given number of control variates m, the most efficient method is indicated in bold. For the Lasso-based methods, the results for the λ selection based on cross-validation (Algorithm 2.1) and dichotomic search (Algorithm 2.2) did not differ much; for the sake of brevity, the figures and the tables report the results associated to the dichotomic search.

Figures 2.1a and 2.1b highlight the success or failure of the OLS estimator depending on the size of m compared to n. In Figures 2.1c and 2.1d, we consider larger values of m and only compare the Lasso-based methods as it takes too much time to solve the OLS. In all our experiments, the LSLASSOX is the clear winner as it has the highest accuracy in almost all configurations. Moreover, the LSLASSOX can be computed much faster than the LSLASSO: in our implementation, preselecting the control variates based on a smaller subsample led to a reduction op the computation time by a factor between three and twenty.

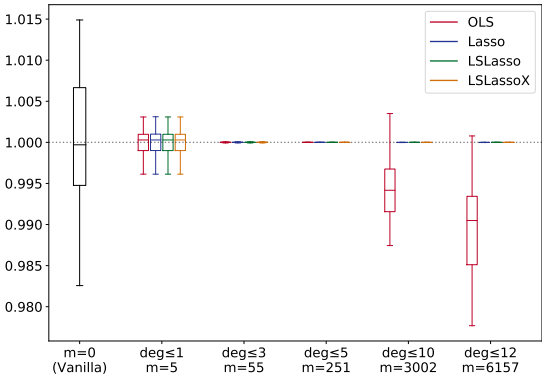
In Figure 2.1a, boxplots of the values returned by each of the methods are provided for φ in (2.14) when $d = 3$ and $n = 10\,000$. In this situation, where m is small compared to n, the OLS performs very well and the LSLASSO procedure selects almost all control variates so it performs as well as OLS. In Figure 2.1b, boxplots of the values returned by each of the methods are provided for g_3 in (2.16) when $d = 5$, $n = 2\,000$, and $N = 700$. In this case, the OLS estimator starts to break down as soon as the number, m, of control variates is of the same order as n. It is then necessary to perform some control variate selection, which is succesfully carried out by the LASSO and LSLASSO. Both of these estimators give the best results. Although the number of sample points used in the selection step of LSLASSOX has been reduced compared to the LSLASSO, the stability of the active set is barely affected. Accordingly, the error distributions for LSLASSO and LSLASSOX are quite similar.

Figures 2.1c and 2.1d reveal the benefits of selecting appropriate control variates before applying the OLS estimator. Figure 2.1c covers the function f_1 in (2.15) when $d = 5$ and $n = 5\,000$, while Figure 2.1d deals with the function g_4 in (2.16) when $d = 8$ and $n = 2\,000$. In the latter case, the number of control variates, $m = 36\,813$, is huge compared to the sample size $n = 2\,000$. However, the Lasso-based methods perform remarkably well in those settings. More precisely, in dimension $d = 5$ with the function f_1, the mean square error of the naive Monte Carlo estimator is of the order 10^{-5} whereas the one of the LSLASSOX is of the order 10^{-10}. Similarly, in dimension $d = 8$ with the function g_4, the mean square error goes down from 10^{-4} to 10^{-8}. Table 2.6 highlights the benefits of the LSLASSO over the LASSO in difficult situations.

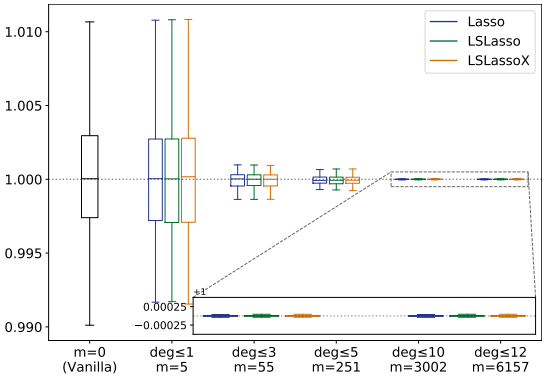
In the recent study South et al. (2022), the authors investigate the use of *regularization* in computing control variates estimates. They focus on the LASSO and ridge regression and they show, based on several examples, that the LASSO generally outperforms the ridge. In the applications they consider, they found that polynomials with relatively small degrees in each direction (k equal to 2 and 3) give the best performance. The examples considered here show a similar pattern as the results do not generally improve beyond degree $k = 3$.



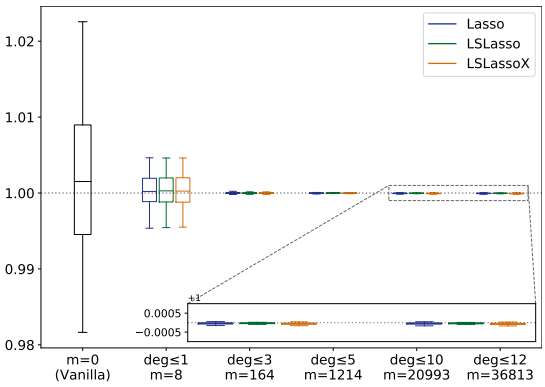
(a) Boxplots for φ in (2.14) with $d = 3$, $n = 10\,000$, $N = 2\,000$.



(b) Boxplots for g_3 in (2.16) with $d = 5$, $n = 2\,000$, $N = 700$.



(c) Boxplots for f_1 in (2.15) with $d = 5$, $n = 5\,000$, $N = 1\,000$.



(d) Boxplots for g_4 in (2.16) with $d = 8$, $n = 2\,000$, $N = 700$.

Figure 2.1 – Boxplots (based on 100 runs) of the values returned by each of the methods for functions φ, g_3, f_1, g_4 in (2.14)–(2.16).

2.5 Bayesian inference

In this section, we compare the different Monte Carlo estimates on Bayesian inference examples. Given some observed data x, the goal is to infer the parameter θ of a statistical model. We have some information through the prior distribution $\pi(\theta)$ and observe the model likelihood $\ell(x|\theta)$. Bayes' rule gives the posterior distribution as

$$p(\theta|x) = \frac{\ell(x|\theta)\pi(\theta)}{\int_\Theta \ell(x|\theta)\pi(\theta)\mathrm{d}\theta}.$$

The normalizing constant in the denominator is called evidence and is of interest for Bayesian model selection:

$$Z = \int_\Theta \ell(x|\theta)\pi(\theta)\mathrm{d}\theta.$$

Typically, this integral is analytically intractable. It is also difficult to compute numerically if the dimension d of the parameter space Θ is large.

We consider the same datasets as in (South et al., 2022): the European dipper capture-recapture data from (Marzolin, 1988) in Section 2.5.1 and the sonar data from (Gorman and Sejnowski, 1988) in Section 2.5.2. The dimensions of the integration domains are

$m =$	3	19	55	285	454
OLS	8.42e00	8.56e02	**2.12e05**	2.49e11	**5.27e14**
LASSO	8.42e00	8.53e02	6.72e04	7.71e04	7.71e04
LSL	8.42e00	**8.58e02**	2.10e05	6.26e05	1.37e06
LSLX	8.42e00	8.51e02	2.09e05	**2.49e11**	2.91e05
QMC	Halton: 8.76e01		Sobol: 3.29e02		

Table 2.3 – Statistical efficiency for φ; see also Figure 2.1a.

$m =$	5	55	251	3002	6157
OLS	2.45e01	5.75e04	7.48e08	1.42e00	4.94e-1
LASSO	2.45e01	5.75e04	4.19e06	4.83e05	4.31e05
LSL	2.45e01	5.75e04	**7.79e08**	**4.83e06**	**4.54e06**
LSLX	2.45e01	5.75e04	1.87e08	1.71e06	5.54e05
QMC	Halton: 3.75e00		Sobol: 1.57e01		

Table 2.4 – Statistical efficiency for g_3; see also Figure 2.1b.

$m =$	5	55	251	3002	6157
LASSO	1.11e00	**6.60e01**	**1.79e02**	8.17e04	8.56e04
LSL	1.11e00	6.59e01	1.76e02	6.77e04	6.83e04
LSLX	1.11e00	6.59e01	1.78e02	**8.97e04**	**9.24e04**
QMC	Halton: 4.60e00		Sobol: 7.21e01		

Table 2.5 – Statistical efficiency for f_1; see also Figure 2.1c.

$m =$	8	164	1214	20993	36813
LASSO	1.98e01	1.52e04	7.94e05	7.94e04	6.05e04
LSL	1.97e01	1.53e04	1.32e06	**1.49e05**	**1.28e05**
LSLX	1.98e01	**1.54e04**	**1.38e06**	1.98e04	1.55e04
QMC	Halton: 3.80e00		Sobol: 2.60e01		

Table 2.6 – Statistical efficiency for g_4; see also Figure 2.1d.

$d = 12$ and $d = 61$, respectively.

As in Section 2.4, we consider multivariate control functions based on univariate orthogonal polynomials by forming tensor products of the form $h_\ell(x_1, \dots, x_d) = \prod_{j=1}^{d} h_{\ell_j}(x_j)$, for a multi-index $\ell = (\ell_1, \dots, \ell_d)$ in $\{0, \dots, k\}^d \setminus \{(0, \dots, 0)\}$. In both examples, the dimension d is so large that considering all tensor products is infeasible. Instead, we focus on combinations where ℓ_j equals 0 for all but one or two coordinates, leading to a total number of $m = kd$ and $m = kd + k^2 d(d-1)/2$ control variates, respectively.

The different Monte Carlo estimates are compared on the basis of their mean squared errors (MSE). In contrast to Section 2.4, the true value of the integral is unknown. An estimate of this value, referred to as the gold standard $Z^\star$, is obtained by naive Monte Carlo with sample size $n = 10^8$. The variance of this estimate, computed on 20 independent runs, is smaller than the variance of all the other considered methods. The different boxplots of Figure 2.2 show the results obtained over 100 independent runs of $\hat{Z}/Z^\star$ where $\hat{Z}$ is the estimate of the evidence. Tables 2.8 to 2.11 provide numerical values for the statistical efficiency $\widehat{MSE}(\text{vanilla})/\widehat{MSE}(\cdot)$. We consider various settings and the parameter configuration is $n \in \{2\,000; 5\,000\}$ for the Monte Carlo sample size with $N \in \{700; 1\,000\}$ for the Monte Carlo subsample size for the LSLASSO. The regularization parameter λ is chosen via dichotomic search (Algorithm 2.2).

2.5.1 European dipper capture-recapture data

The data-set given in Table 2.7 was collected by (Marzolin, 1988) and describes the annual capture and recapture counts of the bird species *Cinclus cinclus*, also known as the European dipper, in eastern France from 1981 to 1987. We observe count data $x_{i,j}$ with $i \in \{1, \dots, I\}$ and $j \in \{i+1, \dots, J\}$, where $x_{i,j}$ denotes the number of birds released in year i and subsequently recaptured for the first time in year j. In the example, we have $I = 6$ and $J = 7$, where 1981 corresponds to year $i = 1$. Also observed is R_i, the number of marked birds released into the population in year i.

Release year	Birds released	Year of recapture: 1981 + ⋯					
		1	2	3	4	5	6
1981	22	11	2	0	0	0	0
1982	60		24	1	0	0	0
1983	78			34	2	0	0
1984	80				45	1	2
1985	88					51	0
1986	98						52

Table 2.7 – European dipper capture-recapture data (Marzolin, 1988). The counts in the triangle refer to the number of birds released in a given year and recaptured for the first time in a later year.

Following (Brooks et al., 2000; Nott et al., 2018; South et al., 2022), we consider a Bayesian approach for the Cormack–Jolly–Seber model (Lebreton et al., 1992). The model parameters are ϕ_i, a bird's survival probability from year i to $(i+1)$ for $i \in \{1, \dots, I\}$, together with p_j, the probability of a bird being recaptured in year $j \in \{2, \dots, J\}$. Let $\nu_{i,j}$ denote the probability that a bird captured and released in year i gets recaptured for the first time in year j. Since the bird must survive from year i to year j, not be recaptured in years $i+1$ to $j-1$ and then finally be recaptured in year j, the probability is modelled as

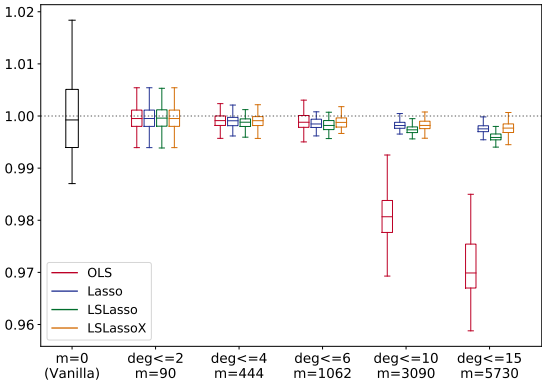
$$\nu_{i,j} = \phi_i p_j \prod_{k=i+1}^{j-1} [\phi_k (1 - p_k)].$$

The number of birds released at year i that are never recaptured at all is equal to $r_i = R_i - \sum_{j=i+1}^{J} x_{i,j}$ while the probability that a bird released in year i is never recaptured is $\chi_i = 1 - \sum_{j=i+1}^{J} \nu_{i,j}$. The resulting likelihood is equal to

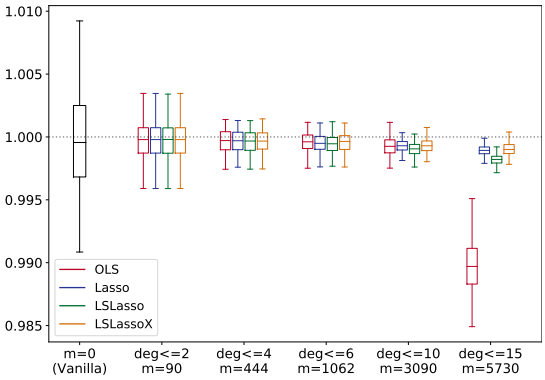
$$\ell(x|\theta) = \prod_{i=1}^{I} \left\{ \chi_i^{r_i} \prod_{j=i+1}^{J} \nu_{i,j}^{x_{i,j}} \right\},$$

where $\theta = (\phi_1, \dots, \phi_6, p_2, \dots, p_7) \in [0,1]^{12}$. The uniform distribution is chosen as prior and we use tensor products of Legendre polynomials with $k = 10$ (Example 2.29) as controls.

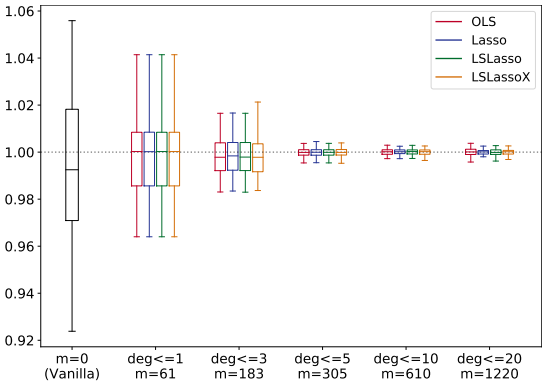
The results for the various integration methos are reported in the same way as in Section 2.4. The boxplots and statistical efficiencies are given in Figures 2.2a and 2.2b and Tables 2.8 and 2.9 respectively. Similarly to the synthetic data, Figures 2.2a and 2.2b reveal the success or failure of the OLS on the capture-recapture data when the number of control variates m is larger than the Monte Carlo sample size n. The variance goes down as m increases. Tables 2.8 and 2.9 show that for $n = 2\,000$, the OLS estimate gives the best performance whereas for $n = 5\,000$, the LASSO-based methods profit from the large number of available control variates. In this case, the LASSO is most efficient while the LSLASSOX performs similarly but at a reduced computing time.



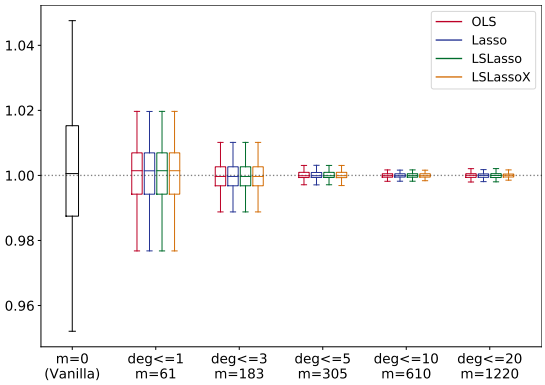
(a) Boxplots for Capture dataset ($n = 2000,\ N = 700$)



(b) Boxplots for Capture dataset ($n = 5000,\ N = 1000$)



(c) Boxplots for Sonar dataset ($n = 2000,\ N = 700$)



(d) Boxplots for Sonar dataset ($n = 5000,\ N = 1000$)

Figure 2.2 – Boxplots (based on 100 runs) of $\hat{Z}/Z^\star$ returned by each of the methods for Capture-Recapture and Sonar examples.

2.5.2 Sonar data

The data were collected by (Gorman and Sejnowski, 1988) and are available from the UCI Machine Learning Repository (Asuncion and Newman, 2007). The data matrix X represents 208 sonar signals, each one composed of 60 attributes within the binary classification framework. A column of 1's is added to the matrix X to represent the intercept so that $X \in \mathbb{R}^{208\times 61}$. The goal is to assess whether the sonar signal bounces off a metal cylinder (label $y = 1$) or a roughly cylindrical rock (label $y = -1$). The different covariates represent the energy within particular frequency bands, integrated over a certain period of time. Using the encoding $y \in \{-1, +1\}$ and following a logistic regression model, the resulting log-likelihood is

$$\log \ell(X, y|\theta) = -\sum_{i=1}^{208} \log\left\{1 + \exp\left(-y_i \sum_{j=1}^{61} X_{ij}\theta_j\right)\right\},$$

where the model coefficient $\theta \in [-1, 1]^{61}$ has a uniform prior distribution. We use the family of Legendre polynomials as control functions with $k = 20$. The boxplots and statistical efficiencies are presented in Figures 2.2c and 2.2d and Tables 2.10 and 2.11,

$m =$	90	444	1 062	3 090	5 730
OLS	9.33	**20.7**	14.7	0.14	0.06
LASSO	9.34	20.3	16.7	**14.4**	**8.57**
LSL	9.33	20.4	12.8	8.43	4.60
LSLX	9.33	19.4	**19.8**	12.9	7.86

Table 2.8 – Capture data: statistical efficiency ($n = 2000$)

$m =$	90	444	1 062	3 090	5 730
OLS	7.67	18.1	22.1	15.2	0.15
LASSO	7.67	**18.4**	**22.3**	**22.8**	12.8
LSL	7.67	18.0	21.3	13.3	5.24
LSLX	7.67	17.8	21.4	21.6	**13.2**

Table 2.9 – Capture data: statistical efficiency ($n = 5000$)

$m =$	61	183	305	610	1220
OLS	3.39	13.3	246	548	330
LASSO	3.39	13.6	**250**	**673**	680
LSL	3.39	13.3	246	564	499
LSLX	3.39	**13.9**	244	558	**680**

Table 2.10 – Sonar data: statistical efficiency ($n = 2000$)

$m =$	61	183	305	610	1220
OLS	4.48	17.0	235	801	601
LASSO	4.49	17.0	**240**	821	721
LSL	4.48	17.0	235	804	629
LSLX	4.48	17.0	241	**833**	**734**

Table 2.11 – Sonar data: statistical efficiency ($n = 5000$)

respectively. Once again, the Lasso-based methods, with their selection strategy, are able to benefit from a larger control variates space. The winner of this competition is LSLASSOX as it offers the best performance combined with a smaller computation time compared to the LSLASSO.

2.6 Conclusion and perspective

The use of high-dimensional control variates with the help of a LASSO-type procedure has been shown to be efficient in order to reduce the variance of the basic Monte Carlo estimate. The method, called LSLASSO(X), that first selects appropriate control variates by the LASSO, possibly on a smaller subsample, and then estimates the control variate coefficients by least squares performs excellently considering the modest computing time required. Several avenues for further research are now discussed.

The construction of control variates by a change of measure (Remark 2.1) presupposes some knowledge on the underlying integration measure in order to choose an appropriate sampling distribution. For instance, if the support of the sampling measure does not cover the whole integration domain then the method will certainly fail. *Adaptive importance sampling* (see, e.g., (Owen and Zhou, 2000; Portier and Delyon, 2018)) offers a possible solution, involving online estimates of the appropriate sampling policy and the optimal linear combination of control variates.

Assumption 2.10 on the sub-Gaussianity of the residuals is key to obtain concentration inequalities. For certain applications, it might be too restrictive, however. In the absence of such an assumption or more generally of suitable bounds on the tails of the residual distribution, other types of results such as almost sure convergence rates might still be pursued.

In the random design setting, the estimators of coefficient vector $\beta^\star(f)$ are all biased, even the OLS estimator. The bias may be removed by sample splitting (Avramidis and Wilson, 1993), but at the cost of an increased variance, especially if the number of control variates is large. For the Lasso-based methods, debiasing methods are studied in (Javanmard and Montanari, 2018) and the references therein. The merits of these techniques for Monte Carlo control variate methods remain to be investigated.

We have presented different control variate methods from the point of view of estimation only. Equally important questions are that of model evaluation and Monte Carlo sample size calculation, assessing the accuracy of the estimate. Several ways can be imagined such as sample splitting (e.g., cross-validation) and plug-in estimation of the residual variance $\sigma^2(f)$, using for instance the estimated residuals.

2.A Proofs

2.A.1 Auxiliary results

Lemma 2.30. *(Sub-Gaussian) Let $X_1, \dots, X_n$ be independent and identically distributed random variables in $(\mathcal{X}, \mathcal{A})$ with distribution π. Let $\varphi_1, \dots, \varphi_p$ be real-valued functions on $\mathcal{X}$ such that $\pi(\varphi_k) = 0$ and $\varphi_k \in \mathcal{G}(\tau^2)$ for all $k = 1, \dots, p$. Then for all $\delta > 0$, we have with probability at least $1 - \delta$,*

$$\max_{k=1,\dots,p} \left| \sum_{i=1}^{n} \varphi_k(X_i) \right| \leq \sqrt{2n\tau^2 \log(2p/\delta)}.$$

Proof For each $k = 1, \dots, p$, the centered random variable $\sum_{i=1}^n \varphi_k(X_i)$ is sub-Gaussian with variance factor $n\tau^2$. By the union bound and by Chernoff's inequality, we have, for each $t > 0$,

$$\begin{aligned} \mathbb{P}\left(\max_{k=1,\dots,p} \left| \sum_{i=1}^{n} \varphi_k(X_i) \right| > t \right) &\leq \sum_{k=1}^{p} \mathbb{P}\left(\left| \sum_{i=1}^{n} \varphi_k(X_i) \right| > t \right) \\ &\leq 2p \exp\left(\frac{-t^2}{2n\tau^2} \right). \end{aligned}$$

Set $t = \sqrt{2n\tau^2 \log(2p/\delta)}$ to find the result. ∎

Lemma 2.31. *(Smallest eigenvalue lower bound) Let $X_1, \dots, X_n$ be independent and identically distributed random variables in $(\mathcal{X}, \mathcal{A})$ with distribution π. Let $g = (g_1, \dots, g_p)^\top$ in $L_2(P)^p$ be such that the $p \times p$ Gram matrix $G = \pi(gg^\top)$ satisfies $\lambda_{\min}(G) > 0$. Define the transformation $\tilde{g} = G^{-1/2} g$ and put $B_{\tilde{g}} := \sup_{x \in \mathcal{X}} \left\| \tilde{g}(x) \right\|_2^2$. Let $\delta, \eta \in (0, 1)$. For $\delta \in (0, 1)$, the empirical Gram matrix $\hat{G}_n = \pi_n(gg^\top)$ satisfies, with probability at least $1 - \delta$,*

$$\lambda_{\min}(\hat{G}_n) > \left(1 - \sqrt{2B_{\tilde{g}} n^{-1} \log(p/\delta)} \right) \lambda_{\min}(G).$$

Proof Suppose that the result is true in the special case that G is the identity matrix. In case of a general Gram matrix G, we could then apply the result for the special case to the vector of functions $\tilde{g} = G^{-1/2}g$, whose Gram matrix is the identity matrix. We would get that $\lambda_{\min}(\pi_n(\tilde{g}\tilde{g}^\top)) > 1 - \eta$ with probability at least $1 - \delta$. Since $\pi_n(\tilde{g}\tilde{g}^\top) = G^{-1/2}\hat{G}_n G^{-1/2}$ and since $u^\top G^{-1} u \leq 1/\lambda_{\min}(G)$ for every unit vector $u \in \mathbb{R}^p$, we would have

$$\begin{aligned}
\lambda_{\min}\left(\pi_n(\tilde{g}\tilde{g}^\top)\right) &= \min_{u^\top u = 1} \left\{ u^\top \pi_n(\tilde{g}\tilde{g}^\top) u \right\} \\
&= \min_{u^\top u = 1} \left\{ \frac{(G^{-1/2}u)^\top \hat{G}_n G^{-1/2} u}{(G^{-1/2}u)^\top G^{-1/2} u} u^\top G^{-1} u \right\} \\
&\leq \lambda_{\min}(\hat{G}_n)/\lambda_{\min}(G).
\end{aligned}$$

It would then follow that

$$\lambda_{\min}(\hat{G}_n) \geq \lambda_{\min}(\pi_n(\tilde{g}\tilde{g}^\top))\, \lambda_{\min}(G) \geq (1 - \eta)\lambda_{\min}(G),$$

as required. Hence we only need to show the result for $G = I$, in which case $\tilde{g} = g$.

We apply the matrix Chernoff inequality in (Tropp, 2015, Theorem 5.1.1) to the random matrices $n^{-1}g(X_i)g(X_i)^\top$. These matrices are independent and symmetric with dimension $p \times p$. Their minimum and maximum eigenvalues are between 0 and $L = B_g/n$, with $B_g = \sup_{x \in \mathcal{X}} \lambda_{\max}(g(x)g(x)^\top) = \sup_{x \in \mathcal{X}} ||g(x)||_2^2$. Their sum is equal to $\pi_n(gg^\top) = \hat{G}_n$, whose expectation is $G = I$ by assumption. In the notation of the cited theorem, we have $\mu_{\min} = \lambda_{\min}(G) = 1$, and thus, by Eq. (5.1.5) in that theorem, we have, for $\eta \in [0, 1)$,

$$\mathbb{P}\{\lambda_{\min}(\hat{G}_n) \leq 1 - \eta\} \leq p \left[\frac{\exp(-\eta)}{(1 - \eta)^{1-\eta}} \right]^{n/B_g}.$$

The term in square brackets is bounded above by $\exp(-\eta^2/2)$. Indeed, we have, for $\eta \in [0, 1)$,

$$\frac{e^{-\eta}}{(1 - \eta)^{1-\eta}} = \exp\{-\eta - (1 - \eta)\log(1 - \eta)\}$$

and

$$\begin{aligned}
\eta + (1 - \eta)\log(1 - \eta) &= \eta - (1 - \eta)\int_0^\eta \frac{\mathrm{d}t}{1 - t} \\
&= \int_0^\eta \left(1 - \frac{1 - \eta}{1 - t}\right) \mathrm{d}t \\
&= \int_0^\eta \frac{\eta - t}{1 - t}\, \mathrm{d}t \\
&\geq \int_0^\eta (\eta - t)\, \mathrm{d}t = \frac{\eta^2}{2}.
\end{aligned}$$

It follows that

$$\mathbb{P}\{\lambda_{\min}(\hat{G}_n) \leq 1 - \eta\} \leq p \exp\left(-\frac{\eta^2 n}{2B_g}\right).$$

Solving $p \exp\left(-\frac{\eta^2 n}{2B_g}\right) = \delta$ in η, we find that, with probability at least $1-\delta$,

$$\lambda_{\min}(\hat{G}_n) > 1 - \sqrt{2B_g n^{-1} \log(p/\delta)}.$$

■

Lemma 2.32 (Upper bound of moments)**.** *Let X be a random variable such that $\mathbb{E}(|X|^{2p}) \leq 2^{p+1}p!$ for every integer $p \geq 1$. Then*

$$\forall \lambda \in \mathbb{R}, \qquad 1 + \sum_{k=2}^{\infty} \frac{\lambda^k}{k!} \mathbb{E}(|X|^k) \leq \exp(9\lambda^2/4), \tag{2.17}$$

in which it is implicitly understood that the series on the left-hand side converges.

Proof

Let $\lambda \in \mathbb{R}$. We split the series in terms with even and odd indices k, leading to

$$1 + \sum_{k=2}^{\infty} \frac{\lambda^k}{k!} \mathbb{E}(|X|^k) = 1 + \sum_{p=1}^{\infty} \frac{\lambda^{2p}}{(2p)!} \mathbb{E}(|X|^{2p}) + \sum_{p=1}^{\infty} \frac{\lambda^{2p+1}}{(2p+1)!} \mathbb{E}(|X|^{2p+1}).$$

We will bound the series on the odd indices in terms of the series on the even indices.

Since the geometric mean of two nonnegative numbers is bounded by their arithmetic mean, we have, for all $x \geq 0$ and all $a > 0$,

$$|x| = \sqrt{\frac{1}{a} \cdot ax^2} \leq \frac{1}{2}\left(\frac{1}{a} + ax^2\right).$$

Applying the previous inequality to $x = \lambda X$ and scalars $a_p > 0$ to be chosen later,

$$\begin{aligned}
\sum_{p=1}^{\infty} \frac{\lambda^{2p+1}}{(2p+1)!} \mathbb{E}(|X|^{2p+1}) &\leq \sum_{p=1}^{\infty} \frac{\lambda^{2p}}{(2p+1)!} \mathbb{E}\left[|X|^{2p} \frac{1}{2}\left(\frac{1}{a_p} + a_p(\lambda X)^2\right)\right] \\
&= \sum_{p=1}^{\infty} \frac{\lambda^{2p}}{2a_p} \frac{\mathbb{E}(|X|^{2p})}{(2p+1)!} + \sum_{p=1}^{\infty} \frac{a_p}{2} \frac{\lambda^{2p+2}}{(2p+1)!} \mathbb{E}(|X|^{2p+2}) \\
&= \sum_{p=1}^{\infty} \frac{\lambda^{2p}}{2a_p} \frac{\mathbb{E}(|X|^{2p})}{(2p+1)!} + \sum_{p=2}^{\infty} \frac{a_{p-1}}{2} \frac{\lambda^{2p}}{(2p-1)!} \mathbb{E}(|X|^{2p}) \\
&= \sum_{p=1}^{\infty} \left(\frac{1}{2a_p(2p+1)} + p a_{p-1} \mathbb{1}_{\{p \geq 2\}}\right) \frac{\lambda^{2p}}{(2p)!} \mathbb{E}(|X|^{2p}).
\end{aligned}$$

Here, $\mathbb{1}$ denotes an indicator function. We obtain

$$\sum_{k=2}^{\infty} \frac{\lambda^k}{k!} \mathbb{E}(|X|^k) \leq \sum_{p=1}^{\infty} \left(1 + \frac{1}{2a_p(2p+1)} + p a_{p-1} \mathbb{1}_{\{p \geq 2\}}\right) \frac{\lambda^{2p}}{(2p)!} \mathbb{E}(|X|^{2p}).$$

Define $b_p = a_p(2p+1)$ and use the hypothesis on $\mathbb{E}(|X|^{2p})$ to see that, for any constants $b_p > 0$,

$$1 + \sum_{k=2}^{\infty} \frac{\lambda^k}{k!}\mathbb{E}(|X|^k) \le 1 + \sum_{p=1}^{\infty} \left(1 + \frac{1}{2b_p} + \frac{p}{2p-1} b_{p-1} \mathbb{1}_{\{p\ge 2\}}\right) \frac{\lambda^{2p}}{(2p)!} 2^{p+1} p!.$$

The objective is to find a constant $c > 0$ as small as possible and such that the right-hand side is bounded by $\exp(c\lambda^2) = 1 + \sum_{p=1}^{\infty} c^p \lambda^{2p}/p!$. Comparing coefficients, this means that we need to determine scalars $b_p > 0$ and $c > 0$ in such a way that for all $p = 1, 2, \ldots$

$$\left(1 + \frac{1}{2b_p} + \frac{p}{2p-1} b_{p-1} \mathbb{1}_{\{p\ge 2\}}\right) \frac{2^{p+1} p!}{(2p)!} \le \frac{c^p}{p!},$$

or, equivalently,

$$\left(2 + \frac{1}{b_p} + \frac{2p}{2p-1} b_{p-1} \mathbb{1}_{\{p\ge 2\}}\right) \prod_{j=1}^{p} \frac{j}{p+j} \le (c/2)^p.$$

The case $p = 1$ gives

$$2 + \frac{1}{b_1} \le c, \tag{2.18}$$

showing that, with this proof technique, we will always find $c > 2$. Setting $b_p \equiv b > 0$ for all integer $p \ge 1$ and $c = 2 + 1/b$, inequality (2.18) is automatically satisfied, so it remains to find $b > 0$ such that forall $p = 2, 3, \ldots$

$$\left(2 + \frac{1}{b} + \frac{2p}{2p-1} b\right) \prod_{j=1}^{p} \frac{j}{p+j} \le (c/2)^p \quad \text{with } c = 2 + \frac{1}{b}.$$

The left-hand side is decreasing in p whereas the right-hand side is increasing in p. It is thus sufficient to have the inequality satisfied for $p = 2$, i.e.,

$$\left(2 + \frac{1}{b} + \frac{4b}{3}\right) \frac{1}{6} \le \left(1 + \frac{1}{2b}\right)^2. \tag{2.19}$$

Equating both sides leads to a nonlinear equation in b that can be solved numerically, giving the root $b \approx 4.006156$. With $b = 4$, inequality (2.19) is satisfied, as can be checked directly ($91/72 \le 81/64$). We conclude that $c = 2 + 1/4 = 9/4$ is a valid choice. ■

Note that the series in (2.17) starts at $k = 2$. If also $\mathbb{E}(X) = 0$, the left-hand side in (2.17) is an upper bound for $\mathbb{E}(\exp(\lambda X))$, and we obtain the following corollary.

Corollary 2.33. *Let Z be a centered random variable such that*

$$\forall p \in \mathbb{N}^\star, \quad \mathbb{E}(|Z|^{2p}) \le 2^{p+1} p!$$

Then $\log \mathbb{E}(\exp(\lambda Z)) \le 9\lambda^2/4$ for all $\lambda \in \mathbb{R}$, i.e., $Z \in \mathcal{G}(9/2)$.

Lemma 2.34. *Let (X, Y) be a pair of uncorrelated random variables. If $X \in \mathcal{G}(\nu)$ and $|Y| \le \kappa$ for some $\nu > 0$ and $\kappa > 0$, then $XY \in \mathcal{G}((9/2)\kappa^2\nu)$.*

Proof The random variable $X/\sqrt{\nu}$ is sub-Gaussian with variance factor 1. As on page 25 in Boucheron et al. (2013a), this implies that $\mathbb{P}(|X/\sqrt{\nu}| > t) \leq 2\exp(-t^2/2)$ for all $t \geq 0$ and thus $\mathbb{E}[|X/\sqrt{\nu}|^{2p}] \leq 2^{p+1}p!$ for all integer $p \geq 1$ (see (Boucheron et al., 2013a, Theorem 2.1)).

Let $Z = XY/(\sqrt{\nu}\kappa)$. Since X is centered and X and Y are uncorrelated, XY is centred too, and therefore also Z. From the previous paragraph, we have $\mathbb{E}(|Z|^{2p}) \leq \mathbb{E}(|X/\sqrt{\nu}|^{2p}) \leq 2^{p+1}p!$ for all integer $p \geq 1$. Corollary 2.33 gives for all $\lambda \in \mathbb{R}$ that $\log \mathbb{E}(\exp(\lambda Z)) \leq 9\lambda^2/4$, from which

$$\log \mathbb{E}(\exp(\lambda XY)) = \log \mathbb{E}(\exp(\lambda\sqrt{\nu}\kappa Z)) \leq \frac{9}{4}\lambda^2\nu\kappa^2.$$

■

Lemma 2.35 (Upper bound for norm-subGaussian random vector)**.** *Let X be a d-dimensional random vector with zero-mean and such that $\mathbb{P}(\|X\|_2 \geq t) \leq 2\exp\left(-t^2/(2\sigma^2)\right)$ for all $t \geq 0$. Then the random matrix Y defined by*

$$Y = \begin{bmatrix} 0 & X^\top \\ X & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \tag{2.20}$$

satisfies $\mathbb{E}(\exp(\theta Y)) \preceq \exp(c\theta^2\sigma^2)I$ for any $\theta \in \mathbb{R}$, with $c = 9/4$, where I denotes the identity matrix.

Proof The non-zero eigenvalues of Y are $\|X\|$ and $-\|X\|$. The non-zero eigenvalues of Y^k are thus $\|X\|^k$ and $(-\|X\|)^k$ for integer $k \geq 1$. It follows that $Y^k \preceq \|X\|^k I$ for all integer $k \geq 1$, and therefore also $\mathbb{E}(Y^k) \preceq \mathbb{E}(\|X\|^k)I$ for all integer $k \geq 1$. Furthermore, the operator norm of Y^k is bounded by $\left\|Y^k\right\| \leq \|X\|^k$.

Since $\mathbb{E}(Y) = 0$, we get, for any $\theta \in \mathbb{R}$,

$$\mathbb{E}(\exp(\theta Y)) = I + \sum_{k=2}^{\infty} \frac{\theta^k}{k!}\mathbb{E}(Y^k) \preceq \left(1 + \sum_{k=2}^{\infty} \frac{\theta^k}{k!}\mathbb{E}(\|X\|^k)\right) I = \left(1 + \sum_{k=2}^{\infty} \frac{(\theta\sigma)^k}{k!}\mathbb{E}(\xi^k)\right) I,$$

where $\xi = \|X\|/\sigma$. The first series converges in operator norm since $\left\|\mathbb{E}(Y^k)\right\| \leq \mathbb{E}(\left\|Y^k\right\|) \leq \mathbb{E}(\|X\|^k)$.

By assumption, $\mathbb{P}(\xi > t) = \mathbb{P}(\|X\| \geq \sigma t) \leq 2e^{-t^2/2}$ for all $t \geq 0$ and thus $\mathbb{E}(|\xi|^{2p}) \leq 2^{p+1}p!$ for all integer $p \geq 1$ But then we can apply Lemma 2.32 with $\lambda = \theta\sigma$ and $X = \xi$, completing the proof. ■

The following result is a special case of Jin et al. (2019, Corollary 7). Our contribution is to make the constant c in the cited result explicit. In passing, we correct an inaccuracy in the proof of Jin et al. (2019, Lemma 4), in which it was incorrectly claimed that the odd moments of a certain random matrix Y as in our Lemma 2.35 are all zero.

Lemma 2.36 (Hoeffding inequality for norm-subGaussian random vectors)**.** *Let the d-dimensional random vectors $Z_1, \dots, Z_n$ be independent, have mean zero, and satisfy*

$$\forall t \geq 0, \forall i = 1, \dots, n, \quad \mathbb{P}(\left\|Z_i\right\|_2 \geq t) \leq 2\exp\left(-\tfrac{t^2}{2\sigma^2}\right) \tag{2.21}$$

for some $\sigma > 0$. Then for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\left\|\sum_{i=1}^n Z_i\right\|_2 \leq 3\sqrt{n\sigma^2 \log(2d/\delta)}.$$

Proof Given Corollary 7 in Jin et al. (2019), the only thing to prove is that their constant can be set equal to 3. Their Corollary 7 follows from their Lemma 6 in which it is shown that when the matrix Y defined in (2.20) satisfies

$$\forall \theta \in \mathbb{R}, \qquad \mathbb{E}[\exp(\theta Y)] \preceq \exp(c\theta^2\sigma^2)I,$$

then we have for any $\theta > 0$, with probability at least $(1 - \delta)$,

$$\left\|\sum_{i=1}^n Z_i\right\|_2 \leq c \cdot \theta n\sigma^2 + \frac{1}{\theta}\log(2d/\delta).$$

Taking $\theta = \sqrt{\log(2d/\delta)/(cn\sigma^2)}$ yields

$$\left\|\sum_{i=1}^n Z_i\right\|_2 \leq 2\sqrt{c}\sqrt{n\sigma^2 \log(2d/\delta)},$$

and we conclude with Lemma 2.35 ($c = 9/4, 2\sqrt{c} = 3$). ■

2.A.2 Proof of Theorem 2.13

The proof is organized as follows. We first provide an upper bound on the error (Step 1). This bound involves the norm of the error made on the rescaled coefficients and is controlled in Step 2. Then (Step 3), we construct an event that has probability at least $1 - \delta$ on which we can control the terms that appear in the upper bound of Step 2. Collecting all the inequalities, we will arrive at the stated bound (Step 4).

Step 1. — Since $f = \pi(f) + \beta^\star(f)^\top h + \epsilon$, the oracle estimate of $\pi(f)$, which uses the unknown, optimal coefficient vector $\beta^\star(f)$, is

$$\hat{\alpha}_n^{\text{or}}(f) = \pi_n[f - \beta^\star(f)^\top h] = \pi(f) + \pi_n(\epsilon).$$

The difference between the OLS and oracle estimates is

$$\hat{\alpha}_n^{\text{ols}}(f) - \hat{\alpha}_n^{\text{or}}(f) = \left(\beta^\star(f) - \hat{\beta}_n^{\text{ols}}(f)\right)^\top \pi_n(h).$$

Let $G = \pi(hh^\top)$ be the $m \times m$ Gram matrix. By assumption, G is positive definite. Write

$$\eta^\star = G^{1/2}\beta^\star(f), \qquad \hat{\eta} = G^{1/2}\hat{\beta}_n^{\text{ols}}(f), \qquad \hbar = G^{-1/2}h.$$

The estimation error of the OLS estimator can thus be decomposed as

$$\begin{aligned} n\Big(\hat{\alpha}_n^{\mathrm{ols}}(f) - \pi(f)\Big) &= n\Big(\hat{\alpha}_n^{\mathrm{or}}(f) - \pi(f)\Big) + \Big(\beta^\star(f) - \hat{\beta}_n^{\mathrm{ols}}(f)\Big)^\top n\,\pi_n(h) \\ &= \sum_{i=1}^n \epsilon(X_i) + \Big(\beta^*(f) - \hat{\beta}_n^{\mathrm{ols}}(f)\Big)^\top \sum_{i=1}^n h(X_i) \\ &= \sum_{i=1}^n \epsilon(X_i) + (\eta^\star - \hat{\eta})^\top \sum_{i=1}^n \hbar(X_i). \end{aligned}$$

By the triangle and Cauchy–Schwarz inequalities,

$$n\left|\hat{\alpha}_n^{\mathrm{ols}}(f) - P(f)\right| \leq \left|\sum_{i=1}^n \epsilon(X_i)\right| + \left\|\eta^\star - \hat{\eta}\right\|_2 \left\|\textstyle\sum_{i=1}^n \hbar(X_i)\right\|_2. \tag{2.22}$$

Step 2. — We will show that, if $\lambda_{\min}(\pi_n(\hbar\hbar^\top)) > \left\|\pi_n(\hbar)\right\|_2^2$, then

$$\left\|\hat{\eta} - \eta^\star\right\|_2 \leq \frac{\left\|\pi_n(\hbar\epsilon)\right\|_2 + \left\|\pi_n(\hbar)\right\|_2 \left|\pi_n(\epsilon)\right|}{\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\|\pi_n(\hbar)\right\|_2^2}. \tag{2.23}$$

and thus, by (2.22),

$$\left|\hat{\alpha}_n^{\mathrm{ols}}(f) - \pi(f)\right| \leq \left|\pi_n(\epsilon)\right| + \frac{\left\|\pi_n(\hbar\epsilon)\right\|_2 + \left\|\pi_n(\hbar)\right\|_2 \left|\pi_n(\epsilon)\right|}{\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\|\pi_n(\hbar)\right\|_2^2} \left\|\pi_n(\hbar)\right\|_2 \tag{2.24}$$

Step 2.1 — Considered the column-centered $n \times m$ design matrices

$$\begin{aligned} H_c &= H - \mathbb{1}_n \pi_n(h)^\top = \Big(h_j(X_i) - \pi_n(h_j)\Big)_{i,j}, \\ \bar{H}_c &= H_c G^{-1/2} = \bar{H} - \mathbb{1}_n \pi_n(\hbar)^\top = \Big(\hbar_j(X_i) - \pi_n(\hbar_j)\Big)_{i,j}. \end{aligned}$$

Since $\bar{H}^\top \mathbb{1}_n = n\pi_n(\hbar)$, we have

$$\begin{aligned} \bar{H}_c^\top \bar{H}_c &= \bar{H}^\top \bar{H} - n\pi_n(\hbar)\pi_n(\hbar)^\top \\ &= n\left(\pi_n(\hbar\hbar^\top) - \pi_n(\hbar)\pi_n(\hbar)^\top\right). \end{aligned}$$

As a consequence, for $u \in \mathbb{R}^m$,

$$\begin{aligned} u^\top \bar{H}_c^\top \bar{H}_c u &= n\left(u^\top \pi_n(\hbar\hbar^\top) u - (\pi_n(\hbar)^\top u)^2\right) \\ &\geq n\left(\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\|\pi_n(\hbar)\right\|_2^2\right) \|u\|_2^2. \end{aligned}$$

by the Cauchy–Schwarz inequality. In particular, $u^\top \bar{H}_c^\top \bar{H}_c u$ is non-zero for non-zero $u \in \mathbb{R}^m$, so that $\bar{H}_c^\top \bar{H}_c$ is invertible, and so is the matrix

$$H_c^\top H_c = G^{1/2} \bar{H}_c \bar{H}_c G^{1/2}.$$

Also, the smallest eigenvalue of $\bar{H}_c^\top \bar{H}_c$ is bounded from below by

$$\lambda_{\min}(\bar{H}_c^\top \bar{H}_c) \geq n \left(\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\| \pi_n(\hbar) \right\|_2^2 \right) > 0.$$

The largest eigenvalue of the inverse matrix $(\bar{H}_c^\top \bar{H}_c)^{-1}$ is then bounded from above by

$$\lambda_{\max}\left((\bar{H}_c^\top \bar{H}_c)^{-1} \right) \leq \frac{1}{n \left(\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\| \pi_n(\hbar) \right\|_2^2 \right)}. \tag{2.25}$$

Step 2.2. — Write $\epsilon_c^{(n)} = (\epsilon(X_i) - \pi_n(\epsilon))_{i=1}^n$ for the centered vector of error terms. Recall $f_c^{(n)} = (f(X_i) - \pi_n(f))_{i=1}^n$, the centered vector of samples from the integrand. As $f = \pi(f) + h^\top \beta^*(f) + \epsilon$, we have

$$f_c^{(n)} = H_c \beta^\star(f) + \epsilon_c^{(n)}.$$

From the characterization (2.4) of the OLS estimate of the coefficient vector and since $H_c^\top H_c$ is invertible,

$$\begin{aligned} \hat{\beta}_n^{\text{ols}}(f) &= (H_c^\top H_c)^{-1} H_c^\top f_c^{(n)} \\ &= (H_c^\top H_c)^{-1} H_c^\top \left(H_c \beta^\star(f) + \epsilon_c^{(n)} \right) \\ &= \beta^\star(f) + (H_c^\top H_c)^{-1} H_c^\top \epsilon_c^{(n)}. \end{aligned}$$

We obtain

$$\begin{aligned} \hat{\eta} - \eta^\star &= G^{1/2} \left(\hat{\beta}_n^{\text{ols}}(f) - \beta^\star(f) \right) \\ &= G^{1/2} (H_c^\top H_c)^{-1} H_c^\top \epsilon_c^{(n)} \\ &= (\bar{H}_c^\top \bar{H}_c)^{-1} \bar{H}_c^\top \epsilon_c^{(n)}. \end{aligned} \tag{2.26}$$

Step 2.3. — We combine the results from Steps 2.1 and 2.2. From the upper bound (2.25) and the identity (2.26), we obtain

$$\left\| \hat{\eta} - \eta^\star \right\|_2 \leq \frac{\left\| \bar{H}_c^\top \epsilon_c^{(n)} \right\|_2}{n \left(\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\| \pi_n(\hbar) \right\|_2^2 \right)}$$

Finally, as $\bar{H}_c = (\hbar_j(X_i) - \pi_n(\hbar_j))_{i,j}$, we find

$$\begin{aligned} n^{-1} \left\| \bar{H}_c^\top \epsilon_c^{(n)} \right\|_2 &= n^{-1} \left\| \textstyle\sum_{i=1}^n \hbar(X_i)\epsilon(X_i) - \pi_n(\hbar) \sum_{i=1}^n \epsilon(X_i) \right\|_2 \\ &= \left\| \pi_n(\hbar\epsilon) - \pi_n(\hbar)\pi_n(\epsilon) \right\|_2 \\ &\leq \left\| \pi_n(\hbar\epsilon) \right\|_2 + \left\| \pi_n(\hbar) \right\|_2 \left| \pi_n(\epsilon) \right|. \end{aligned}$$

Equation (2.23) follows.

Step 3. — In view of (2.24), we need to ensure that $|\pi_n(\epsilon)|$, $||\pi_n(\hbar)||_2$ and $||\pi_n(\hbar\epsilon)||_2$ are small and that $\lambda_{\min}(\pi_n(\hbar\hbar^\top))$ is large. Let $\delta > 0$. We construct an event with probability at least $1-\delta$ on which four inequalities hold simultaneously. Recall $B = \sup_{x\in\mathcal{X}} \left\|\hbar(x)\right\|_2^2$, defined in (2.7).

Step 3.1. — Because $\epsilon \in \mathcal{G}(\tau^2)$, Chernoff's inequality (or Lemma 2.30 with $p = 1$) implies that with probability at least $1 - \delta/4$,

$$\left|\sum_{i=1}^n \epsilon(X_i)\right| \leq \sqrt{2n\tau^2 \log(8/\delta)}. \tag{2.27}$$

Step 3.2. — For the term $\left\|\sum_{i=1}^n \hbar(X_i)\right\|_2$, we apply the vector Bernstein bound in (Hsu et al., 2012, Lemma 31). On the one hand $\sup_{x\in\mathcal{X}} \left\|\hbar(x)\right\|_2 \leq \sqrt{B}$ and on the other hand

$$\sum_{i=1}^n \mathbb{E}[||\hbar(X_i)||_2^2] = \sum_{i=1}^n \sum_{j=1}^m \pi(\hbar_j^2) = nm.$$

The cited vector Bernstein bound gives

$$\forall t \geq 0, \mathbb{P}\left[\left\|\sum_{i=1}^n \hbar(X_i)\right\|_2 > \sqrt{nm}\left(1+\sqrt{8t}\right) + \tfrac{4}{3}t\sqrt{B}\right] \leq e^{-t}.$$

Setting $t = \log(4/\delta)$, we find that, with probability at least $1-\delta/4$, we have

$$\left\|\sum_{i=1}^n \hbar(X_i)\right\|_2 \leq \sqrt{nm}\left(1+\sqrt{8\log(4/\delta)}\right) + \tfrac{4}{3}\log(4/\delta)\sqrt{B}.$$

Since $\log(4/\delta) \geq \log(4)$, we have

$$1 + \sqrt{8\log(4/\delta)} \leq 4\sqrt{\log(4/\delta)}$$

and thus

$$\begin{aligned}\left\|\sum_{i=1}^n \hbar(X_i)\right\|_2 &\leq 4\sqrt{nm\log(4/\delta)} + \tfrac{4}{3}\log(4/\delta)\sqrt{B} \\ &= 4\sqrt{\log(4/\delta)}\left(\sqrt{nm} + \tfrac{1}{3}\sqrt{B\log(4/\delta)}\right).\end{aligned}$$

The condition on n easily implies that

$$\tfrac{1}{3}\sqrt{B\log(4/\delta)} \leq \tfrac{1}{4}\sqrt{nm}$$

and thus

$$\left\|\sum_{i=1}^n \hbar(X_i)\right\|_2 \leq 5\sqrt{nm\log(4/\delta)}. \tag{2.28}$$

Step 3.3. — To control $\left\|\sum_{i=1}^n \hbar(X_i)\epsilon(X_i)\right\|_2$, we apply Lemma 3.18 with $Z_i = \hbar(X_i)\epsilon(X_i)$. The random vectors $\hbar(X_i)\epsilon(X_i)$ for $i = 1, \ldots, n$ are independent and identically distributed and have mean zero. Since $\left\|\hbar(X_i)\right\|_2 \leq \sqrt{B}$ by (2.7) and since $\epsilon \in \mathcal{G}(\tau^2)$ by Assumption 2.10, we have, for all $t > 0$,

$$\mathbb{P}[\left\|\hbar(X_i)\epsilon(X_i)\right\|_2 > t] \le \mathbb{P}[\sqrt{B}\left|\epsilon(X_i)\right| > t] \le 2\exp\left(-\tfrac{t^2}{2B\tau^2}\right),$$

and (3.10) holds with $\sigma^2 = B\tau^2$. Lemma 3.18 then implies that, with probability at least $1-\delta/4$ and $c=3$ that

$$\left\|\sum_{i=1}^n \hbar(X_i)\epsilon(X_i)\right\|_2 \le c\sqrt{nB\tau^2\log(8m/\delta)}. \tag{2.29}$$

Step 3.4. — Recall the $n\times m$ matrix $H=(h_j(X_i))_{i,j}$ and put

$$\bar{H} = HG^{-1/2} = (\hbar_j(X_i))_{i,j}.$$

The empirical Gram matrix of the vector $\hbar=(\hbar_1,\dots,\hbar_m)^\top \in L_2(\pi)^m$ based on the sample $X_1,\dots,X_n$ is

$$P_n(\hbar\hbar^\top) = n^{-1}\bar{H}^\top\bar{H}.$$

We apply Lemma 2.31 with $g=\tilde{g}=\hbar$, $p=m$, and δ replaced by $\delta/4$. We find that, with probability at least $1-\delta/4$,

$$\begin{aligned}\forall u\in\mathbb{R}^m,\ \left\|\bar{H}u\right\|_2^2 &= n\,u^\top P_n(\hbar\hbar^\top)u \\ &\ge n\left(1-\sqrt{2Bn^{-1}\log(4m/\delta)}\right)\|u\|_2^2.\end{aligned} \tag{2.30}$$

Since $P_n(\hbar\hbar^\top)=n^{-1}\bar{H}^\top\bar{H}$, it follows that

$$\lambda_{\min}(\pi_n(\hbar\hbar^\top)) \ge 1-\sqrt{2Bn^{-1}\log(4m/\delta)} \ge \tfrac{2}{3} \tag{2.31}$$

as the assumption on n implies that $2Bn^{-1}\log(4m/\delta)\le 1/9$.

By the union bound, the inequalities (2.27), (2.28), (2.29), and (2.30) hold simultaneously on an event with probability at least $1-\delta$. For the remainder of the proof, we work on this event, denoted by E.

Step 4. — We combine the bound (2.24) on the estimation error with the bounds valid on the event E constructed in Step 3. By (2.31), we have

$$\lambda_{\min}(\pi_n(\hbar\hbar^\top)) - \left\|\pi_n(\hbar)\right\|_2^2 \ge \tfrac{2}{3} - 25mn^{-1}\log(4/\delta) \ge \tfrac{1}{3}$$

since the assumption on n implies that $25mn^{-1}\log(4/\delta)\le 1/3$. As $B\ge m\ge 1$, we have

$$\begin{aligned}\left\|\pi_n(\hbar\epsilon)\right\|_2 + \left\|\pi_n(\hbar)\right\|_2\left|\pi_n(\epsilon)\right| &\le c\sqrt{n^{-1}B\tau^2\log(8m/\delta)} + 5\sqrt{n^{-1}m\log(4/\delta)}\cdot\sqrt{2n^{-1}\tau^2\log(8/\delta)} \\ &\le \sqrt{n^{-1}B\tau^2\log(8m/\delta)}\left(c+5\sqrt{2n^{-1}\log(4/\delta)}\right) \\ &\le (c+\sqrt{2/3})\sqrt{n^{-1}B\tau^2\log(8m/\delta)},\end{aligned}$$

since, by assumption, $n \geq 75m\log(4/\delta)$ which implies that $\sqrt{n^{-1}\log(4/\delta)} \leq 1/(5\sqrt{3})$. We find

$$\begin{aligned}
&\left|\hat{\alpha}_n^{\mathrm{ols}}(f) - \pi(f)\right| \\
&\leq \sqrt{2\tau^2 n^{-1}\log(8/\delta)} + \frac{1}{1/3}\cdot(c+\sqrt{2/3})\sqrt{n^{-1}B\tau^2\log(8m/\delta)}\cdot 5\sqrt{mn^{-1}\log(4/\delta)} \\
&= \sqrt{2\tau^2 n^{-1}\log(8/\delta)} + 15(c+\sqrt{2/3})n^{-1}\sqrt{B\tau^2 m\log(8m/\delta)\log(4/\delta)},
\end{aligned}$$

and the value $c = 3$ gives $15(c+\sqrt{2/3}) \approx 57.2 < 58$ which is the bound stated in Theorem 2.13. □

2.A.3 Proof of Theorem 2.20

For a vector $\beta \in \mathbb{R}^m$ and for a non-empty set $S \subset \{1,\dots,m\}$, write $\beta_S = (\beta_k)_{k\in S}$. For any matrix $A \in \mathbb{R}^{n\times m}$ and $k \in \{1,\dots,m\}$, let A_k denote its k-th column and if $S = \{k_1,\dots,k_\ell\} \subset \{1,\dots,m\}$ with $k_1 < \ldots < k_\ell$, write $A_S = (A_{k_1},\dots,A_{k_\ell}) \in \mathbb{R}^{n\times\ell}$.

The proof is organized in a similar way as the one of Theorem 2.13. We first provide an initial upper bound on the error (Step 1). Then we construct an event that (Step 2) has probability at least $1-\delta$ and (Steps 3, 4, 5) on which we can control each of the terms of the previous upper bound. The combination of all steps to deduce the final statement is made clear in Step 6.

Step 1. — As in the proof of Theorem 2.13, with $\hat{\beta}_n^{\mathrm{ols}}(f)$ replaced by $\hat{\beta}_n^{\mathrm{lasso}}(f)$, the estimation error of the LASSO estimator can be decomposed as

$$n\left(\hat{\alpha}_n^{\mathrm{lasso}}(f) - \pi(f)\right) = \sum_{i=1}^n \epsilon(X_i) + \left(\beta^*(f) - \hat{\beta}_n^{\mathrm{lasso}}(f)\right)^\top \sum_{i=1}^n h(X_i).$$

Writing $\hat{u} = \hat{\beta}_n^{\mathrm{lasso}}(f) - \beta^\star(f)$, we get, by the triangle and Hölder inequalities,

$$n\left|\hat{\alpha}_n^{\mathrm{lasso}}(f) - \pi(f)\right| \leq \left|\sum_{i=1}^n \epsilon(X_i)\right| + \left\|\hat{u}\right\|_1 \max_{k=1,\dots,m}\left|\sum_{i=1}^n h_k(X_i)\right|. \tag{2.32}$$

Step 2. — Let $\delta > 0$. We construct an event, E, with probability at least $1-\delta$ on which four inequalities, namely (2.33), (2.34), (2.35) and (2.36), hold simultaneously.

- Since $\epsilon \in \mathcal{G}(\tau^2)$, we can apply Lemma 2.30 with $p = 1$ to get that, with probability at least $1-\delta/4$,
$$\left|\sum_{i=1}^n \epsilon(X_i)\right| \leq \sqrt{2n\tau^2\log(8/\delta)}. \tag{2.33}$$
- In view of (Boucheron et al., 2013a, Lemma 2.2) and Assumption 2.11, we have $h_k \in \mathcal{G}(U_h^2)$ for all $k = 1,\dots,m$. Hence we can apply Lemma 2.30 with $p = m$ to get that, with probability at least $1-\delta/4$,
$$\max_{k=1,\dots,m}\left|\sum_{i=1}^n h_k(X_i)\right| \leq \sqrt{2nU_h^2\log(8m/\delta)}. \tag{2.34}$$

- By virtue of Assumptions 2.10 and 2.11, we can apply Lemma 2.34 to find $h_k\epsilon \in \mathcal{G}(C\tau^2U_h^2)$ with $C = 9/2$. Hence we can apply Lemma 2.30 to get that, with probability at least $1 - \delta/4$,

$$\max_{k=1,\dots,m} \left| \sum_{i=1}^{n} h_k(X_i)\epsilon(X_i) \right| \leq \sqrt{2nC\tau^2U_h^2 \log(8m/\delta))}. \tag{2.35}$$

- In view of (Boucheron et al., 2013a, Lemma 2.2) and Assumptions 2.11 and 2.23, we have $h_kh_l - P(h_kh_l) \in \mathcal{G}(U_h^4)$ for all $k, l \in \{1, \dots, m\}$. Hence we can apply Lemma 2.30 with $p = m^2$ to get that, with probability at least $1 - \delta/4$,

$$\max_{\substack{1\leq k\leq m\\ 1\leq l\leq m}} \left| \sum_{i=1}^{n} \{h_k(X_i)h_l(X_i) - P(h_kh_l)\} \right| \leq \sqrt{2nU_h^4 \log(8m^2/\delta)}.$$

  Denote by $\Delta = (P_n - P)\{hh^\top\}$. Because by assumption $2(\ell^\star/\gamma^\star)\sqrt{2U_h^4 \log(8m^2/\delta)} \leq \sqrt{n}$, we have that

$$(\ell^\star/\gamma^\star) \max_{1\leq k,l\leq m} |\Delta_{k,l}| \leq 1/2.$$

  Remark that

$$\forall u \in \mathbb{R}^m, \quad n^{-1} \left\|Hu\right\|_2^2 - u^\top Gu = u^\top \Delta u.$$

  Then, following (Bickel et al., 2009, equation (3.3)), use the inequality $|u^\top \Delta u| \leq \|u\|_1^2 \max_{1\leq k,l\leq m} |\Delta_{k,l}|$, to obtain that, with probability $1 - \delta/4$, for all $u \in \mathcal{C}(S^\star; 3)$,

$$\begin{aligned}
\left\|Hu\right\|_2^2 /n &\geq u^\top Gu - \|u\|_1^2 \max_{1\leq k,l\leq m} |\Delta_{k,l}| \\
&\geq u^\top Gu - \|u\|_2^2 \, \ell^\star \max_{1\leq k,l\leq m} |\Delta_{k,l}| \\
&\geq u^\top Gu - (u^\top Gu)(\ell^\star/\gamma^\star) \max_{1\leq k,l\leq m} |\Delta_{k,l}| \\
&\geq (u^\top Gu)/2.
\end{aligned}$$

  It follows that with probability at least $1 - \delta/4$,

$$\left\|Hu\right\|_2^2 \geq (n\gamma^\star/2) \left\|u\right\|_2^2. \tag{2.36}$$

Step 3. — We claim that, on the event E, we have

$$\forall u \in \mathcal{C}(S^\star; 3), \quad \left\|H_cu\right\|_2^2 \geq (n\gamma^\star/4) \left\|u\right\|_2^2 \tag{2.37}$$

We have

$$H_c^\top H_c = H^\top H - n \, \pi_n(h) \, \pi_n(h)^\top$$

and thus,

$$\left\|H_cu\right\|_2^2 \geq \left\|Hu\right\|_2^2 - n \max_{k=1,\dots,m} |\pi_n(h_k)|^2 \left\|u\right\|_1^2.$$

We treat both terms on the right-hand side. On the one hand, we just have obtained a lower bound for the first term. On the other hand, in view of (2.34) and because $\|u\|_1^2 \leq 16 \left\|u_{S^\star}\right\|_1^2 \leq 16\ell^\star \|u\|_2^2$, we have

$$\begin{aligned}\|u\|_1^2 \max_{k=1,\dots,m} |\pi_n(h_k)|^2 = \|u\|_1^2 \, n^{-2} \cdot \max_{k\in S^\star} \left|\sum_{i=1}^n h_k(X_i)\right|^2 \\ \leq 16\ell^\star \|u\|_2^2 \cdot n^{-2} \cdot 2nU_h^2 \log(8m/\delta) \\ \leq \|u\|_2^2 \, \gamma^\star/4\end{aligned}$$

as $n \geq (16 \times 8)\ell^\star(U_h^2/\gamma^\star)\log(8m/\delta)$ by assumption. In combination with (2.36), we find

$$\left\|H_c u\right\|_2^2 \geq n(\gamma^\star/2)\|u\|_2^2 - n(\gamma^\star/4)\|u\|_2^2 = n(\gamma^\star/4)\|u\|_2^2 .$$

Step 4. — We claim that, on the event E, we have

$$\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty \leq (3+\sqrt{2}/8)\sqrt{\log(8m/\delta)}U_h \tau \sqrt{n}. \tag{2.38}$$

Indeed, on the left-hand side in (2.38) we have in virtue of (2.33), (2.34) and (2.35),

$$\begin{aligned}\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty &= \max_{k=1,\dots,m} \left|\sum_{i=1}^n (h_k(X_i) - \pi_n(h_k))(\epsilon(X_i) - \pi_n(\epsilon))\right| \\ &= \max_{k=1,\dots,m} \left|\left(\sum_{i=1}^n h_k(X_i)\epsilon(X_i)\right) - n\pi_n(h_k)\pi_n(\epsilon)\right| \\ &\leq \max_{k=1,\dots,m} \left|\sum_{i=1}^n h_k(X_i)\epsilon(X_i)\right| + n^{-1}\left|\sum_{i=1}^n \epsilon(X_i)\right| \max_{k=1,\dots,m}\left|\sum_{i=1}^n h_k(X_i)\right| \\ &\leq \sqrt{2nC\tau^2 U_h^2 \log(8m/\delta)} + n^{-1}\sqrt{2n\tau^2\log(8/\delta)}\sqrt{2nU_h^2\log(8m/\delta)} \\ &= \sqrt{2nC\tau^2U_h^2\log(8m/\delta)}\left(1+\sqrt{2\log(8/\delta)/(Cn)}\right).\end{aligned}$$

Since $\ell^\star \geq 1$ and $\ell^\star U_h^2 \geq \sum_{k\in S^\star} P(h_k^2) \geq \gamma^\star$, the assumed lower bound on n implies that $n \geq 128\log(8/\delta)$. As $C = 9/2$, the factor $\sqrt{2C}(1+\sqrt{2\log(8/\delta)/(Cn)})$ is bounded by $3+\sqrt{2}/8$ and we get (2.38).

Step 5. — Recall $\hat{u} = \hat{\beta}_n^{\text{lasso}}(f) - \beta^\star(f)$. We claim that, on the event E, we have

$$\left\|\hat{u}\right\|_1 \leq 48\lambda\ell^\star/\gamma^\star. \tag{2.39}$$

To prove this result, we shall rely on the following lemma.

Lemma 2.37. *If* $n\lambda \geq 2\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty$ *then, writing* $\hat{u} = \hat{\beta}_n^{\text{lasso}}(f) - \beta^\star(f)$, *we have* $\hat{u} \in \mathcal{C}(S^\star; 3)$ *and*

$$\left\|H_c\hat{u}\right\|_2^2 \leq 3n\lambda\left\|\hat{u}_{S^\star}\right\|_1 . \tag{2.40}$$

Proof This is just a reformulation of the reasoning on p. 298 in (Tibshirani et al., 2015) with a slightly sharper upper bound. The vector $\hat{\nu}$ at the right-hand side of their Eq. (11.23) can be replaced by $\hat{\nu}_S$. For the sake of completeness, we provide the

details. In the proof we use the shortcuts $\beta^\star = \beta^\star(f)$ and $\hat{\beta}_n^{\text{lasso}} = \hat{\beta}_n^{\text{lasso}}(f)$. Recall $\epsilon_c^{(n)} = f_c^{(n)} - H_c\beta^\star(f)$ and define

$$\begin{aligned} G(u) &= \|f_c^{(n)} - H_c(\beta^\star + u)\|_2^2/(2n) + \lambda\|\beta^\star + u\|_1 \\ &= \|\epsilon_c^{(n)} - H_c u\|_2^2/(2n) + \lambda\|\beta^\star + u\|_1. \end{aligned}$$

Because $G(\hat{u}) \le G(0)$, we have

$$\|H_c\hat{u}\|_2^2/(2n) \le \hat{u}^\top H_c^\top \epsilon_c^{(n)}/n + \lambda(\|\beta^\star\|_1 - \|\beta^\star + \hat{u}\|_1)$$

From the triangle inequality

$$\|(\beta^\star - (-\hat{u}))_{S^\star}\|_1 \ge |\|\beta^\star_{S^\star}\|_1 - \|\hat{u}_{S^\star}\|_1| \ge \|\beta^\star_{S^\star}\|_1 - \|\hat{u}_{S^\star}\|_1,$$

implying that

$$\begin{aligned} &\|\beta^\star\|_1 - \|\beta^\star + \hat{u}\|_1 \\ &= \|\beta^\star\|_1 - \|(\beta^\star + \hat{u})_{S^\star}\|_1 - \|(\beta^\star + \hat{u})_{\overline{S^\star}}\|_1 \\ &\le \|\beta^\star\|_1 - \|\beta^\star_{S^\star}\|_1 + \|\hat{u}_{S^\star}\|_1 - \|(\beta^\star + \hat{u})_{\overline{S^\star}}\|_1 \\ &= \|\hat{u}_{S^\star}\|_1 - \|\hat{u}_{\overline{S^\star}}\|_1. \end{aligned}$$

From Hölder's inequality, we get

$$\left|\hat{u}^\top H_c^\top \epsilon_c^{(n)}\right| \le \left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty \cdot \|\hat{u}\|_1,$$

which leads to

$$\|H_c\hat{u}\|_2^2/(2n) \le \|H_c^\top \epsilon_c^{(n)}\|_\infty \|\hat{u}\|_1/n + \lambda(\|\hat{u}_{S^\star}\|_1 - \|\hat{u}_{\overline{S^\star}}\|_1).$$

Consequently, because $\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty /n \le \lambda/2$ by assumption, we obtain

$$\begin{aligned} 0 \le \|H_c\hat{u}\|_2^2/(2n) &\le \lambda(\|\hat{u}\|_1/2 + \|\hat{u}_{S^\star}\|_1 - \|\hat{u}_{\overline{S^\star}}\|_1) \\ &= (\lambda/2)(3\|\hat{u}_{S^\star}\|_1 - \|\hat{u}_{\overline{S^\star}}\|_1). \end{aligned}$$

The right-hand side must be nonnegative, whence $\|\hat{u}_{\overline{S^\star}}\|_1 \le 3\|\hat{u}_{S^\star}\|_1$, i.e., $\hat{u} \in \mathcal{C}(S;3)$. The bound in (2.40) follows as well. ■

On the event E, the conclusion of Lemma 2.37 is valid because the bound on $\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty$ in (2.38) and the assumption on λ in Theorem 2.20 together imply that $\lambda \ge 2\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty /n$. The cone property of Lemma 2.37 yields $\hat{u} \in \mathcal{C}(S^\star;3)$ so that

$$\|\hat{u}\|_1 = \|\hat{u}_{S^\star}\|_1 + \|\hat{u}_{\overline{S^\star}}\|_1 \le 4\|\hat{u}_{S^\star}\|_1. \tag{2.41}$$

Thanks to (2.37) and Lemma 2.37, and since $|S^\star| = \ell^\star$, we get

$$\begin{aligned} \|\hat{u}_{S^\star}\|_1^2 &\le \ell^\star \|\hat{u}_{S^\star}\|_2^2 \\ &\le \ell^\star \|\hat{u}\|_2^2 \\ &\le \ell^\star \cdot n^{-1}(4/\gamma^\star)\|H_c\hat{u}\|_2^2 \\ &\le \ell^\star \cdot n^{-1}(4/\gamma^\star)\cdot 3n\lambda\|\hat{u}_{S^\star}\|_1 = 12\ell^\star(\lambda/\gamma^\star)\|\hat{u}_{S^\star}\|_1. \end{aligned}$$

It follows that $\left\|\hat{u}_{S^\star}\right\|_1 \leq 12\ell^\star\lambda/\gamma^\star$. In combination with (2.41), we find (2.39).

Step 6. — Equation (2.32) gave a bound on the estimation error involving three terms. On the event E, these terms were shown to be bounded in (2.33), (2.34), and (2.39). It follows that, on E, we finally have

$$n\left|\hat{\alpha}_n^{\text{lasso}}(f) - \pi(f)\right| \leq \sqrt{2n\tau^2\log(8/\delta)} + 48\lambda\ell^\star/\gamma^\star \cdot \sqrt{2nU_h^2\log(8m/\delta)}.$$

Divide by n and use $48\sqrt{2} < 68$ to obtain (2.9). □

2.A.4 Proof of Theorem 2.24

Recall that $S^\star = \{j = 1, \ldots, m : \beta_j^\star(f) \neq 0\}$ with $\ell^\star = \left|S^\star\right|$ and that $\overline{S^\star} = \{1, \ldots, m\} \setminus S^\star$. Further, $H_{c,S^\star}$ is the $n \times \ell^\star$ matrix having columns $H_{c,k}$ for $k \in S^\star$, where $H_{c,k}$ is the k-th column of H_c.

Step 1. — We first establish some (non-probabilistic) properties of $\hat{\beta}_n^{\text{lasso}}(f)$. To this end, we consider the linear regression of the non-active control variates on the active ones: for $k \in \overline{S^\star} = \{j = 1, \ldots, m : \beta_j^\star(f) = 0\}$, this produces the coefficient vector

$$\hat{\theta}_n^{(k)} \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{\ell^\star}} \left\|H_{c,k} - H_{c,S^\star}\theta\right\|_2.$$

Further, we consider the OLS oracle estimate $\hat{\beta}_n^\star$, which is the OLS estimator based upon the active control variables only, i.e.,

$$\hat{\beta}_n^\star \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^{\ell^\star}} \|f_c^{(n)} - H_{c,S^\star}\beta\|_2.$$

Our assumptions will imply that, with large probability, $H_{c,S^\star}$ has rank ℓ^*, in which case

$$\hat{\theta}_n^{(k)} = (H_{c,S^\star}^\top H_{c,S^\star})^{-1} H_{c,S^\star}^\top H_{c,k},$$
$$\hat{\beta}_n^\star = (H_{c,S^\star}^\top H_{c,S^\star})^{-1} H_{c,S^\star}^\top f_c^{(n)}.$$

The following lemma provides a number of (non-probabilistic) properties of $\hat{\beta}_n^{\text{lasso}}(f)$, given certain conditions on H_c and $\epsilon_c^{(n)}$. Recall that a norm $\|\cdot\|$ on $\mathbb{R}^p$ induces a matrix norm on $\mathbb{R}^{p\times p}$ via $\left\|A\right\| = \sup\{\left\|Au\right\| : u \in \mathbb{R}^p, \|u\| = 1\}$ for $A \in \mathbb{R}^{p\times p}$.

Lemma 2.38. *If $H_{c,S^\star}$ has rank $\ell^\star$ and if there exists $\kappa \in (0, 1]$ such that*

$$\max_{k \in \overline{S^\star}} \left\|\hat{\theta}_n^{(k)}\right\|_1 \leq 1 - \kappa, \tag{2.42}$$

$$\max_{k \in \overline{S^\star}} \left|(H_{c,k} - H_{c,S^\star}\hat{\theta}_n^{(k)})^\top \epsilon_c^{(n)}\right| \leq \kappa\lambda n, \tag{2.43}$$

then the minimizer $\hat{\beta}_n^{\text{lasso}}(f)$ in (2.5) is unique, with support $\operatorname{supp}(\hat{\beta}_n^{\text{lasso}}(f)) \subset S^\star$, and it satisfies

$$\max_{k \in S^\star} \left|\hat{\beta}_{n,k}^{\text{lasso}}(f) - \beta_k^\star(f)\right| \leq \max_{k \in S^\star} \left|\hat{\beta}_{n,k}^\star - \beta_k^\star(f)\right| + n\lambda \left\|(H_{c,S^\star}^\top H_{c,S^\star})^{-1}\right\|_\infty. \tag{2.44}$$

Proof The proof of the previous result is actually contained in Tibshirani et al. (2015). The uniqueness of the LASSO solution and the property that it does not select inactive covariates follows directly from the proof of their Theorem 11.3. The only difference is that, in our case, the inequality (2.43) is an assumption whereas in Tibshirani et al. (2015) it is a property of the Gaussian fixed design model. The approach in Tibshirani et al. (2015) is based upon checking the *strict dual feasibility condition*. The bound (2.44) is Eq. (11.37) in Tibshirani et al. (2015). ■

We slightly modify Lemma 2.38 to make the conditions (2.42) and (2.43) easier to check and to make the bound (2.44) easier to use.

Lemma 2.39. *If there exists* $\nu > 0$ *such that*

$$\forall u \in \mathbb{R}^{\ell^\star}, \qquad \left\| H_{c,S^\star} u \right\|_2^2 \geq n\nu \left\| u \right\|_2^2, \tag{2.45}$$

and if there exists $\kappa \in (0,1]$ *such that*

$$\frac{\ell^\star}{\nu n} \max_{k \in \overline{S^\star}} \max_{j \in S^\star} \left| H_{c,j}^\top H_{c,k} \right| \leq 1 - \kappa, \tag{2.46}$$

$$\max_{k=1,\ldots,m} \left| H_{c,k}^\top \epsilon_c^{(n)} \right| \leq \frac{1}{2} \kappa \lambda n, \tag{2.47}$$

then the minimizer $\hat{\beta}_n^{\text{lasso}}(f)$ *in* (2.5) *is unique, with support satisfying* $\operatorname{supp}(\hat{\beta}_n^{\text{lasso}}(f)) \subset S^\star$*, and it holds that*

$$\max_{k \in S^\star} \left| \hat{\beta}_{n,k}^{\text{lasso}}(f) - \beta_k^\star(f) \right| \leq (1 + \kappa/2) \sqrt{\ell^\star} \lambda / \nu. \tag{2.48}$$

Proof

By (2.45), the smallest eigenvalue of the $\ell^\star \times \ell^\star$ matrix $H_{c,S^\star}^\top H_{c,S^\star}$ is positive, so that it is invertible and $H_{c,S^\star}$ has rank $\ell^\star$.

We show that (2.46) implies (2.42). For each $k \in \overline{S^\star}$, the vector $\hat{\theta}_n^{(k)}$ has length $\ell^\star$, so that

$$\left\| \hat{\theta}_n^{(k)} \right\|_1 \leq \sqrt{\ell^\star} \left\| \hat{\theta}_n^{(k)} \right\|_2.$$

Because $\hat{\theta}_n^{(k)}$ is an OLS estimate, using that the largest eigenvalue of $(H_{c,S^\star}^\top H_{c,S^\star})^{-1}$ being bounded from above by $(n\nu)^{-1}$, we obtain

$$\left\| \hat{\theta}_n^{(k)} \right\|_2 = \left\| (H_{c,S^\star}^\top H_{c,S^\star})^{-1} H_{c,S^\star}^\top H_{c,k} \right\|_2 \leq \frac{1}{n\nu} \left\| H_{c,S^\star}^\top H_{c,k} \right\|_2$$

Since $\|x\|_2 \leq \sqrt{m} \, \|x\|_\infty$ for $x \in \mathbb{R}^m$, we can conclude that

$$\left\| \hat{\theta}_n^{(k)} \right\|_2 \leq \frac{\sqrt{\ell^\star}}{\nu n} \max_{j \in S^\star} \left| H_{c,j}^\top H_{c,k} \right|.$$

Combining the two bounds, we find that (2.46) indeed implies (2.42).

Next we show that (2.47) implies (2.43). For $k \in \overline{S^\star}$, we have

$$\begin{aligned}
&\left|(H_{c,k} - H_{c,S^\star}\hat{\theta}_n^{(k)})^\top \epsilon_c^{(n)}\right| \\
&\leq \left|H_{c,k}^\top \epsilon_c^{(n)}\right| + \left|(\hat{\theta}_n^{(k)})^\top H_{c,S^\star}^\top \epsilon_c^{(n)}\right| \\
&\leq \left|H_{c,k}^\top \epsilon_c^{(n)}\right| + \left\|\hat{\theta}_n^{(k)}\right\|_1 \max_{j \in S^\star} \left|H_{c,j}^\top \epsilon_c^{(n)}\right|.
\end{aligned}$$

Using (2.42) and (2.47) we deduce (2.43).

