Université Paris Cité

École doctorale Sciences Mathématiques de Paris Centre
(ED 386)

Laboratoire de Probabilités, Statistiques et Modélisation
(LPSM, UMR 8001)

Thèse de Doctorat
Discipline : Mathématiques Appliquées

Algorithmes Robustes et Autres Contributions à l'Apprentissage Statistique

Robust Algorithms and Other Contributions to Machine Learning

Présentée et soutenue publiquement par

Ibrahim Merad

le 12 décembre 2023

Dirigée par : **Stéphane Gaïffas**
Et par : **Emmanuel Bacry**

Composition du jury :

Anatoli Juditsky	PR, Université Grenoble Alpes	Rapporteur
Gábor Lugosi	PR, Universitat Pompeu Fabra	Rapporteur
Stéphane Gaïffas	PR, Université Paris Cité	Directeur de thèse
Emmanuel Bacry	DR, Université Paris Dauphine PSL	Co-directeur de thèse
Matthieu Lerasle	PR, ENSAE	Président
Po-Ling Loh	PR, University of Cambridge	Examinatrice

Robust Algorithms and Other Contributions to Machine Learning

Abstract

This thesis deals with theoretical and methodological aspects of machine learning. This discipline has found numerous applications thanks to the availability of vast amounts of data. However, empirical evidence suggests that heavy-tailed distributions and corruption can often emerge in training datasets and may compromise the performances of machine learning models. This has motivated the development of robust statistics which seek more dependable methods when data assumptions are weakened.

In this thesis, we propose computationally efficient robust learning algorithms and back them up with theoretical analyses establishing their optimization convergence and the statistical properties of their estimates.

In our first contribution, we propose to use coordinate gradient descent (CGD) with robust scalar estimators of the partial derivatives in order to perform robust learning. This allows to avoid the computational cost of robust vector mean estimation by using only scalar estimates. The resulting procedure is robust to heavy-tails and corruption as attested by the generalization error bounds we show for smooth convex objectives. Moreover, computational overhead is minimal since the complexity is the same as non robust methods. We efficiently implement this method in a Python library called `linlearn` and confirm the advantages of robust CGD through extensive numerical experiments.

Our next contribution deals with robust learning in the high-dimensional setting where optimization is carried out using non-Euclidean methods. We develop a robust high-dimensional learning framework suitable for smooth and non-smooth objectives which uses robust gradient estimation methods tailored to problem-specific non-Euclidean metrics. For the particular case of Vanilla sparse estimation, we obtain an efficient solution algorithm with strong robustness properties. Besides the theoretical analysis establishing these properties, we implement this algorithm in the `linlearn` library and demonstrate its performance through experiments on real data.

The third contribution brings a solution for the streaming data setting where samples are only seen once in a sequential fashion. We propose a clipped SGD algorithm for stochastic optimization using gradient norm quantiles as thresholds. Using Markov chain tools, we prove that the iteration is robust to heavy tails and corrupted data and converges to a limit distribution concentrated around an optimum. In another chapter, we leverage similar tools to study the convergence and concentration properties of standard SGD. In particular, we obtain a non asymptotic concentration bound for Polyak-Ruppert averaging of a tail SGD sequence.

Our contributions also include a new random forest algorithm called WildWood. The latter adds an aggregation mechanism within each tree of a forest which uses out-of-bag samples to compute average predictions over all subtrees. This computation is precise and efficient thanks to the context tree weighting algorithm. As we show theoretically, this allows to nearly match the performance of the best subtree. We propose an efficient implementation in the Python library `wildwood` and experimentally demonstrate the algorithm's competitiveness with popular ensemble methods such as classical random forests and boosting algorithms.

Finally, we present an efficient non Bayesian algorithm for online logistic regression which may achieve optimal regret and provide a preliminary analysis for it.

Keywords: machine learning, robust methods, heavy-tailed data, outliers, generalization error, sparse recovery, stochastic optimization, random forests, online logistique regression.

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université – Campus Pierre et Marie Curie – 4 place Jussieu – 75005 Paris – France

Algorithmes Robustes et Autres Contributions à l'Apprentissage Statistique

Résumé

Cette thèse traite d'aspects théoriques et méthodologiques de l'apprentissage automatique. Cette discipline a trouvé de nombreuses applications grâce aux grandes quantités de données disponibles. Cependant, des constats empiriques suggèrent souvent des distributions à queue lourde et de la corruption dans les jeux de données ce qui pourrait compromettre les performances des modèles. Ceci a motivé le développement de la statistique robuste qui cherche des méthodes plus fiables sous des hypothèses affaiblies sur les données.

Nous présentons des algorithmes d'apprentissage efficaces et robustes avec une analyse théorique établissant la convergence de leur optimisation et les propriétés statistiques de leurs estimateurs.

La première contribution propose d'utiliser la descente de gradient par coordonnées (CGD) avec estimation robuste des dérivées partielles pour effectuer de l'apprentissage robuste. Cela permet d'éviter les calculs coûteux liés à l'estimation de moyenne vectorielle robuste grâce à des estimateurs scalaires. Le procédé obtenu est robuste aux queues lourdes et à la corruption comme attesté par les bornes d'erreur de généralisation établies pour des fonctions convexes et gradient-Lipschitz. De plus, le surplus de calcul est minime vu que la complexité est la même que les méthodes non robustes. Nous proposons une implémentation efficace dans la librairie Python `linlearn` et confirmons les avantages de CGD robuste à travers des expériences numériques.

La seconde contribution traite le cas en haute dimension où l'optimisation se fait par méthode non-Euclidienne. Nous développons un cadre d'apprentissage en haute dimension adapté à plusieurs fonctions objectif qui utilise des méthodes d'estimation de gradient robustes adaptées aux métriques non-Euclidiennes spécifiques à chaque problème. Dans le cas de l'estimation éparse standard, on obtient un algorithme efficace et fortement robuste. En plus de l'analyse théorique établissant ces propriétés, nous implémentons cet algorithme dans la librairie `linlearn` et confirmons ses performances à travers des expériences sur données réelles.

La contribution suivante apporte une solution pour les flux de données où les échantillons ne sont accessibles qu'individuellement et séquentiellement. Nous proposons un algorithme SGD tronqué pour l'optimisation stochastique utilisant comme seuils des quantiles de norme de gradient. Grâce à des outils de chaînes de Markov, nous prouvons que l'itération est robuste aux queues lourdes et aux données corrompues et qu'elle converge vers une distribution limite concentrée autour d'un optimum. Dans un autre chapitre, nous utilisons des outils similaires pour étudier les propriétés de convergence et concentration de l'itération SGD classique. En particulier, nous obtenons une borne de concentration non asymptotique pour les moyennes de Polyak-Ruppert.

Nos contributions comprennent également un nouvel algorithme de forêt aléatoire appelé WildWood. Ce dernier ajoute un mécanisme d'agrégation par arbre utilisant les échantillons bootstrap pour calculer une prédiction moyenne sur les sous-arbres. Ce calcul est précis et efficace grâce à l'algorithme de *context tree weighting*. Un résultat théorique montre que cette agrégation permet d'approcher la performance du meilleur sous-arbre. Nous proposons une implémentation efficace dans la librairie Python `wildwood` et montrons expérimentalement sa compétitivité avec des méthodes d'ensemble connues comme les forêts aléatoires standards et les algorithmes de boosting.

Enfin, nous présentons un algorithme non Bayésien efficace pour la régression logistique en ligne qui peut atteindre le regret optimal et fournissons une analyse préliminaire pour ce dernier.

Mots clés : apprentissage automatique, méthodes robustes, queues lourdes, erreur de généralisation, estimation éparse, optimisation stochastique, forêts aléatoires, régression logistique en ligne.

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université – Campus Pierre et Marie Curie – 4 place Jussieu – 75005 Paris – France

Remerciements

Depuis le début de mon parcours, j'ai bénéficié de l'aide et du soutien considérable de nombreuses personnes. Leur apport fût d'une telle importance qu'il aurait autrement été impensable pour moi d'en arriver là où j'en suis aujourd'hui. Sans avoir la prétention d'être exhaustif, je tiens à exprimer ma reconnaissance envers ces personnes dans les quelques paragraphes qui suivent.

J'aimerais tout d'abord remercier Stéphane pour m'avoir parrainé et m'avoir offert la possibilité d'effectuer une thèse de doctorat sous sa direction tout en me laissant une bonne part de liberté dans mon activité de recherche. Merci d'avoir cru en moi et pour les maintes relectures de mes travaux qui m'ont lentement mais sûrement appris à faire un travail de recherche intéressant et bien présenté. Je garderai un agréable souvenir de nos échanges et discussions pendant ces trois années. Je remercie également Emmanuel pour avoir co-encadré ma thèse. Bien que mes choix de sujets de travail ont fait que nos échanges soient relativement limités, Emmanuel fût toujours prêt à me prêter main forte lors des périodes d'incertitude que j'ai traversées.

Je remercie tous les membres du jury de me faire l'honneur de prendre part à ma soutenance et en particulier Anatoli Juditsky et Gábor Lugosi pour avoir accepté de rapporter mon manuscrit.

Je souhaite également exprimer ma gratitude envers les tuteurs qui m'ont accompagné pendant ma scolarité à l'ENS à savoir Jean Feydy, Gabriel Peyré et Stéphane Gaïffas qui devint mon directeur de thèse par la suite. J'ai grandement bénéficié du système de tutorat qui m'a aidé à faire des choix intelligents dans mon parcours pédagogique tout en prenant conscience des horizons qui se présentaient à moi pour le futur de ma carrière. Je remercie particulièrement Jean Feydy pour m'avoir accordé de nombreuses discussions à travers lesquelles j'ai pu acquérir une vision plus mature du monde scientifique.

J'aimerais également exprimer ma reconnaissance envers Bastien Fernandez et Stéphane Boucheron pour l'intérêt et la bienveillance qu'ils ont manifestés lors du suivi de ma thèse dans le souci d'en assurer le bon déroulement.

En plus de mes encadrants, j'ai aussi eu la chance de collaborer avec un de cercle collègues. Je pense en particulier à Yiyang que je remercie pour nos nombreux échanges et pour m'avoir introduit dans l'équipe alors que je faisais mes premiers pas dans le laboratoire. Je remercie également Jaouad pour plusieurs réunions qui m'ont aidé à développer mon intuition par rapport à certains sujets de recherche auxquels je me suis intéressé.

Je remercie l'équipe administrative pour avoir fait un excellent travail pour faciliter mes démarches et m'avoir toujours bien informé en répondant adroitement et ponctuellement à toutes mes requêtes.

Je remercie aussi les nombreux autres collègues que j'ai eu le plaisir de côtoyer au laboratoire pour les agréables moments de convivialité que nous avons passés ensemble et les discussions intéressantes : Aaraona, Abdelhak, Alessandro, Alexis, Ali, Anna, Arthur, Azar, Dounia, Guillaume, Hiroshi, Hoang, Ilyes, Junchao, Léo, Lionel, Lucas, Massil, Mohan, Nathan, Nesrine, Ons, Orphée, Pablo, Pierre, Raed, Sothea, Sylvain, Xuanye, Yiyang.

Enfin, je remercie bien sûr ma famille pour leur soutien permanent, inconditionnel et inébranlable tout au long de mon parcours. Merci à mes parents qui m'ont encouragé, à mon grand frère qui a toujours su me donner un bon exemple et aussi à mes soeurs qui m'ont constamment soutenu et que j'ai la chance d'avoir à côté.

Contents

Chapitre 1

Introduction en français

Sommaire

Cette thèse vise à apporter des contributions théoriques et méthodologiques au domaine de l'apprentissage automatique. Ce dernier englobe divers algorithmes permettant à un ordinateur à apprendre à exécuter une tâche sur des données en entrée à travers l'exposition répétée d'exemples.

Étant apparu au milieu du vingtième siècle, le développement des méthodes d'apprentissage automatique a progressé à un rythme sans précédent pendant les deux dernières décennies et reste un sujet très actif de nos jours [173, 207, 80, 329]. Cette croissance rapide a été favorisée par de multiples facteurs. Les plus importants parmi ces derniers sont : d'abord, l'augmentation de la puissance de calcul mise à disposition par le matériel informatique permettant d'entraîner des modèles de dimension macroscopique [348, 407, 276].

Deuxièmement, l'explosion des volumes de données régulièrement générées et accessibles sur internet et sur les réseaux sociaux en particulier fournissant des quantités abondantes d'exemples dont les machines peuvent apprendre. Et troisièmement, les multiples et ingénieux développements méthodologiques qui ont étendu la discipline à d'innombrables tâches s'étendant de la vision par ordinateur (notamment grâce aux réseaux convolutionnels [265, 246, 186]) au traitement automatique des langues (pour lequel des performances supérieures ont été atteintes grâces

aux Transformers et aux modèles d'attention [429, 108]) en passant par la robotique (à travers l'apprentissage par renforcement [321, 394]), tout en rendant les algorithmes d'apprentissage de plus en plus efficaces.

Du point de vue mathématique, l'apprentissage automatique trouve son origine dans l'interaction des disciplines de la statistique, des probabilités et de l'optimisation. Cette interaction constituera un schéma récurrent le long des chapitres de ce document et reflète le fait qu'une grande partie des modèles son entraînés à travers la minimisation d'une fonction objectif quantifiant leur performance moyenne sur les échantillons d'un jeu de données. Formellement, le problème d'entraîner un modèle d'apprentissage automatique est souvent exprimé sous la forme suivante

$$\min_{\theta} \mathcal{L}(\theta) := \mathbb{E}_{\zeta}[\ell(\theta, \zeta)], \tag{1.0.1}$$

où θ représente les paramètres du modèle qui doivent être ajustés pour optimiser la performance. La *fonction de perte* ℓ évalue la précision du modèle de paramètre θ sur l'échantillon ζ. L'espérance définissant l'objectif $\mathcal{L}$ est calculée par rapport à la distribution généralement inconnue de ζ.

Le but est de concevoir des algorithmes de résolution efficaces pour des problèmes de la forme (1.0.1) qui utilisent un ensemble d'échantillons de ζ pour calculer un estimateur $\widehat{\theta}$ du minimum de $\mathcal{L}$. Les performances optimales d'un tel procédé en termes de vitesse de convergence et de propriétés statistiques de l'estimateur $\widehat{\theta}$ en sortie sont déterminées par les caractéristiques de la fonction de perte ℓ et de la distribution inconnue de ζ. Nous développerons davantage cet aspect dans la suite.

Ce chapitre introductif donne une description sommaire et informelle du travail de recherche constituant cette thèse qui est élaboré en détail plus tard. Une brève introduction à la statistique robuste est aussi incluse vue sa haute pertinence pour ce travail. Les Chapitres 3 et 4 sont dédiés aux projets présentant des algorithmes d'apprentissage linéaire robustes, le second se concentre sur le cas haute dimension. Le Chapitre 5 traite également le problème de l'apprentissage robuste et propose un algorithme basé sur SGD destiné au contexte où les données ne sont accessibles que sous la forme d'un flux d'échantillons plutôt qu'un jeu de données entier. Les autres chapitres sortent du cadre robuste. Le Chapitre 6 est basé sur des idées en commun avec le Chapitre 5 et contient une étude de l'algorithme SGD standard menant à de nouveaux résultats sur ses propriétés de convergence et de concentration. Le Chapitre 7 présente un nouvel algorithme de forêt aléatoire qui améliore les performances prédictives en usant de moins de ressources de calcul. Enfin, dans l'Appendice A, nous discutons un nouvel algorithme pour la régression logistique en ligne qui pourrait atteindre la borne de regret optimale avec une complexité bien inférieure aux méthodes actuellement connues atteignant cette performance. Tous les chapitres introduisent leur propre cadre formel et peuvent être lus indépendamment.

1.1 Statistique robuste

Le sujet central de cette thèse s'articule autour de la statistique robuste et des algorithmes d'apprentissage robustes en particulier. Le domaine de la statistique robuste est apparu dans les années 1960s et fût lancé par les travaux de Tukey, de Huber et autres [422, 202, 8, 177]. La motivation principale est de développer et analyser des méthodes d'estimation statistiques qui gardent de bonnes performances même sur des données contenant des instances anormales qui diffèrent de la majorité des échantillons. Concernant cet aspect, les idées de bases peuvent être retracées jusqu'au dix-huitième siècle dans les travaux de Boscovitch, Laplace et Edgeworth [142, 387, 140] qui étudièrent une forme de régression robuste utilisant la fonction de perte absolue plutôt que les moindres carrés pour minimiser l'impact de valeurs extrêmes.

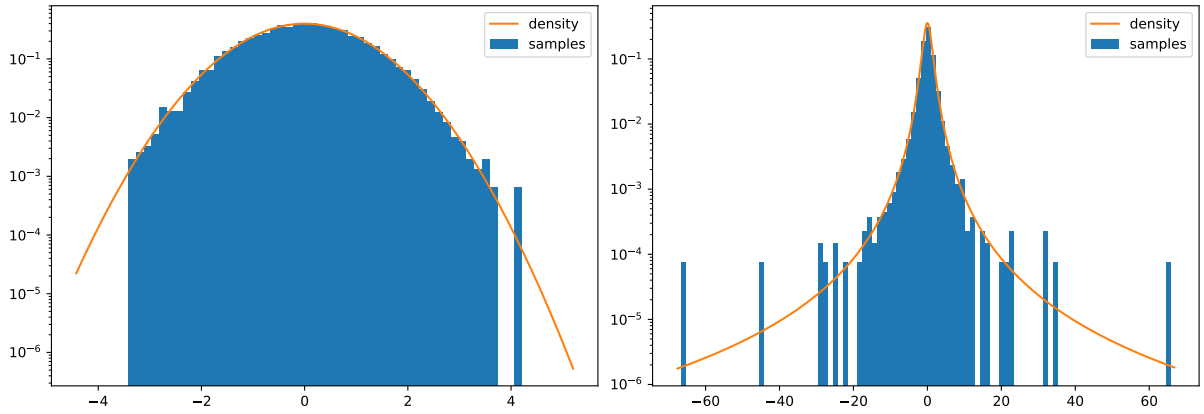

FIGURE 1.1 : Fonction de densité et échantillons d'une distribution Gaussienne standard (à gauche) et d'une distribution Student-t à 2.2 degrés de liberté. Noter que l'axe des y est en échelle logarithmique. Tandis que les échantillons Gaussiens sont concentrés autour de leur moyenne, les échantillons de la distribution Student se produisent souvent à plusieurs écart-types constituant des valeurs aberrantes.