The conditions of Lemma 2.38 have been verified, and so its conclusion holds. We simplify the two terms in the upper bound (2.44). First, we use that

$$\begin{aligned}
\left\|\hat{\beta}_n^\star - \beta^\star(f)\right\|_2 &= \left\|(H_{c,S^\star}^\top H_{c,S^\star})^{-1} H_{c,S^\star}^\top \epsilon_c^{(n)}\right\|_2 \\
&\leq \frac{\sqrt{\ell^\star}}{\nu n} \left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty.
\end{aligned}$$

Second, for any matrix $A \in \mathbb{R}^{p\times p}$, we have $\|A\|_\infty \leq \sqrt{p}\,\|A\|_2$ (e.g., (Horn and Johnson, 2012, page 365)), and this we apply to $(H_{c,S^\star}^\top H_{c,S^\star})^{-1}$. In this way, the upper bound in (2.44) is dominated by

$$\left\|\hat{\beta}_n^\star - \beta^\star(f)\right\|_2 + n\lambda\sqrt{\ell^\star}\left\|(H_{c,S^\star}^\top H_{c,S^\star})^{-1}\right\|_2 \leq \frac{\sqrt{\ell^\star}}{n\nu} \max_{k\in S^\star}\left|H_{c,k}^\top \epsilon_c^{(n)}\right| + n\lambda\sqrt{\ell^\star}\frac{1}{n\nu},$$

since the largest eigenvalue of $(H_{c,S^\star}^\top H_{c,S^\star})^{-1}$ is at most $(n\nu)^{-1}$. Use (2.47) to further simplify the right-hand side, yielding (2.48). ∎

Step 2. — Let $\delta \in (0,1)$ and $n = 1, 2, \ldots$. In a similar way as in the proof of Theorem 2.13, we construct an event of probability at least $1-\delta$. This time, we need five inequalities to hold simultaneously.

- Because $\epsilon \in \mathcal{G}(\tau^2)$, with probability at least $1 - \delta/5$,

$$\left|\sum_{i=1}^n \epsilon(X_i)\right| \leq \sqrt{2n\tau^2 \log(10/\delta)}. \tag{2.49}$$

- In view of (Boucheron et al., 2013a, Lemma 2.2) and Assumption 2.11, we have $h_k \in \mathcal{G}(U_h^2)$ for all $k = 1, \ldots, m$. Hence we can apply Lemma 2.30 with $p = m$ to get that, with probability at least $1 - \delta/5$,

$$\max_{k=1,\ldots,m}\left|\sum_{i=1}^n h_k(X_i)\right| \leq \sqrt{2nU_h^2 \log(10m/\delta)}. \tag{2.50}$$

- By virtue of Assumptions 2.10 and 2.11, we can apply Lemma 2.34 to have $h_k\epsilon \in \mathcal{G}(CU_h^2\tau^2)$, where $C = 9/2$. Hence we can apply Lemma 2.30 to get that, with probability at least $1 - \delta/5$,

$$\max_{k=1,\dots,m} \left| \sum_{i=1}^{n} h_k(X_i)\epsilon(X_i) \right| \le \sqrt{2Cn\tau^2U_h^2 \log(10m/\delta))}. \tag{2.51}$$

- Recall that $B^\star = \sup_{x\in\mathcal{X}} h_{S^\star}^\top(x) G_{S^\star}^{-1} h_{S^\star}(x)$ with

$$B^\star \le \lambda_{\max}(G_{S^\star}^{-1}) \sup_{x\in\mathcal{X}} h_{S^\star}^\top(x) h_{S^\star}(x) \le \ell^\star U_h^2/\gamma^{\star\star},$$

  The assumption on n easily implies that $n \ge 8B^\star \log(5\ell^\star/\delta)$. Applying Lemma 2.31 with $p = \ell^\star$, $g = h_{S^\star}$, and δ replaced by $\delta/5$, we find that, with probability at least $1 - \delta/5$,

$$\left\| H_{S^\star} u \right\|_2^2 \ge n\gamma^{\star\star} \|u\|_2^2/2, \qquad \forall u \in \mathbb{R}^{\ell^\star}. \tag{2.52}$$

- Finally, because $\left|h_j(x)\right| \le U_h$ for all $x \in \mathcal{X}$ and $j \in \{1,\dots,m\}$ and because $P(h_k h_j) = 0$ for all $(k,j) \in \overline{S^\star} \times S^\star$, we have $h_k h_j \in \mathcal{G}(U_h^4)$ for such k and j, and thus, with probability at least $1 - \delta/5$,

$$\max_{k\in\overline{S^\star}} \max_{j\in S^\star} \left| \sum_{i=1}^{n} h_k(X_i)h_j(X_i) \right| \le \sqrt{2nU_h^4 \log(10\ell^\star m/\delta)}. \tag{2.53}$$

By the union bound, the event, say E, on which (2.49), (2.50), (2.51), (2.52) and (2.53) are satisfied simultaneously has probability at least $1 - \delta$. We work on the event E for the rest of the proof.

Step 3. — On the event E, we have

$$\forall u \in \mathbb{R}^{\ell^\star}, \qquad \left\| H_{c,S^\star} u \right\|_2^2 \ge n\alpha\gamma^{\star\star} \|u\|_2^2, \tag{2.54}$$

where $\alpha \in (0, 1/2)$ is an absolute constant whose value will be fixed in Step 6(ii). We have

$$H_{c,S^\star}^\top H_{c,S^\star} = H_{S^\star}^\top H_{S^\star} - n\, \pi_n(h_{S^\star})\, \pi_n(h_{S^\star})^\top$$

and thus, by the Cauchy–Schwarz inequality and by (2.52),

$$\begin{aligned} \left\| H_{c,S^\star} u \right\|_2^2 &\ge \left\| H_{S^\star} u \right\|_2^2 - n \left\| \pi_n(h_{S^\star}) \right\|_2^2 \|u\|_2^2 \\ &\ge n \left(\gamma^{\star\star}/2 - \left\| \pi_n(h_{S^\star}) \right\|_2^2 \right) \|u\|_2^2. \end{aligned}$$

In view of (2.50), we have

$$\left\| \pi_n(h_{S^\star}) \right\|_2^2 \le \frac{\ell^\star}{n^2} 2nU_h^2 \log(10m/\delta) = 2\ell^\star \log(10m/\delta) U_h^2/n.$$

We thus get

$$\left\|H_{c,S^\star}u\right\|_2^2 \geq n\gamma^{\star\star}\left[\frac{1}{2} - \frac{2\ell^\star \log(10m/\delta)U_h^2/\gamma^{\star\star}}{n}\right]\|u\|_2^2$$

A sufficient condition for (2.54) is thus that the term in square brackets is at least α, i.e.,

$$n \geq \frac{2}{1/2-\alpha}\ell^\star \log(10m/\delta)U_h^2/\gamma^{\star\star}$$

Since $\ell^\star \geq 1$ and $U_h^2 \geq \gamma^{\star\star}$, a condition of the form

$$n \geq \rho \log(10\ell^\star m/\delta)[\ell^\star(U_h^2/\gamma^{\star\star})]^2 \tag{2.55}$$

is thus sufficient, with much to spare, provided $\rho > 2/(1/2-\alpha)$. In Step 6(ii), we will choose α in such a way that the constant $\rho = 70$ appearing in the statement of the theorem is sufficient.

Step 4. — On the event E, we have

$$\max_{k\in\overline{S^\star}}\max_{j\in S^\star}|H_{c,j}^\top H_{c,k}| \leq \sqrt{2nU_h^4\log(10\ell^\star m/\delta)} + 2U_h^2\log(10m/\delta). \tag{2.56}$$

Indeed, denote $A = \overline{S^\star}\times S^\star$, in virtue of (2.50) and (2.53), the left-hand side is bounded by

$$\begin{aligned}
&\max_{(k,j)\in A}\left|\left(\sum_{i=1}^n h_k(X_i)h_j(X_i)\right) - n\pi_n(h_k)\pi_n(h_j)\right| \\
&\leq \max_{(k,j)\in A}\left|\sum_{i=1}^n h_k(X_i)h_j(X_i)\right| + \frac{1}{n}\max_{k\in\overline{S^\star}}\left|\sum_{i=1}^n h_k(X_i)\right|\max_{j\in S^\star}\left|\sum_{i=1}^n h_j(X_i)\right| \\
&\leq \max_{(k,j)\in A}\left|\sum_{i=1}^n h_k(X_i)h_j(X_i)\right| + \frac{1}{n}\max_{k=1,\dots m}\left|\sum_{i=1}^n h_k(X_i)\right|^2 \\
&\leq \sqrt{2nU_h^4\log(10\ell^\star m/\delta)} + \frac{1}{n}2nU_h^2\log(10m/\delta),
\end{aligned}$$

which is (2.56).

Step 5. — On the event E, we have

$$\left\|H_c^\top \epsilon_c^{(n)}\right\|_\infty \leq \sqrt{2nC\tau^2U_h^2\log(10m/\delta)}\left(1+\sqrt{2\log(10/\delta)/(Cn)}\right). \tag{2.57}$$

The proof is the same as the first part of the one (2.38).

Step 6. — We will verify that on the event E, the three assumptions of Lemma 2.39 are satisfied with $\kappa = 1/2$ and $\nu = \alpha\gamma^{\star\star}$, with α as in Step 3.

(i) Eq. (2.45) with $\nu = \alpha\gamma^{\star\star}$ is just (2.54).

(ii) Eq. (2.46) with $\nu = \alpha\gamma^{\star\star}$ and $\kappa = 1/2$ follows from (2.56) provided we have

$$\frac{\ell^\star}{\alpha\gamma^{\star\star}n}\left(\sqrt{2nU_h^4\log(10\ell^\star m/\delta)} + 2U_h^2\log(10m/\delta)\right) \le 1 - \frac{1}{2}.$$

To check whether this is satisfied, we will make use of the elementary inequality[2]

$$\forall (a,b,c) \in (0,\infty)^3,\ \forall x \ge \sqrt{b^2+4ac}/a, \qquad ax^2 \ge bx + c.$$

with $x = \sqrt{n}$ and

$$a = \alpha\gamma^{\star\star}/(2\ell^\star), \quad b = \sqrt{2U_h^4\log(10\ell^\star m/\delta)}, \quad c = 2U_h^2\log(10m/\delta).$$

Sufficient is that $n = x^2$ is bounded from below by $(b^2+4ac)/a^2 = (b/a)^2 + 4c/a$, which is

$$\frac{2U_h^4\log(10\ell^\star m/\delta)}{(\alpha\gamma^{\star\star}/(2\ell^\star))^2} + 4\frac{2U_h^2\log(10m/\delta)}{\alpha\gamma^{\star\star}/(2\ell^\star)} = \frac{8}{\alpha^2}\log(10\ell^\star m/\delta)\left(\frac{\ell^\star U_h^2}{\gamma^{\star\star}}\right)^2 + \frac{16}{\alpha}\log(10m/\delta)\left(\frac{\ell^\star U_h^2}{\gamma^{\star\star}}\right).$$

But $\ell^\star \ge 1$ and $\gamma^{\star\star} \le (1/\ell^\star)\sum_{j\in S^\star}\pi(h_j^2) \le U_h^2$, so that a sufficient condition is that

$$n \ge \left(\frac{8}{\alpha^2} + \frac{16}{\alpha}\right)\log(10\ell^\star m/\delta)[\ell^\star(U_h^2/\gamma^{\star\star})]^2.$$

The constant ρ in (2.55) must thus be such that

$$\rho \ge \max\left(\frac{2}{1/2-\alpha}, \frac{8}{\alpha^2} + \frac{16}{\alpha}\right).$$

The minimum of the right-hand side as a function of $\alpha \in (0,1/2)$ occurs at $\alpha = \sqrt{2}/3$ and is equal to $2/(1/2 - \sqrt{2}/3) \approx 69.9$. Taking $\rho = 70$ as in the assumption on n is thus sufficient.

(iii) Eq. (2.47) with $\kappa = 1/2$ follows from (2.57), since

$$\sqrt{n\tau^2U_h^2\log(10m/\delta)}\left(\sqrt{2C} + 2\sqrt{\log(10/\delta)/n}\right) \le \lambda n/4$$

by the assumed lower bound on λ. Indeed, since $\ell^\star \ge 1$ and $U_h^2 \ge \gamma^{\star\star}$, the assumed lower bounds on n imply that $n \ge 70\log(10/\delta)$, so that $2\sqrt{\log(10/\delta)/n}$ is bounded by $2/\sqrt{70}$; recall $C = 9/2$. Since $4\cdot(3+2/\sqrt{70}) \approx 12.9$, the assumed lower bound for λ suffices.

Step 7. — By the previous step, the conclusions of Lemma 2.39 with $\kappa = 1/2$ and $\nu = \alpha\gamma^{\star\star}$ hold on the event E, where $\alpha = \sqrt{2}/3$ was specified in Step 6(ii). The minimizer $\hat\beta_n^{\text{lasso}}$ in (2.5) is thus unique and we have $\mathrm{supp}(\hat\beta_n^{\text{lasso}}(f)) \subset S^\star$.

[2]The convex parabola $x \mapsto ax^2 - bx - c$ has zeroes at $x_\pm = (b \pm \sqrt{b^2+4ac})/(2a)$, and $x_- < 0 < x_+ < \sqrt{b^2+4ac}/a$.

To show the reverse inclusion, we need to verify that $\left|\hat{\beta}^{\text{lasso}}_{n,k}(f)\right| > 0$ for all $k \in S^\star$. To this end, we apply (2.48) with $\kappa = 1/2$ and $\nu = \alpha\gamma^{\star\star}$, which becomes

$$\max_{k\in S^\star}\left|\hat{\beta}^{\text{lasso}}_{n,k}(f) - \beta^\star_k(f)\right| \leq (5/4)\sqrt{\ell^\star}\lambda/(\alpha\gamma^{\star\star}).$$

For any $k \in S^\star$, we thus have

$$\left|\hat{\beta}^{\text{lasso}}_{n,k}(f)\right| \geq \min_{j\in S^\star}\left|\beta^\star_j(f)\right| - (5/(4\alpha))\sqrt{\ell^\star}\lambda/\gamma^{\star\star}.$$

But for $\alpha = \sqrt{2}/3$, we have approximately $5/(4\alpha) \approx 2.65$. Since $\min_{j\in S^\star}\left|\beta^\star_j(f)\right| > 3\sqrt{\ell^\star}\lambda/\gamma^{\star\star}$ by the assumed upper bound for λ, we find $\left|\hat{\beta}^{\text{lasso}}_{n,k}(f)\right| > 0$, as required. □

2.A.5 Proof of Theorem 2.25

Recall that the LSLASSO estimator is defined as an OLS estimate computed on the active variables selected by the LASSO based on a subsample of size $N \in \{1,\dots,n\}$. Let $\hat{\beta}^{\text{lasso}}_N(f)$ denote the LASSO coefficient vector in (2.5) based on the subsample $X_1,\dots,X_N$ and let

$$\hat{S}_N = \text{supp}(\hat{\beta}^{\text{lasso}}_N(f)) = \{k \in \{1,\dots,m\} \,:\, \hat{\beta}^{\text{lasso}}_{N,k}(f) \neq 0\}$$

denote the estimated active set of $\hat{\ell} = |\hat{S}_N|$ control variates. The LSLASSO estimate $\hat{\alpha}^{\text{lslasso}}_n(f)$ based on the full sample $X_1,\dots,X_n$ is defined as the OLS estimator based on the control variates h_k for $k \in \hat{S}_N$: writing $H_{\hat{S}_N}$ for the $n \times \hat{\ell}$ matrix with columns $(h_k(X_i))_{i=1}^n$ with $k \in \hat{S}_N$, we have

$$\left(\hat{\alpha}^{\text{lslasso}}_n(f), \hat{\beta}^{\text{lslasso}}_n(f)\right) \in \operatorname*{arg\,min}_{(\alpha,\beta)\in\mathbb{R}\times\mathbb{R}^{\hat{\ell}}}\left\|f^{(n)} - \alpha\mathbb{1}_n - H_{\hat{S}_N}\beta\right\|_2^2,$$

Therefore, we can derive a concentration inequality by combining the support recovery property (Theorem 2.24) along with the concentration inequality for the OLS estimate (Theorem 2.13) using only the active control variates.

Let $\delta > 0$ and $n \geq 1$. We construct an event with probability at least $1-\delta$ on which the support recovery property and the concentration inequality for the OLS estimate hold simultaneously. Recall that $S^\star = \text{supp}(\beta^\star(f))$ is the true set of $\ell^\star = \left|S^\star\right|$ active control variables.

- Thanks to Theorem 2.24, with probability at least $1-\delta/2$,

$$\hat{S}_N = S^\star. \tag{2.58}$$

  Indeed, the conditions on N and λ in Theorem 2.25 are such that we can apply Theorem 2.24 with n and δ replaced by N and $\delta/2$, respectively.

- Thanks to Theorem 2.13, with probability at least $1-\delta/2$,

$$\left|\hat{\alpha}^{\text{ols}}_n(f, h_{S^\star}) - \pi(f)\right| \leq \sqrt{2\log(16/\delta)}\frac{\tau}{\sqrt{n}} + 58\sqrt{B^\star\ell^\star\log(16\ell^\star/\delta)\log(8/\delta)}\frac{\tau}{n}. \tag{2.59}$$

where for any $S \subset \{1, \ldots, m\}$, $\hat{\alpha}_n^{\text{ols}}(f, h_S)$ is the OLS estimate of $P(f)$ based on the control variates h_S. Indeed, we apply Theorem 2.13 with h and δ replaced by $h_{S^\star}$ and $\delta/2$, respectively. The required lower bound on n is now

$$n \geq \max\left(18B^\star \log(8\ell^\star/\delta), 75\ell^\star \log(8/\delta)\right).$$

By assumption we have $N \geq 75[\ell^\star(U_h^2/\gamma^{\star\star})]^2 \log(20\ell^\star/\delta)$. The required lower bound is already satisfied for N, and thus certainly by n.

By the union bound, the event on which (2.58) and (2.59) are satisfied simultaneously has probability at least $1 - \delta$. On this event, we can, by definition of $\hat{\alpha}_n^{\text{lslasso}}(f)$ and by (2.58), write the integration error as

$$\left|\hat{\alpha}_n^{\text{lslasso}}(f) - \pi(f)\right| = \left|\hat{\alpha}_n^{\text{ols}}(f, h_{\hat{S}_N}) - \pi(f)\right| = \left|\hat{\alpha}_n^{\text{ols}}(f, h_{S^\star}) - \pi(f)\right|.$$

But the right-hand side is bounded by (2.59), yielding (2.13), as required.

Chapter 3

Combining Control Variates and Adaptive Importance Sampling

Contents

3.1 Introduction

In recent years, sequential simulation has emerged as a leading approach to compute multidimensional integrals. A key object in sequential simulation is the sequence of distributions, called the policy, from which to generate the random variables, called particles, used to approximate the integrals of interest. The policy is designed to evolve in the course of the algorithm to mimic the target density, which may itself be known only up to a proportionality constant. While the design of algorithms with adaptive policies has been of major interest recently, only a few studies have focused on using control variates to reduce the variance. This chapter provides a new method to incorporate control variates within standard sequential algorithms. The proposed approach significantly improves the accuracy of the initial algorithm, both theoretically and in practice.

The sequential framework. Consider the problem of approximating the integral $\int f\pi \,\mathrm{d}\lambda = \int_{\mathbb{R}^d} f(x)\pi(x)\,\mathrm{d}x$, where λ is the d-dimensional Lebesgue measure, π is a probability density on $\mathbb{R}^d$ and the integrand f is a real-valued function on $\mathbb{R}^d$. For instance, one may think of π as the posterior density in Bayesian inference. Let $(q_i)_{i\geq 0}$ be the policy of the algorithm, i.e., a sequence of probability densities which evolves adaptively depending on previous outcomes. The particles $(X_i)_{i\geq 1}$ are generated sequentially—at iteration i, particle X_i is drawn from q_{i-1}. The integral $\int f\pi \,\mathrm{d}\lambda$ is estimated by the normalized sum $\Big(\sum_{i=1}^n w_i f(X_i)\Big)/\Big(\sum_{i=1}^n w_i\Big)$, where $w_i = \pi(X_i)/q_{i-1}(X_i)$ are the importance weights. The normalization $\sum_{i=1}^n w_i$ allows to deal with situations where the target density π is known only up to a proportionality constant.

Such an algorithm is part of the *adaptive importance sampling* (AIS) framework. Many different ways have been investigated to update the densities q_i adaptively. Early works that inspired such sequential schemes include Geweke (1989); Kloek and Van Dijk (1978); Oh and Berger (1992) where the sampling policy is chosen out of a parametric family. The parametric approach has been further extended by the Population Monte Carlo framework (Cappé et al., 2008, 2004; Martino et al., 2017). Various asymptotic results have been obtained in Chopin (2004); Douc and Moulines (2008); Portier and Delyon (2018). In Dai et al. (2016); Delyon and Portier (2021); Korba and Portier (2022); Zhang (1996), *nonparametric importance sampling* based on kernel smoothing is studied. The latter bears resemblance to *sequential Monte Carlo* methods (Del Moral et al., 2006; Chopin, 2004), in which the target distribution π changes in the course of the algorithm.

Let $h = (h_1, \dots, h_m)^\top$ be a vector of real-valued functions on $\mathbb{R}^d$ such that for each k, the integral $\int h_k \pi \, \mathrm{d}\lambda$ is known. Without loss of generality, suppose that $\int h\pi \, \mathrm{d}\lambda = 0$. The functions h_k are called control variates and can be obtained in different ways. In Bayesian statistics, Stein control variates (Oates et al., 2017) are constructed by applying the second-order Stein operator to functions satisfying certain regularity conditions (Mira et al., 2013). Other control variates might be created by re-weighting a function h^* that satisfies $\int h^* \, \mathrm{d}\lambda = 0$ via $h = h^*/\pi$. The use of control variates is a well studied variance-reduction technique (Glynn and Szechtman, 2002; Owen and Zhou, 2000). The benefits can be established theoretically in terms of error bounds (see Oates et al. (2017) and chapter 2), weak convergence (Portier and Segers, 2019), the excess risk (Belomestny et al., 2022) and even uniform error bounds over large classes of integrands (Plassier et al., 2020). In practice, the control variates framework has led to efficient procedures in reinforcement learning Jie and Abbeel (2010); Liu et al. (2018) and optimization Wang et al. (2013), to name a few. Importance sampling and control variates in case of a Gaussian target density is explored in Jourdain (2009). The procedure in Kawai (2020) incorporates control variates and is said to involve adaptive importance sampling, but in fact the particles are always sampled from the uniform distribution on the unit cube. To the best of our knowledge, the existing control variate methods do not account for sequential changes in the particle distribution as is the case in AIS.

AISCV estimate. The proposed approach to use control variates within the sequential AIS framework relies on the ordinary least squares expression of control variates (see for instance Portier and Segers (2019)). To take care of the policy changes, some re-weighting must be applied. The AISCV estimate of the integral $\int gf \, \mathrm{d}\lambda$ is defined as the first coordinate of the solution to the weighted least squares problem

$$(\hat{\alpha}_n, \hat{\beta}_n) = \operatorname*{arg\,min}_{a \in \mathbb{R}, b \in \mathbb{R}^m} \sum_{i=1}^{n} w_i \left(f(X_i) - a - b^\top h(X_i) \right)^2,$$

with w_i the importance weights from before. The AISCV estimate $\hat{\alpha}_n$ has several interesting properties: (a) whenever f is of the form $\alpha + \beta^\top h$ for some $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, the error is zero, i.e., $\hat{\alpha}_n = \alpha = \int f\pi \, \mathrm{d}\lambda$; (b) the estimate takes the form of a quadrature rule $\hat{\alpha}_n = \sum_{i=1}^n v_{n,i} f(X_i)$, for quadrature weights $v_{n,i}$ that do not depend on the function g and that can be computed by a single weighted least squares procedure; and (c) it can be computed even when f is known only up to a multiplicative constant. Point (a) suggests that when the linear combinations of the functions h_k span a rich function class, the integration error is likely to be small. Point (b) implies that multiple integrals can be computed just as easily as a single one. Point (c) shows that the approach is applicable for Bayesian computations. In addition, the control variates

can be brought into play in a *post-hoc* scheme, after generation of the particles and importance weights, and this for any AIS algorithm.

Main result. The main theoretical result of the chapter is a probabilistic, non-asymptotic bound on $\hat{\alpha}_n - \alpha$. Under appropriate conditions, the bound scales as $\tau/\sqrt{n}$, where τ^2 is the scale constant in a sub-Gaussian tail condition on the error variable $\varepsilon = f - \alpha - \beta^\top h$ for $(\alpha, \beta) = \arg\min_{a,b} \int (f - a - b^\top h)^2 \pi \, \mathrm{d}\lambda$. Note that ε has the smallest possible variance one could get using control variates h. As a consequence, when the space of control variates is well suited for approximating f, the AISCV estimate will be highly accurate. Also, our bound depends only on the linear function space spanned by the control variates $h_1, \ldots, h_m$, not on the particular basis chosen in that space. The results rely on martingale theory, in particular on a concentration inequality for norm-subGaussian martingales in Jin et al. (2019). In the course of the proof, we develop a novel bound on the smallest eigenvalue of certain random matrices, extending an inequality from (Tropp, 2015) to the martingale case.

Outline. Section 3.2 introduces the general framework of adaptive importance sampling and control variates. Next, Section 3.3 presents the AISCV estimate and the associated quadrature rule. Section 3.4 contains the statements of the theoretical results while Section 3.5 gathers practical considerations, including the construction of control variates. Numerical experiments are presented in Section 3.6 and Section 3.7 concludes the main part of the chapter with a discussion for further research.

3.2 Preliminaries on Monte Carlo integration

The aim of this section is to present the required mathematical framework for Monte Carlo integration and the variance reduction methods of interest, namely adaptive importance sampling and the control variate technique. Recall that $f : \mathbb{R}^d \to \mathbb{R}$ is an integrand and π a probability density on $\mathbb{R}^d$. The aim is to compute $\mathbb{E}_\pi[f] = \int f \pi \, \mathrm{d}\lambda$.

Adaptive importance sampling. In adaptive importance sampling (AIS), $\mathbb{E}_\pi[f]$ is estimated by a weighted mean over a sample of random particles $X_1, \ldots, X_n$ in $\mathbb{R}^d$. Since appropriate sampling densities naturally depend on f and π, we generally cannot simulate from them. They are then approximated in an adaptive manner by a family of tractable densities $(q_i)_{i \geq 0}$ that often evolve towards a density q_{opt} that optimizes some criterion. While the starting density q_0 is fixed, the density q_i for $i \geq 1$ is determined in function of the particles $X_1, \ldots, X_i$ already sampled; think for instance of a parametric family, where the parameter of q_i is a function of $X_1, \ldots, X_i$. Given the particles $X_1, \ldots, X_i$, the next particle, X_{i+1}, is then drawn from q_i. Formally, let $(X_i)_{i \geq 1}$ be a sequence of random vectors on $\mathbb{R}^d$ defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The distribution of the sequence $(X_i)_{i \geq 1}$ is specified by its policy as defined below.

Definition 3.1 (Policy)**.** *A policy is a random sequence of probability density functions $(q_i)_{i \geq 0}$ on $\mathbb{R}^d$ adapted to the σ-field $(\mathcal{F}_i)_{i \geq 0}$ defined by $\mathcal{F}_0 = \{\varnothing, \Omega\}$ and $\mathcal{F}_i = \sigma(X_1, \ldots, X_i)$ for $i \geq 1$. The sequence $(q_i)_{i \geq 0}$ is the policy of $(X_i)_{i \geq 1}$ whenever X_i has density q_{i-1} conditionally on $\mathcal{F}_{i-1}$.*

The (normalized) adaptive importance sampling estimate of $\mathbb{E}_\pi[f]$ is then defined as

$$\hat{\alpha}_n^{(\mathrm{ais})}(f) = \frac{\sum_{i=1}^n w_i f(X_i)}{\sum_{i=1}^n w_i} \quad \text{where} \quad w_i = \frac{\pi(X_i)}{q_{i-1}(X_i)} \quad \text{for } i = 1, \ldots, n. \tag{3.1}$$

The sampling weights w_i reflect the fact that X_i has been sampled from q_{i-1} rather than from π. The division by $\sum_{i=1}^n w_i$ rather than by n has two benefits: first, the integration is exact for constant integrands, and second, π needs to be known only up to a proportionality constant, an advantage for Bayesian inference.

Since updating the density q_i at each iteration may be computationally expensive, it is customary to hold it fixed over a pre-determined number of iterations. Writing $n = n_1 + \cdots + n_T$ in terms of positive integers $(n_t)_{t=1}^T$ called the *allocation policy*, the AIS estimate then becomes

$$\hat{\alpha}_T^{(\mathrm{ais})}(g) = \frac{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} f(X_{t,i})}{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i}} \quad \text{where} \quad w_{t,i} = \frac{\pi(X_{t,i})}{q_t(X_{t,i})} \tag{3.2}$$

for $t = 1, \ldots, T$ and $i = 1, \ldots, n_t$. At stage t, the particles $X_{t,1}, \ldots, X_{t,n_t}$ are sampled independently from q_{t-1}, while all particles sampled up to and including stage t are used to determine the sampling density q_t for stage $t+1$. It is easy to see that the two formulations of the AIS estimate are equivalent: (3.1) arises from (3.2) by setting $n_t = 1$ for all t, while (3.2) can be obtained from (3.1) by constructing the policy in such a way that the densities q_i do not change within integer intervals of the form $\{0, \ldots, n_1 - 1\}$, $\{n_1, \ldots, n_1 + n_2 - 1\}$, and so on. While the shorter representation (3.1) is more convenient for theoretical purposes, formulation (3.2) is the one used in practice (see Section 3.6).

Interestingly, the AIS estimate (3.1) may be seen as a weighted least-squares estimate minimizing the loss function $a \mapsto \sum_{i=1}^n w_i (f(X_i) - a)^2$. This perspective is key to understand control variates.

Control variates. The control variates method is a variance reduction technique that consists in incorporating a new piece of information—the known values of the integrals of some control functions—in a basic Monte Carlo framework. Control variates are simply functions $h_1, \ldots, h_m \in L_2(\pi)$ with known integrals. Without loss of generality, assume that $\mathbb{E}_\pi[h_j] = 0$ for all $j = 1, \ldots, m$. Let $h = (h_1, \ldots, h_m)^\top$ denote the $\mathbb{R}^m$-valued function with the m control variates as elements. For any coefficient vector $\beta \in \mathbb{R}^m$, we have $\mathbb{E}_\pi[f - \beta^\top h] = \mathbb{E}_\pi[f]$. Given an independent random sample $X_1, \ldots, X_n$ from π, any $\beta \in \mathbb{R}^m$ therefore results in an unbiased estimator of $\mathbb{E}_\pi[f]$ by

$$\alpha_n^{(\mathrm{cv})}(f, \beta) = \frac{1}{n} \sum_{i=1}^n \{f(X_i) - \beta^\top h(X_i)\}. \tag{3.3}$$

Provided the $m \times m$ covariance matrix $G = \mathbb{E}_\pi[hh^\top]$ is invertible, there is a unique coefficient vector $\beta^* \in \mathbb{R}^m$ for which the variance of $\alpha_n^{(\mathrm{cv})}(f)$ is minimal and it is given by

$$\beta^* = \Big(\mathbb{E}_\pi[hh^\top]\Big)^{-1} \mathbb{E}_\pi[hf]. \tag{3.4}$$

This vector being generally unknown, it needs to be estimated from the particles $X_1, \ldots, X_n$. Casting the problem in an ordinary least squares framework leads to the control variate estimate

$$\begin{gathered} \alpha_n^{(\mathrm{cv})}(f) = \alpha_n^{(\mathrm{cv})}\Big(f, \hat{\beta}_n^{(\mathrm{cv})}\Big) = \hat{\alpha}_n^{(\mathrm{cv})} \qquad \text{where} \\ \Big(\hat{\alpha}_n^{(\mathrm{cv})}, \hat{\beta}_n^{(\mathrm{cv})}\Big) \in \operatorname*{arg\,min}_{(a,b) \in \mathbb{R} \times \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^n \Big(f(X_i) - a - b^\top h(X_i)\Big)^2. \end{gathered} \tag{3.5}$$

The estimator $\alpha_n^{(\mathrm{cv})}(f)$ is well-defined provided the minimizer $\hat{\alpha}_n^{(\mathrm{cv})}$ to (3.5) is unique. This is the case if and only if there does not exist $b \in \mathbb{R}^m$ such that $b^\top h(X_i) = 1$ for all $i = 1, \dots, n$.

The asymptotic distribution of $\alpha_n^{(\mathrm{cv})}(f)$ as $n \to \infty$ is the same as if the variance-minimizing vector β^* were used in (3.3). In particular, the asymptotic variance of $\alpha_n^{(\mathrm{cv})}(f)$ is $\sigma_m^2(f)/n$ where

$$\sigma_m^2(g) = \min_{\beta \in \mathbb{R}^m} \mathbb{E}_f\Big[(f - \mathbb{E}_\pi[f] - \beta^\top h)^2\Big].$$

Interestingly, when using only the first ℓ out of m control variates, where $\ell \in \{0, 1, \dots, m\}$, we have $\sigma_m^2(f) \le \sigma_\ell^2(f)$. In terms of asymptotic variance, it therefore never harms to add more control variates. Their construction will be addressed in Section 3.5.1.

3.3 Combining adaptive importance sampling with control variates

AISCV estimator. Consider the same integration problem $\mathbb{E}_\pi[f] = \int f\pi \,\mathrm{d}\lambda$ as in Section 3.2. With the idea of performing variance reduction when calculating integrals with respect to the posterior density in Bayesian inference, we incorporate control variates into the AIS estimate. Let the particles $(X_i)_{i\ge1}$ be generated according to a policy $(q_i)_{i\ge0}$ as in Definition 3.1. Let $h = (h_1, \dots, h_m)^\top$ be a vector of control variates, i.e., $h_j \in L_2(\pi)$ and $\mathbb{E}_\pi[h_j] = 0$ for every $j = 1, \dots, m$. Combining (3.1) and (3.3), the proposed estimate takes the form

$$\alpha_n^{(\mathrm{aiscv})}(f, \beta) = \frac{\sum_{i=1}^n w_i \Big(f(X_i) - \beta^\top h(X_i)\Big)}{\sum_{i=1}^n w_i}, \tag{3.6}$$

where $\beta \in \mathbb{R}^m$ remains to be determined. To do so, the ordinary least-squares problem in (3.5) is replaced by a weighted one, yielding the novel AISCV estimator

$$\begin{aligned} &\alpha_n^{(\mathrm{aiscv})}(f) = \alpha_n^{(\mathrm{aiscv})}\Big(f, \hat{\beta}_n\Big) = \hat{\alpha}_n \qquad \text{where} \\ &\Big(\hat{\alpha}_n, \hat{\beta}_n\Big) \in \operatorname*{arg\,min}_{(a,b)\in\mathbb{R}\times\mathbb{R}^m} \sum_{i=1}^n w_i \Big(f(X_i) - a - b^\top h(X_i)\Big)^2. \end{aligned} \tag{3.7}$$

The estimator is well-defined only if the minimizer $\hat{\alpha}_n$ is unique—the minimizer $\hat{\beta}_n$ need not be. We will come back to this in the next paragraph.

As in (3.2), the policy may be divided into T stages in order to reduce the number of times the sampler needs to be updated. Stage $t = 1, \dots, T$ has length n_t, with $\sum_{t=1}^T n_t = n$. Within each stage, the sampling density remains constant. In practice, this leads to the AISCV estimate in Algorithm 3.4.

Quadrature rule. The AIS estimate (3.1) is a quadrature rule with quadrature points X_i and quadrature weights proportional to the sampling weights w_i. The AISCV estimate (3.7) has the same property, but with adapted quadrature weights. Let $e_n = (e_{n,i})_{i=1,\dots,n}$ be the vector of residuals resulting from the weighted least-squares regression of the constant vector $\mathbb{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$ on the control variates but

Algorithm 3.4 Adaptive Importance Sampling with Control Variates (AISCV)

Require: integrand f, target density π (up to a proportionality constant), number of stages $T \in \mathbb{N}^*$, allocation policy $(n_t)_{t=1}^T$, initial density q_0, update rule for the sampling policy

1: **for** $t = 1, \dots, T$ **do**
2: Generate an independent random sample $X_{t,1}, \dots, X_{t,n_t}$ from q_{t-1}
3: Compute the vector of weights $(w_{t,i})_{i=1}^{n_t}$ where $w_{t,i} = \pi(X_{t,i})/q_{t-1}(X_{t,i})$
4: Construct the matrix of control variates $H_t = \Big(h_j(X_{t,i})\Big)_{i=1,\dots,n_t}^{j=1,\dots,m}$
5: Evaluate the integrand in the particles: $(f(X_{t,i}))_{i=1}^{n_t}$
6: Update the sampler q_t based on all previous particles $(X_{s,i} : s = 1, \dots, t; i = 1, \dots, n_s)$
7: **end for**
8: Compute $(\hat{\alpha}_T, \hat{\beta}_T) = \underset{(a,b)\in\mathbb{R}\times\mathbb{R}^m}{\arg\min} \sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} \Big(f(X_{t,i}) - a - b^\top h(X_{t,i})\Big)^2$
9: **return** $\alpha_n^{(\mathrm{aiscv})}(f) = \hat{\alpha}_T$.

without intercept:

$$\begin{aligned} e_{n,i} &= 1 - \hat{\beta}_n(\mathbb{1}_n)^\top h(X_i) \qquad \text{where} \\ \hat{\beta}_n(\mathbb{1}_n) &\in \underset{b\in\mathbb{R}^m}{\arg\min} \sum_{i=1}^n w_i \Big(1 - b^\top h(X_i)\Big)^2. \end{aligned} \tag{3.8}$$

Even though the vector $\hat{\beta}_n(\mathbb{1}_n)$ is not necessarily unique, the weighted least squares fit $(\hat{\beta}_n(\mathbb{1}_n)^\top h(X_i))_{i=1,\dots,n}$ always is. According to the next proposition, the quadrature weights are proportional to $(w_i e_{n,i})_{i=1,\dots,n}$.

Proposition 3.2 (AISCV quadrature rule)**.** *The minimizer $\hat{\alpha}_n$ in (3.7) is unique if and only if $e_n \neq 0$ in (3.8). In that case, the AISCV estimate is*

$$\alpha_n^{(\mathrm{aiscv})}(f) = \hat{\alpha}_n = \frac{\sum_{i=1}^n w_i e_{n,i} f(X_i)}{\sum_{i=1}^n w_i e_{n,i}}. \tag{3.9}$$

If $e_n = 0$, then there exists $b \in \mathbb{R}^m$ such that $b^\top h(X_i) = 1$ for all $i = 1, \dots, n$. In that case, the minimizer $\hat{\alpha}_n$ in (3.7) is not unique and the AISCV estimate is not well-defined. To remedy this, one can for instance reduce the number of control variates. This issue already occurs with the ordinary control variate estimator in (3.3).

Rather than requiring a different weighted least squares problem for every integrand f as in (3.7), the quadrature rule in (3.9) only involves a single weighted least squares problem (3.8), whatever f. Given the quadrature weights, calculating the AISCV estimate for a novel integrand only requires the evaluations of that function on the sampled particles, making the whole procedure a *post-hoc* scheme. The steps in case the sampling policy is divided into T stages are given in Algorithm 3.5, which gives the same result as Algorithm 3.4, but with less effort if multiple integrands f are into play.

Algorithm 3.5 Quadrature Rule – AISCV *post-hoc* scheme

Require: integrand f, $T \in \mathbb{N}^*$, allocation policy $(n_t)_{t=1}^T$, weights $(w_t)_{t=1}^T$ with $w_t = (w_{t,i})_{i=1}^{n_t}$, matrices $(H_t)_{t=1}^T$ with $H_t = \Big(h_j(X_{t,i})\Big)_{i=1,\dots,n_t}^{j=1,\dots,m}$, particles $(X_{t,i} : t = 1,\dots,T; i = 1,\dots,n_t)$

1: Compute $\hat{\beta}_n(\mathbb{1}_n) = \arg\min_{b\in\mathbb{R}^m} \sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} \Big(1 - b^\top h(X_{t,i})\Big)^2$

2: Compute $u_t = \mathrm{diag}(w_t)[\mathbb{1}_{n_t} - H_t\hat{\beta}_n(\mathbb{1}_n)]$ for $t = 1,\dots,T$

3: Compute $s = \sum_{t=1}^T \sum_{i=1}^{n_t} u_{t,i}$

4: Compute weights $v_{t,i} = u_{t,i}/s$ for $t = 1,\dots,T$ and $i = 1,\dots,n_t$

5: **return** $\alpha_T^{(\mathrm{aiscv})}(f) = \sum_{t=1}^T \sum_{i=1}^{n_t} v_{t,i} f(X_{t,i})$

3.4 Theoretical properties of the AISCV estimate

Here we point out several theoretical properties of the novel AISCV estimate. A first point is that the integration rule is exact on the linear span of the control variates and the constant function.

Proposition 3.3 (Exact integration)**.** *For integrands of the form $f = \alpha + \beta^\top h$ for $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, the AISCV estimate is exact: $\alpha_n^{(\mathrm{aiscv})}(f) = \alpha = \mathbb{E}_\pi[f]$.*

A second property is that we may apply arbitrary invertible linear transformations to the control variates without changing the AISCV estimate. This can be advantageous computationally, to make the underlying weighted least squares problem more stable numerically. Also, it means that without loss of generality, we may assume that the control variates are uncorrelated and have unit variance, which simplifies the theoretical performance analysis.

Proposition 3.4 (Invariance)**.** *If the matrix $A \in \mathbb{R}^{m\times m}$ is invertible, then the AISCV estimate based on the control variates Ah is the same as the one based on h.*

Our main result is a non-asymptotic bound on the error of the AISCV estimate for $\int f\pi \,\mathrm{d}\lambda$ when $\int f^2\pi \,\mathrm{d}\lambda$ is finite. First, we introduce some assumptions and definitions.

The first condition that is required concerns the policy given by the AIS part of the algorithm. It is supposed that any element from the policy should dominate the function π.

Assumption 3.5 (Dominated measures)**.** *There exists $c \geq 1$ such that, for all $x \in \mathbb{R}^d$ and for any $i = 1,\dots,n$, we have $\pi(x) \leq c \cdot q_i(x)$.*

This assumption represents a *safe* approach to importance sampling, as the policy will always allow to sample in places where π is positive. A well-known and well-spread (Hesterberg, 1995; Owen and Zhou, 2000; Delyon and Portier, 2021) technique to achieve such a defensive strategy is to a use mixture density $q_i = (1-\eta)\pi_i + \eta q_0$ where $\eta \in (0,1)$ and where q_0 has sufficiently heavy tails to dominate π. Such a mixture allows to choose the densities f_i with some flexibility using in principle any AIS algorithm. Second, the control variates shall be linearly independent and bounded.

Assumption 3.6 (Control variates). *We have* $\sup_{x:\pi(x)>0} |h_j(x)| < \infty$ *for all* $j = 1, \dots, m$. *The matrix* $G = \int hh^\top \pi \, \mathrm{d}\lambda$ *is invertible.*

The previous condition allows to define the standardized vector of control variates as $\hbar = G^{-1/2}h$. By Proposition 3.4, this change does not affect the AISCV estimate. The orthonormal control variates $\hbar$ will play a key role through the following quantity

$$B = \sup_{x:\pi(x)>0} \|\hbar(x)\|_2^2.$$

The quadratic form $\|\hbar(x)\|_2^2 = h(x)^\top G^{-1} h(x)$ is referred to as the *leverage function* in ordinary linear regression as it quantifies the influence of a training point x on the prediction of the observed response. It is invariant with respect to invertible linear transformations of the control variate vector.

Assumption 3.6 and the fact that the integrand g is square integrable with respect to π allows to define the residual function $\varepsilon = f - \int f\pi \, \mathrm{d}\lambda - h^\top \beta^*$ where β^* has been introduced in (3.4) as a minimizer of the residual variance. Since we work in the space $L^2(\pi)$, we assume without loss of generality that f and h vanish outside $\{x : \pi(x) > 0\}$ and we put $\varepsilon(x) = 0$ for $x \in \mathbb{R}^d$ such that $\pi(x) = 0$. The residual function ε should satisfy the following tail condition.

Assumption 3.7 (Residual tail). *There exists* $\tau > 0$ *such that, for all* $t > 0$ *and all integer* $i \geq 1$, *we have* $\mathbb{P}[|w_i \varepsilon(X_i)| > t \mid \mathcal{F}_{i-1}] \leq 2\exp(-t^2/(2\tau^2))$.

The previous assumption concerns both the function ε and the policy sequence $(q_i)_{i\geq 0}$. Since $\mathbb{E}[w_i\varepsilon(X_i) \mid \mathcal{F}_{i-1}] = 0$, it is implied by the so-called sub-Gaussian condition (Boucheron et al., 2013b) that $\mathbb{E}[\exp(\lambda w_i \varepsilon(X_i)) \mid \mathcal{F}_{i-1}] \leq \exp(-\lambda^2\tau^2/2)$ for any $\lambda \in \mathbb{R}$. In the proof of Theorem 3.8, Assumption 3.7 allows to derive concentration bounds on residual-based sums using recent results from Jin et al. (2019); Leluc et al. (2021). We are now in position to state our main result on the error of the AISCV estimate.

Theorem 3.8 (Concentration inequality for AISCV estimate). *If Assumptions 3.5, 3.6 and 3.7 hold, then, for any* $\delta \in (0,1)$ *and for all* $n \geq C_1 c^2 B \log(10m/\delta)$, *we have, with probability at least* $1-\delta$, *that*

$$\left|\alpha_n^{(\mathrm{aiscv})}(f) - \pi(f)\,\mathrm{d}x\right| \leq C_2\tau\sqrt{\frac{\log(10/\delta)}{n}} + C_3 cB\tau\frac{\log(10m/\delta)}{n},$$

where C_1, C_2, C_3 *are universal constants specified in the proof.*

Remark 3.9 (Understanding τ). *The quantity* τ *in Assumption 3.7 is related to the conditional variance* $\mathbb{E}[w_i^2\varepsilon^2(X_i) \mid \mathcal{F}_{i-1}]$. *They actually coincide when* $w_i\varepsilon(X_i)$ *is Gaussian. For a policy satisfying Assumption 3.5,* $\mathbb{E}[w_i^2\varepsilon^2(X_i) \mid \mathcal{F}_{i-1}] \leq c\sigma_m^2$ *which for certain combinations of integrands and control functions scales as* $m^{-s/d}$ *(Portier and Segers, 2019) where the parameter* s *represents the degree of smoothness of* f.

Remark 3.10 (Convergence rates). *Consider an asymptotic regime where the number of control variates* m *tends to infinity with the sample size* n. *The AISCV estimate improves upon the AIS method* ($m = 0$), *which has rate* $1/\sqrt{n}$, *as soon as* $\tau + \tau B\log(m)/\sqrt{n} \to 0$. *To recover the same order of an oracle estimate with rate* $\tau/\sqrt{n}$, *one must have* $B\log(m) = O(\sqrt{n})$ *as* $n \to \infty$.

3.5 Practical considerations

This section presents several ways to build control variates from a practical point of view using either families of polynomials or general functions based on Stein's method. Next, some computations are highlighted in the framework of Bayesian inference.

3.5.1 Control variate constructions

Orthogonal polynomials. When the target density π can be decomposed as a product of univariate densities $\pi = p_1 \otimes \cdots \otimes p_d$, multidimensional control functions may be constructed based on univariate ones. This happens for instance for the uniform distribution over the unit cube $[0,1]^d$ or with uncorrelated Gaussian distributions on $\mathbb{R}^d$. Such univariate control variates may be easily constructed using families of polynomials (Gautschi, 2004), such as Legendre polynomials for the uniform distribution on $[0,1]$ and Hermite polynomials for the Gaussian distribution on $\mathbb{R}$. This technique can also be used when π is dominated by another density π^* having the said product form by transforming zero-mean control variates h^* with respect to π^* via $h = h^*\pi^*/\pi$.

Let $(h_1, \ldots, h_k)$ be a vector of univariate control functions with respect to a density p, i.e., $\mathbb{E}_p[h_j] = 0$ for all $j = 1, \ldots, k$. Let $h_0 = 1$ denote the constant function equal to one. For a multi-index $\ell = (\ell_1, \ldots, \ell_d)$ in $\{0, \ldots, k\}^d \setminus \{(0, \ldots, 0)\}$, multivariate controls with respect to $p^{\otimes d}$ are built by forming tensor products of the form $h_\ell(x_1, \ldots, x_d) = h_{\ell_1}(x_1) \cdots h_{\ell_d}(x_d)$, yielding a total number of $m = (k+1)^d - 1$ control functions. Alternative approaches yielding smaller control spaces consist of imposing $\ell_j = 0$ for all but a small number of coordinates $j = 1, \ldots, d$ or by the constraint $\ell_1 + \cdots + \ell_d \leq Q$ for some $Q \geq 1$.

Stein control variates. In the general case where one has only access to the evaluations of π, control variates may be constructed using Stein's method. The technique relies on the gradient $\nabla_x \log \pi(x)$ which can either be directly computed (see the example of Bayesian regression below) or which may be available through automatic differentiation provided in popular API's such as Tensorflow and PyTorch (Abadi et al., 2016; Paszke et al., 2017). Let $\Delta_x = \nabla_x^\top \nabla_x$ denote the Laplace operator. By definition, the second-order Stein operator $\mathcal{L}$ (Stein, 1972; Gorham and Mackey, 2015) associated to the density π is defined by:

$$\forall \varphi \in \mathcal{C}^2(\mathbb{R}^d, \mathbb{R}), \quad (\mathcal{L}\varphi)(x) = \Delta_x \varphi(x) + \nabla_x \varphi(x)^\top \nabla_x \log \pi(x).$$

The transformation guarantees that $\mathbb{E}_\pi[\mathcal{L}\varphi] = 0$ for all φ with weak regularity conditions (Mira et al., 2013). Therefore, we can build infinitely many control variates $h_\varphi = \mathcal{L}\varphi$ from given functions φ. One simple way is to let φ be a polynomial with bounded total degree: for a degree vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ with $\alpha_1 + \cdots + \alpha_d \leq Q$, define $\varphi_{\boldsymbol{\alpha}}(x) = x_1^{\alpha_1} \cdots x_d^{\alpha_d}$. Given the dimension d and the total degree Q, there are $m = \binom{d+Q}{d} - 1$ such degree vectors, yielding the associated control variates $h_{\boldsymbol{\alpha}} = h_{\varphi_{\boldsymbol{\alpha}}}$. For fast computation, note that, writing $\phi_{\boldsymbol{\alpha}}(x) = \varphi_{\boldsymbol{\alpha}}(x)\mathbb{1}_d$, $D_1(x) = \mathrm{diag}(\alpha_1/x_1, \ldots, \alpha_d/x_d)$ and $D_2(x) = \mathrm{diag}(\alpha_1(\alpha_1-1)/x_1^2, \ldots, \alpha_d(\alpha_d-1)/x_d^2)$, we have $\nabla_x \varphi_{\boldsymbol{\alpha}}(x) = D_1(x)\phi_{\boldsymbol{\alpha}}(x)$ and $\Delta_x \varphi_{\boldsymbol{\alpha}}(x) = \mathbb{1}_d^\top (D_2(x)\phi_{\boldsymbol{\alpha}}(x))$. In practice, all combinations of $\boldsymbol{\alpha}$ are stored in a matrix $A \in \mathbb{N}^{m \times d}$.

3.5.2 Bayesian inference

Given data $\mathcal{D}$ and a parameter of interest $\theta \in \Theta \subset \mathbb{R}^d$, posterior integrals take the form $\int_{\mathbb{R}^d} f(\theta)p(\theta|\mathcal{D})\,\mathrm{d}\theta$, where $p(\theta|\mathcal{D}) \propto \ell(\mathcal{D}|\theta)p(\theta)$ is the posterior distribution, proportional to a prior $p(\theta)$ and a likelihood function $\ell(\mathcal{D}|\theta)$. For instance, when $f(\theta) = \theta$, the integral above recovers the posterior mean. Stein control variates involve the computation of the gradient of the log-posterior $\nabla_\theta \log p(\theta|\mathcal{D})$, which implicitly relies on the score function $\nabla_\theta \log \ell(\mathcal{D}|\theta)$. We point out two common examples—linear and logistic regression—where these functions are easy to compute.

Bayesian linear regression. Consider a linear regression problem comprised of observations $X \in \mathbb{R}^{N\times d}$ with labels $y \in \mathbb{R}^N$. In the Gaussian fixed design setting, the predictor x_i produces the response $y_i = x_i^\top\theta + \varepsilon_i$ where $\varepsilon_1, \dots, \varepsilon_N \sim \mathcal{N}(0, \sigma^2)$ are centered Gaussian noises. The likelihood $\ell(X, y|\theta)$ is proportional to $(\sigma^2)^{-N/2}\exp(-(y-X\theta)^\top(y-X\theta)/(2\sigma^2))$, yielding the score function $\nabla_\theta \log \ell(X, y|\theta) = X^\top(y-X\theta)/(2\sigma^2)$.

Bayesian logistic regression. Next, consider the logistic regression problem comprised of observations $X \in \mathbb{R}^{N\times d}$ with associated binary labels $y \in \{0, 1\}^N$. Letting $\sigma(s) = 1/(1 + e^{-s})$ denote the sigmoid function, the likelihood function is $\ell(X, y|\theta) = \prod_{i=1}^N \sigma(\theta^\top x_i)^{y_i}(1-\sigma(\theta^\top x_i))^{1-y_i}$. The score function is simply $\nabla_\theta \log \ell(X, y|\theta) = X^\top(y-\sigma(X\theta))$.

3.6 Numerical illustration

To compare the finite-sample performance of the AIS and AISCV estimators, we first present in Section 3.6.1 synthetic data examples involving the integration problem over the unit cube $[0, 1]^d$ and then with respect to some Gaussian mixtures as in Cappé et al. (2008). The goal is to compute $\int f\pi\,\mathrm{d}\lambda$ for vectors of integrands $f : \mathbb{R}^d \to \mathbb{R}^p$. We consider various dimensions $d > 1$ and several choices for the number of control variates m. Section 3.6.2 deals with real-world datasets in the context of Bayesian inference. For ease of reproducibility, the code is available online[1] and numerical details with additional results are available in Section 3.D.

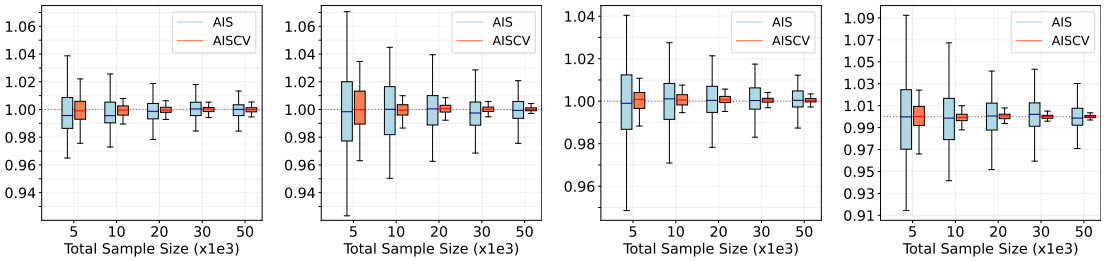
Parameters. In all simulations, the sampling policy is taken within the family of multivariate Student t distributions of degree ν denoted by $\{q_{\mu,\Sigma_0} : \mu \in \mathbb{R}^d\}$ with $\Sigma_0 = \sigma_0 I_d(\nu - 2)/\nu$ and $\nu > 2, \sigma_0 > 0$. Similarly to Portier and Delyon (2018), the mean μ_t is updated at each stage $t = 1, \dots, T$ by the generalized method of moments (GMM), leading to $\mu_t = (\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i}X_{s,i})/(\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i})$. The allocation policy is fixed to $n_t = 1000$ and the number of stages is $T \in \{5; 10; 20; 30; 50\}$. The different Monte Carlo estimates are compared by their mean squared error (MSE) obtained over 100 independent replications.

3.6.1 Synthetic examples

Integration on $[0, 1]^d$. We seek to integrate functions f with respect to the uniform density $\pi(x) = 1$ for $x \in [0, 1]^d$ in dimensions $d \in \{4; 8\}$. We rely on Legendre polynomials for the control variates. Consider the integrands $f_1(x) = 1+\sin(\pi(2d^{-1}\sum_{i=1}^d x_i - 1))$,

[1] https://github.com/RemiLELUC/AISCV

$f_2(x) = \prod_{i=1}^d (2/\pi)^{1/2} x_i^{-1} \mathrm{e}^{-\log(x_i)^2/2}$ and $f_3(x) = \prod_{i=1}^d \log(2) 2^{1-x_i}$, all of which integrate to 1 on $[0,1]^d$. None of the integrands is a linear combination of the control variates. The policy parameters are $\mu_0 = (0.5, \dots, 0.5) \in \mathbb{R}^d$, $\nu = 8$, and $\sigma_0 = 0.1$. The control variates are built out of tensor products of Legendre polynomials where the degree ℓ_j equals 0 for all but two coordinates, leading to a total number of $m = kd + k^2 d(d-1)/2$ control variates. The maximum degree in each variable is $k = 6$, yielding $m = 240$ and $m = 1056$ control variates in dimensions $d = 4$ and $d = 8$ respectively. Figure 3.1 presents the boxplots of the AIS and AISCV estimates. The error reduction obtained thanks to the control variates is huge: the AISCV estimate has a mean squared error smaller than the one of the AIS estimate by a factor at least 10 and up to 100 (see Table 3.1 in the Section 3.D).



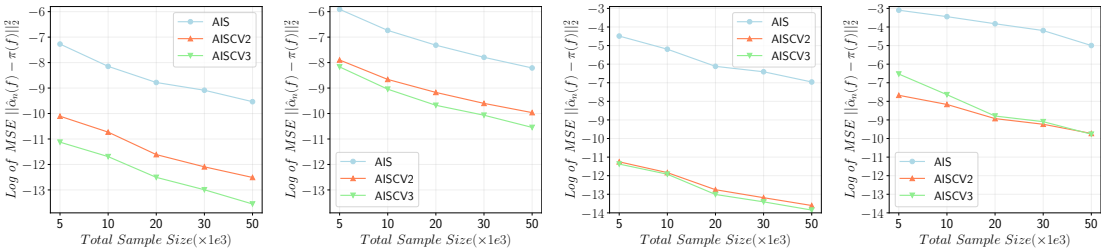
(a) $f_1(d = 4)$ (b) $f_1(d = 8)$ (c) $f_2(d = 4)$ (d) $f_3(d = 8)$

Figure 3.1 – Integration on $[0,1]^d$: boxplots of estimates $\alpha_n^{(\mathrm{ais})}(f)$ and $\alpha_n^{(\mathrm{aiscv})}(f)$ with integrands f_1, f_2, f_3 in dimensions $d \in \{4; 8\}$ obtained over 100 replications. The true integral equals 1.

Gaussian mixture f and Stein control variates. In this setting we assume we only have access to the evaluations of the target density f. We consider the classical example introduced in Cappé et al. (2008) where f is a mixture of two Gaussian distributions. The control variates are built using Stein's method (Section 3.5.1) out of polynomials of total degree at most $Q \in \{2; 3\}$, leading to a number of control variates $m \in \{14; 34\}$ in dimension $d = 4$ and $m \in \{44; 164\}$ in dimension $d = 8$ respectively. We consider two cases: an isotropic and an anisotropic one.

Isotropic case. Let $\pi_\Sigma(x) = 0.5\Phi_\Sigma(x-\mu) + 0.5\Phi_\Sigma(x+\mu)$ where $\mu = (1, \dots, 1)^\top / 2\sqrt{d}, \Sigma = I_d/d$ and Φ_Σ is the multivariate normal density function with zero mean and covariance matrix Σ. The Euclidean distance between the two mixture centers is 1, independently of d. The initial density q_0 is the multivariate Student t distribution with mean $(1, -1, 0, \dots, 0)/\sqrt{d}$ and variance $(5/d)I_d$. The initial mean value differs from the null vector to prevent the naive algorithm using the initial density from having good results due to the symmetrical set-up.
Anisotropic case. In this case, the mixture is unbalanced and each Gaussian is anisotropic. The target density is $\pi_V(x) = 0.75\Phi_V(x - \mu) + 0.25\Phi_V(x + \mu)$ where $\mu = (1, \dots, 1)^\top / 2\sqrt{d}$ and $V = \mathrm{diag}(10, 1, \dots, 1)/d$. The initial density q_0 is the same as for the isotropic case.

Figure 3.2 presents the evolution of the logarithm of the mean squared error $\|\hat{\alpha}_n(f) - \pi(f)\|_2^2$. Once again, the AISCV estimators are the clear winners with a mean squared error smaller by a factor up to 1000 for the anisotropic case (see Table 3.2 in Section 3.D).



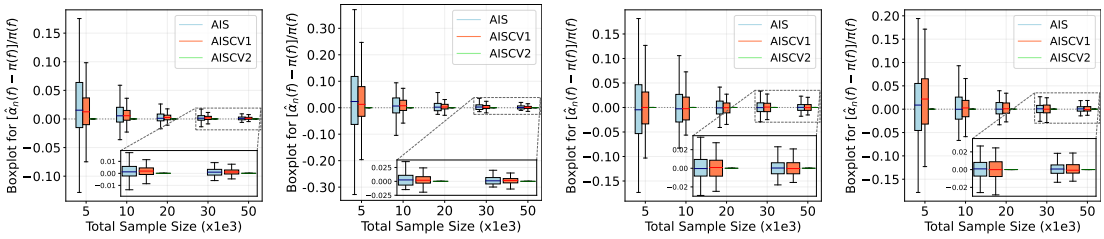
(a) $\pi_\Sigma(d=4)$ (b) $\pi_\Sigma(d=8)$ (c) $\pi_V(d=4)$ (d) $\pi_V(d=8)$

Figure 3.2 – Gaussian mixture density: Logarithm of $\|\hat{\alpha}_n(f) - \pi(f)\|_2^2$ for $f(x) = x$ with target isotropic π_Σ and anisotropic π_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

3.6.2 Real-world examples

We place ourselves in the framework of Bayesian linear regression (Section 3.5.2) with features $X \in \mathbb{R}^{N\times d}$ and continuous responses $y \in \mathbb{R}^N$. The posterior distribution $p(\theta|\mathcal{D})$ involves a Gaussian prior $p(\theta) \sim \mathcal{N}(\mu_a, \Sigma_a)$ and a likelihood function $\ell(\mathcal{D}|\theta)$ proportional to $(\sigma^2)^{-N/2}\exp(-(y - X\theta)^\top(y - X\theta)/(2\sigma^2))$. The noise level is fixed and taken sufficiently large at $\sigma = 50$ to account for general priors. The posterior distribution is Gaussian too: $\mathcal{N}(\mu_b, \Sigma_b)$ with $\mu_b = \Sigma_b(\sigma^{-2}X^\top y + \Sigma_a^{-1}\mu_a)$ and $\Sigma_b = (\sigma^{-2}X^\top X + \Sigma_a^{-1})^{-1}$. The integrand is $f(\theta) = \sum_{i=1}^d \theta_i^2$ and the control variates are built with the Stein operator (Section 3.5.1) out of monomials with total degree $Q \in \{1; 2\}$, leading to the AISCV1 and AISCV2 estimators respectively.

Datasets and parameters. Classical datasets from Dua and Graff (2019) are considered : *housing* ($N = 506; d = 13; m \in \{12; 104\}$); *abalone* ($N = 4177; d = 8; m \in \{7; 44\}$); *red wine* ($N = 1599; d = 11; m \in \{10; 77\}$); and *white wine* ($N = 4898; d = 11; m \in \{10; 77\}$). The initial density is the multivariate Student t distribution with $\nu = 10$ degrees of freedom, zero mean and covariance matrix Σ_b.



(a) Housing (b) Abalone (c) Red Wine (d) White Wine

Figure 3.3 – Bayesian linear regression: boxplots of $(\hat{\alpha}_n(f) - \pi(f))/\pi(f)$ for $f(\theta) = \sum_{j=1}^d \theta_j^2$.

Results. Figure 3.3 presents the boxplots of the relative error $(\hat{\alpha}_n(f) - \pi(f))/\pi(f)$, revealing the benefits of control variates even with polynomials of degree $Q = 1$. When $Q = 2$, the error of the AISCV2 estimate is virtually zero (see Table 3.3 in the supplement), in line with Proposition 3.3. The mean squared error of the AISCV1 estimate is smaller than that of the AIS estimate by a factor ranging between 2 and 10.

3.7 Discussion

While control variates are a well-known tool for Monte Carlo integration, standard methods do not allow the distribution of particles to evolve throughout the algorithm, as is the case for sequential methods. Within the standard adaptive importance sampling framework, we have developed a weighted least-squares procedure to improve numerical integration by incorporating control variates. The underlying adapted weights of this quadrature rule do not depend on the integrand and our non-asymptotic bound highlights the benefits of combining adaptive importance sampling with control variates. Different ways for constructing control variates are proposed. The method is fit for computing integrals with respect to the posterior density in Bayesian analysis, as the target density only needs to be known up to a multiplicative constant.

A limitation of the combined AISCV approach is that it requires the user to make quite some design choices, notably the sampling policy for the AIS part and the control variates for the CV part. These culminate into the factor τ in Assumption 3.7, which appears prominently in the error bound in Theorem 3.8 and which can be interpreted roughly as the standard deviation of $w\varepsilon$, where w is the importance weight – well behaved when the policy is well-chosen in relation to the target density – and where ε is some residual function – well behaved when the control variates are well-chosen with respect to the integrand. Further, choosing too many control variates may result in an ill-conditioned empirical Gram matrix or in overfitting. The least-squares solution could become unstable, requiring some kind of regularization, such as the LASSO (Leluc et al., 2021).

Technical Lemmas and auxiliary results are provided in Appendix 3.A. Section 3.B collects additional theoretical properties of the AISCV estimator while the technical proofs of the Propositions and main theorem are presented in Section 3.C. Finally, Section 3.D presents additional numerical values associated to the numerical experiments on synthetic examples and real-world datasets for Bayesian linear regression.

3.A Auxiliary results

3.A.1 Lemmas on (Random) Matrices inequalities

Definition 3.11. *Let A and Ψ be Hermitian matrices of the same dimension. We say that $A \preceq \Psi$ if and only if $\Psi - A$ is positive semidefinite.*

Definition 3.12 (Tropp (2015), Definition 2.1.2)**.** *Let $f : I \to \mathbb{R}$ where I is an interval of the real line. Consider a $d \times d$ Hermitian matrix A whose eigenvalues are contained in I. Define a $d \times d$ Hermitian matrix $f(A)$, called the standard matrix function, using an eigenvalue decomposition of A, by*

$$f(A) = Q \begin{bmatrix} f(\lambda_1) & & \\ & \ddots & \\ & & f(\lambda_d) \end{bmatrix} Q^* \quad \text{where} \quad A = Q \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix} Q^*.$$

Remark 3.13. *The matrix exponential e^A and the matrix logarithm $\log(A)$ are the standard matrix functions.*

Lemma 3.14 (Tropp (2015), Example 8.3.4)**.** *The trace exponential map is monotone:*

$$A \preceq \Psi \text{ implies } \operatorname{Tr} e^{A} \leq \operatorname{Tr} e^{\Psi}$$

for all Hermitian matrices A and Ψ.

Lemma 3.15 (Tropp (2015), Proposition 3.2.1)**.** *For any random Hermitian matrix Y, for all $t \in \mathbb{R}$, we have*

$$\mathbb{P}\left(\lambda_{\min}(Y) \leq t\right) \leq \inf_{\theta<0} e^{-\theta t}\mathbb{E}[\operatorname{Tr}(e^{\theta Y})].$$

Lemma 3.16 (Tropp (2015), Lemma 5.4.1)**.** *Assume that A is a random matrix with $0 \leq \lambda_{\min}(A)$ and, for some constant $L > 0$, $\lambda_{\max}(A) \leq L$. Then, for all $\theta \in \mathbb{R}$,*

$$\log(\mathbb{E}[e^{\theta A}]) \preceq \eta(\theta)\mathbb{E}[A], \quad \eta(\theta) = L^{-1}(e^{\theta L} - 1).$$

Lemma 3.17 (Tropp (2015), Corollary 3.4.2)**.** *Let Ψ be a fixed Hermitian matrix and A a random Hermitian matrix of the same dimension. Then*

$$\mathbb{E}\left[\operatorname{Tr}\{\exp(\Psi + A)\}\right] \leq \operatorname{Tr}\left[\exp\{\Psi + \log(\mathbb{E}[e^{A}])\}\right].$$

3.A.2 Inequalities for martingales increments and empirical Gram matrices

Lemma 3.18 (Hoeffding inequality for norm-subGaussian martingale increments)**.** *Let the d-dimensional random vectors $Z_1, \ldots, Z_n$ and the natural filtration $\mathcal{F}_n = \sigma(Z_1, \ldots, Z_n)$, $\mathcal{F}_0 = \{\Omega, \emptyset\}$, be such that, for all $i = 1, \ldots n$, $\mathbb{E}[Z_i|\mathcal{F}_{i-1}] = 0$ and*

$$\forall t \geq 0, \forall i = 1, \ldots, n, \qquad \mathbb{P}(\left\|Z_i\right\|_2 \geq t|\mathcal{F}_{i-1}) \leq 2\exp\left(-\tfrac{t^2}{2\sigma^2}\right) \tag{3.10}$$

for some $\sigma > 0$. Then for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\left\|\textstyle\sum_{i=1}^{n} Z_i\right\|_2 \leq K\sigma\sqrt{n\log(2d/\delta)},$$

where $K = 3$.

Proof The proof follows from adapting the proof of Lemma 6 in Leluc et al. (2021) working out their Lemma 5 and Corollary 7 from Jin et al. (2019). ■

Lemma 3.19. *Define $h_k = h(X_k)$, $Q_k = w_k h_k h_k^\top$, $Y_n = \sum_{k=1}^{n} Q_k$. Let the constant $L > 0$ be such that $\lambda_{\max}(Q_k) \leq L$ with probability 1. Then, for all $\zeta \in (0, 1)$, we have*

$$\mathbb{P}\left(\lambda_{\min}(Y_n) \leq (1-\zeta)n\lambda_{\min}(G)\right) \leq m\left[\frac{e^{-\zeta}}{(1-\zeta)^{(1-\zeta)}}\right]^{n\lambda_{\min}(G)/L}.$$

Remark 3.20. *The term in square brackets in Proposition 3.19 is bounded above by $e^{-\zeta^2/2}$ (Leluc et al. (2021), Lemma 2).*

Proof Let $\mathbb{E}_n$ denote the expectation with respect to $\mathcal{F}_{n-1} = \sigma(X_1, \dots, X_{n-1})$ and define $Z_n = \log(\mathbb{E}_n[e^{\nu Q_n}])$. Using Lemma 3.17 with the measurable w.r.t. $\mathcal{F}_{n-1}$ matrix $\Psi = \nu Y_{n-1}$, we have

$$\mathbb{E}\left[\mathrm{Tr}(e^{\nu Y_n})\right] = \mathbb{E}\left[\mathbb{E}_n\left[\mathrm{Tr}(e^{\nu Y_{n-1}+\nu Q_n})\right]\right] \leq \mathbb{E}\left[\mathrm{Tr}(e^{\nu Y_{n-1}+\log(\mathbb{E}_n[e^{\nu Q_n}])})\right] = \mathbb{E}\left[\mathrm{Tr}(e^{\nu Y_{n-1}+Z_n})\right].$$

Using again Lemma 3.17 with the matrix $\Psi = \nu Y_{n-2} + Z_n$, the last term is upper bounded as

$$\mathbb{E}\left[\mathrm{Tr}(e^{\nu Y_{n-1}+Z_n})\right] = \mathbb{E}\left[\mathbb{E}_{n-1}\left[\mathrm{Tr}(e^{\nu Y_{n-2}+\nu Q_{n-1}+Z_n})\right]\right] \leq \mathbb{E}\left[\mathrm{Tr}(e^{\nu Y_{n-2}+Z_{n-1}+Z_n})\right]$$

Applying this inequality several times yields

$$\mathbb{E}[\mathrm{Tr}(e^{\nu Y_n})] \leq \mathbb{E}[\mathrm{Tr}(e^{\sum_{k=1}^n Z_k})].$$

Applying Lemma 3.16 gives $Z_k \preceq \eta(\nu)\mathbb{E}_k[Q_k]$, $\eta(\nu) = L^{-1}(e^{\nu L} - 1)$ for $k = 1, \dots, n$. By Lemma 3.14, we get

$$\mathbb{E}[\mathrm{Tr}(e^{\nu Y_n})] \leq \mathbb{E}[\mathrm{Tr}(e^{\sum_{k=1}^n Z_k})] \leq \mathbb{E}[\mathrm{Tr}(e^{\sum_k \eta(\nu)\mathbb{E}_k[Q_k]})] = \mathrm{Tr}(e^{n\eta(\nu)G}).$$

Now applying Lemma 3.15 and taking into account the fact that $\eta(\nu) < 0$ for $\nu < 0$, we have

$$\begin{aligned}
\mathbb{P}\left(\lambda_{\min}(Y_n) \leq t\right) &\leq \inf_{\nu<0} e^{-\nu t}\mathbb{E}[\mathrm{Tr}(e^{\nu Y_n})] \\
&\leq \inf_{\nu<0} e^{-\nu t}\,\mathrm{Tr}(e^{n\eta(\nu)G}) \\
&\leq \inf_{\nu<0} e^{-\nu t}\,\mathrm{Tr}(e^{n\eta(\nu)\lambda_{\min}(G)I_m}) \\
&\leq \inf_{\nu<0} e^{-\nu t} m e^{n\eta(\nu)\lambda_{\min}(G)}.
\end{aligned}$$

We make the change of variables $t = (1-\zeta)n\lambda_{\min}(G)$ and minimize over $\nu < 0$ the following expression

$$-n\nu(1-\zeta)\lambda_{\min}(G) + n\eta(\nu)\lambda_{\min}(G).$$

The infimum is attained at $\nu = L^{-1}\log(1-\zeta)$ with $\eta(\nu) = -\zeta/L$ which gives the inequality of the Lemma. ■

3.B Additional properties of AISCV estimator

3.B.1 Orthogonal projections

Some geometric considerations help to better understand certain properties of the AISCV estimate (3.7). Let $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$ be a vector of ones and write

$$f^{(n)} = (f(X_1), \dots, f(X_n))^\top, \quad H = (h_j(X_i))_{\substack{i=1,\dots,n \\ j=1,\dots,m}}, \quad \text{and } W = \mathrm{diag}(w_1, \dots, w_n).$$

In matrix form, the weighted least-squares problem (3.7) is

$$(\hat{\alpha}_n, \hat{\beta}_n) \in \underset{(a,b)\in\mathbb{R}\times\mathbb{R}^m}{\arg\min} \|W^{1/2}(f^{(n)} - a\mathbf{1}_n - Hb)\|_2^2. \tag{3.11}$$

For any function $\varphi : \mathbb{R}^d \to \mathbb{R}^p$, let the operator $P_{n,w}$ return the weighted average of the sequence $\varphi(X_1), \dots, \varphi(X_n)$ with the weights $w_1, \dots, w_n$, i.e.,

$$P_{n,w}(\varphi) = \frac{\sum_{i=1}^n w_i \varphi(X_i)}{\sum_{i=1}^n w_i}.$$

The empirically centred integrand and control variates are $f_W^{(n)} = f^{(n)} - \mathbf{1}_n P_{n,w}(f)$ and $H_W = H - \mathbf{1}_n P_{n,w}(h^\top)$. Put $W^{1/2} = \mathrm{diag}(w_1^{1/2}, \dots, w_n^{1/2})$. The solution to (3.11) takes the form

$$\begin{cases} \hat{\alpha}_n = P_{n,w}(f - \hat{\beta}_n^\top h), \\ \hat{\beta}_n \in \arg\min_{b \in \mathbb{R}^m} \|W^{1/2}(f_W^{(n)} - H_W b)\|_2^2, \end{cases} \tag{3.12}$$

If the matrix $H_W^\top W H_W$ is invertible, the optimal vector $\hat{\beta}_n$ is unique and is given by

$$\hat{\beta}_n = (H_W^\top W H_W)^{-1} H_W^\top W f_W^{(n)}. \tag{3.13}$$

3.B.2 Matrix representation

Let us rewrite (3.11) in terms of two nested minimization problems:

$$\hat{\alpha}_n \in \arg\min_{a \in \mathbb{R}} \left[\min_{b \in \mathbb{R}^m} \left\| W^{1/2} \left(f^{(n)} - a\mathbf{1}_n - Hb \right) \right\|_2^2 \right]. \tag{3.14}$$

Let $\Pi \in \mathbb{R}^{n \times n}$ be the orthogonal projection matrix onto the column space of H, when $\mathbb{R}^n$ is endowed with the scalar product $\langle x, y \rangle_W = x^\top W y$ for $x, y \in \mathbb{R}^n$. For $v \in \mathbb{R}^n$, we have

$$\Pi v = H \hat{\beta}_n(v) \quad \text{where} \quad \hat{\beta}_n(v) \in \arg\min_{b \in \mathbb{R}^m} \left\| W^{1/2}(v - Hb) \right\|_2^2.$$

If H has rank m, then the solution to the above minimization problem is unique and $\Pi = H(H^\top W H)^{-1} H^\top W$; otherwise, the matrix Π is still uniquely defined, even though there are then multiple solutions $\hat{\beta}_n(v)$. Given $a \in \mathbb{R}$, the minimum in (3.14) over $b \in \mathbb{R}^m$ is attained as soon as $Hb = \Pi(f^{(n)} - a\mathbf{1}_n)$. Therefore

$$\hat{\alpha}_n \in \arg\min_{a \in \mathbb{R}} \left\| W^{1/2}(I_n - \Pi)(f^{(n)} - a\mathbf{1}_n) \right\|_2^2, \tag{3.15}$$

where I_n is the $n \times n$ identity matrix. Recall the vector e_n in (3.8). In our present notation, we have

$$e_n = (I_n - \Pi)\mathbf{1}_n.$$

Proposition 3.21 (Matrix representation)**.** *The minimizer $\hat{\alpha}_n$ in (3.15) is unique if and only if $e_n \neq 0$, in which case the normalized AISCV estimate is*

$$I_n^{(\mathrm{aiscv})}(f) = \hat{\alpha}_n = \frac{\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)}}{\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi)\mathbf{1}_n} = \frac{\mathbf{1}_n^\top (I_n - \Pi)^\top W f^{(n)}}{\mathbf{1}_n^\top (I_n - \Pi)^\top W \mathbf{1}_n}. \tag{3.16}$$

Proof The objective function on the right-hand side of (3.16) is

$$a^2 \mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi)\mathbf{1}_n - 2a\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)} + \text{constant},$$

where the unspecified constant does not depend on a. The coefficient of a^2 is equal to $e_n^\top W e_n$, which is positive if and only if $e_n \neq 0$. The latter is thus a necessary and sufficient for the minimizer $\hat{\alpha}_n$ to exist and be unique. In that case, the objective function is a convex quadratic function in a, whose minimizer is easily seen to be equal to the stated expression. ■

3.C Proofs of the main results

3.C.1 Proof of Proposition 3.2

Proof We start from Proposition 3.21. Recall that $e_n = (I_n - \Pi)\mathbf{1}_n$. Since $\Pi^\top W = W\Pi$ and $\Pi^2 = \Pi$, we find $(I_n - \Pi)^\top W(I_n - \Pi) = (I_n - \Pi)^\top W$. We obtain

$$\mathbf{1}_n^\top (I_n - \Pi)^\top W(I_n - \Pi) f^{(n)} = \mathbf{1}_n^\top (I_n - \Pi)^\top W g^{(n)} = e_n^\top W g^{(n)} = \sum_{i=1}^n w_i e_{n,i} f(X_i),$$

and similarly $\mathbf{1}_n^\top (I_n - \Pi)^\top W(I_n - \Pi) f^{(n)} = \sum_{i=1}^n w_i e_{n,i}$. ■

3.C.2 Proof of Proposition 3.3

Proof If $f = \alpha + \beta^\top h$ for some $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, then the minimum in (3.7) is clearly attained for $\hat{\alpha}_n = \alpha$ and $\hat{\beta}_n = \beta$. ■

3.C.3 Proof of Proposition 3.4

Proof In (3.7), if b ranges over $\mathbb{R}^m$, then $A^\top b$ ranges over $\mathbb{R}^m$ too, since A is invertible. It follows that the solutions $\hat{\alpha}_n$ in (3.7) do not change if we replace h by Ah, since $b^\top Ah = (A^\top b)^\top h$. ■

3.C.4 Proof of Theorem 3.8

Proof

Step 1: Working out the probability of several bounds. In Step 1, we gather several elementary bounds that will be useful to establish more advanced bounds in Step 2.

Bound 1. To control $\left|\sum_{i=1}^n w_i \varepsilon(X_i)\right|$, we apply Lemma 3.18 with Z_i equal to $w_i\varepsilon(X_i)$. We have $\mathbb{E}[w_i\varepsilon(X_i)|\mathcal{F}_{i-1}] = 0$ and by Assumption 3.7,

$$\mathbb{P}[|w_i\varepsilon(X_i)| > t|\mathcal{F}_{i-1}] \leq 2\exp(-t^2/(2\tau^2))$$

holds, and the sub-Gaussian variance factor is simply τ^2. Therefore, with probability at least $1-\delta/5$, we have

$$\left|\sum_{i=1}^{n} w_i \varepsilon(X_i)\right| \leq K\tau\sqrt{n\log(10/\delta)}.$$

Bound 2. For the term $\left\|\sum_{i=1}^{n} w_i \hbar(X_i)\right\|_2$, we apply Lemma 3.18 with Z_i equal to $w_i\hbar(X_i)$. By Assumptions 3.6 and 3.5, we have $\|w_i\hbar(X_i)\|_2 \leq c\|\hbar(X_i)\|_2 \leq c\sqrt{B}$, which implies that $w_i\hbar(X_i)$ is sub-Gaussian (conditionally on $\mathcal{F}_{i-1}$) with variance factor c^2B (Boucheron et al., 2013b, Lemma 2.2). Hence (3.10) is satisfied with $\sigma^2 = c^2B$. Thus, with probability at least $1-\delta/5$, the inequality

$$\left\|\sum_{i=1}^{n} w_i \hbar(X_i)\right\|_2 \leq Kc\sqrt{nB\log(10m/\delta)}$$

holds.

Bound 3. Now we treat the term $\left\|\sum_{i=1}^{n} w_i \hbar(X_i)\varepsilon(X_i)\right\|_2$ applying again Lemma 3.18 but this time with Z_i equal to $w_i\hbar(X_i)\varepsilon(X_i)$. We have that $\|w_i\hbar(X_i)\varepsilon_i\|_2 \leq \sqrt{B}|w_i\varepsilon_i|$. By Assumption 3.7, we have, for all $t > 0$,

$$\begin{aligned}\mathbb{P}[\left\|w_i\hbar(X_i)\varepsilon(X_i)\right\|_2 > t|\mathcal{F}_i] &\leq \mathbb{P}[\sqrt{B}\left|w_i\varepsilon(X_i)\right| > t|\mathcal{F}_i] \\ &\leq 2\exp\left(-\frac{t^2}{2B\tau^2}\right),\end{aligned}$$

and (3.10) holds with $\sigma^2 = B\tau^2$. Lemma 3.18 then implies that, with probability at least $1-\delta/5$,

$$\left\|\sum_{i=1}^{n} w_i \hbar(X_i)\varepsilon(X_i)\right\|_2 \leq K\sqrt{nB\tau^2\log(10m/\delta)}.$$

Bound 4. By Lemma 3.19 and Remark 3.20, we have, with probability at least $1-\delta/5$,

$$\lambda_{\min}\left(\sum_{i=1}^{n} w_i \hbar(X_i)\hbar^\top(X_i)\right) > (1-\zeta)n\lambda_{\min}(G) = (1-\zeta)n,$$

where, by Assumption 3.6, $G = \int f\hbar\hbar^\top \,\mathrm{d}\lambda = I$, ζ satisfies the equation

$$m\exp(\frac{-\zeta^2 n}{2L}) = \delta/5.$$

with $L = cB$ according to Assumptions 3.6 and 3.5. Solving the last equation, we obtain

$$\zeta = \sqrt{\frac{2L\log(5m/\delta)}{n}}.$$

We choose $\zeta \leq 1/2$ which gives the condition $n \geq 8cB\log(5m/\delta)$ and, with probability at least $1-\delta/5$,

$$\lambda_{\min}\left(\sum_{i=1}^{n} w_i \hbar(X_i)\hbar^\top(X_i)\right) > \big(1-\zeta\big)\, n \geq n/2. \tag{3.17}$$

Bound 5. Now we consider the term $\sum_{i=1}^n w_i$. Since $-1 \leq w_i - 1 \leq c$, $|w_i - 1|$ is bounded by c, and $w_i - 1$ is sub-Gaussian with variance factor c^2. This makes the inequality required in Lemma 3.18 valid and henceforth

$$\left| \sum_{i=1}^n (w_i - 1) \right| \leq Kc\sqrt{n \log(10/\delta)}$$

or

$$-Kc\sqrt{n \log(10/\delta)} + n \leq \sum_{i=1}^n w_i \leq Kc\sqrt{n \log(10/\delta)} + n.$$

We want to have $Kc\sqrt{n \log(10/\delta)} \leq n/2$. It holds if $\sqrt{n} \geq 2Kc\sqrt{\log(10/\delta)}$. Then we get that $n/2 = n - n/2 \leq n - Kc\sqrt{n \log(10/\delta)} \leq \sum_{i=1}^n w_i$. Therefore, with probability at least $1 - \delta/5$, it holds that

$$\sum_{i=1}^n w_i \geq n/2.$$

Step 2: Extending the previous elementary bounds on appropriate quantities. The work in this step consists in showing that under the five previous bounds, and therefore with probability at least $1 - \delta$, we have that

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \hbar_W(X_i) \hbar_W(X_i)^\top \right) \geq n/4, \tag{3.18}$$

$$\left\| \sum_{i=1}^n w_i \hbar_W(X_i) \varepsilon_W(X_i) \right\|_2 \leq 2K\tau\sqrt{nB \log(10m/\delta)}. \tag{3.19}$$

We start by proving (3.18). Recognizing a covariance, we get

$$P_{n,w}\{\hbar_W \hbar_W^\top\} = P_{n,w}(\hbar\hbar^\top) - P_{n,w}(\hbar) P_{n,w}(\hbar)^\top,$$

and then, using Cauchy-Schwarz inequality, we have

$$\lambda_{\min}(P_{n,w}\{\hbar_W \hbar_W^\top\}) \geq \lambda_{\min}(P_{n,w}(\hbar\hbar^\top)) - \|P_{n,w}(\hbar)\|_2^2$$

or, equivalently,

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \hbar_W(X_i) \hbar_W(X_i)^\top \right) \geq \lambda_{\min} \left(\sum_{i=1}^n w_i \hbar(X_i) \hbar(X_i)^\top \right) - \left\| \sum_{i=1}^n w_i \hbar(X_i) \right\|_2^2 \Big/ \sum_{i=1}^n w_i,$$

From Bound 2 and Bound 5,

$$\left\| \sum_{i=1}^n w_i \hbar(X_i) \right\|_2^2 \Big/ \sum_{i=1}^n w_i \leq \frac{K^2 c^2 B n \log(10m/\delta)}{n/2} = 2K^2 c^2 B \log(10m/\delta)$$

Using Bound 4 and the previous inequality, it follows that

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \hbar_W(X_i) \hbar_W(X_i)^\top \right) \geq n/2 - 2K^2 c^2 B \log(10m/\delta).$$

If $n \geq 8K^2c^2B\log(10m/\delta)$,

$$\lambda_{\min}\left(\sum_{i=1}^{n} w_i \hbar_W(X_i)\hbar_W(X_i)^\top\right) \geq n/4.$$

We have just obtained (3.18).

Let us now establish (3.19). Recognizing a covariance, we find

$$P_{n,w}\{\hbar_W \varepsilon_W\} = P_{n,w}(\hbar\varepsilon) - P_{n,w}(\hbar)P_{n,w}(\varepsilon),$$

and it follows that

$$\left\|P_{n,w}\{\hbar_W \varepsilon_W\}\right\|_2 \leq \|P_{n,w}(\hbar\varepsilon)\|_2 + \|P_{n,w}(\hbar)\|_2 |P_{n,w}(\varepsilon)|,$$

or, equivalently,

$$\left\|\sum_{i=1}^{n} w_i \hbar_W(X_i)\varepsilon_W(X_i)\right\|_2 \leq \left\|\sum_{i=1}^{n} w_i \hbar(X_i)\varepsilon(X_i)\right\|_2 + \|P_{n,w}(\hbar)\|_2 \left|\sum_{i=1}^{n} w_i \varepsilon(X_i)\right|.$$

Now using Bound 2 and 5, we find

$$\|P_{n,w}(\hbar)\|_2 \leq 2Kc\sqrt{\frac{B\log(10m/\delta)}{n}}, \tag{3.20}$$

which combined with Bound 1 leads to

$$\begin{aligned}\|P_{n,w}(\hbar)\|_2 \left|\sum_{i=1}^{n} w_i \varepsilon(X_i)\right| &\leq 2K^2c\tau\sqrt{B\log(10m/\delta)\log(10/\delta)} \\ &\leq 2K^2c\tau\sqrt{B}\log(10m/\delta).\end{aligned}$$

The previous inequality and Bound 3 gives

$$\begin{aligned}\left\|\sum_{i=1}^{n} w_i \hbar_W(X_i)\varepsilon_W(X_i)\right\|_2 &\leq K\tau\sqrt{nB\log(10m/\delta)} + 2K^2c\tau\sqrt{B}\log(10m/\delta) \\ &= K\tau\sqrt{nB\log(10m/\delta)}\left(1 + 2Kc\sqrt{\frac{\log(10m/\delta)}{n}}\right) \\ &\leq 2K\tau\sqrt{nB\log(10m/\delta)}\end{aligned}$$

if $n \geq 4K^2c^2\log(10m/\delta)$.

The condition $n \geq 8K^2c^2B\log(10m/\delta)$ (used in establishing (3.18)) implies $n \geq 4K^2c^2\log(10m/\delta)$ (used in proving (3.19)), $n \geq 8cB\log(5m/\delta)$ (used in Bound 4) and $n \geq 4K^2c^2\log(10/\delta)$ (used in Bound 5) since $m \geq 1$, $B \geq m$ and $c \geq 1$. Therefore, the constant C_1 from the statement of the theorem equals $8K^2$.

Step 3. End of the proof. The quantity to be bounded can be written as a sum of two terms as follows

$$\alpha_n^{(\mathrm{aiscv})}(f, \hat{\beta}_n) - \pi(f)\,\mathrm{d}x = P_{n,w}\{\varepsilon\} + P_{n,w}\{h\}^\top(\beta^* - \hat{\beta}_n).$$

Using Bounds 1 and 5, the first term in the right-hand side satisfies

$$|P_{n,w}\{\varepsilon\}| \leq 2K\tau\sqrt{\frac{\log(10/\delta)}{n}}.$$

This corresponds to the first term in the bound of the theorem with the constant C_2 equals $2K$. Hence, it remains to show that

$$|P_{n,w}\{h\}^\top(\beta^* - \hat{\beta}_n)| \leq C_3 cB\tau\log(10m/\delta)/n.$$

Introducing $G^{-1/2}G^{1/2}$, we obtain

$$P_{n,w}\{h\}^\top(\beta^* - \hat{\beta}_n) = P_{n,w}\{\hbar\}^\top G^{1/2}(\beta^* - \hat{\beta}_n).$$

Then, using the identity

$$(\hat{\beta}_n - \beta^*) = (H_W^\top W H_W)^{-1} H_W^\top W \varepsilon_W^{(n)}$$

and Cauchy-Schwarz inequality yields

$$\begin{aligned}
\left|P_{n,w}\{h\}^\top(\beta^* - \hat{\beta}_n)\right| &\leq \left\|P_{n,w}\{\hbar\}\right\|_2 \|G^{1/2}(\beta^* - \hat{\beta}_n)\|_2 \\
&\leq \left\|P_{n,w}\{\hbar\}\right\|_2 \left\|G^{1/2}(H_W^\top W H_W)^{-1} H_W^\top W \varepsilon_W^{(n)}\right\|_2 \\
&\leq \left\|P_{n,w}\{\hbar\}\right\|_2 \left\|G^{1/2}(H_W^\top W H_W)^{-1} G^{1/2}\right\|_2 \left\|G^{-1/2} H_W^\top W \varepsilon_W^{(n)}\right\|_2 \\
&= \left\|P_{n,w}\{\hbar\}\right\|_2 \left\|G^{1/2}(H_W^\top W H_W)^{-1} G^{1/2}\right\|_2 \left\|G^{-1/2} H_W^\top W \varepsilon_W^{(n)}\right\|_2.
\end{aligned}$$

By (3.18), we have

$$\begin{aligned}
\left\|G^{1/2}(H_W^\top W H_W)^{-1} G^{1/2}\right\|_2 &= \left\|\left(\sum_{i=1}^n w_i \hbar_W(X_i)\hbar_W(X_i)^\top\right)^{-1}\right\|_2 \\
&= \left[\lambda_{\min}\left(\sum_{i=1}^n w_i \hbar_W(X_i)\hbar_W(X_i)^\top\right)\right]^{-1} \leq 4/n.
\end{aligned}$$

From (3.19) and (3.20), it follows that

$$\begin{aligned}
\left|P_{n,w}\{h\}^\top(\beta^* - \hat{\beta}_n)\right| &\leq 2K\sqrt{\frac{B\log(10m/\delta)}{n}}\frac{8Kc\tau\sqrt{nB\log(10m/\delta)}}{n} \\
&= 16K^2 cB\tau\frac{\log(10m/\delta)}{n}.
\end{aligned}$$

Therefore, the constant C_3 from the statement of the theorem equals $16K^2$. ■

3.D Additional numerical results

Parameters. In all simulations, the sampling policy is taken within the family of multivariate Student t distributions of degree ν denoted by $\{q_{\mu,\Sigma_0} : \mu \in \mathbb{R}^d\}$ with $\Sigma_0 = \sigma_0 I_d(\nu-2)/\nu$ and $\nu > 2, \sigma_0 > 0$. Similarly to Portier and Delyon (2018), the mean μ_t is updated at each stage $t = 1, \dots, T$ by the generalized method of moments (GMM), leading to

$$\mu_t = \frac{\sum_{s=1}^{t}\sum_{i=1}^{n_s} w_{s,i}X_{s,i}}{\sum_{s=1}^{t}\sum_{i=1}^{n_s} w_{s,i}}.$$

The allocation policy is fixed to $n_t = 1000$ and the number of stages is $T \in \{5; 10; 20; 30; 50\}$. The different Monte Carlo estimates are compared by their mean squared error (MSE) obtained over 100 independent replications. In other words, for each method that returns $\hat{I}(g)$, the mean square error is computed as the average of $\|\hat{I}(f)-\pi(f)\|_2^2$ computed over 100 replicates of $\hat{I}(f)$. When the integrand is real-valued, this quantity is scaled as $([\hat{I}(f) - \pi(f)]/\pi(f))^2$.

The experiments were performed on a laptop Intel Core i7-10510U CPU 1.80GHz × 8.

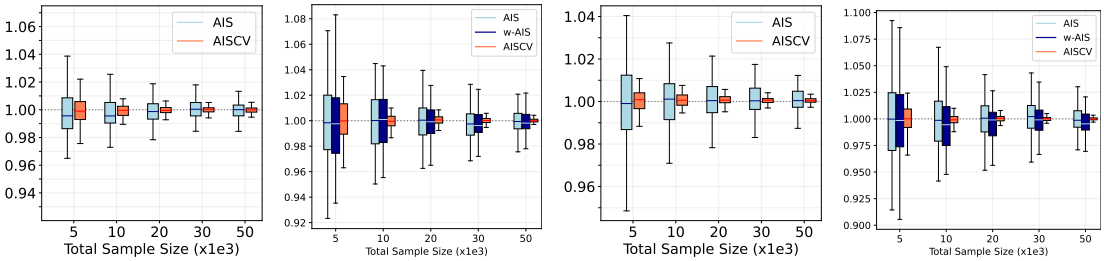
3.D.1 Synthetic examples: integration on $[0,1]^d$

We seek to integrate functions f with respect to the uniform density $\pi(x) = 1$ for $x \in [0,1]^d$ in dimensions $d \in \{4; 8\}$. We rely on Legendre polynomials for the control variates. Consider the integrands $f_1(x) = 1 + \sin(\pi(2d^{-1}\sum_{i=1}^{d} x_i - 1))$, $f_2(x) = \prod_{i=1}^{d}(2/\pi)^{1/2}x_i^{-1}\mathrm{e}^{-\log(x_i)^2/2}$ and $f_3(x) = \prod_{i=1}^{d}\log(2)2^{1-x_i}$, all of which integrate to 1 on $[0,1]^d$. None of the integrands is a linear combination of the control variates. The policy parameters are $\mu_0 = (0.5, \dots, 0.5) \in \mathbb{R}^d$, $\nu = 8$, and $\sigma_0 = 0.1$. The control variates are built out of tensor products of Legendre polynomials where the degree ℓ_j equals 0 for all but two coordinates, leading to a total number of $m = kd + k^2d(d-1)/2$ control variates. The maximum degree in each variable is $k = 6$, yielding $m = 240$ and $m = 1056$ control variates in dimensions $d = 4$ and $d = 8$ respectively. Figure 3.1 presents the boxplots of the AIS and AISCV estimates.

Figure 3.4 presents the boxplots of the different estimates and Table 3.1 gathers the numerical values of the mean squared errors. As a natural competitor to our AISCV estimator, we also implemented the weighted version of standard AIS called *w-AIS* introduced in Portier and Delyon (2018). Interestingly, such a method presents similar or even worse performance than the standard AIS estimate for dimension $d = 4$ but better results for dimension $d = 8$. This good behavior is illustrated in Figure 3.4b and Figure 3.4d. Accordingly, the values of the MSE for w-AIS are smaller than the one of AIS in dimension $d = 8$ but still greater than the ones of AISCV.

3.D.2 Synthetic examples: gaussian mixtures

General target π and Stein method. In this setting we only assume acces to the evaluations of the target density π. We consider the classical example introduced in Cappé et al. (2008) where π is a mixture of two gaussian distributions. The control variates are built using Stein's method with polynomial maps of degree $Q \in \{2; 3\}$ leading to a number of control variates $m \in \{14; 34\}$ in dimension $d = 4$ and $m \in \{44; 164\}$ in dimension $d = 8$ respectively.



(a) $f_1(d=4)$ (b) $f_1(d=8)$ (c) $f_2(d=4)$ (d) $f_3(d=8)$

Figure 3.4 – Integration on $[0,1]^d$: boxplots of estimates $\alpha_n^{(\text{ais})}(f)$ and $\alpha_n^{(\text{aiscv})}(f)$ with integrands f_1, f_2, f_3 in dimensions $d \in \{4; 8\}$ obtained over 100 replications. The true integral equals 1.

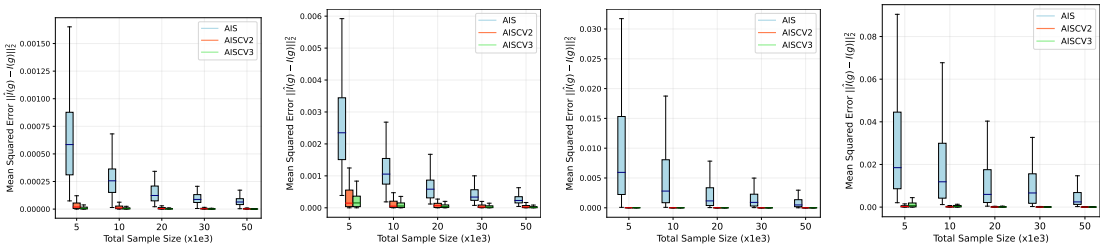
Integrand	Method \ Sample Size n	$5,000$	$10,000$	$20,000$	$30,000$	$50,000$
f_1 $(d=4)$	AIS	2.9e-4	1.5e-4	7.8e-5	5.8e-5	3.7e-5
	wAIS	3.0e-4	1.6e-4	8.3e-5	6.5e-5	4.1e-5
	AISCV	**9.7e-5**	**1.9e-5**	**1.0e-5**	**7.5e-6**	**4.3e-6**
f_1 $(d=8)$	AIS	8.7e-4	4.6e-4	2.3e-4	1.9e-4	1.0e-4
	wAIS	9.2e-4	4.6e-4	2.2e-4	1.6e-4	9.0e-5
	AISCV	**3.2e-4**	**3.2e-5**	**1.1e-5**	**6.0e-6**	**2.5e-6**
f_2 $(d=4)$	AIS	3.4e-4	1.3e-4	7.6e-5	5.9e-5	3.1e-5
	wAIS	3.7e-4	1.6e-4	1.2e-4	1.1e-4	7.9e-5
	AISCV	**3.1e-5**	**1.0e-5**	**4.9e-6**	**2.6e-6**	**1.5e-6**
f_3 $(d=8)$	AIS	1.6e-3	7.8e-4	4.0e-4	3.3e-4	1.9e-4
	wAIS	1.5e-3	7.3e-4	3.6e-4	2.7e-4	1.5e-4
	AISCV	**1.7e-4**	**2.1e-5**	**7.8e-6**	**4.3e-6**	**1.8e-6**

Table 3.1 – Mean Square Errors for f_1, f_2, f_3 with AIS, wAIS (Portier and Delyon, 2018) and AISCV in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

Isotropic case. Let $\pi_\Sigma(x) = 0.5\Phi_\Sigma(x-\mu)+0.5\Phi_\Sigma(x+\mu)$ where $\mu = (1,\dots,1)^\top/2\sqrt{d}, \Sigma = I_d/d$ and Φ_Σ is the multivariate normal density function with zero mean and covariance matrix Σ. Note that the Euclidean distance between the two mixture centers is independent of the dimension as it equals 1. The initial density q_0 is the multivariate student distribution with mean $(1,-1,0,\dots,0)/\sqrt{d}$ and variance $(5/d)I_d$. The initial mean value differs from the null vector to prevent the naive algorithm using the initial density from having good results (due to the symmetry).

Anisotropic case. In this case, the mixture is unbalanced and each gaussian is anisotropic. The target density is $\pi_V(x) = 0.75\Phi_V(x-\mu) + 0.25\Phi_V(x+\mu)$ where $\mu = (1,\dots,1)^\top/2\sqrt{d}$ and $V = Diag(10,1,\dots,1)/d$. The initial density q_0 is the same as for the isotropic case.

Figure 3.5 presents the boxplots of the mean square error $\|\hat{I}(f) - I(f)\|_2^2$ and Table 3.2 gathers the associated numerical values.



(a) $\pi_\Sigma(d=4)$ (b) $\pi_\Sigma(d=8)$ (c) $\pi_V(d=4)$ (d) $\pi_V(d=8)$

Figure 3.5 – Boxplots for $\|\hat{I}(f) - \pi(f)\|_2^2$ for $g(x) = x$ with target isotropic π_Σ and anisotropic π_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

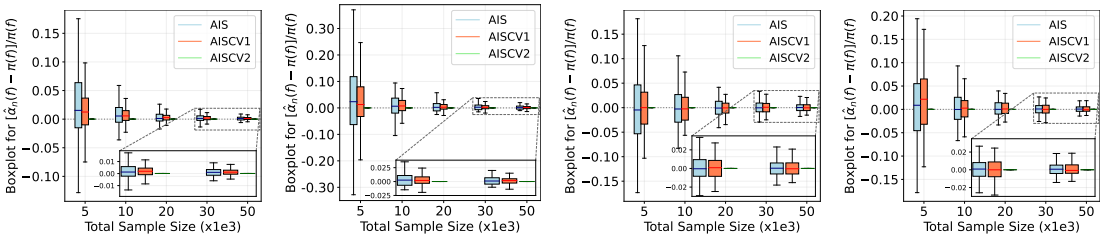
Target	Sample Size n / Method	$5,000$	$10,000$	$20,000$	$30,000$	$50,000$
π_Σ $(d=4)$	AIS	6.9e-4	2.9e-4	1.5e-4	1.1e-4	7.2e-5
	wAIS	6.8e-4	2.9e-4	1.5e-4	1.1e-4	7.3e-5
	AISCV-2	4.1e-5	2.2e-5	9.1e-6	5.6e-6	3.7e-6
	AISCV-3	**1.5e-5**	**8.4e-6**	**3.7e-6**	**2.3e-6**	**1.3e-6**
π_Σ $(d=8)$	AIS	2.7e-3	1.2e-3	6.6e-4	4.1e-4	2.7e-4
	wAIS	2.7e-3	1.2e-3	6.9e-4	4.3e-4	2.8e-4
	AISCV-2	3.7e-4	1.7e-4	1.0e-4	6.8e-5	4.7e-5
	AISCV-3	**2.8e-4**	**1.2e-4**	**6.3e-5**	**4.2e-5**	**2.6e-5**
π_V $(d=4)$	AIS	1.1e-2	5.5e-3	2.2e-3	1.6e-3	9.5e-4
	wAIS	1.1e-2	5.3e-3	2.0e-3	1.3e-3	8.0e-4
	AISCV-2	1.3e-5	7.2e-6	2.9e-6	1.9e-6	1.2e-6
	AISCV-3	**1.1e-5**	**6.6e-6**	**2.2e-6**	**1.5e-6**	**9.6e-7**
π_V $(d=8)$	AIS	4.5e-2	3.2e-2	2.2e-2	1.5e-2	6.8e-3
	wAIS	2.6e-2	1.3e-2	7.8e-3	5.9e-3	3.8e-3
	AISCV-2	**4.6e-4**	**2.8e-4**	**1.3e-4**	**9.7e-5**	6.0e-5
	AISCV-3	1.4e-3	4.8e-4	1.5e-4	1.1e-4	**5.7e-5**

Table 3.2 – Mean Square Errors $\|\hat{I}(f) - \pi(f)\|_2^2$ for $f(x) = x$ with target isotropic π_Σ and anisotropic π_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

3.D.3 Real-world data: Bayesian linear regression

We place ourselves in the framework of Bayesian linear regression with observations $X \in \mathbb{R}^{N\times d}$ and labels $y \in \mathbb{R}^N$. The posterior distribution $p(\theta|\mathcal{D})$ depends on a gaussian prior $p \sim \mathcal{N}(\mu_a, \Sigma_a)$ and a likelihood function $\ell(\mathcal{D}|\theta) \propto (\sigma^2)^{-N/2} \exp(-(y - X\theta)^\top (y - X\theta)/(2\sigma^2))$ where the noise level is fixed and taken sufficiently large $\sigma = 50$ to account general priors. Observe that the posterior distribution is also gaussian $\mathcal{N}(\mu_b, \Sigma_b)$ with $\mu_b = \Sigma_b(\sigma^{-2}X^\top y + \Sigma_a^{-1}\mu_a)$ and $\Sigma_b = (\sigma^{-2}X^\top X + \Sigma_a^{-1})^{-1}$. The integrand is $f(\theta) = \|\theta\|_2^2$ and the control variates are built using Stein method described in Section 3.5.1 with degree $Q \in \{1; 2\}$, leading to the AISCV1 and AISCV2 estimators respectively. Observe that when $Q = 2$, the integrand belongs to the linear span of the control variates so the integration should be exact in light of Proposition 3.3.

Datasets and parameters. Some classical datasets from UCI Machine Learning repository Dua and Graff (2019) are considered : *housing* ($N = 506; d = 13; m \in \{12; 104\}$); *abalone* ($N = 4,177; d = 8; m \in \{7; 44\}$); *red wine* ($N = 1,599; d = 11; m \in$



(a) Housing (b) Abalone (c) Red Wine (d) White Wine

Figure 3.6 – Boxplots of $(\hat{I}(f)-I(f))/I(f)$, $f(\theta) = \|\theta\|_2^2$, obtained over 100 replications.

$\{10; 77\}$) and *white wine* ($N = 4,898; d = 11; m \in \{10; 77\}$). The initial density is the multivariate student distribution with $\nu = 10$ degrees of freedom, zero mean and covariance matrix Σ_b.

Results. Figure 3.6 presents the boxplots of the error $(\hat{I}(f)-\pi(f))/\pi(f)$ and Table 3.3 gathers the associated numerical values. Observe the benefits of using control variates even with polynomials of degree $Q = 1$. Observe that when $Q = 2$, the error of the AISCV2 estimator is almost equal to zero which is in line with Proposition 3.3. Accordingly when looking at the MSE, the AISCV1 error is smaller than the AIS one by a factor ranging between 2 and 10 and the MSE of AISCV2 is of order 10^{-9}.

Sample Size n / Dataset	Method	5,000	10,000	20,000	30,000	50,000
Housing	AIS	2.2e-2	4.4e-3	3.1e-4	2.7e-4	2.5e-4
	AISCV1	2.9e-3	7.0e-4	1.7e-4	1.6e-4	5.2e-5
	AISCV2	5.6e-9	5.6e-9	5.6e-9	5.6e-9	5.6e-9
Abalone	AIS	6.2e-2	2.6e-2	1.1e-2	6.5e-3	3.1e-3
	AISCV1	6.3e-3	1.2e-3	4.7e-4	3.1e-4	1.8e-4
	AISCV2	5.1e-9	6.1e-9	6.1e-9	6.1e-9	6.1e-9
Red Wine	AIS	3.0e-2	1.3e-2	7.0e-3	4.7e-3	2.8e-3
	AISCV1	3.7e-3	1.5e-3	8.7e-4	6.4e-4	4.2e-4
	AISCV2	5.1e-10	5.1e-10	5.1e-10	5.1e-10	5.1e-10
White Wine	AIS	1.1e-2	2.6e-3	8.1e-4	4.2e-4	1.8e-4
	AISCV1	7.1e-3	1.5e-3	4.0e-4	2.1e-4	9.2e-5
	AISCV2	2.4e-9	2.4e-9	2.4e-9	2.4e-9	2.4e-9

Table 3.3 – Mean Square Errors for different datasets with $f(\theta) = \|\theta\|_2^2$ obtained over 100 replications.

Sample Size n	5,000	10,000	20,000	30,000
Housing	5.5e-5	2.7e-5	1.9e-5	1.2e-5
Abalone	1.8e-4	8.6e-5	6.7e-5	5.6e-5
Red Wine	2.7e-4	1.8e-4	9.5e-5	5.2e-5
White Wine	3.8e-4	1.6e-4	8.5e-5	7.3e-5

Table 3.4 – MSE with $f(\theta) = \|\theta\|_2^2$ obtained over 30 chains of NUTS sampler.

Monte Carlo Markov Chain. We run a state-of-the-art MCMC method called NUTS Hoffman et al. (2014), which is a self-tuning variant of Hamiltonian Monte Carlo. It may be hard to compare precisely this method against the AIS based methods since

they are different in nature. Indeed, the goal of MCMC methods is to sample from a target distribution whereas AISCV methods are meant for variance reduction. In both cases there are hyperparameters to tune. For AIS-based methods, there is the choice of the policy $(q_i)_{i\geq 0}$, the choice of the control variates and the number of particles n_t to draw at each step. For the NUTS sampler, there is among others, the number of samples used for the tuning phase and the initialization of the Markov kernel. A reasonable comparison is obtained based on the overall number of sampled particles. Table 3.4 above presents the mean squared errors obtained over 30 chains of NUTS sampler with default configuration of the parameters from the Python library *pymc3* Patil et al. (2010).

Chapter 4

Speeding up Monte Carlo: Nearest Neighbors as Control Variates

Contents

4.1 Introduction

Consider the numerical integration problem to approximate the value of an integral $\pi(f) = \int f(x)\pi(x)\mathrm{d}x$ where π is a probability density on $\mathbb{R}^d$ and the integrand $f : \mathbb{R}^d \to \mathbb{R}$ is a real-valued function defined on the support of π. Suppose that random draws from the density π are available and calls to the function f are possible. The standard Monte Carlo estimate consists in averaging $f(X_i)$ over $i = 1, \dots, n$, where the particles X_i are identically and independently drawn from π. While easy to implement and fast to compute, a recognized drawback of the Monte Carlo estimate is its slow convergence rate in $\mathrm{O}(n^{-1/2})$. In some applications, one may only have access to a very limited number of evaluations $f(X_i)$ due to expensive calls of the integrand, *e.g.* in complex Bayesian inference models (Higdon et al., 2015). The $\sqrt{n}$-convergence of the standard Monte Carlo estimate becomes too slow and leads to high-variance estimation.

As detailed in Novak (2016), the complexity of integration algorithms may be analyzed through the convergence rate of the error. Any randomized procedure based on n particles yields an estimate $\hat{\pi}_n(f)$ of the integral $\pi(f)$. In this context, the error of the procedure is defined as $\mathbb{E}[|\hat{\pi}_n(f) - \pi(f)|^2]^{1/2}$. For the specific problem of integration with respect to the uniform measure over the unit cube $[0,1]^d$ with $d \geq 1$, the complexity rate of randomized methods for Lipschitz integrands is known to be $\mathrm{O}(n^{-1/2}n^{-1/d})$ (see Novak (2016)). Furthermore, when the integrand has bounded s first derivatives, the convergence rate becomes $\mathrm{O}(n^{-1/2}n^{-s/d})$. These complexity rates are informative as they advocate the use of random methods over deterministic integration rules. The convergence rates indicate that random methods have some $\mathrm{O}(n^{-1/2})$ gain compared to deterministic methods with complexity rates in $\mathrm{O}(n^{-s/d})$. In addition, they show that the naive Monte Carlo estimate is suboptimal from the convergence rate perspective. This supports the idea that there is room for improvement by relying in particular on the regularity of the integrand. Several approaches are already known for improving upon the Monte Carlo benchmark in terms of convergence rate. They can be classified according to their convergence rates while keeping in mind the lower bound $\mathrm{O}(n^{-1/2}n^{-s/d})$.

The control variate method (Glasserman, 2004) is a powerful technique that allows to reduce the variance of the Monte Carlo estimate using some approximation of the integrand function. Relying on some nonparametric statistical approximation of f, the Monte Carlo rate of convergence can be improved using control variates as demonstrated in Oates et al. (2017); Portier and Segers (2019); Leluc et al. (2021); South et al. (2022). In Portier and Segers (2019), when using m control variates, the convergence rate is $\mathrm{O}(n^{-1/2}m^{-s/d})$ where s is the regularity of f and the measure π is arbitrary. The associated computation of optimal control variates relies on ordinary least squares regression. To avoid ill-conditioning and for numerical stability, it requires that m should be of a smaller order than n and thus, it prevents from achieving the optimal rate. Relying on some control function constructed in a reproducing kernel Hilbert space, Oates et al. (2017) derived an acceleration compared to the naive $\sqrt{n}$-convergence rate and obtained $\mathrm{O}(n^{-7/12})$ for a specific class of functions.

Determinantal sampling has been used for Monte Carlo integration in Bardenet and Hardy (2020) in which a stochastic quadrature rule is proposed. It allows to reduce the error to $\mathrm{O}(n^{-1/2}n^{-1/2d})$ when the function f is differentiable with continuous derivatives. This interesting acceleration still remains slower than the optimal lower bound.

Another reliable technique to improve the rate of convergence of standard Monte Carlo is stratification. This technique consists in partitioning the space and sampling over each element of the partition. It has allowed to improve the convergence rate of Monte Carlo estimates (Haber, 1966, 1967) and to derive a general framework called stochastic quadrature rules (Haber, 1969). Recently, Haber's work has been extended to take advantage of higher smoothness in the integrand (Chopin and Gerber, 2022). To the best of found knowledge, the works of Haber (1966) and Chopin and Gerber (2022) are the only ones achieving the best rate of convergence for Lipschitz function and for general regularity space.

Still concerned about the integration problem with respect to the uniform measure on $[0,1]^d$, other methods such as Quasi-Monte Carlo and Randomized Quasi-Monte Carlo have been studied (Caflisch, 1998; Dick and Pillichshammer, 2010) . These methods are fitted for specific functions having finite Hardy–Krause variation and can attain an error bound of order $\mathrm{O}(\log(n)^d/n)$. This type of methods is therefore associated to other complexity rates (Novak, 2016).

Observe that the methods in Haber (1966) and in Chopin and Gerber (2022), even though they achieve the optimal convergence rate, are only valid for integration over the unit cube. In addition they involve a geometric number (ℓ^d) of evaluations of the integrand f which is problematic in practice for applications with small computational budget as in complex bayesian models. Interestingly, as mentioned in Chopin and Gerber (2022), their stratification method is related to a specific control variates construction relying on a piecewise constant control function which has a very low bias compared to traditional regression estimate. This precise idea of using an estimate with small bias is the starting point of this chapter. It is relevant to the considered framework because the function f is accessible without noise. Note that this kind of estimates - with small bias - has also been successfully used in the related topic of adaptive rejection sampling (Achddou et al., 2019) allowing to reach optimal rate. All the different properties of the mentioned Monte Carlo estimates are summarized in Table below.

Monte Carlo method	Super-$\sqrt{n}$ convergence	Optimal rate	General π
Vanilla Monte Carlo	×	✓	✓
Quasi Monte Carlo (QMC and RQMC) (Caflisch, 1998; Dick and Pillichshammer, 2010)	✓	✓	×
OLS-based Control Functionals (Oates et al., 2017; Portier and Segers, 2019)	✓	×	✓
Cubic Stratification (Haber, 1966; Chopin and Gerber, 2022)	✓	✓	×
Control neighbors	✓	✓	✓

In this chapter, a new Monte Carlo method called *control neighbors* is introduced. This method constructs an estimate $\hat{\pi}_n(f)$ to approximate the integral $\pi(f)$ for general probability measure π and the core idea follows from using 1-Nearest Neighbor estimates as control variates. This novel estimate is shown to achieve the optimal convergence rate in $\mathrm{O}(n^{-1/2}n^{-1/d})$ for Lipschitz functions. To the best of found knowledge, obtaining the optimal convergence rate for general probability measure makes this method the first of its kind. The most remarkable properties of the *control neighbors* estimate are:

(a) The *control neighbors* estimate can be obtained under the same framework as standard Monte Carlo, *i.e.*, as soon as one can both *(i)* draw random particles from π and *(ii)* evaluate the integrand f. Contrary to the classical control variates framework (Portier and Segers, 2019), the proposed estimate does not require the existence of control variates with known integrals.

(b) *control neighbors* takes the form of a linear integration rule $\sum_{i=1}^{n} w_{i,n} f(X_i)$ where the weights $w_{i,n}$ do not depend on the integrand f but only on the sampled particles $X_1, \ldots, X_n$. This key property allows computational benefits when several integrals are to be computed with respect to the same density π.

(c) The convergence rate is shown to be optimal for Lipschitz functions, *i.e.*, the integration error decreases as $\mathrm{O}(n^{-1/2}n^{-1/d})$ whenever f is Lipschitz (Novak, 2016). Other approaches (for general measure π) that have been developed recently, e.g., (Oates et al., 2017; Portier and Segers, 2019) do not achieve this rate.

(d) Since the weights $w_{n,i}$ are built using nearest neighbor estimates, complete practical tools are already available, including effective nearest neighbor search with k-dimensional tree (Bentley, 1975) and efficient compression and parallelization (Pedregosa et al., 2011; Johnson et al., 2019).

(e) The proposed approach is *post-hoc* in the sense that it can be run after sampling the particles and independently from the sampling mechanism. In particular, it implies that the approach can be implemented for other sampling design including MCMC or adaptive importance sampling.

Section 4.2 presents a unified view of the control functionals framework and motivates the use of nearest neighbor estimates acting as control variates. Then, the mathematical foundations of nearest neighbor estimates are gathered in Section 4.3. The theoretical properties of the proposed *control neighbors* estimates are stated in Section 4.4. Finally, Section 4.5 contains several convincing numerical experiments.

4.2 From control functionals to the method of Control Neighbors

4.2.1 General view of control functionals

The goal of this section is to introduce the framework and the main ideas of control functionals. By considering several examples, we present the key ingredients of the proposed approach of *control neighbors*.

Consider the classical *numerical integration* problem where given a target density function π on $\mathbb{R}^d$ and a squared-integrable function $f \in L_2(\pi)$, the goal is to compute

$$\pi(f) = \mathbb{E}_\pi[f(X)] = \int_{\mathbb{R}^d} f(x)\pi(x)\mathrm{d}x.$$

The standard Monte Carlo estimate approximates this value by using independent samples $X_1, \ldots, X_n$ drawn from π and takes the average as

$$\hat{\pi}_n^{(MC)}(f) = \frac{1}{n}\sum_{i=1}^{n} f(X_i).$$

This unbiased estimate is consistent and provided that $f(X_1)$ has finite variance, it satisfies the central limit theorem. In particular, it converges to $\pi(f)$ at the rate $\mathrm{O}(n^{-1/2})$ which may be prohibitive for complex statistical methods where the integrand f is expensive to evaluate. While the use of control variates has been recognized as a useful variance reduction tool in many situations (Glasserman, 2004; Owen, 2013), it is only recently that control variates have been cast into a general functional approximation problem (Oates et al., 2017; Portier and Segers, 2019). This method of control variates or *control functionals* consists of two steps as indicated below.

Control functionals main steps.

1. Build a surrogate function $\hat{f}$ with known integral $\pi(\hat{f})$.
2. Use the centered variables $\hat{f}(X_i) - \pi(\hat{f})$ to derive the following enhanced Monte Carlo estimate with control variates

$$\hat{\pi}_n^{(CV)}(f) = \frac{1}{n}\sum_{i=1}^{n}\left\{ f(X_i) - \left(\hat{f}(X_i) - \pi(\hat{f})\right)\right\}.$$

Whenever the function $\hat{f}$ is constructed from another *surrogate* sample $\tilde{X}_1, \ldots, \tilde{X}_N$ being either deterministic or independent from $X_1, \ldots X_n$, the error analysis is simple and can be conducted using the mean squared error conditionally to $\tilde{X}_1, \ldots, \tilde{X}_N$. It gives the following proposition in which the integrated mean squared error of $\hat{f}$ estimating f, $\int \mathbb{E}[(f - \hat{f})^2]\mathrm{d}\pi$ plays an important role.

Proposition 4.1. *Let $(X_1, \ldots X_n)$ be an independent and identically distributed collection of random variables with distribution π. Suppose that $\hat{f}$ depends only on a surrogate sample $\tilde{X}_1, \ldots, \tilde{X}_N$ which is independent from $(X_1, \ldots X_n)$, then*

$$\mathbb{E}\left[|\hat{\pi}_n^{(CV)}(f) - \pi(f)|^2\right] \leq n^{-1}\mathbb{E}\Big[\int (f - \hat{f})^2 \mathrm{d}\pi\Big].$$

Clearly, the success of the approach depends on the size of the (random) squared L_2-error $\int (f - \hat{f})^2 \, \mathrm{d}\pi$. This promotes the use of the most accurate estimate $\hat{f}$ of f in the functional space $L_2(\pi)$. To address this function approximation problem, many different control functional estimates have been investigated in the literature.

The use of reproducing kernel Hilbert spaces (RKHS) is considered in Oates et al. (2017). Ordinary least squares control variates based on different function bases are promoted in Portier and Segers (2019), Leluc et al. (2021) and South et al. (2022). Observe that these methods are based on regression models and may be suboptimal by not exploiting the noiseless nature of the integrand. Indeed, note that in the Monte Carlo framework, the integrand f is accessible without noise. As a consequence, expected convergence rates of the L_2-error is then $n^{-1/d}$ (Kohler and Krzyżak, 2013) rather than $n^{-1/(d+2)}$ as in standard regression (Stone, 1982). Reaching the optimal convergence rate when estimating f by the control variate is the cornerstone to speed up the convergence rate of Monte Carlo integration.

Consider now the control variate approach described in Chopin and Gerber (2022) which is related to the stratification method of Haber (1966) (see their section 2.1). Suppose that the support of π is $[0,1]^d$ and let $\{\tilde{X}_1, \ldots, \tilde{X}_N\}$ be the $(1/\ell)$-equidistant grid of $[0,1]^d$ with $N = \ell^d$, $\ell \geq 1$. The control variate estimate is then given by $\hat{f}(x) = \sum_{i=1}^N f(\tilde{X}_i) \mathbb{1}_{R_i}(x)$ where $(R_i)_{i=1,\ldots,N}$ is the partition of $[0,1]^d$ made of the rectangles induced by the elements of the grid. Standard results give $\int (f - \hat{f})^2 = \mathrm{O}(N^{-2/d})$ and, from Proposition 4.1, one obtains that the associated integration method has convergence rate $\mathrm{O}(n^{-1/2} N^{-1/d})$. Minimizing the previous upper bound under a fixed budget $(n + N)$ implies choosing N and n of a similar order. This leads to the convergence rate $\mathrm{O}(n^{-1/2} n^{-1/d})$.

Though the optimal convergence rate is achieved by the previous rectangle control variate method, there are several important issues coming from the basic nature of the implied partitioning. These restrictive conditions may be prohibitive in many practical situations. First the support of π needs to be the unit cube so that that the equidistant grid forms a reasonable partitioning. Second, the equidistant grid requires the number of evaluations to be of the form $n = \ell^d$ which turns out to be quite restrictive in terms of computational efficiency in high-dimensional settings.

The proposed method, called *control neighbors*, is based on the following idea: use a nearest neighbor estimate for $\hat{f}$ instead of a regular grid-based estimate. Given a surrogate sample $\{\tilde{X}_1, \ldots, \tilde{X}_N\}$, let $\hat{f}$ be the 1-NN estimate of f, that is, $\hat{f}(x) = \sum_{i=1}^N f(\tilde{X}_i) \mathbb{1}_{S_{N,i}}(x)$ where $(S_{N,i})_{i=1,\ldots,N}$ are the Voronoï cells associated to the sample $\{\tilde{X}_1, \ldots, \tilde{X}_N\}$, *i.e.*, each cell $S_{N,i}$ contains all the points that are closer to $\tilde{X}_i$ than any other point within the surrogate sample. The resulting method is similar to the rectangle approach described before as a partitioning estimate is also employed. However, note that with this approach, the surrogate sample $\{\tilde{X}_1, \ldots, \tilde{X}_N\}$ can be any set of points within the support of π. Hence, neither a strong assumption on the support of π is needed, nor a restriction on the computational budget n.

Finally, by following a "leave-one-out" strategy, the control variate estimate is built directly from the initial sample $\{X_1, \ldots, X_n\}$ which allows to ultimately reduce the number of evaluations of f from $(n + N)$ to only n.

4.2.2 Control Neighbors estimates

For any $i = 1, \dots, n$, denote by $\hat{f}_n^{(i)}$ the 1-NN estimate of f constructed without the i-th sample point and $\mathcal{X}_n^{(i)} = \{X_1, \dots, X_n\} \setminus X_i$. Introduce the following *control neighbors* Monte Carlo estimate

$$\hat{\pi}_n^{(\mathrm{NN-loo})}(f) = \frac{1}{n} \sum_{i=1}^{n} \left\{ f(X_i) - \left(\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)}) \right) \right\}, \tag{4.1}$$

in which the function $\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)})$ acts as control variate. A simple conditioning argument implies that $\mathbb{E}[\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)})] = \mathbb{E}[\mathbb{E}[\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)}) \,|\, \mathcal{X}_n^{(i)}]] = 0$ which is sufficient to get

$$\mathbb{E}\left[\hat{\pi}_n^{(\mathrm{NN-loo})}(f) \right] = \pi(f).$$

Note that the n additional evaluations $\hat{f}_n^{(i)}(X_i)$ are not computationally difficult as no additional evaluations of f are necessary. However, computing the terms $\pi(\hat{f}_n^{(i)}))$, for $i = 1, \dots, n$ requires the evaluation of n additional integrals. Intuitively, since $\hat{f}_n^{(i)}$ is similar to $\hat{f}_n$ then their integral values should be close. This is stated in the Appendix and one consequence is that

$$\frac{1}{n} \sum_{i=1}^{n} \pi(\hat{f}_n^{(i)}) = \pi(\hat{f}_n) + \mathrm{O}_{\mathbb{P}}(n^{-1/2} n^{-1/d}).$$

Based on this remark, one may replace the n integral evaluations $\pi(\hat{f}_n^{(i)})$ by only a single integral $\pi(\hat{f}_n)$ to compute. This gives the following *control neighbors* Monte Carlo estimate

$$\hat{\pi}_n^{(\mathrm{NN})}(f) = \frac{1}{n} \sum_{i=1}^{n} \left\{ f(X_i) - \left(\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n) \right) \right\}. \tag{4.2}$$

Both estimates (4.1) and (4.2) may be written as linear integration rules with weights that do not depend on the integrand (see Section 4.4 below). In practice, the working estimate is the control neighbor estimate (4.2) as it involves less integrals computations.

4.2.3 Control Neighbors implementation

We end this section by specifying the algorithm for computing the *control neighbors* estimate (4.2). This estimate is based on the evaluations $f(X_i)$ of the integrand and the evaluations $\hat{f}_n^{(i)}(X_i)$ of the leave-one-out nearest neighbors estimates. It also requires to compute the integral $\pi(\hat{f})$ of the 1-NN estimate $\hat{f}_n$. Several practical remarks regarding the computations of all these quantities are given right after Algorithm 4.6.

Remark 4.2 (Tree search)**.** *The naive neighbor search implementation involves the brute-force computation of distances between all pairs of points in the training samples and may be computationally prohibitive. To address such practical inefficiencies, a variety of tree-based data structures have been invented so the cost of a nearest neighbors search can be reduced. The KD-Tree (Bentley, 1975) is a binary tree structure which recursively partitions the parameter space along the data axes, dividing it into nested orthotropic regions into which data points are filed. Once constructed, the query of a*

Algorithm 4.6 Control Neighbors for Monte Carlo integration

Require: integrand f, density π, number of samples n.

1. Generate an independent random sample $X_1, \dots, X_n$ from π.
2. Compute evaluations $f(X_1), \dots, f(X_n)$.
3. Compute nearest neighbors evaluations $\hat{f}_n^{(1)}(X_1), \dots, \hat{f}_n^{(n)}(X_n)$.
4. Compute integral of nearest neighbor estimate $\pi(\hat{f}_n)$.
5. Return $\hat{\pi}_n^{(\mathrm{NN})}(f) = \frac{1}{n}\sum_{i=1}^n \left\{ f(X_i) - (\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n)) \right\}$.

nearest neighbor in a KD-Tree can be done in logarithmic time. However,in high dimension, the query cost increases and the structure of Ball-Tree (Omohundro, 1989) is favored. Where KD trees partition data along Cartesian axes, Ball trees partition data in a series of nesting hyper-spheres, making tree construction more costly than KD tree, but results in a efficient data structure even in very high dimensions. In practice, there exists many software libraries containing implementations of KD-tree and Ball-Tree with efficient compression and parallelization (Pedregosa et al., 2011; Johnson et al., 2019).

Remark 4.3 (Evaluation of $\hat{f}_n^{(i)}(X_i)$)**.** *The evaluations of the leave-one-out nearest neighbors estimates can be efficiently computed with nearest neighbor search and masks evaluations. More precisely, denote by $\mathcal{F} = [f(X_1), \dots, f(X_n)]$ the vector of evaluations of the integrand. Any query of a nearest neighbor algorithm produces a vector containing the indices of neighbors of the corresponding query points. After fitting a KD-Tree on the particles $X_1, \dots, X_n$, one can query the 2-nearest neighbor of each X_i to produce the vector of indices $\mathcal{I}$ such that $\mathcal{I}_i$ is the index of the nearest neighbor of X_i among $\mathcal{X}_n^{(i)}$. The leave-one-out evaluations $[\hat{f}_n^{(1)}(X_1), \dots, \hat{f}_n^{(n)}(X_n)]$ are then simply obtained using the slicing operation on array $\mathcal{F}[\mathcal{I}]$.*

Remark 4.4 (Evaluation of $\pi(\hat{f}_n)$)**.** *In the case of a complex probability measure π, the Voronoi volumes may be hard to compute but can always be approximated. The integral of the nearest neighbor estimate $\pi(\hat{f}_n)$ may be replaced by a Monte Carlo estimate that uses M particles. That is $\pi(\hat{f}_n) \simeq M^{-1}\sum_{i=1}^M \hat{f}_n(\tilde{X}_i)$ where the variables $\tilde{X}_i$ are drawn independently from π. Observe that such an approach does not involve additional evaluations of f. The error of this naive Monte Carlo approximation is in $\mathrm{O}(M^{-1/2})$ meaning that large values of the form $M = n^2$ and $M = n^3$ can be taken to compare with the optimal convergence rate in $\mathrm{O}(n^{-1/2}n^{-1/d})$ of the control neighbors estimate.*

Remark 4.5 (Voronoi volume when π is uniform)**.** *The quantity $\pi(\hat{f})$ may be written as a sum of the evaluations $f(X_i)$ weighted by the value of the Voronoi volumes associated to the corresponding sample point X_i (see Definition 4.9 in the next section). In case the measure π is the uniform measure on $[0,1]^d$, one may be able to explicitly compute those volumes. Starting from the pioneer work of Richards (1974) in the context of protein structures, there has been advances to perform efficient Voronoi volume computations using Delaunay triangulations and taking advantage of graphic hardware (Hoff III et al., 1999). For $2d$ and $3d$ Voronoi diagrams, one can refer to the software Voro++ (Rycroft, 2009): a software library for carrying out computations of the Voronoi tessellation. Note however that this type of algorithm are subjected to the curse of dimensionality and might be inefficient when d is large.*

Remark 4.6 (k-NN estimates)**.** *A natural variant of the proposed method is obtained by replacing the* 1*-NN estimate* $\hat{f}_n$ *in Eq.*(4.2) *by a* k*-NN estimate* $\hat{f}_n^{(k)}$ *which averages the evaluations of the* k *nearest neighbors of a given point. This estimate is defined by* $\hat{f}_n^{(k)}(x) = k^{-1}\sum_{j=1}^{k} f(\hat{N}_{n,j}(x))$ *where* $\hat{N}_{n,j}(x)$ *is the* j*-nearest neighbor of* x*. Note that it involves both the tuning of the hyper-parameter* $k \geq 1$ *and some extra computation due to the associated nearest neighbors search. In regression or classification, high values of* k *can reduce the variance of the estimate by averaging the model noise at the cost of added computations. In contrast, the control neighbors estimate* $(k = 1)$ *is free of these additional costs and it takes advantage of the noiseless evaluations (see Chapter 15 in Biau and Devroye (2015)) of the integrand mentioned in Section 4.2.1.*

4.3 Nearest Neighbor estimation

This section presents the mathematical framework of nearest neighbor estimates with reminders on Voronoi cells and central quantities for the analysis, namely the degree of a point and the average cell volume. Throughout this section, we consider the following assumptions which are related to the *strong density assumption* of Audibert and Tsybakov (2007). This condition ensures that the density π has a *regular* support and that it is bounded away from zero and infinity.

(A1) $X, X_1, X_2, \ldots$ are independent and identically distributed random vectors in $\mathbb{R}^d$ drawn from the density π having support $\mathcal{X} = \{x : \pi(x) > 0\}$.

(A2) There exists $0 < b, U < +\infty$ and $c, r_0 > 0$ such that:

- $\forall x \in \mathcal{X}, \quad b \leq \pi(x) \leq U.$
- $\forall 0 < r \leq r_0, \forall x \in \mathcal{X}, \quad \lambda(\mathcal{X} \cap B(x, r)) \geq c\lambda(B(x, r)).$

The existence of a density function π facilitates the introduction of nearest neighbor estimates and related quantities. Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$ and x be a given point in $\mathbb{R}^d$. As the sample $\|x - X_1\|, \ldots, \|x - X_n\|$ is independently generated from a continuous distribution defined on $\mathbb{R}$, we have, with probability one, the existence of a unique minimum value among the previous collection of positive numbers and therefore a unique nearest neighbor to x. However to establish formulas valid for any given sample points in $\mathbb{R}^d$ and any $x \in \mathbb{R}^d$, it is convenient to take care of the presence of ties when introducing nearest neighbor and associated distance. This is done in the next definition using, when a tie occurs, the indexes of the concerned points to select the one nearest neighbor. Though convenient to obtain some formulas in the proofs, this arbitrary choice does not affect the results of the chapter.

Definition 4.7 (Nearest neighbors and distances)**.** *Given any point* $x \in \mathbb{R}^d$ *and any collection* $X_1, \ldots, X_n$ *in* $\mathbb{R}^d$*, define* $\hat{N}_n(x)$ *as the nearest neighbor of* x *among* $X_1, \ldots, X_n$ *and* $\hat{\tau}_n(x)$ *the associated distance, i.e.,*

$$\hat{N}_n(x) \in \underset{Y \in \{X_1, \ldots, X_n\}}{\arg\min} \|x - Y\|, \qquad \hat{\tau}_n(x) = \|\hat{N}_n(x) - x\|.$$

When the above arg min *is not unique, then* $\hat{N}_n(x)$ *is defined as the one point among the* arg min *having smallest index.*

The next Lemma follows from standard considerations in the k-NN literature (Biau and Devroye, 2015) and relies on the uniform lower bounds required in (A2). Observe that this condition plays an important role in the analysis of nearest neighbors estimates as it allows a uniform control on the radius of the Voronoï cells. Such a uniform bound on the radius is the key to study the convergence of general k-NN estimates. When dealing with densities having general supports, one can also consider some minimal mass assumption (Gadat et al., 2016) to guarantee that no region has no point. Futhermore, note that this question of necessary conditions for general uniform bounds remains an active field of research with recent progress for unbounded data (Kohler et al., 2006) and relaxations through tail assumptions (Gadat et al., 2016). Extending the present analysis to such general measure is beyond the scope of the present chapter and left for further research.

Let Γ denote the standard gamma function and V_d be the volume of the unit ball, *i.e.*, $V_d = \int_{B(0,1)} \mathrm{d}x$ with $B(0,1) = \{x \in \mathbb{R}^d \,:\, \|x\| \leq 1\}$.

Lemma 4.8 (Upper bound of distance moments)**.** *Under (A1) and (A2), we have, for any $q \geq 1$,*

$$\forall x \in \mathcal{X}, \quad \mathbb{E}[\hat{\tau}_n(x)^q] \leq \big(nV_d bc\big)^{-q/d}\,\Gamma(q/d+1).$$

The sample $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ defines a natural (random) partition of the integration domain when considering the associated Voronoi cells. Any such cell is associated to a given sample point, say X_i, and contains all the points x such that their nearest neighbor is X_i, as detailed below.

Definition 4.9 (Voronoi cells and volumes)**.** *The Voronoi cells are given by*

$$\forall i = 1, \ldots, n, \qquad S_{n,i} = \{x \in \mathbb{R}^d \,:\, \hat{N}_n(x) = X_i\},$$

and its associated Voronoi volume is $V_{n,i} = \pi(S_{n,i})$.

Voronoi cells are strongly related to the 1-nearest neighbor predictor. The 1-NN estimate of f is simply defined as $\hat{f}_n(x) = f(\hat{N}_n(x))$ for all $x \in \mathbb{R}^d$. As a consequence, it is piece-wise constant on the Voronoi cells, i.e., $\hat{f}_n(x) = \sum_{i=1}^n f(X_i)\mathbb{1}_{S_{n,i}}(x)$. A useful property for the analysis is the following regularity condition on f.

(A3) The function $f : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz with respect to $\|\cdot\|$, *i.e.* there is $L > 0$ such that

$$\forall x, y \in \mathbb{R}^d, \quad |f(x) - f(y)| \leq L\|x - y\|.$$

When the function f is Lipschitz, the distance $\hat{\tau}_n$ is key in the analysis of the functional approximation problem of f by the estimate $\hat{f}_n$. This is stated in the next Lemma whose proof is in the Appendix.

Lemma 4.10 (1-NN estimation of f)**.** *Under (A1), (A2) and (A3), we have almost surely $|\hat{f}_n(x) - f(x)| \leq L\hat{\tau}_n(x)$.*

The leave-one-out rule is a general technique to introduce independence between the prediction and the evaluation point. It is used as a cross-validation strategy in order to tune hyper-parameters of certain procedure (Stone, 1974; Craven and Wahba, 1978). The leave-one-out version of $\hat{f}_n$ without the i-th sample is denoted $\hat{f}_n^{(i)}$ and is obtained

in the exact same way as $\hat{f}_n$ except that a slightly different sample - in which the i-th observation has been removed - is used. It is therefore useful to introduce the leave-one-out nearest neighbor and the leave-one-out Voronoi cells.

Definition 4.11 (Leave-one-out neighbors, Voronoi cells and volumes)**.** *Let $\mathcal{X}_n^{(i)} = \{X_1, \ldots, X_n\} \backslash X_i$. The leave-one-out neighbor of x is given by*

$$\hat{N}_n^{(i)}(x) \in \underset{Y \in \mathcal{X}_n^{(i)}}{\arg\min} \|x - Y\|.$$

When the above arg min *is not unique, then $\hat{N}_n^{(i)}(x)$ is defined as the one point among the* arg min *having smallest index. The leave-one-out Voronoi cell $S_{n,j}^{(i)}$ denotes the j-th Voronoi cell in $\mathcal{X}_n^{(i)}$, i.e.*

$$\forall i \neq j \in \{1, \ldots, n\} \qquad S_{n,j}^{(i)} = \{x \in \mathbb{R}^d : \hat{N}_n^{(i)}(x) = X_j\}.$$

The leave-one-out Voronoi volume is defined as $V_{n,j}^{(i)} = \pi(S_{n,j}^{(i)})$.

Now that we have at hand the previous definition, we can introduce formally the leave-one-out 1-NN predictor $\hat{f}_n^{(i)}(x) = f(\hat{N}_n^{(i)}(x))$ (which was already used in previous section to define the proposed integral estimate). A key property is that $\hat{f}_n^{(i)}$ and $\hat{f}_n$ coincide on $S_{n,j}$ for $j \neq i$. On the cell $S_{n,i}$, when the function f is Lipschitz, their difference is of the same order as the nearest neighbor distance. In terms of the L_1-norm, their difference is even smaller as the cell $S_{n,i}$ has a small volume. Relevant for our numerical integration problem is that the average of the integrals $\pi(\hat{f}_n^{(i)})$ is close to $\pi(\hat{f}_n)$, as stated in the following proposition.

Lemma 4.12. *Let $\bar{f}_n(x) = \sum_{i=1}^n \hat{f}_n^{(i)}(x) \mathbb{1}_{S_{n,i}}(x)$. Under (A1) and (A2), we have*

$$\sum_{i=1}^{n} \{\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\} = \pi(\bar{f}_n - \hat{f}_n).$$

A central quantity that reflects how much a point is surrounded within the sample is given by enumerating how many times a point, say X_i, is the nearest neighbor of points from the sample $\mathcal{X}_n^{(i)}$. Another important quantity that qualifies the isolation of a point is obtained by summing the Voronoi volumes. These two notions are formally stated in the next definition.

Definition 4.13 (Degree and cumulative volume)**.** *For all $j = 1, \ldots, n$, the degree $\hat{d}_{n,j}$ represents the number of times X_j is a nearest neighbor of a point X_i for $i \neq j$. The associated j-th cumulative Voronoi volume is denoted by $\hat{c}_{n,j}$, that is*

$$\hat{d}_{n,j} = \sum_{i:i \neq j} \mathbb{1}_{S_{n,j}^{(i)}}(X_i), \qquad \hat{c}_{n,j} = \sum_{i:i \neq j} V_{n,j}^{(i)}.$$

Interestingly, the degree of a point and its cumulative Voronoi volume have the same expectation.

Lemma 4.14. *Under (A1) and (A2), it holds that $\mathbb{E}[\hat{d}_{n,j}] = \mathbb{E}[\hat{c}_{n,j}] = 1$.*

The two quantities $\hat{d}_{n,j}$ and $\hat{c}_{n,j}$ will be useful in the next section to express the control neighbors estimate as a linear integration rule. For now, one can compute weighted sum of $f(X_j)$ using $\hat{d}_{n,j}$ and $\hat{c}_{n,j}$ as weights and notice that these weighted sum are related to the leave-one-out estimate.

Lemma 4.15. *Under (A1) and (A2), it holds that*

$$\sum_{i=1}^{n} f(X_i)\,\hat{d}_{n,i} = \sum_{i=1}^{n} \hat{f}_n^{(i)}(X_i) \qquad \text{and} \qquad \sum_{i=1}^{n} f(X_i)\,\hat{c}_{n,i} = \sum_{i=1}^{n} \pi(\hat{f}_n^{(i)}).$$

4.4 Main results

This section gathers the main theoretical properties of the *control neighbors* estimates (4.1) and (4.2) presented in Section 4.2.3. First, these estimates can be written as simple linear integration rules with weights that only depend on the nearest neighbor estimates and may be efficiently computed in practice. Then the convergence rate of the error $\mathbb{E}[|\hat{\pi}_n(f) - \pi(f)|^2]^{1/2}$ is derived for the proposed estimates.

4.4.1 Linear integration rules

The *control neighbors* estimates $\hat{\pi}_n^{(\mathrm{NN})}(f)$ and $\hat{\pi}_n^{(\mathrm{NN-loo})}(f)$ can be expressed as linear integration rules of the form $\sum_{i=1}^n w_{i,n} f(X_i)$ where the weights $w_{i,n}$ do not depend on the integrand g. The integration weights involve the degrees $\hat{d}_{n,i}$, the (cumulative) volumes $V_{n,i}$ and $\hat{c}_{n,i}$ in Definition 4.13.

Proposition 4.16 (Quadrature rules)**.** *The estimates $\hat{\pi}_n^{(\mathrm{NN})}(f)$ and $\hat{\pi}_n^{(\mathrm{NN-loo})}(f)$ can be expressed as linear estimates of the form*

$$\hat{\pi}_n^{(\mathrm{NN})}(f) = \sum_{i=1}^{n} w_{i,n}^{(\mathrm{NN})} f(X_i) \quad \text{and} \quad \hat{\pi}_n^{(\mathrm{NN-loo})}(f) = \sum_{i=1}^{n} w_{i,n}^{(\mathrm{NN-loo})} f(X_i)$$

where $w_{i,n}^{(\mathrm{NN})} = (1 + nV_{n,i} - \hat{d}_{n,i})/n$ and $w_{i,n}^{(\mathrm{NN-loo})} = (1 + \hat{c}_{n,i} - \hat{d}_{n,i})/n$.

In light of the previous proposition, the proposed approach consists in a simple modification of $\hat{\pi}_n^{(\mathrm{NN-loo})}(f)$ as we can recover $\hat{\pi}_n^{(\mathrm{NN})}(f)$ from $\hat{\pi}_n^{(\mathrm{NN-loo})}(f)$ by replacing $\hat{c}_{n,j}$, which requires to compute $n-1$ Voronoï volumes, by $nV_{n,i}$. The difference between both is in fact of order $n^{-1/2-1/d}$ as shown in the next section.

4.4.2 Convergence rate of the leave-one-out version

The first result provides a finite-sample bound on the mean-squared error of the leave-one-out *control neighbors* estimate.

Proposition 4.17. *Under (A1), (A2) and (A3), if $n \geq 4$, then*

$$\mathbb{E}\left[\left|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)\right|^2\right]^{1/2} \leq C_{\mathrm{NN-loo}} n^{-1/2} n^{-1/d},$$

where $C_{\mathrm{NN-loo}} = 16L\left(V_d bc\right)^{-1/d}(U/bc)^{-1/2}$.

Interestingly, the rate obtained before matches the complexity rate described in Novak (2016). Note that the results in the aforementioned paper are concerned about a slightly more precise context as they assert that no random integration rule (see the paper for more details) can reach a better accuracy – measured in terms of mean-squared error – than $\mathrm{O}(n^{-1-2/d})$ when the integration measure is the uniform measure over the unit cube and the function f is Lipschitz. Proposition 4.17 states that the optimal rate is in fact achieved by some integration rule in situations where the integration measure's density is not necessarily uniform but only lower and upper bounded.

4.4.3 Back to the proposed estimate

The leave-one-out version, though its rate of convergence matches the optimal rate, suffers from the difficulty of computing n integrals value which might represent a computational burden. The proposed estimate $\hat{\pi}_n^{(\mathrm{NN})}$ is actually a mild modification of the leave-one-out estimate as

$$\hat{\pi}_n^{(\mathrm{NN})}(f) - \hat{\pi}_n^{(\mathrm{NN-loo})}(f) = \pi(\hat{f}_n) - \frac{1}{n}\sum_{i=1}^{n} \pi(\hat{f}_n^{(i)})$$

that benefits from computational advantages (as detailed in Section 4.2 in the remarks stated after the algorithm). Based on the previous property and using Lemma 4.12 from previous section, we can obtain that the mean-squared distance between the leave-one-out version and the proposed estimate is of order $\mathrm{O}(n^{-1/2-1/d})$ as $n \to \infty$ (a precise statement is given in the Appendix, Lemma 4.7). Therefore, one obtains that $\hat{\pi}_n^{(\mathrm{NN})}$ has the same convergence rate as $\hat{\pi}_n^{(\mathrm{NN-loo})}$.

Proposition 4.18. *Under (A1), (A2) and (A3), we have*

$$\mathbb{E}\left[\left|\hat{\pi}_n^{(\mathrm{NN})}(f) - \pi(f)\right|^2\right]^{1/2} \leq C_{\mathrm{NN}} n^{-1/2} n^{-1/d},$$

where $C_{\mathrm{NN}} = 39L(V_d bc)^{-1/d}(U/bc)^{-1/2}$.

4.5 Numerical experiments

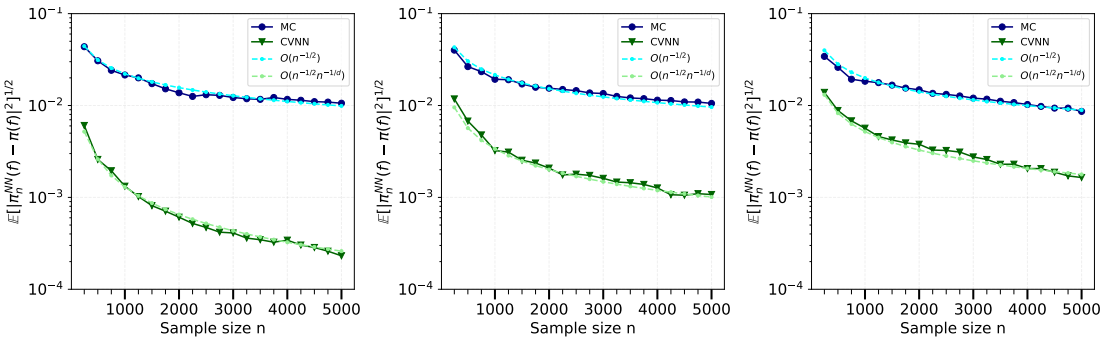
To illustrate the finite-sample performance of the proposed estimator, we first present in Section 4.5.1 synthetic data examples involving two standard integration problems with uniform and Gaussian measures. Then Section 4.5.2 presents an application of the method in finance for Monte Carlo exotic option pricing under the standard Black-Scholes model with constant volatility and the more difficult Heston model with stochastic volatility. Finally Section 4.5.3 deals with marginalising hyper-parameters in Bayesian models. This is the same case study used in Oates et al. (2017) where the evaluations of the integrand f are very costly. This framework is particularly well suited for the nearest neighbor control variates estimate. In all the experiments, the method MC represents the naive Monte Carlo estimate and CVNN returns the value of $\hat{\pi}_n^{(\mathrm{NN})}(f)$ for which the integral $\int \hat{f}_n \mathrm{d}\pi$ is replaced by a Monte Carlo estimate that uses $M = n^2$ particles.

4.5.1 Simulated data

The aim of this section is to empirically validate the $\mathrm{O}(n^{-1/2}n^{-1/d})$ convergence rate of the *control neighbors* estimate. Similarly to Oates et al. (2017), consider the integrands

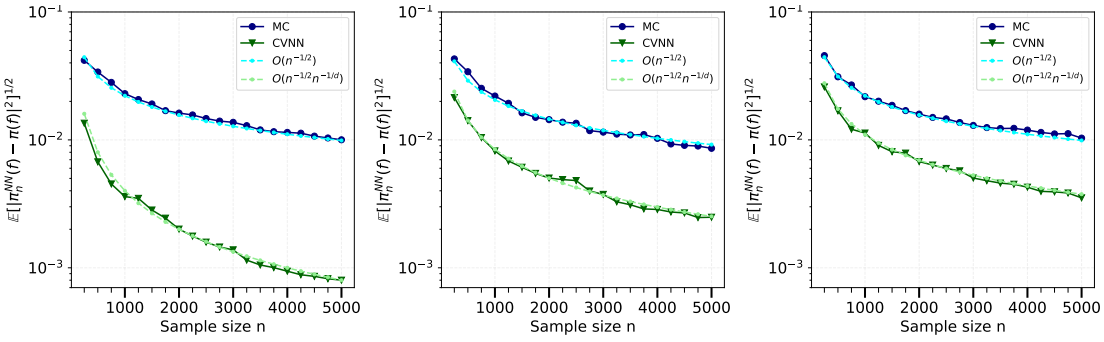
$$f_1(x_1,\dots,x_d) = \sin\Big(\pi\Big(\frac{2}{d}\sum_{i=1}^{d} x_i - 1\Big)\Big), \qquad f_2(x_1,\dots,x_d) = \sin\Big(\frac{\pi}{d}\sum_{i=1}^{d} x_i\Big).$$

The goal is to compute $\int f_1(x)\mathbb{1}_{[0,1]^d}(x)\mathrm{d}x$ and $\int f_2(x)\varphi(x)\mathrm{d}x$ where $\varphi(\cdot)$ denotes the probability density function of the multivariate Gaussian distribution $\mathcal{N}(0, I_d)$. Different dimensions $d \in \{2;4;6\}$ are considered and the sample size evolves from $n = 250$ to $n = 5\,000$. Figures 4.1 and 4.2 display the evolution of the root mean squared error $n \mapsto \mathbb{E}[|\hat{\pi}_n^{(\mathrm{NN})}(f) - \pi(f)|^2]^{1/2}$ for integrands f_1 and f_2 respectively, where the expectation is computed over 100 independent replications.



(a) $f_1, d = 2$ (b) $f_1, d = 4$ (c) $f_1, d = 6$

Figure 4.1 – Root mean squared errors obtained over 100 replications for functions f_1 in dimension $d \in \{2;4;6\}$ (left to right).



(a) $f_2, d = 2$ (b) $f_2, d = 4$ (c) $f_2, d = 6$

Figure 4.2 – Root mean squared errors obtained over 100 replications for function f_2 in dimension $d \in \{2;4;6\}$ (left to right).

Interestingly, the different error curves validate the optimal convergence rate in $\mathrm{O}(n^{-1/2}n^{-1/d})$ for the *control neighbors* estimate. For small dimensions ($d = 2$ and $d = 4$), the root mean squared error of the CVNN estimate can be reduced by a factor ten compared to the standard Monte Carlo approach.

4.5.2 Monte Carlo Option Pricing

Finance background. Options are financial derivatives based on the value of underlying securities. They give the buyer the right to buy (call option) or sell (put option) the underlying asset at a pre-determined price within a specific time frame. The price of an option may be expressed as the expectation, under the so-called risk-neutral measure, of the payoff discounted to the present value. Consider a contract of European type, which specifies a payoff $V(S_T)$, depending on the level of the underlying asset S_t at maturity $t = T$. The value V of the contract at time $t = 0$, conditional on an underlying value S_0 is given by

$$V(S_0) = \mathbb{E}_Q[e^{-rT}V(S_T)], \tag{4.3}$$

where $\mathbb{E}_Q$ denotes the expectation under the risk-neutral measure and r is the risk-free interest rate. Such a representation suggests a straightforward Monte Carlo based method for its calculation by simulating random paths of the underlying asset, calculating each time the resulting payoff and taking the average of the result. This approach is particularly useful when dealing with exotic options which present no closed-form expression as is the case for barrier options.

Barrier options (Merton, 1973) are considered exotic options because they are more complex than basic American or European options. Barrier options are also considered a type of path-dependent option because their value fluctuates as the underlying's value changes during the option's contract term. In other words, a barrier option's payoff is based on the underlying asset's price path. The option becomes worthless or may be activated upon the crossing of a price point barrier denoted H. More precisely, Knock-Out (KO) options are options that expire worthless when the underlying's spot crosses the prespecified barrier level whereas Knock-In (KI) options only come into existence if the prespecified barrier level is crossed by the underlying asset's price.

The payoff of a European call option with strike price K is given by $V(S_T) = (S_T - K)_+$ and depends only on the level of the underlying asset S_t at maturity time $t = T$. In contrast, the payoff a of barrier option depends on the path $(S_t)_{t\in[0,T]}$. The payoffs of *up-in* (UI) and *up-out* (UO) barrier options with barrier price K are given by

$$V_{(\mathrm{UI})}(S) = (S_T - K)_+ \mathbb{1}\{\max_{t\in[0,T]} S_t \geq H\}, \tag{4.4}$$

$$V_{(\mathrm{UO})}(S) = (S_T - K)_+ \mathbb{1}\{\max_{t\in[0,T]} S_t < H\}. \tag{4.5}$$

Market Dynamics. The Black–Scholes model (Black and Scholes, 1973) is a mathematical model for pricing option contracts. It is based on geometric Brownian motion with constant drift and volatility so that the underlying stock S_t satisfies the following stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where μ represents the drift rate of growth of the underlying stock, σ is the volatility and W denotes a Wiener process. Although simple and widely used in practice, the Black-Scholes model has some limitations. In particular, it assumes constant values for the risk-free rate of return and volatility over the option duration. Neither of those necessarily remains constant in the real world. The Heston Model (Heston, 1993) is a type of stochastic volatility model that can be used for pricing options on various securities. For the Heston model, the previous constant volatility σ is replaced by a

stochastic volatility v_t which follows an Ornstein-Uhlenbeck process. The underlying stock S_t satisfies the following equations

$$\begin{cases} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^S, \\ dv_t &= \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dW_t^v, \qquad dW_t^S dW_t^v = \rho\ dt. \end{cases}$$

with stochastic volatility v_t, drift term μ, long run average variance θ, rate of mean reversion κ and volatility of volatility ξ. Essentially the Heston model is obtained by just simulating a standard geometric Brownian motion with non-constant volatility, where the change in S has relationship ρ with the change in volatility.

Monte Carlo procedures. The application of standard Monte Carlo methods to option pricing takes the following form:

(1) Simulate a large number n of price paths for the underlying asset: $(S_{(1)}, \ldots, S_{(n)})$.

(2) Compute the associated payoff using Eq.(4.4) for the option of each path: $(V_1, \ldots, V_n)$.

(3) Average the payoffs and discount them to today: $\hat{V}_n = (e^{-rT}/n) \sum_{i=1}^n V_i$.

In practice the price paths are simulated using a Euler scheme with a discretization of the time period $[0, T]$ comprised of m times $t_1 = 0 < t_2 < \ldots < t_m = T$. Each price path S_i for $i = 1, \ldots, n$ is actually a vector $(S_{(i)}^{(1)}, \ldots, S_{(i)}^{(m)})$ so that the indicator function of the barrier options is computed on the discretized prices. Common values for m are the number of trading days per year which is $m = 252$ for $T = 1$ year.