Plus généralement, le but est d'étendre la méthodologie et les garanties théoriques à des hypothèses plus faibles sur les données. Par exemple, un cadre statistique très courant est de considérer des échantillons $X_1, \ldots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ Gaussiens indépendants et identiquement distribués (i.i.d). Cependant, cet ensemble d'hypothèses est souvent discutable en pratique : des exemples dans les variations du cours d'actifs financiers [296] et de données web [170] suggèrent fortement que la concentration Gaussienne n'est pas une propriété des données réelles où des distributions *à queue lourdes* sont souvent observées. De plus, des valeurs anormales venant d'erreurs de mesures, de *bêtises* ou de fautes lors du rassemblement des données se retrouvent souvent dans les jeux de données et constituent une part de 1–10% des échantillons [205] ce qui invalide l'hypothèse de distribution identique.

En tenant compte du fait que les problèmes d'apprentissage de la forme (1.0.1) sont optimisés grâce à des échantillons de données de distribution inconnue, la précision des estimations obtenues de l'optimum, et donc les performances des modèles entraînés, dépendent fortement de la qualité des données utilisées dans ce but. Les anciennes garanties [427, 157, 298] furent obtenues sous de fortes hypothèses impliquant des données i.i.d Gaussiennes ou bornées et en utilisant des moyennes empiriques. Par conséquent, des méthodes plus robustes doivent être conçues pour tenir compte du caractère irrégulier des données observé en pratique. Cela peut se manifester de plusieurs façons :

Les queues lourdes. Les données semblent souvent suivre une distribution à queue lourde. On dit qu'une distribution réelle est à queue lourde si sa fonction de densité n'est pas bornée exponentiellement au voisinage de l'infini. Cela est à la différence des distributions à queue légère telles que les distributions sous-Gaussiennes ou sous-exponentielles dont les queues ont respectivement une tendance en $\exp(-\Omega(x^2))$ ou $\exp(-\Omega(|x|))$ au voisinage de l'infini. La décroissance lente des distributions à queue lourde augmentent leur tendance à produire des échantillons extrêmes (valeurs aberrantes) qui s'égarent loin de la moyenne. Ceci est illustré sur la Figure 1.1 qui montre la densité et des échantillons de la distribution Gaussienne standard et d'une distribution Student-t qui est un exemple de queue lourde.

η-Contamination. Ceci est le premier modèle de données non i.i.d introduit par [202, 205] et généralement désigné comme le modèle d'η-contamination de Huber. Ce dernier postule que les échantillons sont tirés d'une distribution mélange $(1-\eta)P + \eta Q$ pour un taux de contamination $\eta \in]0, 1/2[$ avec P la vraie distribution de données ciblées par l'analyse et Q une distribution de contamination inconnue sans hypothèses. La source de données parasites est modélisée comme inconsciente et indépendante ce qui exclut un comportement adverse.

Données corrompues. Un cadre plus général est de supposer que l'ensemble des indices est divisé en une paire inconnue $\mathcal{I} \cup \mathcal{O}$ telle que les échantillons dans $\mathcal{I}$ suivent la vraie distribution cible et ceux de $\mathcal{O}$ sont arbitrairement corrompus. Le nombre de corruptions est toujours supposé inférieur à celui des vrais échantillons $|\mathcal{O}| < |\mathcal{I}|$ et peut représenter au plus une fraction $\eta \in]0, 1/2[$ du nombre total d'échantillons (η-corrpution). Ce cadre a été étudié dans [257, 261] et permet aux échantillons corrompus de dépendre des vrais et même des méthodes utilisées pour le traitement des données afin de modéliser une entité adverse.

L'estimation d'une moyenne est probablement l'une des tâche les plus basiques en statistique et peut être considérée comme block de base apparaissant dans la plupart des procédures. On propose d'explorer quelques approches à la robustesse à travers ce problème.

1.1.1 Estimation robuste d'une moyenne scalaire

D'abord, considérons le cas simple d'un échantillon i.i.d de variables Gaussiennes $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2)$. Le choix standard pour l'estimation de la moyenne μ est de calculer l'estimateur de maximum de vraisemblance qui, dans ce cas, correspond à la moyenne empirique

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i. \tag{1.1.1}$$

Sous ces hypothèses, cette dernière vérifie l'inégalité en grande probabilité suivante

$$\mathbb{P}\Big(|\widehat{\mu} - \mu| > \sqrt{\frac{2\sigma^2 \log(2/\delta)}{n}} \Big) \leqslant \delta. \tag{1.1.2}$$

Il est à remarquer que la borne de déviation est sous-Gaussienne et est de l'ordre de $\sqrt{\log(1/\delta)}$ pour les petits niveaux de confiance $\delta \to 0$ ce qui est optimal et correspond à la limite asymptotique du théorème central limite. Cependant, si on retire l'hypothèse que les X_i suivent une distribution Gaussienne et qu'on considère une distribution source à queue lourde de moyenne $\mathbb{E}[X] = \mu$ et de variance $\mathbb{E}[(X-\mu)^2] = \sigma^2$, alors en utilisant l'inégalité de Tchebyshev, on obtient ce qui suit pour la moyenne empirique

$$\mathbb{P}\Big(|\widehat{\mu} - \mu| > \sqrt{\frac{\sigma^2}{\delta n}} \Big) \leqslant \delta,$$

où la nouvelle dépendance en $\sqrt{1/\delta}$ aboutit à des intervalles de confiance bien plus larges pour les petites valeurs de δ. De plus, il est possible de concevoir des distributions qui saturent l'inégalité ci-dessus [73]. Ceci reflète la faible précision de la moyenne empirique pour les distributions à queue lourde.

Il est cependant possible de définir des estimateurs de la moyenne avec le taux sous-Gaussien de (1.1.2) pour des distributions satisfaisant seulement une hypothèse de variance finie. Un exemple particulier est donné par les M-estimateurs qui sont définis comme la solution en $\widehat{\mu}_\rho$ du

problème

$$\sum_{i=1}^{n} \operatorname{sign}(\widehat{\mu}_\rho - X_i)\rho\big(|\widehat{\mu}_\rho - X_i|/\beta\big) = 0 \tag{1.1.3}$$

où $\beta > 0$ est un paramètre d'échelle et ρ est une *fonction d'influence* telle que ρ est croissante sur $\mathbb{R}^+$ et satisfait $\rho(0) = 0$.

On voit facilement que la moyenne empirique est retrouvée en prenant ρ comme la fonction identité. Il est aussi à remarquer qu'en prenant ρ constante égale à 1, on retrouve la médiane. Ces deux choix extrêmes illustrent l'intention de la définition précédente de moduler l'influence d'échantillons individuels en interpolant entre la moyenne standard (qui un estimateur non biaisé mais non robuste) et la médiane (qui est généralement biaisée pour la moyenne mais très robuste). En particulier, l'influence de valeurs aberrantes sur l'estimateur $\widehat{\mu}_\rho$ est limitée en choisissant ρ telle que $\rho(x) = o(x)$ pour les grandes valeurs de x. Les principaux exemples pour ρ sont les suivants :

1. la fonction d'influence de Huber $\rho_H(x) = x\mathbf{1}_{x\leqslant 1} + \mathbf{1}_{x>1}$, qui fût introduite par [202].
2. La fonction d'influence de Catoni $\rho_C(x) = \log(1 + x + x^2/2)$, introduite par [73].
3. la fonction d'influence polynomiale $\rho_P(x) = \frac{x}{1+x^{1-1/p}}$ pour un degré polynomial $p > 1$.

Ces fonctions d'influence et les propriétés des M-estimateurs associés sont étudiées dans [300]. Pour un choix approprié de β (qui dépend de la variance des échantillons aléatoires), ces estimateurs ont une borne de déviation sous-Gaussienne similaire à (1.1.2). En particulier, L'estimateur de Catoni, obtenu avec $\rho_C(x)$, a la propriété remarquable de satisfaire cette borne avec des constantes optimales [110].

Les principaux inconvénients des M-estimateurs se trouvent dans la nécessité d'ajuster le paramètre d'échelle β, ce qui doit être fait selon la variance inconnue des données. De plus, la résolution de (1.1.3) peut être lente. De manière plus importante, si ρ est choisie non bornée, toutes garantie sur l'estimateur associé devient nulle dès que l'un des échantillons est corrompu (i.e. s'il est d'origine inconnue ou potentiellement adversaire)

La robustesse d'un estimateur aux échantillons corrompus peut se mesurer par la notion de *point d'échec* (en anglais "*breakdown point*") introduite par [129]. On dit qu'un estimateur a pour point d'échec $\varepsilon \in [0, 1/2]$ si ε est la plus petite proportion requise d'instances corrompues pour provoquer des valeurs arbitraires de la part de l'estimateur. Formellement, pour un estimateur $T : \cup_{n\geqslant 0}\mathbb{R}^n \to \mathbb{R}$, on propose de définir $\varepsilon := \lim_{n\to\infty} \varepsilon_n$ où

$$\varepsilon_n := \min\left\{\frac{m}{n}, m \geqslant 0 \ : \sup_{X_j, Y_j \,:\, \sum_j \mathbf{1}_{X_j \neq Y_j} \leqslant m} \big|T(X_1, \dots, X_n) - T(Y_1, \dots, Y_n)\big| = \infty\right\},$$

(voir aussi [299, Section 2.1.3]). Par exemple, la moyenne empirique a pour point d'échec $\varepsilon = 0$ puisqu'elle peut être infiniment biaisée par une seule valeur arbitraire (on dit aussi que le point d'échec à échantillon fini est $\varepsilon_n = 1/n$). plus généralement, la même chose est vraie pour les M-estimateurs avec ρ non bornée. Noter que le point d'échec le plus élevé possible est de $1/2$ puisqu'il devient impossible de distinguer la distribution des données de la corruption lorsque cette dernière affecte plus de la moitié des données [384]. Cette valeur de point d'échec est atteinte par la médiane et aussi par tout M-estimateur avec ρ bornée [379].

On mentionne deux estimateurs robustes supplémentaires d'une moyenne scalaire qui peuvent supporter des échantillons corrompus mais n'appartiennent pas aux M-estimateurs.

L'estimateur médiane des moyennes (MOM) [7, 217, 339] partitionne les échantillons en K

blocks de tailles égales $\{1, \dots, n\} = \cup_{k=1}^{K} B_k$ et calcule la médiane des moyennes par block

$$\widehat{\mu}^{K}_{\mathrm{MOM}} = \mathrm{median}\left(\Big(\frac{1}{|B_k|} \sum_{i \in B_k} X_i \Big)_{k=1}^{K} \right).$$

Sous l'hypothèse précédente de variance finie, L'estimateur MOM satisfait la borne suivante [284]

$$\mathbb{P}\Big(\big| \widehat{\mu}^{K}_{\mathrm{MOM}} - \mu \big| > 2\sigma\sqrt{K/n} \Big) \leqslant e^{-K/8},$$

où l'on peut choisir le nombre de blocks comme $K = \lceil 8 \log(1/\delta) \rceil$ pour un niveau de confiance δ de manière à obtenir une borne sous-Gaussienne comme dans (1.1.2). On peut également montrer qu'une telle borne est vérifiée si l'échantillon contient jusqu'à $K/2$ corruptions, bien qu'avec de pires constantes. Ces propriétés en plus de sa facilité de calcul font de MOM un estimateur robuste intéressant qui a trouvé de nombreuses applications [283, 259, 261, 196]. Cependant, le nombre d'échantillons corrompus autorisé est limité par le nombre de blocks K de sorte qu'il serait nécessaire de prendre $K = \Omega(n)$ pour avoir la robustesse à une fraction de corruptions dans les données. Ceci mène à un comportement de MOM plus proche de la médiane que de la moyenne de sorte qu'il n'est pas à adapté au cadre d'η-corruption.

Le dernier estimateur robuste de moyenne scalaire dont nous discuterons est la moyenne tronquée introduite par [288]. Étant donné un indice de quantile $\epsilon \in]0, 1/2[$, ce dernier estime la moyenne en calculant

$$\widehat{\mu}^{\epsilon}_{TM} = \frac{2}{n} \sum_{i=n/2+1}^{n} X^{(\epsilon n/2)} \vee X_i \wedge X^{((1-\epsilon)n/2)},$$

où $a \wedge b := \min(a, b), a \vee b := \max(a, b), [x]$ est la partie entière de $x \in \mathbb{R}$ et $X^{(j)}$ est la j-ème statistique d'ordre de $(X_i)_{i \leqslant n/2}$. L'échantillon est donc divisé en moitié. La première est utilisée pour estimer les quantiles d'ordre ϵ et $1 - \epsilon$ et la deuxième est utilisée pour estimer la moyenne en ramenant toutes les valeurs entre les deux quantiles précédents. L'estimateur ci-dessus est parfois appelé "la moyenne winsorizée" and ne doit pas être confondu avec d'autres définitions qui excluent simplement les valeurs à extérieur de l'intervalle de quantiles plutôt que les ramener dedans. Il est également à remarquer que la division de l'échantillon est nécessaire pour l'étude théorique pour s'assurer de l'indépendance de l'estimation des quantiles mais est généralement négligée en pratique. En considérant une suite $X_1, \dots, X_n$ d'échantillons tirés d'une distribution à queue lourde de variance finie σ^2 et contenant une proportion η de corruptions (potentiellement adversaires), l'estimateur de moyenne tronquée à quantile $\epsilon = 8\eta + 24\frac{\log(4/\delta)}{n}$ satisfait la borne de déviation suivante [288, Theorem 1]

$$\mathbb{P}\Big(\big| \widehat{\mu}^{\epsilon}_{\mathrm{TM}} - \mu \big| > 12\sigma\sqrt{2\eta} + 2\sigma\sqrt{\frac{\log(4/\delta)}{n}} \Big) \leqslant \delta. \tag{1.1.4}$$

Ainsi, l'estimateur de moyenne tronquée est sous-Gaussien tout en étant robuste à l'η-corruption et a la dépendance optimal en $\sqrt{\eta}$ en cette dernière. Noter que, similairement à l'estimateur MOM, la moyenne tronquée dépend d'un paramètre qui doit être fixé selon le niveau de confiance δ souhaité. Cette dépendance, qui se manifeste aussi pour les M-estimateurs comme la moyenne de Catoni, est en fait inévitable puisqu'aucun estimateur unique n'existe pour tout les niveau de confiance δ [110]. Revenant à la moyenne tronquée $\widehat{\mu}^{\epsilon}_{\mathrm{TM}}$, la seule tâche non triviale pour son calcul est la détermination des quantiles $X^{(\epsilon n/2)}$ et $X^{((1-\epsilon)n/2)}$ qui peut être effectuée en temps linéaire en utilisant l'algorithme de médiane des médianes (voir par exemple [98, Chapter 9]) ou

simplement l'algorithme quickselect [189]. Grâce à la réunion de ces propriétés, nous verrons dans les chapitres suivants que la moyenne tronquée mène généralement au meilleurs compromis entre forte robustesse et vitesse de calcul pour la résolution de plusieurs problèmes d'apprentissage dans le cadre robuste.

Les estimateurs précédents fournissent des solutions assez satisfaisantes pour l'estimation robuste de moyenne scalaire. Malheureusement, nous verrons que cela s'étend difficilement à l'estimation de moyenne multidimensionnelle pour laquelle la plupart des algorithmes manquent de robustesse ou sont coûteux en calculs. Ceci représente un obstacle important à l'apprentissage robuste puisque la solution de problèmes de la forme (1.0.1) dépend crucialement de l'estimation précise de gradients comme moyennes multidimensionnelles. Un moyen de contourner ce problème est exploré dans le Chapitre 3 en combinant des estimateurs scalaires robustes avec la descente de gradient par coordonnées (CGD) pour résoudre des problèmes d'apprentissage linéaire.

1.1.2 Estimation robuste de moyenne vectorielle

Avant d'explorer les estimateurs robustes, nous devons déterminer le meilleur taux de concentration atteignable. Dans le cas multidimensionnel, un choix plus large de façons de mesurer la distance entre un estimateur $\widehat{\mu}$ et la vraie valeur μ est possible et influence le type de bornes obtensibles. Puisque la distance Euclidienne est le choix le plus adopté, nous nous concentrerons sur ce cas ici. Un choix naturel de référence pour al concentration est de considérer le taux Gaussien asymptotique énoncé par le théorème central limite. Par conséquent, notre référence sera la borne de déviation satisfaite par la moyenne standard $\widehat{\mu}$ définie dans (1.1.1) sur un échantillon Gaussien i.i.d $X_1, \dots, X_n \sim \mathcal{N}(\mu, \Sigma)$ où $\mu \in \mathbb{R}^d$ pour $d > 1$ et $\Sigma \in \mathbb{R}^{d\times d}$ est une matrice de covariance symétrique définie positive. Dans ce cas, la borne en grande probabilité suivante peut être obtenue en utilisant l'inégalité de Borell-TIS [418, 50, 267]

$$\mathbb{P}\Big(\|\widehat{\mu} - \mu\| > \sqrt{\frac{\mathrm{Tr}(\Sigma)}{n}} + \sqrt{\frac{2\|\Sigma\|_{\mathrm{op}} \log(1/\delta)}{n}}\Big) \leqslant \delta,$$

où $\|\cdot\|_{\mathrm{op}}$ représente la norme d'opérateur. Un trait important de la borne sous-Gaussienne ci-dessus est que la déviation pour différent niveaux de confiance δ est modulée par $\|\Sigma\|_{\mathrm{op}}$ qui peut être plus petite que $\mathrm{Tr}(\Sigma)$ d'un facteur d la dimension de l'espace.