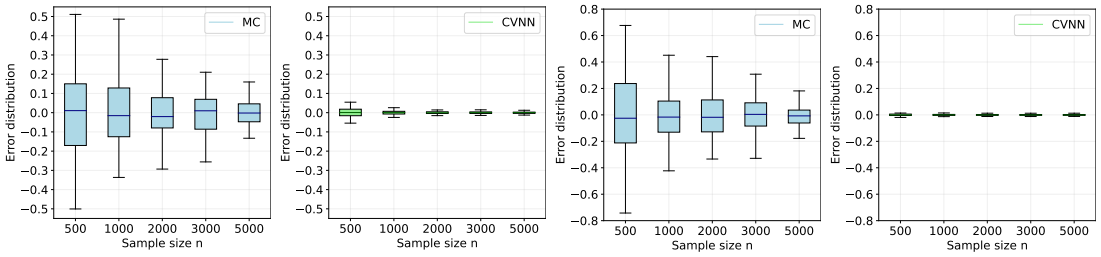
Parameters. Several numerical experiments are performed for the pricing of European Barrier call options "up-in" and "up-out". The number of sampled paths evolves as $n \in \{500; 1,000; 2,000; 3,000; 5,000\}$ and the granularity of the grid is equal to $m = 240$. Two different mathematical models are considered when simulating the underlying assets:

(1) Black-Scholes model with constant volatility $\sigma = 0.30$.

(2) Heston model with initial volatility $v_0 = 0.1$, long-run average variance $\theta = 0.02$, rate of mean reversion $\kappa = 4$, instanteneous correlation $\rho = 0.8$ and volatility of volatility $\xi = 0.9$.

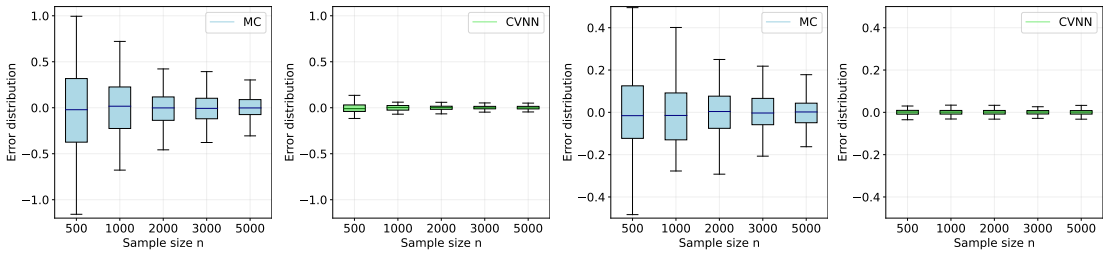
In both cases the fixed parameters are: spot price $S_0 = 100$, interest rate $r = 0.10$, maturity $T = 2$ months, strike price $K = S_0 = 100$ and barrier price $H = 130$.

Figure 4.3 below shows the error distribution of the different Monte Carlo estimates (naive MC and CVNN) for the pricing of Barrier call options "up-in" and "up-out" in the Black-Scholes model. The boxplots are computed over 100 independent replications. Accordingly Figure 4.4 gathers the results in the Heston model. The gain in terms of variance reduction is huge when using the *control neighbors* estimate compared to the standard Monte Carlo approach.



(a) "Up-In", MC (b) "Up-In", CVNN (c) "Up-Out", MC (d) "Up-Out", CVNN

Figure 4.3 – Barrier option pricing under Black-Scholes model with spot price $S_0 = 100$, strike $K = S_0$, maturity $T = 2$ months, risk-free rate $r = 0.1$, constant volatility $\sigma = 0.3$, barrier price $H = 130$. The boxplots are obtained over 100 replications.



(a) "Up-In", MC (b) "Up-In", CVNN (c) "Up-Out", MC (d) "Up-Out", CVNN

Figure 4.4 – Barrier option pricing with Heston Model with spot ptice $S_0 = 100$, strike $K = S_0$, barrier price $H = 130$, maturity $T = 2$ months, risk-free rate $r = 0.1$, initial volatility $v_0 = 0.1$, long-run average variance $\theta = 0.02$, rate of mean reversion $\kappa = 4$, instanteneous correlation $\rho = 0.8$ and volatility of volatility $\xi = 0.9$. The boxplots are obtained over 100 replications.

4.5.3 Sarcos Robot Arm

Hierarchical model. Similarly to Oates et al. (2017), we consider the problem of marginalising over hyper-parameters in a fully Bayesian treatment of hierarchical models. In this case study, the underlying model is a D-dimensional regression with Gaussian Process (GP) prior (Rasmussen, 2003). The dataset is comprised of state/response pairs $(y_i, z_i)_{i=1}^N$ with $z_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. Using a fixed and known variance parameter $\sigma > 0$ and a transformation $T : \mathbb{R}^D \to \mathbb{R}$, the regression model is given by

$$\forall i = 1, \dots, N, \quad y_i = T(z_i) + \varepsilon_i \qquad \text{with } \varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

The transformation T is taken as a GP prior $T \sim \mathcal{GP}(0, c(z_1, z_2; \theta))$ where the cost function c depends on a parameter vector $\theta = (\theta_1, \theta_2)$ that controls how the training data are used for prediction on a test point $z^\star$. The cost function is defined as

$$c(z_1, z_2; \theta) = \theta_1 \exp(-\frac{\|z_1 - z_2\|_2^2}{2\theta_2^2}).$$

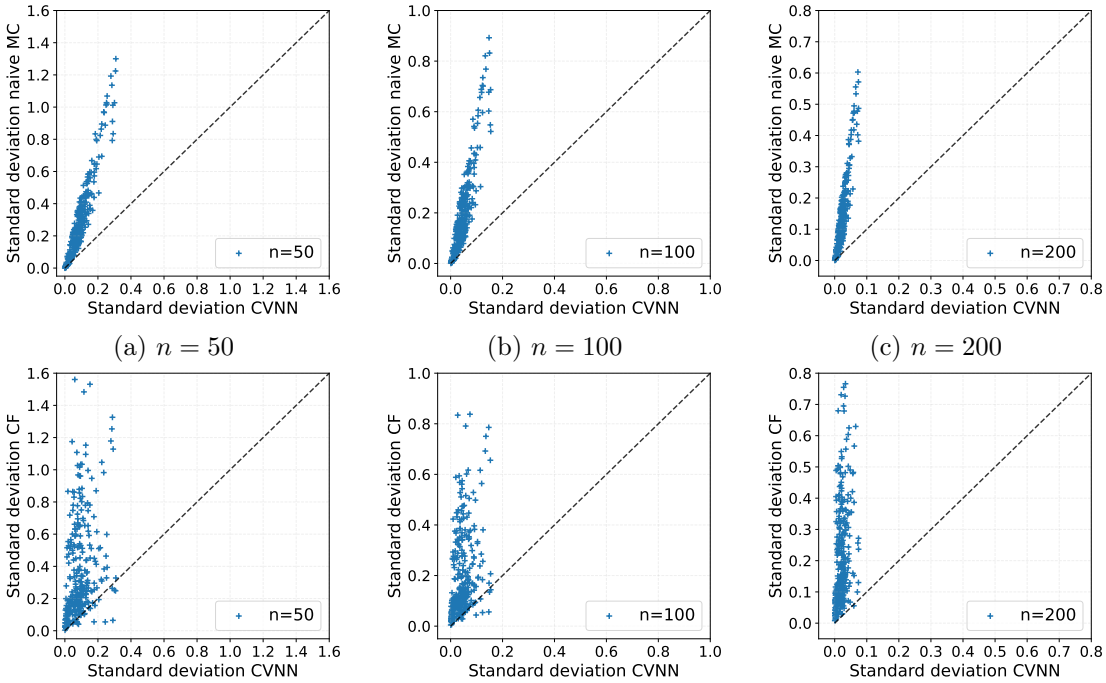
In the Bayesian framework, the parameters θ_1 and θ_2 are hyper-priors with Gamma distributions $\theta_1 \sim \Gamma(\alpha, \beta)$ and $\theta_2 \sim \Gamma(\gamma, \delta)$ in the shape/scale parameterization, which joint density is written as $\pi(\theta)$. Given an unseen state vector $z^\star$, the goal is to predict the value of the response $y^\star$ using the posterior mean $\hat{y}^\star := \mathbb{E}[y^\star|y] = \int \mathbb{E}[y^\star|y, \theta]\pi(\theta)\mathrm{d}\theta$

which is implicitly conditioned on the covariates $z_1, \ldots, z_N, z^\star$. Since this integral is intractable, one may rely on Monte Carlo estimates by sampling $\theta_1, \ldots, \theta_n$ independently from the prior $\pi(\theta)$ and evaluating the following integrand

$$\mathbb{E}[y^\star|y, \theta] = C_{*,N}(C_N + \sigma^2 I_{N\times N})^{-1} y$$

where the cost matrices are $(C_N)_{i,j} = c(z_i, z_j; \theta)$ and $(C_{*,N})_{1,j} = c(z_*, z_j; \theta)$. Note that each evaluation of the integrand requires $O(N^3)$ operations due to the matrix inversion. This computational issue is addressed by using a *subset of regressors* comprised of $N' < N$ samples (see Sec. 8.3.1 of Williams and Rasmussen, 2006, for full details)

$$f(\theta) = C_{*,N'}(C_{N',N}C_{N,N'} + \sigma^2 C_{N'})^{-1} C_{N',N} y \tag{4.6}$$



(a) $n = 50$ (b) $n = 100$ (c) $n = 200$

(d) $n = 50$ (e) $n = 100$ (f) $n = 200$

Figure 4.5 – Sampling standard deviation of Monte Carlo estimates for the posterior predictive mean $\mathbb{E}[y^\star|y]$, computed over 100 independent realisations.

Dataset and parameters. The goal is to estimate the inverse dynamics of a seven degrees-of-freedom SARCOS anthropomorphic robot arm. The task, as described in Williams and Rasmussen (2006, Sec. 8.3.1), is to map from a 21-dimensional input space (7 positions, 7 velocities, 7 accelerations) to the corresponding 7 joint torques using the hierarchical GP model mentioned above. Following Oates et al. (2017) we present results below on just one of the mappings, from the 21 input variables to the first of the seven torques. Similarly to the experiment of Oates et al. (2017) which investigates the sampling distribution of estimators, we take a random subset of $N = 1,000$ training points and a subset of regressors approximation with $N' = 100$. The inputs were translated and scaled to have mean zero and unit variance on the training set. The outputs were centered so as to have mean zero on the training set. Here $\sigma = 0.1$, $\alpha = \gamma = 25$, $\beta = \delta = 0.04$, so that each hyper-parameter θ_i has a prior mean of 1 and a prior standard deviation of 0.2.

For randomly selected test points z_* we estimated the sampling standard deviation of $\hat{y}^*$ over 100 independent realisations of the Monte Carlo sampling procedure. Along with the naive Monte Carlo estimate and the *control neighbor* estimate, we implemented the *control functionals* (CF) estimate from Oates et al. (2017) with default hyper-parameters $\alpha_1 = 0.1$, $\alpha_2 = 1$, the latter reflecting the fact that the training data were standardised. The estimator standard deviations were estimated in this way for 300 randomly selected test samples and the full results are shown in Fig. 4.5. Note that each test sample corresponds to a different integrand f and thus these results are quite objective, encompassing hundreds of different Monte Carlo integration problems.

4.A Proofs

Section 4.A.1 is concerned with the radius of k-NN estimate. Section 4.A.2 gathers the technical proofs of all the lemmas while Section 4.A.3 is concerned with the proofs of the different propositions.

4.A.1 k-Nearest Neighbor distance

Definition 4.19 (Nearest neighbors and distances)**.** *Given any point $x \in \mathbb{R}^d$ and any collection $X_1, \dots, X_n$ in $\mathbb{R}^d$, for $k = 1, \dots, n$ define $\hat{N}_{n,k}(x)$ as the k-nearest neighbor of x among $X_1, \dots, X_n$ and $\hat{\tau}_{n,k}(x)$ the associated distance, i.e., $\hat{\tau}_{n,k}(x) = \|\hat{N}_{n,k}(x) - x\|$. As before, when some ties are observed we use the lexicographic order.*

Lemma 4.20 (Upper bound of distance moments)**.** *Under (A1) and (A2), if $2k \leq n$, we have, for any $q \geq 1$,*

$$\forall x \in \mathcal{X}, \quad \mathbb{E}[\hat{\tau}_{n,k}(x)^q] \leq 2^{2q/d+1}\Gamma(q/d+1)(nV_d bc/k)^{-q/d}.$$

4.A.2 Proofs of Lemmas

Proof of Lemma 4.8 and Lemma 4.20

First, concerning the moments of 1-NN distance, the proof bears resemblance with the proof of Theorem 2.3 in Biau and Devroye (2015). Let $x \in \mathcal{X}$ and start with

$$\begin{aligned}
\mathbb{P}(|\hat{\tau}_n(x)| > t) &= \mathbb{P}\left(\min_{i=1,\dots,n} \|X_i - x\| > t\right) \\
&= [\mathbb{P}(\|X_1 - x\| > t)]^n \\
&= [1 - \mathbb{P}(B(x,t))]^n \\
&\leq \exp[-n\,\mathbb{P}(B(x,t))] \\
&\leq \exp(-nt^d V_d bc).
\end{aligned}$$

Then

$$\begin{aligned}\mathbb{E}\left[\hat{\tau}_n(x)^q\right] &= \int_0^\infty \mathbb{P}\left(|\hat{\tau}_n(x)| > t^{1/q}\right) \mathrm{d}t \\ &\leq \int_0^\infty \exp(-nt^{d/q} V_d bc)\, \mathrm{d}t \\ &= \left(nV_d bc\right)^{-q/d} (q/d) \int_0^\infty \exp(-u) u^{q/d-1}\, \mathrm{d}u \\ &= \left(nV_d bc\right)^{-q/d} \Gamma(q/d+1).\end{aligned}$$

Then, concerning the moments of k-NN distance, the proof is based on the one of Theorem 2.4 in Biau and Devroye (2015). Partition the set $X_1, \ldots, X_n$ into $2k$ sets of sizes $n_1, \ldots, n_{2k}$, with

$$\sum_{j=1}^{2k} n_j = n \quad \text{and} \quad \left\lfloor \frac{n}{2k} \right\rfloor \leq n_j \leq \left\lfloor \frac{n}{2k} \right\rfloor + 1.$$

Let $\hat{N}_{n_j}(x, j)$ be the nearest neighbor of x among all X_i's in the j-th group. Observe that, deterministically,

$$\|\hat{N}_{n,k}(x) - x\| \leq \frac{1}{k} \sum_{j=1}^{2k} \|\hat{N}_{n_j}(x, j) - x\|$$

and, similarly,

$$\|\hat{N}_{n,k}(x) - x\|^q \leq \frac{1}{k} \sum_{j=1}^{2k} \|\hat{N}_{n_j}(x, j) - x\|^q,$$

because at least k of these nearest neighbors have values that are at least $\|\hat{N}_{n,k}(x) - x\|$. This last inequality may be written as

$$\|\hat{N}_{n,k}(x) - x\|^q \leq \frac{1}{k} \sum_{j=1}^{2k} \hat{\tau}_{n_j}(x)^q.$$

Applying the previous upper bound for 1-NN moment gives

$$\begin{aligned}\mathbb{E}\left[\|\hat{N}_{n,k}(x) - x\|^q\right] &\leq \frac{1}{k} \sum_{j=1}^{2k} \left(n_j V_d bc\right)^{-q/d} \Gamma(q/d+1) \\ &= \frac{(V_d bc)^{-q/d} \Gamma(q/d+1)}{k} \sum_{j=1}^{2k} \left(\frac{1}{n_j}\right)^{q/d} \\ &= \frac{2^{q/d} (V_d bc)^{-q/d} \Gamma(q/d+1)}{k} \sum_{j=1}^{2k} \left(\frac{1}{2n_j}\right)^{q/d} \\ &\leq \frac{2^{q/d} (V_d bc)^{-q/d} \Gamma(q/d+1)}{k} \sum_{j=1}^{2k} \left(\frac{2k}{n}\right)^{q/d} \\ &= 2^{2q/d+1} \Gamma(q/d+1) (nV_d bc/k)^{-q/d}.\end{aligned}$$

Proof of Lemma 4.10

We have, by (A3),

$$|\hat{f}_n(x) - f(x)| = |\hat{f}(\hat{N}_n(x)) - f(x)| \leq L|\hat{N}_n(x) - x| = L\hat{\tau}_n(x).$$

Proof of Lemma 4.12

Given any collection $X_1, \ldots, X_n$ of distant points, if $j \neq i$, $\hat{f}_n^{(i)}$ and $\hat{g}_n$ are the same on $S_{n,j}$. It holds that

$$\hat{f}_n^{(i)}(x) - \hat{f}_n(x) = (\hat{f}_n^{(i)}(x) - \hat{f}_n(x))\mathbb{1}_{S_{n,i}}(x).$$

Now using that $\bar{f}_n$ and $\hat{f}_n^{(i)}$ are the same on $S_{n,i}$, it follows that

$$\hat{f}_n^{(i)}(x) - \hat{f}_n(x) = (\bar{f}_n(x) - \hat{f}_n(x))\mathbb{1}_{S_{n,i}}(x).$$

Taking the sum and using that $\sum_{i=1}^n \mathbb{1}_{S_{n,i}}(x) = 1$ gives

$$\sum_{i=1}^n \{\hat{f}_n^{(i)}(x) - \hat{f}_n(x)\} = (\bar{f}_n(x) - \hat{f}_n(x)),$$

and the result follows by integrating with respect to π. □

Proof of Lemma 4.14

First observe that

$$\mathbb{E}[\hat{d}_{n,j}] = \sum_{i:i\neq j} \mathbb{E}[V_{n,j}^{(i)}] = \mathbb{E}[\hat{c}_{n,j}]$$

and the first equality comes. Since $\sum_{j:j\neq i} \mathbb{E}[V_{n,j}^{(i)}] = 1$ and the variables $V_{n,j}^{(i)}$ for distinct i and j are identically distributed, we get $\mathbb{E}[V_{n,j}^{(i)}] = 1/(n-1)$ and thus $\mathbb{E}[\hat{d}_{n,j}] = 1$. □

Proof of Lemma 4.15

Because the Voronoi cells define a partition of $\mathbb{R}^d$, we have for any $x \in \mathbb{R}^d$,

$$\hat{f}_n^{(i)}(x) = \sum_{j:j\neq i} f(X_j)\mathbb{1}_{S_{n,j}^{(i)}}(x)$$

and in particular

$$\hat{f}_n^{(i)}(X_i) = \sum_{j:j\neq i} f(X_j)\mathbb{1}_{S_{n,j}^{(i)}}(X_i)$$

from which we deduce

$$\sum_{i=1}^n \hat{f}_n^{(i)}(X_i) = \sum_{j=1}^n f(X_j) \sum_{i:i\neq j} \mathbb{1}_{S_{n,j}^{(i)}}(X_i) = \sum_{j=1}^n f(X_j)\, \hat{d}_{n,j}.$$

Further, we have

$$\pi(\hat{f}_n) = \sum_{i=1}^n \pi\left(f(X_i)\mathbb{1}_{S_{n,i}}\right) = \sum_{i=1}^n f(X_i)\, \pi(S_{n,i}) = \sum_{i=1}^n f(X_i)\, V_{n,i}$$

and

$$\sum_{i=1}^n \pi(\hat{f}_n^{(i)}) = \sum_{i=1}^n \sum_{j:j\neq i} \pi\left(f(X_j)\mathbb{1}_{S_{n,j}^{(i)}}\right) = \sum_{j=1}^n f(X_j) \sum_{i:i\neq j} V_{n,j}^{(i)} = \sum_{j=1}^n f(X_j)\, \hat{c}_{n,j}.$$

4.A.3 Proofs of Propositions

Proof of Proposition 4.1

By conditioning on $\tilde{X}_1, \dots, \tilde{X}_N$, we obtain that

$$\begin{aligned}\mathbb{E}\left[|\hat{\pi}_n^{(CV)}(f) - \pi(f)|^2 | \tilde{X}_1, \dots, \tilde{X}_N\right] &= n^{-1} \operatorname{Var}[(f(X_1) - \hat{f}(X_1))|\tilde{X}_1, \dots, \tilde{X}_N] \\ &\leq n^{-1}\mathbb{E}[(f(X_1) - \hat{f}(X_1))^2|\tilde{X}] \\ &= n^{-1} \int (f - \hat{f})^2 d\pi.\end{aligned}$$

and taking the expectation with respect to the $\tilde{X}_1, \dots, \tilde{X}_N$ leads to the result. □

Proof of Proposition 4.16 Using Lemma 4.15, we find

$$\begin{aligned}\hat{\pi}_n^{(\mathrm{NN})}(f) &= \frac{1}{n}\sum_{i=1}^n [f(X_i) - \{\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n)\}] \\ &= \frac{1}{n}\sum_{i=1}^n f(X_i) - \frac{1}{n}\sum_{j=1}^n f(X_j)\,\hat{d}_{n,j} + \sum_{i=1}^n f(X_i)\,V_{n,i} \\ &= \frac{1}{n}\sum_{i=1}^n \left(1 - \hat{d}_{n,i} + nV_{n,i}\right) f(X_i)\end{aligned}$$

and, similarly,

$$\begin{aligned}\hat{\pi}_n^{(\mathrm{NN-loo})} &= \frac{1}{n}\sum_{i=1}^n [f(X_i) - \{\hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)})\}] \\ &= \frac{1}{n}\sum_{i=1}^n f(X_i) - \frac{1}{n}\sum_{j=1}^n f(X_j)\hat{d}_{n,j} + \frac{1}{n}\sum_{j=1}^n f(X_j)\hat{c}_{n,j} \\ &= \frac{1}{n}\sum_{i=1}^n \left(1 - \hat{d}_{n,i} + \hat{c}_{n,i}\right) f(X_i),\end{aligned}$$

as required. □

Proof of Proposition 4.17

Let $Y_{n,i} = \hat{f}_n^{(i)}(X_i) - \pi(\hat{f}_n^{(i)})$ and write

$$\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f) = \frac{1}{n}\sum_{i=1}^n \{Y_i - Y_{n,i}\}$$

with $Y_i = f(X_i) - \pi(f)$. Then write

$$\begin{aligned}n^2\mathbb{E}[(\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f))^2] &= \sum_{i=1}^n \mathbb{E}[\{Y_i - Y_{n,i}\}^2] + \sum_{i\neq j}\mathbb{E}[\{Y_i - Y_{n,i}\}\{Y_j - Y_{n,j}\}] \\ &= n\mathbb{E}[\{Y_1 - Y_{n,1}\}^2] + n(n-1)\mathbb{E}[\{Y_1 - Y_{n,1}\}\{Y_2 - Y_{n,2}\}]\end{aligned}$$

Now it is suitable to decompose $Y_{n,1}$ into two terms, one of which does not depend on X_2. We also use the fact that the Voronoi partition made with $(n-1)$ element is more detailed than the one constructed with $(n-2)$ points, *i.e.* $S_{n-1,i}^{(1)} \subset S_{n-2,i}^{(1,2)}$

for $i = 3, \ldots, n$. Define the map $\mathcal{N}^{(1,2)} : \mathbb{R}^d \to \mathbb{R}^d$ such that $\mathcal{N}^{(1,2)}(x)$ is the nearest neighbor to x among the sample $\{X_1, \ldots, X_n\}$ without X_1 and X_2. We write (using that $\mathcal{N}^{(1,2)}(x) = X_i$ whenever $x \in S^{(1)}_{n-1,i}$ for $i \geq 3$),

$$\begin{aligned}
\hat{f}_n^{(1,2)}(x) &= g(\mathcal{N}^{(1,2)}(x)) \\
&= f(\mathcal{N}^{(1,2)}(x)) \left(\sum_{i=2}^{n} \mathbb{1}_{S^{(1)}_{n-1,i}}(x) \right) \\
&= f(\mathcal{N}^{(1,2)}(x)) \mathbb{1}_{S^{(1)}_{n-1,2}}(x) + \sum_{i=3}^{n} f(X_i) \mathbb{1}_{S^{(1)}_{n-1,i}}(x) \\
&= (f(\mathcal{N}^{(1,2)}(x)) - f(X_2)) \mathbb{1}_{S^{(1)}_{n-1,2}}(x) + \sum_{i=2}^{n} f(X_i) \mathbb{1}_{S^{(1)}_{n-1,i}}(x).
\end{aligned}$$

It follows that

$$\hat{g}_n^{(1)}(x) = \hat{L}^{(1)}(x) + \hat{f}_n^{(1,2)}(x)$$

with $\hat{L}^{(1)}(x) = (f(X_2) - f(\mathcal{N}^{(1,2)}(x))) \mathbb{1}_{S^{(1)}_{n-1,2}}(x)$. Therefore,

$$Y_1 - Y_{n,1} = Y_1 - (\hat{L}^{(1)}(X_1) - \pi(\hat{L}^{(1)})) - (\hat{f}_n^{(1,2)}(X_1) - \pi(\hat{f}_n^{(1,2)})).$$

Denote

$$\begin{aligned}
A_1 &= Y_1, \\
A_2 &= Y_2, \\
B_1 &= \hat{L}^{(1)}(X_1) - \pi(\hat{L}^{(1)}), \\
B_2 &= \hat{L}^{(2)}(X_2) - \pi(\hat{L}^{(2)}), \\
C_1 &= \hat{f}_n^{(1,2)}(X_1) - \pi(\hat{f}_n^{(1,2)}), \\
C_2 &= \hat{f}_n^{(1,2)}(X_2) - \pi(\hat{f}_n^{(1,2)}),
\end{aligned}$$

where $\hat{L}^{(2)}(x) = (f(X_1) - f(\mathcal{N}^{(1,2)}(x))) \mathbb{1}_{S^{(2)}_{n-1,1}}(x)$. Then

$$\begin{aligned}
\mathbb{E}[\{Y_1 - Y_{n,1}\}\{Y_2 - Y_{n,2}\}] = {} & \mathbb{E}[A_1 A_2] + \mathbb{E}[A_1 B_2] + \mathbb{E}[A_1 C_2] \\
& + \mathbb{E}[B_1 A_2] + \mathbb{E}[B_1 B_2] + \mathbb{E}[B_1 C_2] \\
& + \mathbb{E}[C_1 A_2] + \mathbb{E}[C_1 B_2] + \mathbb{E}[C_1 C_2].
\end{aligned}$$

Since A_1 and A_2 are independent, $\mathbb{E}[A_1 A_2] = 0$. This also applies to $\mathbb{E}[A_1 C_2]$ and $\mathbb{E}[A_2 C_1]$. Considering $\mathbb{E}[A_1 B_2]$ gives

$$\begin{aligned}
\mathbb{E}[A_1 B_2] &= \mathbb{E}\Big[Y_1 (\hat{L}^{(2)}(X_2) - \pi(\hat{L}^{(2)})) \Big] \\
&= \mathbb{E}\Big[\mathbb{E}\Big[Y_1 (\hat{L}^{(2)}(X_2) - \pi(\hat{L}^{(2)})) \mid X_1, X_3, \ldots, X_n \Big] \Big] \\
&= \mathbb{E}\Big[Y_1 \mathbb{E}\Big[(\hat{L}^{(2)}(X_2) - \pi(\hat{L}^{(2)})) \mid X_1, X_3, \ldots, X_n \Big] \Big] = 0.
\end{aligned}$$

Due to similar reasoning, $\mathbb{E}[B_1 A_2] = 0$, $\mathbb{E}[B_1 C_2] = 0$ and $\mathbb{E}[C_1 B_2] = 0$. For $\mathbb{E}[C_1 C_2]$, we have

$$\begin{aligned}
\mathbb{E}[C_1 C_2] &= \mathbb{E}\Big[(\hat{f}_n^{(1,2)}(X_1) - \pi(\hat{f}_n^{(1,2)}))(\hat{f}_n^{(1,2)}(X_2) - \pi(\hat{f}_n^{(1,2)})) \Big] \\
&= \mathbb{E}\Big[\mathbb{E}\Big[(\hat{f}_n^{(1,2)}(X_1) - \pi(\hat{f}_n^{(1,2)}))(\hat{f}_n^{(1,2)}(X_2) - \pi(\hat{f}_n^{(1,2)})) \mid X_3, \ldots, X_n \Big] \Big] = 0.
\end{aligned}$$

Therefore, we get

$$\mathbb{E}[\{Y_1 - Y_{n,1}\}\{Y_2 - Y_{n,2}\}] = \mathbb{E}[\{\hat{L}^{(1)}(X_1) - \pi(\hat{L}^{(1)})\}\{\hat{L}^{(2)}(X_2) - \pi(\hat{L}^{(2)})\}].$$

The use of Cauchy-Schwarz inequality gives $\|(A-B)(C-D)\|_1 \leq \|A-B\|_2\|C-D\|_2$ and the fact that B and D are conditional expectation of A and C, respectively, leads to $\|(A-B)(C-D)\|_1 \leq \|A\|_2\|C\|_2 = \|A\|_2^2$. As a result,

$$\mathbb{E}[\{Y_1 - Y_{n,1}\}\{Y_2 - Y_{n,2}\}] \leq \mathbb{E}\Big[\hat{L}^{(1)}(X_1)^2\Big].$$

Using the Lipschitz property, we obtain

$$\begin{aligned}
|\hat{L}^{(1)}(x)| &= |f(X_2) - f(\mathcal{N}^{(1,2)}(x))|\mathbb{1}_{S^{(1)}_{n-1,2}}(x) \\
&= |f(\mathcal{N}^{(1)}(x)) - f(\mathcal{N}^{(1,2)}(x))|\mathbb{1}_{S^{(1)}_{n-1,2}}(x) \\
&\leq L\|\mathcal{N}^{(1)}(x) - \mathcal{N}^{(1,2)}(x)\|\mathbb{1}_{S^{(1)}_{n-1,2}}(x) \\
&\leq 2L\|x - \mathcal{N}^{(1,2)}(x)\|\mathbb{1}_{S^{(1)}_{n-1,2}}(x) \\
&= 2L\|x - \mathcal{N}^{(1,2)}(x)\|\mathbb{1}_{B(x,\hat{\tau}^{(1)}(x))}(X_2) \\
&\leq 2L\|x - \mathcal{N}^{(1,2)}(x)\|\mathbb{1}_{B(x,\hat{\tau}^{(1,2)}(x))}(X_2)
\end{aligned}$$

Hence

$$\mathbb{E}\Big[|\hat{L}^{(1)}(x)|^2 \mid X_3, \ldots, X_n\Big] \leq 4L^2\|x - \mathcal{N}^{(1,2)}(x)\|^2\pi\{B(x,\hat{\tau}^{(1,2)}(x))\}.$$

Moreover,

$$\begin{aligned}
\pi\{B(x,\hat{\tau}^{(1,2)}(x))\} &= \int_{B(x,\hat{\tau}^{(1,2)}(x))} \pi(z)\mathrm{d}z \\
&\leq U\int_{B(x,\hat{\tau}^{(1,2)}(x))\cap\mathcal{X}} \mathrm{d}z \\
&\leq U\hat{\tau}^{(1,2)}(x)^d V_d
\end{aligned}$$

Using that $\hat{\tau}^{(1,2)}(x) = \|x - \mathcal{N}^{(1,2)}(x)\|$, we obtain that

$$\mathbb{E}\Big[|\hat{L}^{(1)}(x)|^2 \mid X_3, \ldots, X_n\Big] \leq 4UV_dL^2\|x - \mathcal{N}^{(1,2)}(x)\|^{2+d}.$$

Applying to the term

$$\{Y_1 - Y_{n,1}\}^2 = \Big\{f(X_1) - \hat{f}_n^{(1)}(X_1) - (\pi(f) - \pi(\hat{f}_n^{(1)}))\Big\}^2$$

the same reasoning as above with $A = C = f(X_1) - \hat{f}_n^{(1)}(X_1)$ and $B = D = \pi(f) - \pi(\hat{f}_n^{(1)})$, we get

$$\mathbb{E}[\{Y_1 - Y_{n,1}\}^2] \leq \mathbb{E}\Big[\Big\{f(X_1) - \hat{f}_n^{(1)}(X_1)\Big\}^2\Big].$$

All this together gives

$$\begin{aligned}
\mathbb{E}\Big[|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)|^2\Big] &\leq n^{-1}\mathbb{E}\Big[|f(X_1) - \hat{f}^{(1)}(X_1)|^2\Big] + 4UV_dL^2\mathbb{E}\Big[\|X_1 - \mathcal{N}^{(1,2)}(X_1)\|^{2+d}\Big] \\
&= n^{-1}\mathbb{E}\Big[|f(X_1) - f(\mathcal{N}^{(1)}(X_1))|^2\Big] + 4UV_dL^2\mathbb{E}\Big[\|X_1 - \mathcal{N}^{(1,2)}(X_1)\|^{2+d}\Big] \\
&\leq L^2n^{-1}\mathbb{E}\Big[\|X_1 - \mathcal{N}^{(1)}(X_1)\|^2\Big] + 4UV_dL^2\mathbb{E}\Big[\|X_1 - \mathcal{N}^{(1,2)}(X_1)\|^{2+d}\Big] \\
&= L^2n^{-1}\mathbb{E}[\hat{\tau}_{n-1}(X_1)^2] + 4UV_dL^2\mathbb{E}[\hat{\tau}_{n-2}(X_1)^{2+d}].
\end{aligned}$$

Applying Lemma 4.8 to $\mathbb{E}[\hat{\tau}_{n-1}(X_1)^2]$ and to $\mathbb{E}[\hat{\tau}_{n-2}(X_1)^{2+d}]$, we get

$$\mathbb{E}[\hat{\tau}_{n-1}(X_1)^2] \leq \Big((n-1)V_d bc\Big)^{-2/d} \Gamma(2/d+1)$$

and

$$\mathbb{E}[\hat{\tau}_{n-2}(X_1)^{2+d}] \leq \Big((n-2)V_d bc\Big)^{-2/d-1} \Gamma(2/d+2).$$

Therefore,

$$\begin{aligned}\mathbb{E}\Big[|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)|^2\Big] \leq\ & L^2 n^{-1} \Big((n-1)V_d bc\Big)^{-2/d} \Gamma(2/d+1) \\ & + 4UV_d L^2 \Big((n-2)V_d bc\Big)^{-2/d-1} \Gamma(2/d+2).\end{aligned}$$

Rearranging the terms gives

$$\begin{aligned}&\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)\Big|^2\Big] \\ &\leq L^2 n^{-1} \Big((n-1)V_d bc\Big)^{-2/d} \Gamma(2/d+1) + 4UV_d L^2 \Big((n-2)V_d bc\Big)^{-2/d-1} \Gamma(2/d+2) \\ &\leq L^2 n^{-1} \Big((n-1)V_d bc\Big)^{-2/d} \Gamma(2/d+1) + 4(U/bc)L^2 \Big((n-2)V_d bc\Big)^{-2/d} (n-2)^{-1}\Gamma(2/d+2) \\ &\leq L^2 (n-2)^{-1} \Big((n-2)V_d bc\Big)^{-2/d} \Big[\Gamma(2/d+1) + 4(U/bc)\Gamma(2/d+2)\Big]\end{aligned}$$

Since $d \geq 1$, it holds that both $(2/d+1)$ and $(2/d+2)$ are in $[1,4]$. Using that $1 \leq \Gamma(x) \leq 6$ whenever $1 \leq x \leq 4$ we first get

$$\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)\Big|^2\Big] \leq L^2 (n-2)^{-1} \Big((n-2)V_d bc\Big)^{-2/d} 6(1 + 4(U/bc)).$$

Then, since $n \geq 4$ we have $n - 2 \geq n/2$ and obtain

$$\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)\Big|^2\Big] \leq 48L^2 n^{-1} \big(nV_d bc\big)^{-2/d} (U/bc)\Big[(bc/U) + 4\Big].$$

Using $(bc/U) \leq 1$ finally gives the stated bound. □

Proof of Proposition 4.18

The proof follows from combining Proposition 4.17 and the next inequality: for $n \geq 4$, we have

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^{n} \pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\right|^2\right] \leq 516L^2 (V_d bc)^{-2/d} n^{-1-2/d}, \tag{4.7}$$

whcih is established below. By Minkowski's inequality, we have

$$\begin{aligned}&\left(\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN})}(f) - \pi(f)\Big|^2\Big]\right)^{1/2} \\ &\qquad \leq \left(\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN})}(f) - \hat{\pi}_n^{(\mathrm{NN-loo})}(f)\Big|^2\Big]\right)^{1/2} + \left(\mathbb{E}\Big[\Big|\hat{\pi}_n^{(\mathrm{NN-loo})}(f) - \pi(f)\Big|^2\Big]\right)^{1/2}.\end{aligned}$$

Conclude by using the bounds of Proposition 4.17 and (4.7) along with $\sqrt{240} \leq 16$; $\sqrt{516} \leq 23$.

Proof of (4.7). Using the fact that, for $\hat{f}_n^{(i)}$ and $\hat{f}_n(x)$ coincides outside $S_{n,i}$ and that $\hat{f}_n(x) = g(X_i)$ for $x \in S_{n,i}^\circ$, we have

$$\begin{aligned}\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n) &= \pi(\hat{f}_n^{(i)} - \hat{f}_n) \\ &= \pi((\hat{f}_n^{(i)} - \hat{f}_n)\mathbb{1}_{S_{n,i}^\circ}) \\ &= \pi((\hat{f}_n^{(i)} - f(X_i))\mathbb{1}_{S_{n,i}^\circ}).\end{aligned}$$

Denote $R_n = \left|\frac{1}{n}\sum_{i=1}^n \left[\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\right]\right|$. Then using the triangle inequality and the definition of $\hat{f}$ gives

$$\begin{aligned}R_n &\leq \frac{1}{n}\sum_{i=1}^n \left|\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\right| \\ &= \frac{1}{n}\sum_{i=1}^n \left|\pi((\hat{f}_n^{(i)} - f(X_i))\mathbb{1}_{S_{n,i}^\circ})\right| \\ &\leq \frac{1}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \left|\hat{f}_n^{(i)}(x) - f(X_i)\right| \mathrm{d}\pi(x) \\ &= \frac{1}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \left|f(\hat{N}_n^{(i)}(x)) - f(X_i)\right| \mathrm{d}\pi(x).\end{aligned}$$

And because g is L-Lipschitz, one has

$$\begin{aligned}R_n &\leq \frac{L}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \|\hat{N}_n^{(i)}(x) - X_i\| \mathrm{d}\pi(x) \\ &\leq \frac{L}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \Big(\|\hat{N}_n^{(i)}(x) - x\| + \|x - X_i\|\Big) \mathrm{d}\pi(x) \\ &= \frac{L}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \Big[\hat{\tau}_n^{(i)}(x) + \hat{\tau}_n(x)\Big] \mathrm{d}\pi(x).\end{aligned}$$

Concerning R_n^2, one can use Jensen's inequality to obtain

$$\begin{aligned}R_n^2 &= \left|\frac{1}{n}\sum_{i=1}^n \left[\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\right]\right|^2 \\ &\leq \frac{1}{n}\sum_{i=1}^n \left|\pi(\hat{f}_n^{(i)}) - \pi(\hat{f}_n)\right|^2 \\ &\leq \frac{L^2}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \Big(\|\hat{N}_n^{(i)}(x) - x\| + \|x - X_i\|\Big)^2 \mathrm{d}\pi(x) \\ &\leq \frac{2L^2}{n}\sum_{i=1}^n \int_{S_{n,i}^\circ} \Big(\|\hat{N}_n^{(i)}(x) - x\|^2 + \|x - X_i\|^2\Big) \mathrm{d}\pi(x).\end{aligned}$$

For $x \in S_{n,i}^{\circ}$, the nearest neighbor in $\{X_1, \ldots, X_n\}$ is X_i. Hence, for $x \in S_{n,i}^{\circ}$,

$$\hat{\tau}_n^{(i)}(x) = \hat{\tau}_{n,2}(x)$$

is the distance to the second nearest neighbor in $\{X_1, \ldots, X_n\}$. We get

$$\begin{aligned} R_n &\leq \frac{L}{n} \sum_{i=1}^{n} \int_{S_{n,i}} \Big[\hat{\tau}_{n,2}(x) + \hat{\tau}_n(x)\Big] \mathrm{d}\pi(x) \\ &= \frac{L}{n} \int_{\mathcal{X}} \Big[\hat{\tau}_{n,2}(x) + \hat{\tau}_n(x)\Big] \mathrm{d}\pi(x) \end{aligned}$$

and

$$\begin{aligned} R_n^2 &\leq \frac{2L^2}{n} \sum_{i=1}^{n} \int_{S_{n,i}} \Big[\hat{\tau}_{n,2}(x)^2 + \hat{\tau}_n(x)^2\Big] \mathrm{d}\pi(x) \\ &= \frac{2L^2}{n} \int_{\mathcal{X}} \Big[\hat{\tau}_{n,2}(x)^2 + \hat{\tau}_n(x)^2\Big] \mathrm{d}\pi(x). \end{aligned}$$

Consequently, by Lemma 4.8,

$$\begin{aligned} \mathbb{E}[R_n] &\leq \frac{L}{n} \int_{\mathcal{X}} \mathbb{E}\Big[\hat{\tau}_{n,2}(x) + \hat{\tau}_n(x)\Big] \mathrm{d}\pi(x) \\ &\leq \frac{L}{n} \left(\sup_{x \in \mathcal{X}} \mathbb{E}\Big[\hat{\tau}_{n,2}(x)\Big] + \sup_{x \in \mathcal{X}} \mathbb{E}\Big[\hat{\tau}_n(x)\Big] \right) \\ &\leq \frac{L}{n} \Big(2^{2/d+1} \Gamma(1/d+1)(nV_d bc/2)^{-1/d} + (nV_d bc)^{-1/d} \Gamma(1/d+1) \Big) \\ &= (2^{3/d+1} + 1) L (V_d bc)^{-1/d} \Gamma(1/d+1) n^{-1-1/d} \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}[R_n^2] &\leq \frac{2L^2}{n} \int_{\mathcal{X}} \mathbb{E}\Big[\hat{\tau}_{n,2}(x)^2 + \hat{\tau}_n(x)^2\Big] \mathrm{d}\pi(x) \\ &\leq \frac{2L^2}{n} \left(\sup_{x \in \mathcal{X}} \mathbb{E}\Big[\hat{\tau}_{n,2}(x)^2\Big] + \sup_{x \in \mathcal{X}} \mathbb{E}\Big[\hat{\tau}_n(x)^2\Big] \right) \\ &\leq \frac{2L^2}{n} \Big(2^{4/d+1} \Gamma(2/d+1)(nV_d bc/2)^{-2/d} + (nV_d bc)^{-2/d} \Gamma(2/d+1) \Big) \\ &= 2(2^{6/d+1} + 1) L^2 (V_d bc)^{-2/d} \Gamma(2/d+1) n^{-1-2/d} \end{aligned}$$

Now use that $1 \leq \Gamma(x) \leq 2$ for $1 \leq x \leq 3$ and $(2^{6/d+1} + 1) \leq 2^7 + 1 = 129$ to obtain the stated bound.

Part III

Stochastic Approximation: Conditioning, Adaptive Sampling

"If people do not believe that mathematics is simple,
it is only because they do not realize how complicated life is."

(**John von Neumann**, 1st meeting of the Association for Computing Machinery, 1947)

Chapter 5

Asymptotic Analysis of Conditioned Stochastic Gradient Descent

Contents

In this chapter, we investigate a general class of stochastic gradient descent (SGD) algorithms, called *conditioned* SGD, based on a preconditioning of the gradient direction. Using a discrete-time approach with martingale tools, we establish the weak convergence of the rescaled sequence of iterates for a broad class of conditioning matrices including stochastic first-order and second-order methods. Almost sure convergence results, which may be of independent interest, are also presented. When the conditioning matrix is an estimate of the inverse Hessian, the algorithm is asymptotically optimal. For the sake of completeness, we provide a practical procedure to achieve this minimum variance.

5.1 Introduction

Consider some classical unconstrained optimization problem of the following form: $\min_{\theta\in\mathbb{R}^d}\{F(\theta) = \mathbb{E}_\xi[f(\theta,\xi)]\}$, where f is a loss function and ξ is a random variable. This key methodological problem, known under the name of *stochastic programming* (Shapiro et al., 2014), includes many flagship machine learning applications such as *empirical risk minimization* (Bottou et al., 2018), *adaptive importance sampling* (Delyon and Portier, 2018) and *reinforcement learning* (Sutton and Barto, 2018). When F is differentiable, a common appproach is to rely on first-order methods. However, in many scenarios and particularly in large-scale learning, the gradient of F may be hard to evaluate or even intractable. Instead, a random unbiased estimate of the gradient is available at a cheap computing cost and the state-of-the-art algorithm, *stochastic gradient descent* (SGD), just moves along this estimate at each iteration. It is an iterative algorithm, simple and computationally fast, but its convergence towards the optimum is generally slow. *Conditioned* SGD, which consists in multiplying the gradient estimate by some conditioning matrix at each iteration, can lead to better performance as shown in several recent studies ranging from natural gradient (Amari, 1998; Kakade, 2002) and stochastic second-order methods with quasi-Newton (Byrd et al., 2016) and (L)-BFGS methods (Liu and Nocedal, 1989) to diagonal scalings and adaptive methods such as AdaGrad (Duchi et al., 2011), RMSProp (Tieleman et al., 2012), Adam (Kingma and Ba, 2014) and AMSGrad (Reddi et al., 2018).

Conditioned SGD generalizes *standard SGD* by adding a conditioning step to refine the descent direction. Starting from $\theta_0 \in \mathbb{R}^d$, the algorithm of interest is defined by the following iteration

$$\theta_{k+1} = \theta_k - \gamma_{k+1} C_k g(\theta_k, \xi_{k+1}), \qquad k \geq 0,$$

where $g(\theta_k, \xi_{k+1})$ is some unbiased gradient valued in $\mathbb{R}^d$, $C_k \in \mathbb{R}^{d \times d}$ is called *conditioning matrix* and $(\gamma_k)_{k \geq 1}$ is a decreasing learning rate sequence. Interestingly, the optimal choice according to the asymptotic variance is the inverse of the Hessian matrix at optimal point, i.e., $C_k = \nabla^2 F(\theta^\star)^{-1}$; see (Benveniste et al., 2012, Chapter 3) or Section 5.2.3 in this paper. With this matrix, the rate of convergence remains the same and only the asymptotic variance can be reduced; e.g., Agarwal et al. (2009). An important question, which is still open to the best of our knowledge, is whether the optimal variance can be achieved by such an algorithm for non-convex objective F. We show that the answer is positive under mild conditions on the matrix C_k.

Related work. Seminal works around standard SGD ($C_k = I_d$) were initiated by Robbins and Monro (1951) and Kiefer et al. (1952). Since then, a large literature known as *stochastic approximation*, has developed. The almost sure convergence is studied in Robbins and Siegmund (1971) and Bertsekas and Tsitsiklis (2000); rates of convergence are investigated in Kushner and Huang (1979) and Pelletier (1998a); non-asymptotic bounds are given in Moulines and Bach (2011). The asymptotic normality can be obtained using two different approaches: a diffusion-based method is employed in Pelletier (1998b) and Benaïm (1999) whereas martingale tools are used in Sacks (1958) and Kushner and Clark (1978). We refer to Nevelson and Khas'minskiĭ (1976); Delyon (1996); Benveniste et al. (2012); Duflo (2013) for general textbooks on *stochastic approximation*.

The aforementioned results do not apply directly to *conditioned* SGD because of the presence of the matrix sequence $(C_k)_{k \geq 0}$ involving an additional source of randomness in the algorithm. Seminal papers dealing with the weak convergence of *conditioned* SGD are Venter (1967) and Fabian (1968). Within a restrictive framework (univariate case $d = 1$ and strong assumptions on the function F), their results are encouraging because the limiting variance of the procedure is shown to be smaller than the limiting variance of standard SGD. Venter's and Fabian's results have then been extended to more general situations (Fabian, 1973; Nevelson and Khas'minskiĭ, 1976; Wei, 1987). In Wei (1987), the framework is still restrictive not only because the random errors are assumed to be independent and identically distributed but also because the objective F must satisfy their assumption (4.10) which hardly extends to objectives other than quadratic.

More recently, Bercu et al. (2020) have obtained the asymptotic normality as well as the efficiency of certain *conditioned* SGD estimates in the particular case of *logistic regression*. The previous approach has been generalized not long ago in Boyer and Godichon-Baggioni (2020) where the use of the Woodbury matrix identity is promoted to compute the Hessian inverse in the online setting. Several theoretical results, including the weak convergence of *conditioned* SGD, are obtained for convex objective functions. An alternative to *conditioning*, called *averaging*, developed by Polyak (1990) and Polyak and Juditsky (1992), allows to recover the same asymptotic variance as *conditioned SGD*. When dealing with convex objectives, the theory behind this averaging technique is a well-studied topic (Moulines and Bach, 2011; Gadat and Panloup, 2017; Dieuleveut et al., 2020; Zhu et al., 2021). However, it is inevitably associated with a

large bias caused by poor initialization. Furthermore, *conditioned SGD* methods proved to be the current state-of-the-art for training machine learning models (Zhang, 2004; LeCun et al., 2012) and are implemented in widely used programming tools (Pedregosa et al., 2011; Abadi et al., 2016).

Contributions. The main result of this chapter deals with the weak convergence of the rescaled sequence of iterates. Interestingly, our asymptotic normality result consists of the following continuity property: whenever the matrix sequence $(C_k)_{k\geq 0}$ converges to a matrix C and the iterates $(\theta_k)_{k\geq 0}$ converges to a minimizer $\theta^\star$, the algorithm behaves in the same way as an oracle version in which C would be used instead of C_k. We stress that contrary to Boyer and Godichon-Baggioni (2020), no convexity assumption is needed on the objective function and no rate of convergence is required on the sequence $(C_k)_{k\geq 0}$. This is important because, in most cases, deriving a convergence rate on $(C_k)_{k\geq 0}$ requires a specific convergence rate on the iterates $(\theta_k)_{k\geq 0}$ which, in general, is unknown at this stage of the analysis. Another result of independent interest dealing with the almost sure convergence of the gradients is also provided. Finally, for the sake of completeness, we present practical ways to compute the *conditioning* matrix C_k and show that the resulting procedure satisfies the high-level conditions of our main Theorem. This yields a feasible algorithm which achieves minimum variance.

To obtain these results, instead of approximating the rescaled sequence of iterates by a continuous diffusion (as for instance in Pelletier (1998b)), we rely on a discrete-time approach where the recursion scheme is directly analyzed (as for instance in Delyon (1996)). More precisely, the sequence of iterates is studied with the help of an auxiliary linear algorithm whose limiting distribution can be deduced from the central limit theorem for martingale increments (Hall and Heyde, 1980). The limiting variance is derived from a discrete time matrix-valued dynamical system algorithm. It corresponds to the solution of a Lyapunov equation involving the matrix C. It allows a special choice for C which guarantees an optimal variance. Finally, in order to examine the remaining part, a particular recursion is identified. By studying it on a particular event, we show that this remaining part is negligible.

Outline. Section 5.2 introduces the framework of standard SGD with asymptotic results. Section 5.3 is dedicated to *conditioned* SGD: it first presents popular optimization methods that fall in the considered framework and then presents our main results, namely the weak convergence and asymptotic optimality. Section 5.4 gathers practical tools to meet the developed theoretical framework and Section 5.5 concludes the chapter with a discussion of avenues for further research.

5.2 Mathematical background

In this section, the mathematical background of stochastic gradient descent (SGD) methods is presented and illustrated with the help of some examples. Then, to motivate the use of *conditioning* matrices, we present a known result from Pelletier (1998b) about the weak convergence of SGD given the almost sure convergence of the iterates.

5.2.1 Problem setup

Consider the problem of finding a minimizer $\theta^\star \in \mathbb{R}^d$ of a function $F : \mathbb{R}^d \to \mathbb{R}$, that is,

$$\theta^\star \in \arg\min_{\theta\in\mathbb{R}^d} F(\theta).$$

In many scenarios and particularly in large scale learning, the gradient of F cannot be fully computed and only a stochastic unbiased version of it is available. The SGD algorithm moves the iterate along this direction. To increase the efficiency, the random generators used to derive the unbiased gradients might evolve during the algorithm, *e.g.*, using the past iterations. To analyse such algorithms, we consider the following probabilistic setting.

Definition 5.1. *A stochastic algorithm is a sequence $(\theta_k)_{k\geq 0}$ of random variables defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and valued in $\mathbb{R}^d$. Define $(\mathcal{F}_k)_{k\geq 0}$ as the natural σ-field associated to the stochastic algorithm $(\theta_k)_{k\geq 0}$, i.e., $\mathcal{F}_k = \sigma(\theta_0, \theta_1, \ldots, \theta_k)$, $k \geq 0$. A policy is a sequence of random probability measures $(P_k)_{k\geq 0}$, each defined on a measurable space $(S, \mathcal{S})$ that are adapted to $\mathcal{F}_k$.*

Given a *policy* $(P_k)_{k\geq 0}$ and a *learning rates* sequence $(\gamma_k)_{k\geq 1}$ of positive numbers, the SGD algorithm (Robbins and Monro, 1951) is defined by the update rule

$$\theta_{k+1} = \theta_k - \gamma_{k+1} g(\theta_k, \xi_{k+1}) \quad \text{with} \quad \xi_{k+1} \sim P_k, \tag{5.1}$$

where $g : \mathbb{R}^d \times S \to \mathbb{R}^d$ is called the gradient generator. Hence the policy $(P_k)_{k\geq 0}$ is used at each iteration to produce random gradients through the function g. Those gradients are assumed to be unbiased.

Assumption 5.2 (Unbiased gradient)**.** *The gradient generator $g : \mathbb{R}^d \times S \to \mathbb{R}^d$ is such that for all $\theta \in \mathbb{R}^d$, $g(\theta, \cdot)$ is measurable, and we have:*

$$\forall k \geq 0, \quad \mathbb{E}\left[g(\theta_k, \xi_{k+1}) | \mathcal{F}_k\right] = \nabla F(\theta_k).$$

We emphasize three important examples covered by the developed approach. In each case, explicit ways to generate the stochastic gradient are provided.

Example 1. *(Empirical Risk Minimization)* Given some observed data $z_1, \ldots, z_n \in \mathbb{R}^p$ and a differentiable loss function $\ell : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$, the objective function F approximates the true expected risk $\mathbb{E}_z[\ell(\theta, z)]$ using its empirical counterpart $F(\theta) = n^{-1} \sum_{i=1}^n \ell(\theta, z_i)$. Classically, the gradient estimates at θ_k are given by the policy

$$g(\theta_k, \xi_{k+1}) = \nabla_\theta \ell(\theta_k, \xi_{k+1}) \quad \text{with} \quad \xi_{k+1} \sim \sum_{i=1}^n \delta_{z_i}/n.$$

Another one, more subtle, referred to as mini-batching (Gower et al., 2019), consists in generating uniformly a set of n_k samples $(z_1, \ldots, z_{n_k})$ and computing the gradient as the average $n_k^{-1} \sum_{j=1}^{n_k} \nabla_\theta \ell(\theta_k, z_j)$. Note that interestingly, we allow changes of the minibatch size throughout the algorithm. Our framework also includes adaptive non-uniform sampling (Papa et al., 2015) and survey sampling (Clémençon et al., 2019), which use $P_k = \sum_{i=1}^n w_i^{(k)} \delta_{z_i}$ with $\mathcal{F}_k$-adapted weights satisfying $\sum_{i=1}^n w_i^{(k)} = 1$ for each $k \geq 0$.

Example 2. *(Adaptive importance sampling)* Given a target function f, which might result from the likelihood of some data, and a parametric family of sampler $\{q_\theta : \theta \in \Theta\}$, the objective function is $F(\theta) = -\int \log(q_\theta(y)) f(y) dy$. Other losses can be considered

and we refer to Delyon and Portier (2018) for some details and further references about adaptive importance sampling. A common choice in practice for the policy is given by

$$g(\theta_k, \xi_{k+1}) = -\nabla_\theta \log(q_{\theta_k}(\xi_{k+1})) \frac{f(\xi_{k+1})}{q_{\theta_k}(\xi_{k+1})}, \quad \xi_{k+1} \sim q_{\theta_k}.$$

Example 3. *(Policy-gradient methods)* In reinforcement learning (Sutton and Barto, 2018), the goal of the agent is to find the best action-selection policy to maximize the expected reward. Policy-gradient methods (Baxter and Bartlett, 2001; Williams, 1992) use a parameterized policy $\{\pi_\theta : \theta \in \Theta\}$ to optimize an expected reward function F given by $F(\theta) = \mathbb{E}_{\xi\sim\pi_\theta}[\mathcal{R}(\xi)]$ where ξ is a trajectory including nature states and selected actions. Using the policy gradient theorem, one has $\nabla F(\theta) = \mathbb{E}_{\xi\sim\pi_\theta}\left[\mathcal{R}(\xi)\nabla_\theta \log \pi_\theta(\xi)\right]$, leading to the REINFORCE algorithm (Williams, 1992) given by

$$g(\theta_k, \xi_{k+1}) = \mathcal{R}(\xi_{k+1})\nabla_\theta \log \pi_{\theta_k}(\xi_{k+1}), \quad \xi_{k+1} \sim \pi_{\theta_k}.$$

5.2.2 Weak convergence of SGD

This section is related to the weak convergence property of the normalized sequence of iterates $(\theta_k - \theta^\star)/\sqrt{\gamma_k}$. The working assumptions include the almost sure convergence of the sequence of iterates $(\theta_k)_{k\geq 0}$ towards a stationary point $\theta^\star$. Note that, given Assumptions 5.2 and 5.3, there exist many criteria on the objective function that give such almost sure convergence. For these results, we refer to Bertsekas and Tsitsiklis (2000); Benveniste et al. (2012); Duflo (2013). In addition to this high-level assumption of almost sure convergence, we require the following classical assumptions. Let $\mathcal{S}_d^{++}(\mathbb{R})$ denote the space of real symmetric positive definite matrices and define for all $k \geq 0$,

$$w_{k+1} = \nabla F(\theta_k) - g(\theta_k, \xi_{k+1})$$
$$\Gamma_k = \mathbb{E}\left[w_{k+1} w_{k+1}^\top | \mathcal{F}_k\right].$$

Assumption 5.3 (Learning rates)**.** *The sequence of step-size is* $\gamma_k = \alpha k^{-\beta}$ *with* $\beta \in (1/2, 1]$.

Assumption 5.4 (Hessian)**.** *The Hessian matrix at stationary point is positive definite, i.e.,* $H = \nabla^2 F(\theta^\star) \in \mathcal{S}_d^{++}(\mathbb{R})$ *and the mapping* $\theta \mapsto \nabla^2 F(\theta)$ *is continuous at* $\theta^\star$.

Assumption 5.5 (Covariance matrix)**.** *There exists* $\Gamma \in \mathcal{S}_d^{++}(\mathbb{R})$ *such that almost surely* $\Gamma_k \overset{k\to+\infty}{\longrightarrow} \Gamma$.

Assumption 5.6 (Lyapunov bound)**.** *There exist* $\delta, \varepsilon > 0$ *such that:*

$$\sup_{k\geq 0} \mathbb{E}[\|w_{k+1}\|_2^{2+\delta} | \mathcal{F}_k] \mathbb{1}_{\{\|\theta_k - \theta^\star\| \leq \varepsilon\}} < \infty \quad a.s.$$

Assumption 5.5 is needed to identify the limiting distribution while Assumption 5.6 is a stability condition, often referred to as the Lyapunov condition, required for tightness. The following result can be either derived from (Pelletier, 1998b, Theorem 1) or as a direct corollary of our main result, Theorem 5.10, given in Section 5.3.2.

Theorem 5.7 (Weak convergence of SGD). *Let $(\theta_k)_{k\geq 0}$ be obtained by the SGD rule (5.1). Suppose that Assumptions 5.2, 5.3, 5.4, 5.5, 5.6 are fulfilled and that $\theta_k \to \theta^\star$ almost surely. If moreover, $(H-\zeta I)$ is positive definite with $\zeta = \mathbb{1}_{\{\beta=1\}}/2\alpha$, it holds that*

$$\frac{1}{\sqrt{\gamma_k}}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0,\Sigma), \qquad \text{as } k \to \infty,$$

where the covariance matrix Σ satisfies the following Lyapunov equation

$$(H-\zeta I_d)\Sigma + \Sigma(H-\zeta I_d)^\top = \Gamma.$$

Several remarks are to be explored. Since Γ and $(H-\zeta I)$ are positive definite matrices, there exists a unique solution Σ to the Lyapunov equation $(H-\zeta I_d)\Sigma+\Sigma(H-\zeta I_d)^\top = \Gamma$ given by $\Sigma = \int_0^{+\infty} \exp[-t(H-\zeta I_d)]\Gamma \exp[-t(H-\zeta I_d)^\top]\mathrm{d}t$. Second, the previous result can be expressed as $k^{\beta/2}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0,\alpha\Sigma)$. Hence, the fastest rate of convergence is obtained when $\beta = 1$ for which we recover the classical $1/\sqrt{k}$-rate of a Monte-Carlo estimate. In this case, the coefficient α should be chosen large enough to ensure the convergence through the condition $H - I_d/(2\alpha) \succ 0$, but also such that the covariance matrix $\alpha\Sigma$ is small. The choice of α is discussed in the next section and should be replaced with a matrix gain.

5.2.3 Minimum variance with deterministic conditioning

To motivate the use of *conditioning* matrices in SGD, we raise the question of variance optimality when γ_k decreases as $1/k$, so the rate of convergence in Theorem 5.7 is optimal, and the scalar gain α is replaced by a *conditioning* matrix $C \in \mathcal{S}_d^{++}(\mathbb{R})$. That is, we consider the iteration scheme, for $k \geq 1$,

$$\theta_{k+1} = \theta_k - \left(\frac{C}{k+1}\right) g(\theta_k, \xi_{k+1}). \tag{5.2}$$

As a corollary of Theorem 5.10 (given below) or inferring from the results in Pelletier (1998b), we can derive the following result. Define $\mathcal{C}_H$ as the set of real symmetric positive definite matrices $C \in \mathcal{S}_d^{++}(\mathbb{R})$ such that all the eigenvalues of the matrix $CH - (I/2)$ are positive.

Proposition 5.8. *Let $(\theta_k)_{k\geq 0}$ be obtained by (5.2) with $C \in \mathcal{C}_H$. Suppose that Assumptions 5.2, 5.4, 5.5, 5.6 are fulfilled and that $\theta_k \to \theta^\star$ almost surely. Then we have*

$$\sqrt{k}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0,\Sigma_C), \qquad \text{as } k \to \infty,$$

where Σ_C satisfies:

$$(CH - I_d/2)\Sigma_C + \Sigma_C(CH - I_d/2)^\top = C\Gamma C^\top.$$

The best *conditioning* matrix C that could be chosen regarding the asymptotic variance is specified in the next proposition whose proof is given in Section 5.A.2.

Proposition 5.9 (Optimal choice). *The choice $C^\star = H^{-1}$ is optimal in the sense that $\Sigma_{C^\star} \preceq \Sigma_C$ for all $C \in \mathcal{C}_H$. Moreover, we have $\Sigma_{C^\star} = H^{-1}\Gamma H^{-1}$.*

In deterministic gradient descent, it is well-known that the rate of convergence is improved when the gradient is multiplied by the inverse of the Hessian matrix, referred to as the Newton algorithm, whose convergence rate is quadratic, instead of linear for gradient descent. Due to Proposition 5.9 where we see that the smallest limiting variance is nonzero, a faster rate of convergence cannot be expected with *conditioned* SGD. However, an improvement in the limiting variance is still possible.

5.3 The asymptotics of conditioned stochastic gradient descent

This Section first presents practical optimization schemes that fall in the framework of *conditioned* SGD. Then it contains our main results, namely the weak convergence and asymptotic optimality. Another result of independent interest dealing with the almost sure convergence of the gradients and the iterates is also provided.

5.3.1 Framework and Examples

We introduce the general framework of *conditioned* SGD as an extension of the standard SGD presented in Section 6.2. It is defined by the following update rule, for $k \geq 0$,

$$\theta_{k+1} = \theta_k - \gamma_{k+1} C_k g(\theta_k, \xi_{k+1}), \tag{5.3}$$

where the matrix $C_k \in \mathcal{S}_d^{++}(\mathbb{R})$, the *conditioning matrix*, is a $\mathcal{F}_k$-measurable real symmetric positive definite matrix so that the search direction always points to a descent direction. In convex optimization, inverse of the Hessian is a popular choice but *(1)* it may be hard to compute, *(2)* it is not always positive definite and *(3)* it may increase the noise of SGD especially when the Hessian is ill-conditioned.

Quasi-Newton. These methods build approximations of the Hessian $C_k \approx \nabla^2 f(\theta_k)^{-1}$ with gradient-only information, and are applicable for convex and nonconvex problems. For scalability issue, variants with limited memory are the most used in practice. Following Newton's method idea with the secant equation, the update rule is based on pairs (s_k, y_k) tracking the differences of iterates and stochastic gradients, *i.e.*, $s_k = \theta_{k+1} - \theta_k$ and $y_k = g(\theta_{k+1}, \xi_{k+1}) - g(\theta_k, \xi_{k+1})$. Let $\rho_k = 1/(s_k^\top y_k)$ then the Hessian updates are

$$C_{k+1} = (I - \rho_k y_k s_k^\top)^\top C_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top.$$

In the deterministic setting, the BFGS update formula above is well-defined as long as $s_k^\top y_k > 0$. Such condition preserves positive definite approximations and may be obtained in the stochastic setting by replacing the Hessian matrix with a Gauss-Newton approximation and using regularization.

Adaptive methods and Diagonal scalings. These methods adapt locally to the structure of the optimization problem by setting C_k as a function of past stochastic gradients. General adaptive methods differ in the construction of the *conditioning* matrix and whether or not they add a momentum term. Using different representations such as dense or sparse conditioners also modify the properties of the underlying algorithm. For instance, the optimizers Adam and RMSProp maintain an exponential moving average of past stochastic gradients with a factor $\tau \in (0, 1)$ but fail to guarantee $C_{k+1} \preceq C_k$. Such behaviour can lead to large fluctuations and prevent convergence of the iterates. Instead, AdaGrad and AMSGrad ensure the monotonicity $C_{k+1} \preceq C_k$.

Denote by $g_k = g(\theta_k, \xi_{k+1})$ a gradient estimate and $m \in [0, 1)$ a momentum parameter. General adaptive gradient methods are defined by

$$\theta_{k+1} = \theta_k - \gamma_{k+1} C_k \hat{g}_k, \quad \hat{g}_k = m\hat{g}_{k-1} + (1-m)g_k.$$

Different optimizers are summarized in Table 5.1 below. They all rely on a gradient matrix G_k which accumulates the information of stochastic gradients. The *conditioning* matrix is equal to $C_k = G_k^{-1/2}$ except for AMSGrad which uses $C_k = \max\{C_{k-1}; G_k^{-1/2}\}$. Starting from $G_0 = \delta I$ with $\delta > 0$, G_{k+1} is updated either in a dense or sparse (diagonal) manner or using an exponential moving average.

Optimizer	**Gradient matrix $\mathbf{G_{k+1}}$**	**m**
AdaFull	$G_k + g_k g_k^\top$	0
AdaNorm	$G_k + \|g_k\|_2^2$	0
AdaDiag	$G_k + diag(g_k g_k^\top)$	0
RMSProp	$\tau G_k + (1-\tau) diag(g_k g_k^\top)$	0
Adam	$[\tau G_k + (1-\tau) diag(g_k g_k^\top)]/(1-\tau^k)$	m
AMSGrad	$[\tau G_k + (1-\tau) diag(g_k g_k^\top)]/(1-\tau^k)$	m

Table 5.1 – Adaptive Gradient Methods.

A common assumption made in the literature of adaptive methods is that *conditioning* matrices are well-behaved in the sense that their eigenvalues are bounded in a fixed interval. This property is easy to check for diagonal matrices and can always be implemented in practice using projection.

5.3.2 Main result

Similarly to the weak convergence of the SGD iterates, it is interesting to search for an appropriate rescaled process to obtain some convergence rate and asymptotic normality results. In fact the only additional assumption needed, compared to SGD, is the almost sure convergence of the sequence $(C_k)_{k\geq 0}$. This makes Theorem 5.7 a particular case of the following Theorem which is the main result of the paper.

Theorem 5.10 (Weak convergence of conditioned SGD)**.** *Let $(\theta_k)_{k\geq 0}$ be obtained by conditioned SGD (6.3). Suppose that Assumptions 5.2, 5.3, 5.4, 5.5, 5.6 are fulfilled and that $\theta_k \to \theta^\star$ almost surely. If moreover, $C_k \to C \in \mathcal{S}_d^{++}(\mathbb{R})$ almost surely and all the eigenvalues of $(CH - \zeta I)$ are positive with $\zeta = \mathbb{1}_{\{\beta=1\}}/2\alpha$, it holds that*

$$\frac{1}{\sqrt{\gamma_k}}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0, \Sigma_C), \qquad \text{as } k \to \infty,$$

where Σ_C satisfies:

$$\left(CH - \zeta I_d\right)\Sigma_C + \Sigma_C \left(CH - \zeta I_d\right)^\top = C\Gamma C^\top.$$

Sketch of the proof. In a similar spirit as in Delyon (1996), the proof is based on the Taylor approximation $\nabla F(\theta_k) = \nabla F(\theta^\star) + H(\theta_k - \theta^\star) + o(\theta_k - \theta^\star) \simeq H(\theta_k - \theta^\star)$ and relies on the introduction of a linear stochastic algorithm. Avoiding some technicalities

related to the introduction of some event, we introduce the matrix $K = CH$ along with the iteration

$$\widetilde{\Delta}_{k+1} = \widetilde{\Delta}_k - \gamma_{k+1} K \widetilde{\Delta}_k + \gamma_{k+1} C_k w_{k+1}, \qquad k \geq 1,$$

and prove that the difference $(\theta_k - \theta^\star) - \widetilde{\Delta}_k$ is negligible. The analysis of $\widetilde{\Delta}_k$ is carried out with martingale tools where the limiting covariance is derived from a discrete time matrix-valued dynamical system algorithm.

Comparison with previous works. Theorem 5.10 stated above is comparable to Theorem 1 given in Pelletier (1998b). However, our result on the weak convergence cannot be recovered from the one of Pelletier (1998b) due to their Assumption (A1.2) about convergence rates. Indeed, this assumption would require that the sequence $(C_k)_{k\geq 0}$ converges towards C faster than $\sqrt{\gamma_k}$. This condition is either hardly meet in practice or difficult to check. Unlike this prior work, our result only requires the almost sure convergence of the sequence $(C_k)_{k\geq 0}$.

In a more restrictive setting of convex objective and online learning framework, *i.e.* in which data becomes available in a sequential order, another way to obtain the weak convergence of the rescaled sequence of iterates $(\theta_k - \theta^\star)/\sqrt{\gamma_k}$ is to rely on the results of Boyer and Godichon-Baggioni (2020). However, once again, their work rely on a particular convergence rate for the matrix sequence $(C_k)_{k\geq 0}$. This implies the derivation of an additional result on the almost sure convergence rate of the iterates. To overcome all these issues, we show in Section 5.4 that our conditions on the matrices C_k are easily satisfied in common situations.

5.3.3 Asymptotic optimality of Conditioned SGD

Another remarkable result, which directly follows from the Theorem 5.10 is now stated as a corollary.

Corollary 5.11 (Asymptotic optimality)**.** *Under the assumptions of Theorem 5.10, if $\gamma_k = 1/k$ and $C = H^{-1}$, then*

$$\sqrt{k}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0, H^{-1}\Gamma H^{-1}), \qquad \text{as } k \to \infty.$$

Moreover, let $(Z_1, \ldots, Z_d) \sim \mathcal{N}(0, I_d)$ and $(\lambda_k)_{k=1,\ldots,d}$ be the eigenvalues of the matrix $H^{-1/2}\Gamma H^{-1/2}$, we have the convergence in distribution

$$k(F(\theta_k) - F(\theta^\star)) \rightsquigarrow \sum_{k=1}^{d} \lambda_k Z_k^2, \qquad \text{as } k \to \infty.$$

The previous result shows the success of the proposed approach as the asymptotic variance obtained is the optimal one. It provides the user a practical choice for the sequence of rate, $\gamma_k = 1/k$ and also removes the assumption that $2\alpha H \succ I_d$ which is usually needed in SGD (see Theorem 5.7). Concerning the almost sure convergence of the *conditioning* matrices, we provide in Section 5.4 an explicit way to ensure that $C_k \to H^{-1}$. The above statement also provides insights about the convergence speed. It first claims that the convergence rate of $F(\theta_k)$ towards the optimum $F(\theta^\star)$, in $1/k$, is faster than the convergence rate of the iterates, in $1/\sqrt{k}$. Another important feature, which is a consequence of Proposition 5.9, is that the eigenvalues $(\lambda_k)_{k=1,\ldots,d}$ that appear in the limiting distribution are the smallest ones among all the other possible version of *conditioned* SGD (defined by the *conditioning* matrix C).

5.3.4 Convergence of iterates (θ_k) of Conditioned SGD

In order to apply both Theorem 5.10 and Corollary 5.11, it remains to check the almost sure convergence of the iterates $(\theta_k)_{k\geq 0}$ of *conditioned* SGD. Note that, in a general non-convex setting, the iterates of stochastic first-order methods can only reach local optima in the sense of stationary points, *i.e.* the iterates are expected to converge to the following set $\mathcal{S} = \{\theta \in \mathbb{R}^d : \nabla F(\theta) = 0\}$. Going in this direction, we first prove the almost sure convergence of the gradients towards zero for general *conditioned* SGD methods under mild assumptions. This theoretical result may be of independent interest. Then, under an identifiability condition on $\mathcal{S}$, one may uniquely identify a limit point $\theta^\star$ and consider the event $\{\theta_k \to \theta^\star\}$ which is needed for the weak convergence results. The following analysis is based on classical assumptions which are used in the literature to obtain the convergence of standard SGD.

Assumption 5.12 (L-smooth)**.** *The objective function $F : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and the gradient function $\nabla F : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz continuous with Lipschitz constant $L > 0$,*

$$\forall \theta, \eta \in \mathbb{R}^d, \quad \|\nabla F(\theta) - \nabla F(\eta)\|_2 \leq L\|\theta - \eta\|_2.$$

Assumption 5.13 (Lower bound)**.** *There exists $F^\star \in \mathbb{R}$ such that: $\forall \theta \in \mathbb{R}^d, F^\star \leq F(\theta)$.*

To handle the stochastic noise associated to the gradient estimates, we consider a relatively weak growth condition, related to the notion of *expected smoothness* as introduced in Gower et al. (2019) (see also Gazagnadou et al. (2019); Gower et al. (2021)). In particular, we extend the condition of Gower et al. (2019) to our general context in which the sampling distributions are allowed to change along the algorithm.

Assumption 5.14 (Growth condition)**.** *With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ such that for all $\theta \in \mathbb{R}^d, k \in \mathbb{N}$,*

$$\mathbb{E}\left[\|g(\theta, \xi_{k+1})\|_2^2 | \mathcal{F}_k\right] \leq 2\mathcal{L}(F(\theta) - F^\star) + \sigma^2.$$

This almost-sure bound on the stochastic noise $\mathbb{E}\left[\|g(\theta, \xi_k)\|_2^2 | \mathcal{F}_{k-1}\right]$ is the key to prove the almost sure convergence of the *conditioned* SGD algorithm. This weak growth condition on the stochastic noise is general and can be achieved in practice with a general Lemma available in Section 5.A.3.

Note that Assumption 6.13, often referred to as a *growth condition*, is mild since it allows the noise to be large when the iterate is far away from the optimal point. In that aspect, it contrasts with uniform bounds of the form $\mathbb{E}\left[\|g(\theta_k, \xi_{k+1})\|_2^2 | \mathcal{F}_k\right] \leq \sigma^2$ for some deterministic $\sigma^2 > 0$ (see Nemirovski et al. (2009); Nemirovski and Yudin (1983); Shalev-Shwartz et al. (2011)). Observe that such uniform bound is recovered by taking $\mathcal{L} = 0$ in Assumption 6.13 but cannot hold when the objective function F is strongly convex (Nguyen et al., 2018). Besides, fast convergence rates have been derived in Schmidt and Roux (2013) under the *strong-growth condition*: $\mathbb{E}[\|g(\theta, \xi_{k+1})\|_2^2 | \mathcal{F}_k] \leq M\|\nabla F(\theta)\|_2^2$ for some $M > 0$. Similarly to our growth condition, Bertsekas and Tsitsiklis (2000) and Bottou et al. (2018) performed an analysis under the condition $\mathbb{E}[\|g(\theta, \xi_{k+1})\|_2^2 | \mathcal{F}_k] \leq M\|\nabla F(\theta)\|_2^2 + \sigma^2$ for $M, \sigma^2 > 0$. Under Assumptions 6.11 and 5.13, we have $\|\nabla F(\theta)\|_2^2 \leq 2L\left(F(\theta) - F(\theta^\star)\right)$ (Gower et al., 2019, Proposition A.1) so our growth condition is less

restrictive. If F satisfies the Polyak-Lojasiewicz condition (Karimi et al., 2016), then our growth condition becomes a bit stronger. Another weak growth condition has been used for a non-asymptotic study in Moulines and Bach (2011).

The success of the proposed approach relies on the following condition which may be seen as an extended Robbins-Monro condition. Such condition guarantees a suitable control on the eigenvalues of the *conditioning* matrices.

Assumption 5.15 (Eigenvalues and learning rates)**.** *Let $(\mu_k)_{k\geq 1}$ and $(\nu_k)_{k\geq 1}$ be such that:*

$$\forall k \geq 1, \quad \mu_k I_d \preceq C_{k-1} \preceq \nu_k I_d.$$

The sequences $(\gamma_k)_{k\geq 1}, (\mu_k)_{k\geq 1}, (\nu_k)_{k\geq 1}$ are positive and satisfy $\sum_{k\geq 1} \gamma_k \nu_k = +\infty$, $\sum_{k\geq 1} (\gamma_k \nu_k)^2 < +\infty$ and $\limsup_k \nu_k/\mu_k < \infty$ a.s.

Observe that the last condition deals with the ratio (ν_k/μ_k) which may be seen as a conditioned number and ensures that the matrices C_k are well-conditioned. The following Theorem reveals that all these assumptions are sufficient to ensure the almost sure convergence of the gradients of *conditioned* SGD.

Theorem 5.16 (Almost sure convergence)**.** *Suppose that Assumptions 5.2, 6.11, 5.13, 6.13, 5.15 are fulfilled. Then the sequence of iterates $(\theta_k)_{k\geq 0}$ obtained by the conditioned SGD (6.3) satisfies $\nabla F(\theta_k) \to 0$ as $k \to \infty$ almost surely.*

Other convergence results concerning the sequence of iterates towards global minimizers may be obtained by considering stronger assumptions such as convexity or that F is coercive and the level sets of stationary point $\mathcal{S} \cap \{\theta, F(\theta) = y\}$ are locally finite for every $y \in \mathbb{R}^d$ (see Gadat et al. (2018)). In our analysis, the proof of Theorem 5.16 reveals that $\theta_{k+1} - \theta_k \to 0$ in L^2 and almost surely. Therefore, as soon as the stationary points are isolated, *i.e.* the objective function does not present any plateau, the sequence of iterates will converge towards a unique stationary point $\theta^\star \in \mathbb{R}^d$. This result is stated in the next Corollary.

Corollary 5.17 (Almost sure convergence)**.** *Under the assumptions of Theorem 5.16, assume that F is coercive and let $(\theta_k)_{k\geq 0}$ be the sequence of iterates obtained by the conditioned SGD (6.3), then $d(\theta_k, \mathcal{S}) \to 0$ as $k \to \infty$. In particular, if $\mathcal{S}$ is a finite set, (θ_k) converges to some $\theta^\star \in \mathcal{S}$.*

5.4 Practical procedure

For the sake of completeness, the aim of this Section is to derive a feasible procedure that achieves the optimal asymptotic variance described in Corollary 5.11. First, we present a practical way to compute the *conditioning* matrix C_k and then we show that the resulting algorithm satisfies the high-level conditions of Theorem 5.10, namely the almost sure convergence of the iterates $(\theta_k)_{k\geq 0}$ to $\theta^\star$ and of the *conditioning* matrices $(C_k)_{k\geq 0}$ to H^{-1}. This method is considered in a numerical illustration along with a novel variant of AdaGrad.

Construction of the conditioning matrix C_k. Similarly to the unavailability of exact gradients, one may not have access to values of the Hessian matrix but only stochastic versions of it (see details in numerical experiments below). As a consequence,

we consider the following framework which involves random Hessian matrices. As for gradients, a policy $(P'_k)_{k\geq 0}$ is used at each iteration to produce random Hessians through $H(\theta_k, \xi'_{k+1})$ with $\xi'_{k+1} \sim P'_k$. We work under the following property.

Assumption 5.18 (Unbiased and bounded Hessians)**.** *The Hessian generator $H : \mathbb{R}^d \times S \to \mathbb{R}^{d\times d}$ is uniformly bounded around the minimizer and is such that for all $\theta \in \mathbb{R}^d$, $H(\theta, \cdot)$ is measurable and*

$$\forall k \geq 0, \quad \mathbb{E}\left[H(\theta_k, \xi'_{k+1})|\mathcal{F}_k\right] = \nabla^2 F(\theta_k).$$

An estimate of the Hessian matrix $H = \nabla^2 F(\theta^\star)$ is now introduced as the weighted average

$$\Phi_k = \sum_{j=0}^{k} \omega_{j,k} H(\theta_j, \xi'_{j+1}) \quad \text{with} \quad \sum_{j=0}^{k} \omega_{j,k} = 1. \tag{5.4}$$

The previous estimate has two advantages. First, thanks to averaging, the noise associated to each evaluation $H(\theta_j, \xi'_{j+1})$ will eventually vanished due to the sum of martingale increments. Second, the weights $\omega_{j,k}$ may help to give more importance to most recent iterates. In the idea that θ_k lies near $\theta^\star$ eventually, it might be helpful to reduce the bias when estimating $H = \nabla^2 F(\theta^\star)$.

Proposition 5.19. *Let $(\Phi_k)_{k\geq 0}$ be obtained by (5.4). Suppose that Assumptions 5.4 and 5.18 are fulfilled and that $\theta_k \to \theta^\star$ almost surely . If $\sup_{0\leq j\leq k} \omega_{j,k} = O(1/k)$, then we have $\Phi_k \to H = \nabla^2 F(\theta^\star)$ almost surely.*

A common choice is to take equal weights $\omega_{j,k} = (k+1)^{-1}$. However, since the last iterates are more likely to bring more relevant information through their Hessian estimates, we advocate the use of adaptive weights of the form $\omega_{j,k} \propto \exp(-\eta\|\theta_j - \theta_k\|_1)$ with a parameter $\eta \geq 0$ that recovers equal weights with $\eta = 0$. These two weights sequences satisfy the assumption of Proposition 5.19. They are considered in the numerical illustration of the next Section. While inverting Φ_k would produce a simple estimate of H^{-1}, such an approach might result in a certain instability in practice caused by large jumps towards wrong directions (large eigenvalues) or a too restrictive visit along other components (vanishing eigenvalues). To overcome this issue, we rely on the following filter which clamps the eigenvalues of a symmetric matrix. For any symmetric matrix S and two positive numbers $0 < a < b$, denote by $S[a, b]$ the associated matrix where all the eigenvalues are clamped to $[a, b]$, *i.e.*, any eigenvalue λ of S is modified as $\lambda \leftarrow \max\{a, \min\{\lambda, b\}\}$.

Let $(\lambda_k^{(m)})_{k\geq 1}$ and $(\lambda_k^{(M)})_{k\geq 1}$ be two sequence of positive numbers such that $\lambda_k^{(m)} \leq \lambda_k^{(M)}$ for all $k \geq 1$. Define the matrices

$$\forall k \in \mathbb{N}, \quad C_k = \left(\Phi_k[(\lambda_{k+1}^{(M)})^{-1}, (\lambda_{k+1}^{(m)})^{-1}]\right)^{-1}. \tag{5.5}$$

Observe that such a definition guarantees two important properties. First, C_k is a real symmetric positive definite matrix which satisfies the matrix inequality

$$\lambda_{k+1}^{(m)} I_d \preceq C_k \preceq \lambda_{k+1}^{(M)} I_d.$$

Second, in virtue of Proposition 5.19, the matrix Φ_k converges almost surely to H so that, as soon as the sequences $(\lambda_k^{(m)})_{k\geq 1}$ and $(\lambda_k^{(M)})_{k\geq 1}$ go to 0 and $+\infty$ respectively, the matrix C_k converges almost surely to H^{-1} (as recommended by Corollary 5.11). Therefore, we obtain a feasible procedure leading to asymptotic optimality.

Theorem 5.20 (Asymptotic optimality of the iterates)**.** *Let $(\theta_k)_{k\geq 0}$ be obtained by conditioned SGD (6.3) with $\gamma_k = 1/k$, Φ_k defined by (5.4), $\lambda_k^{(m)} \to 0, \lambda_k^{(M)} \to +\infty$ and C_k given by (5.5). Suppose that Assumptions 5.2 to 5.15 are fulfilled and $\sup_{0\leq j\leq k} \omega_{j,k} = O(1/k)$. We have*

$$\sqrt{k}(\theta_k - \theta^\star) \rightsquigarrow \mathcal{N}(0, H^{-1}\Gamma H^{-1}), \qquad \text{as } k \to \infty.$$

This algorithm is theoretically asymptotically optimal. However in practice, adaptive gradient methods described in Table 5.1 have become the workhorse for training deep learning models as they take advantage of low rank-approximations and diagonal scalings. Interestingly, the *conditioned* matrices involved in these methods are linked to gradient estimates and thus to covariance matrices Γ_k (see Assumption 5.5) rather than the Hessian H. Indeed, since $\theta^\star \in \mathcal{S}$, we have for the limiting covariance $\Gamma = \mathbb{E}_\xi[g(\theta^\star, \xi)g(\theta^\star, \xi)^\top]$. Consider a variant of AdaGrad which accumulates the average gradients $G_k = \delta I + (1/k)\sum_{i=1}^k g_i g_i^\top$ and $C_k = G_k^{-1/2}$. Averaging allows to anneal the stochastic noise of the gradient estimate. By the law of large numbers, the limiting matrix in our Theorem 5.10 will be $C = (\Gamma + \delta I)^{-1/2}$. For illustrative purposes, this novel method is considered in experiments with futher details in Appendix.

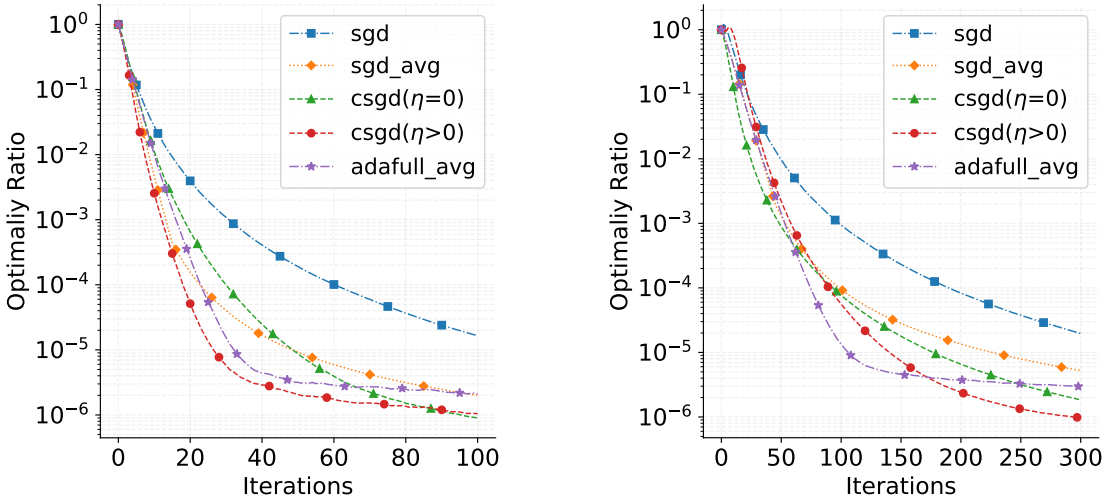
Numerical illustration. Consider the empirical risk minimization framework applied to Generalized Linear Models. Given a data matrix $X = (x_{i,j}) \in \mathbb{R}^{n\times d}$ with labels $y \in \mathbb{R}^n$ and a regularization parameter $\lambda > 0$, we are interested in $\min_{\theta\in\mathbb{R}^d}\{F(\theta) = (1/n)\sum_{i=1}^n f_i(\theta)\}$, $f_i(\theta) = \mathcal{L}(x_i^\top\theta, y_i) + \lambda\Omega(\theta)$, $\mathcal{L} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is smooth loss function and $\Omega : \mathbb{R}^d \to \mathbb{R}_+$ is a smooth convex regularizer chosen as Tikhonov regularization $\Omega(\theta) = \frac{1}{2}\|\theta\|_2^2$. The gradient and Hessian of each component f_i are given for all $i = 1, \dots, n$ by

$$\nabla f_i(\theta) = \mathcal{L}'(x_i^\top\theta, y_i)x_i + \lambda\theta$$
$$\nabla^2 f_i(\theta) = \mathcal{L}''(x_i^\top\theta, y_i)x_i x_i^\top + \lambda I_d,$$

where $\mathcal{L}'(\cdot,\cdot)$ and $\mathcal{L}''(\cdot,\cdot)$ are the first and second derivative of $\mathcal{L}(\cdot,\cdot)$ with respect to the first argument. As stated in Example 1 of Section 6.2, stochastic versions of both the gradient and the Hessian of the objective F can be easily computed using only a batch $B \subset \{1, \dots, n\}$ of data and $\nabla_B F(\theta) = \sum_{i\in B}\nabla f_i(\theta)/|B|$ (resp. $\nabla_B^2 F(\theta) = \sum_{i\in B}\nabla^2 f_i(\theta)/|B|$) for the gradient (resp. Hessian) estimate. Note that these random generators meet Assumptions 5.2 and 5.18 as they produce unbiased estimates of the gradient and the Hessian matrix respectively.

We focus on Ridge regression on simulated data with $n = 10,000$ samples in dimensions $d \in \{20; 100\}$ with $|B| = 16$. Starting from the null vector $\theta_0 = (0, \dots, 0) \in \mathbb{R}^d$, we use optimal learning rate of the form $\gamma_k = \alpha/(k + k_0)$ (Bottou et al., 2018) and set $\lambda_k^{(m)} \equiv 0, \lambda_k^{(M)} = \Lambda\sqrt{k}$ where α, k_0 and Λ are tuned using a grid search. The means of the optimality ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$, obtained over 100 independent runs, are presented in Figure 5.1. The methods in competition are *sgd*: standard stochastic gradient descent; *sgd_ avg*: Polyak-averaging variant with a burn-in period; *csgd*($\eta = 0$) and *csgd*($\eta > 0$): *conditioned* sgd methods with equal and adaptive

weights where the matrix Φ_k is given by Equation (5.4); *adafull_ avg*: The variant of Adagrad presented above with an average for G_k instead of the cumulative sum provided in the literature of Adagrad.



(a) Ridge $d = 20$)

(b) Ridge $d = 100$

Figure 5.1 – Ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$ for Ridge regression in dimension $d \in \{20; 100\}$.

5.5 Conclusion and Discussion

We derived an asymptotic theory for *conditioned* stochastic gradient descent methods in a general non-convex setting. We showed that, compared to standard SGD methods, the only additional assumption required to obtain the weak convergence is the almost sure convergence of the *conditioning* matrices. The use of appropriate *conditioning* matrices with the help of Hessian estimates is the key to achieve asymptotic optimality in the sense of minimal variance. While our study focuses on the weak convergence of the rescaled sequence of iterates - an appropriate tool to deal with efficiency issues because algorithms can be easily compared through their asymptotic variances - it would be interesting to complement our asymptotic results with concentration inequalities and non-asymptotic bounds. This research direction, left for future work, may be done at the expense of additional assumptions, *e.g.*, strong convexity of the objectve function combined with bounded gradients.

From a practical standpoint, the approach of Section 5.4 may not be computationally optimal as it requires eigenvalue decomposition. However, *conditioned* SGD methods and especially stochastic second-order methods do not actually require the full computation of a matrix decomposition but rely on matrix-vector products which may be performed in $O(d^2)$ operations. Futhermore, using low-rank approximation with BFGS algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) and its variant L-BFGS (Liu and Nocedal, 1989), those algorithms approximately invert Hessian matrices in $O(d)$ operations. More recently, this technique was extended to the online learning framework (Schraudolph et al., 2007) and a purely stochastic setting (Moritz et al., 2016). Similarly, the different adaptive optimizers presented in Section 5.3.1 are concerned with both fast computations and high precision. Designing an efficient *con-*

ditioned SGD algorithm involves a careful trade-off between the low-memory storage of the scaling matrix representation C_k and the quality of its approximation of either the inverse Hessian $\nabla^2 F(\theta^\star)^{-1}$ or the information brought in by the underlying geometry of the problem.

5.A Proofs

Appendix 5.A.1 presents the proof of the weak convergence of *conditioned* SGD. Appendix 5.A.2 gives additional propositions, namely the optimality of the inverse of the Hessian matrix and the almost sure convergence of the matrix Φ_k built in Equation (5.4). Appendix 5.A.3 deals with auxiliary results on the expected smoothness condition and its links with our growth condition. Appendix 5.A.4 is concerned about the almost sure convergence (Theorem 5.16). Appendix 5.B gathers Robbins-Siegmund theorem and technical Lemmas that are useful for the analysis. For illustrative purposes, Appendix 5.B.3 gathers numerical experiments.

5.A.1 Proof of the weak convergence (Theorem 5.10)

For any matrix $A \in \mathbb{R}^{d\times d}$, we denote by $\|A\| = \max_{\|u\|_2=1} \|Au\|_2$ the operator norm associated to the Euclidian norm and by $\rho(A)$ the spectral radius of A, *i.e.*, $\rho(A) = \max\{|\lambda_1|, \dots, |\lambda_n|\}$ where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of A. We also introduce $\lambda_{\min}(A) = \min\{|\lambda_1|, \dots, |\lambda_n|\}$. Note that when A is symmetric $\|A\| = \rho(A)$ and recall that the spectral radius is a (submultiplicative) norm on the real linear space of symmetric matrices.

Structure of the proof.

In virtue of Assumption 5.6, there exist $\delta, \varepsilon > 0$ such that almost surely

$$\sup_{k\geq 0} \mathbb{E}[\|w_{k+1}\|_2^{2+\delta} | \mathcal{F}_k] \mathbb{1}_{\{\|\theta_k - \theta^\star\|_2 \leq \varepsilon\}} < \infty. \tag{5.6}$$

An important event in the following is

$$\mathcal{A}_k = \{\|\theta_k - \theta^\star\|_2 \leq \varepsilon,\ \|C_k\| < 2\|C\|,\ \|\Gamma_k\| \leq 2\|\Gamma\|\}.$$

By assumption, this event has probability going to 1.

Introduce the difference

$$\Delta_k = \theta_k - \theta^\star,$$

and remark that Δ_k is subjected to the iteration:

$$\begin{aligned}
&\Delta_0 = \theta_0 - \theta^\star, \\
&\Delta_{k+1} = \Delta_k - \gamma_{k+1} C_k \nabla F(\theta_k) + \gamma_{k+1} C_k w_{k+1}, \qquad k \geq 0,
\end{aligned}$$

with $w_{k+1} = \nabla F(\theta_k) - g(\theta_k, \xi_{k+1})$. We have by assumption that $C_k \to C$ almost surely and we can define $K = \lim_{k\to\infty} C_k H = CH$. The proof relies on the introduction of an auxiliary stochastic algorithm which follows the iteration:

$$\begin{aligned}
&\widetilde{\Delta}_0 = \theta_0 - \theta^\star \\
&\widetilde{\Delta}_{k+1} = \widetilde{\Delta}_k - \gamma_{k+1} K \widetilde{\Delta}_k + \gamma_{k+1} C_k w_{k+1} \mathbb{1}_{\mathcal{A}_k}, \qquad k \geq 0
\end{aligned}$$

The previous algorithm is a linear approximation of the algorithm that defines Δ_k in the sense that $\nabla F(\theta_k) = \nabla F(\theta^\star) + H(\theta_k - \theta^\star) + o(\theta_k - \theta^\star) \simeq H(\theta_k - \theta^\star)$ has been linearly expanded around $\theta^\star$. Writing

$$\Delta_k = \widetilde{\Delta}_k + (\Delta_k - \widetilde{\Delta}_k),$$

and invoking the Slutsky lemma, the proof will be complete as soon as we obtain that

$$\gamma_k^{-1/2}\widetilde{\Delta}_k \rightsquigarrow \mathcal{N}(0, \Sigma), \tag{5.7}$$

$$(\Delta_k - \widetilde{\Delta}_k) = o_{\mathbb{P}}(\gamma_k^{1/2}). \tag{5.8}$$

Denote by $\sqrt{H}$ the positive square root of the real symmetric positive definite matrix H and consider the transformation $\Theta_k = \sqrt{H}\widetilde{\Delta}_k$ which satisfies

$$\begin{aligned}\Theta_0 &= \sqrt{H}\widetilde{\Delta}_0\\ \Theta_{k+1} &= \Theta_k - \gamma_{k+1}\widetilde{K}\Theta_k + \gamma_{k+1}\sqrt{H}C_k w_{k+1}\mathbb{1}_{\mathcal{A}_k}, \qquad k \geq 1,\end{aligned}$$

where $\widetilde{K} = \sqrt{H}C\sqrt{H} \in \mathcal{S}_d^{++}(\mathbb{R})$ is a real symmetric positive definite matrix. The sequence $(\Theta_k)_{k\geq 0}$ is easier to study than $\widetilde{\Delta}_k$ because contrary to $\widetilde{K}$, the matrix $K = CH$ is not symmetric in general unless C and H commute. In view of Assumption 5.4, the eigenvalues of $\widetilde{K}$ are real and positive. Denote by λ_m (*resp.* λ_M) the smallest (*resp.* the largest) eigenvalue of $\widetilde{K}$, *i.e.*,

$$\lambda_m = \lambda_{\min}(\widetilde{K}), \quad \lambda_M = \lambda_{\max}(\widetilde{K}).$$

Because CH is similar to $\widetilde{K}$, they share the same eigenvalues. Since by assumption, the eigenvalues of $(CH - \zeta I_d)$ are positive, we have $2\alpha\lambda_m > \mathbb{1}_{\{\beta=1\}}$. For all $k \geq 1$, introduce the real symmetric matrix $A_k = I - \gamma_k\widetilde{K}$. Observe that all these matrices commute, i.e., for any $i, j \geq 0$, we have $A_iA_j = A_jA_i$. For any $k, n \geq 0$, denote the matrices product

$$\begin{cases} \Pi_{n,k} &= A_n \dots A_{k+1} \text{ if } k < n\\ \Pi_{n,k} &= I_d \text{ if } k \geq n, \Pi_n = \Pi_{n,0}\end{cases}$$

Since the matrices A_k commute, we have $\Pi_{n,k}^\top = \Pi_{n,k}$ is also real symmetric.

Step 1. Proof of Equation (5.7).
The random process $(\Theta_k)_{k\geq 0}$ follows the recursion equation

$$\Theta_k = A_k\Theta_{k-1} + \gamma_k\sqrt{H}C_{k-1}w_k\mathbb{1}_{\mathcal{A}_{k-1}}.$$

We have by induction

$$\Theta_n = \Pi_n\Theta_0 + \sum_{k=1}^{n}\gamma_k\Pi_{n,k}\sqrt{H}C_{k-1}w_k\mathbb{1}_{\mathcal{A}_{k-1}},$$

and the rescaled process is equal to

$$\frac{\Theta_n}{\sqrt{\gamma_n}} = \underbrace{\frac{\Pi_n}{\sqrt{\gamma_n}}\Theta_0}_{\text{initial error } Y_n} + \underbrace{\sum_{k=1}^{n}\frac{\gamma_k}{\sqrt{\gamma_n}}\Pi_{n,k}\sqrt{H}C_{k-1}w_k\mathbb{1}_{\mathcal{A}_{k-1}}}_{\text{sampling error } M_n}.$$

Bound on the initial error.
Define $\tau_n = \sum_{k=1}^n \gamma_k$ the partial sum of the learning rates. Since Π_n is symmetric, we have $\rho(\Pi_n\Theta_0) \leq \rho(\Pi_n)\|\Theta_0\|_2$. In view of Lemma 5.29, since $\gamma_k \to 0$, there exists $j \geq 1$ such that

$$\rho(\Pi_n) \leq \rho(\Pi_j)\exp(-\lambda_m(\tau_n - \tau_j)).$$

Therefore, the initial error is bounded by

$$\rho(Y_n) \leq \rho(\Pi_j)\exp(\lambda_m\tau_j)\|\Theta_0\|_2\exp(d_n) \quad \text{with} \quad d_n = -\lambda_m\tau_n - \log(\sqrt{\gamma_n}).$$

Using Lemma 5.30, we can treat the two cases $\beta < 1$ and $\beta = 1$. On the one hand, if $\beta < 1$ then we always have $d_n \to -\infty$. On the other hand, if $\beta = 1$, we have $d_n \sim \left(\frac{1}{2} - \gamma\lambda_m\right)\log(n)$ and the condition $2\alpha\lambda_m - 1 > 0$ ensures $d_n \to -\infty$. In both cases we get $\exp(d_n) \to 0$ and the initial error vanishes to 0.

Weak convergence of the sampling error.
Consider the random process

$$M_n = \gamma_n^{-1/2}\sum_{k=1}^n \gamma_k\Pi_{n,k}\sqrt{H}C_{k-1}w_k\mathbb{1}_{\mathcal{A}_{k-1}}.$$

Note that θ_k, $\mathcal{A}_k$ and C_k are $\mathcal{F}_k$-measurable. As a consequence, M_n is a sum of martingale increments and we may rely on the following central limit theorem for martingale arrays.

Theorem 5.21. *(Hall and Heyde, 1980, Corollary 3.1) Let $(W_{n,i})_{1\leq i\leq n,\, n\geq 1}$ be a triangular array of random vectors such that*

$$\mathbb{E}[W_{n,i} \mid \mathcal{F}_{i-1}] = 0, \quad \textit{for all } 1 \leq i \leq n, \tag{5.9}$$

$$\sum_{i=1}^n \mathbb{E}[W_{n,i}W_{n,i}^\top \mid \mathcal{F}_{i-1}] \to V^* \geq 0, \quad \textit{in probability,} \tag{5.10}$$

$$\sum_{i=1}^n \mathbb{E}[\|W_{n,i}\|^2\mathbb{1}_{\{\|W_{n,i}\|>\varepsilon\}} \mid \mathcal{F}_{i-1}] \to 0, \quad \textit{in probability,} \tag{5.11}$$

then, $\sum_{i=1}^n W_{n,i} \rightsquigarrow \mathcal{N}(0, V^*)$, as $n \to \infty$.

We start by verifying (5.10). Let $D_k = \sqrt{H}C_{k-1}\Gamma_{k-1}C_{k-1}^T\sqrt{H}\mathbb{1}_{\mathcal{A}_{k-1}} \in \mathcal{S}_d(\mathbb{R})$. The quadratic variation of M_n is given by

$$\Sigma_n = \gamma_n^{-1}\sum_{k=1}^n \gamma_k^2\Pi_{n,k}D_k\Pi_{n,k}^\top.$$

First we can check that Σ_n is bounded. Using the triangle inequality and since the operator norm is submultiplicative, we have

$$\|\Sigma_n\| \leq \gamma_n^{-1}\sum_{k=1}^n \gamma_k^2\|\Pi_{n,k}D_k\Pi_{n,k}^T\| \leq \gamma_n^{-1}\sum_{k=1}^n \gamma_k^2\|D_k\|\|\Pi_{n,k}\|^2 = \gamma_n^{-1}\sum_{k=1}^n \gamma_k^2\|D_k\|\rho(\Pi_{n,k})^2,$$

where we use in the last equality that $\Pi_{n,k}$ is real symmetric so $\|\Pi_{n,k}\| = \rho(\Pi_{n,k})$. On the event $\mathcal{A}_{k-1}$, the matrices C_{k-1} and Γ_{k-1} are bounded as $\|C_{k-1}\| \leq 2\|C\|$ and $\|\Gamma_{k-1}\| \leq 2\|\Gamma\|$ leading to the following bound for the matrix D_k,

$$\begin{aligned}\|D_k\| &= \|\sqrt{H}C_{k-1}\Gamma_{k-1}C_{k-1}^T\sqrt{H}\mathbb{1}_{\mathcal{A}_{k-1}}\| \\ &\leq \|H\|\|\Gamma_{k-1}\|\|C_{k-1}\|^2\mathbb{1}_{\mathcal{A}_{k-1}} \\ &\leq 8\|H\|\|\Gamma\|\|C\|^2 = U_D.\end{aligned}$$

It follows that

$$\|\Sigma_n\| \leq U_D\gamma_n^{-1}\sum_{k=1}^{n}\gamma_k^2\rho(\Pi_{n,k})^2.$$

In view of Lemma 5.29, we shall split the summation from $k = 1, \ldots, j$ and $k = j+1, \ldots, n$ as

$$\begin{aligned}\gamma_n^{-1}\sum_{k=1}^{n}\gamma_k^2\rho\left(\Pi_{n,k}\right)^2 &= \gamma_n^{-1}\sum_{k=1}^{j}\gamma_k^2\rho\left(\Pi_{n,k}\right)^2 + \gamma_n^{-1}\sum_{k=j+1}^{n}\gamma_k^2\rho\left(\Pi_{n,k}\right)^2 \\ &\leq \underbrace{\gamma_n^{-1}\sum_{k=1}^{j}\gamma_k^2\rho\left(\Pi_{n,k}\right)^2}_{a_n} + \underbrace{\gamma_n^{-1}\sum_{k=j+1}^{n}\gamma_k^2\prod_{i=k+1}^{n}(1-\lambda_m\gamma_i)^2}_{b_n}.\end{aligned}$$

For the first term a_n, we have for all $k = 1, \ldots, j$

$$\rho(\Pi_{n,k}) \leq \rho(\Pi_{n,j}) \leq \prod_{i=j+1}^{n}(1-\lambda_m\gamma_i) \leq \exp(-\lambda_m(\tau_n - \tau_j)),$$

which implies since (γ_k) is decreasing with $\gamma_1 = \alpha$ that

$$\sum_{k=1}^{j}\gamma_k^2\rho\left(\Pi_{n,k}\right)^2 \leq \alpha\tau_j\exp(-2\lambda_m(\tau_n - \tau_j)).$$

Therefore, similarly to the initial error term, we get

$$a_n \leq \alpha\tau_j\exp(2\lambda_m\tau_j))\exp(d_n) \quad \text{with} \quad d_n = -2\lambda_m\tau_n - \log(\gamma_n),$$

and the condition $2\alpha\lambda_m - 1 > 0$ ensures $d_n \to -\infty$ so that a_n goes to 0 and is almost surely bounded by U_a.

For the second term b_n, we can apply Lemma 5.27 and need to distinguish between the two cases:

• ($\beta = 1$) If $\gamma_n = \alpha/n$, since $2\alpha\lambda_m > 1$, we can apply Lemma 5.27 ($p = 1, m = 2, \lambda = \lambda_m\alpha, x_j = 0, \varepsilon_k = \alpha^2$) and obtain

$$b_n \leq \frac{\alpha^2}{2\alpha\lambda_m - 1} = U_b.$$

• ($\beta < 1$) If $\gamma_n = \gamma/n^\beta$, we deduce the same as before because $\lambda_m > 0$.

Finally in both cases, we get

$$\|\Sigma_n\| \leq U_D \left(U_a + U_b\right). \tag{5.12}$$

We now derive the limit of Σ_n. We shall use a recursion equation to recover a stochastic approximation scheme. Note that

$$\gamma_n \Sigma_n = \sum_{k=1}^{n} \gamma_k^2 \Pi_{n,k} D_k \Pi_{n,k}^T \tag{5.13}$$

$$= \gamma_n^2 D_n + A_n \left(\sum_{k=1}^{n-1} \gamma_k^2 \Pi_{n-1,k} D_k \Pi_{n-1,k}^T \right) A_n^\top, \tag{5.14}$$

and recognize

$$\gamma_n \Sigma_n = \gamma_n^2 D_n + \gamma_{n-1} A_n \Sigma_{n-1} A_n^\top.$$

Replacing the symmetric matrix $A_n = I - \gamma_n \widetilde{K}$, we get (because Σ_n is bounded almost surely)

$$\begin{aligned}\gamma_n \Sigma_n &= \gamma_n^2 D_n + \gamma_{n-1}(I - \gamma_n \widetilde{K}) \Sigma_{n-1} (I - \gamma_n \widetilde{K}) \\ &= \gamma_n^2 D_n + \gamma_{n-1} \left[\Sigma_{n-1} - \gamma_n \Sigma_{n-1} \widetilde{K} - \gamma_n \widetilde{K} \Sigma_{n-1} + O(\gamma_n^2) \right].\end{aligned}$$

Divide by γ_n to obtain

$$\Sigma_n = \gamma_n D_n + \frac{\gamma_{n-1}}{\gamma_n} \left[\Sigma_{n-1} - \gamma_n (\widetilde{K} \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K}) + O(\gamma_n^2) \right],$$

and we recognize a stochastic approximation scheme

$$\Sigma_n = \Sigma_{n-1} - \gamma_n \left[\widetilde{K} \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K} - D_n \right] + \frac{\gamma_{n-1} - \gamma_n}{\gamma_n} \Sigma_{n-1} + O(\gamma_{n-1}\gamma_n + |\gamma_{n-1} - \gamma_n|)$$

Recall that when $\beta < 1$ we have

$$\frac{1}{\gamma_n} - \frac{1}{\gamma_{n-1}} \to 0, \text{ i.e., } \frac{\gamma_{n-1} - \gamma_n}{\gamma_n} = o(\gamma_n).$$

- ($\beta = 1$) If $\gamma_n = \alpha/n$ we get

$$\Sigma_n = \Sigma_{n-1} - \frac{\alpha}{n} \left[\widetilde{K} \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K} - \frac{1}{\alpha} \Sigma_{n-1} - D_n \right] + O(n^{-2})$$

$$\Sigma_n = \Sigma_{n-1} - \frac{\alpha}{n} \left[\left(\widetilde{K} - \frac{I}{2\alpha} \right) \Sigma_{n-1} + \Sigma_{n-1} \left(\widetilde{K} - \frac{I}{2\alpha} \right) - D_n \right] + O(n^{-2}).$$

- ($\beta < 1$) If $\gamma_n = \alpha/n^\beta$ we get

$$\Sigma_n = \Sigma_{n-1} - \gamma_n \left[\widetilde{K} \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K} - D_n \right] + o(\gamma_n).$$

Recall that $\zeta = \mathbb{1}_{\{\beta=1\}}/(2\alpha)$ and define $\widetilde{K}_\zeta = \widetilde{K} - \zeta I$, so that in both cases, the recursion equation becomes

$$\Sigma_n = \Sigma_{n-1} - \gamma_n \left[\widetilde{K}_\zeta \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K}_\zeta^\top - D_n \right] + o(\gamma_n).$$

We can vectorize this equation. The vectorization of an $m \times n$ matrix $A = (a_{i,j})$, denoted $\mathrm{vec}(A)$, is the $mn \times 1$ column vector obtained by stacking the columns of the matrix A on top of one another:

$$\mathrm{vec}(A) = [a_{1,1}, \dots, a_{m,1}, a_{1,2}, \dots, a_{m,2}, \dots, a_{1,n}, \dots, a_{m,n}]^T.$$

Applying this operator to our stochastic approximation scheme gives

$$\mathrm{vec}(\Sigma_n) = \mathrm{vec}(\Sigma_{n-1}) - \gamma_n \left[\mathrm{vec}\left(\widetilde{K}_\zeta \Sigma_{n-1} + \Sigma_{n-1} \widetilde{K}_\zeta^\top \right) - \mathrm{vec}(D_n) \right] + o(\gamma_n).$$

Denote by $\otimes$ the Kronecker product, we have the following property

$$\mathrm{vec}\left(K_\zeta \Sigma_{n-1} + \Sigma_{n-1} K_\zeta^\top \right) = \left(I_d \otimes K_\zeta + K_\zeta^\top \otimes I_d \right) \mathrm{vec}(\Sigma_{n-1}).$$

Define D as the almost sure limit of D_n, *i.e.*

$$D = \lim_{n \to \infty} D_n = \sqrt{H} C \Gamma C \sqrt{H}.$$

Introduce $v_n = \mathrm{vec}(\Sigma_n)$ and $Q = \left(I_d \otimes \widetilde{K}_\zeta + \widetilde{K}_\zeta \otimes I_d \right)$. We have almost surely

$$\begin{aligned} v_n &= v_{n-1} - \gamma_n \left(Q v_{n-1} - \mathrm{vec}(D) \right) + \gamma_n \mathrm{vec}(D_n - D) + o(\gamma_n) \\ &= v_{n-1} - \gamma_n \left(Q v_{n-1} - \mathrm{vec}(D) \right) + \varepsilon_n \gamma_n \end{aligned}$$

where $\varepsilon_n \to 0$ almost surely. This is a stochastic approximation scheme with the affine function $h(v) = Qv - \mathrm{vec}(D)$ for $v \in \mathbb{R}^{d^2}$. Let $v^\star$ be the solution of $h(v) = 0$ which is well defined since $Q = \left(I_d \otimes \widetilde{K}_\zeta + \widetilde{K}_\zeta^\top \otimes I_d \right)$ is invertible. Indeed, the eigenvalues of Q are $\mu_i + \mu_j$, $1 \leq i, j \leq d$, where the μ_i, $i = 1, \dots, d$ are the eigenvalues of $\widetilde{K}_\zeta$. Equivalently, the eigenvalues of Q are of the form $(\lambda_i - \zeta) + (\lambda_j - \zeta)$ where the λ_i, $i = 1, \dots, d$ are the eigenvalues of $\widetilde{K}$. Because $\lambda_m > \zeta$, we have that $Q \succ 0$. As a consequence

$$\begin{aligned} \left(v_n - v^\star \right) &= \left(v_{n-1} - v^\star \right) - \gamma_n \left(h(v_{n-1}) - h(v^\star) \right) + \varepsilon_n \gamma_n \\ &= \left(v_{n-1} - v^\star \right) - \gamma_n Q \left(v_{n-1} - v^\star \right) + \varepsilon_n \gamma_n \\ &= B_n \left(v_{n-1} - v^\star \right) + \varepsilon_n \gamma_n, \end{aligned}$$

with $B_n = (I_{d^2} - \gamma_n Q)$. By induction, we obtain

$$\left(v_n - v^\star \right) = \left(B_n \dots B_1 \right) \left(v_0 - v^\star \right) + \sum_{k=1}^{n} \gamma_k \left(B_n \dots B_{k+1} \right) \varepsilon_k,$$

Define $\lambda_Q = \lambda_{\min}(Q) > 0$ and remark that

$$\| B_n \dots B_{k+1} \| \leq \prod_{j=k+1}^{n} \| B_j \| = \prod_{j=k+1}^{n} (1 - \gamma_j \lambda_Q).$$

It follows that

$$\|v_n - v^\star\|_2 \leq \|B_n \dots B_1\| \|v_0 - v^\star\|_2 + \sum_{k=1}^{n} \gamma_k \|B_n \dots B_{k+1}\| \|\varepsilon_k\|_2$$
$$\leq \prod_{j=1}^{n} (1 - \gamma_j \lambda_Q) \|v_0 - v^\star\|_2 + \sum_{k=1}^{n} \gamma_k \prod_{j=k+1}^{n} (1 - \gamma_j \lambda_Q) \|\varepsilon_k\|_2$$

Applying Lemma 5.27 we obtain that the right-hand side term goes to 0. The left-hand side term goes to 0 under the effect of the product by definition of $(\gamma_k)_{k\geq 1}$. We therefore conclude that $v_n \to v^\star$ almost surely. From easy manipulation involving $\mathrm{vec}(\cdot)$ and $\otimes$, this is equivalent to $\Sigma_n \to \Sigma$, where Σ is the solution of the Lyapunov equation

$$(\widetilde{K} - \zeta I)\Sigma + \Sigma(\widetilde{K} - \zeta I) = D.$$

Now we turn our attention to (5.11). We need to show that almost surely,

$$\gamma_n^{-1} \sum_{k=1}^{n} \gamma_k^2 \mathbb{E}[\|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2^2 \mathbb{1}_{\{\gamma_k \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2 > \varepsilon \gamma_n^{1/2}\}} \mid \mathcal{F}_{k-1}] \mathbb{1}_{\mathcal{A}_{k-1}} \to 0.$$

We have

$$\mathbb{E}[\gamma_n^{-1} \gamma_k^2 \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2^2 \mathbb{1}_{\{\gamma_k \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2 > \varepsilon \gamma_n^{1/2}\}} \mid \mathcal{F}_{k-1}]$$
$$\leq \varepsilon^{-\delta} \mathbb{E}[(\gamma_n^{-1/2} \gamma_k \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2)^{2+\delta} \mid \mathcal{F}_{k-1}]$$
$$\leq \varepsilon^{-\delta} (\gamma_n^{-1/2} \gamma_k \|\Pi_{n,k} \sqrt{H} C_{k-1}\|^{2+\delta} \mathbb{E}[\|w_k\|_2^{2+\delta} \mid \mathcal{F}_{k-1}].$$

Let $U(\omega) = \sup_{k\geq 1} \mathbb{E}[\|w_k\|_2^{2+\delta} \mid \mathcal{F}_{k-1}] \mathbb{1}_{\mathcal{A}_{k-1}}$ which is almost surely finite by Assumption 5.6. We get

$$\mathbb{E}[\gamma_n^{-1} \gamma_k^2 \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2^2 \mathbb{1}_{\{\gamma_k \|\Pi_{n,k} \sqrt{H} C_{k-1} w_k\|_2 > \varepsilon \gamma_n^{1/2}\}} \mid \mathcal{F}_{k-1}] \mathbb{1}_{\mathcal{A}_{k-1}}$$
$$\leq \varepsilon^{-\delta} \left(2 \|\sqrt{H}\| \|C\|\right)^{2+\delta} U(\omega) (\gamma_n^{-1/2} \gamma_k \rho(\Pi_{n,k}))^{2+\delta}$$

Hence by showing that

$$\sum_{k=1}^{n} (\gamma_n^{-1/2} \gamma_k \rho(\Pi_{n,k}))^{2+\delta} \to 0,$$

we will obtain (5.11). The previous convergence can be deduced from Lemma 5.27 with $p = 1 + \delta/2$, $m = 2 + \delta$, $\epsilon_k = \gamma_k^{\delta/2}$, checking that $(2 + \delta)\alpha\lambda_m > 1 + \delta/2$.

Step 2. Proof of Equation (5.8).

A preliminary step to the derivation of Equation (5.8) is to obtain that $\widetilde{\Delta}_k \to 0$ almost surely. For any θ and η in $\mathbb{R}^d$, we have

$$\|\theta\|^2 = \|\eta\|^2 + 2\eta^\top(\theta - \eta) + \|\theta - \eta\|^2$$

implying that for all $k \geq 0$

$$\mathbb{E}[\|\Theta_{k+1}\|^2 | \mathcal{F}_k] = \|\widetilde{\Theta}_k\|^2 - 2\gamma_{k+1} \Theta_k^\top \widetilde{K} \Theta_k + \gamma_{k+1}^2 \mathbb{E}[\|\widetilde{K}\Theta_k - C_k w_{k+1} \mathbb{1}_{\mathcal{A}_k}\|^2 | \mathcal{F}_k].$$

Since (w_k) is a martingale increment and because on $\mathcal{A}_k$, $\rho(C_k) \leq 2\rho(C)$, we get

$$\begin{aligned}\mathbb{E}[\|\widetilde{K}\Theta_k - C_k w_{k+1}\mathbb{1}_{\mathcal{A}_k}\|^2|\mathcal{F}_k] &= \mathbb{E}[\|\widetilde{K}\Theta_k\|^2|\mathcal{F}_k] + \mathbb{E}[\|C_k w_{k+1}\mathbb{1}_{\mathcal{A}_k}\|^2|\mathcal{F}_k] \\ &\leq \lambda_M^2\|\Theta_k\|^2 + \rho(C_k)^2\mathbb{E}[\|w_{k+1}\mathbb{1}_{\mathcal{A}_k}\|^2|\mathcal{F}_k] \\ &\leq \lambda_M^2\|\Theta_k\|^2 + 4\rho(C)^2\mathbb{E}[\|w_{k+1}\mathbb{1}_{\mathcal{A}_k}\|^2|\mathcal{F}_k],\end{aligned}$$

Injecting this bound in the previous equality yields

$$\mathbb{E}[\|\Theta_{k+1}\|^2|\mathcal{F}_k] \leq \|\Theta_k\|^2(1+\gamma_{k+1}^2\lambda_M^2) - 2\gamma_{k+1}\Theta_k^\top\widetilde{K}\Theta_k + 4\rho(C)^2\gamma_{k+1}^2\mathbb{E}[\|w_{k+1}\|^2|\mathcal{F}_k]\mathbb{1}_{\mathcal{A}_k}.$$

Since, using (5.6),

$$\sum_{k\geq 0}\gamma_{k+1}^2\mathbb{E}[\|w_{k+1}\|^2|\mathcal{F}_k]\mathbb{1}_{\mathcal{A}_k} \leq \left(\sup_{k\geq 0}\mathbb{E}[\|w_{k+1}\|^2|\mathcal{F}_k]\mathbb{1}_{\mathcal{A}_k}\right)\left(\sum_{k\geq 0}\gamma_{k+1}^2\right) < \infty,$$

we are in position to apply the Robbins-Siegmund Theorem 6.33 and we obtain the almost sure convergence of $\sum_k \gamma_{k+1}\Theta_k^\top\widetilde{K}\Theta_k$ and $\|\Theta_k\|_2^2 \to V_\infty$. Because $\widetilde{K}$ is positive definite, it gives that, with probability 1, $\sum_{k\geq 0}\gamma_{k+1}\|\Theta_k\|^2 < +\infty$, from which, we deduce $\liminf_k \|\Theta_k\|^2 = 0$. Therefore one can extract a subsequence Θ_k such that $\|\Theta_k\|^2 \to 0$. Using the above second condition yields $V_\infty = 0$ and we conclude that $\widetilde{\Delta}_k = H^{-1/2}\Theta_k \to 0$.

Define the difference

$$E_k = \Delta_k - \widetilde{\Delta}_k.$$

Since $\theta \mapsto \nabla^2 F(\theta)$ is continous at $\theta^\star$, we can apply a coordinate-wise mean value theorem. Indeed, for any $\theta \in \mathbb{R}^d$, we have $\nabla F(\theta) = (\partial_1 F(\theta), \ldots, \partial_d F(\theta))$ where for all $j = 1, \ldots, d$, the partial derivatives functions $\partial_j F : \mathbb{R}^d \to \mathbb{R}$ are Lipschitz continuous. Denote by $\nabla(\partial_j F) : \mathbb{R}^d \to \mathbb{R}^d$ the gradient of the partial derivative $\partial_j F$, *i.e.*, $\nabla(\partial_j F)(\theta) = (\partial_{1,j}^2 F(\theta), \ldots, \partial_{d,j}^2 F(\theta))$. For any $\theta, \eta \in B(\theta^\star, \varepsilon)$, there exists $\xi_j \in \mathbb{R}^d$ such that

$$\partial_j F(\theta) - \partial_j F(\eta) = \nabla(\partial_j F)(\xi_j)(\theta - \eta).$$

We construct a Hessian matrix by rows $H(\xi) = H(\xi_1, \ldots, \xi_d)$ where the j-th row is equal to $\nabla(\partial_j F)(\xi_j) = (\partial_{1,j}^2 F(\xi_j), \ldots, \partial_{d,j}^2 F(\xi_j))$

$$H(\xi) = \begin{bmatrix} \partial_{1,1}^2 F(\xi_1) & \ldots & \partial_{1,d}^2 F(\xi_1) \\ \vdots & \ddots & \vdots \\ \partial_{d,1}^2 F(\xi_d) & \ldots & \partial_{d,d}^2 F(\xi_d) \end{bmatrix}$$

and we can write

$$\nabla F(\theta) - \nabla F(\eta) = H(\xi)(\theta - \eta).$$

There exists $\xi_k = (\xi_k^{(1)}, \ldots, \xi_k^{(d)})$ with $\xi_k^{(j)} \in [\theta^\star + E_k, \theta_k]$ and $\xi_k' = (\xi_k'^{(1)}, \ldots, \xi_k'^{(d)})$ with $\xi_k'^{(j)} \in [\theta^\star + E_k, \theta^\star]$ such that

$$\nabla F(\theta^\star + E_k) - \nabla F(\theta_k) = -H(\xi_k)\widetilde{\Delta}_k \tag{5.15}$$

$$\nabla F(\theta^\star + E_k) = H(\xi_k')E_k. \tag{5.16}$$

Let $\eta > 0$ such that $2\alpha\lambda_m(1-3\eta) > 1$. This choice will come clear at the end of the reasoning. On the one hand, we have $C_k \to C$. On the other hand, using Lemma 5.28, the spectrum of C_kH is real and positive. Hence, we have the convergence of the eigenvalues of C_kH towards the eigenvalues of $K = CH$. This follows from the definition of eigenvalues as roots of the characteristic polynomial and the fact that the roots of any polynomial $P \in \mathbb{C}[X]$ are continuous functions of the coefficients (Zedek, 1965). Consequently, there exists $n_1(\omega)$ such that for all $k \geq n_1(\omega)$,

$$(1-\eta)\lambda_m \leq \lambda_{\min}(C_kH) \leq \lambda_{\max}(C_kH) \leq (1+\eta)\lambda_M. \tag{5.17}$$

We can define $n_2(\omega)$ such that for all $k \geq n_2(\omega)$

$$\mathcal{A}_k \text{ is realized.} \tag{5.18}$$

Since $\|\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d\| \to 0$ as $k \to \infty$, there is $n_3(\omega)$ and $n_4(\omega)$ such that for all $k \geq n_3(\omega)$

$$\|\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d\| \leq \frac{\eta}{1+\eta}\frac{\lambda_m}{\lambda_M}, \tag{5.19}$$

and for all $k \geq n_4(\omega)$,

$$\|\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}}\| \leq 1. \tag{5.20}$$

Since $\gamma_k \to 0$, there is n_5 such that for all $k \geq n_5$

$$\gamma_{k+1} \leq \frac{2\eta\lambda_m}{(1+\eta)^2\lambda_M^2}. \tag{5.21}$$

To use the previous local properties, define $n_0(\omega) = n_1(\omega) \vee n_2(\omega) \vee n_3(\omega) \vee n_4(\omega) \vee n_5$ and introduce the set $\mathcal{E}_j$ along with its complement $\mathcal{E}_j^c$, defined by

$$\mathcal{E}_j = \{\omega \,:\, j \geq n_0(\omega)\}.$$

Let $\delta > 0$ and take $j \geq 1$ large enough such that $\mathbb{P}(\mathcal{E}_j^c) \leq \delta$. Invoking the Markov inequality, we have for all $a > 0$

$$\begin{aligned}
\mathbb{P}(\gamma_k^{-1/2}\|E_k\| > a) &= \mathbb{P}(\gamma_k^{-1/2}\|E_k\| > a,\, \mathcal{E}_j) + \mathbb{P}(\gamma_k^{-1/2}\|E_k\| > a,\, \mathcal{E}_j^c) \\
&\leq \mathbb{P}(\gamma_k^{-1/2}\|E_k\| > a,\, \mathcal{E}_j) + \delta \\
&\leq \gamma_k^{-1/2}a^{-1}\mathbb{E}[\|E_k\|\mathbb{1}_{\mathcal{E}_j}] + \delta
\end{aligned}$$

Because δ is arbitrary, we only need to show that for any value of $j \geq 1$,

$$e_k := \mathbb{E}[\|E_k\|\mathbb{1}_{\mathcal{E}_j}] = o(\gamma_k^{1/2}).$$

To prove this fact, we shall recognize a stochastic algorithm for the sequence e_k.

Let $k \geq j$ and assume further that $\mathcal{E}_j$ is realized. We have, because of (5.18),

$$E_{k+1} = \Delta_k - \widetilde{\Delta}_k - \gamma_{k+1}C_k\nabla F(\theta_k) + \gamma_{k+1}K\widetilde{\Delta}_k.$$

Introducing $\widetilde{E}_k = \sqrt{H}E_k$, we find

$$\widetilde{E}_{k+1} = \widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta_k) + \gamma_{k+1}\sqrt{H}K\widetilde{\Delta}_k,$$

and using (5.15), it comes that

$$\begin{aligned}\widetilde{E}_{k+1} &= \widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta^\star + E_k) - \gamma_{k+1}\sqrt{H}C_kH(\xi_k)\widetilde{\Delta}_k + \gamma_{k+1}\sqrt{H}K\widetilde{\Delta}_k \\ &= \widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta^\star + E_k) + \gamma_{k+1}\sqrt{H}(K - C_kH(\xi_k))\widetilde{\Delta}_k.\end{aligned}$$

Using Minkowski inequality, we have

$$\|\widetilde{E}_{k+1}\| \leq \|\widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta^\star + E_k)\| + \|\gamma_{k+1}\sqrt{H}(K - C_kH(\xi_k))\widetilde{\Delta}_k\|.$$

We shall now focus on the first term. Still on the set $\mathcal{E}_j$, we have

$$\begin{aligned}&\|\widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta^\star + E_k)\|^2 \\ &= \|\widetilde{E}_k\|^2 - 2\gamma_{k+1}\langle\widetilde{E}_k, \sqrt{H}C_k\nabla F(\theta^\star + E_k)\rangle + \gamma_{k+1}^2\|\sqrt{H}C_k\nabla F(\theta^\star + E_k)\|^2 \qquad (5.22)\end{aligned}$$

We have on the one hand using (5.16)

$$\begin{aligned}\langle\widetilde{E}_k, \sqrt{H}C_k\nabla F(\theta^\star + E_k)\rangle &= \langle\widetilde{E}_k, \sqrt{H}C_kH(\xi_k')E_k\rangle \\ &= \langle\widetilde{E}_k, \sqrt{H}C_kHE_k\rangle + \langle\widetilde{E}_k, \sqrt{H}C_k(H(\xi_k') - H)E_k\rangle\end{aligned}$$

Due to (5.17), the first term satisfies

$$\begin{aligned}\langle\widetilde{E}_k, \sqrt{H}C_kHE_k\rangle &= \langle\widetilde{E}_k, \sqrt{H}C_k\sqrt{H}\widetilde{E}_k\rangle \\ &\geq \lambda_{\min}(C_kH)\|\widetilde{E}_k\|^2 \\ &\geq (1-\eta)\lambda_m\|\widetilde{E}_k\|^2\end{aligned}$$

The second term satisfies

$$\begin{aligned}\langle\widetilde{E}_k, \sqrt{H}C_k(H(\xi_k') - H)E_k\rangle &= \langle\widetilde{E}_k, \sqrt{H}C_k\sqrt{H}(\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d)\widetilde{E}_k\rangle \\ &\geq -\left|\langle\widetilde{E}_k, \sqrt{H}C_k\sqrt{H}(\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d)\widetilde{E}_k\rangle\right|\end{aligned}$$

Using Cauchy-Schwarz inequality, the submultiplicativity of the norm, (5.17) and (5.19), we have

$$\begin{aligned}&\left|\langle\widetilde{E}_k, \sqrt{H}C_k\sqrt{H}(\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d)\widetilde{E}_k\rangle\right| \\ &\leq \|\sqrt{H}C_k\sqrt{H}\|\|\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}} - I_d\|\|\widetilde{E}_k\|^2 \\ &\leq \eta\lambda_m\|\widetilde{E}_k\|^2.\end{aligned}$$

Finally, it follows that

$$\langle\widetilde{E}_k, \sqrt{H}C_k\nabla F(\theta^\star + E_k)\rangle \geq (1-2\eta)\lambda_m\|\widetilde{E}_k\|^2 \qquad (5.23)$$

On the other hand using (5.16), (5.17) and (5.20),

$$\begin{aligned}\|\sqrt{H}C_k\nabla F(\theta^\star + E_k)\|^2 &= \|\sqrt{H}C_kH(\xi_k')E_k\|^2 \\ &= \|\sqrt{H}C_k\sqrt{H}(\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}})\widetilde{E}_k\|^2 \\ &\leq \lambda_{\max}(C_kH)^2\|\sqrt{H^{-1}}H(\xi_k')\sqrt{H^{-1}}\|^2\|\widetilde{E}_k\|^2 \\ &\leq (1+\eta)^2\lambda_M^2\|\widetilde{E}_k\|^2 \qquad (5.24)\end{aligned}$$

Putting together (5.22), (5.23), (5.24) and using (5.21) gives that, on $\mathcal{E}_j$,

$$\begin{aligned}
&\|\widetilde{E}_k - \gamma_{k+1}\sqrt{H}C_k\nabla F(\theta^\star + E_k)\|^2 \\
&\leq \|\widetilde{E}_k\|^2(1 - 2\gamma_{k+1}(1-2\eta)\lambda_m + \gamma_{k+1}^2(1+\eta)^2\lambda_M^2) \\
&\leq \|\widetilde{E}_k\|^2(1 - 2\gamma_{k+1}(1-3\eta)\lambda_m).
\end{aligned}$$

By the Minkowski inequality and the fact that $(1-x)^{1/2} \leq 1 - x/2$, on $\mathcal{E}_j$, it holds

$$\begin{aligned}
\|\widetilde{E}_{k+1}\| &\leq \|\widetilde{E}_k\|(1 - 2\gamma_{k+1}(1-3\eta)\lambda_m)^{1/2} + \gamma_{k+1}\|\sqrt{H}(K - C_kH(\xi_k))\widetilde{\Delta}_k\| \\
&\leq \|\widetilde{E}_k\|(1 - \gamma_{k+1}(1-3\eta)\lambda_m) + \gamma_{k+1}\|\sqrt{H}\|\|(K - C_kH(\xi_k))\widetilde{\Delta}_k\|
\end{aligned}$$

Hence, we have shown that for any $k \geq j$,

$$\|\widetilde{E}_k\|\mathbb{1}_{\mathcal{E}_j} \leq \|\widetilde{E}_k\|\mathbb{1}_{\mathcal{E}_j}(1 - \gamma_{k+1}(1-3\eta)\lambda_m) + \gamma_{k+1}\|\sqrt{H}\|\|(K - C_kH(\xi_k))\mathbb{1}_{\mathcal{E}_j}\widetilde{\Delta}_k\|.$$

It follows that, for any $k \geq j$,

$$e_{k+1} \leq e_k(1 - \gamma_{k+1}(1-3\eta)\lambda_m) + \gamma_{k+1}\|\sqrt{H}\|\mathbb{E}[\|U_k\widetilde{\Delta}_k\|],$$

with $U_k = (K - C_kH(\xi_k))\mathbb{1}_{\mathcal{E}_j}$. Because with probability 1, $\|U_k\|$ is bounded, we can apply the Lebesgue dominated convergence theorem to obtain that $\varepsilon_k = \mathbb{E}[\|U_k\|^2] \to 0$. From the Cauchy-Schwarz inequality, we get

$$\mathbb{E}[\|U_k\widetilde{\Delta}_k\|] \leq \sqrt{\varepsilon_k}\sqrt{\mathbb{E}[\|\widetilde{\Delta}_k\|_2^2]}.$$

On the other hand, we have already shown in (5.12) that $\rho(\Sigma_k) = \|\Sigma_k\| \leq U_D\left(U_a + U_b\right)$. Since $\widetilde{\Delta}_k = \sqrt{H^{-1}}\Theta_k = \sqrt{H^{-1}}\sqrt{\gamma_k}(Y_k + M_k)$, we have

$$\mathbb{E}[\|\widetilde{\Delta}_k\|_2^2] \leq 2(\gamma_k/\lambda_m)(\|Y_k\|_2^2 + \mathbb{E}[\|M_k\|_2^2]),$$

where the last term is the leading term and satisfies

$$\mathbb{E}[\|M_k\|_2^2]] = \mathbb{E}[\mathrm{Tr}(\Sigma_k)] \leq d\mathbb{E}[\rho(\Sigma_k)].$$

Therefore, we have

$$\mathbb{E}[\|\widetilde{\Delta}_k\|_2^2] \leq \gamma_k A$$

for some $A > 0$. Consequently, for all $k \geq j$,

$$e_{k+1} \leq e_k(1 - \gamma_{k+1}(1-3\eta)\lambda_m) + \gamma_{k+1}^{3/2}A'\|\sqrt{H}\|\varepsilon_k^{1/2}.$$

The condition $2\alpha\lambda_m(1-3\eta) > 1$ ensures that we can apply Lemma 5.27 with $(m\lambda > p), m = 1, p = 1/2, \lambda = \alpha(1-3\eta)\lambda_m$. we finally get

$$\limsup_k (e_k/\gamma_k^{1/2}) = 0.$$

As a consequence, $e_k = o(\sqrt{\gamma_k})$, which concludes the proof.

Since $\gamma_k^{-1/2}\sqrt{H}\widetilde{\Delta}_k \to \mathcal{N}(0,\Sigma)$, we have $\gamma_k^{-1/2}\widetilde{\Delta}_k \to \mathcal{N}(0,\widetilde{\Sigma})$ where $\widetilde{\Sigma} = \sqrt{H^{-1}}\Sigma\sqrt{H^{-1}}$. Recall that Σ satisfies the Lyapunov equation

$$(\sqrt{H}C\sqrt{H} - \zeta I_d)\Sigma + \Sigma(\sqrt{H}C\sqrt{H} - \zeta I_d) = \sqrt{H}C\Gamma C\sqrt{H}.$$

Multiplying on the left and right sides by $\sqrt{H^{-1}}$, we get

$$C\sqrt{H}\Sigma\sqrt{H^{-1}} - \zeta\sqrt{H^{-1}}\Sigma\sqrt{H^{-1}} + \sqrt{H^{-1}}\Sigma\sqrt{H}C - \zeta\sqrt{H^{-1}}\Sigma\sqrt{H^{-1}} = C\Gamma C,$$

where we recognize the following Lyapunov equation

$$(CH - \zeta I_d)\widetilde{\Sigma} + \widetilde{\Sigma}(CH - \zeta I_d)^\top = C\Gamma C.$$

5.A.2 Additional propositions

This section gathers the proofs of Proposition 5.9 about the optimal choice for the *conditioning* matrix and of Proposition 5.19 about the almost sure convergence of the *conditioning* matrices.

Proposition 5.9. *The choice* $C^\star = H^{-1}$ *is optimal in the sense that* $\Sigma_{C^*} \preceq \Sigma_C$, $\forall C \in \mathcal{C}_H$. *Moreover,* $\Sigma_{C^\star} = H^{-1}\Gamma H^{-1}$.

Proof Define $\Delta_C = \Sigma_C - H^{-1}\Gamma H^{-1}$ and check that Δ_C satisfies

$$\left(CH - I_d/2\right)\Delta_C + \Delta_C\left(CH - I_d/2\right)^\top = (C - H^{-1})\Gamma(C - H^{-1}).$$

Because Γ is symmetric positive semi-definite, we have using Lemma 5.32 that the term on the right side is symmetric positive semi-definite. Therefore, in view of Proposition 5.33, we get that Δ_C is symmetric positive semi-definite $\Delta_C \succeq 0$ which implies $\Sigma_C \succeq H^{-1}\Gamma H^{-1}$ for all $C \in \mathcal{C}_H$. The equality is reached for $C^\star = H^{-1}$ with $\Delta_C = 0, \Sigma_{C^\star} = H^{-1}\Gamma H^{-1}$. ■

Proposition 5.19. *Let* $(\Phi_k)_{k\geq 0}$ *be obtained by* (5.4). *Suppose that Assumptions 5.4 and 5.18 are fulfilled and that* $\theta_k \to \theta^\star$ *almost surely . If* $\sup_{0\leq j\leq k}\omega_{j,k} = O(1/k)$, *then we have* $\Phi_k \to H = \nabla^2 F(\theta^\star)$ *almost surely.*

Proof We use the decomposition

$$\Phi_k - H = \sum_{j=0}^{k}\omega_{j,k}\left(\nabla^2 F(\theta_j) - H\right) + \sum_{j=0}^{k}\omega_{j,k}\left(H(\theta_j, \xi'_{j+1}) - \nabla^2 F(\theta_j)\right).$$

The continuity of $\nabla^2 F$ at $\theta^\star$ and the fact that $\theta_j \to \theta^\star$ a.s. implie that $\left\|\nabla^2 F(\theta_j) - H\right\| \to 0$ a.s. Since $\sup_{0\leq j\leq k}\omega_{j,k} = O(1/k)$, there exists $a > 0$ such that

$$\left\|\sum_{j=0}^{k}\omega_{j,k}\left(\nabla^2 F(\theta_j) - H\right)\right\| \leq \frac{a}{k+1}\sum_{j=0}^{k}\left\|\nabla^2 F(\theta_j) - H\right\|,$$

which goes to 0 in virtue of Cesaro's Lemma, therefore $\lim_{k\to\infty}\sum_{j=0}^{k}\omega_{j,k}\left(\nabla^2 F(\theta_j) - H\right) = 0$. The second term is a sum of martingale increments and shall be treated with Freedman inequality and Borel-Cantelli Lemma. Introduce the martingale increments

$$\forall 0 \leq j \leq k, \quad X_{j+1,k} = \omega_{j,k}\left(H(\theta_j, \xi'_{j+1}) - \nabla^2 F(\theta_j)\right).$$

For a fixed k, we have $X_{j+1,k} = \left(x_{j+1}^{(i,l)}\right)_{1\leq i,l\leq d}$ where we remove the index k for the sake of clarity. Because the Hessian generator is unbiased, we have for all coordinates

$$\mathbb{E}\left[x_{j+1}^{(i,l)}|\mathcal{F}_j\right] = 0 \quad \text{for all } 0 \leq j \leq k.$$

By definition of the Hessian generator and using that $(\nabla^2 F(\theta_j))$ is bounded, we get that $\left\|H(\theta_j, \xi'_{j+1}) - \nabla^2 F(\theta_j)\right\| = O(1)$ for all $j \geq 0$. For any $b > 0$, consider the following event

$$\Omega_b = \left\{ \sup_{k\geq 0} \max_{j=0,\dots,k} (k+1) \left| x_{j+1}^{(i,l)} \right| \leq b \right\},$$

and note that since $\omega_{j,k} = O(1/k)$ we have $\mathbb{P}(\Omega_b) \to 1$ as $b \to \infty$. On this event, the martingale increments and the variance term are bounded as

$$\max_{j=0,\dots,k} \left| x_{j+1}^{(i,l)} \right| \leq b(k+1)^{-1}, \quad \sum_{j=0}^{k} \mathbb{E}\left[\left(x_{j+1}^{(i,l)} \right)^2 \mid \mathcal{F}_j \right] \leq b^2 (k+1)^{-1}.$$

Using Freedman inequality (Theorem 5.31), we have for all coordinates $i, l = 1, \dots, d$,

$$\mathbb{P}\left(\left| \sum_{j=0}^{k} x_{j+1}^{(i,l)} \right| > \varepsilon, \Omega_b \right) \leq 2 \exp\left(-\frac{\varepsilon^2 (k+1)}{2b(b+\varepsilon)} \right).$$

The last term is the general term of a convergent series. Apply Borel-Cantelli Lemma (Borel, 1909) to finally get almost surely on Ω_b that $\lim_{k\to\infty} \sum_{j=0}^{k} x_{j+1}^{(i,l)} = 0$. Since $b > 0$ is arbitrary and $\mathbb{P}(\Omega_b) \to 1$ when $b \to \infty$, we have almost surely $\lim_{k\to\infty} \sum_{j=0}^{k} x_{j+1}^{(i,l)} = 0$. This is true for all the coordinates of the martingale increments and therefore

$$\lim_{k\to\infty} \sum_{j=0}^{k} \omega_{j,k} \left(H(\theta_j, \xi'_{j+1}) - \nabla^2 F(\theta_j) \right) = 0 \text{ a.s.}$$

■

5.A.3 Auxiliary results on expected smoothness

The following Lemma gives sufficient conditions to meet the weak growth condition on the stochastic noise as stated in Assumption 6.13.

Lemma 5.22. *Suppose that for all $k \geq 1, \theta \in \mathbb{R}^d, F(\theta) = \mathbb{E}\left[f(\theta, \xi_k) | \mathcal{F}_{k-1} \right]$ with $\xi_k \sim P_{k-1}$. Assume that for all $\xi_k \sim P_{k-1}$, the function $\theta \mapsto f(\theta, \xi_k)$ is L-smooth almost surely and there exists $m \in \mathbb{R}$ such that for all $\theta \in \mathbb{R}^d, f(\theta, \xi_k) \geq m$. Then a gradient estimate is given by $g(\theta, \xi) = \nabla f(\theta, \xi)$ and the growth condition of Assumption 6.13 is satisfied with $\sigma^2 = 2L(F^\star - m)$ and*

$$\forall \theta \in \mathbb{R}^d, \forall k \in \mathbb{N}, \quad \mathbb{E}\left[\|g(\theta, \xi_k)\|_2^2 | \mathcal{F}_{k-1} \right] \leq 2L \left(F(\theta) - F^\star \right) + \sigma^2.$$

Proof For all $\xi_k \sim P_{k-1}$, Lipschitz continuity of the gradient $\theta \mapsto \nabla f(\theta, \xi_k)$ implies (see Nesterov (2013))

$$f(y, \xi_k) \leq f(\theta, \xi_k) + \langle \nabla f(\theta, \xi_k), y - \theta \rangle + (L/2)\|y - \theta\|_2^2.$$

Plug $y = \theta - (1/L)\nabla f(\theta, \xi_k)$ and use the lower bound $f(y, \xi_k) \geq m$ to obtain

$$\frac{1}{2L}\|\nabla f(\theta, \xi_k)\|_2^2 \leq f(\theta, \xi_k) - f(y, \xi_k) \leq f(\theta, \xi_k) - m,$$

which gives,

$$\|g(\theta, \xi_k)\|_2^2 \leq 2L\left(f(\theta, \xi_k) - f(\theta^\star, \xi_k)\right) + 2L\left(f(\theta^\star, \xi_k) - m\right)$$

and conclude by taking the conditional expectation with respect to $\mathcal{F}_{k-1}$. ■

The next Lemma links our weak growth condition with the notion of expected smoothness as introduced in Gower et al. (2019). In particular, this notion can be extended to our general context where the sampling distribution can evolve through the stochastic algorithm.

Lemma 5.23. *(Expected smoothness) Assume that with probability one,*

$$\sup_{k\geq 1}\sup_{x\neq x^\star}\frac{\mathbb{E}\left[\|g(\theta, \xi_k) - g(\theta^\star, \xi_k)\|_2^2|\mathcal{F}_{k-1}\right]}{F(\theta) - F^\star} < \infty \quad \textit{and} \quad \sup_{k\geq 1}\mathbb{E}\left[\|g(\theta^\star, \xi_k)\|_2^2|\mathcal{F}_{k-1}\right] < \infty.$$

Then there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ *such that*

$$\forall \theta \in \mathbb{R}^d, \forall k \in \mathbb{N}, \quad \mathbb{E}\left[\|g(\theta, \xi_k)\|_2^2|\mathcal{F}_{k-1}\right] \leq 2\mathcal{L}\left(F(\theta) - F^\star\right) + 2\sigma^2.$$

Proof For all $\theta \in \mathbb{R}^d$ and all $k \in \mathbb{N}$, we have

$$\begin{aligned}\|g(\theta, \xi_k)\|_2^2 &= \|g(\theta, \xi_k) - g(\theta^\star, \xi_k) + g(\theta^\star, \xi_k)\|_2^2 \\ &\leq 2\|g(\theta, \xi_k) - g(\theta^\star, \xi_k)\|_2^2 + 2\|g(\theta^\star, \xi_k)\|_2^2.\end{aligned}$$

Using the expected smoothness, with probability one, there exists $0 \leq \mathcal{L} < \infty$ such that

$$\mathbb{E}\left[\|g(\theta, \xi_k) - g(\theta^\star, \xi_k)\|_2^2|\mathcal{F}_{k-1}\right] \leq \mathcal{L}\left(F(\theta) - F^\star\right).$$

Since the noise at optimal point is almost surely finite there exists $0 \leq \sigma^2 < \infty$ such that

$$\mathbb{E}\left[\|g(\theta^\star, \xi_k)\|_2^2|\mathcal{F}_{k-1}\right] \leq \sigma^2,$$

which allows to conclude by taking the conditional expectation. ■

5.A.4 Proof of the almost sure convergence (Theorem 5.16)

The idea behind the proof of the almost sure convergence is to apply the Robbins-Siegmund Theorem (Theorem 6.33) (which can be found in Section 5.B) in combination with the following key deterministic result.

Lemma 5.24 (Deterministic result)**.** *Let* $F : \mathbb{R}^d \to \mathbb{R}$ *be a* L*-smooth function and* (θ_t) *a random sequence obtained by the SGD update rule* $\theta_{t+1} = \theta_t - \gamma_{t+1}C_t g_t$ *where* $(\gamma)_{t\geq 1}$ *a positive sequence of learning rates and* $C_{t-1} \preceq \nu_t I_d$ *are such that* $\sum_t \gamma_t\nu_t = \infty$*. Let* $\omega \in \Omega$ *such that the following limits exist:*

$$(i)\ \sum_{t\geq 0}\gamma_{t+1}\nu_{t+1}\|\nabla F(\theta_t(\omega))\|_2^2 < \infty \quad (ii)\ \sum_{t\geq 1}\gamma_t C_{t-1}(g_{t-1}(\omega) - \nabla F(\theta_{t-1}(\omega))) < \infty$$

then $\nabla F(\theta_t(\omega)) \to 0$ *as* $t \to \infty$*.*

Proof. The proof (and in particular the reasoning by contradiction) is inspired from the proof of Proposition 1 in Bertsekas and Tsitsiklis (2000). For ease of notation we omit the ω in the proof. Note that condition (i) along with $\sum_t \gamma_t \nu_t = \infty$ implie that $\liminf_t \|\nabla F(\theta_t)\| = 0$. Now, by contradiction, let $\varepsilon > 0$ and assume that

$$\limsup_t \|\nabla F(\theta_t)\| > \varepsilon$$

We have that there is infinitely many t such that $\|\nabla F(\theta_t)\| < \varepsilon/2$ and also infinitely many t such that $\|\nabla F(\theta_t)\| > \varepsilon$. It follows that there is infinitely many *crossings* between the sets $\{t \in \mathbb{N} : \|\nabla F(\theta_t)\| < \varepsilon/2\}$ and $\{t \in \mathbb{N} : \|\nabla F(\theta_t)\| > \varepsilon\}$. A *crossing* is a collection of indexes $I_k = \{L_k, L_k + 1, \dots, U_k - 1\}$ with $L_k \leq U_k$ ($I_k = \emptyset$ when $L_k = U_k$) such that for all $t \in I_k$,

$$\|\nabla F(\theta_{L_k - 1})\| < \varepsilon/2 \leq \|\nabla F(\theta_t)\| \leq \varepsilon < \|\nabla F(\theta_{U_k})\|.$$

Define the following partial Cauchy sequence $R_k = \sum_{t=L_k}^{U_k} \gamma_t (g_{t-1} - \nabla F(\theta_{t-1}))$ and note that condition (ii) implies that $R_k \to 0$ as $k \to \infty$. For all $k \geq 1$,

$$\begin{aligned} \varepsilon/2 &\leq \|\nabla F(\theta_{U_k})\|_2 - \|\nabla F(\theta_{L_k - 1})\|_2 \\ &\leq \|\nabla F(\theta_{U_k}) - \nabla F(\theta_{L_k - 1})\|_2 \\ &\leq L \|\theta_{U_k} - \theta_{L_k - 1}\|_2, \end{aligned}$$

where we use that ∇F is L-Lipschitz. Then using the update rule $\theta_t - \theta_{t-1} = -\gamma_t C_{t-1} g_{t-1}$, we have by sum

$$\begin{aligned} \varepsilon/2 &\leq L \Big\| \sum_{t=L_k}^{U_k} \theta_t - \theta_{t-1} \Big\|_2 = L \Big\| \sum_{t=L_k}^{U_k} \gamma_t C_{t-1} g_{t-1} \Big\|_2 \\ &\leq L \Big\| \sum_{t=L_k}^{U_k} \gamma_t C_{t-1} \nabla F(\theta_{t-1}) \Big\|_2 + L \Big\| \sum_{t=L_k}^{U_k} \gamma_t C_{t-1} (g_{t-1} - \nabla F(\theta_{t-1})) \Big\|_2 \\ &\leq L \sum_{t=L_k}^{U_k} \gamma_t \nu_t \|\nabla F(\theta_{t-1})\|_2 + L \|R_k\|_2 \end{aligned}$$

Since in the previous equation $\|\nabla F(\theta_{t-1})\|_2 > \varepsilon/2$, we get

$$(\varepsilon/2)^2 \leq L \sum_{t=L_k}^{U_k} \gamma_t \nu_t \|\nabla F(\theta_{t-1})\|_2^2 + (\varepsilon/2) L \|R_k\|_2$$

But since $\sum_{t \geq 0} \gamma_{t+1} \nu_{t+1} \|\nabla F(\theta_t)\|^2$ is finite and $\lim_k R_k = 0$, the previous upper bound goes to 0 and implies a contradiction. □

It remains to show that points (i) and (ii) in Lemma 5.24 are valid with probability one. Since $\theta \mapsto F(\theta)$ is L-smooth, we have the quadratic bound (see Nesterov (2013))

$$\forall \theta, \eta \in \mathbb{R}^d \quad F(\eta) \leq F(\theta) + \langle \nabla F(\theta), \eta - \theta \rangle + \frac{L}{2} \|\eta - \theta\|_2^2.$$

Using the update rule $\theta_{k+1} = \theta_k - \gamma_{k+1} C_k g(\theta_k, \xi_{k+1})$, we get

$$\begin{aligned} F(\theta_{k+1}) &\leq F(\theta_k) + \langle \nabla F(\theta_k), \theta_{k+1} - \theta_k \rangle + \frac{L}{2} \|\theta_{k+1} - \theta_k\|_2^2 \\ &= F(\theta_k) - \gamma_{k+1} \langle \nabla F(\theta_k), C_k g(\theta_k, \xi_{k+1}) \rangle + \frac{L}{2} \gamma_{k+1}^2 \|C_k g(\theta_k, \xi_{k+1})\|_2^2. \end{aligned}$$

The last term can be upper bounded using the matrix norm and Assumption 5.15 as

$$\|C_k g(\theta_k, \xi_{k+1})\|_2^2 \leq \|C_k\|^2 \|g(\theta_k, \xi_{k+1})\|_2^2 \leq \nu_{k+1}^2 \|g(\theta_k, \xi_{k+1})\|_2^2,$$

and we have the inequality

$$F(\theta_{k+1}) \leq F(\theta_k) - \gamma_{k+1} \langle \nabla F(\theta_k), C_k g(\theta_k, \xi_{k+1}) \rangle + \frac{L}{2} (\gamma_{k+1} \nu_{k+1})^2 \|g(\theta_k, \xi_{k+1})\|_2^2.$$

Introduce $u_k = \gamma_k \nu_k$ and $v_k = \gamma_k \mu_k$, we have $\sum_{k \geq 1} v_k = +\infty$ and $\sum_{k \geq 1} u_k^2 < +\infty$ a.s. in virtue of Assumption 5.15. The random variables $F(\theta_k), C_k$ are $\mathcal{F}_k$-measurable and the gradient estimate is unbiased with respect to $\mathcal{F}_k$. Taking the conditional expectation denoted by $\mathbb{E}_k$ leads to

$$\begin{aligned}
\mathbb{E}_k \left[F(\theta_{k+1}) \right] - F(\theta_k) &\leq -\gamma_{k+1} \langle \nabla F(\theta_k), \mathbb{E}_k \left[C_k g(\theta_k, \xi_{k+1}) \right] \rangle + \frac{L}{2} u_{k+1}^2 \mathbb{E}_k \left[\|g(\theta_k, \xi_{k+1})\|_2^2 \right] \\
&= -\gamma_{k+1} \nabla F(\theta_k)^\top C_k \nabla F(\theta_k) + \frac{L}{2} u_{k+1}^2 \mathbb{E}_k \left[\|g(\theta_k, \xi_{k+1})\|_2^2 \right].
\end{aligned}$$

On the one hand for the first term, using Assumption 5.15 ,

$$\nabla F(\theta_k)^\top C_k \nabla F(\theta_k) \geq \lambda_{\min}(C_k) \|\nabla F(\theta_k)\|_2^2 \geq \mu_{k+1} \|\nabla F(\theta_k)\|_2^2.$$

On the other hand, using Assumption 6.13, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ such that almost surely

$$\forall k \in \mathbb{N}, \quad \mathbb{E}_k \left[\|g(\theta_k, \xi_{k+1})\|_2^2 \right] \leq 2\mathcal{L} \left(F(\theta_k) - F^\star \right) + \sigma^2.$$

Inject these bounds in the previous inequality and substract $F(\theta^\star)$ on both sides to have

$$\mathbb{E}_k \left[F(\theta_{k+1}) - F^\star \right] \leq (1 + L\mathcal{L} u_{k+1}^2)(F(\theta_k) - F^\star) - v_{k+1} \|\nabla F(\theta_k)\|_2^2 + (L/2) u_{k+1}^2 \sigma^2.$$

Introduce $V_k = F(\theta_k) - F^\star, W_k = v_{k+1} \|\nabla F(\theta_k)\|_2^2, a_k = L\mathcal{L} u_{k+1}^2$ and $b_k = (L/2) u_{k+1}^2 \sigma^2$. These four random sequences are non-negative $\mathcal{F}_k$-measurable sequences with $\sum_k a_k < \infty$ and $\sum_k b_k < \infty$ almost surely. Moreover we have

$$\forall k \in \mathbb{N}, \quad \mathbb{E} \left[V_{k+1} | \mathcal{F}_k \right] \leq (1 + a_k) V_k - W_k + b_k.$$

We can apply Robbins-Siegmund Theorem 6.33 to have

$$(a) \ \sum_{k \geq 0} W_k < \infty \ a.s. \qquad (b) \ V_k \xrightarrow{a.s.} V_\infty, \mathbb{E} \left[V_\infty \right] < \infty. \qquad (c) \ \sup_{k \geq 0} \mathbb{E} \left[V_k \right] < \infty.$$

Therefore we have the almost sure convergence of the series $\sum v_{k+1} \|\nabla F(\theta_k)\|_2^2$ which, given that $\limsup_k \nu_k / \mu_k$ exists, implies that $\sum u_{k+1} \|\nabla F(\theta_k)\|_2^2$ is finite. Hence we obtain (i) in Lemma 5.24. We now show that (ii) in Lemma 5.24 is also valid. The term of interest is a sum of martingale increments. The quadratic variation is given by

$$\begin{aligned}
\sum_{t \geq 1} \gamma_t^2 \mathbb{E}_t [\|C_{t-1}(g_{t-1}(\omega) - \nabla f(\theta_{t-1}(\omega)))\|_2^2] &\leq \sum_{t \geq 1} \gamma_t^2 \nu_t^2 \mathbb{E}_t [\|(g_{t-1}(\omega) - \nabla F(\theta_{t-1}(\omega)))\|_2^2] \\
&\leq \sum_{t \geq 1} \gamma_t^2 \nu_t^2 \mathbb{E}_t [\|g_{t-1}(\omega)\|_2^2] \\
&\leq \sum_{t \geq 1} \gamma_t^2 \nu_t^2 (2\mathcal{L}(F(\theta_{t-1}) - F^\star) + \sigma^2).
\end{aligned}$$

Now we can use that $V_k = F(\theta_k) - F^\star \xrightarrow{a.s.} V_\infty$ (which was deduced from Robbins-Siegmund Theorem) to obtain that the previous series converges. Invoking Theorem 2.17 in Hall and Heyde (1980), we obtain (ii) in Lemma 5.24. Furthermore we can prove that $\theta_{k+1} - \theta_k \to 0$ almost surely and in L^2. Indeed, we have

$$\mathbb{E}\left[\|\theta_{k+1} - \theta_k\|_2^2\right] = \mathbb{E}\left[\|\gamma_{k+1} C_k g(\theta_k, \xi_{k+1}\|_2^2\right] \leq u_{k+1}^2 \left(2\mathcal{L}\left(F(\theta_k) - F^\star\right) + \sigma^2\right).$$

In virtue of the almost sure convergence of $V_k = F(\theta_k) - F^\star$, the last term in parenthesis is upper bounded by a constant so that in view of the convergence of $\sum u_{k+1}^2$, we have the convergence of the series $\sum \mathbb{E}\left[\|\theta_{k+1} - \theta_k\|_2^2\right]$. We then deduce that $\mathbb{E}\left[\|\theta_{k+1} - \theta_k\|_2^2\right] \to 0$ and $\sum \left[\|\theta_{k+1} - \theta_k\|_2^2\right] < +\infty$ almost surely. In particular, $\theta_{k+1} - \theta_k \to 0$ in L^2 and almost surely. The last point follows from the fact that, for every $\delta > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{k\geq n} \|\theta_{k+1} - \theta_k\| \geq \delta\right) \leq \delta^{-2} \lim_{n\to\infty} \sum_{k\geq n} \mathbb{E}\left[\|\theta_{k+1} - \theta_k\|_2^2\right] = 0.$$

5.A.5 Proof of Corollary 5.17

First observe that since F is coercive, the convergence of $(F(\theta_k))$ obtained by Robbins-Siegmund theorem implies that the sequence of iterates $(\theta_k)_{k\geq 0}$ remains in a compact subset $\mathcal{K} \subset \mathbb{R}^d$. Let $\varepsilon > 0$. Since $\theta \mapsto d(\theta, \mathcal{S})$ is continuous, the set $\mathcal{D}(\varepsilon) = \{\theta \in \mathbb{R}^d : d(\theta, \mathcal{S}) \geq \varepsilon\}$ is closed and the set $\mathcal{K}(\varepsilon) = \mathcal{K} \cap \mathcal{D}(\varepsilon)$ is compact. On this set, the map $\theta \mapsto \|\nabla F(\theta)\|_2$ is stricly positive and there exists $\eta_\varepsilon > 0$ such that: $\theta \in \mathcal{K}(\varepsilon) \Rightarrow \|\nabla F(\theta)\|_2 > \eta_\varepsilon$. Thus, $\mathbb{P}(\theta \in \mathcal{K}(\varepsilon)) \leq \mathbb{P}(\|\nabla F(\theta)\|_2 > \eta_\varepsilon)$ and this last quantity goes to zero which proves the convergence in probability $d(\theta_k, \mathcal{S}) \to 0$. Actually the almost sure convergence $\nabla F(\theta_k) \to 0$ implies the convergence of the distances. Define $A_k(\varepsilon) = \{\omega : \theta_k(\omega) \in \mathcal{K}(\varepsilon)\}$ and $B_k(\varepsilon) = \{\omega : \|\nabla F(\theta_k(\omega))\|_2 > \eta_\varepsilon\}$. We have $A_k(\varepsilon) \subset B_k(\varepsilon)$ then $\cup_{n\geq 1} \cap_{k\geq n} A_k(\varepsilon) \subset \cup_{n\geq 1} \cap_{k\geq n} B_k(\varepsilon)$. Conclude by using the almost sure convergence $\mathbb{P}(\cup_{n\geq 1} \cap_{k\geq n} B_k(\varepsilon)) = 0$ for each $\varepsilon > 0$. If $\mathcal{S}$ is finite, it is in particular a compact set so the distance is attained for every $k \geq 0$, $d(\theta_k, \mathcal{S}) = \min_{s\in\mathcal{S}} d(\theta_k, s) \to 0$. Since $\theta_{k+1} - \theta_k \to 0$, the sequence of iterates can only converge to a single point of $\mathcal{S}$.