Estimateurs intractables

Un certain nombre d'estimateurs sont connus [131, 289, 288, 84] qui ne sont pas utilisables en pratique à cause de la complexité exponentielle de leur calcul par rapport à la dimension (intractable). A exemple remarquable est la *médiane de Tukey* qui est définie sur la base de la profondeur de Tukey. Pour un point $u \in \mathbb{R}^d$, sa profondeur de Tukey par rapport à un ensemble de points $v_1, \dots, v_n \in \mathbb{R}^d$ est définie comme

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\big|\{1 \leqslant j \leqslant n : \langle w, u - v_j\rangle > 0\}\big|,$$

où $|\cdot|$ représente le cardinal d'ensemble. la médiane de Tukey est alors définie en maximisant cette mesure de profondeur relativement à l'ensemble d'échantillons. Ceci se traduit par un point de *centralité maximale* similaire à l'estimateur sous-Gaussien de [289]. Il est à noter que le minimum par rapport à $w \in \mathbb{R}^d$ dans la définition peut être restreint aux vecteurs de norme unité. Les problèmes d'optimisation de ce type par rapport à toutes les directions de l'espace appraîssent également dans la définition d'autres estimateurs vectoriels robustes et sont la raison de leur intractabilité. L'étude effectuée par [84] a montrée que la médiane de Tukey est robuste

à l'η-contamination de Huber si la vraie distribution de données est Gaussienne avec covariance identité, auquel cas, elle satisfait une borne de déviation de la forme suivante

$$\mathbb{P}\Big(\|\widehat{\mu}_{\text{Tukey}} - \mu\| > \sqrt{\frac{d}{n}} \vee \eta + \sqrt{\frac{\log(1/\delta)}{n}}\Big) \leqslant \delta.$$

Ces hypothèses relativement restrictives sont relaxées par l'estimateur de moyenne tronquée généralisée [288] qui satisfait la borne de déviation optimale suivante pour des données η-corrompues à covariance finie Σ :

$$\mathbb{P}\Big(\|\widehat{\mu}_{\text{TM}} - \mu\| > C\Big(\sqrt{\|\Sigma\|_{\text{op}}\eta} + \sqrt{\frac{\text{Tr}(\Sigma)}{n}} + \sqrt{\frac{\|\Sigma\|_{\text{op}}\log(1/\delta)}{n}}\Big)\Big) \leqslant \delta, \tag{1.1.5}$$

où C est une constante absolue. Il est à remarquer que la dépendance en le taux de corruption, précédemment en η pour la données Gaussiennes, empire en $\sqrt{\eta}$ dans le cas de covariance finie.

Malheureusement, la définition de la moyenne tronquée généralisée est basée sur le calcul des quantiles des vecteurs de données projetés par rapport à toutes les directions de l'espace. Par conséquent, une implémentation directe aurait une complexité exponentielle en la dimension ce qui rend l'estimateur intractable.

Estimateurs robustes de complexité polynomiale

Nous nous tournons maintenant vers les estimateurs calculables. Une première idée naïve est d'utiliser un estimateur scalaire robuste séparément pour chaque coordonnée. Si l'estimateur de base est sous-Gaussien, alors pour des échantillons à covariance finie Σ, ceci mène à une borne de la forme suivante

$$\mathbb{P}\Big(\|\widehat{\mu} - \mu\| > C\sqrt{\frac{\text{Tr}(\Sigma)\log(d/\delta)}{n}}\Big) \leqslant \delta,$$

où C est une constante absolue. Malheureusement, cette nouvelle borne est clairement moins bonne puisque le niveau de confiance a dû être remplacé par δ/d pour utiliser une borne d'union. De manière plus importante, les fluctuations sont modulées par $\text{Tr}(\Sigma)$ plutôt que $\|\Sigma\|_{\text{op}}$ ce qui implique une pire dépendance en la dimension.

Une solution légèrement meilleure est fournie par une généralisation multidimensionnelle de l'estimateur MOM appelée la médiane des moyennes géométrique qui fût proposée par [315] (voir aussi [196] pour une autre généralisation ayant des propriétés similaires). Cette dernière calcule des moyennes par block et les combine grâce à la généralisation suivante de la médiane pour des vecteurs $v_1, \dots, v_K \in \mathbb{R}^d$:

$$\text{median}(v_1, \dots, v_K) = \operatorname*{arg\,min}_{u \in \mathbb{R}^d} \sum_{i=1}^{K} \|u - v_i\|.$$

En fixant le nombre de blocks à $K = \lceil 8\log(1/\delta)\rceil$, l'estimateur obtenu satisfait la borne légèrement meilleure

$$\mathbb{P}\Big(\|\widehat{\mu} - \mu\| > C\sqrt{\frac{\text{Tr}(\Sigma)\log(1/\delta)}{n}}\Big) \leqslant \delta,$$

avec l'avantage additionnel de robustesse à un nombre $\leqslant K/2$ de valeurs aberrantes dans l'échantillon. Par contre, cet estimateur n'est toujours pas sous-Gaussien vu le facteur $\text{Tr}(\Sigma)$ devant $\log(1/\delta)$.

Le premier estimateur de moyenne vectoriel sous-Gaussien et de complexité polynomiale fût proposé par [195] en concevant un algorithme capable de certifier le critère de centralité de [289] en utilisant l'optimisation semi-définie positive (SDP). Cette approche par certification de concen-

tration est aussi la base des algorithmes robustes d'estimation de moyenne utilisant le paradigme des "sommes de carrés" (*Sum-Of-Squares*) [244, 194] et dont la complexité est également polynomiale. Des améliorations progressives dans des travaux ultérieurs ont baissé la complexité jusqu'au temps quasi-linéaire [92, 107]. Ces résultats qui ont initialement considéré peu ou pas de corruption au départ ont été complémentés par [268, 117] qui ont introduit l'η-corruption adversaire et ont obtenu des performances se rapprochant de (1.1.5) (à facteurs logarithmiques près) tout en remplaçant l'optimisation SDP par des méthodes spectrales plus simples.

L'approche par stabilité pour l'estimation de moyenne vectorielle robuste

Le formalisme de [117] représente l'état de l'art à ce jour et mérite une discussion plus élaborée. Ils conçoivent des algorithmes qui effectuent un filtrage itératif sur l'échantillon des données et obtiennent un estimateur $\widehat{\mu}_{\mathrm{DKP}}$ satisfaisant la borne suivante pour un échantillon η-corrompu par un adversaire et suivant une distribution de covariance finie [117, Proposition 1.5]

$$\mathbb{P}\Big(\|\widehat{\mu}_{\mathrm{DKP}} - \mu\| > C\Big(\sqrt{\|\Sigma\|_{\mathrm{op}}\eta} + \sqrt{\frac{\mathrm{Tr}(\Sigma)\log(\mathrm{r}(\Sigma))}{n}} + \sqrt{\frac{\|\Sigma\|_{\mathrm{op}}\log(1/\delta)}{n}}\Big)\Big) \leqslant \delta, \qquad (1.1.6)$$

où $\mathrm{r}(\Sigma) = \mathrm{Tr}(\Sigma)/\|\Sigma\|_{\mathrm{op}}$ est le rang stable de Σ. Le facteur $\log(\mathrm{r}(\Sigma)) \leqslant \log(d)$ est la seule sous-optimalité dans cette borne et peut être supprimé grâce à une étape de pré-traitement par Médiane-de-moyennes. Cependant, la quantité de corruption η toléré par cette méthode est fortement restreinte à ce moment (voir aussi Section 4.5.2). Le formalisme de [117] est basé sur la propriété suivante.

Définition 1.1 (Stabilité)**.** *Fixons $\epsilon \in (0, 1/2)$ et $\tau \geqslant \epsilon$. Un ensemble fini $S \subset \mathbb{R}^d$ est dit (ϵ, τ)-stable par rapport à $\mu \in \mathbb{R}^d$ et σ^2 si pour tout $S' \subseteq S$ avec $|S'| \geqslant (1-\epsilon)|S|$ on a*

$$\|\mu_{S'} - \mu\| \leqslant \sigma\tau \quad \text{et} \quad \|\overline{\Sigma}_{S'} - \sigma^2 I_d\|_{\mathrm{op}} \leqslant \sigma^2\tau^2/\epsilon,$$

où $\mu_{S'}$ et $\overline{\Sigma}_{S'}$ sont la moyenne et la covariance de l'ensemble S' respectivement.

Étant donnée cette définition, l'énoncé mathématique clef motivant l'approche par stabilité est le suivant.

Théorème 1.1 ([117, Théorème 1.4] informel)**.** *Soient $\delta > 0, \eta > 0$ et $S = \{X_1, \ldots, X_n\}$ un ensemble d'échantillons i.i.d d'une distribution sur $\mathbb{R}^d$ de moyenne μ et covariance Σ. Supposons que $\epsilon := \Theta(\log(1/\delta)/n + \eta)$ soit suffisamment petit. Avec probabilité au moins $1-\delta$, il existe un sous-ensemble $S' \subseteq S$ tel que $|S'| \geqslant (1-\epsilon)n$ et S' est $(2\epsilon, \tau)$-stable par rapport à μ et $\|\Sigma\|_{\mathrm{op}}$ où $\tau = O\big(\sqrt{\mathrm{r}(\Sigma)\log\mathrm{r}(\Sigma)/n} + \sqrt{\eta} + \sqrt{\log(1/\delta)/n}\big)$.*

En conséquence de ce résultat, avec grande probabilité, un ensemble η-corrompu d'échantillons suivant une distribution de covariance finie contient un sous-ensemble (ϵ', τ')-stable où (ϵ', τ') sont equivalents à (ϵ, τ) dans l'énoncé ci-dessus à facteurs constants près. Par suite, le calcul d'un estimateur satisfaisant (1.1.6) peut se faire en identifiant un tel sous-ensemble et en utilisant la première inégalité de la Définition 1.1. Cette exigence est en fait relaxée à la recherche de poids $w^* \in \mathbb{R}^n_+$ avec $\sum_i w_i^* = 1$ sur l'ensemble des échantillons pour calculer l'estimateur comme

$$\widehat{\mu}_{\mathrm{DKP}} = \mu_{w^*} := \sum_i w_i^* X_i,$$

The task of determining w^* is executed by an iterative filtering algorithm which proceeds roughly as follows (see [116, 117] for details). Given ϵ and τ, start from uniform weights $w \in \mathbb{R}^n_+$ and loop over the steps :

La détermination de w^* se fait par un algorithme de filtrage itératif qui procède à peu près comme suit (voir [116, 117] pour plus de détails). Étant donnés ϵ et τ, commencer à partir de poids uniformes $w \in \mathbb{R}^n_+$ et répéter les étapes suivantes :

- Calculer la moyenne et la covariance pondérées

$$\mu_w = \sum_i w_i X_i \quad \text{et} \quad \Sigma_w = \sum_i w_i (X_i - \mu_w)(X_i - \mu_w)^\top,$$

  par rapport aux poids actuels.
- Calculer le vecteur propre v de Σ_w associé à la plus grande valeur propre.
- Diminuer les poids de la proportion ϵ des échantillons à plus grande contribution à la variance suivant la direction de v.

La boucle continue jusqu'à ce que le critère d'arrêt $\|\Sigma_w\|_{\text{op}} \leqslant \|\Sigma\|_{\text{op}}(1 + O(\tau^2/\epsilon))$ soit satisfait.

La complexité de cet algorithme est déclarée polynomiale mais n'est pas précisément spécifiée par les auteurs de [117]. En pratique, ce procédé implique plusieurs boucles sur les données avec des opérations d'algèbre linéaire non négligeable à chaque tour. Ceci fait que cette méthode est sensiblement plus lente que le calcul de la médiane des moyennes géométrique par exemple.

Estimation de moyenne vectorielle robuste par rapport à une métrique non-Euclidienne

Comme mentionné plus haut, la mesure de précision d'un estimateur de moyenne vectoriel dépend de la norme choisie pour ce faire. Pour une norme arbitraire, la première question importante est de déterminer le taux optimal correspondant qui servira de repère pour juger d'un estimateur candidat. Cette question fût traitée par [285] qui a donné une borne inférieure basée sur l'entropie sur la performance optimale ainsi qu'un estimateur (intractable) pour l'atteindre. Cette borne inférieure fût prouvée étroite au détail près que le caractère large de la borne de minoration de Sudakov [267, 432] peut mener à un écart logarithmique dans le cas Euclidien. Cet écart fût comblé par [106] en remplaçant l'entropie par la largeur Gaussienne (Gaussian width) comme mesure de complexité statistique. [106] propose également un algorithme d'optimisation convexe pour l'estimation sur données corrompues et à queue lourde.

Dans le Chapitre 4, nous considérons le problème apprentissage linéaire robuste dans le cadre de haute dimension. Nous utilisons une approche basée sur l'optimisation non-Euclidienne pour minimiser l'impact de la haute dimension. Ceci implique que l'erreur d'estimation du gradient est mesurée par une norme non-Euclidienne. Cet avantage est utilisé pour proposer une procédure d'apprentissage robuste et efficace utilisant des estimateurs adaptés.

1.1.3 La robustesse pour les flux de données

Jusqu'à maintenant, nous avons discuté de l'estimation robuste dans la cas *batch* où l'on suppose un accès libre à l'intégralité des échantillons de données sans contrainte de mémoire particulière. Cependant, de nombreux contextes existent où ce luxe n'est pas permis et l'on doit constituer un estimateur en utilisant seulement un échantillons individuel à la fois sans la possibilité de revoir les précédents. Cette situation se présente, par exemple, pour l'optimisation stochastique sur flux de données [47, 49, 307] et dans les problèmes d'apprentissage en ligne [183, 77, 347] où les données sont disponibles sous forme de flux d'échantillons.

Nous proposons de discuter la robustesse pour les flux de données à travers un problème d'optimisation stochastique avec une fonction objectif similaire à (1.0.1).

$$\min_\theta \mathcal{L}(\theta) := \mathbb{E}_\zeta[\ell(\theta, \zeta)]. \tag{1.1.7}$$

On suppose que l'objective $\mathcal{L}$ satisfait au moins les propriétés de convexité et de gradient Lipschitz et qu'il admet un minimum $\mathcal{L}^\star$ de sorte qu'il puisse être optimisé par une méthode de gradient. Il est à remarquer que l'estimation de moyenne peut être formulée comme cas particulier de ce problème en posant $\zeta = X \in \mathbb{R}^d$ et $\ell(\theta, X) = \|\theta - X\|^2$. Étant donnée une suite d'échantillons $\zeta_1, \ldots, \zeta_T$, l'objectif précédent peut être optimisé par la descente de gradient stochastique (SGD) avec échantillons de gradient $\nabla\ell(\theta, \zeta_t)$ (en supposant la différentiabilité de ℓ) à travers l'itération

$$\theta_{t+1} = \theta_t - \beta_t \nabla\ell(\theta_t, \zeta_t) \tag{1.1.8}$$

lancée à partir de $\theta_0 \in \mathbb{R}^d$ avec avec pas de descente $\beta_t > 0$. La qualité d'un estimateur $\widehat{\theta}_T$ après T itérations est mesurée soit en termes de l'excès de risque $\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}^\star$ soit en termes de la distance carrée $\|\widehat{\theta}_T - \theta^\star\|^2$ à un $\theta^\star$ optimal tel que $\mathcal{L}(\theta^\star) = \mathcal{L}^\star$ lorsque ce dernier existe et est unique (c'est le cas pour les fonctions fortement convexes par exemple).

Anciennes garanties pour SGD

Des travaux anciens on obtenu des garanties en espérance sur le défaut d'optimalité après T itérations. Par exemple, [371] obtint le résultat suivant.

Théorème 1.2 ([371, Théorème 1])**.** *Supposons que la fonction objectif $\mathcal{L}$ soit μ-fortement convexe et L-gradient Lipschitz et que $\mathbb{E}[\|\nabla\ell(\theta_t, \zeta_t)\|^2] \leqslant G^2$ pour tout t. Soit θ_T le résultat de T itérations de* (1.1.8) *à partir de $\theta_0 = 0$ et avec pas de descente $\beta_t = (\mu t)^{-1}$, on a*

$$\mathbb{E}[\mathcal{L}(\theta_T) - \mathcal{L}^\star] \leqslant \frac{2LG^2}{\mu^2 T}. \tag{1.1.9}$$

Bien qu'ils garantissent qu'une solution précise soit trouvée en moyenne, les résultats en espérance donnent peu de confiance pour une estimation particulière $\widehat{\theta}$. En effet, l'application de l'inégalité de Markov à l'Inégalité (1.1.9) donne seulement une borne sur $\mathcal{L}(\theta_T) - \mathcal{L}^\star$ avec une dépendance en δ^{-1} pour le niveau de confidence δ.

L'obtention de bornes avec des fluctuations mieux contrôlées est possible en supposons des propriétés de concentration sur les gradients aléatoires. En particulier, les hypothèses concernent généralement le comportement des erreurs de gradient

$$\varepsilon_\zeta(\theta) := \nabla\ell(\theta, \zeta) - \nabla\mathcal{L}(\theta). \tag{1.1.10}$$

L'un des premiers résultats de ce type est paru dans [17] qui a utilisé une hypothèse L_p sur $\varepsilon_\zeta(\theta)$ (avec $p > 1$) pour montrer la convergence L_p de $\mathcal{L}(\overline{\theta}_T) - \mathcal{L}^\star$ pour un pas de descent constant $\beta = O(1/p)$ où $\overline{\theta}_T = T^{-1}\sum_{t=0}^{T-1}\theta_t$ est la moyenne des itérés. Cela mène à une borne de meilleure confiance de l'ordre de $O(\delta^{-1/p})$ sur le défaut d'optimalité $\mathcal{L}(\overline{\theta}_T) - \mathcal{L}^\star$ pour le niveau de confiance δ (nous négligeons les autres constantes impliquées pour simplifier, voir les Théorème 2 et Corollaire 1 de [17] pour plus de détails).)

Résultats de haute-confiance pour SGD

On s'intéresse aux bornes de haute-confiance sur le défaut d'optimalité en termes de $\log(1/\delta)$ pour un niveau de confiance δ. Par exemple, un tel résultat fût présenté par [179] qui définit un estimateur $\widehat{\theta}_T$ basé sur T itérations de SGD et satisfaisant la borne suivante

$$\mathbb{P}\Big(\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}^\star \geqslant O\Big(\frac{1}{T} \cdot \frac{L \cdot \log(1/\delta) + L^2}{\mu}\Big)\Big) \leqslant \delta,$$

où l'objectif $\mathcal{L}$ est supposé L-Lipschitz et μ-fortement convexe. Cependant, la borne précédente a été prouvée sous la forte condition de bornitude des erreurs de gradient $\varepsilon_\zeta(\theta)$. Des résultats similaires furent obtenus pour des erreurs de gradient sous-Gaussiennes par [338] mais cela reste une forte hypothèse. Du point de vue des statistiques robustes, une première question est si de telles bornes de haute-confiance peuvent être prouvées lorsque la distribution de $\varepsilon_\zeta(\theta)$ est à queue lourde.