5.B Auxiliary results

5.B.1 Robbins-Siegmund Theorem

Theorem 5.25. *(Robbins and Siegmund (1971)) Consider a filtration $\left(\mathcal{F}_n\right)_{n\geq 0}$ and four sequences of random variables $\left(V_n\right)_{n\geq 0}, \left(W_n\right)_{n\geq 0}, \left(a_n\right)_{n\geq 0}$ and $\left(b_n\right)_{n\geq 0}$ that are adapted and non-negative. Assume that almost surely $\sum_k a_k < \infty$ and $\sum_k b_k < \infty$. Assume moreover that $\mathbb{E}\left[V_0\right] < \infty$ and for all $n \in \mathbb{N}$, $\mathbb{E}[V_{n+1}|\mathcal{F}_n] \leq (1 + a_n)V_n - W_n + b_n$. Then it holds*

$$(a)\ \sum_k W_k < \infty\ a.s. \qquad (b)\ V_n \xrightarrow{a.s.} V_\infty, \mathbb{E}\left[V_\infty\right] < \infty. \qquad (c)\ \sup_{n\geq 0} \mathbb{E}\left[V_n\right] < \infty.$$

5.B.2 Auxiliary lemmas

Lemma 5.26. *Let $(u_n)_{n\geq 1}, (v_n)_{n\geq 1}$ and $(\gamma_n)_{n\geq 1}$ be non-negative sequences such that $\gamma_n \to 0$ and $\sum_n \gamma_n = +\infty$. Assume that there exists a real number $m \geq 1$ and $j \geq 1$ such that for all $n \geq j$, $\quad u_n \leq (1-\gamma_n)^m u_{n-1} + \gamma_n v_n$. Then it holds that $\limsup_{n\to+\infty} u_n \leq \limsup_{n\to+\infty} v_n$.*

Proof Denote $x_+ = \max(x, 0)$. One has $(x+y)_+ \leq x_+ + y_+$. Set $\varepsilon > 0$ and $v = \limsup_n v_n + \varepsilon$. Then there exists an integer $N \geq 1$ such that $(1-\gamma_n)^m \leq (1-\gamma_n)$ and $v_n < v$, i.e., $(v_n - v)_+ = 0$ for $n \geq N$. We have for large enough $n \geq N \vee j$,

$$u_n - v \leq (1-\gamma_n)(u_{n-1} - v) + \gamma_n(v_n - v),$$

and taking the positive part gives

$$(u_n - v)_+ \leq (1-\gamma_n)(u_{n-1} - v)_+ + \gamma_n(v_n - v)_+ = (1-\gamma_n)(u_{n-1} - v)_+.$$

Since $\sum_n \gamma_n = +\infty$, this inequality implies that $(u_n - v)_+$ tends to zero, but this is true for all $\varepsilon > 0$ so v is arbitrarily close to $\limsup_n v_n$ and the result follows. ■

Lemma 5.27. *Let $(\gamma_n)_{n\geq 1}$ be a non-negative sequence converging to zero, and λ, m and p three real numbers with $\lambda > 0, m \geq 1, p \geq 0$. Consider two non-negative sequences $(x_n), (\varepsilon_n)$ and an integer $j \geq 1$ such that*

$$\forall n \geq j, \quad x_n = (1-\lambda\gamma_n)^m x_{n-1} + \gamma_n^{p+1}\varepsilon_n,$$
$$i.e., \quad x_n = \prod_{i=j}^{n}(1-\lambda\gamma_i)^m x_{j-1} + \sum_{k=j}^{n}\gamma_k^{p+1}\left(\prod_{i=k+1}^{n}(1-\lambda\gamma_i)^m\right)\varepsilon_k.$$

The following holds

- *if $\gamma_n = n^{-\beta}, \beta \in (1/2, 1)$, then for any p*

$$\limsup_{n\to+\infty} \frac{x_n}{\gamma_n^p} \leq \frac{1}{m\lambda}\limsup_{n\to+\infty}\varepsilon_n.$$

- *if $\gamma_n = 1/n$, then for any $p < m\lambda$*

$$\limsup_{n\to+\infty} \frac{x_n}{\gamma_n^p} \leq \frac{1}{m\lambda - p}\limsup_{n\to+\infty}\varepsilon_n.$$

In particular, when $\varepsilon_n \to 0$ with $j = 1$ and $x_0 = 0$,

$$\lim_{n\to+\infty}\sum_{k=1}^{n}\gamma_k\prod_{i=k+1}^{n}(1-\lambda\gamma_i)^m\varepsilon_k = 0,$$
$$(m\lambda > 1)\quad \lim_{n\to+\infty}\frac{1}{\gamma_n}\sum_{k=1}^{n}\gamma_k^2\prod_{i=k+1}^{n}(1-\lambda\gamma_i)^m\varepsilon_k = 0.$$

Before proving this result, note that if we consider $\gamma_n = \gamma/n^\beta$ then we can write

$$x_n = (1-\lambda\gamma_n)^m x_{n-1} + \gamma_n^{p+1}\varepsilon_n = (1-(\lambda\gamma)n^{-\beta})^m x_{n-1} + (n^{-\beta})^{p+1}(\gamma^{p+1}\varepsilon_n)$$

and apply the result with $\tilde{\lambda} = \gamma\lambda$ and $\tilde{\varepsilon}_n = \gamma^{p+1}\varepsilon_n$.

Proof We apply Lemma 5.26 to the sequence $u_n = \frac{x_n}{\gamma_n^p}$. We have for all $n \geq j$,

$$\begin{aligned} u_n &= \frac{1}{\gamma_n^p}\Big((1-\lambda\gamma_n)^m x_{n-1} + \gamma_n^{p+1}\varepsilon_n\Big) \\ &= \left(\frac{\gamma_{n-1}}{\gamma_n}\right)^p (1-\lambda\gamma_n)^m u_{n-1} + \gamma_n\varepsilon_n \\ &= \exp\left(p\log\left(\frac{\gamma_{n-1}}{\gamma_n}\right) + m\log(1-\lambda\gamma_n)\right) u_{n-1} + \gamma_n\varepsilon_n. \end{aligned}$$

Define

$$\lambda_n = \frac{1}{\gamma_n}\left(1 - \exp\left(p\log\left(\frac{\gamma_{n-1}}{\gamma_n}\right) + m\log(1-\lambda\gamma_n)\right)\right),$$

so we get the recursion equation

$$\forall n \geq j, \quad u_n = (1-\lambda_n\gamma_n)u_{n-1} + \lambda_n\gamma_n\frac{\varepsilon_n}{\lambda_n}.$$

• if $\gamma_n = n^{-\beta}, \beta \in (1/2, 1)$ then $1/\gamma_n - 1/\gamma_{n-1} \to 0$ and the ratio γ_{n-1}/γ_n tends to 1 with

$$\log\left(\frac{\gamma_{n-1}}{\gamma_n}\right) = \left(\frac{\gamma_{n-1}}{\gamma_n} - 1\right)\Big(1 + o(1)\Big) = \gamma_{n-1}\left(\frac{1}{\gamma_n} - \frac{1}{\gamma_{n-1}}\right)\Big(1 + o(1)\Big) = o(\gamma_n).$$

Besides, $m\log(1-\lambda\gamma_n) = -m\lambda\gamma_n + o(\gamma_n)$ when $n \to +\infty$ and we get

$$\lambda_n = \frac{1}{\gamma_n}\Big[1 - \exp\Big(-m\lambda\gamma_n + o(\gamma_n)\Big)\Big],$$

which implies that λ_n converges to $m\lambda$. We conclude with Lemma 5.26.
• if $\gamma_n = 1/n$ then the ratio γ_{n-1}/γ_n tends to 1 with

$$\log\left(\frac{\gamma_{n-1}}{\gamma_n}\right) = \log\left(1 + \frac{1}{n-1}\right) = \gamma_n + o(\gamma_n).$$

We still have $m\log(1-\lambda\gamma_n) = -m\lambda\gamma_n + o(\gamma_n)$ when $n \to +\infty$ and therefore

$$\lambda_n = \frac{1}{\gamma_n}\Big[1 - \exp\Big((p - m\lambda)\gamma_n + o(\gamma_n)\Big)\Big],$$

which implies λ_n converges to $(m\lambda - p)$ and we conclude in the same way. ■

Lemma 5.28. *Let $A, B \in \mathcal{S}_d^{++}(\mathbb{R})$ then the eigenvalues of AB are real and positive with $Sp(AB) \subset [\lambda_{\min}(A)\lambda_{\min}(B); \lambda_{\max}(A)\lambda_{\max}(B)]$.*

Proof Denote by $\sqrt{B}$ the unique positive square root of B. The matrix AB is similar to the real symmetric positive definite matrix $\sqrt{B}A\sqrt{B}$. Therefore its eigenvalues are real and positive. Since $A \mapsto \lambda_{\max}(A)$ is a sub-multiplicative matrix norm on $\mathcal{S}_d^{++}(\mathbb{R})$, $\lambda_{\max}(AB) \leq \lambda_{\max}(A)\lambda_{\max}(B)$ which gives $\lambda_{\max}(\left(AB\right)^{-1}) \leq \lambda_{\max}(A^{-1})\lambda_{\max}(B^{-1})$, *i.e.,* $\lambda_{\min}\left(AB\right)^{-1} \leq \lambda_{\min}(A)^{-1}\lambda_{\min}(B)^{-1}$, and finally $\lambda_{\min}(A)\lambda_{\min}(B) \leq \lambda_{\min}\left(AB\right)$. ■

Lemma 5.29. *Let $S \in \mathcal{S}_d^{++}(\mathbb{R})$ be a real symmetric positive definite matrix. Let $(\gamma_k)_{k\geq 1}$ be a positive decreasing sequence converging to 0 such that $\sum_k \gamma_k = +\infty$. Denote by λ_m the smallest eigenvalue of S. It holds that there exists $j \geq 1$ such that for any $k > j$, all the eigenvalues of the real symmetric matrix $A_k = I - \gamma_k S$ are positive and we have*

$$\rho(\Pi_n) = \rho(A_n \dots A_1) \overset{n\to+\infty}{\longrightarrow} 0,$$
$$\forall k > j, \quad \rho(\Pi_{n,k}) = \rho(A_n \dots A_{k+1}) \leq \prod_{i=k+1}^{n} (1 - \gamma_i \lambda_m).$$

Proof For any $k \in \mathbb{N}$, the eigenvalues of the real symmetric matrix $A_k = I - \gamma_k S$ are given by $Sp(A_k) = \{(1 - \gamma_k\lambda), \lambda \in Sp(S)\}$. Since $\gamma_k \to 0$, there exists $j \geq 1$ such that $\gamma_k\lambda_m < 1$ for all $k > j$. Therefore for any $k > j$, we have $Sp(A_k) \subset \mathbb{R}_+^\star$ and the largest eigenvalue is $\rho(A_k) = 1 - \gamma_k\lambda_m$. Since ρ is a sub-multiplicative norm for real symmetric matrices, we get $\rho(\Pi_n) \leq \prod_{k=1}^n \rho(A_k) = \prod_{k=1}^j \rho(A_k) \prod_{k=j+1}^n \rho(A_k)$. The second product can be upper bounded with the convexity of exponential,

$$\prod_{k=j+1}^{n} \rho(A_k) = \prod_{k=j+1}^{n} (1 - \gamma_k\lambda_m) \leq \prod_{k=j+1}^{n} \exp\left(-\gamma_k\lambda_m\right) = \exp\left(-\lambda_m(\tau_n - \tau_j)\right) \overset{n\to+\infty}{\longrightarrow} 0.$$

Similarly we have for all $k > j, \rho(\Pi_{n,k}) \leq \prod_{i=k+1}^n \rho(A_i) \leq \prod_{i=k+1}^n (1 - \gamma_i\lambda_m)$. ■

Lemma 5.30. *Let $\gamma_n = \alpha n^{-\beta}$ with $\beta \in (1/2, 1]$ then it holds*

$$(\beta < 1) \sum_{k=1}^{n} \gamma_k \sim \frac{n\gamma_n}{1-\beta} = \frac{\alpha}{1-\beta} n^{1-\beta}, \quad (\beta = 1) \sum_{k=1}^{n} \gamma_k \sim \alpha \log(n).$$

Proof By series-integral comprison, $\int_1^{n+1} t^{-\beta} dt \leq \sum_{k=1}^n k^{-\beta} \leq 1 + \int_1^n t^{-\beta} dt$. ■

Theorem 5.31. *(Delyon and Portier, 2021, Theorem 17)(Freedman inequality) Let $(X_j)_{1\leq j\leq n}$ be random variables such that $\mathbb{E}[X_j|\mathcal{F}_{j-1}] = 0$ for all $1 \leq j \leq n$ then, for all $t \geq 0$ and $v, m > 0$,*

$$\mathbb{P}\left(\left|\sum_{j=1}^{n} X_j\right| \geq t, \max_{j=1,\dots,n} |X_j| \leq m, \sum_{j=1}^{n} \mathbb{E}\left[X_j^2 \mid \mathcal{F}_{j-1}\right] \leq v\right) \leq 2\exp\left(-\frac{t^2/2}{v + tm/3}\right).$$

Lemma 5.32. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite matrix. Then for any $B \in \mathbb{R}^{m\times n}$, the matrix $BAB^\top \in \mathbb{R}^{m\times m}$ is symmetric positive semi-definite.*

Proof First note that $(BAB^\top)^\top = (B^\top)^\top A^\top B^\top = BAB^\top$ because A is symmetric. Then for any vector $x \in \mathbb{R}$, we have $x^\top(BAB^\top)x = (B^\top x)^\top A(B^\top x) \geq 0$ since A is positive semi-definite. ■

Proposition 5.33. *(Khalil, 2002, Theorem 4.6) Let H be a positive definite matrix and Γ a symmetric positive definite matrix of same dimension. Then there exists a symmetric positive definite matrix Σ, unique solution of the Lyapunov equation $H\Sigma + \Sigma H^\top = \Gamma$, which is given by $\Sigma = \int_0^{+\infty} e^{-tH}\Gamma e^{-tH^\top} \mathrm{d}t$.*

The results remains true if the matrix Γ is only symmetric positive semi-definite: in that case the matrix Σ is also symmetric positive semi-definite and is the solution of the Lyapunov equation.

5.B.3 Numerical illustration details

Consider the empirical risk minimization framework applied to Generalized Linear Models. Given a data matrix $X = (x_{i,j}) \in \mathbb{R}^{n\times d}$ with labels $y \in \mathbb{R}^n$ and a regularization parameter $\lambda > 0$, we are interested in solving $\min_{\theta\in\mathbb{R}^d} F(\theta)$ with

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta), \quad f_i(\theta) = \mathcal{L}(x_i^\top\theta, y_i) + \lambda\Omega(\theta),$$

$\mathcal{L} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is smooth loss function and $\Omega : \mathbb{R}^d \to \mathbb{R}_+$ is a smooth convex regularizer chosen as Tikhonov regularization $\Omega(\theta) = \frac{1}{2}\|\theta\|_2^2$. The gradient and Hessian of each component f_i are given for all $i = 1, \dots, n$ by

$$\nabla f_i(\theta) = \mathcal{L}'(x_i^\top\theta, y_i)x_i + \lambda\theta$$
$$\nabla^2 f_i(\theta) = \mathcal{L}''(x_i^\top\theta, y_i)x_i x_i^\top + \lambda I_d,$$

where $\mathcal{L}'(\cdot,\cdot)$ and $\mathcal{L}''(\cdot,\cdot)$ are the first and second derivative of $\mathcal{L}(\cdot,\cdot)$ with respect to the first argument. Consider two well-known losses, namely least-squares and logistic. These losses are respectively associated to the Ridge regression problem with $y \in \mathbb{R}^n$ and the binary classication task with $y \in \{-1, +1\}^n$. The regularization parameter is set to the classical value $\lambda = 1/n$. Denote by $\sigma(z) = 1/(1 + \exp(-z))$ the sigmoid function, we have the following closed-form equations

(Ridge Regression)

$$\begin{cases} \mathcal{L}(x_i^\top\theta, y_i) &= \frac{1}{2}(y_i - x_i^\top\theta)^2 \\ \mathcal{L}'(x_i^\top\theta, y_i) &= x_i^\top\theta - y_i \\ \mathcal{L}''(x_i^\top\theta, y_i) &= 1 \end{cases}$$

(Logistic Regression)

$$\begin{cases} \mathcal{L}(x_i^\top\theta, y_i) &= \log(1 + \exp(-y_i x_i^\top\theta)) \\ \mathcal{L}'(x_i^\top\theta, y_i) &= \sigma(x_i^\top\theta) - y_i \\ \mathcal{L}''(x_i^\top\theta, y_i) &= \sigma(x_i^\top\theta)(1 - \sigma(x_i^\top\theta)) \end{cases}$$

For the sake of completeness and illustrative purposes, we compare the performance of classical stochastic gradient descent (sgd) and the *conditionned* variant (csgd) presented in Section 5.4 where the matrix Φ_k is an averaging of past Hessian estimates as given in Equation (5.4). We shall compare equal weights $\omega_{j,k} = (k+1)^{-1}$ and adaptive weights $\omega_{j,k} \propto \exp(-\eta\|\theta_j - \theta_k\|_1)$ with $\eta > 0$ to give more importance to Hessian estimates associated to iterates which are closed to the current point. Furthermore, for computational reason, we consider a novel adaptive stochastic first-order method which is a variant of Adagrad.

Starting from the null vector $\theta_0 = (0, \ldots, 0) \in \mathbb{R}^d$, we use optimal learning rate of the form $\gamma_k = \alpha/(k + k_0)$ (Bottou et al., 2018) and set $\lambda_k^{(m)} \equiv 0, \lambda_k^{(M)} = \Lambda\sqrt{k}$ in the experiments where γ, k_0 and Λ are tuned using a grid search. The means of the optimality ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$, obtained over 100 independent runs, are presented in Figures below.

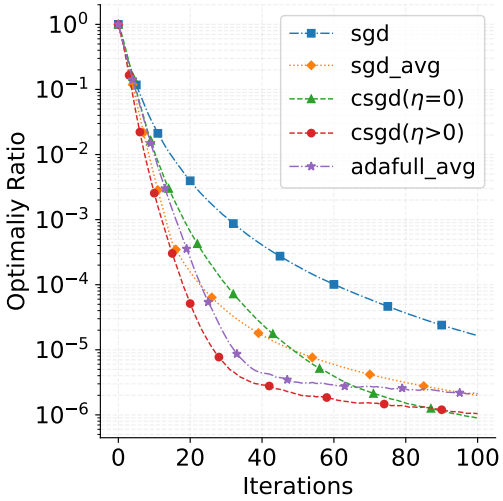
Methods in competition. The different methods in the experiments are:

- *sgd*: standard stochastic gradient descent.
- *sgd_ avg*: Polyak-averaging stochastic gradient descent , with a burn-in period to avoid the poor performance of bad initialization.
- *csgd*($\eta = 0$) and *csgd*($\eta > 0$): *conditioned* stochastic gradient descent methods with equal and adaptive weights where the matrix Φ_k is an averaging of past Hessian estimates as given in Equation (5.4).
- *adafull_ avg*: The variant of Adagrad presented in Section 5.4 where the gradient matrix G_k is updated as an average $G_k = \delta I + (1/k)\sum_{i=1}^k g_i g_i^\top$ and $C_k = G_k^{-1/2}$ instead of the cumulative sum provided in the literature of Adagrad. Note that averaging here allows to anneal the stochastic noise whereas classical versions of Adagrad often rely on true gradients and may use cumulative sums. The parameter δ is also tuned using a grid search.

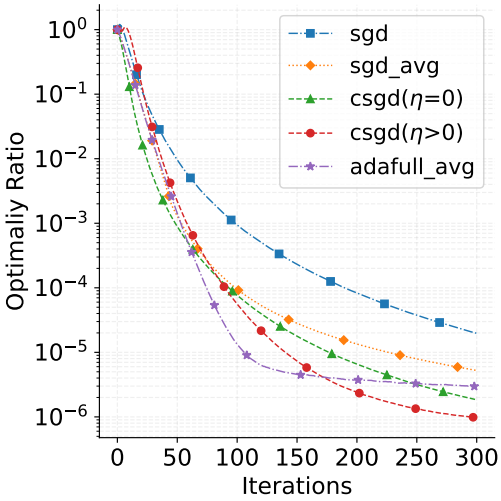
We focus on Ridge regression on simulated data with $n = 10,000$ samples in dimensions $d \in \{20; 100\}$. Stochastic gradient methods are known to greatly benefit from mini-batch instead of picking a single random sample when computing the gradient estimate. We use a batch-size equal to $|B| = 16$. In Figure 5.2, we can see that *conditioned* SGD outperforms standard SGD. Furthermore, adaptive weights ($\eta > 0$) improve the convergence speed of *conditioned* SGD methods. Interestingly, the novel approach *adafull_ avg* offers great performance at a cheap computing cost. Indeed, the update of C_{k+1} relies on the inverse of an average. This operation can be carried out in an efficient way thanks to Woodbury matrix identity.

Real-world data. We now turn our attention to real-world data and consider again the Ridge regression problem on the following datasets: *Boston Housing dataset* (Harrison Jr and Rubinfeld, 1978) ($n = 506; d = 14$) and *Diabetes dataset* (Dua and Graff, 2017) ($n = 442; d = 10$).
• *Boston Housing dataset* (Harrison Jr and Rubinfeld, 1978): This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It contains $n = 506$ samples in dimension $d = 14$.
• *Diabetes dataset* (Dua and Graff, 2017): Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of $n = 442$ diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

The means of the optimality ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$, obtained over 100 independent runs, are presented in Figure 5.3. Once again, the *conditioned* SGD methods offer better performance than plain SGD. For these datasets, it is the *conditioning* matrix with adaptive weights as given in Equation (5.4) which presents the best results.

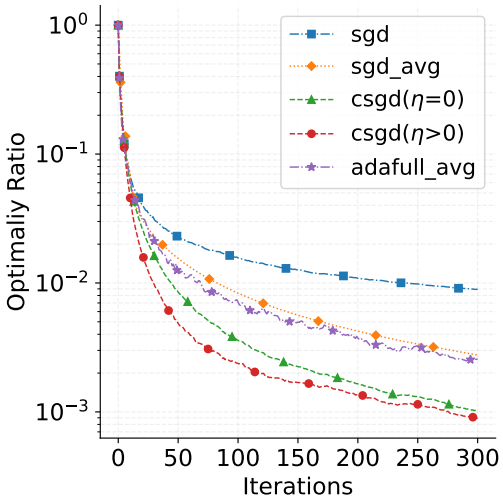


(a) Ridge $d = 20$)

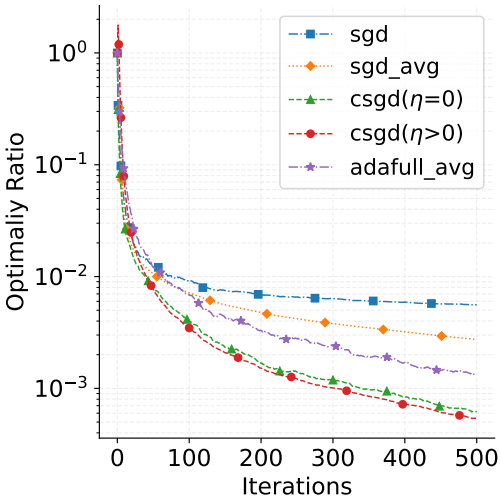


(b) Ridge $d = 100$

Figure 5.2 – Ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$ for Ridge regression in dimension $d \in \{20; 100\}$.



(a) Boston Dataset



(b) Diabetes dataset

Figure 5.3 – Ratio $k \mapsto [F(\theta_k) - F(\theta^\star)]/[F(\theta_0) - F(\theta^\star)]$ for Ridge regression on datasets Boston and Diabetes.

Chapter 6

SGD with Coordinate Sampling: Theory and Practice

Contents

While classical forms of stochastic gradient descent algorithm treat the different coordinates in the same way, a framework allowing for adaptive (*non uniform*) coordinate sampling is developed to leverage structure in data. In a non-convex setting and including zeroth-order gradient estimate, almost sure convergence as well as non-asymptotic bounds are established. Within the proposed framework, we develop an algorithm, MUSKETEER, based on a reinforcement strategy: after collecting information on the noisy gradients, it samples the most promising coordinate (*all for one*); then it moves along the one direction yielding an important decrease of the objective (*one for all*). Numerical experiments on both synthetic and real data examples confirm the effectiveness of MUSKETEER in large scale problems.

6.1 Introduction

Coordinate Descent (CD) algorithms have become unavoidable in modern machine learning because they are tractable (Nesterov, 2012) and competitive to other methods when dealing with key problems such as support vector machines, logistic regression, LASSO regression and other ℓ_1-regularized learning problems (Wu et al., 2008; Friedman et al., 2010). They are applied in a wide variety of problems ranging from linear systems (Lee and Sidford, 2013; Beck and Tetruashvili, 2013) to finite sum optimization (Necoara et al., 2014; Lu and Xiao, 2015) and composite functions (Richtárik and Takáč, 2014) with parallel (Fercoq and Richtárik, 2015; Richtárik and Takáč, 2016b), distributed (Fercoq et al., 2014; Qu et al., 2015) and dual (Shalev-Shwartz and Zhang,

2013; Csiba et al., 2015; Perekrestenko et al., 2017) variants. In many contributions (Loshchilov et al., 2011; Richtárik and Takáč, 2016a; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017), the choice of the coordinate sampling policy is conducted through some optimality criterion estimated along the algorithm. On the one hand, efficient forms of CD methods rely on a deterministic procedure (Nutini et al., 2015) which adapts to the underlying structure in data at the expense of higher calculation and thus, may be costly. On the other hand, stochastic gradient descent (SGD) methods are computationally efficient but often treat all coordinates equally and thus, may be sub-optimal. In the spirit of adaptive schemes, we tend to bridge the gap between the best of both worlds by developing, within a noisy gradient framework, a general stochastic coordinate descent method with a particular selection strategy.

We are interested in solving unconstrained optimization problems of the following form $\min_{\theta\in\mathbb{R}^d} f(\theta)$, where the objective function f may be either known exactly or accessed through noisy observations. When f is differentiable, a common appproach is to rely on the gradient of f. However, in many scenarios and particularly in large-scale learning, the gradient may be hard to evaluate or even intractable. Hence, one usually approximates the gradient using zeroth or first order estimates (Ghadimi and Lan, 2013; Lian et al., 2016). The former constructs pseudo-gradients by sampling some perturbed points or using finite differences (Flaxman et al., 2005; Duchi et al., 2012; Nesterov and Spokoiny, 2017; Shamir, 2017) (see Liu et al. (2020) for a recent survey and numerous references) leading to biased gradient estimates while the latter often relies on data sampling techniques (Needell et al., 2014; Papa et al., 2015) to obtain unbiased gradient estimates. In both cases, a random gradient estimate is available at a cheap computing cost and the method consists in moving along this estimate at each iteration. Early seminal works on such stochastic algorithms include Robbins and Monro (1951); Kiefer et al. (1952) and a recent review dealing with large scale learning problems is given in Bottou et al. (2018).

Starting from an initial point $\theta_0 \in \mathbb{R}^d$, the SGD algorithm is defined by the update rule

$$\forall t \geq 0, \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

where $g_t \in \mathbb{R}^d$ is a gradient estimate at θ_t (possibly biased) and $(\gamma_t)_{t\geq 1}$ is some learning rate sequence that should decrease throughout the algorithm. While the computation of g_t may be cheap, it still requires the computation of a vector of size d which may be a critical issue in high-dimensional problems. To address this difficulty, we rely on sampling well-chosen coordinates of the gradient estimate at each iteration.

We consider the framework of stochastic coordinate gradient descent (SCGD) which modifies standard stochastic gradient descent methods by adding a selection step to perform random coordinate descent. The SCGD algorithm is defined by the following iteration

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

where ζ_{t+1} is a random variable valued in $[\![1, d]\!]$ which selects a coordinate of the gradient estimate. The distribution of ζ_t is called the *coordinate sampling policy*. Note that the SCGD framework is very general as it contains as many methods as there are ways to generate both the gradient estimate g_t and the random variables ζ_t.

Contributions. The main contributions are as follows

(i)(Theory) We show the almost-sure convergence of the SCGD iterates $(\theta_t)_{t\in\mathbb{N}}$ towards stationary points in the sense that $\nabla f(\theta_t) \to 0$ almost surely as well as non-asymptotic bounds on the optimality gap $\mathbb{E}[f(\theta_t) - f^\star]$ where $f^\star$ is a lower bound of f. The working conditions are relatively weak as the function f is only required to be L-smooth (classical in non-convex problems) and the stochastic gradients are possibly biased with unbounded variance, using a growth condition related to *expected smoothness* (Gower et al., 2019).

(ii)(Practice) We develop a new algorithm, called MUSKETEER, for *MUltivariate Stochastic Knowledge Extraction Through Exploration Exploitation Reinforcement*. In the image of the motto 'all for one and one for all', this procedure belongs to the SCGD framework with a particular design for the *coordinate sampling policy*. It compares the value of all past gradient estimates g_t to select a descent direction (*all for one*) and then moves the current iterate according to the chosen direction (*one for all*). The heuristic is the one of reinforcement learning in the sense that large gradient coordinates represent large decrease of the objective and can be seen as high rewards. The resulting directions should be favored compared to the path associated to small gradient coordinates. By updating the *coordinate sampling policy*, the algorithm is able to detect when a direction becomes rewarding and when another one stops being engaging.

Related work. The authors of (Nutini et al., 2015) investigate the deterministic Gauss-Southwell rule which consists of picking the coordinate with maximum gradient value. In trusting large gradients, this rule looks like the one of MUSKETEER except that no stochastic noise -neither in the gradient evaluation nor in the coordinate selection- is present in their algorithm. In that aspect, our method differs from all the previous CD studies (Loshchilov et al., 2011; Richtárik and Takáč, 2016a; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017) which rely on ∇f. Among the SGD literature, compression and sparsification methods (Alistarh et al., 2017; Wangni et al., 2018) were developed for communication efficiency. The former use compression operators to select a few components of the gradient estimates at the cost of full gradient computation and coordinate sorting. The latter use a gradient estimate g which is sparsified using probability weights to reach an unbiased estimate of the gradient. In contrast, the SCGD framework allows the gradient to be biased as no importance re-weighting is performed. Note also that, to cover zeroth-order methods, the gradient estimate itself g_t is allowed to be biased as for instance in the recent study of Ajalloeian and Stich (2020). The proofs of the asymptotic convergence results are based on ideas from Bertsekas and Tsitsiklis (2000) with particular extensions in the framework of biased gradient estimates. Finally, the non-asymptotic bounds are inspired from Moulines and Bach (2011) where the authors provide a non-asymptotic analysis for standard SGD.

Outline. Section 6.2 introduces the mathematical framework with the different sampling strategies and Section 6.3 contains our main theoretical results. Section 6.4 is dedicated to MUSKETEER algorithm and a numerical analysis is performed in Section 6.5. Proofs, technical details and additional experiments may be found in auxiliary sections.

6.2 Mathematical Background

6.2.1 Notation and problem set-up

Notation. Denote by $(e_1, \dots, e_d)$ the canonical basis of $\mathbb{R}^d$ and for $k \in [\![1, d]\!]$, $C(k) = e_k e_k^T \in \{0,1\}^{d\times d}$ is a diagonal matrix with a 1 in position k. $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are respectively the Euclidian and infinity norm. For any $u \in \mathbb{R}^d$, $u^{(k)}$ is the k-th coordinate of u; $\mathbb{1}_A$ is the indicator function of the event A, *i.e.*, $\mathbb{1}_A = 1$ is A is true and $\mathbb{1}_A = 0$ otherwise. Denote by $\mathcal{U}([\![1, d]\!])$ the uniform distribution over $[\![1, d]\!]$. For a vector of probability weights $p = (p^{(1)}, \dots, p^{(d)})$ with $\sum_{k=1}^d p^{(k)} = 1$, denote by $Q(p)$ the associated categorical distribution.

Problem set-up. Consider the classical stochastic optimization problem

$$\min_{\theta\in\mathbb{R}^d} \Big\{ f(\theta) = \mathbb{E}_\xi[f(\theta, \xi)] \Big\},$$

where ξ is a random variable. In many scenarios, *e.g.* empirical risk minimization or reinforcement learning, the gradient ∇f cannot be computed in a reasonable time and only a stochastic version, possibly biased, is available. The distribution of ξ is called the *data sampling policy* as it refers to the sampling mechanism in the empirical risk minimization (ERM) framework. This running example is presented below and shall be considered throughout the paper. Other classical optimization problems where stochastic gradients are available include adaptive importance sampling (Delyon and Portier, 2018), policy gradient methods (Hanna et al., 2019) and optimal transport (Genevay et al., 2016).

Running Example (ERM). Given some observed data $z_1, \dots, z_n \subset \mathcal{Z}$ and a loss function $\ell : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$, the objective function f approximates the risk $\mathbb{E}_z[\ell(\theta, z)]$ by the so-called empirical risk defined as

$$\forall \theta \in \mathbb{R}^d, \quad f(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(\theta, z_i).$$

Evaluating f or its gradient is prohibitive in large scale machine learning as it requires seeing all the samples in the dataset. Instead, after picking at random an index $j = \xi$, uniformly distributed over $[\![1, n]\!]$, the k-th coordinate of the gradient estimate may be computed as $(\ell(\theta + he_k, z_j) - \ell(\theta, z_j))/h$. When differentiation is possible, another gradient estimate is offered by $\nabla_\theta \ell(\theta, z_j)$. These two gradient estimates are of a different nature: the first one, often referred to as zeroth-order estimate, is biased whereas the second one, often referred to as first order estimate, is unbiased.

6.2.2 Gradient estimates

Throughout the paper, the gradient generator is denoted by $g_h(\cdot, \xi)$ where the parameter $h \geq 0$ represents the underlying bias as claimed in the next assumption. This level of generality allows to include zeroth-order estimate as discussed right after the assumption.

Assumption 6.1 (Biased gradient)**.** *There exists a constant $c \geq 0$ such that:*

$$\forall h > 0, \forall \theta \in \mathbb{R}^d, \quad \|\mathbb{E}_\xi[g_h(\theta, \xi)] - \nabla f(\theta)\|_2 \leq ch.$$

This general assumption enables to work with classical unbiased gradient in the framework of first order estimates by taking $c = 0$. Furthermore, Assumption 6.1 is satisfied for the following well-spread zeroth-order estimates.

Example 1 (smoothing). The smoothed gradient estimate (Nesterov and Spokoiny, 2017) is given for all $\theta \in \mathbb{R}^d$ by $g_h(\theta, \xi) = h^{-1}[f(\theta + hU, \xi) - f(\theta, \xi)]U$ where U is a standard Gaussian vector (independent from ξ). An alternative version consists in taking U uniformly distributed over the unit sphere.

Example 2 (finite differences). The finite differences gradient estimate is given for all $\theta \in \mathbb{R}^d$ by $g_h(\theta, \xi) = \sum_{k=1}^d g_h(\theta, \xi)^{(k)} e_k$ where for all $k = 1, \ldots, p$ the coordinates are $g_h(\theta, \xi)^{(k)} = h^{-1}[f(\theta + he_k, \xi) - f(\theta, \xi)]$.

Both previous examples share the following general property. There exists a probability measure ν satisfying $\int_{\mathbb{R}^d} xx^\top \nu(\mathrm{d}x) = I_d$ such that,

$$\forall h > 0, \theta \in \mathbb{R}^d, \quad \mathbb{E}_\xi[g_h(\theta, \xi)] = \int_{\mathbb{R}^d} x \left\{ \frac{f(\theta + hx) - f(\theta)}{h} \right\} \nu(\mathrm{d}x). \tag{6.1}$$

The smoothed gradient estimate is recovered when ν is the standard Gaussian measure and taking $\nu = \sum_{k=1}^d \delta_{e_k}/d$ covers the finite differences estimate. As detailed in the next subsection, an interesting framework is to use a measure ν that evolves through time and put different weights on the different directions. As stated in the following proposition, when the function f is L-smooth, i.e., ∇f is L-Lipschitz, the bias of the gradient estimate (6.1) is of order h and thus satisfies Assumption 6.1.

Proposition 6.2. *Under Eq. (6.1), if f is L-smooth, then Assumption 6.1 holds true with $c = \sqrt{C}L/2$ where $C = \int_{\mathbb{R}^d} \|x\|_2^6 \nu(\mathrm{d}x) < \infty$.*

The previous proposition allows us to cover the two methods: smoothing and finite difference. Note that for the latter, the constant C is equal to 1.

6.2.3 Coordinate Sampling Policy

Let $(\xi_t)_{t\geq 1}$ be a sequence of independent and identically distributed random variables. Let $(\gamma_t)_{t\geq 1}$ be a sequence of positive numbers called *learning rates*. Let $(h_t)_{t\geq 1}$ be a sequence of positive numbers called *smoothing parameters*. Denote by $g_t = g_{h_{t+1}}(\theta_t, \xi_{t+1})$ the gradient estimate at time t. The classical SGD update rule is given by

$$\theta_{t+1} = \theta_t - \gamma_{t+1} g_t, \quad t \geq 0, \tag{6.2}$$

For any $t \in \mathbb{N}, \mathcal{F}_t = \sigma(\theta_0, \theta_1, \ldots, \theta_t)$ is the σ-field associated to the sequence of iterates $(\theta_t)_{t\in\mathbb{N}}$.

The framework of SCGD is introduced thanks to random coordinate sampling. At each step, only one coordinate of the parameter of interest is updated. This coordinate is selected at random according to a distribution valued in $[\![1, d]\!]$ which is allowed to evolve during the algorithm. The iteration of the coordinate sampling algorithm is given coordinate-wise by

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases} \tag{6.3}$$

where ζ_{t+1} is a random variable valued in $[\![1, d]\!]$. Hence ζ_{t+1} selects the coordinate along which the t-th descent shall proceed. The distribution of ζ_{t+1} is called the *coordinate sampling policy* as opposed to the *data sampling policy* governed by the random variable ξ_{t+1}. The distribution of ζ_{t+1} is characterized by the probability weights vector $p_t = (p_t^{(1)}, \dots, p_t^{(d)})$ defined by

$$p_t^{(k)} = \mathbb{P}(\zeta_{t+1} = k | \mathcal{F}_t), \quad k \in [\![1, d]\!].$$

The categorical distribution on $[\![1, d]\!]$ associated to p_t is denoted by $Q(p_t)$, *i.e.*, conditionally to $\mathcal{F}_t$, we have:

$$\forall t \geq 0, \quad \zeta_{t+1} \sim Q(p_t) \quad \text{with} \quad p_t = (p_t^{(1)}, \dots, p_t^{(d)}).$$

Running Example (ERM). The CD algorithm defined by Equation (6.3) can easily be applied in the ERM framework. The *coordinate sampling* strategy $\zeta \sim Q(p_t)$ combined with the uniform *data sampling* $\xi \sim \mathcal{U}([\![1, n]\!])$ leads to $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - (\gamma_{t+1}/h_{t+1})(\ell(\theta_t + h_{t+1}e_\zeta, z_\xi) - \ell(\theta_t, z_\xi))$ (zeroth-order) and $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1}\partial_{\theta_\zeta}\ell(\theta_t, z_\xi)$ (first order).

Given the past, the *data sampling* and *coordinate sampling* draws should not be related.

Assumption 6.3 (Conditional Independence)**.** *ζ_{t+1} is independent from ξ_{t+1} conditionally on $\mathcal{F}_t$.*

This assumption is natural in the ERM context as in most cases there is no particular link between the sample indexes and the coordinates. Futhermore, the independence property plays an important role in our proofs. The SCGD algorithm defined in (6.3) is simply written with matrix notation as

$$\theta_{t+1} = \theta_t - \gamma_{t+1} C(\zeta_{t+1}) g_t,$$

where $C(k) = e_k e_k^\top \in \mathbb{R}^{d \times d}$ has its entries equal to 0 except the (k, k) which is 1. Observe that the distribution of the random matrix $C(\zeta_{t+1})$ is fully characterized by the matrix

$$C_t = \mathbb{E}[C(\zeta_{t+1}) | \mathcal{F}_t] = \mathrm{Diag}(p_t^{(1)}, \dots, p_t^{(d)}).$$

Note that under Assumptions 6.1 and 6.3, the average move of SCGD follows a biased gradient direction. For instance, when $c = 0$ the average move of SCGD is given by $\mathbb{E}[\theta_{t+1} - \theta_t | \mathcal{F}_t] = -\gamma_{t+1} C_t \nabla f(\theta_t)$ which bears resemblance to the Conditioned-SGD iteration (Bottou et al., 2018, Section 6.2). Such preprocessing is meant to refine the gradient direction through a matrix mulitplication for a better understanding of the underlying structure of the data. A natural question rises on the choice of the matrix C_t among all the possible coordinate sampling distributions.

The SCGD framework is efficient as soon as one can compute each coordinate of the gradient estimate. This is the case for zeroth-order (ZO) optimization with finite differences where the full gradient estimate uses d partial derivatives, each of them requiring two queries of the objective function. SCGD reduces this cost to a single coordinate update.

Remark 6.4 (Batch coordinates)**.** *A natural extension is to consider subsets of coordinates, a.k.a. block-coordinate descent. Note that this framework is covered by our approach as the proofs can be extended by summing different matrices $C(\zeta)$. Similarly*

to mini-batching (Gower et al., 2019), one can consider multiple draws for the coordinates that are to be updated. The selecting random matrix $C(\zeta_{t+1})$ may be replaced by a diagonal matrix with $m(< d)$ non-zero coefficients. For that matter, it is enough to have multiple draws from the categorical distribution $Q(p_t)$.

Remark 6.5 (Parallelization)**.** *Several families of communication-reduction methods such as quantization (Alistarh et al., 2017), gradient sparsification (Wangni et al., 2018; Alistarh et al., 2018) or local-SGD (Patel and Dieuleveut, 2019) have been proposed to reduce the overheads of distribution. The SCGD framework can benefit from such data parallelization techniques. When a fixed number m of machines is available, it is then possible to gain computational acceleration by drawing m times the coordinate distribution $Q(p_t)$ on the different machines and then transmit the batch of selected coordinates to the workers.*

6.2.4 Adaptive and Unbiased Policies

To understand more clearly the differences between SGD and SCGD, we shall rely on a more general iteration scheme. This framework is useful to compare different algorithms in terms of adaptive policies and unbiased estimates. Consider the following general update rule

$$\theta_{t+1} = \theta_t - \gamma_{t+1} h(\theta_t, \omega_{t+1}), \quad t \geq 0 \tag{6.4}$$

where h is a gradient generator and $(\omega_t)_{t\geq 1}$ is a sequence of random variables which are not necessarily independent nor identically distributed. Observe that both frameworks, SGD and SCGD, are instances of (6.4). For example, the randomness of SCGD can be expressed through $\omega_t = (\xi_t, \zeta_t)$.

Definition 6.6 (Policy)**.** *Denote by P_t the distribution of ω_{t+1} given $\mathcal{F}_t$. The sequence $(P_t)_{t\geq 0}$ is called the policy of the stochastic algorithm.*

The policy of a stochastic algorithm is an important tool as it determines the randomness introduced over time. On the one hand, it provides insights on the expected behavior of the algorithm. On the other hand, it measures the ability to adapt through the iterations.

Definition 6.7 (Unbiased and Adaptive)**.** *A policy $(P_t)_{t\geq 0}$ is called "unbiased" if: $\forall \theta \in \mathbb{R}^d, t \geq 0$, $\int h(\theta, \omega) P_t(\mathrm{d}\omega) \propto \nabla f(\theta)$. It is called "naive" if P_t does not change with t, otherwise it is adaptive.*

With these definitions in mind, it is clear that the SGD policy (6.2) under Assumption 6.1 with $c = 0$ is unbiased and naive, and so does the policy induced by first order gradient in ERM.

Within the framework of SCGD, a policy cannot be unbiased and adaptive as claimed in the next proposition.

Proposition 6.8 (Unbiased coordinate policy)**.** *Suppose that Assumption 6.1 is fulfilled with $c = 0$ and that $\mathrm{Span}\{\nabla f(\theta) : \theta \in \mathbb{R}^d\}$ is dense in $\mathbb{R}^d$, then the only unbiased coordinate sampling policy is $C_t = I_d/d$. It corresponds to uniform coordinate sampling.*

When working under Assumption 6.1 with $c = 0$, SCGD with uniform coordinate sampling is unbiased and hence similar to SGD. This is confirmed in the numerical experiments (Appendix 6.E and 6.G). However, a uniform sampling does not use any available information to favor coordinates among others. Thus, the approach promoted in the paper is different: past gradient values are used to update the probability weights of C_t. The resulting method is an adaptive algorithm which is biased.

Remark 6.9 (Importance Coordinate Sampling)**.** *Note that the general framework defined above includes the particular case where the coordinates are selected according to ζ then reweighted as proposed in (Wangni et al., 2018). This corresponds to the choice $h(\theta, \omega_{t+1}) = C_t^{-1} C(\zeta_{t+1}) g(\theta, \xi_{t+1})$. Even though such a policy is adaptive and unbiased, it turns out -from our numerical experiments (Appendix 6.F)- that it behaves similarly to the uniform version and is therefore sub-optimal.*

6.3 Main Theoretical Results

In a general non-convex setting, we investigate the almost sure convergence of SCGD algorithms as well as non-asymptotic bounds. The following assumptions on the objective function f are classical among the SGD literature.

Assumption 6.10 (Lower bound)**.** *There exists $f^\star \in \mathbb{R}$ such that: $\forall \theta \in \mathbb{R}^d, f(\theta) \geq f^\star$.*

Assumption 6.11 (Smoothness)**.** *The objective $f : \mathbb{R}^d \to \mathbb{R}$ is twicely continuously differentiable and L-smooth: $\forall \theta, \eta \in \mathbb{R}^d, \quad \|\nabla f(\theta) - \nabla f(\eta)\|_2 \leq L \|\theta - \eta\|_2$.*

Remark 6.12 (Coordinate smoothness)**.** *Note that this assumption may be refined using the notion of coordinate smoothness with parameters $(L_1, \dots, L_p)$ where for all $k = 1, \dots, d$, $\partial_k f(\cdot)$ is L_k-Lipschitz, i.e., for all $\theta \in \mathbb{R}^d, \delta \in \mathbb{R}, |\partial_k f(\theta + \delta e_k) - \partial_k f(\theta)| \leq L_k |\delta|$. Within this framework, small values of L_k are associated to a high degree of smoothness in the k-th direction. Conversely, large values of L_k are associated to more difficult minimization problems along that direction. Intuitively, it requires more energy to minimize f along these directions and one should assign more sampling probability on coordinates with larger L_k (see Proposition 6.32 in the appendix).*

When dealing with stochastic algorithms, the stochastic noise associated to the gradient estimates is the keystone for the theoretical analysis. To treat this term, we consider a weak growth condition, related to the notion of *expected smoothness* as introduced in Gower et al. (2019) (see also Gazagnadou et al. (2019); Gower et al. (2021)).

Assumption 6.13 (Growth condition)**.** *With probability 1, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ such that for all $\theta \in \mathbb{R}^d$ and $h > 0$, we have: $\mathbb{E}\left[\|g_h(\theta, \xi)\|_2^2\right] \leq 2\mathcal{L}\left(f(\theta) - f^\star\right) + \sigma^2$.*

This bound on the stochastic noise $\mathbb{E}\left[\|g(\theta, \xi)\|_2^2\right]$ is the key to prove the almost sure convergence of the algorithm. Note that Assumption 6.13 is weak as it allows the noise to be large when the iterate is far away from the optimal point. In that aspect, it contrasts with uniform bounds of the form $\mathbb{E}\left[\|g(\theta, \xi)\|_2^2\right] \leq \sigma^2$ for some deterministic $\sigma^2 > 0$ (Nemirovski and Yudin, 1983; Nemirovski et al., 2009; Shalev-Shwartz et al., 2011). Observe that such uniform bound is recovered by taking $\mathcal{L} = 0$ in Assumption 6.13 but cannot hold when the objective function f is strongly convex (Nguyen et al.,

2018). The standard Robbins-Monro condition, $\sum_{t\geq 1}\gamma_t = +\infty$ and $\sum_{t\geq 1}\gamma_t^2 < +\infty$ is required in the next theorem which serves as a starting point for a comparison between SGD and SCGD methods.

Theorem 6.14 (Almost sure convergence of biased SGD)**.** *Suppose that Assumptions 6.1 to 6.13 are fulfilled and let $(\theta_t)_{t\in\mathbb{N}}$ be the sequence of iterates defined by (6.2). If the learning rates satisfy the Robbins-Monro condition and $h_t^2 = O(\gamma_t)$ then $\nabla f(\theta_t) \to 0$ a.s. when $t \to +\infty$.*

The SCGD framework is very general in the sense that it covers as many algorithms as there are ways to generate both the gradient estimate g_t and the random variables ζ_t that select the coordinates. The next theorem provides the almost sure convergence of particular instances of SCGD algorithms where the true gradient is known and used to define the *coordinate sampling* policy. It recovers the deterministic Gauss-Southwell rule (Nutini et al., 2015) and extends it to the case where the coordinate weights are proportional to any norm of the current gradient $\nabla f(\theta_t)$.

Theorem 6.15 (Almost sure convergence of particular SCGD)**.** *Suppose that Assumptions 6.1 to 6.13 are fulfilled and let $(\theta_t)_{t\in\mathbb{N}}$ be the sequence of iterates defined by (6.3), i.e., $\theta_{t+1} = \theta_t - \gamma_{t+1}C(\zeta_{t+1})g_t$. If the learning rates satisfy the standard Robbins-Monro and $h_t^2 = O(\gamma_t)$, then the two following results hold:*

- *(a) (maximum gradient) if the selected coordinate follows the maximum coordinate of the gradient $\zeta_{t+1} = \arg\max_{k=1,\dots,d} |\partial_k f(\theta_t)|$ then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.*
- *(b) (gradient weights) if the selection weights are proportional to the gradient norm $C_t \propto (|\nabla_k f(\theta_t)|^q)_{1\leq k\leq d}$ with $q > 0$ then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.*

Remark 6.16 (Sparse Gradient)**.** *In light of the sparsity assumption used in Wang et al. (2018)(Assumption A5), note that SCGD methods with weights proportional to the gradient coordinates can outperform uniform coordinate sampling as they only select the relevant directions throughout the procedure. Such sparsity framework happens for instance in hyper-parameter tuning problems of learning systems: usually the performance of the system is insensitive to some hyper-parameters which implies the sparsity of the gradients.*

In the general case, one may not have access to the true gradient and can only rely on the estimate g_t. Another assumption is therefore needed on the weights of the *coordinate sampling* policy to ensure that all the coordinates of interest are selected throughout the algorithm. The success of the proposed approach relies on the following restrictions between the *learning rates* sequence $(\gamma_t)_{t\in\mathbb{N}}$ and the weights of the *coordinate policy*. This is formally stated in the following assumption, referred to as the extended Robbins-Monro condition. Denote by β_{t+1} the smallest probability weight at time t, *i.e.*, $\beta_{t+1} = \min_{1\leq k\leq p} p_t^{(k)}$.

Assumption 6.17 (Extended Robbins-Monro condition)**.** *$(\gamma_t)_{t\geq 1}$, $(\beta_t)_{t\geq 1}$ are positive sequences such that $\sum_{t\geq 1}\gamma_t\beta_t = +\infty$ and $\sum_{t\geq 1}\gamma_t^2 < +\infty$.*

From a practical point of view, those are not restrictive as they can always be implemented by the user. In the case $C_t = I_d$, this is simply the standard Robbins-Monro condition.

Theorem 6.18 (Almost sure convergence of general SCGD)**.** *Suppose that Assumptions 6.1 to 6.13 are fulfilled and let $(\theta_t)_{t\in\mathbb{N}}$ be the sequence of iterates defined by (6.3). Assume moreover that the learning rates satisfy Assumption 6.17, $h_t^2 = O(\gamma_t)$ and that (β_t) has a positive lower bound, then $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.*

Remark 6.19 (Global convergence)**.** *Other convergence results concerning the sequence of iterates towards global minimizers may be obtained by considering stronger assumptions including that f is coercive and the level sets of stationary points $\{\theta, \nabla f(\theta) = 0\} \cap \{\theta, f(\theta) = y\}$ are locally finite for every $y \in \mathbb{R}^d$ (see Gadat et al. (2018) or Appendix 6.B.1).*

For a non-asymptotic analysis, we place ourselves under the Polyak–Łojasiewicz (PL) condition (Polyak, 1963) which does not assume convexity of f but retains many properties of strong convexity, *e.g.* the fact that every stationary point is a global minimum.

Assumption 6.20 (PL inequality)**.** *There exists a constant $\mu > 0$ such that:*

$$\forall \theta \in \mathbb{R}^d, \|\nabla f(\theta)\|_2^2 \geq 2\mu \left(f(\theta) - f^\star\right).$$

Similarly to (Moulines and Bach, 2011), we introduce $\varphi_\alpha : \mathbb{R}_+^\star \to \mathbb{R}, \varphi_\alpha(t) = \alpha^{-1}(t^\alpha - 1)$ if $\alpha \neq 0$ and $\varphi_\alpha(t) = \log(t)$ if $\alpha = 0$. Denoting $\delta_t = \mathbb{E}[f(\theta_t) - f^\star]$ and assuming that $\beta_{t+1} \geq \beta > 0$, one can obtain the recursion equation: $\delta_t \leq \left(1 - 2\mu\beta\gamma_t + L\mathcal{L}\gamma_t^2\right)\delta_{t-1} + \gamma_t^2(\sigma^2 L + c^2)/2$, leading to the following theorem on non-asymptotic bounds for SCGD methods.

Theorem 6.21 (Non-asymptotic bounds)**.** *Suppose that Assumptions 6.1 to 6.20 are fulfilled and let $(\theta_t)_{t\in\mathbb{N}}$ defined in (6.3) with $\gamma_t = \gamma t^{-\alpha}$ and $h_t = \sqrt{\gamma_t}$. Denote by $\delta_t = \mathbb{E}[f(\theta_t) - f^\star]$ and assume that there exists $\beta > 0$ such that $\beta_{t+1} \geq \beta > 0$. We have for $\alpha \in [0, 1]$:*

- *If $0 \leq \alpha < 1$ then*

$$\delta_t \leq 2\exp\left(2L\mathcal{L}\gamma^2\varphi_{1-2\alpha}(t)\right)\exp\left(-\frac{\mu\beta\gamma}{4}t^{1-\alpha}\right)\left(\delta_0 + \frac{\sigma^2 + 2c^2}{2\mathcal{L}}\right) + \frac{\gamma(\sigma^2 L + 2c^2)}{\mu\beta}t^{-\alpha}$$

- *If $\alpha = 1$ then*

$$\delta_t \leq 2\exp\left(L\mathcal{L}\gamma^2\right)\left(\delta_0 + \frac{\sigma^2 + 2c^2}{2\mathcal{L}}\right)t^{-\mu\beta\gamma} + \left(\frac{\sigma^2 L}{2} + c^2\right)\gamma^2\varphi_{\mu\beta\gamma/2-1}(t)t^{-\mu\beta\gamma/2}$$

Remark 6.22 (Importance weights)**.** *The conclusion of Theorem 6.18 remains valid for the update rule $\theta_{t+1} = \theta_t - \gamma_{t+1}W_t C(\zeta_{t+1})g_t$ where W_t is a diagonal matrix with coefficients $(w_t^{(1)}, \ldots, w_t^{(d)})$ such that $\beta_{t+1} = \min_{1\leq k\leq d} w_t^{(k)} p_t^{(k)}$.*

Remark 6.23 (Norms and constants)**.** *A quick inspection of the proof reveals that Assumptions 6.1 and 6.13 may be replaced respectively by: $\forall \theta \in \mathbb{R}^d, h > 0$, $\|\mathbb{E}_\xi[g_h(\theta, \xi)] - \nabla f(\theta)\|_\infty \leq ch$ and $\max_{k=1,\ldots,d} \mathbb{E}[g_h^{(k)}(\theta, \xi)^2] \leq 2\mathcal{L}(f(\theta) - f(\theta^\star)) + \sigma^2$. Since $\|\cdot\|_\infty \leq \|\cdot\|_2 \leq \sqrt{d}\|\cdot\|_\infty$, the above constant scales more efficiently with the dimension.*

Remark 6.24 (Rates)**.** *The optimal convergence rate in Theorem 6.21 is of order $O(1/t)$, obtained with $\alpha = 1$ under the condition $\mu\beta\gamma > 2$. Such rate matches optimal asymptotic minimax rate for stochastic approximation (Agarwal et al., 2012) and recovers the rate of (Ajalloeian and Stich, 2020) for SGD with biased gradients.*

6.4 MUSKETEER Algorithm

This section is dedicated to the algorithm MUSKETEER which performs an adaptive reweighting of the coordinate sampling probabilities to leverage the data structure. Note that this procedure is general and may be applied on top of any stochastic optimization algorithm as soon as one has acces to coordinates of a gradient estimate. In view of Theorem 6.15 and Remark 6.16, the main idea is to rely on a stochastic version of the Gauss-southwell rule where the coordinates of the gradients are only available through random estimates. The algorithm of interest alternates between two elementary blocks: one for the *exploration* phase and another one for the *exploitation* phase.

Exploration phase. The goal of this phase is twofold: perform stochastic coordinate gradient descent and collect information about the noisy directions of the gradient. The former task is done using the current coordinate sampling distribution $Q(p_n)$ which is fixed during this phase whereas the latter is computed through cumulative gains.

Exploitation phase. This phase is the cornerstone of the probability updates since it exploits the knowledge of the cumulative gains to update the coordinate sampling probability vector p_n in order to sample more often the relevant directions of the optimization problem.

Algorithm 6.7 MUSKETEER

Require: $\theta_0 \in \mathbb{R}^d$, $N, T \in \mathbb{N}$, $(\gamma_t)_{t\geq 0}$, $(\lambda_n)_{n\geq 0}$, $\eta > 0$.

1. Initialize probability weights $p_0 = (1/d, \dots, 1/d)$ // start with uniform sampling
2. Initialize cumulative gains $G_0 = (0, \dots, 0)$
3. **for** $n = 0, \dots, N-1$ **do**
4. Initialize current gain $\widetilde{G}_0 = (0, \dots, 0)$
5. Run **Explore**(T, p_n) // to compute current gain $\widetilde{G}_T$
6. Run **Exploit**$(G_n, \widetilde{G}_T, \lambda_n, \eta)$ // to update weights p_{n+1}
7. **end for**
8. Return final point θ_N

Consider a fixed iteration $n \in \mathbb{N}$ of MUSKETEER's main loop. The *exploration* phase may be seen as a multi-armed bandit problem (Auer et al., 2002a) where the arms are the gradient coordinates for $k \in [\![1, d]\!]$. At each time step $t \in [\![1, T]\!]$, a coordinate ζ is drawn according to $Q(p_n)$ and the relative gradient $g_t^{(\zeta)}/p_n^{(\zeta)}$, representing the reward, is observed. Note that an importance sampling strategy is used to produce an unbiased estimate of the gradient when dealing with first order methods. The rewards are then used to build cumulative gains $\widetilde{G}_T$ which can be written in a vectorized form as an empirical sum of the visited gradients during the *exploration* phase

$$\forall k \in [\![1, d]\!], \quad \widetilde{G}_T^{(k)} = \frac{1}{T}\sum_{t=1}^{T} \frac{g_t^{(k)}}{p_n^{(k)}} \mathbb{1}_{\{\zeta_{t+1}=k\}}, \quad i.e. \quad \widetilde{G}_T = \frac{1}{T}\sum_{t=1}^{T} C_n^{-1} C(\zeta_{t+1}) g(\theta_t, \xi_{t+1}). \tag{6.5}$$

This average reduces the noise induced by the gradient estimates but may be sign-dependent. Thus, one may rely on the following cumulative gains which are also considered in the experiments,

$$\widetilde{G}_T = \frac{1}{T}\sum_{t=1}^{T} C_n^{-1} C(\zeta_{t+1})|g(\theta_t, \xi_{t+1})| \quad \text{or} \quad \widetilde{G}_T = \frac{1}{T}\sum_{t=1}^{T} C_n^{-1} C(\zeta_{t+1}) g(\theta_t, \xi_{t+1})^2. \quad (6.6)$$

Starting from $G_0 = (0, \dots, 0)$, the total gain G_n is updated in a online manner during the *exploitation* phase using the update rule $G_{n+1} = G_n + (\widetilde{G}_T - G_n)/(n+1)$. Once the average cumulative gains are computed, one needs to normalize them to obtain probability weights. Such normalization can be done by a natural ℓ_1-reweighting or a softmax operator with a parameter $\eta > 0$. To cover both cases, consider the normalizing function $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ defined by $\varphi(x)^{(k)} = |x^{(k)}|/\sum_{j=1}^p |x^{(j)}|$ or $\varphi(x)^{(k)} = \exp(\eta|x^{(k)}|)/\sum_{j=1}^p \exp(\eta|x^{(j)}|)$. Following the sequential approach of the EXP3 algorithm (Auer et al., 2002a,b), the probability weights are updated through a mixture between the normalized average cumulative gains $\varphi(G_n)$ and a uniform distribution. The former term takes into account the knowledge of the gains by exploiting the rewards while the latter ensures exploration. Given a sequence $(\lambda_n) \in [0,1]^{\mathbb{N}}$, we have for all $k \in [\![1, d]\!]$,

$$p_{n+1}^{(k)} = (1 - \lambda_n)\varphi(G_n)^{(k)} + \lambda_n \frac{1}{d}. \quad (6.7)$$

Algorithm Explore(T, p_n)

1. **for** $t = 1, \dots, T$ **do**
2. Sample $\zeta \sim Q(p_n)$ and data ξ
3. Update $\theta_{t+1}^{(\zeta)} = \theta_t^{(\zeta)} - \gamma_{t+1} g_h^{(\zeta)}(\theta_t, \xi)$
4. Update gain $\widetilde{G}_{t+1}^{(\zeta)}$ using (6.5) or (6.6)
5. **end for**
6. Return vector of gains $\widetilde{G}_T$

Algorithm Exploit$(G_n, \widetilde{G}_T, \lambda_n, \eta)$

1. Update total average gain G_n in an online manner
2. Compute normalized gains $\varphi(G_n)$ with ℓ_1-weights or softmax
3. Update probability weights p_{n+1} with the mixture of Eq.(6.7)

In view of Theorem 6.18, the convergence of the sequence of iterates $(\theta_t)_{t\in\mathbb{N}}$ obtained by MUSKETEER relies on the extended Robbins-Monro condition $\sum_{t\geq 1} \beta_t \gamma_t = +\infty$ which is implied by the weaker condition $\sum_{t\geq 1} \lambda_t \gamma_t = +\infty$ for both ℓ_1 and softmax weights. Observe that such a constraint is easily verified with either a fixed value $\lambda_t \equiv \lambda$ in the mixture update or more generally a slowly decreasing sequence, e.g. $\lambda_t = 1/\log(t)$. Since the gradients $\nabla f(\theta_t)$ get smaller through the iterations, the softmax weights get closer to $1/d$. Thus, in the asymptotic regime, there is no favorable directions among all the possible gradient directions. Hence, near the optimum, the *coordinate sampling policy* of MUSKETEER with softmax weights is likely to treat all the coordinates equally.

Theorem 6.25. *(Weak convergence) Suppose that Assumptions 6.1 to 6.13 are fulfilled and that the learning rates satisfy the standard Robbins-Monro condition. Then MUSKETEER's coordinate policy $(Q(p_n))_{n\in\mathbb{N}}$ with softmax normalization converges weakly to the uniform distribution, i.e., $Q(p_n) \rightsquigarrow \mathcal{U}([\![1, d]\!])$ as $n \to +\infty$.*

Remark 6.26. *(On the choice of λ_n and η) The uniform term in Equation (6.7) ensures that all coordinates are eventually visited. Taking $\lambda_n \to 0$ at a specific rate (which can be derived from the proof) gives more importance to the cumulative gains. The parameter η is fixed during the algorithm and may be tuned through an analysis of the regret (Auer et al., 2002a).*

Remark 6.27. *(Choice of Exploration Size T) Choosing the value of T is a central question known as the exploration-exploitation dilemma in reinforcement learning. As T gets large, the exploration phase gathers more information leading to fewer but more accurate updates. Conversely, with a small value of T, the probabilities get updated more often, at the price of less collected information. Setting $T = d$ ensures that, in average, all the coordinates are visited once during the exploration phase. Nevertheless, a smaller value $T = \lfloor \sqrt{d} \rfloor$ is taken in the experiments and lead to great performance.*

Remark 6.28. *(Asymptotic behavior) The previous results highlight two main features of MUSKETEER: the sequence of iterates converges almost surely and the coordinate policy converges weakly. The latter point suggests that, in the long run, MUSKETEER is similar to the uniform coordinate version of SCGD. However, the weak convergence of the rescaled process $(\theta_t - \theta^\star)/\sqrt{\gamma_t}$ remains an open question. In light of the link between SCGD and Conditioned-SGD, discussed in Section 6.2.3, we conjecture that the behavior of MUSKETEER with softmax weights is asymptotically equivalent to SCGD with uniform policy. This is in line with the continuity property obtained in Leluc and Portier (2020) within the Conditioned-SGD framework and relates to the convergence of stochastic Newton algorithms (Boyer and Godichon-Baggioni, 2020).*

6.5 Numerical Experiments

In this section, we empirically validate the SCGD framework by running MUSKETEER and competitors on synthetic and real datasets. First, we focus on regularized regression problems adopting the data generation process of (Namkoong et al., 2017)ch the covariates exhibit a certain block structure. Second, MUSKETEER is employed to train different neural networks models on real datasets for multi-label classification task. For ease of reproducibility, the code is available online[1]. Technical details and additional results (with different data settings, normalization and hyperparameters) are available in the appendix.

Methods in competition. The set of methods is restricted to zeroth-order methods. This choice leads to an honest comparison based on the number of function queries. MUSKETEER is implemented according to Section 6.4 with $T = \lfloor \sqrt{d} \rfloor$, softmax and ℓ_1 normalization for the simulated and real data respectively. The different cumulative gains of Eq. (6.6) are considered, namely AVG, SQR and ABS for the gradients, their squares or their absolute value respectively. The method FULL is the finite difference gradient estimate computed over all coordinates and UNIFORM stands for the uniform coordinate sampling policy. NESTEROV implements the gaussian smoothing of (Nesterov and Spokoiny, 2017). In all cases, the initial parameter is set to $\theta_0 = (0, \ldots, 0)^\top \in \mathbb{R}^d$ and the optimal SGD learning rate of the form $\gamma_k = \gamma/(k + k_0)$ is used.

Regularized linear models. We apply the Empirical Risk Minimization paradigm to regularized linear problems. Given a data matrix $X = (x_{i,j}) \in \mathbb{R}^{n \times d}$, labels $y \in \mathbb{R}^n$ or $\{-1, +1\}^n$ and a regularization parameter $\mu > 0$, the *Ridge regression* objective is

[1] https://github.com/RemiLELUC/SCGD-Musketeer

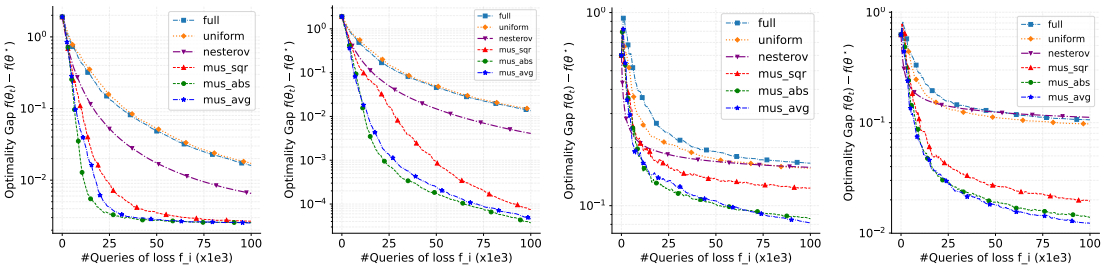
defined by

$$f(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} x_{i,j}\theta_j)^2 + \frac{\mu}{2}\|\theta\|_2^2$$

and the ℓ_2-*regularized logistic regression* is given by

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i\sum_{j=1}^{d} x_{i,j}\theta_j)) + \mu\|\theta\|_2^2.$$

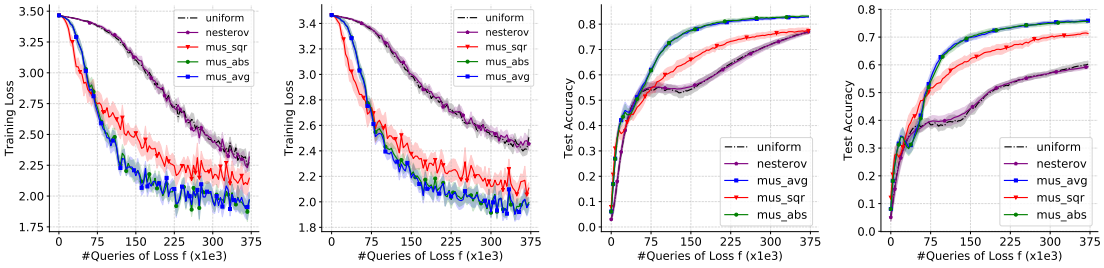
Similarly to (Namkoong et al., 2017), we endow the data matrix X with a block structure. The columns are drawn as $X[:,k] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, d]\!]$. The parameters are set to $n = 10,000$ samples in dimension $d = 250$ with an exploration size equal to $T = \lfloor\sqrt{d}\rfloor = 15$. The regularization parameter is set to the classical value $\mu = 1/n$. Figure 6.1 provides the graphs of the optimaliy gap $t \mapsto f(\theta_t) - f(\theta^\star)$ averaged over 20 independent simulations for different values of $\alpha \in \{2; 5; 10\}$. First, note that the uniform sampling strategy shows similar performance to the classical full gradient estimate. Besides, MUSKETEER with average or absolute gains shows the best performance in all configurations. Greater values of α, *i.e.* stronger block structure, improve our relative performance with respect to the other methods as shown by Figures 6.1b and 6.1d.



(a) Ridge $\alpha = 5$ (b) Ridge $\alpha = 5$ (c) Logistic $\alpha = 2$ (d) Logistic $\alpha = 5$

Figure 6.1 – $[f(\theta_t) - f(\theta^\star)]$ for Ridge and Logistic on Synthetic data with different block structures.

Neural Networks. We focus on the training of neural networks within the framework of multi-label classification. The datasets in the experiments are popular publicly available deep learning datasets: MNIST (Deng, 2012) and Fashion-MNIST (Xiao et al., 2017). Given an image, the goal is to predict its label among ten classes. The neural architecture is based on linear layers in dimension $d = 55,050$ with $T = 234$. Figure 6.2 shows the means and standard deviations of the training losses of the different ZO methods averaged over 10 independent runs. Interestingly, the performance of MUSKETEER also benefit from the adaptive structure in terms on accuracy of the test set (see Figures 6.2c and 6.2d). This allows to quantify the statistical gain brought by MUSKETEER over standard ZO methods.



(a) Mnist (b) Fashion (c) Mnist (d) Fashion

Figure 6.2 – Evolution of Training Loss (a),(b) and Test Accuracy (c),(d).

6.A Technical Proofs

6.A.1 Proof of Proposition 6.2

Under Eq.(6.1), using Jensen inequality, we find

$$\begin{aligned}\|\mathbb{E}_\xi[g_\mu(\theta,\xi)] - \nabla f(\theta)\|_2^2 &= \left\| \int_{\mathbb{R}^d} x \left(\frac{f(\theta+\mu x) - f(\theta)}{\mu} - x^\top \nabla f(\theta) \right) \nu(\mathrm{d}x) \right\|_2^2 \\ &\leq \int_{\mathbb{R}^d} \|x\|_2^2 \left(\frac{f(\theta+\mu x) - f(\theta)}{\mu} - x^\top \nabla f(\theta) \right)^2 \nu(\mathrm{d}x) \\ &= \mu^{-2} \int_{\mathbb{R}^p} \|x\|_2^2 \Big(f(\theta+\mu x) - f(\theta) - \mu x^\top \nabla f(\theta) \Big)^2 \nu(\mathrm{d}x)\end{aligned}$$

Using the quadratic bound of L-smooth functions, we obtain

$$\|\mathbb{E}_\xi[g_\mu(\theta,\xi)] - \nabla f(\theta)\|_2^2 \leq \mu^{-2} \frac{L^2}{4} \int \|x\|_2^2 \|\mu x\|_2^4 \nu(\mathrm{d}x) = \mu^2 \frac{L^2}{4} \int \|x\|_2^6 \nu(\mathrm{d}x).$$

6.A.2 Deterministic results for convergence of gradients

In this section we provide results ensuring the convergence to 0 of several gradient descent algorithms. They are meant to be *high-level* as they may be applied in different situations and *deterministic* because no randomness is measured but only an inclusion of events is considered. The results are key in the proofs.

Lemma 6.29 (Deterministic result 1)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a L-smooth function, $(\gamma_t)_{t\geq 1}$ a positive sequence of learning rates such that $\sum_t \gamma_t = \infty$. Let (θ_t) a random sequence obtained by the SGD update rule $\theta_{t+1} = \theta_t - \gamma_{t+1} g_t$. Let $\omega \in \Omega$ such that the following limits exist:*

$$(i)\ \sum_{t\geq 0} \gamma_{t+1} \|\nabla f(\theta_t(\omega))\|_2^2 < \infty \quad (ii)\ \sum_{t\geq 1} \gamma_t (g_{t-1}(\omega) - \nabla f(\theta_{t-1}(\omega))) < \infty$$

then $\nabla f(\theta_t(\omega)) \to 0$ as $t \to \infty$.

The next Lemma is the equivalent of Lemma 6.29 for a specific procedure which, at each iteration, moves only one well-chosen coordinate: the one with highest gradient value.

Lemma 6.30 (Deterministic result 2)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a L-smooth function (with respect to $|\cdot|_\infty$), $(\gamma_t)_{t\geq 1}$ a positive sequence of learning rates such that $\sum_t \gamma_t = \infty$. Let (θ_t) a random sequence obtained by the SCGD update rule $\theta_{t+1} = \theta_t - \gamma_{t+1} C(\zeta_{t+1}) g_t$ with $\zeta_{t+1} = \arg\max_{k=1,\dots,d} |\partial_k f(\theta_t)|$. Let $\omega \in \Omega$ such that the following limits exist:*

$$(i)\ \sum_{t\geq 0} \gamma_{t+1} |\nabla f(\theta_t(\omega)|_\infty^2 < \infty \quad (ii)\ \sum_{t\geq 1} \gamma_t C(\zeta_t)(g_{t-1}(\omega) - \nabla f(\theta_{t-1}(\omega))) < \infty,$$

then $\nabla f(\theta_t(\omega)) \to 0$ as $t \to \infty$.

We conclude with one last result which is valid for procedure where only one coordinate (chosen randomly) is moved at each iteration.

Lemma 6.31 (Deterministic result 3)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a L-smooth function, $(\gamma_t)_{t\geq 1}$ a positive sequence of learning rates such that $\sum_t \gamma_t = \infty$. Let (θ_t) a random sequence obtained by the SCGD update rule $\theta_{t+1} = \theta_t - \gamma_{t+1} C(\zeta_{t+1}) g_t$ where $\zeta_{t+1} \sim Q(p_t)$. Let $\omega \in \Omega$ such that the following limits exist:*

$$(i)\ \sum_{t\geq 0} \gamma_{t+1} \|\nabla f(\theta_t(\omega))\|_2^2 < \infty \quad (ii)\ \sum_{t\geq 1} \gamma_t (C(\zeta_t(\omega)) g_{t-1}(\omega) - C_{t-1} \nabla f(\theta_{t-1}(\omega))) < \infty$$

then $\nabla f(\theta_t(\omega)) \to 0$ as $t \to \infty$.