Ceci fût accompli pour la première fois par [166] qui considère un objectif L-gradient Lipschitz et suppose une borne uniforme sur la variance de l'erreur de gradient

$$\mathbb{E}[\|\varepsilon_\zeta(\theta)\|^2] \leqslant \sigma^2 < +\infty. \tag{1.1.11}$$

Pour cela, l'itération de SGD tronqué avec minibatch est utilisée

$$\theta_{t+1} = \theta_t - \beta_t \operatorname{clip}\big(\widehat{\nabla}_m \ell(\theta_t), \lambda_t\big) \quad \text{with} \quad \operatorname{clip}(v,\lambda) := \min\Big(1, \frac{\lambda}{\|v\|}\Big) \cdot v, \tag{1.1.12}$$

où $\beta_t > 0$ sont des pas de descente, $\lambda_t > 0$ des seuils de troncage et $\widehat{\nabla}_m \ell(\theta) = \frac{1}{m}\sum_{j=1}^m \nabla \ell(\theta, \zeta_j)$ est une moyenne de gradient minibatch de taille m. Ceci conduit à l'énoncé suivant

Théorème 1.3 ([166, Théorème 3.1] informel)**.** *Soit $\mathcal{L}$ convexe et L-gradient Lipschitz avec minimum unique $\theta^\star \in \mathbb{R}^d$ et supposons* (1.1.11)*. Soit $\theta_0 \in \mathbb{R}^d$ un paramètre initial tel que $R_0 = \|\theta_0 - \theta^\star\|$ et $T \geqslant 1$ l'horizon. Pour un niveau de confiance $\delta > 0$, supposons que l'Itération* (1.1.12) *soit exécutée avec*

$$m = \Theta\Big(\frac{T\sigma^2}{R_0^2 L^2 \log(T/\delta)}\Big), \quad \lambda_t = \lambda = \Theta(LR_0) \quad \textit{et} \quad \beta_t = \beta = O\big((L\log(T/\delta))^{-1}\big),$$

alors on a la borne suivante pour $\overline{\theta}_T = \frac{1}{T}\sum_{t=0}^{T-1}$:

$$\mathbb{P}\Big(\mathcal{L}(\overline{\theta}_T) - \mathcal{L}(\theta^\star) \geqslant O\Big(\frac{LR_0^2 \log(T/\delta)}{T}\Big)\Big) \leqslant \delta. \tag{1.1.13}$$

Cet article fût cependant précédé par [330] qui obtint des garanties similaires pour la descente stochastique miroir (SMD) en utilisant une méthode de troncage semblable. Toujours grâce à SGD tronqué, [417] a prouvé un résultat équivalent pour les objectifs fortement convexes sans minibatching et en utilisant des pas décroissants $\beta_t = \Theta(1/t)$ et un seuil de troncage fixé $\lambda = \Omega\big(\sqrt{T/\log(1/\delta)}\big)$. À part le fait d'éliminer la nécessité du minibatching, la principale amélioration apportée par [417] est de relaxer l'hypothèse de borne uniforme sur la variance (1.1.11) vers

$$\mathbb{E}[\|\varepsilon_\zeta(\theta)\|^2] \leqslant a\|\theta - \theta^\star\|^2 + b \quad \text{avec} \quad a, b > 0,$$

qui est une hypothèse bien plus raisonnable pour des problèmes habituels comme la régression linéaire. Il est important de remarquer que [166] et [417] utilisent tous les deux des seuils de troncage élevés par rapport aux valeurs attendues. Pour [166], des moyennes minibatch de taille $m = \Omega(T)$ sont utilisées de sorte que, au voisinage de l'optimum, $\|\widehat{\nabla}_m \ell(\theta)\|$ est d'ordre $O(1/\sqrt{T})$ tandis que seuil appliqué est d'ordre constant. De manière analogue, [417] utilise $m = 1$ de sorte que $\|\widehat{\nabla}_m \ell(\theta)\|$ soit d'ordre constant près de l'optimum et le seuil de troncage est fixé à $\Omega(\sqrt{T})$. Cette valeur élevée est compensée par le pas de descente décroissant qui est de l'ordre de $O(1/T)$ vers la fin de l'itération. Dans les deux cas, l'estimateur final $\widehat{\theta}$ satisfait $\|\widehat{\theta} - \theta^\star\| = O(1/\sqrt{T})$ qui mène au taux $O(1/T)$ comme dans (1.1.13) (à facteur logarithmique près) puisque la propriété

L-gradient Lipschitz implique

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) \leqslant \frac{L}{2}\|\theta - \theta^\star\|^2.$$

Des travaux ultérieurs [389, 346] ont davantage affaibli les hypothèses sur $\varepsilon(\theta)$ à l'existence d'un moment d'ordre α fini avec $\alpha \in]1, 2]$. Noter que dans ce cas, le taux optimal se dégrade de $O(1/T)$ à $O(T^{-2(\alpha-1)/\alpha})$. Des développements supplémentaires pour des objectifs non gradient-Lipschitz fûrent également apportés récemment par [277].

Robustesse à la corruption pour les flux de données

Au moment de cette écriture, peu de travaux ont considéré des méthodes robustes à la corruption dans le cas en ligne. La principale référence à ce sujet est [118] qui traite de l'estimation de moyenne vectorielle à partir d'un flux d'échantillon avec corrompus par un adversaire et à queue lourde. L'algorithme proposé adapte la procédure de filtration qui paru dans [117] pour cas batch en la restreignant à des minibatchs de taille logarithmique et en introduisant un certain nombre d'optimisations pour les opérations d'algèbre linéaire nécessaires pour éviter une complexité en $O(d^2)$ par itération. L'algorithme obtenu atteint les taux robustes optimaux et des applications de ce dernier sont explorées pour les tâches d'apprentissage convexes telles que la régression linéaire ou logistique. Cependant, l'étude de l'algorithme est purement théorique et son implémentation risque d'être difficile vu les étapes complexes et le nombre de constantes absolues inconnues dont il dépend.

Au Chapitre 5, nous proposons un stratégie de troncage spéciale pour SGD pour le rendre robuste à la fois aux queues lourdes et à la corruption. La procédure proposée est assez simple à implémenter et est appuyée par une analyse théorique rigoureuse confirmant sa robustesse.

1.2 Apprentissage robuste avec modèles linéaires

En apprentissage automatique, la question est souvent de concevoir un modèle qui prédise une réponse $Y \in \mathcal{Y}$ à partir de *features* $X \in \mathcal{X}$ aussi précisément que possible. On considère le cas très courant où l'espace des features est Euclidien $\mathcal{X} = \mathbb{R}^d$. Un modèle de prédiction est une application mesurable $\phi : \mathcal{X} \to \widehat{\mathcal{Y}}$ qui, à une instance X associe $\phi(X)$ qui prédit la réalité Y. La précision d'un modèle peut être mesurée de différentes manières. Celles ci utilisent généralement une fonction de perte $\ell : \widehat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}_+$ qui évalue l'écart entre prédiction et réalité par $\ell(\phi(X), Y)$. La performance globale de ϕ est mesurée par son *risque* qui correspond à l'espérance par rapport à la distribution de données de (X, Y) :

$$\mathcal{R}(\phi) := \mathbb{E}[\ell(\phi(X), Y)].$$

Selon la nature des labels Y qui doivent être prédits, on appelle la tâche d'apprentissage un problème de régression lorsque $Y \in \mathcal{Y} = \mathbb{R}$ est une valeur réelle et une problème de classification lorsque $Y \in \mathcal{Y} = \{\pm 1\}$ est une classe binaire (des extensions multi-classe peuvent aussi être envisagées). À ce moment, la fonction de perte ℓ est choisie selon la tâche d'apprentissage.

Le prédicteur ϕ est généralement sélectionné parmi une classe de fonctions Φ sur la base de sa performance sur un échantillon d'entraînement $(X_1, Y_1), \ldots, (X_n, Y_n)$. Cette étape vise à trouver un prédicteur minimisant le risque $\mathcal{R}(\phi)$ avec de bonnes propriétés de généralisation i.e. une bonne performance de prédiction sur de nouveaux échantillons de test en dehors de l'échantillon d'entraînement. Évidemment, la taille de la classe Φ détermine la performance atteignable selon la capacité des fonctions $\phi \in \Phi$ à exprimer la dépendance entre X et Y. Il faut cependant faire

attention au piège d'utiliser des modèles trop complexes puisque cela peut compliqué la tâche de trouver le prédicteur optimal ou mener au *sur-apprentissage*.

En effet, il est commun d'entraîner des modèles de prédictions en minimisant le *risque empirique* par rapport à un échantillon d'entraînement $(X_i, Y_i)_{i=1}^n$ défini par

$$\widehat{\mathcal{R}}_n(\phi) := \frac{1}{n}\sum_{i=1}^{n} \ell(\phi(X_i), Y_i). \tag{1.2.1}$$

Cependant, $\widehat{\mathcal{R}}_n$ pourrait ne pas être un bon substitut pour $\mathcal{R}$ et le sur-apprentissage se produit lorsqu'un prédicteur ϕ atteint une haute performance sur les données d'entraînement mesurée par $\widehat{\mathcal{R}}_n$ mais ne généralise pas sur de nouvelles données de test, ce qui suggère une haute valeur du risque $\mathcal{R}$ [329, 60]. Ce phénomène est exacerbé par les petites tailles d'échantillon n et par une classes de fonctions Φ excessivement complexes qui ont tendance à capturer trop de bruit à partir des données. Une solution pour atténuer ce problème est d'ajouter un terme de régularisation au risque empirique (1.2.1) qui pénalise les prédicteurs selon leur complexité et favorise des solutions plus "simples" [36].

Les modèles linéaires correspondent à prendre Φ comme la classe des fonctions linéaires ce qui est un choix courant pour plusieurs problèmes. En effet, cet ensemble de modèles fourni souvent une puissance de prédiction satisfaisante toute permettant l'usage de méthodes d'optimisation génériques pour déterminer un bon candidat $\phi^\star \in \Phi$. Chaque modèle peut être représenté par un vecteur $\theta \in \mathbb{R}^d$ tel que $\phi(X) = \theta^\top X$ et le problème de trouver un bon prédicteur $\phi^\star$ est équivalent à celui de déterminer le vecteur associé $\theta^\star$. En choisissant une fonction de perte ℓ convexe et gradient-Lipschitz, le problème de trouver un prédicteur à bas risque revient à celui de l'optimisation convexe

$$\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta) := \mathbb{E}[\ell(\theta^\top X, Y)], \tag{1.2.2}$$

qui peut être résolu efficacement grâce à une méthode de gradient. Deux des exemples les plus courants dans ce cadre sont :

- La régression moindres carrées : qui correspond à la prédiction d'une valeur réelle Y en définissant l'objectif à partir de la perte carrée $\ell(y, z) = \frac{1}{2}(y - z)^2$.
- La régression logistique : dans laquelle un label binaire $Y \in \{\pm 1\}$ est prédit en optimisant l'objectif $\mathcal{L}$ définie avec la fonction de perte logistique $\ell(y, z) = \log(1 + \exp(-yz))$.

La cadre de prédiction linéaire peut aussi être étendu à des modèles plus complexes en considérant des features polynomiaux en les variables de X ou des méthodes à noyaux [403, 245].

Cependant, la méthode à adopter pour résoudre le problème (1.2.2) n'est pas tout à fait évidente puisque les données (X, Y) suivent une distribution qui est inconnue en général et n'est accessible qu'à travers l'échantillon d'entraînement $(X_i, Y_i)_{i=1}^n$. Comme mentionné plus haut, une solution standard est de remplacer le vrai risque $\mathcal{L}$ par le risque empirique

$$\widehat{\mathcal{L}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta^\top X_i, Y_i), \tag{1.2.3}$$

qui est défini à partir de l'échantillon d'entraînement, et peu être directement optimisé pour obtenir un estimateur $\widehat{\theta}_{\text{erm}}$ (minimum du risque empirique). Pour garantir que ce dernier est bon optimiseur de $\mathcal{L}$, il est nécessaire que $\widehat{\mathcal{L}}_n$ fournisse une bonne approximation de l'objectif réel $\mathcal{L}$ dans le sens ou la différence $|\widehat{\mathcal{L}}_n(\theta) - \mathcal{L}(\theta)|$ satisfait une borne sous-Gaussienne semblable à (1.1.2). Ceci peut être prouvé sous de fortes hypothèses sur la distribution des données (X, Y) telle que la concentration sous-Gaussienne ou la bornitude [427, 298, 157]. Cependant, ces garanties ne

tiennent pas pour des données à queue lourde ou corrompues auquel cas une méthode plus robuste est nécessaire. Pour un problème de régression, une ancienne proposition fût de remplacer la perte carrée par la perte de Huber

$$\ell_H(y,z) = \begin{cases} \frac{1}{2}(y-z)^2 & \text{si} \quad |y-z| \leqslant 1 \\ |y-z| - \frac{1}{2} & \text{sinon.} \end{cases} \tag{1.2.4}$$

Cette dernière est capable de minimiser l'effet de valeurs aberrantes grâce à son taux linéaire plutôt que quadratique pour $|y-z| \gg 1$. Bien que cela permette traiter le cas des labels Y à queue lourde, ça reste insuffisant quand les hypothèses sur les covariées X sont affaiblies. Une autre possibilité est d'utiliser des estimations scalaire's robustes répétées de l'objectif $\mathcal{L}$ pour l'optimiser [57]. Par contre, cette procédure s'avère instable et difficile à implémenter.

Vu que les méthodes de gradient constituent la meilleure approche pour les tâches d'optimisation en général, une idée naturelle est de se concentrer sur l'estimation du gradient de $\mathcal{L}$ plutôt que l'objectif même et d'exécuter une descente de gradient [364]. Malheureusement, cela requière usage d'algorithmes d'estimation de moyenne vectorielle robustes qui, malgré de nombreux efforts, restent coûteux en calcul ou manquent de fiabilité dans certains cadres robustes comme discuté en Section 1.1.2.

1.2.1 Contribution : Apprentissage robuste par la descente de gradient par coordonnées

Pour permettre l'apprentissage robuste sans être confronté à la difficulté de l'estimation robuste de moyenne vectorielle, nous proposons au Chapitre 3 d'effectuer la descente de gradient par coordonnées (CGD) [341, 442] en utilisant un estimateur robuste scalaire des dérivées partielles. Pour des itérés $\theta^{(t)}$, ceci correspond à la mise à jour

$$\theta_j^{(t+1)} = \begin{cases} \theta_j^{(t)} - \beta_j \widehat{g}_j(\theta^{(t)}) & \text{si } j = j_t \\ \theta_j^{(t)} & \text{sinon,} \end{cases} \tag{1.2.5}$$

où $1 \leqslant j_t \leqslant d$, les β_js sont des pas de descente et $\widehat{g}_j(\theta^{(t)})$ est un estimateur robuste de $\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial \theta_j}$. Pour un tel estimateur $\widehat{g}$ des dérivées partielles, on définit son vecteur d'erreur $\epsilon(\delta)$ qui, pour une probabilité d'échec $\delta \in (0,1)$, satisfait pour tout $j \in [\![d]\!]$:

$$\mathbb{P}\Big[\sup_{\theta} \big|\widehat{g}_j(\theta) - g_j(\theta)\big| \leqslant \epsilon_j(\delta) \Big] \geqslant 1 - \delta.$$

Vue cette définition, nous avons le résultat suivant pour l'apprentissage robuste avec CGD.

Théorème 1.4 (Théorème 3.1 informel)**.** *Supposons que l'objectif $\mathcal{L}(\theta)$ soit λ-fortement convexe et L_j-gradient-Lipschitz par rapport à chaque coordonnée θ_j. Soit $\theta^{(T)}$ la sortie de l'Itération* (1.2.5) *avec pas de descente $\beta_j = 1/L_j$, un vecteur initial $\theta^{(0)}$, et des coordonnées j_t échantillonnées selon la distribution d'échantillonnage par importance $p_j = L_j / \sum_{k \in [\![d]\!]} L_k$, on a*

$$\mathbb{E}\big[\mathcal{L}(\theta^{(T)})\big] - \mathcal{L}^\star \leqslant \big(\mathcal{L}(\theta^{(0)}) - \mathcal{L}^\star\big)\Big(1 - \frac{\lambda}{\sum_{j \in [\![d]\!]} L_j}\Big)^T + \frac{1}{2\lambda}\big\|\epsilon(\delta)\big\|_2^2,$$

avec probabilité au moins $1-\delta$, où l'espérance est par rapport à l'échantillonnage des coordonnées j_t.