Proof of Lemma 6.29. The proof (and in particular the reasoning by contradiction) is inspired from the proof of Proposition 1 in Bertsekas and Tsitsiklis (2000). For ease of notation we omit the ω in the proof. Note that condition (i) along with $\sum_t \gamma_t = \infty$ implie that $\liminf_t \|\nabla f(\theta_t)\| = 0$. Now, by contradiction, let $\varepsilon > 0$ and assume that

$$\limsup_t \|\nabla f(\theta_t)\| > \varepsilon$$

We have that there is infinitely many t such that $\|\nabla f(\theta_t)\| < \varepsilon/2$ and also infinitely many t such that $\|\nabla f(\theta_t)\| > \varepsilon$. It follows that there is infinitely many *crossings* between the sets $\{t \in \mathbb{N} : \|\nabla f(\theta_t)\| < \varepsilon/2\}$ and $\{t \in \mathbb{N} : \|\nabla f(\theta_t)\| > \varepsilon\}$. A *crossing* is a collection of indexes $I_k = \{L_k, L_k + 1, \dots, U_k - 1\}$ with $L_k \leq U_k$ ($I_k = \emptyset$ when $L_k = U_k$) such that for all $t \in I_k$,

$$\|\nabla f(\theta_{L_k - 1})\| < \varepsilon/2 \leq \|\nabla f(\theta_t)\| \leq \varepsilon < \|\nabla f(\theta_{U_k})\|.$$

Define the following partial Cauchy sequence $R_k = \sum_{t=L_k}^{U_k} \gamma_t (g_{t-1} - \nabla f(\theta_{t-1}))$ and note that condition (ii) implies that $R_k \to 0$ as $k \to \infty$. For all $k \geq 1$,

$$\begin{aligned} \varepsilon/2 &\leq \|\nabla f(\theta_{U_k})\|_2 - \|\nabla f(\theta_{L_k - 1})\|_2 \\ &\leq \|\nabla f(\theta_{U_k}) - \nabla f(\theta_{L_k - 1})\|_2 \\ &\leq L \|\theta_{U_k} - \theta_{L_k - 1}\|_2, \end{aligned}$$

where we use that ∇f is L-Lipschitz. Then using the update rule $\theta_t - \theta_{t-1} = -\gamma_t g_{t-1}$, we have by sum

$$\begin{aligned}\varepsilon/2 &\le L\|\sum_{t=L_k}^{U_k} \theta_t - \theta_{t-1}\|_2 = L\|\sum_{t=L_k}^{U_k} \gamma_t g_{t-1}\|_2 \\ &\le L\|\sum_{t=L_k}^{U_k} \gamma_t \nabla f(\theta_{t-1})\|_2 + L\|\sum_{t=L_k}^{U_k} \gamma_t (g_{t-1} - \nabla f(\theta_{t-1}))\|_2 \\ &\le L\sum_{t=L_k}^{U_k} \gamma_t \|\nabla f(\theta_{t-1})\|_2 + L\|R_k\|_2\end{aligned}$$

Since in the previous equation $\|\nabla f(\theta_{t-1})\|_2 > \varepsilon/2$, we get

$$(\varepsilon/2)^2 \le L\sum_{t=L_k}^{U_k} \gamma_t \|\nabla f(\theta_{t-1})\|_2^2 + (\varepsilon/2)L\|R_k\|_2$$

But since $\sum_{t\ge 0} \gamma_{t+1}\|\nabla f(\theta_t)\|^2$ is finite and $\lim_k R_k = 0$, the previous upper bound goes to 0 and implies a contradiction. □

Proof of Lemma 6.30. For ease of readability, the variable ω is removed during the proof. By assumption, $|\nabla_{\zeta_{t+1}} f(\theta_t)| = |\nabla f(\theta_t)|_\infty$. Hence, (i) yields that $\liminf_t |\nabla f(\theta_t)|_\infty = 0$. The proof is by contradiction. Suppose that $\limsup_t |\nabla f(\theta_t)|_\infty > \epsilon$. There exists a sequence of *crossings* between the sets $\{t \in \mathbb{N} : |\nabla f(\theta_t)|_\infty < \varepsilon/2\}$ and $\{t \in \mathbb{N} : |\nabla f(\theta_t)|_\infty > \varepsilon\}$. Formally, there is a collection of indexes $I_k = \{L_k, L_k + 1, \dots, U_k - 1\}$ with $L_k \le U_k$ ($I_k = \emptyset$ when $L_k = U_k$) such that for all $t \in I_k$,

$$|\nabla f(\theta_{L_k - 1})|_\infty < \varepsilon/2 \le |\nabla f(\theta_t)|_\infty \le \varepsilon < |\nabla f(\theta_{U_k})|_\infty.$$

Define

$$R_k = \sum_{t=L_k}^{U_k} \gamma_t C_{\zeta_t}(g_{t-1} - \nabla f(\theta_{t-1}))$$

and use that ∇f is L-smooth to get

$$\begin{aligned}(\varepsilon/2) &\le |\nabla f(\theta_{U_k})|_\infty - |\nabla f(\theta_{L_k-1})|_\infty \\ &\le L|\theta_{U_k} - \theta_{L_k-1}|_\infty \\ &\le L|\sum_{t=L_k}^{U_k} \gamma_t C_{\zeta_t} \nabla f(\theta_{t-1})|_\infty + L\left|\sum_{t=L_k}^{U_k} \gamma_t C_{\zeta_t}(g_{t-1} - \nabla f(\theta_{t-1}))\right|_\infty \\ &= L|\sum_{t=L_k}^{U_k} \gamma_t C_{\zeta_t} \nabla f(\theta_{t-1})|_\infty + L|R_k|_\infty \\ &\le L\sum_{t=L_k}^{U_k} \gamma_t |C_{\zeta_t} \nabla f(\theta_{t-1})|_\infty + L|R_k|_\infty\end{aligned}$$

Noting that $|C_{\zeta_t} \nabla f(\theta_{t-1})|_\infty = |\nabla f(\theta_{t-1})|_\infty > \varepsilon/2$, we get

$$(\varepsilon/2)^2 \le L\sum_{t=L_k}^{U_k} \gamma_t |\nabla f(\theta_{t-1})|_\infty^2 + (\varepsilon/2)L|R_k|_\infty.$$

As the previous upper bound converges to 0 by assumption we reach a contradiction. □

Proof of Lemma 6.31. Following the proof of Lemma 6.29, we assume that $\limsup_t \|\nabla f(\theta_t)\|_2 > \varepsilon$ and consider the same collection of crossing indexes (L_k, U_k) to obtain that

$$\varepsilon/2 \leq L \sum_{t=L_k}^{U_k} \gamma_t \|C_{t-1}\nabla f(\theta_{t-1})\|_2 + L\|R_k\|_2$$

where $R_k = \sum_{t=L_k}^{U_k} \gamma_t (C(\zeta_t)g_{t-1} - C_{t-1}\nabla f(\theta_{t-1}))$ is a sequence that goes to 0. Since in the previous equation $C_{t-1} \preceq I_d$ and $\|\nabla f(\theta_{t-1})\|_2 > \varepsilon/2$, we get

$$(\varepsilon/2)^2 \leq L \sum_{t=L_k}^{U_k} \gamma_t \|\nabla f(\theta_{t-1})\|_2^2 + (\varepsilon/2)L\|R_k\|_2$$

and a contradiction follows as the above term goes to 0. □

6.A.3 Proof of Theorem 6.14

The proof follows from applying Lemma 6.29 in which two conditions are required:

$$(i) \sum_{t\geq 0} \gamma_{t+1}\|\nabla f(\theta_t(\omega))\|_2^2 < \infty \quad (ii) \sum_{t\geq 1} \gamma_t (g_{t-1}(\omega) - \nabla f(\theta_{t-1}(\omega))) < \infty.$$

Proof of condition (i). We classically rely on the Robbins-Siegmund Theorem (Theorem 6.33 in Section 6.B.4). Since $\theta \mapsto f(\theta)$ is L-smooth, we have the quadratic bound $f(\eta) \leq f(\theta) + \langle \nabla f(\theta), \eta - \theta\rangle + \frac{L}{2}\|\eta - \theta\|_2^2$. Using the update rule $\theta_{t+1} = \theta_t - \gamma_{t+1}g_t$, we get

$$\begin{aligned} f(\theta_{t+1}) &\leq f(\theta_t) + \langle \nabla f(\theta_t), \theta_{t+1} - \theta_t\rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|_2^2 \\ &= f(\theta_t) - \gamma_{t+1}\langle \nabla f(\theta_t), g_t\rangle + \frac{L}{2}\gamma_{t+1}^2\|g_t\|_2^2. \end{aligned}$$

Using that

$$2\langle a, b\rangle = \|a\|_2^2 + \|b\|_2^2 - \|a - b\|_2^2 \geq \|a\|_2^2 - \|a - b\|_2^2$$

and taking the conditional expectation, we get

$$\begin{aligned} \mathbb{E}_t\left[f(\theta_{t+1})\right] &\leq f(\theta_t) - \gamma_{t+1}\langle \nabla f(\theta_t), \mathbb{E}_t[g_t]\rangle + \frac{L}{2}\gamma_{t+1}^2\mathbb{E}_t[\|g_t\|_2^2] \\ &\leq f(\theta_t) - \frac{\gamma_{t+1}}{2}\|\nabla f(\theta_t)\|_2^2 + \frac{\gamma_{t+1}}{2}\|\nabla f(\theta_t) - \mathbb{E}_t[g_t]\|_2^2 + \frac{L}{2}\gamma_{t+1}^2\mathbb{E}_t[\|g_t\|_2^2] \end{aligned}$$

On the one hand, using Assumption 6.1, we obtain

$$\|\nabla f(\theta_t) - \mathbb{E}_t[g_t]\|_2^2 \leq h_{t+1}^2 c^2$$

On the other hand, using Assumption 6.13, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ such that almost surely

$$\forall t \in \mathbb{N}, \quad \mathbb{E}_t\left[\|g_t\|_2^2\right] = \mathbb{E}_\xi\left[\|g(\theta_t, \xi)\|_2^2\right] \leq 2\mathcal{L}\left(f(\theta_t) - f^\star\right) + \sigma^2.$$

Injecting $-f^\star$ on both sides, it follows that

$$\mathbb{E}_t\left[f(\theta_{t+1}) - f^\star\right] \leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \frac{\gamma_{t+1}}{2}\|\nabla f(\theta_t)\|_2^2 + \gamma_{t+1}h_{t+1}^2c^2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2$$

Introduce $V_t = f(\theta_t) - f^\star, W_t = \gamma_{t+1}\|\nabla f(\theta_t)\|_2^2/2$, $a_t = L\mathcal{L}\gamma_{t+1}^2$ and $b_t = c^2h_{t+1}^2\gamma_{t+1} + (L/2)\gamma_{t+1}^2\sigma^2$. These four random sequences are non-negative $\mathcal{F}_t$-measurable sequences with $\sum_t a_t < \infty$ and $\sum_t b_t < \infty$ almost surely. We have: $\forall t \in \mathbb{N}, \mathbb{E}\left[V_{t+1}|\mathcal{F}_t\right] \leq (1 + a_t)V_t - W_t + b_t$. We can apply Robbins-Siegmund Theorem to have

$$(a)\ \sum_{t\geq 0} W_t < \infty\ a.s. \qquad (b)\ V_t \xrightarrow{a.s.} V_\infty, \mathbb{E}\left[V_\infty\right] < \infty. \qquad (c)\ \sup_{t\geq 0}\mathbb{E}\left[V_t\right] < \infty.$$

Therefore we have a.s. that $(f(\theta_t))$ converges to a finite value $f_\infty \in L^1$ and $\sum_{t\geq 0}\gamma_{t+1}\|\nabla f(\theta_t)\|_2^2 < +\infty$. There exists an event $\Omega_0 \subset \Omega$ such that, $\mathbb{P}(\Omega_0) = 1$ and for every $\omega \in \Omega_0$, $\lim_t f(\theta_t(\omega)) < \infty$ and $\sum_{t\geq 0}\gamma_{t+1}\|\nabla f(\theta_t(\omega))\|_2^2 < \infty$.

Proof of condition (ii). We place ourselves on the event Ω_0 and omit the ω in notation for ease of clarity. First, since $\limsup_t f(\theta_t) < \infty$, we have that $(f(\theta_t))$ is bounded almost surely. It yields, in virtue of Assumption 6.13 that $\mathbb{E}_t\left[\|g_t\|_2^2\right] \leq 2\mathcal{L}\left(f(\theta_t) - f^\star\right) + \sigma^2 \leq C$ where C is a some finite random variable and the latter holds almost surely. It then follows that, almost surely $\sum_{t\geq 1}\gamma_t^2\mathbb{E}_t[\|g_t\|^2] \leq C\sum_{t\geq 1}\gamma_t^2 < \infty$. Now, observe that condition (ii) is satisfied as soon as

$$(a)\ \|\sum_{t\geq 0}\gamma_{t+1}(g_t - \mathbb{E}_t[g_t])\|_2 < \infty \quad \text{and} \quad (b)\ \|\sum_{t\geq 0}\gamma_{t+1}(\mathbb{E}_t[g_t] - \nabla f(\theta_t))\|_2 < \infty.$$

Equation (a) involves martingale increments whose quadratic variation satisfies

$$\sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|g_t - \mathbb{E}_t[g_t]\|^2] \leq \sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|g_t\|^2] < \infty,$$

which ensures that $\sum_{t\geq 0}\gamma_{t+1}(g_t - \mathbb{E}_t[g_t]) < \infty$ a.s. in virtue of Theorem 2.17 in **?**. The term in equation (b) is bounded using assumption 6.1 and we have

$$\sum_{t\geq 0}\gamma_{t+1}^2\|\mathbb{E}_t[g_t] - \nabla f(\theta_t)\|_2^2 \leq c^2\sum_{t\geq 0}\gamma_{t+1}^2h_t^2 < \infty,$$

which finally proves

$$(ii)\ \sum_{t\geq 0}\gamma_{t+1}(g_t(\omega) - \nabla f(\theta_t(\omega))) < \infty$$

and gives, in virtue of Lemma 6.29 the conclusion $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.

6.A.4 Proof of Theorem 6.15

Part (a) Maximum gradient. The proof follows from applying Lemma 6.30 in which two conditions are required:

$$(i)\ \sum_{t\geq 0}\gamma_{t+1}|\nabla f(\theta_t(\omega)|_\infty^2 < \infty \quad (ii)\ \sum_{t\geq 0}\gamma_{t+1}C(\zeta_{t+1})(g_t(\omega) - \nabla f(\theta_t(\omega))) < \infty.$$

Proof of condition (i). Again, we rely on the quadratic bound

$$\begin{aligned} f(\theta_{t+1}) &\leq f(\theta_t) - \gamma_{t+1}\langle \nabla f(\theta_t), C(\zeta_{t+1})g_t\rangle + \frac{L}{2}\gamma_{t+1}^2 \|C(\zeta_{t+1})g_t\|_2^2 \\ &= f(\theta_t) - \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t) g_t^{(\zeta_{t+1})} + \frac{L}{2}\gamma_{t+1}^2 g_t^{(\zeta_{t+1})2} \end{aligned}$$

Taking the expectation with respect to ξ_{t+1} and using Assumption 6.3, we find

$$\mathbb{E}_{\xi_{t+1}}[f(\theta_{t+1}) - f^\star] \leq f(\theta_t) - f^\star - \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t)\tilde{g}_t^{(\zeta_{t+1})} + \frac{L}{2}\gamma_{t+1}^2 \mathbb{E}_{\xi_{t+1}}[g_t^{(\zeta_{t+1})2}]$$

where $\tilde{g}_t = E_\xi[g_{h_{t+1}}(\theta_t, \xi)]$. We use the inequality $2ab \geq a^2 - (a-b)^2$ and Assumption 6.1 to get

$$\begin{aligned} 2\nabla_{\zeta_{t+1}} f(\theta_t)\tilde{g}_t^{(\zeta_{t+1})} &\geq \nabla_{\zeta_{t+1}} f(\theta_t)^2 - (\nabla_{\zeta_{t+1}} f(\theta_t) - \tilde{g}_t^{(\zeta_{t+1})})^2 \\ &\geq \nabla_{\zeta_{t+1}} f(\theta_t)^2 - \max_{k=1,\dots,d} (\partial_k f(\theta_t) - \tilde{g}_t^{(k)})^2 \\ &\geq \nabla_{\zeta_{t+1}} f(\theta_t)^2 - c^2 h_{t+1}^2 \end{aligned}$$

We also have, invoking Assumption 6.13, that

$$\mathbb{E}_{\xi_{t+1}}[g_t^{(\zeta_{t+1})2}] \leq \max_{k=1,\dots,d} \mathbb{E}_{\xi_{t+1}}[g_t^{(k)2}] \leq 2\mathcal{L}(f(\theta_t) - f^\star) + \sigma^2.$$

We finally obtain that

$$\begin{aligned} &\mathbb{E}_{\xi_{t+1}}[f(\theta_{t+1}) - f^\star] \\ &\leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t)^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2. \end{aligned}$$

Apply Robbins-Siegmund Theorem to obtain that almost surely

$$\sum_{t\geq 0} \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t)^2 = \sum_{t\geq 0} \gamma_{t+1}\|\nabla f(\theta_t)\|_\infty^2 < \infty.$$

Proof of condition (ii). Note that from the proof of Theorem 1, we already have $\sum_{t\geq 0}\gamma_{t+1}(g_t(\omega) - \nabla f(\theta_t(\omega))) < \infty$, so using that

$$\|C(\zeta_{t+1})(g_t(\omega) - \nabla f(\theta_t(\omega)))\|_2 \leq \|(g_t(\omega) - \nabla f(\theta_t(\omega)))\|_2,$$

we deduce the convergence $\sum_{t\geq 0}\gamma_{t+1}C(\zeta_{t+1})(g_t(\omega) - \nabla f(\theta_t(\omega))) < \infty$ which gives, in virtue of Lemma 6.30 the result $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$. □

Part (b) gradient weights. Here we assume that the weights of the coordinate sampling policy are proportional to any norm of the current gradient: $C_t \propto (|\partial_k f(\theta_t)|^q)_{1\leq k\leq d}$ with $q > 0$. As before, the proof follows from applying Lemma 6.30. The proof of condition (i) relies on the equivalence of the norms in finite dimension.

Proof of condition (i). From the proof of Theorem 6.15, we get

$$\begin{aligned} &\mathbb{E}_{\xi_{t+1}}[f(\theta_{t+1}) - f^\star] \\ &\leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t)^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2. \end{aligned}$$

Taking the expectation with respect to ζ_{t+1}, we get

$$\mathbb{E}_t[f(\theta_{t+1}) - f^\star] \leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\sum_{k=1}^d p_{t,k}\partial_k f(\theta_t)^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2.$$

Apply Robbins-Siegmund Theorem to obtain $\sum_{t\geq 0}\gamma_{t+1}\nabla f(\theta_t)^\top C_t \nabla f(\theta_t) < \infty$ almost surely. Now observe that since $C_t \propto (|\partial_k f(\theta_t)|^q)_{1\leq k\leq d}$, it means that for all $k = 1, \dots, d$ we have $p_{t,k} \propto |\partial_k f(\theta_t)|^q / \|\nabla f(\theta_t)\|_q^q$ and

$$\nabla f(\theta_t)^\top C_t \nabla f(\theta_t) = \sum_{k=1}^d p_{t,k}\partial_k f(\theta_t)^2 \propto \sum_{k=1}^d \frac{|\partial_k f(\theta_t)|^q}{\|\nabla f(\theta_t)\|_q^q}\partial_k f(\theta_t)^2 \propto \frac{\|\nabla f(\theta_t)\|_{q+2}^{q+2}}{\|\nabla f(\theta_t)\|_q^q}.$$

All norms are equivalent on $\mathbb{R}^d$ and using Hölder's inequality we have for $0 < p < q$ that $\|\cdot\|_l \leq d^{1/p-1/q}\|\cdot\|_q$ so the last term is lower bounded as

$$\frac{\|\nabla f(\theta_t)\|_{q+2}^{q+2}}{\|\nabla f(\theta_t)\|_q^q} \geq C\|\nabla f(\theta_t)\|_{q+2}^2 \quad \text{with } C = d^{-2/(q+2)},$$

and again using the equivalence of the norms we get the square of the infinity norm $\nabla f(\theta_t)^\top C_t \nabla f(\theta_t) \propto \|\nabla f(\theta_t)\|_\infty^2$ which finally proves

$$(i)\ \sum_{t\geq 0}\gamma_{t+1}\|\nabla f(\theta_t)\|_\infty^2 < \infty.$$

Proof of condition (ii). It is the same as for *Part (a) maximum gradient*. We deduce the convergence $\sum_{t\geq 0}\gamma_{t+1}C(\zeta_{t+1})(g_t(\omega) - \nabla f(\theta_t(\omega))) < \infty$ which gives, in virtue of Lemma 6.30 the result $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$. □

6.A.5 Proof of Theorem 6.18

Similarly to the proof of Theorem 6.14, we rely on Lemma 6.31 where g_{t-1} is replaced by $C(\zeta_t)g_{t-1}$. Therefore we need to check that, with probability 1, it holds that

$$(i)\ \sum_{t\geq 0}\gamma_{t+1}\|\nabla f(\theta_t(\omega))\|_2^2 < \infty \quad (ii)\ \sum_{t\geq 0}\gamma_{t+1}(C(\zeta_t)g_t(\omega) - C_t\nabla f(\theta_t(\omega))) < \infty.$$

Proof of condition (i). From the proof of Theorem 6.15, we get

$$\mathbb{E}_{\xi_{t+1}}[f(\theta_{t+1}) - f^\star] \leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\nabla_{\zeta_{t+1}} f(\theta_t)^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2.$$

Taking the expectation with respect to ζ_{t+1} and using that $\min_{k=1,\dots,d} p_{t,k} \geq \beta$ gives

$$\begin{aligned}\mathbb{E}_t[f(\theta_{t+1}) - f^\star] &\leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\sum_{k=1}^d p_{t,k}\partial_k f(\theta_t)^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2 \\ &\leq (1 + L\mathcal{L}\gamma_{t+1}^2)(f(\theta_t) - f^\star) - \gamma_{t+1}\beta\|\nabla f(\theta_t)\|_2^2/2 + c^2\gamma_{t+1}h_{t+1}^2/2 + \frac{L}{2}\gamma_{t+1}^2\sigma^2,\end{aligned}$$

and Robbins-Siegmund Theorem allows to conclude $\sum_{t\geq 0}\gamma_{t+1}\|\nabla f(\theta_t)\|_2^2 < +\infty$.

Proof of condition (ii). Again, we place ourselves on the event Ω_0 and omit the ω in notation for ease of clarity. First, note that $\|g_t\|_2^2 \leq g_t^{(\zeta_{t+1})2} \leq \|g_t\|_2^2$. As a consequence, $\sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|C(\zeta_{t+1})g_t\|_2^2] \leq \sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|g_t\|_2^2]$ and this last series converges as shown in the proof of Theorem 6.14. Now observe that condition (ii) is satisfied as soon as

$$(a)\ \|\sum_{t\geq 0}\gamma_{t+1}(C(\zeta_{t+1})g_t - \mathbb{E}_t[C(\zeta_{t+1})g_t])\|_2 < \infty$$
$$(b)\ \|\sum_{t\geq 0}\gamma_{t+1}(\mathbb{E}_t[C(\zeta_{t+1})g_t] - C_t\nabla f(\theta_t))\|_2 < \infty$$

Note that equation (a) involves martingale increments whose quadratic variation satisfies

$$\sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|C(\zeta_{t+1})g_t - \mathbb{E}_t[C(\zeta_{t+1})g_t]\|_2^2] \leq \sum_{t\geq 0}\gamma_{t+1}^2\mathbb{E}_t[\|C(\zeta_{t+1})g_t\|^2] < \infty,$$

which proves Equation (a). Finally the term in equation (b) is bounded using assumption 6.1 and $\|C_t\|_2 \leq 1$. We have $\sum_{t\geq 0}\gamma_{t+1}^2\|\mathbb{E}_t[C(\zeta_{t+1})g_t] - C_t\nabla f(\theta_t)\|_2^2 \leq c^2\sum_{t\geq 0}\gamma_{t+1}^2 h_t^2 < \infty$ which finally proves condition (ii) and gives, in virtue of Lemma 6.31 that $\nabla f(\theta_t) \to 0$ almost surely as $t \to +\infty$.

6.A.6 Proof of Theorem 6.21

From the proof of Theorem 6.18 and using β as a uniform lower bound on β_{t+1}, we have

$$\mathbb{E}_t\left[f(\theta_{t+1}) - f^\star\right] \leq \left(1 + L\mathcal{L}\gamma_{t+1}^2\right)\left[f(\theta_t) - f^\star\right] - \gamma_{t+1}\beta\|\nabla f(\theta_t)\|_2^2 + \frac{\sigma^2 L + c^2}{2}\gamma_{t+1}^2.$$

Inject the PL inequality $\|\nabla f(\theta_t)\|_2^2 \geq 2\mu(f(\theta_t) - f(\theta^*))$ from Assumption 6.20 to have

$$\mathbb{E}_t\left[f(\theta_{t+1}) - f^\star\right] \leq \left(1 - 2\mu\beta\gamma_{t+1} + L\mathcal{L}\gamma_{t+1}^2\right)\left[f(\theta_t) - f^\star\right] + \frac{\sigma^2 L + c^2}{2}\gamma_{t+1}^2.$$

Define $\delta_t = \mathbb{E}\left[f(\theta_t) - f^\star\right]$ to finally obtain the recursion equation

$$\delta_t \leq \left(1 - 2\mu\beta\gamma_t + L\mathcal{L}\gamma_t^2\right)\delta_{t-1} + \frac{\sigma^2 L + c^2}{2}\gamma_t^2$$

Applying the same result from (Moulines and Bach, 2011) with the family of functions φ_α defined by $\varphi_\alpha(t) = \alpha^{-1}(t^\alpha - 1)$ if $\alpha \neq 0$ and $\varphi_\alpha(t) = \log(t)$ if $\alpha = 0$ along with the learning rates $\gamma_t = \gamma t^{-\alpha}$.

$$\delta_t \leq \begin{cases} 2\exp\left(2L\mathcal{L}\gamma^2\varphi_{1-2\alpha}(t)\right)\exp\left(-\frac{\mu\beta\gamma}{4}t^{1-\alpha}\right)\left(\delta_0 + \frac{\sigma^2+2c^2}{2\mathcal{L}}\right) + \frac{\gamma(\sigma^2 L+2c^2)}{\mu\beta}t^{-\alpha} & \text{if } \alpha < 1 \\ 2\exp\left(L\mathcal{L}\gamma^2\right)\left(\delta_0 + \frac{\sigma^2+2c^2}{2\mathcal{L}}\right)t^{-\mu\beta\gamma} + \left(\frac{\sigma^2 L}{2} + c^2\right)\gamma^2\varphi_{\mu\beta\gamma/2-1}(t)t^{-\mu\beta\gamma/2} & \text{if } \alpha = 1 \end{cases}$$

6.A.7 Proof of Theorem 6.25

Starting from $G_0 = (0, \dots, 0)$, the total average gain G_n is updated in a online manner during the exploitation phase and collects all the empirical sums of the gradient gradient estimates as

$$G_n = \frac{1}{nT} \sum_{t=1}^{nT} C_t^{-1} C(\zeta_{t+1}) g(\theta_t, \xi_{t+1}), \qquad \mathbb{E}\left[G_n\right] = \frac{1}{nT} \sum_{t=1}^{nT} \nabla f(\theta_t).$$

The goal is to show that $G_n \to 0$ using martingale properties. Thanks to Theorem 6.18, we have the almost sure convergence $\theta_t \to \theta^*$ which gives, since $\theta \mapsto \nabla f(\theta)$ is continuous, that $\nabla f(\theta_t) \to 0$ almost surely. Applying Cesaro's Lemma, it holds that $\mathbb{E}\left[G_n\right] \to 0$. It is enough to consider the difference $\left(G_n^{(k)} - \mathbb{E}\left[G_n^{(k)}\right]\right)$ for each $k \in [\![1, d]\!]$. Introducing the martingale increments

$$\Delta_{t+1}^{(k)} = \frac{g(\theta_t, \xi_{t+1})^{(k)}}{p_t^{(k)}} \mathbb{1}_{\{\zeta_{t+1}=k\}} - \partial_k f(\theta_t), \qquad \mathbb{E}\left[\Delta_{t+1}^{(k)} | \mathcal{F}_t\right] = 0.$$

It remains to show that, with probability 1,

$$G_n^{(k)} - \mathbb{E}\left[G_n^{(k)}\right] = \frac{1}{nT} \sum_{t=1}^{nT} \Delta_{t+1}^{(k)} \to 0.$$

Or equivalently, that, for each coordinate $k \in [\![1, d]\!]$

$$\sum_{t=1}^{nT} \Delta_{t+1}^{(k)} = o(n). \tag{6.8}$$

The latter being a sum of martingale increments, we are in position to apply the strong law of large numbers for martingales which can be find as Assertion 2 of Theorem 1.18 in (Bercu et al., 2015). Using Assumption 6.13, there exist $0 \leq \mathcal{L}, \sigma^2 < \infty$ such that almost surely

$$\forall t \in \mathbb{N}, \quad \mathbb{E}\left[(g(\theta_t, \xi_{t+1})^{(k)})^2 | \mathcal{F}_t\right] \leq 2\mathcal{L}\left(f(\theta_t) - f^\star\right) + \sigma^2.$$

Using the almost sure convergence $\theta_t \to \theta^\star$, we deduce that there is exist a compact set K which contains the sequence of iterates $(\theta_t)_{t\in\mathbb{N}}$ and using that f is continuous gives the upper bound

$$\forall k \in [\![1, d]\!] \quad \mathbb{E}\left[(g(\theta_t, \xi_{t+1})^{(k)})^2 | \mathcal{F}_t\right] \leq M = 2\mathcal{L} \sup_{\theta \in K} (f(\theta) - f(\theta^*)) + \sigma^2.$$

Hence, the quadratic variation is bounded as follows

$$\begin{aligned}
\sum_{t=1}^{nT} \mathbb{E}\left[(\Delta_{t+1}^{(k)})^2 | \mathcal{F}_t\right] &\leq \sum_{t=1}^{nT} \mathbb{E}\left[\left(\frac{g(\theta_t, \xi_{t+1})^{(k)}}{p_t^{(k)}}\right)^2 | \mathcal{F}_t\right] \\
&\leq (d/\lambda)^2 \sum_{t=1}^{nT} \mathbb{E}[(g(\theta_t, \xi_{t+1})^{(k)})^2 | \mathcal{F}_t] \\
&\leq (d/\lambda)^2 nTM.
\end{aligned}$$

Equation (6.8) follows from applying the previously mentioned law of large number.

6.B Additional Results

6.B.1 Almost sure convergence under stronger assumptions

Similary to Gadat et al. (2018), we consider some stronger assumptions where the function f is coercive and there exists a unique stationary point $\theta^\star$. In such framework, the sequences of iterates $(\theta_t)_{t\geq 0}$ obtained by both SGD and SCGD satisfy $\theta_t \to \theta^\star$ almost surely as $t \to +\infty$.

f is coercive and $\{\theta \in \mathbb{R}^d : \nabla f(\theta) = 0\} = \{\theta^\star\}$. Following the proofs of Theorems 6.14 and 6.18, we may apply Robbins-Siegmund Theorem. There exists an event $\Omega_0 \subset \Omega$ such that, $\mathbb{P}(\Omega_0) = 1$ and for every $\omega \in \Omega_0$, $\limsup_t f(\theta_t(\omega)) < \infty$ and the series $\sum_t \eta_{t+1}\|\nabla f(\theta_t(\omega))\|_2^2$ converges (where $\eta_t = \gamma_t$ for SGD and $\eta_t = \gamma_t\beta_t$ for SCGD). Since $\lim_{\|\theta\|\to\infty} f(\theta) = \infty$, we deduce that for every $\omega \in \Omega_0$, the sequence $(\theta_t(\omega))_{t\geq 0}$ is bounded in $\mathbb{R}^p$. Therefore the limit set $\chi_\infty(\omega)$ (set of accumulation points) of the sequence $(\theta_t(\omega))$ is non-empty. The convergence of the series $\sum_t \eta_{t+1}\|\nabla f(\theta_t(\omega))\|_2^2 < \infty$ along with the condition $\sum_t \eta_{t+1} = +\infty$ only implie that : $\liminf_{t\to\infty} \|\nabla f(\theta_t(\omega))\|_2^2 = 0, \quad \mathbb{P}-a.s.$
Hence, since $\theta \mapsto \nabla f(\theta)$ is continuous, there exits a limit point $\theta_\infty(\omega) \in \chi_\infty(\omega)$ such that $\|\nabla f(\theta_\infty(\omega))\|_2^2 = 0$, *i.e.*, $\nabla f(\theta_\infty(\omega)) = 0$. Because the set of solutions $\{\theta \in \mathbb{R}^p, \nabla f(\theta) = 0\}$ is reduced to the singleton $\{\theta^\star\}$, we have $\theta_\infty(\omega) = \theta^\star$. Since $(f(\theta_t(\omega)))$ converges, it implies that $\lim_t f(\theta_t(\omega)) = f^\star$ and for every limit point $\theta \in \chi_\infty(\omega)$, we have $f(\theta) = f^\star$. Since the set $\{\theta \in \mathbb{R}^d, f(\theta) = f^\star\}$ is equal to $\{\theta^\star\}$, the limit set $\chi_\infty(\omega)$ is also reduced to $\{\theta^\star\}$.

6.B.2 Almost sure convergence of MUSKETEER

By definition, we have for all $k \in [\![1, d]\!]$,

$$p_{t+1}^{(k)} = (1 - \lambda_t)\varphi(G_t)^{(k)} + \lambda_t \frac{1}{d}$$

implying that $\beta_{t+1} = \min_{k\in[\![1,d]\!]} p_t^{(k)} \geq \lambda_t/d$. As a consequence, as soon as $\sum_{t\geq 1} \lambda_t\gamma_t = +\infty$, the assumption $\sum_{t\geq 1} \beta_t\gamma_t = +\infty$ is satisfied. Applying Theorem 6.18 we obtain the almost sure convergence of MUSKETEER. The condition $\sum_{t\geq 1} \lambda_t\gamma_t = +\infty$ is easily satisfied with a fixed value $\lambda_t \equiv \lambda$ in the mixture update and one can also use a slowly decreasing sequence, e.g. $\lambda_t = 1/\log(t)$.

6.B.3 Regret analysis in the convex case

In order to better understand the benefits of the adaptive sampling strategies over standard uniform sampling, let us consider a particular setting where the objective function f is convex. The following proposition available in Namkoong et al. (2017) presents a regret analysis which is useful for interpretability.

Proposition 6.32 (Regret analysis for convex f and unbiased estimates)**.** *Assume that f is convex and consider the sequence of iterates obtained by $\theta_{t+1} = \theta_t - \gamma C_t^{-1} C(\zeta_{t+1}) g_t$ with constant step size $\gamma > 0$. We have*

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} \theta_t \right) - f(\theta^\star) \right] \leq \frac{\|\theta^\star\|^2}{2\gamma T} + \frac{\gamma}{2T} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{d} \frac{|\partial_k f(\theta_t)|^2}{p_t^{(k)}} \right].$$

Proof Assume that the objective f is convex and consider the average estimate $\bar{\theta}_T = \frac{1}{T}\sum_{t=1}^{T} \theta_t$. along with the following quantity: $S(f, \hat{\theta}) = \mathbb{E}[f(\hat{\theta})] - f^\star$. Using convexity we have on the one hand $f(\theta_t) - f^\star \leq \langle \theta_t - \theta^\star, \nabla f(\theta_t) \rangle$ and on the other hand

$$f(\bar{\theta}_T) - f^\star \leq \frac{1}{T} \sum_{t=1}^{T} \Big(f(\theta_t) - f^\star \Big)$$

which give together the following upper bound

$$f(\bar{\theta}_T) - f^\star \leq \frac{1}{T} \sum_{t=1}^{T} \langle \theta_t - \theta^\star, \nabla f(\theta_t) \rangle.$$

Using an unbiased gradient estimate v_t, *i.e.* $\mathbb{E}_t[v_t] = \nabla f(\theta_t)$, we can write

$$\mathbb{E}[f(\bar{\theta}_T)] - f^\star \leq \mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} \langle \theta_t - \theta^\star, \mathbb{E}_t[v_t]) \rangle \right].$$

The term in the expectation is bounded using Lemma 6.34 with $v_t = C_t^{-1} C(\zeta_{t+1}) g_t$ as

$$\frac{1}{T} \sum_{t=1}^{T} \langle \theta_t - \theta^\star, v_t \rangle \leq \frac{\|\theta^\star\|^2}{2\gamma T} + \frac{\gamma}{2T} \sum_{t=1}^{T} \|C_t^{-1} C(\zeta_{t+1}) g_t\|^2.$$

Take the expectation on both side to control the regret as

$$S(f, \bar{\theta}_T) \leq \frac{\|\theta^\star\|^2}{2\gamma T} + \frac{\gamma}{2T} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{d} \frac{|\partial_k f(\theta_t)|^2}{p_t^{(k)}} \right].$$

■

The term in expectation should be minimized with respect to the probability weights $p_t^{(k)}$. Intuitively, in order to maintain the overall sum as small as possible, the large gradient coordinates should be sampled more often, *i.e.* we would like to have $p_t^{(k)}$ large whenever $|\partial_k f(\theta_t)|^2$ is large. This is in line with the framework of coordinate smoothness discussed in Remark 6.12 and the work of Allen-Zhu et al. (2016).

(Uniform Coordinate Sampling) For all $k \in [\![1, d]\!]$, we have $p_t^{(k)} = 1/d$ so that

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{d} \frac{|\partial_k f(\theta_t)|^2}{p_t^{(k)}} \right] = \frac{d}{T} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{d} |\partial_k f(\theta_t)|^2 \right] = \frac{d}{T} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla f(\theta_t)\|^2 \right].$$

(MUSKETEER) For all $k \in [\![1, d]\!]$, we have $p_t^{(k)} = (1 - \lambda_{t-1})\varphi(G_{t-1})^{(k)} + \lambda_{t-1}/d$ so that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{k=1}^{d}\frac{|\partial_k f(\theta_t)|^2}{p_t^{(k)}}\right] = \frac{d}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{k=1}^{d}\frac{|\partial_k f(\theta_t)|^2}{(1-\lambda_{t-1})d\varphi(G_{t-1})^{(k)} + \lambda_{t-1}}\right],$$

where the denominator is stricly larger than 1 for all the coordinates associated to large gains. Indeed, let $k \in [\![1, d]\!]$ the index of such coordinate. Since it is a rewarding coordinate, the normalizing step implies that $\varphi(G_{t-1})^{(k)} > 1/p$ and $(1 - \lambda_{t-1})d\varphi(G_{t-1})^{(k)} + \lambda_{t-1} > 1$. This property translates the adaptive nature of the probability weights used in the MUSKETEER strategy.

6.B.4 Auxiliary Results

Theorem 6.33. *(Robbins and Siegmund, 1971) Consider a filtration $(\mathcal{F}_n)_{n\geq 0}$ and four sequences of random variables $(V_n)_{n\geq 0}, (W_n)_{n\geq 0}, (a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ that are adapted and non-negative. Assume that almost surely $\sum_k a_k < \infty$ and $\sum_k b_k < \infty$. Assume moreover that $\mathbb{E}[V_0] < \infty$ and $\forall n \in \mathbb{N} : \mathbb{E}[V_{n+1}|\mathcal{F}_n] \leq (1 + a_n)V_n - W_n + b_n$. Then it holds*

$$(a)\ \sum_k W_k < \infty\ a.s. \qquad (b)\ V_n \xrightarrow{a.s.} V_\infty, \mathbb{E}[V_\infty] < \infty. \qquad (c)\ \sup_{n\geq 0}\mathbb{E}[V_n] < \infty.$$

Lemma 6.34. *Let $\theta_1, \ldots, \theta_T$ be an arbitrary sequence of vectors. Any algorithm with initialization $\theta_1 = 0$ and update rule $\theta_{t+1} = \theta_t - \gamma v_t$ satisfies*

$$\sum_{t=1}^{T}\langle \theta_t - \theta^\star, v_t\rangle \leq \frac{\|\theta^\star\|^2}{2\gamma} + \frac{\gamma}{2}\sum_{t=1}^{T}\|v_t\|^2.$$

In particular, for $B, \rho > 0$, if we have $\|v_t\| \leq \rho$ and we set $\gamma = \sqrt{B^2/(\rho^2 T)}$ then for every $\theta^\star$ with $\|\theta^\star\| \leq B$, we have $T^{-1}\sum_{t=1}^{T}\langle \theta_t - \theta^\star, v_t\rangle \leq B\rho/\sqrt{T}$.

6.C Illustrative Example (stochastic first order)

We perform a comparison on a simple example in dimension $d = 2$ with the functions $f(x, y) = (x^2 + y^2)/2$ and $h(x, y) = x^2/2$. Note that the function h only depends on the first coordinate and an adaptive coordinate descent method should favor this direction. Figure 6.3 presents the optimization paths of the different methods: SGD, Uniform and MUKSTEER. With the function f which does not present any particular design or favorable descent direction, the Uniform and Musketeer policies perform as good as classical SGD. More interestingly, when dealing with the function h, our method MUSKETEER (red) finds that the horizontal direction associated to axis (Ox) is the relevant one for optimization. After collecting some information during the exploration phase, the probability weights got updated to favor the horizontal direction, leading to a faster convergence. For a visual demonstration of these optimization paths, please refer to the mp4-files *optimize_f.mp4* and *optimize_h.mp4* available online[2].

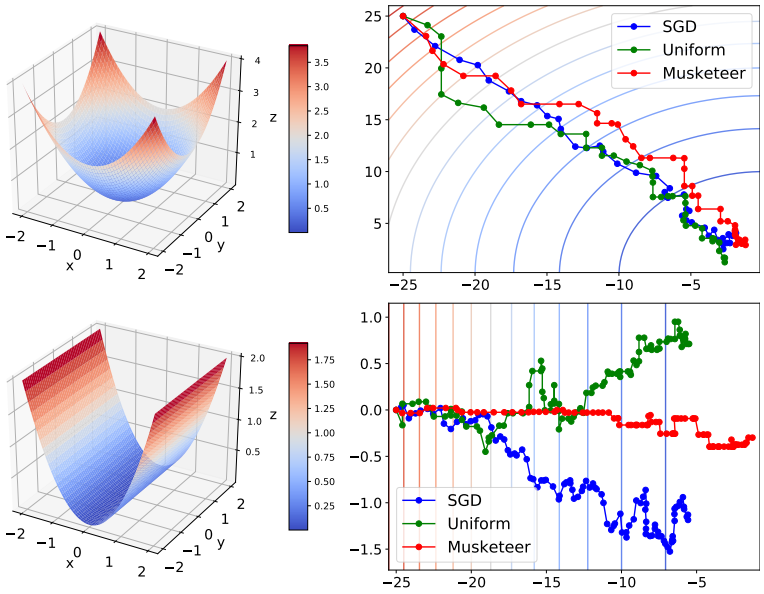


Figure 6.3 – Comparison of SGD/Uniform/Musketeer on simple 2D-examples

6.D Numerical Experiments Details

6.D.1 Regularized linear models

We consider the ERM paradigm with linear models, namely regularized regression problems with objectives of the form $f(\theta) = (1/n)\sum_{i=1}^{n} f_i(\theta) + \mu\|\theta\|^2$. Similarly to (Namkoong et al., 2017), we endow the data matrix X with a block structure. The columns are drawn as $X[:, k] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, d]\!]$. The parameters are set to $n = 10,000$ samples in dimension $d = 250$ with an exploration size equal to $T = \lfloor\sqrt{d}\rfloor = 15$. The regularization parameter is set to the classical value $\mu = 1/n$. We update the parameter vector with the optimal learning rate $\gamma_k = \gamma/(k+k_0)$ in the experiments. Other learning rates in the framework of stochastic first order meth-

[2] https://github.com/RemiLELUC/SCGD-Musketeer

ods are considered in Appendix 6.G.
• (zeroth-order) For the Ridge regression, we set $\gamma = 3, k_0 = 10$ and for the logistic regression $\gamma = 10, k_0 = 5$. The gradient estimate g is computed using queries of a function f_i where $i \sim \mathcal{U}([\![1, n]\!])$. We use the ℓ_1-reweighting with $\lambda_t = 1/\log(t)$ or softmax with $\lambda_n \equiv 0.5$, which both satisfy Assumption 6.17.
• (first order) The learning rate is equal to $\gamma_k = 1/k$ ($\gamma = 1, k_0 = 0$). The gradient estimate g is computed using mini-batches of size 8. The weighting parameter $\eta > 0$ in the softmax part of the probability weights is set to $\eta = 1$ and the parameter λ in Equation (6.7) is chosen as $\lambda_t = 1/\log(t)$ which satisfies the extended Robbins-Monro condition 6.17.

6.D.2 Neural Networks

Dataset description and parameter configuration. The three datasets in the experiments are popular publicly available deep learning datasets. The underlying machine learning task is the one of multi-label classification.

• **MNIST** (Deng, 2012): a database of handwritten digits with a training set of 60,000 examples and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field. Each training and test example is assigned to the corresponding handwritten digit between 0 and 9.

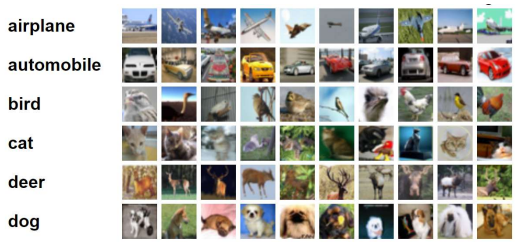
• **Fashion-MNIST** (Xiao et al., 2017): a dataset of Zalando's article images, composed of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. It shares the same image size and structure of training and testing splits as the MNIST database. Each training and test example is assigned to one of the following labels: T-shirt/top (0); Trouser (1); Pullover (2); Dress (3); Coat (4); Sandal (5); Shirt (6); Sneaker (7); Bag (8); Ankle boot (9).

• **Kuzushiji-MNIST**: This dataset is a drop-in replacement for the MNIST dataset (28x28 grayscale, 70,000 images), provided in the original MNIST format as well as a NumPy format. Since MNIST is restricted to 10 classes, one character here represents each of the 10 rows of Hiragana when creating Kuzushiji-MNIST.

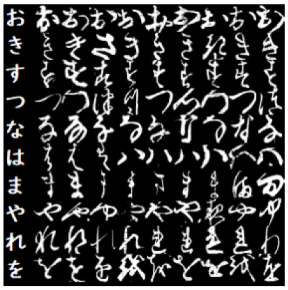
• **CIFAR10** (Krizhevsky et al., 2009): The CIFAR-10 dataset consists of $60,000$ 32×32 colour images in 10 classes, with $6,000$ images per class. There are $50,000$ training images and $10,000$ test images. The dataset is divided into five training batches and one test batch, each with $10,000$ images. The test batch contains exactly $1,000$ randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly $5,000$ images from each class. Each training and test example is assigned to one of the following labels: airplane (0); automobile (1); bird (2); cat (3); deer (4); dog (5); frog (6); horse (7); ship (8); truck (9).

(a) MNIST

(b) Fashion-MNIST



(c) CIFAR10



(d) MNIST

Figure 6.4 – Samples for Mnist, Fashion-Mnist, K-Mnist and CIFAR-10.

Two different neural networks are used in the experiments: one with linear layers for MNIST, Fashion-MNIST, K-MNIST another one with convolutional layers for CIFAR10. For the first network, the total number of parameters is $d = 55,050$. For the second network,the dimension is $d = 64,862$. In both cases, the exploration size is $T = \lfloor \sqrt{d} \rfloor$. In the experiments with stochastic first order methods, we use batches of coordinates with $m = d/10$.

6.D.3 Hyperparameters and Hardware.

Hyperparameters.

When training neural networks with linear layers, we use:

batch_size = 32; input_size = 28*28; hidden_size = 32; output_size = 64

• (zeroth-order) $\gamma = 10$ (Mnist and Fashion-Mnist) γ=15 (Kmnist); h = 0.01; ℓ_1 normalization with $\lambda_n = 1/\log(n)$; softmax normalization with $\lambda_n \equiv 0.2$ and $\eta = 5$.

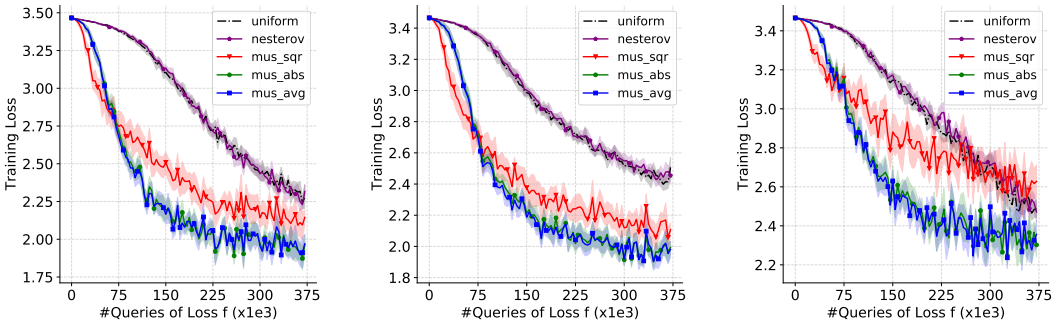
• (first order) $\gamma = 0.01$ (Mnist,Fashion-Mnist,Cifar10); normalization = softmax with $\eta \in \{1, 2, 10\}$; $\lambda_t = 0$ (only exponential weights).

Hardware.

The experiments of linear models are run using a processor Intel Core i7-10510U CPU 1.80GHz × 8.

The neural networks are trained using GPU from Google Colab (GPU: Nvidia K80 / T4; GPU Memory: 12GB/16GB; GPU Memory Clock: 0.82GHz/1.59GHz; Performance: 4.1 TFLOPS / 8.1 TFLOPS)

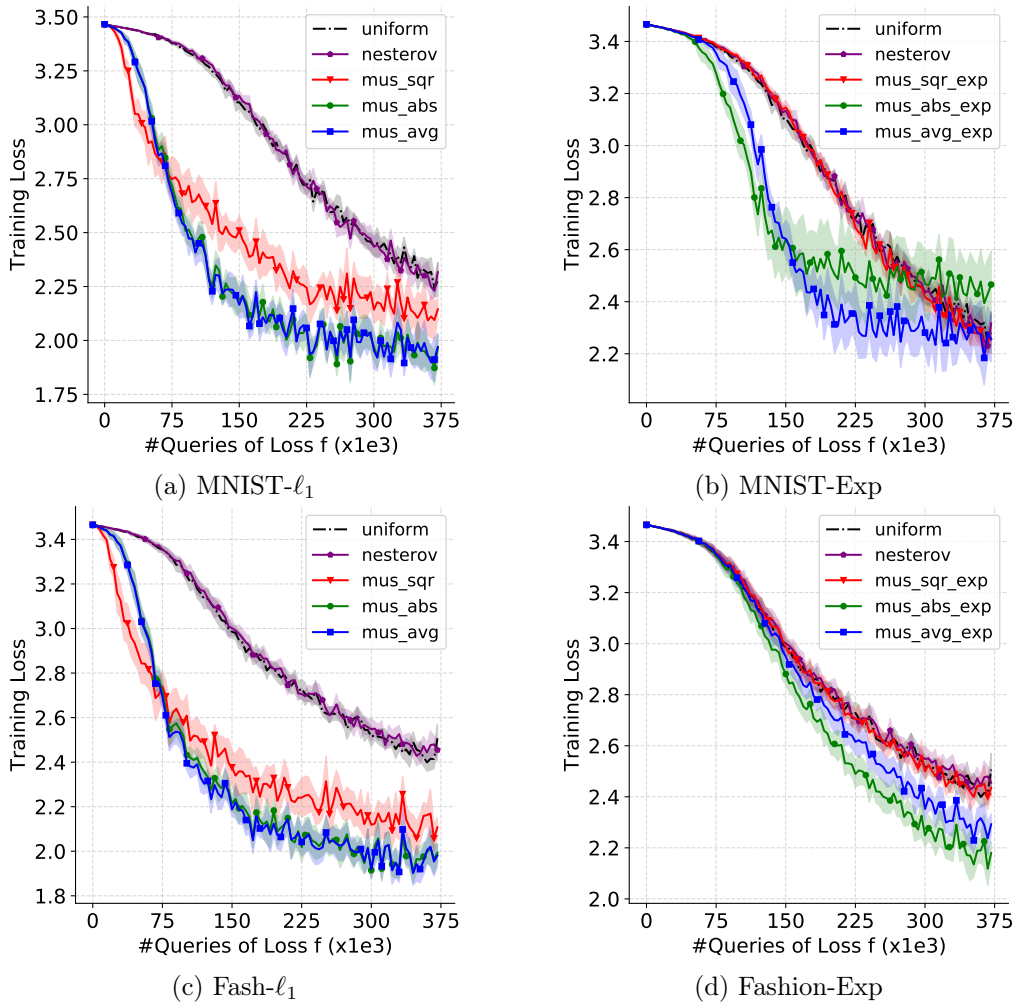
ZO Neural Networks with ℓ_1 normalization.



(a) MNIST (b) Fashion-MNIST (c) KMNIST

Figure 6.5 – Training Loss ZO Neural Networks with ℓ_1 normalization.

ZO Neural Networks, Comparison of ℓ_1 and Softmax normalizations.



(a) MNIST-ℓ_1 (b) MNIST-Exp

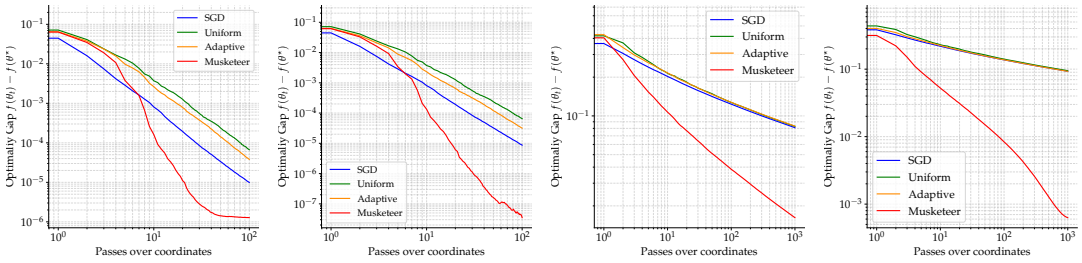
(c) Fash-ℓ_1 (d) Fashion-Exp

Figure 6.6 – Training Loss ZO Neural Networks with ℓ_1 and Softmax normalizations.

6.E Numerical Experiments with stochastic first order methods

In this section, we empirically validate the SCGD framework by running MUSKETEER and competitors on synthetic and real datasets problems with stochastic first order methods. First, we focus on ridge regression and regularized logistic regression problems adopting the data generation process of (Namkoong et al., 2017) in which the covariates exhibit a certain block structure. Second, MUSKETEER is employed to train different neural networks models on real datasets for multi-label classification task. From a practical point of view, the optimization procedure is implemented through a PyTorch optimizer which allows an easy deployment and integration.

Methods in competition. The set of methods in competition is restricted to stochastic coordinate-based methods along with standard SGD playing the role of the baseline. This choice allows an honest comparison as the parameter tuning can be the same for all methods. MUSKETEER is implemented according to Section 6.4 with an exploration size $T = \lfloor \sqrt{d} \rfloor$ and different values of η are used to feed the discussion on the adaptiveness. The method UNIFORM stands for the uniform coordinate sampling policy in SCGD. The method ADAPTIVE is the importance sampling based method described in Remark 6.9. This method is no longer part of the SCGD framework and corresponds to the one developed in (Wangni et al., 2018). Among the different methods, MUSKETEER is the only one exhibiting a bias when generating gradients. In all cases, $\theta_0 = (0, \dots, 0)^\top \in \mathbb{R}^d$ and the optimal SGD learning rate $\gamma_k = 1/k$ is used. For a fair comparison of SGD against SCGD, we normalize the number of passes over the coordinates: one SGD step updates the p coordinates of θ so we allow to take d steps for the coordinate-based methods in the mean time.



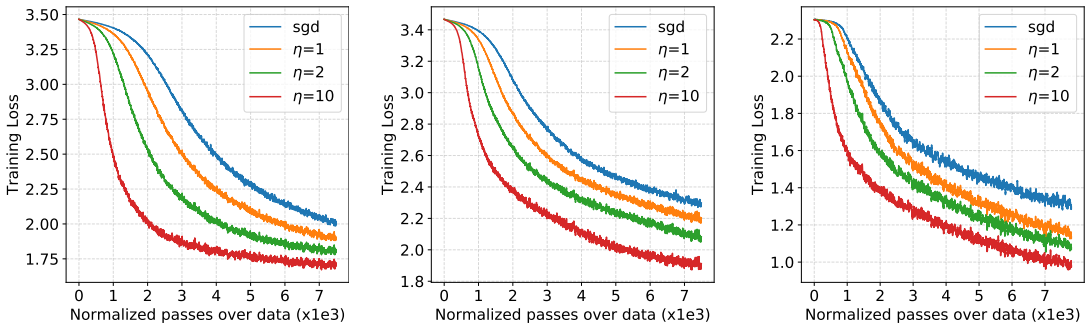
(a) Ridge $\alpha = 5$ (b) Ridge $\alpha = 10$ (c) Logistic $\alpha = 2$ (d) Logistic $\alpha = 5$

Figure 6.7 – $[f(\theta_t) - f(\theta^\star)]$ for Linear Models on Synthetic data with different block structures.

Linear models. We apply ERM to regularized regression and classification problems. Similarly to (Namkoong et al., 2017), we endow the data matrix X with a block structure. The columns are drawn as $X[:, k] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for $k \in [\![1, d]\!]$. The parameters are set to $n = 10,000$ samples in dimension $d = 250$ and $T = 15$. Figure 6.7 provides the graphs of the optimaliy gap $t \mapsto f(\theta_t) - f^\star$ averaged over 20 independent simulations for different values of $\alpha \in \{2; 5; 10\}$. First, note that the uniform sampling strategy shows similar performance to the classical SGD and that the (unbiased) importance sampling version ADAPTIVE is also of the same order. Besides, the clear winner is MUSKETEER as it offers the best performance in all configurations. Greater

values of α (stronger block structure) improve our relative performance with respect to the other methods as shown by Figures 6.7b and 6.7d.

Neural Networks. To asses the practical performance of MUSKETEER, we focus on the training of neural networks within the framework of multi-label classification. The datasets in the experiments are popular publicly available deep learning datasets: MNIST (Deng, 2012), Fashion-MNIST (Xiao et al., 2017) and CIFAR10 (Krizhevsky et al., 2009). Given an image, the goal is to predict its label among ten classes. Two different neural networks are used in the experiments: one with linear layers for MNIST and Fashion-MNIST ($d = 55,050$ and $T = 234$) , another one with convolutional layers for CIFAR10 ($d = 64,862$ and $T = 254$).



(a) MNIST (b) Fashion-MNIST (c) CIFAR-10

Figure 6.8 – Training Loss of SGD vs. MUSKETEER on real-world datasets.

Figure 6.8 compares the evolution of the training loss of SGD against MUSKETEER averaged over 10 independent simulations with different values of η. A great value of this parameter strengthens the adaptive scheme as it gives more importance to the weights in Equation (6.7), leading to stronger decrease of the objective function. Interestingly, the performance of MUSKETEER also benefit from such adaptive structure in terms on accuracy of the test set (see Table 6.1). This allows to quantify the statistical gain brought by MUSKETEER over SGD.

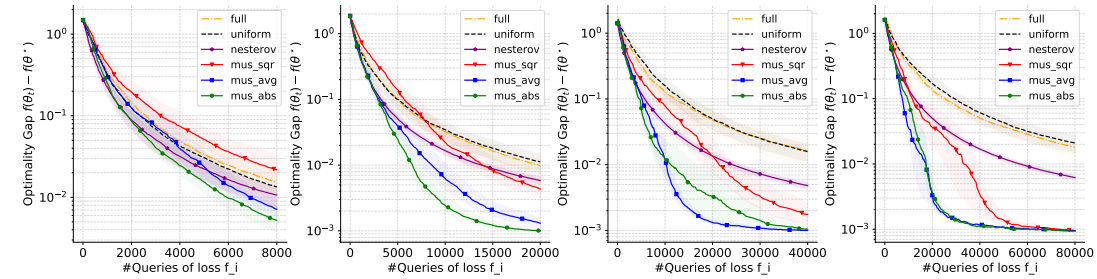
	SGD	$\eta = 1$	$\eta = 2$	$\eta = 10$
MNIST	84.7±1.0	86.7±0.5	88.9±0.4	**91.3±0.2**
FASHION	64.7±1.2	68.5±1.0	71.2±0.7	**77.1±0.8**
CIFAR10	51.4±1.4	57.7±0.8	59.7±1.0	**62.7±0.8**

Table 6.1 – Test Accuracy (in %).

6.F Further Numerical Experiments with zeroth-order methods

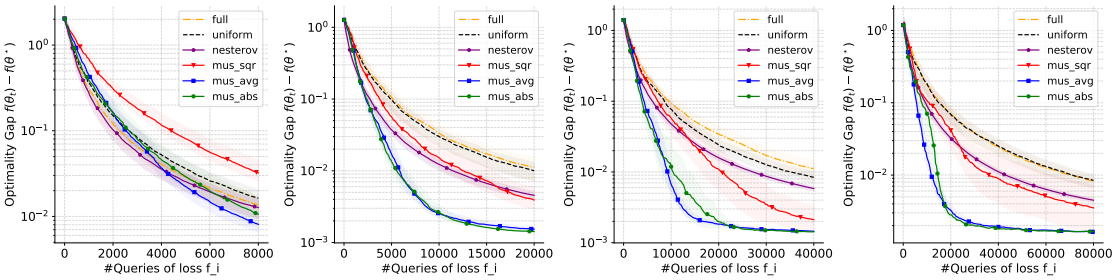
6.F.1 Ridge Regression (ℓ_1-reweighting) with different (n, d)

We consider the Ridge regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![0, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 10$ for the Ridge regression. The parameter α represents the block structure and is set to $\alpha = 5$. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs. The learning rates is the same for all methods, fixed to $\gamma_k = 1/(k+10)$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$. We use the ℓ_1 normalization in Equation (6.7) with $\lambda_n = 1/\log(n)$.



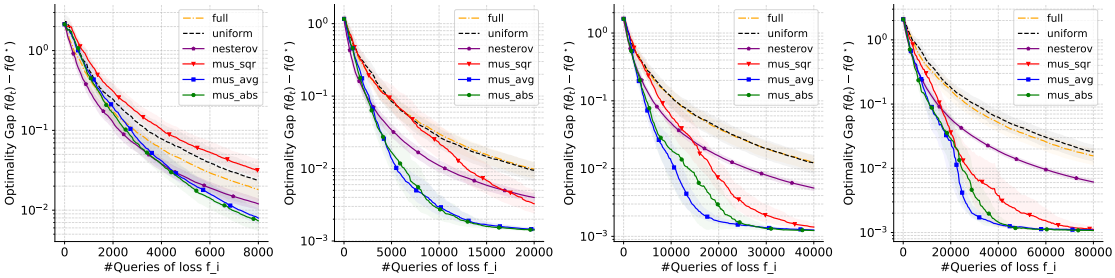
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.9 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.10 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 2000$ and $d = 20, 50, 100, 200$

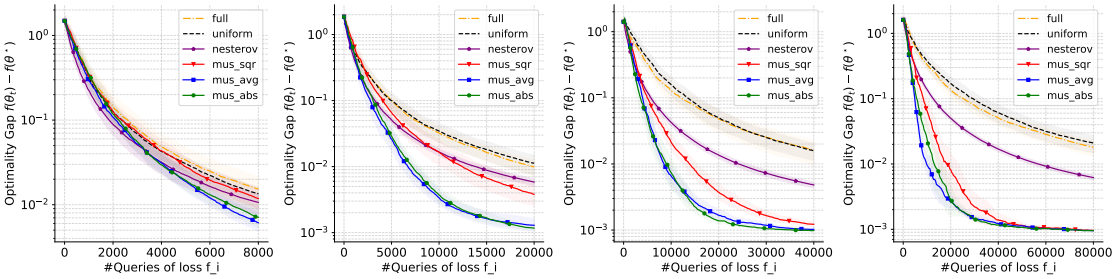


(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

Figure 6.11 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 5000$ and $d = 20, 50, 100, 200$

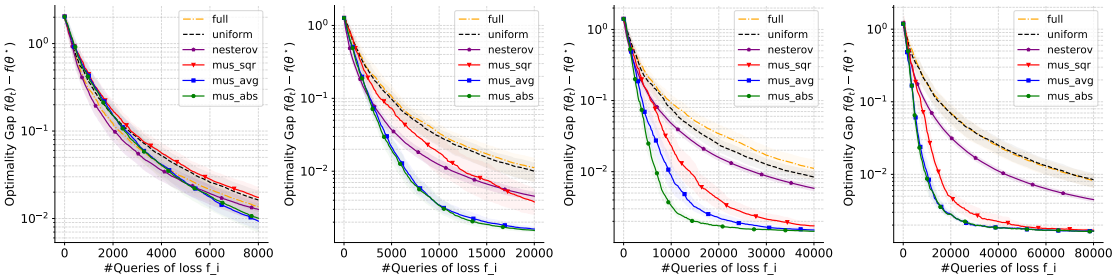
6.F.2 Ridge Regression (softmax reweighting) with different (n, d)

We consider the Ridge regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![0, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 10$ for the Ridge regression. The parameter α represents the block structure and is set to $\alpha = 5$. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs. The learning rates is the same for all methods, fixed to $\gamma_k = 1/(k+10)$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$. We use the softmax normalization in Equation (6.7) with $\lambda_n \equiv 0.5$ and $\eta = 1$.



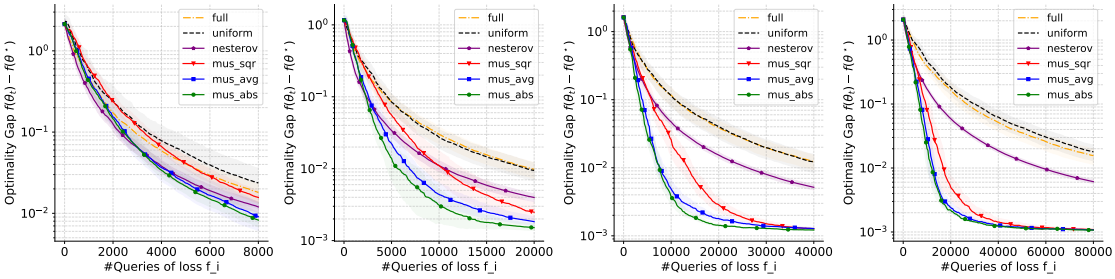
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.12 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.13 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 2000$ and $d = 20, 50, 100, 200$

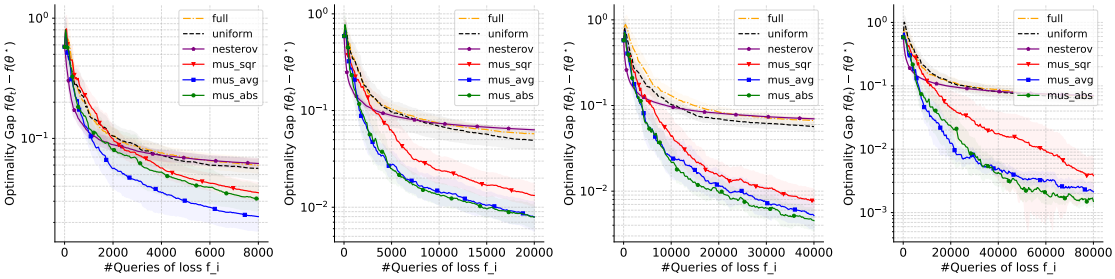


(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

Figure 6.14 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 5000$ and $d = 20, 50, 100, 200$

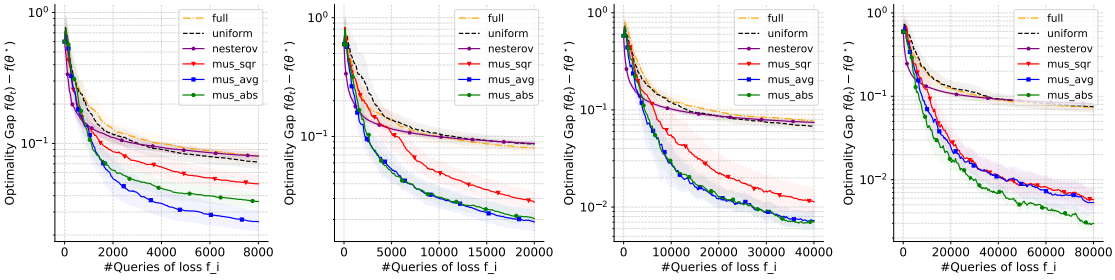
6.F.3 Logistic Regression (ℓ_1-reweighting) with different (n, d)

We consider the ℓ_2-Logistic regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 5$ for the Logistic regression. The parameter α represents the block structure and is set to $\alpha = 5$. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs. The learning rates is the same for all methods, fixed to $\gamma_k = 10/(k+5)$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$. We use the ℓ_1 normalization in Equation (6.7) with $\lambda_n = 1/\log(n)$.



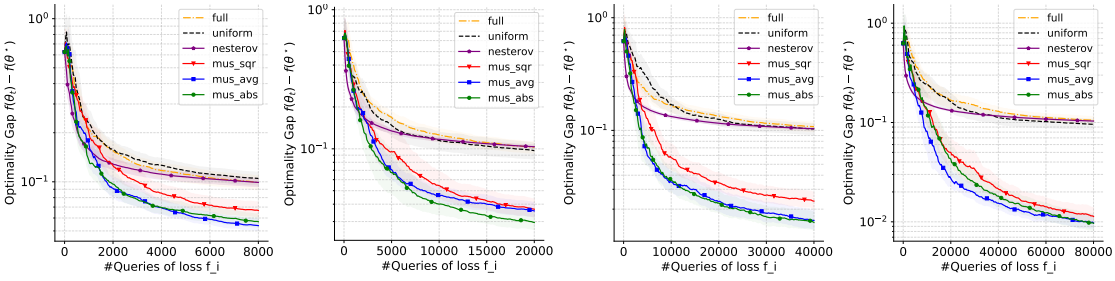
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.15 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.16 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 2000$ and $d = 20, 50, 100, 200$

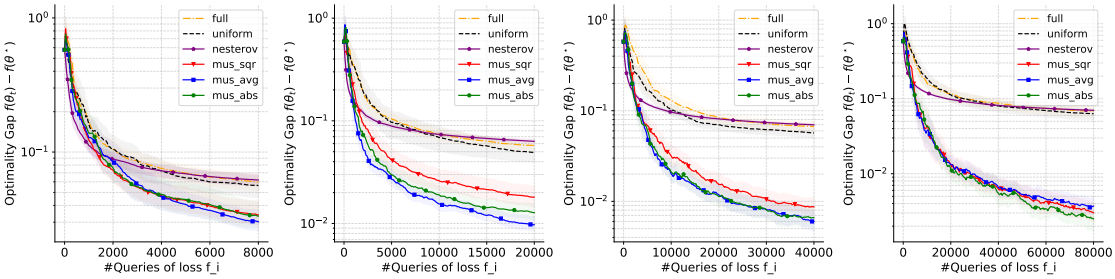


(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

Figure 6.17 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 5000$ and $d = 20, 50, 100, 200$

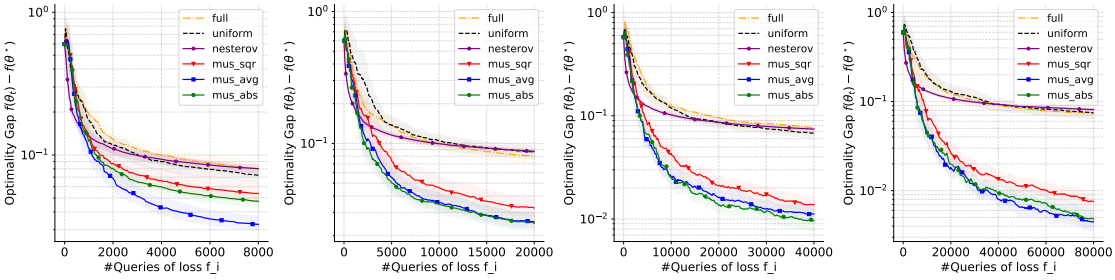
6.F.4 Logistic Regression (softmax reweighting) with different (n, d)

We consider the ℓ_2-Logistic regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 5$ for the Logistic regression. The parameter α represents the block structure and is set to $\alpha = 5$. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs. The learning rates is the same for all methods, fixed to $\gamma_k = 10/(k+5)$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$. We use the softmax normalization in Equation (6.7) with $\lambda_n \equiv 0.5$ and $\eta = 1$.



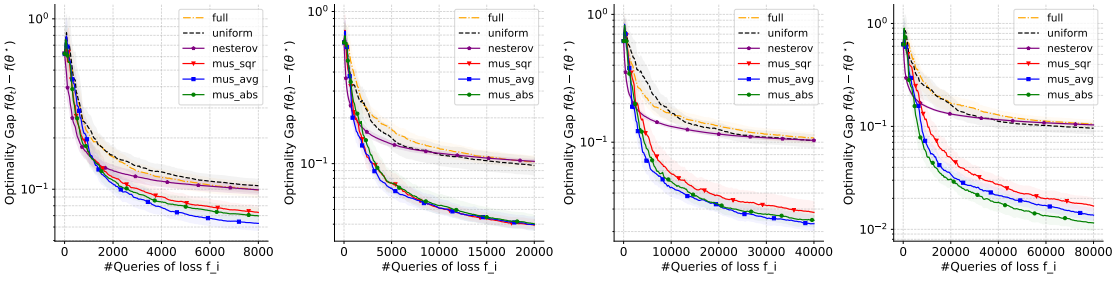
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.18 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.19 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 2000$ and $d = 20, 50, 100, 200$



(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

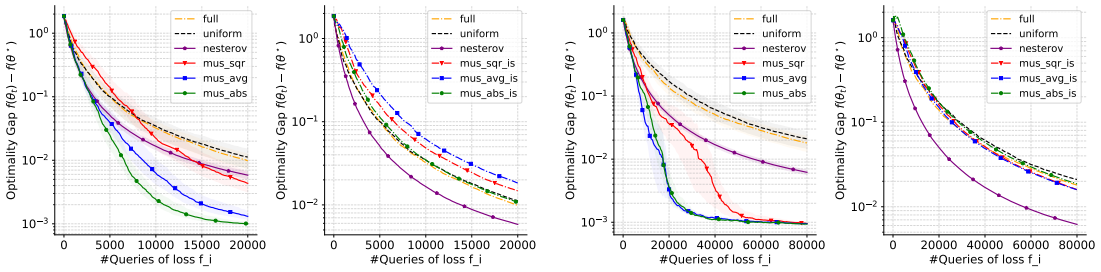
Figure 6.20 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 5000$ and $d = 20, 50, 100, 200$

6.F.5 Effect of Importance Sampling (IS) on Ridge Regression

We consider the same setting as in Subsection 6.F.1 and study the effect of using importance sampling weights in the update rule of MUSKETEER. Indeed, MUSKETEER update rule is defined with the following biased gradient estimate $\theta_{t+1} = \theta_t - \gamma_{t+1} C(\zeta_{t+1}) g_t$ and the importance sampling (IS) strategy consists in adding C_t^{-1} to reach an unbiased estimate

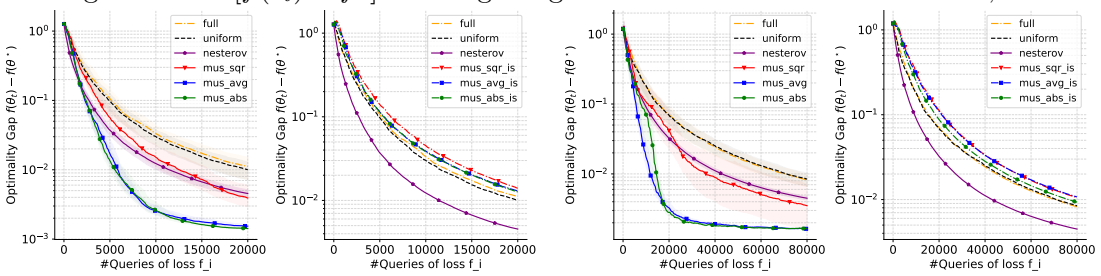
$$\theta_{t+1} = \theta_t - \gamma_{t+1} C_t^{-1} C(\zeta_{t+1}) g_t.$$

For the different configurations, we compare the MUSKETEER methods with their importance sampling counterparts. The Figures below show that the importance sampling methods perform similarly to the uniform coordinate sampling strategy and are therefore sub-optimal.



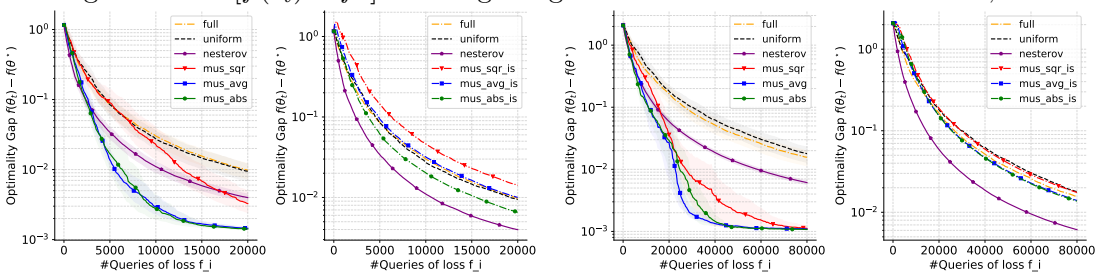
(a) $n = 1000, d = 50$ (b) $n = 1000, d = 200$

Figure 6.21 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 1000$ and $d = 50, 200$



(a) $n = 2000, d = 50$ (b) $n = 2000, d = 200$

Figure 6.22 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 2000$ and $d = 50, 200$



(a) $n = 5000, d = 50$ (b) $n = 5000, d = 200$

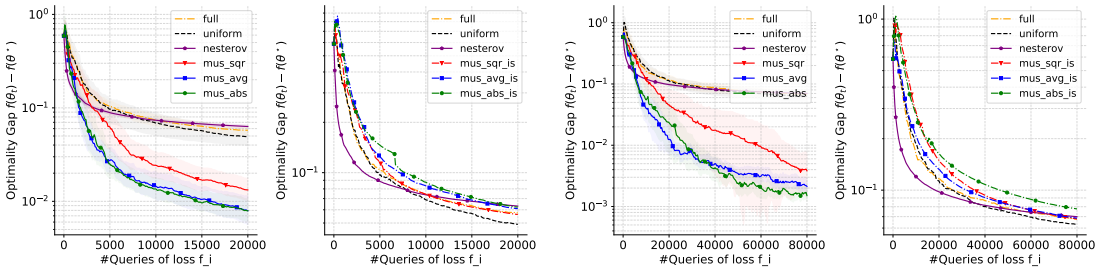
Figure 6.23 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 5000$ and $d = 50, 200$

6.F.6 Effect of Importance Sampling (IS) on Logistic Regression

We consider the same setting as in Subsection 6.F.3 and study the effect of using importance sampling weights in the update rule of MUSKETEER. Indeed, MUSKETEER update rule is defined with the following biased gradient estimate $\theta_{t+1} = \theta_t - \gamma_{t+1} C(\zeta_{t+1}) g_t$ and the importance sampling (IS) strategy consists in adding C_t^{-1} to reach an unbiased estimate

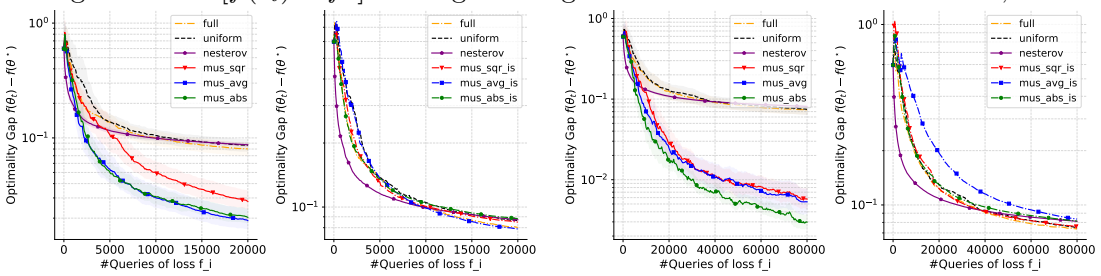
$$\theta_{t+1} = \theta_t - \gamma_{t+1} C_t^{-1} C(\zeta_{t+1}) g_t.$$

For the different configurations, we compare the MUSKETEER methods with their importance sampling counterparts. The Figures below show that the importance sampling methods perform similarly to the uniform coordinate sampling strategy and are therefore sub-optimal.