La valeur de $\|\epsilon(\delta)\|_2^2$ dans le Théorème 1.4 dépend de l'estimateur de gradient utilisé dans l'Itération (1.2.5). En utilisant des estimateurs de moyenne scalaire robustes, on obtient des

procédures robustes avec un surplus minime de calcul par rapport à la descente de gradient non robuste utilisant le risque empirique (1.2.3). Nous obtenons également des résultats semblables au Théorème 1.4 pour différentes stratégies d'échantillonnage des coordonnées pour CGD. Pour un objectif gradient-Lipschitz mais non fortement convexe, nous montrons que l'optimisation converge malgré les erreurs potentielles en troncant les gradients estimés dans les limites des intervalles de confiances qu'ils satisfont. Nous considérons trois estimateurs : l'estimateur de

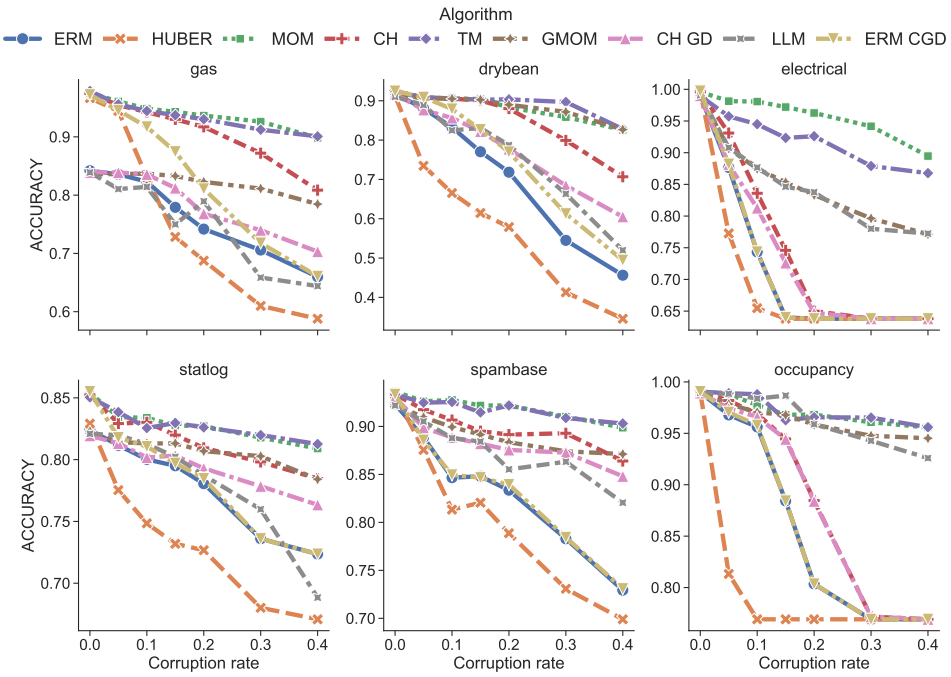

FIGURE 1.2 : Précision sur jeu de test (axe des y) contre la proportion d'échantillons corrompus (axe des x) pour six jeux de données et différents algorithmes. `MOM`, `CH` et `TM` représentent CGD utilisant MOM, l'estimateur de Catoni et la moyenne tronquée respectivement. `ERM` et `ERM CGD` représentent la descente de gradient (GD) et CGD en utilisant la moyenne empirique. `HUBER` utilise GD et la perte de Huber modifiée [452] pour la classification. Les algorithmes restants sont tirés de [262, 364, 191].

Catoni, la médiane des moyennes et le moyenne tronquée que nous avons définis précédemment. Ceux ci permettent de gérer les gradient corrompus et à queue lourde, y compris le cas de variance infinie. Nous écrivons une implémentation efficace des algorithmes d'apprentissage associés et fournissons le code en accès publique à travers une librairie Python appelée `linlearn`[1].

En utilisant notre implementation, nous faisons une comparaison expérimentale extensive des méthodes robustes à base de CGD avec les autres méthodes d'apprentissage robustes dans la littérature [262, 364, 191]. La figure 1.2 est tirées du Chapitre 3 et montre la comparaison en termes de précision sur plusieurs jeux de données de classification pour des taux de corruption croissants des données. Les résultats montrent que les algorithmes basés sur CGD (en particulier `MOM` et `TM`) sont robustes même à de hauts niveaux de corruption et conservent de bonnes performances même dans ce cas.

[1] `https://github.com/linlearn/linlearn`

1.2.2 Le cas de la grande dimension

L'apprentissage automatique avec modèle linéaire est parfois confronté à des situations où les données sont représentées dans un espace de grande dimension $X \in \mathcal{X} = \mathbb{R}^d$ avec $d \gg 1$. À part les soucis computationnels, la situation où la dimension dépasse le nombre d'échantillons ($d > n$) provoque la surdétermination et nécessite une modification du modèle. Pour résoudre ce problème, l'usage est d'ajouté une hypothèse d'*éparsité* [66, 61, 425] qui postule que le label Y peut être prédit en utilisant seulement un sous-ensemble de taille $s < d$ des features disponibles. Ceci est équivalent à supposer que l'optimum $\theta^\star = \arg\min_\theta \mathcal{L}(\theta)$ soit s-épars i.e. toutes sauf s de ses coordonnées sont égales à zéro.

Puisque ce sous-ensemble de coordonnées non nulles n'est pas connu a priori, l'optimisation de l'objectif (1.2.2) doit quand même se faire dans l'espace à grande dimension où la descente de gradient conventionnelle serait une solution inefficace. Des méthodes plus appropriées dans ce contexte sont offerte par des schémas d'optimisation *non-Euclidiens* tels que la descente miroir [339] et l'algorithme du *dual averaging* [343]. En effet, pour une fonction f et un pas de descente β, la descente de gradient classique effectue l'itération

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

Cette dernière considère implicitement la métrique Euclidienne sur l'espace des paramètres de sorte qu'il soit isométrique avec son *dual* qui est l'espace des gradients. En effet, l'itération ci-dessus est équivalente à

$$\theta_{t+1} = \arg\min_\theta \Big\{ \langle \theta, \beta \nabla f(\theta_t) \rangle + \frac{1}{2} \|\theta - \theta_t\|^2 \Big\},$$

qui utilise la norme Euclidienne sur l'espace des paramètres. La descente miroir peut être vue comme une généralisation de la descente de gradient qui remplace la métrique Euclidienne par une *divergence de Bregman* [339] $V(\theta, \theta')$ ce qui se traduit par l'itération

$$\theta_{t+1} = \arg\min_\theta \Big\{ \langle \theta, \beta \nabla f(\theta_t) \rangle + V(\theta, \theta_t) \Big\}.$$

De cette manière, selon le choix de V, une métrique différente est induite sur l'espace des gradients par dualité. En particulier, un choix judicieux permet de rendre l'optimisation plus efficace dans le cas de la grande dimension.

1.2.3 Contribution : Apprentissage robuste en grande dimension

Au Chapitre 4, nous traitons des problèmes d'apprentissage avec modèle linéaire et fonction de perte gradient-Lipschitz dans le cadre de grande dimension en utilisant la descente miroir. Nous nous inspirons de [223] pour définir un algorithme d'apprentissage multi-étapes robuste basé sur la descente miroir que nous appelons AMMD (*Approximate Multistage Mirror Descent* ou descente miroir approchée multi-étapes). Ce dernier peut être utilisé pour différentes tâches d'apprentissage y compris l'estimation éparse de base, par groupes ou l'estimation de matrice de bas rang en insérant un estimateur de gradient approprié dans chacun de ces cas.

Selon le problème, différentes divergences de Bregman sont utilisées pour la descente miroir induisant différentes métriques sur l'espace des gradients. Pour l'estimation éparse de base en particulier, l'espace des paramètres est muni de la métrique ℓ_1 qui induit la métrique ℓ_∞ sur l'espace des gradients. Dans ce cas, le gradient peut être estimé avec taux statistique quasi-optimal et avec surplus de calculs minime en utilisant simplement une moyenne tronquée coordonnée par coordonnée. On effet, on a l'énoncé suivant

Lemme 1.1. *Soient $X_1, \dots, X_n$ des échantillons i.i.d η-corrompus d'un vecteur aléatoire $X \in \mathbb{R}^d$ tel que chaque coordonnée de X a une variance inférieur à $\overline{\sigma}^2 < +\infty$. Soient $\delta > 0$ et $\widetilde{\mu}^{\epsilon}_{\mathrm{TM}}$ l'estimateur par moyenne tronquée cordonnée par coordonnée de $\mu = \mathbb{E}[X]$ calculer avec $(X_i)_{i=1}^n$ avec paramètre $\epsilon = 8\eta + 24\frac{\log(4/\delta)}{n}$, on a*

$$\mathbb{P}\left(\left\| \widetilde{\mu}^{\epsilon}_{\mathrm{TM}} - \mu \right\|_{\infty} > 12\overline{\sigma}\sqrt{2\eta} + 2\overline{\sigma}\sqrt{\frac{\log(d) + \log(4/\delta)}{n}} \right) \leqslant \delta.$$

Le Lemme 1.1 est une conséquence de (1.1.4) et d'un argument *Union Bound*. Ceci conduit à un estimateur $\widehat{\theta}$ robuste aux queues lourdes et à la corruption satisfaisant la borne d'estimation suivante (voir Chapitre 4 Corollaire 4.1 pour plus de details)

$$\mathbb{P}\left(\left\| \widehat{\theta} - \theta^{\star} \right\|_2 > \frac{2^{-K/2}R}{\sqrt{\bar{s}}} + \frac{140\sqrt{2\bar{s}}\sigma_{\max}}{\kappa}\sqrt{4\eta + 6\frac{\log(4/\widetilde{\delta}) + \log(d)}{\widetilde{n}}} \right) \leqslant \delta,$$

où $\widetilde{\delta} = \delta/T$ et $\widetilde{n} = n/T$ avec T le nombre d'itérations, $\bar{s}$ est une borne supérieure sur l'éparsité s, K est le nombre d'étapes exécutées par l'algorithme et $\sigma_{\max}$ est une borne uniforme sur la variance des coordonnées du gradient.

Nous étendons également l'analyse aux objectifs non gradient-Lipschitz en définissant un algorithme similaire à AMMD basé sur l'algorithme de *dual averaging* [343]. De plus, nous proposons des estimateurs robustes atteignant des taux statistiques presque optimaux pour traiter les problèmes d'estimation de vecteur épars par groupes et de matrice de bas rang selon les métriques qui apparaîssent dans chaque cas. Nous implémentons ces algorithmes dans la librairie `linlearn` et appuyons les résultats théoriques par des expériences numériques sur de vrais jeux de données qui confirment l'efficacité des algorithmes proposés comparés à ceux déjà connus.

1.3 Robustesse et propriétés de concentration de SGD

Après avoir traité de méthodes d'apprentissages robuste dans le cas batch, nous nous tournons maintenant vers le même problème dans le cadre de flux de données en considérant le problème d'optimisation stochastique (1.1.7). Parmi les références les plus pertinentes pour ce problème, on trouve les articles de [166, 417] qui traitent chacun du problème (1.1.7) avec échantillons de gradient à queue lourde en utilisant SGD tronqué (1.1.12) et obtiennent des bornes de déviation sous-Gaussiennes sur le défaut d'optimalité final. Ces travaux utilisent le cadre des martingales pour étudier l'erreur d'estimation le long de l'itération et appliquent l'inégalité de Freedman [150] (aussi appelée inégalité de Bernstein pour les martingales) pour obtenir leur borne de concentration finale.

Bien que SGD avec gradients à queue lourde ait reçu pas mal d'attention [166, 417, 389, 346], une solution pouvant supporter des données corrompus manquait encore. Nous considérons ce problème au Chapitre 5 sous un modèle de corruption semblable à celui de Huber et dans lequel, à chaque itération, l'échantillon de gradient reçu est corrompu avec probabilité $\eta < 1/2$ indépendamment des autres itérations. Une simple intuition suggère que, si l'on venait à exécuter SGD en plafonnant la contribution de chaque échantillon à un seuil suffisamment bas, alors l'optimisation ne serait pas semée par la corruption puisque cette dernière ne constitue qu'une minorité des échantillons ($\eta < 1/2$). En d'autres termes, une version tronquée de SGD est robuste à notre modèle d'η-corruption.

Afin d'éviter de ralentir excessivement l'optimisation, le seuil de troncage doit être fixé de manière adaptative et robuste. Un choix satisfaisant ces critères est fourni, par exemple, par la norme médiane des échantillons de gradient qui peut être estimée en pratique en gardant un

historique des normes de gradient. Afin d'analyser cette procédure, nous remplaçons le formalisme de martingale utilisé par [166, 417] par un formalisme de chaîne de Markov qui permet de contourner la tâche combinatoire d'énumérer les suites possibles d'échantillons corrompus et leurs probabilités.

1.3.1 Contribution : SGD robuste aux queues lourdes et au données corrompues

Nous faisons l'analyse théorique de l'algorithme informellement décrit ci-dessus au Chapitre 5. Pour ce faire, nous utilisons des outils de la théorie de chaîne de Markov à espaces généraux [313] pour établir la convergence en loi et caractériser la distribution limite de l'itération. Comme le montre l'analyse, l'itération est idéalement définie en utilisant un quantile de la norme du gradient comme seuil de troncage. En notant $G(\theta, \zeta)$ un échantillon de gradient aléatoire suivant la distribution corrompue et $\widetilde{G}(\theta, \zeta)$ un échantillon de la vraie distribution (i.e. tel que $\mathbb{E}_\zeta[\widetilde{G}(\theta, \zeta)] = \nabla\mathcal{L}(\theta)$), on obtient l'itération suivante que nous appelons QC-SGD (*quantile clipped SGD*)

$$\theta_{t+1} = \theta_t - \alpha_{\theta_t}\beta G(\theta_t, \zeta_t) \quad \text{with} \quad \alpha_{\theta_t} = \min\left(1, \frac{Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)}{\|G(\theta_t, \zeta_t)\|}\right), \tag{1.3.1}$$

où $\beta > 0$ est un pas de descent constant et α_{θ_t} est un facteur de troncage[2] avec seuil défini par $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ le quantile d'ordre p de $\|\widetilde{G}(\theta_t, \zeta_t)\|$ avec $p \in]0, 1[$.

Nous montrons que l'itération ci-dessus, vue comme une chaîne de Markov, converge vers une distribution invariante qui est concentrée autour de l'optimum $\theta^\star = \arg\min \mathcal{L}(\theta)$ lorsque ce dernier existe. Ceci est le sujet de l'énoncé suivante tiré du Chapitre 5

Proposition 1.1 (Proposition 5.1 informelle)**.** *Supposons que $\mathcal{L}$ soit gradient-Lipschitz et fortement convexe, que les échantillons de gradient $G(\theta, \zeta)$ soient η-corrompus et que les erreurs de gradient satisfassent*

$$\mathbb{E}\big[\|\varepsilon_\zeta(\theta)\|^q|\theta\big]^{1/q} = \mathbb{E}\big[\|\widetilde{G}(\theta, \zeta) - \nabla\mathcal{L}(\theta)\|^q|\theta\big]^{1/q} \leqslant A_q\|\theta - \theta^\star\| + B_q,$$

pour $q > 1$. Alors, pour un pas β et un ordre de quantile p bien choisis, la chaîne de Markov QC-SGD (1.3.1) *converge géométriquement en distance variation totale vers une distribution invariante $\pi_{\beta,p}$ telle que*

$$\mathbb{E}_{\theta\sim\pi_{\beta,p}}\big[\|\theta - \theta^\star\|^2\big] \leqslant 20\Big(\frac{\eta^{1-1/q}B_q}{\kappa}\Big)^2. \tag{1.3.2}$$

Thus, QC-SGD is robust both to corrupted and heavy-tailed gradients, including the infinite variance case ($q \in (1, 2)$). Moreover, by restricting Iteration (1.3.1) to a bounded set (via projection), we show that the limit distribution $\pi_{\beta,p}$ is sub-Gaussian if $\eta = 0$ (no corruption) and sub-exponential otherwise. Indeed, we argue that $\pi_{\beta,p}$ cannot be sub-Gaussian when corruption is present.

Ainsi, QC-SGD est robuste à la fois aux gradients corrompus et à queue lourde, y compris le cas de variance infinie ($q \in]1, 2[$). De plus, en restreignant l'Itération (1.3.1) à un ensemble borné (par projection), nous montrons que la distribution limite $\pi_{\beta,p}$ est sous-Gaussienne si $\eta = 0$ (pas de corruption) et sous-exponentielle sinon. En effet, nous soutenons que $\pi_{\beta,p}$ ne peut être sous-Gaussienne en présence de corruption.

Nous montrons également un résultat de convergence similaire pour les fonctions objectif $\mathcal{L}$ gradient-Lipschitz et positives. Cette convergence se produit à une vitesse sous-linéaire (en $1/t$)

[2]Cette nouvelle notation pour le troncage sera plus pratique dans la suite.

et vers une distribution limite satisfaisant la propriété suivante pour $\theta \sim \pi_{\beta,p}$ (voir Chapitre 5 Proposition 5.3)

$$\mathbb{E}_{\theta\sim\pi_{\beta,p}}\left[\left\|\nabla\mathcal{L}(\theta)\right\|^2\right] \leqslant O\left(\eta^{2-\frac{2}{q}}B_q^2\right). \tag{1.3.3}$$

La distribution limite est ainsi concentrée autour d'un point critique. Il est à remarquer que la convexité de $\mathcal{L}$ n'est pas requise pour obtenir résultat.

Les énoncés de convergence en variance totale de chaîne de Markov du Chapitre 5 (et du Chapitre 6) sont basés sur des résultats d'ergodicité de [313]. Afin de prouver l'ergodicité géométrique (i.e. convergence à vitesse géométrique), on prouve, pour des objectifs $\mathcal{L}$ fortement convexes, que la chaîne de Markov satisfait la condition de *dérive géométrique* suivante (*geometric drift* en anglais, voir [313, Chapitre 15]) par rapport à la fonction $V(\theta) = 1 + \|\theta - \theta^\star\|^2$:

$$\mathbb{E}[V(\theta_{t+1})|\theta_t] \leqslant \kappa V(\theta_t) + b \cdot \mathbf{1}_{\theta_t\in\mathcal{C}},$$

où $\kappa < 1$ est un facteur contractant, $b < +\infty$ et $\mathcal{C}$ est un voisinage borné de $\theta^\star$. En d'autres termes, la distance à l'optimum est approximativement contractée d'un facteur κ après chaque itération de (1.3.1) en moyenne pour $\theta_t \notin \mathcal{C}$. Pour les objectifs gradient Lipschitz, l'ergodicité avec convergence à vitesse sous-linéaire est établie en montrant la propriété de dérive plus faible suivante

$$\mathbb{E}[V(\theta_{t+1})|\theta_t] - V(\theta_t) \leqslant -1 + b \cdot \mathbf{1}_{\theta_t\in\mathcal{C}},$$

avec V égale à l'objectif $\mathcal{L}$ multiplié par un facteur d'échelle. À savoir, $\mathcal{L}$ décroît d'une quantité constante après chaque itération en moyenne en dehors de $\mathcal{C}$.

Un élément crucial pour prouver les propriétés de concentration (1.3.2) et (1.3.3) de la distribution limite invariante au Chapitre 5 est la définition suivante du biais du gradient dans l'Itération (1.3.1) :

$$\alpha_\theta G(\theta,\zeta) - \overline{\alpha}_\theta \nabla\mathcal{L}(\theta),$$

où $\overline{\alpha}_\theta = \mathbb{E}[\alpha_\theta \mid \theta]$ est la valeur moyenne du facteur de troncage (voir Section 5.9 Lemme 5.2 pour une définition plus precise). Ceci diffère de la définition habituelle prenant $\nabla\mathcal{L}(\theta)$ comme référence plutôt que $\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)$ et utilisée dans [166, 417]. La distinction vient du fait que [166, 417] supposent des échantillons de gradient non biaisés et utilisent des seuils de troncage bien au dessus des valeurs attendues de sorte qu'ils sont rarement excédés (voir discussion suivant le Théorème 1.3 ci-dessus). Dans ce cas, le troncage sert principalement à mitiger l'effet d'une distribution à queue lourde. En revanche, notre stratégie de troncage prévoit, en plus, de filtrer la corruption ce qui requiert un seuil inférieur et, par conséquent, une redéfinition du biais pour l'analyse.

Une implémentation directe de l'Itération (1.3.1) n'est pas possible vu que les quantiles $Q_p(\|\widetilde{G}(\theta_t,\zeta_t)\|)$ ne sont généralement pas connus. Néanmoins, nous implémentons une variante appelée "rolling QC-SGD" (RQC-SGD) qui remplace $Q_p(\|\widetilde{G}(\theta_t,\zeta_t)\|)$ par le quantile d'ordre p d'un historique $(\|G(\theta_{t-s},\zeta_{t-s})\|)_{s=0}^{S-1}$ de taille $S \in \mathbb{N}^*$.

Nous effectuons des expériences numériques qui montrent les propriétés de robustesse de RQC-SGD suggérées par la théorie. Les expériences furent exécutées sur des données synthétiques pour l'estimation de moyenne vectorielle, la régression linéaire et la régression logistique.

La Figure 1.3 est tirée du Chapitre 5 et compare RQC-SGD à l'estimateur de Huber [204] et à SGD tronqué avec des seuils constants pour la régression linéaire et logistique avec données corrompues. Tandis que l'estimateur de Huber n'est pas robuste à la corruption, c'est le cas pour SGD tronqué avec seuil constant. Cependant, ce dernier manque des propriétés d'adaptivité de RQC-SGD et peut ne converger que lentement et aboutir à des estimations finales imprécises.

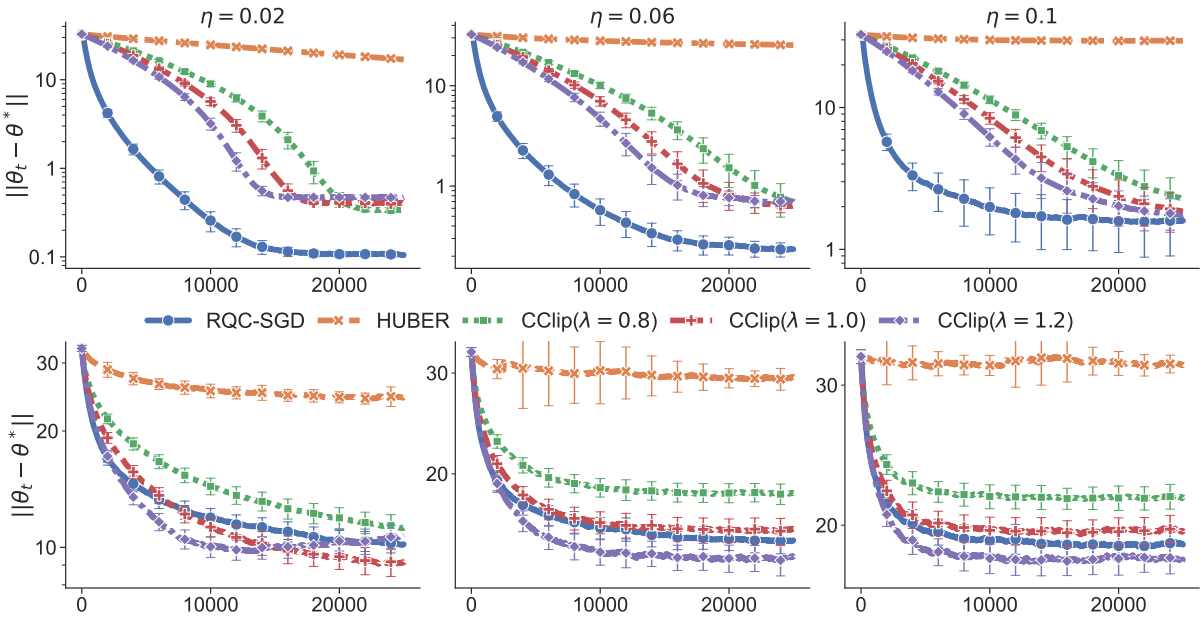

FIGURE 1.3 : Évolution de $\|\theta_t - \theta^\star\|$ sur les tâches de régression linéaire (ligne au dessus) et de régression logistique (ligne en bas) moyennée sur 100 exécutions à niveaux de corruption croissants (les barres d'erreur représentent la moitié des écart-types). Les estimateurs basés sur la fonction de perte de Huber sont fortement affectés par la corruption des données. SGD avec troncage à seuil constant est robuste mais converge lentement pour la régression linéaire et requiert un ajustement du seuil pour améliorer la précision finale. RQC-SGD réunit une convergence rapide avec une bonne précision finale grâce à la stratégie de troncage adaptative.

1.3.2 Contribution : Propriétés de convergence et de concentration de SGD

Les outils de chaînes de Markov utilisés au Chapitre 5 nous permettent également d'étudier les propriété de l'algorithme SGD classique avec gradients sans biais en dehors du cadre robuste. Nous montrons au Chapitre 6 que l'itération de SGD standard (1.1.8) avec pas constant converge vers une distribution limite invariante qui hérité des propriétés de concentration sous-Gaussienne et sous-exponentielle lorsque ces dernières sont satisfaites par la distribution du gradient i.e. nous avons l'énoncé suivant.

Proposition 1.2 (Proposition 6.2 informelle)**.** *Supposons que $\mathcal{L}$ soit μ-fortement convexe et que les erreurs de gradient $G(\theta,\zeta) - \nabla\mathcal{L}(\theta)$ soient sous-Gaussiennes (resp. sous-exponentielles) avec constante K pour tout θ. Alors l'itération SGD standard* (1.1.8) *exécutée avec pas de descente constant $\beta_t = \beta$ suffisamment petit converge vers une distribution limite invariante π_β sous-Gaussienne (resp. sous-exponentielle) avec constante $O\big(K\sqrt{\beta/\mu}\big)$.*

L'aspect important de l'énoncé précédent est que la constante sous-Gaussienne (resp sous-expoentielle) de la distribution invariante soit modulée par la valeur de β ce qui permet d'obtenir des bornes de déviation en grande probabilité sur un estimateur final $\widehat{\theta}_T$ après T itérations avec un petit pas de descente. Sous des hypothèses légèrement plus fortes sur les propriétés de concentration des échantillons de gradient, nous montrons que les conclusions de la Proposition 1.2 sont vérifiées avec une constante K indépendante de la dimension, ce qui permet d'obtenir des bornes de concentration sous-Gaussiennes sur $\widehat{\theta}_T$.

Nous obtenons également des bornes en grand probabilité pour les estimateurs $\widehat{\theta}$ définis comme une moyenne d'itérés finaux de SGD (moyennes de Polyak-Ruppert [358, 386]). Nous avons le résultat non asymptotique suivant.

Proposition 1.3 (Proposition 6.6 informelle)**.** *Supposons que $\mathcal{L}$ soit gradient-Lipschitz et μ-fortement convexe et que le gradient $\nabla\mathcal{L}(\theta)$ soit linéaire. Supposons aussi que les erreurs de gradient $\varepsilon_\zeta(\theta) = G(\theta,\zeta) - \nabla\mathcal{L}(\theta)$ soient sous-Gaussiennes avec constante K et satisfassent pout tous θ, θ' :*

$$\mathcal{W}_2^2\big(\mathcal{D}(\varepsilon_\zeta(\theta)), \mathcal{D}(\varepsilon_\zeta(\theta'))\big) \leqslant L_{\mathcal{W}}\|\theta - \theta'\|_2^2 \tag{1.3.4}$$

avec $L_{\mathcal{W}} < +\infty$, où $\mathcal{D}(\varepsilon_\zeta(\theta))$ est la distribution de $\varepsilon_\zeta(\theta)$ avec $\mathcal{W}_2$ la distance de Wasserstein-2.

Soit $(\theta_t)_{t\geqslant 0}$ la chaîne de Markov de SGD lancé à partir de $\theta_0 \sim \nu$ avec pas de descente β suffisamment petit. Alors, il existe $\rho < 1$ et $M < +\infty$ tels que pour $\delta > 0$ et $n, n_0 > 0$:

$$\Big\|\frac{1}{n}\sum_{t=n_0+1}^{n_0+n}\theta_t - \theta^\star\Big\| \leqslant \sqrt{\frac{2}{n}\frac{1+\alpha}{1-\alpha}\Big(\alpha_{\mathcal{W}}^{n_0}\mathcal{W}_2^2(\nu,\pi_\beta) + \frac{\beta\sigma^2}{\mu}\Big)} + \frac{4K\sqrt{\beta/\mu}}{1-\alpha_{\mathcal{W}}}\sqrt{\frac{\log(1/\delta)}{n}} \tag{1.3.5}$$

avec probabilité au moins $1 - \Upsilon(\nu, n_0)\delta$, où

$$\alpha = 1 - \beta\mu, \quad \alpha_{\mathcal{W}} = \sqrt{\alpha^2 + \beta^2 L_{\mathcal{W}}} \quad \textit{et} \quad \Upsilon(\nu, n_0) = 1 + M\rho^{n_0}\Big\|\frac{d\nu}{d\pi_\beta}\Big\|_\infty.$$

La preuve de l'énoncé ci-dessus combine plusieurs propriétés d'intérêt potentiel indépendant telles que la convergence en distance en variation totale et en distance de Wasserstein et un phénomène de décorrélation géométrique entre les itérés successifs θ_t lorsque le gradient $\nabla\mathcal{L}$ est linéaire. En particulier, nous obtenons la convergence par rapport à la métrique de Wasserstein grâce à l'hypothèse de l'Inégalité (1.3.4) qui une condition plus générale que celle précédemment connue dans la littérature comme il est démontré en Section 6.5. Enfin, nous discutons l'application de ces résultats aux exemple de régression linéaire et logistique.

1.4 Régression logistique en ligne avec regret optimal

Dans cette section, on considère le problème de régression logistique en ligne qui consiste à prédire une séquence de labels binaires $y_t \in \mathcal{Y} = \{\pm 1\}$ à partir d'entrée Euclidiennes $x_t \in \mathcal{X} = \mathbb{R}^d$ pour $t \geqslant 1$ (on se concentre sur le cas de labels binaires mais l'extension au cas multi-classe est possible).

Dans le cadre en ligne, les instances $(x_t, y_t)_{t\geqslant 1}$ arrivent séquentiellement et l'*agent* fait une prédiction $\widehat{y}_t$ à chaque tout qui est évaluée par la perte logistique

$$\ell(\widehat{y}_t, y_t) = \log(1 + \exp(-\widehat{y}_t y_t)). \tag{1.4.1}$$

Plus précisément, les deux étapes suivantes se produisent pour chaque $t \geqslant 1$:

- L'agent reçoit un échantillon x_t de la part de l'environnement et l'utilise avec l'historique jusqu'à lors $H_t = \{(x_1, y_1), \dots, (x_{t-1}, y_{t-1})\}$ afin de faire une prédiction $\widehat{y}_t$ de y_t.
- Ensuite, l'environnement révèle la valeur du label y_t et l'agent subit la perte logistique $\ell(\widehat{y}_t, y_t)$.

La performance globale des prédictions de l'agent est évaluée à travers la notion de regret qui est conventionnellement définie par rapport au meilleur prédicteur linéaire a postériori dans une classe de comparaison $\Theta \subset \mathbb{R}^d$. Ceci correspond à la définition suivante après n tours

$$R_n := \sum_{t=1}^n \ell(\widehat{y}_t, y_t) - \min_{\theta\in\Theta}\sum_{t=1}^n \ell(\theta^\top x_t, y_t). \tag{1.4.2}$$

Le cadre classique pour la minimisation du regret considère des données bornées $\|x_t\| \leqslant R$ et fixe la classe de comparaison comme la boule Euclidienne de rayon B i.e. $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\| \leqslant B\}$.

Attention à ne pas confondre cette tâche de régression logistique en ligne avec les problèmes d'apprentissage robustes abordés précédemment puisqu'ils diffèrent sur plusieurs aspects :

- La suite (x_t, y_t) est purement arbitraire dans ce cas et ne suit aucune distribution de probabilité.
- Les x_t ne peuvent pas être à queue lourde puisqu'ils sont supposés bornés.
- Malgré que la suite (x_t, y_t) pourrait être constituée par un adversaire, aucun échantillon ne peut être considéré comme faux ou corrompu puisqu'ils comptent tous dans le regret (1.4.2).

La qualité d'une stratégie d'apprentissage pour la minimisation du regret est mesurée par le taux asymptotique qu'elle atteint en termes du nombre de tours n. Il est important de concevoir des stratégies avec regret sous-linéaire $R_n = o(n)$ puisque, dans le cas contraire, un regret linéaire signifie que l'agent ne parvient pas à apprendre la tâche en subissant une perte d'ordre au moins constant par tour en moyenne.

De multiples stratégies de minimisation du regret furent proposées dans la littérature avec des bornes de regret progressivement améliorées en tirant parti des propriétés de la perte logistique (1.4.1). Par exemple, la descente de gradient en ligne (en anglais *Online Gradient Descent* ou OGD) [460] obtient un regret en $O(BR\sqrt{n})$ grâce à la convexité de ℓ et la descente de Newton enligne (en anglais *Online Newton Step* ou ONS) [184] obtient la borne $O(de^{BR}\log(n))$ en tirant avantage de la forte convexité.

Un progrès important fût accompli par [149] qui a montré qu'un regret logarithmic est possible sans le facteur exponentiel e^{BR} apparaîssant dans la borne satisfaite par ONS. Ceci est faisable grâce à un algorithme *impropre* qui fait des prédictions $\widehat{y}_t = \theta_t^\top x_t$ où le paramètre linéaire θ_t est calculé en utilisant la connaissance de x_t en plus des échantillons de l'historique $(x_s, y_s)_{s=1}^{t-1}$. Grâce à une approche de prédiction Bayésienne, [149] obtient un algorithme avec borne de regret en $O(d\log(BRn))$ qui reste inégalé au moment de cette écriture. Cependant, cet algorithme ne peut être utilisé en pratique vu sa grande complexité de l'ordre $O((BR)^6 n^{12}(BRn+d)^{12})$.

Des travaux ultérieurs [218, 4] on conçu des algorithmes impropres plus pratiques qui atteignent un regret logarithmique avec un faible coût en calculs de l'ordre de $O(nd^2)$. Toutefois, tous ces algorithmes ont un regret de l'ordre de $O(dBR\log(n))$ qui est sous-optimal d'un facteur BR par rapport à [149].

Dans l'Appendice A, nous discutons d'un algorithme efficace qui pourrait fournir une solution comblant ce fossé. Ce dernier se base sur le *Sample Minmax Predictor* (SMP) qui fût introduit par [325] et pour lequel la borne suivante fût prouvée sur l'excès de risque pour la régression logistique batch à n échantillons (voir définition (1.2.2) en Section 2.2)

$$\mathcal{L}(\widehat{\theta}_{\text{SMP}}) - \min_{\theta\in\Theta} \mathcal{L}(\theta) \leqslant \frac{ed + B^2R^2}{n}. \tag{1.4.3}$$

SMP est un estimateur de non Bayésien qui peut être calculé en résolvant deux problème de régression logistique batch. Ce qui le rend bien plus efficace que les algorithmes Bayésiens nécessitant l'usage de méthodes d'intégration MCMC coûteuses comme pour [149]. Vue la borne (1.4.3) sur l'excès de risque batch, une version en ligne de SMP est susceptible d'atteindre un regret en $O((d + B^2R^2)\log(n))$ qui serait équivalente à la performance de [149] puisqu'on peut soutenir que $BR \lesssim \sqrt{d}$ (voir [326, Remarque 2]).

Une variante en ligne de SMP est introduite en Appendice A en reprenant les idées de [326] dans le cadre en ligne. De plus, nous présentons une analyse préliminaire du regret basée sur la forte convexité de la perte logistique afin de montrer un regret logarithmique. Vu qu'on ne sait

pas encore comment la borne cible en $O((d + B^2R^2)\log(n))$ peut être prouvée, nous discutons de la difficulté sous-jacente dans l'analyse et des moyens possibles pour la dépasser.

1.5 De l'arbre aléatoire à WildWood

Comme mentionné précédemment, la plupart des problèmes d'apprentissage automatique consistent à choisir un bon prédicteur $\phi : \mathcal{X} \to \widehat{\mathcal{Y}}$ parmi une classe de fonctions candidates Φ. Nous avons vu que les modèles linéaires correspondent à utiliser comme Φ l'ensemble des fonctions linéaires. Dans cette partie, on considère les forêts aléatoires qui correspondent à définir Φ comme l'ensemble des fonctions constantes par morceaux. Il est évident que cette ensemble a une forte capacité d'approximation vu qu'il permet d'approcher (par exemple) n'importe quelle fonction continue bornée à précision arbitraire en utilisant une partition fine de l'espace $\mathcal{X}$.

En effet, une fonction constante par morceaux ϕ peut être définie par une partition de l'espace en régions disjointes $\mathcal{X} = \cup_j C_j$ et par les valeurs $v_j \in \mathcal{Y}$ prises par ϕ sur chacune de ces régions de sorte qu'on ait $\phi(x) = \sum_j \mathbf{1}_{x \in C_j} v_j$. Contrairement aux fonctions linéaires, l'ensemble des fonctions constantes par morceaux a un caractère non paramétrique qui exclu les méthodes de gradient pour leur entraînement. Au lieu de cela, les modèles constants par morceaux sont entraînés grâce une méthode récursive de *croissance d'arbre* qu'on décrit ci-dessous.