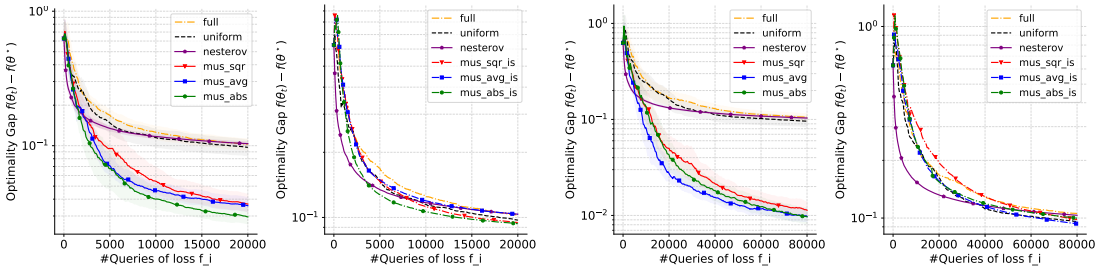
(a) $n = 1000, d = 50$ (b) $n = 1000, d = 200$

Figure 6.24 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 1000$ and $d = 50, 200$



(a) $n = 2000, d = 50$ (b) $n = 2000, d = 200$

Figure 6.25 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 2000$ and $d = 50, 200$



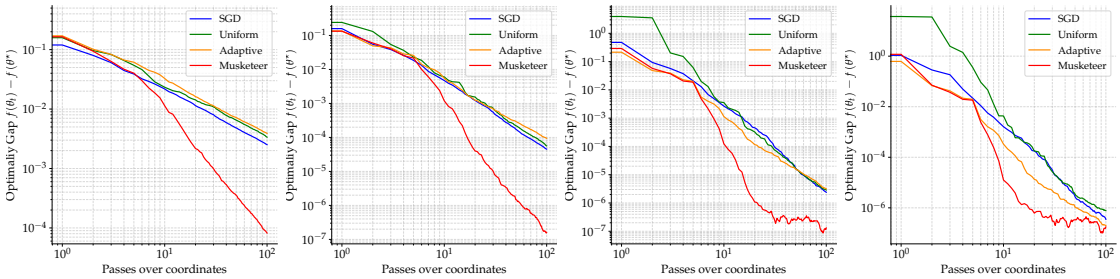
(a) $n = 5000, d = 50$ (b) $n = 5000, d = 200$

Figure 6.26 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 5000$ and $d = 50, 200$

6.G Further Experiments with stochastic first order methods

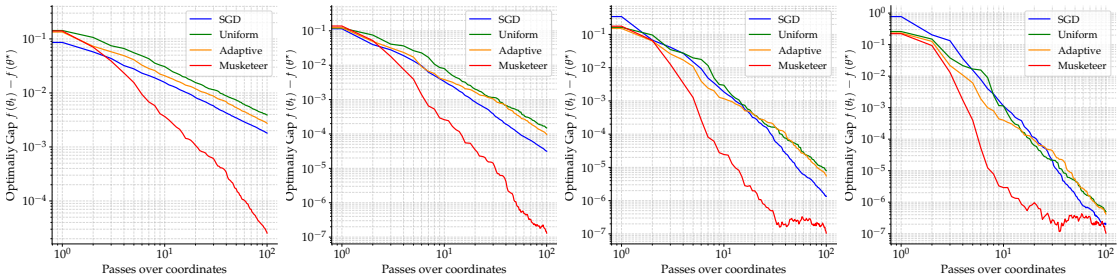
6.G.1 Comparing learning rates

This section investigates the effect of different learning rates $\gamma_k = \gamma/k$ with $\gamma \in \{0.5; 1; 1.5; 2\}$. It reveals a safe behavior of MUSKETEER as it performs better than the other methods in all configurations with a stronger difference when dealing with small values of γ. We consider the Ridge regression problem with regularization parameter $\mu = 1/n$ and run several experiments in the setting $n = 5,000$ samples and dimension $d \in \{20; 100; 200\}$. We endow the data matrix X with a block structure. The columns are drawn as $X[:, k] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, d]\!]$. The parameter α of block structure is $\alpha = 8$. The gradient estimate g is computed using mini-batches of size 4. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs for $N = 100$ iterations with normalized passes over coordinates.



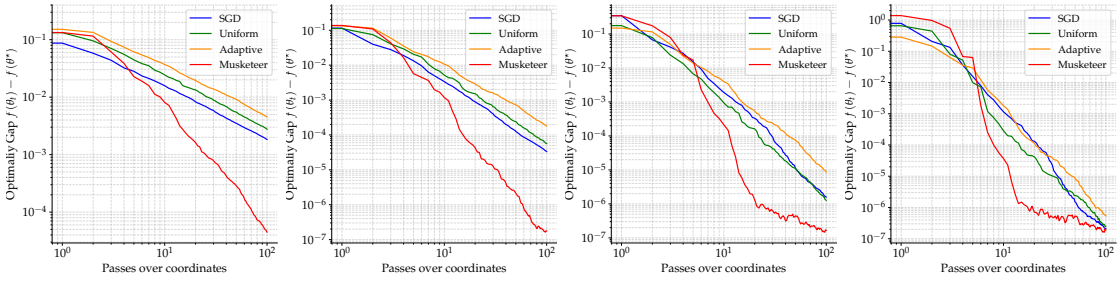
(a) $d = 20, \gamma = 0.5$ (b) $d = 20, \gamma = 1$ (c) $d = 20, \gamma = 1.5$ (d) $d = 20, \gamma = 02$

Figure 6.27 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $d = 20$ and $\gamma \in \{0.5; 1; 1.5; 2\}$



(a) $d = 100, \gamma = 0.5$ (b) $d = 100, \gamma = 1$ (c) $d = 100, \gamma = 1.5$ (d) $d = 100, \gamma = 02$

Figure 6.28 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $d = 100$ and $\gamma \in \{0.5; 1; 1.5; 2\}$

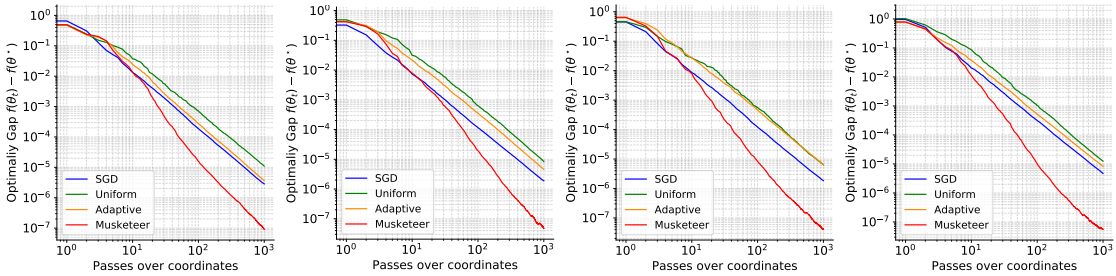


(a) $d = 200, \gamma = 0.5$ (b) $d = 200, \gamma = 1$ (c) $d = 200, \gamma = 1.5$ (d) $d = 200, \gamma = 02$

Figure 6.29 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $d = 200$ and $\gamma \in \{0.5; 1; 1.5; 2\}$

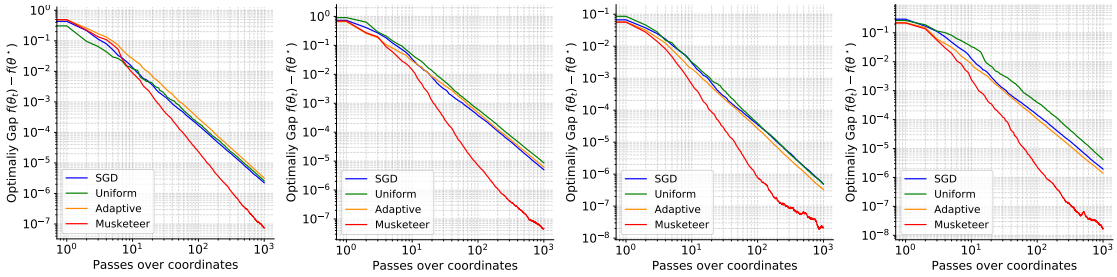
6.G.2 Ridge Regression with different settings of (n, d)

We consider the Ridge regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![0, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 5$ for the Ridge regression. The parameter α represents the block structure and is set to $\alpha = 10$. The data sampling process ξ of gradient estimate g is computed using mini-batches of size 8. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs for $N = 1000$ iterations with normalized passes over coordinates. The learning rates is the same for all methods, fixed to $\gamma_k = 1/k$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$.



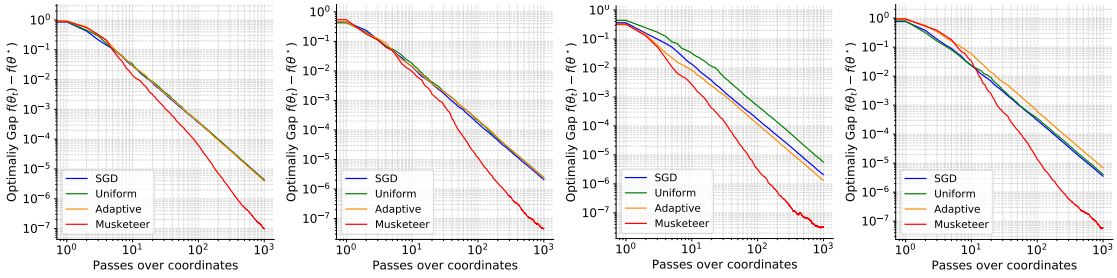
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.30 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.31 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 2000$ and $d = 20, 50, 100, 200$

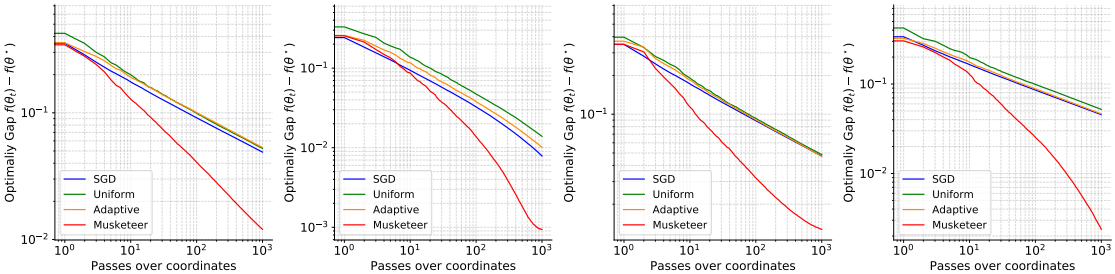


(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

Figure 6.32 – $[f(\theta_t) - f^\star]$ for Ridge Regression with $n = 5000$ and $d = 20, 50, 100, 200$

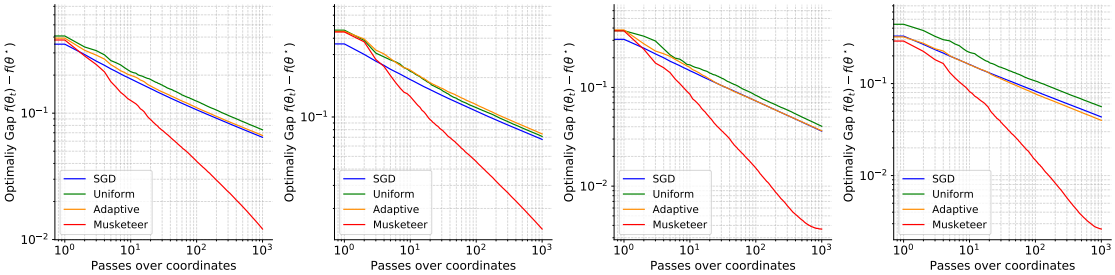
6.G.3 Logistic Regression with different settings of (n, d)

We consider the ℓ_2-Logistic regression problem with the classical regularization parameter value $\mu = 1/n$ and run several experiments in various settings of (n, d). We endow the data matrix X with a block structure. The columns are drawn as $X[:, kB+1 : kB+B] \sim \mathcal{N}(0, \sigma_k^2 I_n)$ with $\sigma_k^2 = k^{-\alpha}$ for all $k \in [\![1, (d/B)-1]\!]$. The parameter B is the block-size and is set to $B = 2$ for the Logistic regression. The parameter α represents the block structure and is set to $\alpha = 5$. The data sampling process ξ of gradient estimate g is computed using mini-batches of size 32. The different Figures below present the evolution of the optimality gap $t \mapsto [f(\theta_t) - f^\star]$ averaged over 20 independent runs for $N = 1000$ iterations with normalized passes over coordinates. The learning rates is the same for all methods, fixed to $\gamma_k = 1/k$. The different settings are: number of samples $n \in \{1,000; 2,000; 5,000\}$ and dimension $d \in \{20; 50; 100; 200\}$.



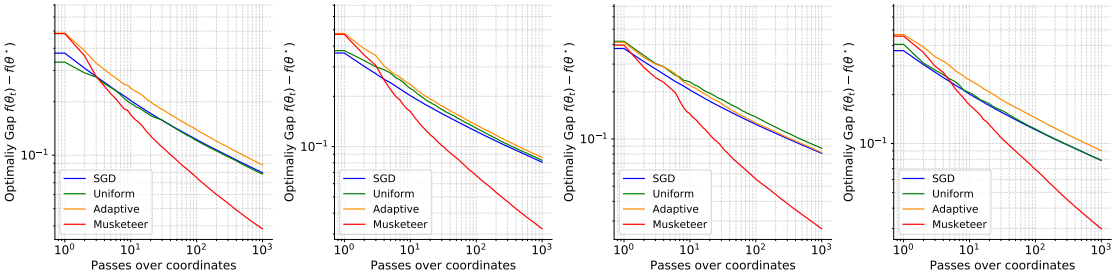
(a) $n = 1000, d = 20$ (b) $n = 1000, d = 50$ (c) $n = 1000, d = 100$ (d) $n = 1000, d = 200$

Figure 6.33 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 1000$ and $d = 20, 50, 100, 200$



(a) $n = 2000, d = 20$ (b) $n = 2000, d = 50$ (c) $n = 2000, d = 100$ (d) $n = 2000, d = 200$

Figure 6.34 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 2000$ and $d = 20, 50, 100, 200$



(a) $n = 5000, d = 20$ (b) $n = 5000, d = 50$ (c) $n = 5000, d = 100$ (d) $n = 5000, d = 200$

Figure 6.35 – $[f(\theta_t) - f^\star]$ for Logistic Regression with $n = 5000$ and $d = 20, 50, 100, 200$

Chapter 7

Conclusion and Perspectives

We conclude this dissertation with a short summary of the thesis and some perspectives and open questions related to the different areas and problems we have worked with.

7.1 Conclusion

Monte Carlo methods continue to be one of the most useful approaches to solve numerical integration and gradient estimation due to their simplicity and general applicability. On the one hand, the particular variance reduction technique of *control variates* offers many advantages as it relies on a simple and intuitive paradigm that is to take into account the more information we have in order to solve complex problems. On the other hand, the use of *conditioning matrices* for stochastic optimization algorithms is key to achieve optimal variance and leverage structure in data. Based on various research directions, we have developed new theoretical and practical tools.

Through the different chapters of Part II, we have developed new Monte Carlo estimators that present interesting properties: we first derived the (LS)LASSOMC estimate which allows the use of high-dimensional control variates; then we developed a weighted least-squares estimate, called AISCV, to incorporate control variates within the adaptive importance sampling framework; finally, we proposed a Monte Carlo method with control variates based on nearest neighbors estimates to achieve optimal convergence rate for Lipschitz functions.

In the second Part of this thesis, we focused on stochastic optimization algorithms through the lens of conditioning and adaptive sampling: we first derived a general asymptotic theory for *conditioned* SGD methods in a general non-convex setting, then we presented a general framework to perform coordinate sampling for SGD algorithms. Within this particular framework which leverages structure in data, we developed an algorithm, called MUSKETEER, based on a reinforcement strategy.

7.2 Perspectives and Future work

Some avenues for further research are presented for all the different chapter and associated research questions of this thesis.

On Chapter 2. The construction of control variates by a change of measure (Remark 2.1) presupposes some knowledge on the underlying integration measure in order to choose an appropriate sampling distribution. For instance, if the support of the sampling measure does not cover the whole integration domain then the method will certainly fail. Adaptive importance sampling offers a possible solution, involving online estimates of the appropriate sampling policy and the optimal linear combination of control variates. Assumption 2.10 on the sub-Gaussianity of the residuals is key to obtain

concentration inequalities. For certain applications, it might be too restrictive, however. In the absence of such an assumption or more generally of suitable bounds on the tails of the residual distribution, other types of results such as almost sure convergence rates might still be pursued. In the random design setting, the estimators of coefficient vector $\beta^\star(f)$ are all biased, even the OLS estimator. The bias may be removed by sample splitting (Avramidis and Wilson, 1993), but at the cost of an increased variance, especially if the number of control variates is large. For the LASSO-based methods, debiasing methods are studied in (Javanmard and Montanari, 2018) and the references therein. The merits of these techniques for control variate methods remain to be investigated.

On Chapter 3. The combined AISCV approach has certain design choices that the user must make, such as the sampling policy for the AIS part and the control variates for the CV part. These choices are reflected in the factor τ of Theorem 3.8 (Chapter 3), which is related to the standard deviation of the importance weight and the residual function. If too many control variates are chosen, it can lead to an ill-conditioned empirical Gram matrix or overfitting, which could cause the least-squares solution to become unstable. To prevent this, regularization techniques based on LASSO-type procedures such as the one presented in Chapter 2 can be used.

On Chapter 4. The use of nearest neighbors estimates acting as control variates with the help of a leave-one-out procedure has been shown to be efficient in order to speed up the convergence rate of Monte Carlo integration and achieve the optimal $\mathrm{O}(n^{-1/2}n^{-1/d})$ rate for Lipschitz functions. The method, called *control neighbors*, that first builds a surrogate function using 1-nearest neighbor estimates and then estimates the integral of interest by using centered variables as control variates performs very-well considering the modest computing time required. For future work, it would be interesting to continue the analysis in order to establish concentration bounds with high probability by using tools such as Mac Diarmid's inequality to treat the leave-one-out estimates.

On Chapter 5. By deriving an asymptotic theory for *conditioned* stochastic gradient descent methods in a general non-convex setting, we have revealed in Chapter 5 that the only additional assumption required to attain weak convergence is the almost sure convergence of the *conditioning* matrices. Utilizing appropriate *conditioning* matrices with the help of Hessian estimates is crucial for achieving asymptotic optimality in the sense of minimal variance. Our study focuses primarily on the weak convergence of the rescaled sequence of iterates, which is a useful tool for handling efficiency issues as algorithms can easily be compared through their asymptotic variances. It would also be beneficial to complement our asymptotic results with concentration inequalities and non-asymptotic bounds. This research direction may require additional assumptions such as strong convexity of the objective function combined with bounded gradients.

The approach described in Section 5.4 may not be computationally optimal, as it requires eigenvalue decomposition. However, *conditioned* SGD methods and especially stochastic second-order methods can be used, as they only require matrix-vector products which can be computed in $O(d^2)$ operations. Low-rank approximation with BFGS algorithm and its variant L-BFGS can help approximate the inversion of Hessian matrices in $O(d)$ operations. Furthermore, this technique has been extended to the online learning framework as well as a purely stochastic setting. The adaptive optimizers discussed in Section 5.3.1 aim to create a balance between low-memory storage of the scaling matrix representation C_k and the quality of its approximation of either the inverse Hessian $\nabla^2 f(\theta^\star)^{-1}$ or the information obtained through the geometry of the problem.

On Chapter 6. In light of Chapter 5 and the derived asymptotic theory for *conditioned*-SGD methods, a future direction of research concerns the behavior of the rescaled sequence of iterates $(\theta_t - \theta^\star)/\sqrt{\gamma_t}$ towards a Gaussian distribution. The associated asymptotic covariance matrix should reflect the information brought in by the adaptive selection matrix C_t. In particular, for the algorithm MUSKETEER, following the continuity property established for *conditioned* SGD, it is expected to have asymptotic normality with a limiting conditioning matrix C proportional to the identity matrix I_p since the coordinate sampling policy has a uniform behavior in the asymptotic regime. Furthermore, when the objective function f is s-sparse – in the sense that it only depends on $s < d$ coordinates – the associated coordinate selection matrix C_t should become degenerated with non-negative weights only for the coordinates in the support of f. Finally, in the perspective of accelerated coordinate descent methods, one may be interested in pursuing the analysis of the SCGD framework with acceleration techniques such as adding a momentum term or using particular stochastic variance reduction techniques (Johnson and Zhang, 2013).

Appendix: Additional results on Monte Carlo estimates

This appendix provides additional numerical results for the different Monte Carlo estimates of Part II. In particular, while the different Chapters 2, 3 and 4 present the numerical results in the form of mean squared errors and statistical efficiency, it is important to also take into account the computation times. For that matter, two different metrics are used in the following to evaluate the performance of the different Monte Carlo procedures: the *standard efficiency* and the *global efficiency*.

As mentioned in Chapter 2, the *standard efficiency* is defined as the ratio between the mean squared error of the naive Monte Carlo estimate and the mean squared error of the candidate procedure. Since the proposed control variate techniques rely on heavy computations through the matrix of control variates, they are most valuable when the sampling algorithm is expensive or when evaluations of the integrands are costly.

Similarly to (South et al., 2022), we consider the metric of *global efficiency* which reweights the *standard efficiency* by the computing times of the different methods. More precisely, for any method $\mathcal{M}$, the two different metrics are

$$\text{standard efficiency}(\mathcal{M}) = \frac{MSE(vanilla)}{MSE(\mathcal{M})},$$
$$\text{global efficiency}(\mathcal{M}) = \frac{MSE(vanilla)}{MSE(\mathcal{M})} \times \frac{Time(vanilla)}{Time(\mathcal{M})}.$$

The *standard efficiency* is a relevant criterion when one is only interested in obtaining the best accuracy whatever the cost in computing time. In some situations, the final precision is the only thing that matters. In other scenarios, it may be interesting to consider the computation time, because if a method is for example four times more accurate that the vanilla estimate but requires two times more computation, then reasonably we can only say that it presents a gain of two. Of course, one needs to keep in mind that it is hard to precisely evaluate the general computing time of a particular method because the run time is subject to the programming language and efficiency of the code. In the considered examples, all the code is written in Python (version 3) and the run time are computed with the method *time()*.

A.1 Capture and Sonar datasets (Chapter 2)

$m =$	90	444	1 062	3 090	5 730
OLS	9.33	**20.7**	14.7	0.14	0.06
LASSO	9.34	20.3	16.7	**14.4**	**8.57**
LSL	9.33	20.4	12.8	8.43	4.60
LSLX	9.33	19.4	**19.8**	12.9	7.86

Table 7.1 – Capture data: standard efficiency ($n = 2000$)

$m =$	90	444	1 062	3 090	5 730
OLS	7.67	18.1	22.1	15.2	0.15
LASSO	7.67	**18.4**	**22.3**	**22.8**	12.8
LSL	7.67	18.0	21.3	13.3	5.24
LSLX	7.67	17.8	21.4	21.6	**13.2**

Table 7.2 – Capture data: standard efficiency ($n = 5000$)

$m =$	61	183	305	610	1220
OLS	3.39	13.3	246	548	330
LASSO	3.39	13.6	**250**	**673**	680
LSL	3.39	13.3	246	564	499
LSLX	3.39	**13.9**	244	558	**680**

Table 7.3 – Sonar data: standard efficiency ($n = 2000$)

$m =$	61	183	305	610	1220
OLS	4.48	17.0	235	801	601
LASSO	4.49	17.0	**240**	821	721
LSL	4.48	17.0	235	804	629
LSLX	4.48	17.0	241	**833**	**734**

Table 7.4 – Sonar data: standard efficiency ($n = 5000$)

$m =$	90	444	1 062	3 090	5 730
OLS	8.23	10.3	5.21	0.01	5e-3
LASSO	7.84	10.5	5.88	2.80	0.85
LSL	7.70	10.4	4.54	1.42	0.43
LSLX	7.59	9.77	7.58	2.73	1.04

Table 7.5 – Capture data: global efficiency ($n = 2000$)

$m =$	90	444	1 062	3 090	5 730
OLS	5.21	9.56	8.31	1.28	3e-3
LASSO	5.16	9.69	8.59	4.87	1.72
LSL	5.16	9.59	7.88	2.49	0.59
LSLX	5.15	9.55	8.15	4.51	1.72

Table 7.6 – Capture data: global efficiency ($n = 5000$)

$m =$	61	183	305	610	1220
OLS	0.27	0.33	3.87	4.68	1.47
LASSO	0.27	0.35	3.96	5.55	3.00
LSL	0.26	0.33	3.85	4.90	2.19
LSLX	0.26	0.35	3.80	4.81	3.17

Table 7.7 – Sonar data: global efficiency ($n = 2000$)

$m =$	61	183	305	610	1220
OLS	0.29	0.41	3.66	6.70	2.57
LASSO	0.28	0.41	3.73	6.85	3.10
LSL	0.28	0.41	3.56	6.66	2.68
LSLX	0.28	0.41	3.70	6.95	3.17

Table 7.8 – Sonar data: global efficiency ($n = 5000$)

A.2 AISCV synthetic data and real data (Chapter 3)

Integrand	Efficiency	Sample Size n $5,000$	$10,000$	$20,000$	$30,000$	$50,000$
f_1	standard	2.97	7.87	7.56	7.81	9.64
$(d = 4)$	global	0.76	1.88	1.63	1.53	1.47
f_1	standard	2.70	14.3	20.7	30.7	41.8
$(d = 8)$	global	0.12	0.63	0.96	1.65	2.10
f_2	standard	11.0	12.6	15.5	22.7	20.7
$(d = 4)$	global	9.90	10.7	12.6	18.0	15.9
f_3	standard	9.12	37.1	51.8	78.4	102
$(d = 8)$	global	2.52	10.6	14.3	21.3	26.2

Table 7.9 – Standard and global efficiencies for AISCV compared to AIS for f_1, f_2, f_3 in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

Dataset	Efficiency	Sample Size n $5,000$	$10,000$	$20,000$	$30,000$	$50,000$
Housing	standard	7.60	6.77	19.3	17.2	53.0
	global	3.24	3.26	9.39	8.38	26.0
Abalone	standard	10.4	21.3	23.6	21.1	17.3
	global	5.63	12.2	13.5	12.0	9.85
Red	standard	8.25	9.25	8.03	7.33	6.49
Wine	global	3.84	4.66	4.01	3.66	3.24
White	standard	1.60	1.74	2.06	2.03	1.96
Wine	global	0.77	0.88	1.05	1.04	1.01

Table 7.10 – Standard and global efficiencies for AISCV1 compared to AIS for Bayesian Linear Regression on real-world datasets obtained over 100 replications.

Dataset	Efficiency	Sample Size n $5,000$	$10,000$	$20,000$	$30,000$
Housing	standard	376	155	157	228
	global	50.4	15.6	17.0	24.7
Abalone	standard	342	300	162	114
	global	10.0	12.9	9.70	5.48
Red	standard	111	77.9	83.3	95.0
Wine	global	9.58	9.39	9.40	12.4
White	standard	29.1	15.9	9.65	5.73
Wine	global	2.48	1.45	0.92	0.56

Table 7.11 – Standard and global efficiencies for NUTS sampler compared to AIS for Bayesian Linear Regression on real-world datasets obtained over 100 replications.

A.3 Control Neighbors for Barrier option (Chapter 4)

Sample Size n / Options	Efficiency	500	$1,000$	$2,000$	$3,000$	$5,000$
(Black-Scholes)	standard	65	260	297	317	130
Up-In	global	0.27	1.30	2.02	2.34	1.20
(Black-Scholes)	standard	1518	1019	725	461	189
Up-Out	global	9.0	9.6	12.7	12.3	8.0
(Heston)	standard	44.3	54.2	57.8	60.5	36.0
Up-In	global	0.21	0.42	0.60	0.75	0.60
(Heston)	standard	215	159	76.3	54.5	31.7
Up-Out	global	1.67	1.36	1.20	1.23	1.16

Table 7.12 – Standard and global efficiencies for CVNN compared to naive MC for Barrier option "Up-In" and "Up-Out" with Black-Scholes or Heston models, obtained over 100 replications.

Résumé des contributions

Motivés par les différentes questions de recherche (RQ) mentionnées dans les sections 1.2 et 1.3, nous fournissons maintenant un aperçu détaillé des contributions de cette thèse où chaque chapitre est dédié à l'une des directions de recherche.

Partie II : Monte Carlo methods and Variance Reduction

• Chapitre 2 : *Control Variate Selection for Monte Carlo Integration* (*QR*#1)

Pour faire face aux problèmes de calcul liés à l'utilisation d'un grand nombre de *variables de contrôle*, il a été proposé dans South et al. (2022) de régulariser l'estimation OLS en ajoutant un terme de pénalité ℓ_1 dans le problème de minimisation, tout comme dans le LASSO (Tibshirani, 1996). Les résultats de simulation dans South et al. (2022) montrent que cette approche, appelée LASSO, apporte de grandes améliorations en pratique. Toutefois, ces résultats pratiques ne sont pas étayés par un taux d'erreur asymptotique ni par une borne d'erreur non asymptotique. L'objectif principal de ce chapitre est de fournir une théorie non asymptotique pour l'utilisation des variables de contrôle dans les simulations de Monte Carlo.

Contributions. Les contributions sont les suivantes.

(1) Une *nouvelle méthode* appelée LSLASSO est proposée. Dans l'esprit de (Belloni and Chernozhukov, 2013), elle consiste à sélectionner les meilleures variables de contrôle via le LASSO, en utilisant le sous-échantillonnage pour diminuer le temps de calcul, puis à appliquer une régression OLS avec les variables de contrôle sélectionnées.

(2) *Recouvrement de support* : on montre que la procédure LASSO permet de sélectionner les bonnes variables de contrôle avec une grande probabilité.

(3) *Inégalités de concentration* sont dérivées pour les erreurs d'intégration des régressions OLS, LASSO et LSLASSO. Celle pour OLS met en évidence un compromis entre l'erreur d'approximation de l'intégrande dans le sous-espace vectoriel des variables de contrôle et les multicollinéarités entre les variables de contrôle. Celles pour (LS)LASSO montrent des améliorations significatives concernant les effets de la multicollinéarité.

L'approche pour les preuves combine des inégalités de concentration sous-gaussiennes bien connues (Boucheron et al., 2013a), ainsi qu'une borne inférieure pour la plus petite valeur propre d'une matrice de Gram empirique, basée sur une inégalité de Chernoff pour les matrices (Tropp, 2015, Theorem 5.1.1).

• Chapitre 3 : *A Quadrature Rule Combining Control Variates and Adaptive Importance Sampling* (*QR*#2)

L'utilisation des variables de contrôle est une technique de réduction de variance bien étudiée (Glynn and Szechtman, 2002; Owen and Zhou, 2000). Les avantages peuvent être établis théoriquement en termes de bornes d'erreur (voir Oates et al. (2017) et

Chapitre 2), de convergence faible (Portier and Segers, 2019), d'excès de risque (Belomestny et al., 2022) et même de bornes d'erreur uniformes sur de grandes classes d'intégrandes (Plassier et al., 2020). En pratique, le cadre des variables de contrôle a conduit à des procédures efficaces en apprentissage par renforcement Jie and Abbeel (2010); Liu et al. (2018) et en optimisation Wang et al. (2013), pour n'en citer que quelques-unes. L'échantillonnage par importance et les variables de contrôle dans le cas d'une densité cible gaussienne sont explorés dans Jourdain (2009). Récemment, la procédure de Kawai (2020) incorpore des variables de contrôle et prétend impliquer un échantillonnage adaptatif d'importance, mais en fait les particules sont toujours échantillonnées à partir de la distribution uniforme sur le cube unitaire. À notre connaissance, les méthodes de variables de contrôle existantes ne tiennent pas compte des changements séquentiels dans la distribution des particules, comme c'est le cas dans l'échantillonnage adaptatif d'importance. L'objectif principal de ce Chapitre est de développer un cadre permettant de combiner les variables de contrôle et l'échantillonnage par importance adaptatif.

Contributions. Les contributions peuvent être résumées comme suit:

(1) Une approche simple des moindres carrés pondérés est proposée pour améliorer la procédure des algorithmes séquentiels avec des variables de contrôle. L'estimation proposée, appelée AISCV, améliore considérablement la précision de l'algorithme initial, à la fois en théorie et en pratique.

(2) Plusieurs propriétés théoriques de l'estimation AISCV sont fournies. En particulier, nous dérivons une limite probabiliste et non asymptotique sur l'erreur d'intégration.

(3) Des considérations pratiques et des implémentations des variables de contrôle sont présentées, ainsi que des expériences numériques convaincantes.

L'approche proposée pour utiliser les variables de contrôle dans le cadre séquentiel repose sur l'expression des moindres carrés ordinaires des variables de contrôle (voir par exemple Portier and Segers (2019)). Pour prendre en compte les changements de politique, une certaine repondération doit être appliquée. L'estimation AISCV de l'intégrale $\int f\pi \,\mathrm{d}\lambda$ est définie comme la première coordonnée de la solution au problème des moindres carrés pondérés

$$(\hat{\alpha}_n, \hat{\beta}_n) = \underset{a\in\mathbb{R}, b\in\mathbb{R}^m}{\arg\min} \sum_{i=1}^{n} w_i \left(f(X_i) - a - b^\top h(X_i) \right)^2,$$

avec w_i les poids d'importance précédents. L'estimation AISCV $\hat{\alpha}_n$ possède plusieurs propriétés intéressantes :

(a) A chaque fois que g est de la forme $\alpha + \beta^\top h$ pour tous $\alpha \in \mathbb{R}$ et $\beta \in \mathbb{R}^m$, l'erreur est nulle, c'est-à-dire, $\hat{\alpha}_n = \alpha = \int f\pi \,\mathrm{d}\lambda$.

(b) L'estimation prend la forme d'une règle de quadrature $\hat{\alpha}_n = \sum_{i=1}^n v_{n,i} f(X_i)$, pour des poids de quadrature $v_{n,i}$ qui ne dépendent pas de la fonction f et qui peuvent être calculés par une simple procédure de moindres carrés pondérés.

(c) Elle peut être calculée même lorsque π n'est connu qu'à une constante multiplicative près.

Le point (a) suggère que lorsque les combinaisons linéaires des fonctions h_k couvrent une riche classe de fonctions, l'erreur d'intégration est susceptible d'être faible. Le point (b) implique que plusieurs intégrales peuvent être calculées aussi facilement qu'une seule. Le point (c) montre que l'approche est applicable aux calculs bayésiens. De plus, les variables de contrôle peuvent être mises en jeu dans un schéma *post-hoc*, après la génération des particules et des poids d'importance, et ce pour tout algorithme AIS.

Le principal résultat théorique de ce Chapitre est une majoration probabiliste non asymptotique sur $\hat{\alpha}_n - \alpha$. Dans des conditions appropriées, cette limite est égale à $\tau/\sqrt{n}$, où τ^2 est la constante d'échelle d'une condition de queue sous-gaussienne sur la variable d'erreur $\varepsilon = f - \alpha - \beta^\top h$ pour $(\alpha, \beta) = \arg\min_{a,b} \int (f - a - b^\top h)^2 \pi \, \mathrm{d}\lambda$. Notez que ε a la plus petite variance possible que l'on pourrait obtenir en utilisant les variables de contrôle h. Par conséquent, lorsque l'espace des variables de contrôle est bien adapté à l'approximation de g, l'estimation AISCV sera très précise. De plus, notre limite ne dépend que de l'espace des fonctions linéaires couvert par les variables de contrôle $h_1, \ldots, h_m$, et non de la base particulière choisie dans cet espace. Les résultats reposent sur la théorie des martingales, en particulier sur une inégalité de concentration pour les martingales sous-gaussiennes dans Jin et al. (2019). Au cours de la preuve, nous développons une nouvelle borne sur la plus petite valeur propre de certaines matrices aléatoires, en étendant une inégalité de (Tropp, 2015) au cas des martingales.

• Chapitre 4: *Speeding up Monte Carlo Integration: Nearest Neighbors Estimates as Control Variates*(*QR*#3)

Ce chapitre traite de l'utilisation de *variables de contrôle* du point de vue des taux de complexité. Comme mentionné dans la section 1.2, les méthodes de Haber (1966) et de Chopin and Gerber (2022), même si elles atteignent le taux de convergence optimal, ne sont valables que pour l'intégration sur le cube unitaire. De plus, elles impliquent un nombre géométrique (ℓ^d) d'évaluations de l'intégrande f, ce qui est problématique en pratique pour les applications à petit budget de calcul comme dans les modèles bayésiens complexes. Il est intéressant de noter que, comme mentionné dans Chopin and Gerber (2022), leur méthode de stratification est liée à une construction spécifique des variables de contrôle reposant sur une fonction de contrôle constante par morceaux qui présente un biais très faible par rapport à l'estimation par régression traditionnelle.

Cette idée précise d'utiliser une estimation avec un faible biais est le point de départ de cet article. Elle est pertinente dans le cadre considéré car la fonction f est accessible sans bruit. Il est à noter que ce type d'estimation - avec un faible biais - a également été utilisé avec succès dans le domaine connexe de l'échantillonnage par rejet adaptatif (Achddou et al., 2019) permettant d'atteindre un taux optimal. L'objectif principal de ce chapitre est de développer le cadre de *control neighbors* qui utilise les plus proches voisins comme variables de contrôle pour atteindre un taux de convergence optimal pour l'erreur d'intégration.

Contributions. Les contributions peuvent être résumées comme suit :

(1) Une nouvelle méthode de Monte Carlo appelée *Control Neighbors* est présentée. Cette méthode construit une estimation $\hat{\alpha}_n(f)$ pour approcher l'intégrale $\pi(f)$ pour une mesure de probabilité générale π et l'idée centrale découle de l'utilisation d'estimations de 1-Plus Proches Voisins comme variables de contrôle.

(2) Cette estimation permet d'atteindre le taux de convergence optimal en $\mathrm{O}(n^{-1/2}n^{-1/d})$ pour les fonctions Lipschitz. À notre connaissance, l'obtention du taux de convergence optimal pour une mesure de probabilité générale fait de cette méthode la première de son genre.

(3) Plusieurs considérations pratiques sur les voisins de contrôle sont présentées ainsi que des expériences numériques prometteuses.

Les propriétés les plus remarquables de l'estimation *control neighbors* sont :

(a) L'estimation par les voisins de contrôle peut être obtenue dans le même cadre que le Monte Carlo standard, *i.*, dès lors que l'on peut à la fois *(i)* tirer des particules aléatoires de π et *(ii)* évaluer l'intégrande f. Contrairement au cadre classique des variables de contrôle (Portier and Segers, 2019), l'estimation proposée ne nécessite pas l'existence de variables de contrôle dont les intégrales sont connues.

(b) *control neighbors* prend la forme d'une règle d'intégration linéaire $\sum_{i=1}^{n} w_{i,n} f(X_i)$ où les poids $w_{i,n}$ ne dépendent pas de l'intégrande f mais seulement des particules échantillonnées $X_1, \ldots, X_n$. Cette propriété clé permet de bénéficier d'avantages informatiques lorsque plusieurs intégrales doivent être calculées par rapport à la même mesure μ. cette propriété permet de réduire le temps de calcul des poids par rapport au temps de calcul pour évaluer les intégrandes.

(c) On montre que le taux de convergence est optimal pour les fonctions de Lipschitz, c'est-à-dire que l'erreur d'intégration diminue comme $\mathrm{O}(n^{-1/2}n^{-1/d})$ chaque fois que f est de Lipschitz (Novak, 2016). D'autres approches (pour la mesure générale π) qui ont été développées récemment, par exemple (Oates et al., 2017; Portier and Segers, 2019) n'atteignent pas ce taux.

(d) Puisque les poids $w_{n,i}$ sont construits en utilisant les estimations du plus proche voisin, des outils pratiques complets sont déjà disponibles, notamment une recherche efficace du plus proche voisin avec un arbre à k dimensions (Bentley, 1975) et une compression et une parallélisation efficaces (Pedregosa et al., 2011; Johnson et al., 2019).

(e) L'approche proposée est *post-hoc* dans le sens où elle peut être exécutée après l'échantillonnage des particules et indépendamment du mécanisme d'échantillonnage. En particulier, elle peut être mise en œuvre pour d'autres plans d'échantillonnage, notamment MCMC ou AIS.

Part III : Stochastic Approximation: Conditioning, Sampling

• Chapitre 5 : *Asymptotic Analysis of Conditioned Stochastic Gradient Descent* ($QR\#4$)

À la lumière de la question de recherche ($QR\#4$), ce chapitre concerne les problèmes d'optimisation de la forme suivante : $\min_{\theta\in\mathbb{R}^d}\{f(\theta) = \mathbb{E}_\xi[f(\theta,\xi)]\}$, où f est une fonction de perte et ξ une variable aléatoire. *Conditioned* SGD généralise *SGD standard* en ajoutant une étape de conditionnement pour affiner la direction de descente. En partant de $\theta_0 \in \mathbb{R}^d$, l'algorithme d'intérêt est défini par l'itération suivante

$$\theta_{t+1} = \theta_t - \gamma_{t+1} C_t g(\theta_t, \xi_{t+1}), \qquad t \geq 0,$$

où $g(\theta_t, \xi_{t+1})$ est un gradient sans biais évalué dans $\mathbb{R}^d$, $C_t \in \mathbb{R}^{d\times d}$ est appelée *matrice de conditionnement* et $(\gamma_t)_{t\geq 1}$ est une séquence de taux d'apprentissage décroissant.

Travaux connexes. Les travaux fondateurs autour de SGD standard ($C_k = I_d$) ont été initiés par Robbins and Monro (1951) et Kiefer et al. (1952). Depuis lors, une importante littérature connue sous le nom de *approximation stochastique*, s'est développée. La convergence presque certaine est étudiée dans Robbins and Siegmund (1971) et Bertsekas and Tsitsiklis (2000) ; les taux de convergence sont étudiés dans Kushner and Huang (1979) et Pelletier (1998a) ; des bornes non asymptotiques sont données dans Moulines and Bach (2011). La normalité asymptotique peut être obtenue par deux approches différentes : une méthode basée sur la diffusion est employée dans Pelletier (1998b) et Benaïm (1999) alors que les outils de martingale sont utilisés dans Sacks (1958) et Kushner and Clark (1978). Nous renvoyons à Nevelson and Khas'minskiĭ (1976); Delyon (1996); Benveniste et al. (2012); Duflo (2013) pour les manuels de réference sur *l'approximation stochastique*.

Les résultats susmentionnés ne s'appliquent pas directement au *conditioned* SGD en raison de la présence de la séquence de matrices $(C_k)_{k\geq 0}$ impliquant une source supplémentaire d'aléa dans l'algorithme. Les articles précurseurs traitant de la convergence faible de *conditioned*-SGD sont Venter (1967) et Fabian (1968). Dans un cadre restrictif (cas univarié $d = 1$ et hypothèses fortes sur la fonction f), leurs résultats sont encourageants car ils montrent que la variance limite de la procédure est plus petite que la variance limite de SGD standard. Les résultats de Venter et Fabian ont ensuite été étendus à des situations plus générales : (Fabian, 1973; Nevelson and Khas'minskiĭ, 1976; Wei, 1987). Dans Wei (1987), le cadre reste restrictif non seulement parce que les erreurs aléatoires sont supposées indépendantes et identiquement distribuées mais aussi parce que l'objectif f doit satisfaire leur hypothèse (4.10) qui ne s'étend guère aux objectifs autres que quadratiques.

Plus récemment, Bercu et al. (2020) ont obtenu la normalité asymptotique ainsi que l'efficacité de certaines procédures *conditioned* SGD dans le cas particulier de la *régression logistique*. L'approche précédente a été généralisée il n'y a pas longtemps dans Boyer and Godichon-Baggioni (2020) où l'utilisation de l'identité matricielle de Woodbury est promue pour calculer l'inverse de la Hessienne dans le cadre en ligne. Plusieurs résultats théoriques, dont la faible convergence de *conditioned* SGD, sont obtenus pour les fonctions objectives convexes.

Contributions. Les principaux résultats de ce Chapitre sont les suivants :

(1) Un résultat de haut niveau traitant de la convergence faible de la séquence réechelonnée des itérés $(\theta_t - \theta^\star)/\sqrt{\gamma_t}$ est fourni pour les méthodes générales de *conditioned* SGD.

(2) Un autre résultat d'intérêt indépendant traitant de la convergence presque sûre des gradients $\nabla f(\theta_t) \to 0$ est également présenté.

(3) Nous présentons des méthodes de calcul de la matrice de *conditionnement* C_t et montrons que la procédure résultante satisfait les conditions de notre théorème principal. Cela donne un algorithme réalisable qui atteint une variance minimale.

Il est intéressant de noter que notre résultat de normalité asymptotique consiste en la propriété de continuité suivante : chaque fois que la séquence de matrices $(C_t)_{t\geq 0}$ converge vers une matrice C et que les itérés $(\theta_t)_{t\geq 0}$ convergent vers un minimiseur $\theta^\star$, l'algorithme se comporte de la même manière qu'une version oracle dans laquelle C serait utilisé au lieu de C_t. Nous soulignons que contrairement à Boyer and Godichon-Baggioni (2020), aucune hypothèse de convexité n'est nécessaire sur la fonction objectif et aucun taux de convergence n'est requis sur la séquence $(C_t)_{t\geq 0}$. Ceci est important car, dans la plupart des cas, la dérivation d'un taux de convergence sur $(C_t)_{t\geq 0}$ nécessite un taux de convergence spécifique sur les itérations $(\theta_t)_{t\geq 0}$ qui, en général, est inconnu à ce stade de l'analyse.

Pour obtenir ces résultats, au lieu d'approximer la séquence des itérés redimensionnés par une diffusion continue (comme par exemple dans Pelletier (1998b)), nous nous basons sur une approche en temps discret où le schéma de récursion est directement analysé (comme par exemple dans Delyon (1996)). Plus précisément, la séquence des itérés est étudiée à l'aide d'un algorithme linéaire auxiliaire dont la distribution limite peut être déduite du théorème central limite pour les incréments de martingale (Hall and Heyde, 1980). La variance limite est dérivée d'un algorithme de système dynamique à valeurs matricielles en temps discret. Elle correspond à la solution d'une équation de Lyapunov impliquant la matrice C. Elle permet un choix spécial pour C qui garantit une variance optimale. Enfin, afin d'examiner la partie restante, une récursion particulière est identifiée. En l'étudiant sur un événement particulier, on montre que cette partie restante est négligeable.

• Chapitre 6 : *SGD with Coordinate Sampling: Theory and Practice* ($QR\#5$)

Pour rappel, l'algorithme SGD est défini par la règle de mise à jour suivante

$$\forall t \geq 0, \quad \theta_{t+1} = \theta_t - \gamma_{t+1} g_t$$

où $g_t \in \mathbb{R}^d$ est une estimation du gradient à θ_t (éventuellement biaisée) et $(\gamma_t)_{t\geq 1}$ est une séquence de pas d'apprentissage qui diminue tout au long de l'algorithme. Bien que le calcul de g_t puisse être bon marché, il nécessite toujours le calcul d'un vecteur de taille d, ce qui peut être un problème critique dans les problèmes de haute dimension. Pour résoudre cette difficulté, nous nous appuyons sur l'échantillonnage de coordonnées bien choisies de l'estimation du gradient à chaque itération.

Dance ce chapitre, nous développons le cadre de la descente de gradient par coordonnées stochastique (SCGD) qui modifie les méthodes de descente de gradient stochastique standard en ajoutant une étape de sélection pour effectuer une descente de coordonnées

aléatoire. L'algorithme SCGD est défini par l'itération suivante

$$\begin{cases} \theta_{t+1}^{(k)} = \theta_t^{(k)} & \text{if } k \neq \zeta_{t+1} \\ \theta_{t+1}^{(k)} = \theta_t^{(k)} - \gamma_{t+1} g_t^{(k)} & \text{if } k = \zeta_{t+1} \end{cases}$$

où ζ_{t+1} est une variable aléatoire évaluée dans $[\![1, d]\!]$ qui sélectionne une coordonnée de l'estimation du gradient. La distribution de ζ_t est appelée *politique d'échantillonnage des coordonnées*. Notons que le cadre SCGD est très général puisqu'il contient autant de méthodes qu'il existe de façons de générer à la fois l'estimation du gradient g_t et les variables aléatoires ζ_t.

Travaux connexes. Les auteurs de (Nutini et al., 2015) étudient la règle déterministe de Gauss-Southwell qui consiste à choisir la coordonnée dont la valeur du gradient est maximale. En faisant confiance à de grands gradients, cette règle ressemble à celle de MUSKETEER, sauf qu'aucun bruit stochastique - ni dans l'évaluation du gradient ni dans la sélection des coordonnées - n'est présent dans leur algorithme. Sous cet aspect, notre méthode diffère de toutes les études CD précédentes (Loshchilov et al., 2011; Richtárik and Takáč, 2016a; Glasmachers and Dogan, 2013; Qu and Richtárik, 2016; Allen-Zhu et al., 2016; Namkoong et al., 2017) qui reposent sur ∇f.

Parmi la littérature SGD, des méthodes de compression et de sparsification (Alistarh et al., 2017; Wangni et al., 2018) ont été développées pour l'efficacité de la communication. Les premières utilisent des opérateurs de compression pour sélectionner quelques composantes des estimations du gradient au prix du calcul complet du gradient et du tri des coordonnées. Les seconds utilisent une estimation du gradient g qui est sparsifiée en utilisant des poids de probabilité pour atteindre une estimation non biaisée du gradient. En revanche, le cadre SCGD permet au gradient d'être biaisé car aucune repondération d'importance n'est effectuée. Notons également que, pour couvrir les méthodes d'ordre zéro, l'estimation du gradient lui-même g_t peut être biaisée, comme par exemple dans l'étude récente de Ajalloeian and Stich (2020).

Contributions. L'objectif de ce Chapitre est double : d'un point de vue théorique, il s'agit de développer et d'étudier un cadre général permettant l'échantillonnage de coordonnées dans le cadre de SGD ; d'un point de vue pratique, il s'agit de fournir un algorithme efficace pour réaliser une optimisation stochastique. Les contributions sont les suivantes :

(1) *(Théorie)* Nous étudions en détail les méthodes SCGD avec échantillonnage adaptatif. Ce cadre général couvre une grande classe d'algorithmes et est bien adapté à l'optimisation d'ordre zéro. Nous montrons la convergence presque certaine des itérés SCGD $(\theta_t)_{t\in\mathbb{N}}$ vers des points stationnaires dans le sens où $\nabla f(\theta_t) \to 0$ presque sûrement ainsi que des bornes non-asymptotiques sur l'écart d'optimalité $\mathbb{E}[f(\theta_t) - f^\star]$ où $f^\star$ est une borne inférieure de f. Les conditions de travail sont relativement faibles puisque la fonction f doit seulement être L lisse (classique dans les problèmes non convexes) et les gradients stochastiques sont éventuellement biaisés avec une variance non bornée, en utilisant une condition de croissance liée à la *growth condition* de Gower et al. (2019).

(2) *(Pratique)* Nous développons un nouvel algorithme, appelé MUSKETEER, pour MUltivariate Stochastic Knowledge Extraction Through Exploration Exploitation Reinforcement. À l'image de la devise "tous pour un et un pour tous", cette

procédure appartient au cadre SCGD avec une conception particulière pour le *politique d'échantillonnage des coordonnées*. Elle compare la valeur de toutes les estimations passées du gradient g_t pour sélectionner une direction de descente (*tous pour un*) et déplace ensuite l'itération courante selon la direction choisie (*un pour tous*). L'heuristique est celle de l'apprentissage par renforcement dans le sens où les grandes coordonnées de gradient représentent une grande diminution de l'objectif et peuvent être considérées comme des récompenses élevées. Les directions résultantes doivent être favorisées par rapport au chemin associé aux coordonnées de gradient faibles. En mettant à jour le *politique d'échantillonnage des coordonnées*, l'algorithme est capable de détecter quand une direction devient gratifiante et quand une autre cesse d'être engageante.

Les preuves des résultats de convergence asymptotique sont basées sur les idées de Bertsekas and Tsitsiklis (2000) avec des extensions particulières dans le cadre des estimations de gradient biaisées. Enfin, les bornes non-asymptotiques sont inspirées de Moulines and Bach (2011) où les auteurs fournissent une analyse non-asymptotique pour l'algorithme SGD standard.

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. pages 98, 148

J. Achddou, J. Lam-Weil, A. Carpentier, and G. Blanchard. A minimax near-optimal algorithm for adaptive rejection sampling. In *Algorithmic Learning Theory*, pages 94–126. PMLR, 2019. pages 37, 117, 236

A. Agarwal, M. J. Wainwright, P. L. Bartlett, and P. K. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009. pages 34, 147

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012. page 193

A. Ajalloeian and S. U. Stich. Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020. pages 41, 186, 193, 240

D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017. pages 40, 186, 190, 240

D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, pages 5973–5983, 2018. page 190

Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119, 2016. pages 40, 185, 186, 208, 240

S.-I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998. pages 34, 146

A. Amini, A. Soleimany, S. Karaman, and D. Rus. Spatial uncertainty sampling for end-to-end control. *arXiv preprint arXiv:1805.04829*, 2018. page 32

A. Asuncion and D. Newman. UCI Machine Learning Repository, 2007. https://archive.ics.uci.edu/ml/index.php. page 66

J.-Y. Audibert and A. B. Tsybakov. Fast learning rates for plug-in classifiers. 35(2): 608–633, 2007. page 123

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a. pages 194, 195

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b. page 195

A. N. Avramidis and J. R. Wilson. A splitting scheme for control variates. *Operations Research Letters*, 14(4):187–198, 1993. pages 68, 227

R. Bardenet and A. Hardy. Monte carlo with determinantal point processes. *The Annals of Applied Probability*, 30(1):368–417, 2020. page 117

J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001. page 150

A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013. page 184

A. Belloni and V. Chernozhukov. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013. pages 35, 47, 51, 234

D. Belomestny, L. Iosipoi, E. Moulines, A. Naumov, and S. Samsonov. Variance reduction for Markov chains with application to MCMC. *Statistics and Computing*, 30: 973–997, 2020. page 46

D. Belomestny, L. Iosipoi, Q. Paris, and N. Zhivotovskiy. Empirical variance minimization with applications in variance reduction and optimal control. *Bernoulli*, 28(2): 1382–1407, 2022. pages 36, 91, 235

M. Benaïm. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII*, pages 1–68. Springer, 1999. pages 39, 147, 238

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. pages 38, 118, 121, 237

A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012. pages 34, 39, 147, 150, 238

B. Bercu, B. Delyon, and E. Rio. *Concentration inequalities for sums and martingales*. Springer, 2015. page 206

B. Bercu, A. Godichon, and B. Portier. An efficient stochastic newton algorithm for parameter estimation in logistic regressions. *SIAM Journal on Control and Optimization*, 58(1):348–367, 2020. pages 39, 147, 238

D. P. Bertsekas and J. N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000. pages 38, 41, 147, 150, 155, 174, 186, 199, 238, 241

G. Biau and L. Devroye. *Lectures on the nearest neighbor method*, volume 246. Springer, 2015. pages 123, 124, 133, 134

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009. pages 54, 79

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973. pages 22, 129

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. page 22

M. É. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 27(1):247–271, 1909. page 172

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018. pages 33, 34, 146, 155, 158, 181, 185, 189

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities*. Oxford University Press, 2013a. pages 36, 47, 52, 72, 78, 79, 84, 234

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013b. pages 97, 107

C. Boyer and A. Godichon-Baggioni. On the asymptotic rate of convergence of stochastic newton algorithms and their weighted averaged versions. *arXiv preprint arXiv:2011.09706*, 2020. pages 39, 147, 148, 154, 196, 238, 239

S. P. Brooks, E. A. Catchpole, and B. J. T. Morgan. Bayesian animal survival estimation. *Statistical Science*, 15(4):357–376, 2000. page 65

N. Brosse, A. Durmus, S. Meyn, É. Moulines, and A. Radhakrishnan. Diffusion approximations and control variates for MCMC. *arXiv preprint arXiv:1808.01665*, 2018. page 46

C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970. page 159

R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016. pages 34, 146

R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998. pages 58, 117, 118

O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004. page 91

O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008. pages 91, 99, 100, 111

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. page 18

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004. page 91

N. Chopin and M. Gerber. Higher-order stochastic integration through cubic stratification. *arXiv preprint arXiv:2210.01554*, 2022. pages 29, 37, 117, 118, 120, 236

N. A. Chriss and N. Chriss. *Black Scholes and beyond: option pricing models*. McGraw-Hill, 1997. page 22

S. Clémençon, P. Bertail, E. Chautru, and G. Papa. Optimal survey schemes for stochastic gradient descent with applications to m-estimation. *ESAIM: Probability and Statistics*, 23:310–337, 2019. page 149

P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978. page 124

D. Csiba, Z. Qu, and P. Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. In *International Conference on Machine Learning*, pages 674–683, 2015. page 185

B. Dai, N. He, H. Dai, and L. Song. Provable Bayesian inference via particle mirror descent. In *Artificial Intelligence and Statistics*, pages 985–994. PMLR, 2016. page 91

R. Davis, T. do Rego Sousa, and C. Klüppelberg. Indirect inference for time series using the empirical characteristic function and control variates. *Journal of Time Series Analysis*, 42, 01 2021. doi: 10.1111/jtsa.12582. page 47

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(3):411–436, 2006. ISSN 13697412, 14679868. pages 21, 91

B. Delyon. General results on the convergence of stochastic algorithms. *IEEE Transactions on Automatic Control*, 41(9):1245–1255, 1996. pages 39, 40, 147, 148, 153, 238, 239

B. Delyon and F. Portier. Asymptotic optimality of adaptive importance sampling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3138–3148. Curran Associates Inc., 2018. pages 146, 150, 187

B. Delyon and F. Portier. Safe adaptive importance sampling: A mixture approach. *The Annals of Statistics*, 49(2):885–917, 2021. pages 91, 96, 179

L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. pages 197, 211, 215

J. Dick and F. Pillichshammer. *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010. pages 117, 118

A. Dieuleveut, A. Durmus, and F. Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains. *Annals of Statistics*, 48(3):1348–1382, 2020. page 147

R. Douc and E. Moulines. Limit theorems for weighted samples with applications to sequential monte carlo methods. *The Annals of Statistics*, pages 2344–2376, 2008. page 91

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml. page 181

D. Dua and C. Graff. Uci Machine Learning Repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of Information and Computer Science*, 25:27, 2019. pages 101, 113

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011. pages 33, 34, 146

J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012. page 185

M. Duflo. *Random iterative models*, volume 34. Springer Science & Business Media, 1st edition, 2013. pages 39, 147, 150, 238

V. Fabian. On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, 39(4):1327–1332, 1968. pages 39, 147, 238

V. Fabian. Asymptotically efficient stochastic approximation; the rm case. *The Annals of Statistics*, 1(3):486–495, 1973. pages 39, 147, 238

O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. page 184

O. Fercoq, Z. Qu, P. Richtárik, and M. Takáč. Fast distributed coordinate descent for non-strongly convex losses. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2014. page 184

A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 385–394, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857. page 185

R. Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13 (3):317–322, 1970. page 159

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010. pages 34, 184

S. Gadat and F. Panloup. Optimal non-asymptotic bound of the ruppert-polyak averaging without strong convexity. *arXiv preprint arXiv:1709.03342*, 2017. page 147

S. Gadat, T. Klein, and C. Marteau. Classification in general finite dimensional spaces with the k-nearest neighbor rule. *The Annals of Statistics*, 44(3):982–1009, 2016. page 124

S. Gadat, F. Panloup, S. Saadane, et al. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018. pages 19, 156, 193, 207

W. Gautschi. *Orthogonal polynomials: computation and approximation*. OUP Oxford, 2004. page 98

N. Gazagnadou, R. Gower, and J. Salmon. Optimal mini-batch and step sizes for saga. In *International conference on machine learning*, pages 2142–2150. PMLR, 2019. pages 155, 191

A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016. page 187

J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989. page 91

S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. page 185

T. Glasmachers and U. Dogan. Accelerated coordinate descent with adaptive coordinate frequencies. In *Asian Conference on Machine Learning*, pages 72–86, 2013. pages 40, 185, 186, 240

P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, New York, NY, USA, 2004. pages 18, 22, 46, 49, 117, 119

P. W. Glynn and R. Szechtman. Some new perspectives on the method of control variates. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 27–49. Springer, 2002. pages 26, 36, 47, 50, 91, 234

E. Gobet and C. Labart. Solving bsde with adaptive control variate. *SIAM Journal on Numerical Analysis*, 48(1):257–277, 2010. doi: 10.1137/090755060. page 46

D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970. page 159

J. Gorham and L. Mackey. Measuring sample quality with Stein's method. *Advances in Neural Information Processing Systems*, 28, 2015. page 98

R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural networks*, 1(1):75–89, 1988. pages 63, 66

R. Gower, N. Le Roux, and F. Bach. Tracking the gradients using the hessian: a new look at variance reducing stochastic methods. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 707–715, Canary Islands, Spain, 2018. PMLR. page 46

R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209. PMLR, 2019. pages 32, 41, 149, 155, 173, 186, 190, 191, 240

R. M. Gower, P. Richtárik, and F. Bach. Stochastic quasi-gradient methods: Variance reduction via jacobian sketching. *Mathematical Programming*, 188(1):135–192, 2021. pages 155, 191

S. Haber. A modified monte-carlo quadrature. *Mathematics of Computation*, 20(95): 361–368, 1966. pages 29, 37, 117, 118, 120, 236

S. Haber. A modified monte-carlo quadrature. ii. *Mathematics of Computation*, 21(99): 388–397, 1967. pages 29, 117

S. Haber. Stochastic quadrature formulas. *Mathematics of Computation*, 23(108):751–764, 1969. pages 29, 117

P. Hall and C. Heyde. *Martingale Limit Theory and Its Application*. Probability and mathematical statistics. Academic Press, 1980. ISBN 9781483240244. URL https://books.google.fr/books?id=wdLajgEACAAJ. pages 40, 148, 162, 176, 239

J. Hanna, S. Niekum, and P. Stone. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pages 2605–2613. PMLR, 2019. page 187

D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978. page 181

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009. page 30

T. Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995. page 96

S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993. page 129

D. Higdon, J. D. McDonnell, N. Schunck, J. Sarich, and S. M. Wild. A bayesian approach for parameter estimation and prediction using a computationally intensive model. *Journal of Physics G: Nuclear and Particle Physics*, 42(3):034009, 2015. page 116

J. Hirschberg and C. D. Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015. page 18

K. E. Hoff III, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286, 1999. page 122

M. D. Hoffman, A. Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014. page 114

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012. page 84

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pages 9–1. JMLR Workshop and Conference Proceedings, 2012. pages 53, 76

A. Javanmard and A. Montanari. Debiasing the lasso: Optimal sample size for Gaussian designs. *The Annals of Statistics*, 46(6A):2593–2622, 2018. pages 68, 227

T. Jie and P. Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. pages 46, 91, 235

C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019. pages 37, 72, 73, 92, 97, 103, 236

J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. pages 38, 118, 122, 237

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013. page 228

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. page 21

B. Jourdain. Adaptive variance reduction techniques in finance. *Radon Series Comp. Appl. Math*, 8:1–18, 2009. pages 36, 91, 235

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. pages 18, 20

S. M. Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002. pages 34, 146

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016. page 156

R. Kawai. Adaptive importance sampling and control variates. *Journal of Mathematical Analysis and Applications*, 483(1):123608, 2020. pages 36, 91, 235

H. K. Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. URL https://cds.cern.ch/record/1173048. page 180

J. Kiefer, J. Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952. pages 38, 147, 185, 238

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. pages 34, 146

T. Kloek and H. K. Van Dijk. Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. *Econometrica: Journal of the Econometric Society*, pages 1–19, 1978. page 91

J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. page 20

M. Kohler and A. Krzyżak. Optimal global rates of convergence for interpolation problems with random design. *Statistics & Probability Letters*, 83(8):1871–1879, 2013. page 120

M. Kohler, A. Krzyżak, and H. Walk. Rates of convergence for partitioning and nearest neighbor regression estimates with unbounded data. *Journal of Multivariate Analysis*, 97(2):311–323, 2006. page 124

A. Korba and F. Portier. Adaptive importance sampling meets mirror descent: a bias-variance tradeoff. In *International Conference on Artificial Intelligence and Statistics*, pages 11503–11527. PMLR, 2022. page 91

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. pages 211, 215

S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. page 22

H. J. Kushner and D. S. Clark. Stochastic approximation methods for constrained and unconstrained systems. 1978. pages 39, 147, 238

H. J. Kushner and H. Huang. Rates of convergence for stochastic approximation type algorithms. *SIAM Journal on Control and Optimization*, 17(5):607–617, 1979. pages 39, 147, 238

J.-D. Lebreton, K. P. Burnham, J. Clobert, and D. R. Anderson. Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies. *Ecological monographs*, 62(1):67–118, 1992. page 65

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012. page 148

Y. T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 147–156. IEEE, 2013. page 184

R. Leluc and F. Portier. Towards asymptotic optimality with conditioned stochastic gradient descent. *arXiv preprint arXiv:2006.02745*, 2020. pages 10, 196

R. Leluc and F. Portier. Sgd with coordinate sampling: Theory and practice. *Journal of Machine Learning Research*, 23(342):1–47, 2022. URL http://jmlr.org/papers/v23/21-1240.html. page 10

R. Leluc, F. Portier, and J. Segers. Control variate selection for Monte Carlo integration. *Statistics and Computing*, 31, 07 2021. doi: 10.1007/s11222-021-10011-z. pages 9, 97, 102, 103, 117, 120

R. Leluc, F. Portier, J. Segers, and A. Zhuman. A Quadrature Rule combining Control Variates and Adaptive Importance Sampling. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11842–11853. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4d4e8614a37f0aff841ba87ed1a898c1-Paper-Conference.pdf. page 9

X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. page 185

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. pages 34, 146, 159

H. Liu, Y. Feng, Y. Mao, D. Zhou, J. Peng, and Q. Liu. Action-dependent control variates for policy optimization via stein identity. In *ICLR 2018 Conference*, February 2018. pages 46, 91, 235

S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37 (5):43–54, 2020. page 185

I. Loshchilov, M. Schoenauer, and M. Sebag. Adaptive coordinate descent. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 885–892, 2011. pages 40, 185, 186, 240

Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015. page 184

L. Martino, V. Elvira, D. Luengo, and J. Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27(3):599–623, 2017. page 91

G. Marzolin. Polygynie du Cincle plongeur (Cinclus cinclus) dans les côtes de Lorraine. *Oiseau et la Revue Francaise d'Ornithologie*, 58(4):277–286, 1988. pages 63, 64, 65

H. B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010. page 33

R. C. Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973. page 129

N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949. page 23

T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369, 2001. page 21

A. Mira, R. Solgi, and D. Imparato. Zero variance Markov Chain Monte Carlo for Bayesian estimators. *Statistics and Computing*, 23(5):653–662, 2013. pages 91, 98

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. page 20

S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020. page 19

P. Moritz, R. Nishihara, and M. Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics*, pages 249–258. PMLR, 2016. page 159

E. Moulines and F. R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011. pages 39, 41, 147, 156, 186, 193, 205, 238, 241

H. Namkoong, A. Sinha, S. Yadlowsky, and J. C. Duchi. Adaptive sampling probabilities for non-smooth optimization. In *International Conference on Machine Learning*, pages 2574–2583, 2017. pages 40, 185, 186, 196, 197, 207, 210, 214, 240

R. M. Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993. page 21

I. Necoara, Y. Nesterov, and F. Glineur. A random coordinate descent method on large-scale optimization problems with linear constraints. Technical report, Technical Report, 2014. page 184

D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*, volume 27, pages 1017–1025, 2014. page 185

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009. pages 155, 191

A. S. Nemirovski and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983. pages 155, 191

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. pages 34, 184

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013. pages 172, 174

Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017. pages 185, 188, 196

M. B. Nevelson and R. Z. Khas'minskiĭ. *Stochastic approximation and recursive estimation*, volume 47. American Mathematical Soc., 1976. pages 39, 147, 238

W. K. Newey. Convergence rates and asymptotic normality for series estimators. *Journal of econometrics*, 79(1):147–168, 1997. page 53

L. Nguyen, P. H. Nguyen, M. Dijk, P. Richtárik, K. Scheinberg, and M. Takác. Sgd and hogwild! convergence without the bounded gradients assumption. In *International Conference on Machine Learning*, pages 3750–3758. PMLR, 2018. pages 155, 191

D. J. Nott, C. C. Drovandi, K. Mengersen, M. Evans, et al. Approximation of Bayesian predictive p-values with regression ABC. *Bayesian Analysis*, 13(1):59–83, 2018. page 65

E. Novak. Some results on the complexity of numerical integration. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 161–183. Springer, 2016. pages 20, 28, 38, 116, 117, 118, 127, 237

J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641, 2015. pages 35, 40, 185, 186, 192, 240

C. J. Oates, M. Girolami, and N. Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017. pages 29, 36, 38, 46, 49, 91, 117, 118, 119, 120, 127, 128, 131, 132, 133, 234, 237

M.-S. Oh and J. O. Berger. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation*, 41(3-4):143–168, 1992. pages 21, 91

R. Okuda, Y. Kajiwara, and K. Terashima. A survey of technical trend of adas and autonomous driving. In *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, pages 1–4. IEEE, 2014. page 20

S. M. Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989. page 122

A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000. pages 36, 49, 67, 91, 96, 234

A. B. Owen. *Monte Carlo Theory, Methods and Examples*. 2013. http://statweb.stanford.edu/~owen/mc/. pages 47, 119

G. Papa, P. Bianchi, and S. Clémençon. Adaptive sampling for incremental optimization using stochastic gradient descent. In *International Conference on Algorithmic Learning Theory*, pages 317–331. Springer, 2015. pages 149, 185

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. page 98

K. K. Patel and A. Dieuleveut. Communication trade-offs for local-sgd with large step size. *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32(CONF), 2019. page 190

A. Patil, D. Huard, and C. J. Fonnesbeck. Pymc: Bayesian stochastic modelling in python. *Journal of statistical software*, 35(4):1, 2010. page 115

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. pages 38, 59, 118, 122, 148, 237

M. Pelletier. On the almost sure asymptotic behaviour of stochastic algorithms. *Stochastic processes and their applications*, 78(2):217–244, 1998a. pages 39, 147, 238

M. Pelletier. Weak convergence rates for stochastic approximation with application to multiple targets and simulated annealing. *Annals of Applied Probability*, pages 10–44, 1998b. pages 39, 40, 147, 148, 150, 151, 154, 238, 239

D. Perekrestenko, V. Cevher, and M. Jaggi. Faster coordinate descent via adaptive importance sampling. In *Artificial Intelligence and Statistics*, pages 869–877. PMLR, 2017. page 185

V. Plassier, F. Portier, and J. Segers. Risk bounds when learning infinitely many response functions by ordinary linear regression. *arXiv preprint arXiv:2006.09223. To appear in Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, 2020. pages 36, 91, 235

B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963. page 193

B. T. Polyak. A new method of stochastic approximation type. *Avtomatika i telemekhanika*, (7):98–107, 1990. page 147

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. page 147

F. Portier and B. Delyon. Asymptotic optimality of adaptive importance sampling. *Advances in Neural Information Processing Systems*, 31:3134–3144, 2018. pages 67, 91, 99, 111, 112

F. Portier and J. Segers. Monte Carlo integration with a growing number of control variates. *Journal of Applied Probability*, 56(4):1168–1186, 2019. doi: 10.1017/jpr.2019.78. pages 26, 29, 36, 38, 47, 50, 53, 54, 56, 91, 97, 117, 118, 119, 120, 235, 237

Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016. pages 40, 185, 186, 240

Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in neural information processing systems*, pages 865–873, 2015. page 184

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33, pages 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR. page 46

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. page 131

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. pages 34, 146

F. M. Richards. The interpretation of protein structures: total volume, group volume distributions and packing density. *Journal of molecular biology*, 82(1):1–14, 1974. page 122

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014. page 184

P. Richtárik and M. Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016a. pages 40, 185, 186, 240

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016b. page 184

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. pages 32, 38, 147, 149, 185, 238

H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pages 233–257. Elsevier, 1971. pages 38, 147, 176, 209, 238

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 2 of *Springer Texts in Statistics*. Springer, second edition, 1999. page 19

W. Rudin. *Real and Complex Analysis*. Tata McGraw-Hill Education, 2006. pages 26, 47

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. page 33

C. Rycroft. Voro++: A three-dimensional voronoi cell library in c++. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009. page 122

J. Sacks. Asymptotic distribution of stochastic approximation procedures. *The Annals of Mathematical Statistics*, 29(2):373–405, 1958. pages 39, 147, 238

A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017. page 20

M. Schmidt and N. L. Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013. page 155

N. N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *Artificial intelligence and statistics*, pages 436–443. PMLR, 2007. page 159

P. L. Seidel. *Ueber ein verfahren, die gleichungen, auf welche die methode der kleinsten quadrate führt, sowie lineäre gleichungen überhaupt, durch successive annäherung aufzulösen*, volume 11. Verlag d. Akad., 1873. page 34

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013. page 184

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011. pages 155, 191

O. Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017. page 185

D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970. page 159

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014. page 146

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. page 20

L. South, C. Oates, A. Mira, and C. Drovandi. Regularized zero-variance control variates. *Bayesian Analysis*, 1(1):1–24, 2022. pages 35, 47, 62, 63, 65, 117, 120, 230, 234

C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, volume 6, pages 583–603. University of California Press, 1972. page 98

C. J. Stone. Optimal global rates of convergence for nonparametric regression. *The annals of statistics*, pages 1040–1053, 1982. page 120

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974. page 124

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018. pages 20, 146, 150

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. pages 35, 47, 50, 234

R. Tibshirani, M. Wainwright, and T. Hastie. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 2015. pages 51, 54, 59, 80, 83

T. Tieleman, G. Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012. pages 34, 146

J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. arXiv:1501.01571. pages 36, 37, 47, 69, 92, 102, 103, 234, 236

S. A. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009. page 54

J. H. Venter. An extension of the robbins-monro procedure. *The Annals of Mathematical Statistics*, 38(1):181–190, 1967. pages 39, 147, 238

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. page 20

C. Wang, X. Chen, A. Smola, and E. Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. pages 46, 91, 235

Y. Wang, S. Du, S. Balakrishnan, and A. Singh. Stochastic zeroth-order optimization in high dimensions. In A. Storkey and F. Perez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1356–1365. PMLR, PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/wang18e.html. page 192

J. Wangni, J. Wang, J. Liu, and T. Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages 1299–1309, 2018. pages 40, 186, 190, 191, 214, 240

C. Wei. Multivariate adaptive stochastic approximation. *The Annals of Statistics*, 15 (3):1115–1130, 1987. pages 39, 147, 238

C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006. page 132

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. pages 20, 150

T. T. Wu, K. Lange, et al. Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, 2(1):224–244, 2008. pages 34, 184

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. pages 197, 211, 215

M. Zedek. Continuity and location of zeros of linear combinations of polynomials. *Proceedings of the American Mathematical Society*, 16(1):78–84, 1965. page 168

A. Zhang, L. D. Brown, and T. T. Cai. Semi-supervised inference: General theory and estimation of means. *The Annals of Statistics*, 47(5):2538–2566, 2019. page 47

P. Zhang. Nonparametric importance sampling. *J. Amer. Statist. Assoc.*, 91(435): 1245–1253, 1996. page 91

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116, 2004. page 148

W. Zhu, X. Chen, and W. B. Wu. Online covariance matrix estimation in stochastic gradient descent. *Journal of the American Statistical Association*, pages 1–12, 2021. page 147

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Méthodes de Monte Carlo et Approximation Stochastique: Théorie et Applications au Machine Learning

Mots clés : Monte Carlo, Approximation Stochastique, Réduction de Variance, Echantillonage Adaptatif

Résumé : Dans de nombreux domaines de recherche, que ce soit l'inférence variationnelle, l'inférence Bayésienne ou l'apprentissage par renforcement, le besoin d'un calcul précis et efficace d'intégrales et de paramètres minimisant des fonctions de risque apparaît, faisant des méthodes d'optimisation stochastiques et de Monte Carlo l'un des problèmes fondamentaux de la recherche en statistique et en apprentissage automatique. Cette thèse se concentre sur des méthodes d'intégration par Monte Carlo et d'optimisation stochastique, tant d'un point de vue théorique que pratique, où l'idée centrale est d'utiliser l'aléatoire pour résoudre des problèmes numériques déterministes. D'un point de vue technique, l'étude se concentre sur la réduction de la variance et des techniques d'échantillonnage adaptatif. La première partie de la thèse se concentre sur diverses techniques de variables de contrôle pour l'intégration de Monte Carlo. L'étude est basée sur des outils mathématiques issus de la théorie des probabilités et des statistiques visant à comprendre le comportement de certains algorithmes existants et à en concevoir de nouveaux avec une analyse approfondie de l'erreur d'intégration. Nous présentons une procédure LASSO pour utiliser les variables de contrôle en grande dimension. Une estimation pondérée des moindres carrés est ensuite proposée pour incorporer les variables de contrôle dans le cadre de l'échantillonnage adaptatif par importance. Enfin, une méthode de Monte Carlo basée sur des estimateurs des plus proches voisins est proposée. La deuxième partie traite d' algorithmes d'optimisation stochastique. Nous étudions d'abord une classe d'algorithmes de descente de gradient stochastique (SGD) basée sur un préconditionnement de la direction du gradient. Nous présentons ensuite un cadre général pour effectuer un échantillonnage adaptatif des coordonnées. Alors que les formes classiques d'algorithmes SGD traitent les différentes coordonnées de la même manière, un cadre permettant l'échantillonnage adaptatif (non uniforme) des coordonnées est développé pour exploiter la structure des données. Tous les algorithmes sont implémentés et testés par rapport aux méthodes de l'état de l'art et des expériences numériques approfondies sont fournies pour permettre la reproductibilité. Tous les algorithmes développés dans cette thèse sont libres de droits et disponibles en ligne.

Title : Monte Carlo Methods and Stochastic Approximation: Theory and Applications to Machine Learning

Keywords : Monte Carlo, Stochastic Approximation, Variance Reduction, Adaptive Sampling

Abstract : Across a breadth of research areas, whether in Bayesian inference, reinforcement learning or variational inference, the need for accurate and efficient computation of integrals and parameters minimizing risk functions arises, making stochastic optimization and Monte Carlo methods one of the fundamental problems of statistical and machine learning research. This thesis focuses on Monte Carlo integration and stochastic optimization methods, both from a theoretical and practical perspectives, where the core idea is to use randomness to solve deterministic numerical problems. From a technical standpoint, the study is mainly based on two standard concepts: variance reduction and adaptive sampling techniques. The first part of the thesis focuses on various control variates techniques for Monte Carlo integration. The study is based on mathematical tools coming from probability theory and statistics aiming to understand the behavior of certain existing algorithms and to design new ones with thorough analysis of the integration error. First, we present a LASSO-type procedure to allow the use of high-dimensional control variates. Then, a weighted least-squares estimate, called AISCV, is proposed to incorporate control variates within the adaptive importance sampling framework. Finally, a Monte Carlo method with control variates based on nearest neighbors estimates, called Control Neighbors, is provided. The second part of the thesis deals with stochastic optimization algorithms. First, we investigate a general class of stochastic gradient descent (SGD) algorithms, called conditioned SGD, based on a preconditioning of the gradient direction. Then we present a general framework to perform coordinate sampling for SGD algorithms. While classical forms of SGD algorithms treat the different coordinates in the same way, a framework allowing for adaptive (non uniform) coordinate sampling is developed to leverage structure in data. To emphazise the practical applications of the proposed methods, all algorithms are implemented and tested against state-of-the-art procedures and extensive numerical experiments are provided to allow reproducibility. All algorithms developed in this thesis are open-sourced and available online.

Institut Polytechnique de Paris
91120 Palaiseau, France