L'algorithme de croissance d'arbre

On considère un espace Euclidien $\mathcal{X} = \mathbb{R}^d$. En partant d'une partition triviale $\mathcal{P}_0 = \{C_{\texttt{root}}\} = \{\mathbb{R}^d\}$ à $t = 0$, on exécute les étapes suivantes :

1. Choisir une cellule $C_{\mathbf{v}} \in \mathcal{P}_t$ parmi la partition actuelle.
2. Choisir une coordonnées (ou *feature*) $j \in \{1, \dots, d\}$ et un seuil $s \in \mathbb{R}$.
3. Définir les sous-cellules $C_{\mathbf{v}} = C_{\mathbf{v}0} \cup C_{\mathbf{v}1}$ où $C_{\mathbf{v}0} = \{x \in C_{\mathbf{v}} : x_j \leqslant s\}$ et $C_{\mathbf{v}1} = C_{\mathbf{v}} \setminus C_{\mathbf{v}0}$.
4. Mettre à jour la partition comme $\mathcal{P}_{t+1} = (\mathcal{P}_t \setminus C_{\mathbf{v}}) \cup \{C_{\mathbf{v}0}, C_{\mathbf{v}1}\}$.

Après une itération, on peut définir le modèle de prédiction en choisissant les valeurs constantes v_0, v_1 sur chaque membre de la partition $C_0 \cup C_1$. Afin d'obtenir un modèle plus précis, les étapes précédentes peuvent être répétées récursivement pour affiner la partition $\mathcal{P}_t$ jusqu'à ce qu'un niveau de précision satisfaisant soit atteint.

La partition de l'espace donnée par ce procédé peut être représentée par une structure hiérarchique d'arbre. La Figure 1.4 donne un exemple d'illustration des deux étapes initiales de cet algorithme pour $d = 2$. La structure d'arbre associée encode l'information sur les coupures définissant la partition dans ces nœuds intérieurs (coordonnées des coupures et leurs seuils) tandis que les feuilles représentent les régions (aussi appelées cellules) formant la partition. Idéalement, étant donné un échantillon d'entraînement $(X_i, Y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, le but serait de trouver le meilleur modèle $\phi^\star$ de ce type qui minimise une fonction objectif comme (1.2.1). Cependant, le problème de déterminer les coupures définissant la structure d'arbre d'une tel $\phi^\star$ s'avère être NP-difficile [256]. Pour éviter cette difficulté, l'algorithme de croissance d'arbre est utilisé pour entraîner un substitut du prédicteur optimal de manière gloutonne une coupure à la fois.

La croissance d'arbre comme méthode d'apprentissage

Pour obtenir un bon prédicteur pour un jeu de données $(X_i, Y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ via la croissance d'arbre, chaque coupure est faite de manière à maximiser l'homogénéité des sous-cellules en termes des labels Y_i des échantillons X_i qui s'y retrouvent. Ceci se fait en choisissant les

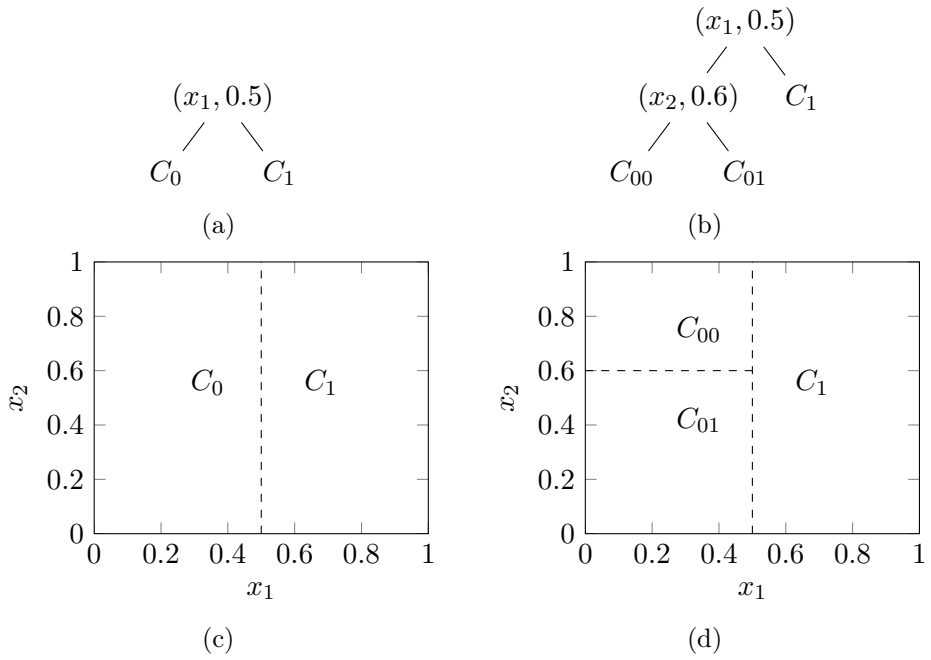

FIGURE 1.4 : Illustration de deux étapes initiales de l'algorithme de croissance d'arbre en dimension 2 où la première coupure est faite suivant la coordonnée x_1 avec seuil 0.5 donnant $\mathbb{R}^d = C_0 \cup C_1$ (à gauche). la cellule C_0 est ensuite coupée à nouveau le long de la coordonnée x_2 avec seuil 0.6 (à droite).

paramètres j et s de coupure à l'étape 2. ci-dessus en minimisant un critère heuristique d'impureté des sous-cellules. Étant donnée une fonction $\mathcal{I}$ mesurant l'impureté d'une distribution de probabilité et en notant δ_x la mesure de Dirac en x, cela correspond à l'optimisation suivante

$$\underset{s\in\mathbb{R}, j\in\{1,\dots,d\}}{\arg\min} \quad \mathcal{I}\big(\mathcal{D}_0\big) + \mathcal{I}\big(\mathcal{D}_1\big) \quad \text{où} \quad \mathcal{D}_k = \frac{1}{|\{i : X_i \in C_{\mathbf{v}k}\}|} \sum_{i:X_i\in C_{\mathbf{v}k}} \delta_{Y_i}. \tag{1.5.1}$$

Pour les problèmes de classification (les Y_i sont des labels discrets), un choix habituel de $\mathcal{I}$ et la fonction d'entropie tandis que pour les problèmes de régression (les Y_i sont des valeurs réelles) on peut utiliser la variance en guise de $\mathcal{I}$. Ainsi, l'espace $\mathcal{X}$ est subdivisé en voisinages contenant des échantillons X_i ayant des labels Y_i similaires, ce qui permet de définir un prédicteur de haute précision [55, 423].

En plus de cette heuristique, la procédure est randomisée grâce au *sous-échantillonnage de coordonnées* (en anglais *feature subsampling*) qui consiste à limiter les coordonnées candidates pour chaque coupure à un sous-ensemble aléatoire de l'ensemble total des coordonnées i.e. $\{1, \dots, d\}$ est remplacé par un sous-ensemble échantillonné aléatoirement à l'étape 2. ci-dessus et dans (1.5.1).

La structure finale est appelée *arbre binaire de décision* [370, 56] et permet d'implémenter efficacement le prédicteur associé en s'orientant dans la structure d'arbre pour une requête $X \in \mathcal{X}$. Le terme d'*arbre aléatoire* [51, 53] est aussi souvent utilisé pour faire référence à l'aléa introduit dans la croissance d'arbre à travers le sous-échantillonnage de coordonnées.

Selon la taille de l'échantillon d'entraînement $(X_i, Y_i)_{i=1}^n$, la partition de l'espace peut être raffinée en poursuivant le processus de croissance d'arbre et en coupant les feuilles de l'arbre jusqu'à ce qu'un critère d'arrêt soit satisfait. Dans le cas extrême, le processus peut se poursuivre jusqu'à ce que la partition soit si fine que chaque feuille de l'arbre ne contient plus qu'un seul échantillon X_i du jeu de données. Par contre, cela mène souvent au sur-apprentissage et il est

préférable d'éviter de faire croître l'arbre jusqu'à ce point pour avoir de meilleures performances.

Méthodes d'ensemble à base d'arbres

Les modèles de décision à base d'arbres sont apparus comme méthode d'apprentissage automatique dans les années 1960 [322, 312, 370] avant d'être popularisées par l'étude théorique et pratique approfondie de [54]. Ces modèles ont suscité l'intérêt grâce à leur simplicité, intuitivité et facilité d'usage. Cependant, leur forte capacité d'approximation et la nature bruitée des données en pratique leur donnent une tendance au sur-apprentissage qui provoque des performances prédictives médiocres. Ce problème fût traité en utilisant des *méthodes d'ensemble* qui consistent à entraîner plusieurs instances d'un modèle simple et combiner leurs prédictions dans un modèle global pour obtenir de meilleurs résultats. Deux grandes familles de méthodes d'ensemble méritent d'être mentionnées :

- Le Boosting : présenté dans [392, 152], le boosting consiste à construire un modèle d'ensemble de manière incrémentale en ajoutant plusieurs modèles simples entraînés pour corriger les erreurs de l'ensemble actuel. Le Boosting à base d'arbres utilisent des arbres peu profonds comme modèles de base et les critères de coupure appliqués sont directement liés à l'optimisation d'un objectif empirique (1.2.1). Plusieurs implémentations sont disponibles dans des librairies telles que XGBoost [87], LightGBM [231] ou CatBoost [367].

- le Bagging : ou *Bootstrap Aggregation* [51] entraîne simultanément des modèles sur des jeux de données bootstrap et les combine dans un modèle d'ensemble global qui fait ses prédictions suivant des règles de majorité à partir des prédictions des sous-modèles. Le bagging à base d'arbres utilise des critères d'impureté pour la croissance d'arbres et la profondeur est contrôlée par des heuristiques comme une profondeur limite maximale ou un nombre minimum d'échantillons par cellule pour éviter le sur-apprentissage.

Nous nous intéressons de plus près au Bagging. Les jeux de données bootstrap utilisés sont obtenus en échantillonnant n éléments (avec n le nombre total d'échantillons) du jeu de données d'entraînement avec remise pour chaque sous-modèle. Par conséquent, chaque sous-modèle est entraîné sur un sous-ensemble des échantillons d'entraînement appelé les échantillons *in-the-bag* tandis que les échantillons *out-of-bag* restants sont ignorés. Cette méthode bootstrap vise à créer une certaine indépendance entre les sous-modèles et d'éliminer le bruit dans leurs prédictions en les agrégeant avec une moyenne.

Le bagging à base d'arbre avec sous-échantillonnage de coordonnées correspond au fameux algorithme de forêt aléatoire [188, 52] implémenté en Python dans la librairie `scikit-learn` [355]. Au Chapitre 7, nous introduisons un nouvel algorithme appelé WildWood qui améliore l'algorithme de forêt aléatoire classique en ajoutant un mécanisme de régularisation par arbre efficace.

1.5.1 Contribution : WildWood : un nouvel algorithme de forêt aléatoire

Soit $\mathcal{T}$ un arbre binaire obtenu par le procédé de croissance d'arbre précédent sur un jeu de données $(X_i, Y_i)_{i=1}^n$, notons $\phi_{\mathcal{T}} : \mathcal{X} \to \widehat{\mathcal{Y}}$ sa fonction de prédiction.

Rappelons que la performance de $\phi_{\mathcal{T}}$ est souvent affectée négativement par le fait que la croissance $\mathcal{T}$ se poursuive jusqu'à une profondeur excessive puisque, au delà d'une certain précision, la partition de l'espace associée cesse de capturer de l'information pertinente sur la distribution des données et commence à suivre du bruit non informatif. Une solution à ce problème est d'utiliser un *élagage* (ou sous-arbre) $T \subset \mathcal{T}$ de l'arbre complet $\mathcal{T}$ qui néglige une partie des coupures qui le définissent (par exemple, l'arbre à gauche sur la Figure 1.4 est un sous-arbre de l'arbre à droite). Pour un sous-arbre bien choisi, on obtient un meilleur prédicteur ϕ_T basé sur une partition plus

simple de l'espace qui minimise l'effet du bruit tout en préservant une bonne capacité de prédiction. Néanmoins, vu que le nombre sous-arbres possibles $T \subset \mathcal{T}$ est hautement combinatoire, un algorithme qui les énumère simplement afin de trouver celui qui donne le meilleure prédicteur ϕ_t serait excessivement inefficace.

Afin de contourner cette difficulté, WildWood définit le prédicteur $\widehat{\phi}_{\mathcal{T}}$ d'un arbre $\mathcal{T}$ comme la moyenne exponentiellement pondérée suivante

$$\widehat{\phi}_{\mathcal{T}}(x) = \frac{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T} \phi_T(x)}{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T}} \quad \text{avec} \quad L_T = \sum_{i \in I_{\mathtt{oob}}} \ell(\phi_T(X_i), Y_i) \quad \text{et} \quad \pi(T) = 2^{-\|T\|}, \tag{1.5.2}$$

où la somme porte sur tous les sous-arbres possibles $T \subset \mathcal{T}$, L_T est la perte associée à ϕ_T sur l'échantillon out-of-bag $I_{\mathtt{oob}}$ de $\mathcal{T}$, $\|T\|$ mesure la complexité de T et $\eta > 0$ est un paramètre de température.

La moyenne exponentielle (1.5.2) pondère les sous-arbres $T \subset \mathcal{T}$ selon leur performance prédictive sur l'échantillon out-of-bag et leur simplicité (les sous-arbres peu profonds sont favorisés). La calcul de (1.5.2) est implémenté grâce à une adaptation de l'algorithme de *Context Tree Weighting* [441, 440] qui permet d'éviter l'énumération explicite des sous-arbres $T \subset \mathcal{T}$ et de maintenir la même complexité que l'algorithme de forêt aléatoire original.

L'implémentation de WildWood (publiquement accessible sur GitHub[3]) inclut également la stratégie de coupure par histogramme utilisée par les librairies de boosting [87, 231, 367]. Cette dernière consiste à quantifier les valeurs de X_i afin d'accélérer les optimisations (1.5.1) pour la recherche de coupures ce qui permet d'entraîner le modèle plus rapidement.

Le mécanisme d'agrégation défini par (1.5.2) permet d'obtenir un modèle dont les performances se rapprochent de celles du meilleur sous-arbre $T \subset \mathcal{T}$ en termes de précision des prédictions i.e. nous avons le résultat suivant.

Théorème 1.5 (Théorème 7.2 informel)**.** *Supposons que la fonction de perte ℓ soit η-exp-concave. Alors, la fonction de prédiction $\widehat{\phi}_{\mathcal{T}}$ donnée par* (1.5.2) *satisfait l'inégalité d'oracle suivante*

$$\frac{1}{n_{\mathtt{oob}}} \sum_{i \in I_{\mathtt{oob}}} \ell(\widehat{\phi}_{\mathcal{T}}(X_i), Y_i) \leqslant \inf_{T \subset \mathcal{T}} \left\{ \frac{1}{n_{\mathtt{oob}}} \sum_{i \in I_{\mathtt{oob}}} \ell(\phi_T(X_i), Y_i) + \frac{\log 2}{\eta} \frac{\|T\|}{n_{\mathtt{oob}} + 1} \right\},$$

où $\|T\|$ est le nombre de nœuds de T moins de nombre de ses feuilles qui sont aussi feuilles de $\mathcal{T}$ et $n_{\mathtt{oob}} = |I_{\mathtt{oob}}|$.

Nous évaluons les performances de WildWood en pratique et le comparons au forêts aléatoires classique et aux algorithmes de boosting à travers des expériences numériques qui confirment ses gain prédictifs sur des jeux de données de classification publiquement accessibles. Nous montrons également que WildWood obtient de bons résultats avec des modèles plus petits et plus légers qui sont plus rapides à entraîner et donnent des frontières de décision plus simples pour les problèmes de classification ainsi qu'une meilleure interprétabilité.

Ceci est illustré par les Figures 1.5 et 1.6 qui sont tirées du Chapitre 7. La première montre que WildWood obtient des modèles plus performants avec moins d'arbres que les algorithmes de forêts aléatoires précédemment connus tandis que la deuxième illustre l'effet régularisateur du mécanisme d'agrégation sur la fonction de décision de chaque arbre sur un problème de classification jouet. L'agrégation permet d'obtenir des frontières de décision plus simples qui minimisent l'effet du bruit et évitent le sur-apprentissage.

[3] `https://github.com/pyensemble/wildwood.git`

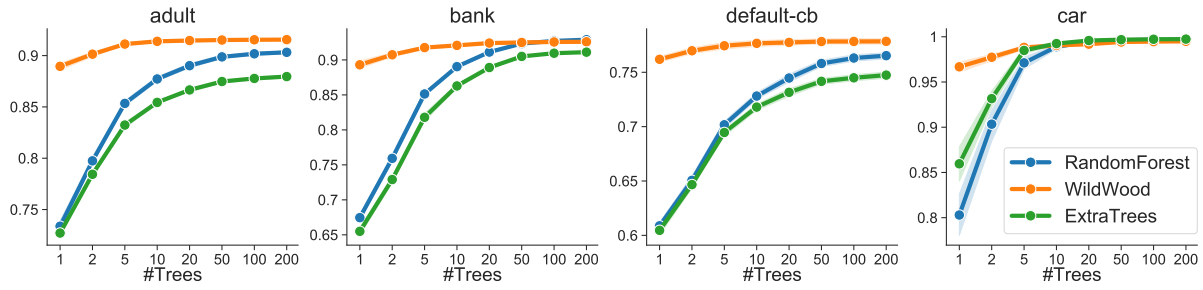

FIGURE 1.5 : Scores AUC pour la classification sur certains jeux de données et pour différents algorithmes de forêts aléatoires pour un nombre d'arbres croissant.

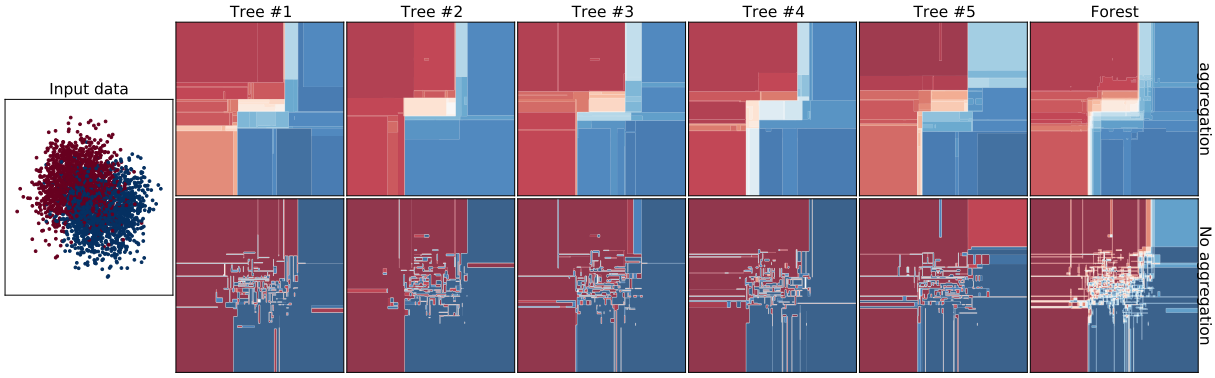

FIGURE 1.6 : WildWood decision functions illustrated on a toy dataset (left) with subtrees aggregation (top) and without it (bottom). Subtrees aggregation improves trees predictions, as illustrated by smoother decision functions in the top compared with the bottom, improving overall predictions of the forest (last column).

1.6 Liste des travaux

Nous fournissons ci-dessous une liste des articles de recherche écrits pendant cette thèse et leur statut :

- S. Gaïffas, I. Merad et Y. Yu. "WildWood : a new Random Forest algorithm". Dans : *IEEE Transactions on Information Theory* (2023).
- I. Merad et S. Gaïffas. "Robust supervised learning with coordinate gradient descent". Soumis.
- I. Merad et S. Gaïffas. "Robust methods for high-dimensional linear learning". Dans : *Journal of Machine Learning Research* (2023).
- I. Merad et S. Gaïffas. "Robust stochastic optimization via gradient quantile clipping". Soumis.
- I. Merad et S. Gaïffas. "Convergence and concentration properties of constant step-size SGD through Markov chains". Soumis.

Chapter 2

Introduction

Contents

This thesis aims to bring theoretical and methodological contributions to the field of machine learning. The latter encompasses a variety of algorithms which allow computers to learn to perform a task on data inputs through repeated exposure to examples. Appearing in the mid twentieth century, the development of machine learning methods has progressed at an unprecedented pace during the last two decades and remains a very hot topic in the present day [173, 207, 80, 329]. This fast growth was made possible by multiple factors. The most important of which are: first, the increasing computational power made available by hardware allowing to train models of macroscopic dimension [348, 407, 276]. Second, the explosion of the volumes of regularly generated data accessible on the internet and particularly on social networks providing plentiful amounts of examples machines can learn from. And third, the multiple ingenuous methodological developments which expanded the discipline to countless tasks spanning computer vision (notably using convolutional neural networks [265, 246, 186]), natural language processing (for which superior performances were achieved using Transformers and attention models [429, 108]) and robotics (through reinforcement learning methods [321, 394]), all while making the learning algorithms ever more efficient.

From a mathematical point of view, machine learning has originated from the interplay between the fields of statistics, probability and optimization. The interaction between these three subjects will be a recurring pattern along the chapters of this document and reflects the

fact that a large part of models learn by minimizing an objective function quantifying their average performance over the samples of a data set. Formally, the problem of training a machine learning model is often expressed in the following form

$$\min_{\theta} \mathcal{L}(\theta) := \mathbb{E}_{\zeta}[\ell(\theta, \zeta)], \tag{2.0.1}$$

where θ represents the parameters of the model which need to be adjusted to optimize performance. The *loss function* ℓ evaluates the accuracy of the model with parameter θ on the sample ζ. The expectation defining the objective $\mathcal{L}$ is computed with respect to the generally unknown distribution of ζ.

The goal is to design efficient solution algorithms for problems of the form (2.0.1) which use a set of samples of ζ to compute an estimator $\widehat{\theta}$ of the minimum of $\mathcal{L}$. The optimal performances of such a procedure in terms of convergence speed and statistical properties of the output estimator $\widehat{\theta}$ are determined by the characteristics of the loss function ℓ and the unknown distribution of ζ. We will elaborate more on this matter further below.

This introductory chapter gives a summary and informal description of the research constituting this thesis which is elaborated in full detail later on. A brief introduction to robust statistics is also included due to its high relevance to this work. Chapters 3 and 4 are dedicated to projects proposing robust linear learning algorithms, the second of which focuses on the high-dimensional case. Chapter 5 also deals with the robust learning problem and proposes an SGD-based algorithm destined to the setting where data is only accessible in the form of streaming samples as opposed to a full batch data set. The remaining chapters diverge from the robust setting. Chapter 6 is based on common insights with Chapter 5 and contains a study of the standard SGD algorithm leading to new results about its convergence and concentration properties. Chapter 7 presents a new random forest algorithm which improves predictive performance using fewer computational resources. Finally, in Appendix A, we discuss a new algorithm for online logistic regression which may achieve the optimal regret upper bound with much lower complexity than currently known methods reaching this performance. All chapters introduce their own formal setting and may be read independently.

2.1 Robust statistics

The central subject of this thesis revolves around robust statistics and robust learning algorithms in particular. The field of robust statistics appeared in the 1960s and was pioneered by the works of Tukey and Huber among others [422, 202, 8, 177]. The core motivation is to develop and analyze statistical estimation methods which maintain good performance even on data containing abnormal observations which differ from the majority of samples. Regarding this aspect, basic ideas can be traced back as far as the eighteenth century to the works of Boscovitch, Laplace and Edgeworth [142, 387, 140] who studied a form of robust regression using the absolute loss instead of the least squares to minimize the impact of extreme values.

More generally, the purpose is to extend the methodology and theoretical guarantees to looser data assumptions. For instance, a very common statistical setting is to consider samples $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ of independent and identically distributed (i.i.d) Gaussians. However, this set of assumptions is often questionable in practice: examples in stock price variations [296] and web data [170] highly suggest that Gaussian concentration does not hold for real world data where *heavy-tailed* distributions are often observed. In addition, abnormal values stemming from measurement errors, blunders or mistakes during data collection often find their way into a dataset and make up a 1–10% share of samples [205] invalidating the identical distribution assumption.

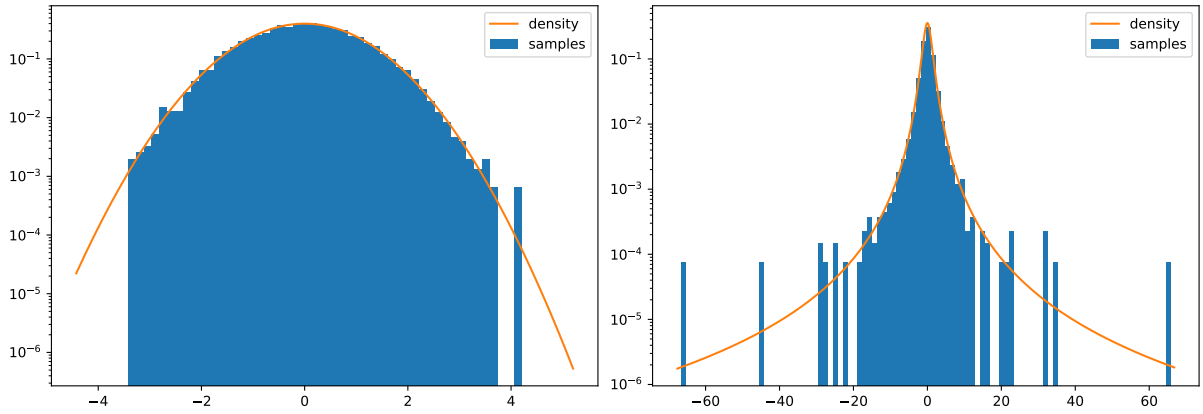

Figure 2.1: Point density function and samples from the standard Gaussian distribution (left) and the Student-t distribution with 2.2 degrees of freedom. Note that the y-axis is in log-scale. While Gaussian samples are well concentrated around their mean, samples from the Student distribution often occur several standard-deviations away producing outliers.

Considering that learning problems in the form of (2.0.1) are optimized using data samples from an unknown distribution, the precision of the obtained estimates of the optimum, and hence the performance of the trained models, strongly depend on the quality of the data used for this purpose. Early guarantees for learning algorithms [427, 157, 298] were obtained under strong assumptions such as i.i.d Gaussian or bounded data and using empirical means. Therefore, more robust methods need to be designed which can handle the unruly nature of data observed in practice. This can manifest in a few ways:

Heavy-tails. Data often appear to follow a heavy-tailed distribution. We say that a real distribution is heavy-tailed if its density function is not exponentially bounded near infinity. This is in contrast with light-tailed distributions such as sub-Gaussians or sub-exponentials whose tails decay as $\exp(-\Omega(x^2))$ or $\exp(-\Omega(|x|))$ near infinity respectively. The slow density decay of a heavy-tailed distribution makes it more likely to output extreme samples (sometimes referred to as outliers) which stray far away from the distribution's mean. This is illustrated on Figure 2.1 which displays the density and samples from the standard Gaussian distribution and a Student-t distribution which is a heavy-tailed example.

η-Contamination. This is the first model for non i.i.d data introduced by [202, 205] and generally referred to as Huber's η-contamination model. The latter posits that samples are drawn from a mixed distribution $(1-\eta)P+\eta Q$ for some contamination rate $\eta \in (0, 1/2)$ with P the true data distribution targeted by the analysis and Q an unknown contamination distribution with no assumptions. The source of spurious data is modelled as oblivious and independent which excludes adversary behavior.

Corrupted data. A more general setting is to assume that the index set is disjointly partitioned into an unknown pair $\mathcal{I} \cup \mathcal{O}$ such that samples in $\mathcal{I}$ (inliers) follow the true target distribution and those in $\mathcal{O}$ (outliers) are arbitrary corruptions. The number of corruptions is always assumed inferior to the true samples $|\mathcal{O}| < |\mathcal{I}|$ and is allowed to represent at most a fraction $\eta \in (0, 1/2)$ of the total number of samples (η-corruption). This setting was studied in [257, 261] and allows corrupted samples to depend on the true ones and even on the methods used for data processing in order to model an adversary entity.

Mean estimation is arguably one of the most fundamental tasks in statistics and may be seen as a basic building block which appears in most procedures. We propose to explore a few approaches to robustness through this problem.

2.1.1 Robust scalar mean estimation

First, let us consider the simple case of a sample of i.i.d Gaussian variables $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2)$. The standard choice for the estimation of the mean μ is to compute the maximum likelihood estimator which, in this case, corresponds to the empirical mean

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i. \tag{2.1.1}$$

Under the present assumptions, the latter enjoys the following high probability inequality

$$\mathbb{P}\Big(|\widehat{\mu} - \mu| > \sqrt{\frac{2\sigma^2 \log(2/\delta)}{n}} \Big) \leqslant \delta. \tag{2.1.2}$$

Observe that the deviation bound is sub-Gaussian and scales in $\sqrt{\log(1/\delta)}$ for small confidence level $\delta \to 0$ which is optimal and corresponds to the asymptotic limit of the central limit theorem. However, if we remove the assumption that the X_is follow a Gaussian distribution and consider a heavy-tailed source distribution with mean $\mathbb{E}[X] = \mu$ and variance $\mathbb{E}[(X-\mu)^2] = \sigma^2$, then using Tchebyshev's inequality, we obtain the following for the empirical mean

$$\mathbb{P}\Big(|\widehat{\mu} - \mu| > \sqrt{\frac{\sigma^2}{\delta n}} \Big) \leqslant \delta,$$

where the new dependency in $\sqrt{1/\delta}$ yields much wider confidence intervals for small values of δ. Moreover, it is possible to design distributions which saturate the above inequality [73]. This reflects the poor accuracy of the empirical mean on heavy-tailed distributions.

Fortunately, it is possible to define mean estimators which satisfy the sub-Gaussian rate of (2.1.2) for distributions only satisfying a finite variance assumption. A notable example is given by M-estimators which are defined as the solution in $\widehat{\mu}_\rho$ to the problem

$$\sum_{i=1}^{n} \operatorname{sign}(\widehat{\mu}_\rho - X_i) \rho\big(|\widehat{\mu}_\rho - X_i| / \beta \big) = 0 \tag{2.1.3}$$

where $\beta > 0$ is a scale parameter and ρ is an *influence function* such that ρ is non-decreasing on $\mathbb{R}^+$ and satisfies $\rho(0) = 0$.

It is easy to see that one recovers the empirical mean by setting ρ as the identity function. Note also that setting ρ constant equal to 1 recovers the median. These two extreme choices illustrate the purpose of the previous definition to modulate the influence of individual samples by interpolating between the standard mean (which is an unbiased but non robust estimator) and the median (which is generally biased for the mean but very robust). In particular, the influence of outliers on the estimator $\widehat{\mu}_\rho$ is limited by choosing ρ such that $\rho(x) = o(x)$ for high values of x. The main examples for ρ are:

1. Huber's influence function $\rho_H(x) = x\mathbf{1}_{x \leqslant 1} + \mathbf{1}_{x>1}$, which was introduced by [202].
2. Catoni's influence function $\rho_C(x) = \log(1 + x + x^2/2)$, introduced by [73].
3. The polynomial influence function $\rho_P(x) = \frac{x}{1+x^{1-1/p}}$ for some polynomial degree $p > 1$.

These influence functions and the properties of the associated M-estimators are studied in [300]. For a properly chosen scale β (which depends on the random samples' variance), these estimators have a sub-Gaussian deviation bound similar to (2.1.2). In particular, Catoni's estimator, obtained using $\rho_C(x)$, has the remarkable property of satisfying this bound with optimal constants [110].

The main shortcomings of M-estimators lie in the need to adjust the scale parameter β which must be done according to the unknown data variance. In addition, solving (2.1.3) may be computationally slow. More importantly, if ρ is chosen unbounded, all guarantees on the resulting estimator fail as soon as a single sample is corrupted (i.e. has unknown or potentially adversarial origin).

The robustness of an estimator to corrupted samples can be measured by the notion of *breakdown point* introduced in [129]. We say that an estimator has breakdown point $\varepsilon \in [0, 1/2]$ if ε is the smallest required proportion of corrupted instances in order to make the estimator output arbitrary values. Formally, for an estimator $T : \cup_{n\geqslant 0}\mathbb{R}^n \to \mathbb{R}$, we propose to define $\varepsilon := \lim_{n\to\infty} \varepsilon_n$ where

$$\varepsilon_n := \min\left\{\frac{m}{n}, m \geqslant 0 \ : \sup_{X_j, Y_j \,:\, \sum_j \mathbf{1}_{X_j \neq Y_j} \leqslant m} \big|T(X_1, \dots, X_n) - T(Y_1, \dots, Y_n)\big| = \infty\right\},$$

(see also [299, Section 2.1.3]). For example, the empirical mean has breakdown point $\varepsilon = 0$ since it can be thrown off by a single outlier with arbitrary value (we also say that the finite-sample breakdown point is $\varepsilon_n = 1/n$). More generally, the same holds for M-estimators with unbounded ρ. Note that the highest possible breakdown point is $1/2$ since it becomes impossible to distinguish the underlying data distribution from the corruption when the latter affects over half the data [384]. This breakdown point value is reached by the median and also by all M-estimators with bounded ρ [379].

We mention two additional robust scalar mean estimators which can handle corrupted samples but do not belong to M-estimators.

The Median-Of-Means (MOM) estimator [7, 217, 339] partitions the samples into K blocks of equal size $\{1, \dots, n\} = \cup_{k=1}^K B_k$ and returns the median of the blockwise means

$$\widehat{\mu}^K_{\mathrm{MOM}} = \mathrm{median}\left(\Big(\frac{1}{|B_k|}\sum_{i\in B_k} X_i\Big)_{k=1}^K\right).$$

Under the previous finite variance assumption, the MOM estimator satisfies the following bound [284]

$$\mathbb{P}\Big(\big|\widehat{\mu}^K_{\mathrm{MOM}} - \mu\big| > 2\sigma\sqrt{K/n}\Big) \leqslant e^{-K/8},$$

where one may set the number of blocks as $K = \lceil 8\log(1/\delta)\rceil$ for some confidence level δ in order to obtain a sub-Gaussian bound as in (2.1.2). One can also show that a similar bound holds if the sample contains up to $K/2$ corruptions, although with worse constants. These properties together with its ease of computation make MOM an interesting robust estimator which has found use in multiple applications [283, 259, 261, 196]. However, the number of allowed corrupted samples is limited by the number of blocks K so that it would be necessary to set $K = \Omega(n)$ to obtain robustness to a fraction of corruptions in the data. This in turn makes MOM behave more like a median rather than a mean so that it is unfit for the η-corrupted setting.

The last robust scalar mean estimator we discuss is the trimmed mean introduced by [288].

Given a quantile index $\epsilon \in (0, 1/2)$, the latter estimates the mean by computing

$$\widehat{\mu}^{\epsilon}_{\mathrm{TM}} = \frac{2}{n} \sum_{i=n/2+1}^{n} X^{([\epsilon n/2])} \vee X_i \wedge X^{([(1-\epsilon)n/2])},$$

where $a \wedge b := \min(a, b), a \vee b := \max(a, b), [x]$ is the integer part of $x \in \mathbb{R}$ and $X^{(j)}$ is the j-th order statistic of $(X_i)_{i \leqslant n/2}$. In words, the sample is split in two halves. The first one is used to estimate the ϵ and $1 - \epsilon$ quantiles and the second one is used to compute a mean estimate by clipping all values between the previous quantiles. The above estimator is sometimes also called the "winsorized mean" and should not be confused with other definitions which simply discard samples outside the quantile interval rather than clipping them. Note also that dividing the sample is necessary for the theoretical study in order to ensure independence of the quantiles' estimation but is generally dropped in practice. Considering a sequence $X_1, \dots, X_n$ of samples drawn from a heavy-tailed distribution with finite variance σ^2 and which contains a proportion η of corruptions (possibly adversarial), the trimmed mean estimator with quantile $\epsilon = 8\eta + 24\frac{\log(4/\delta)}{n}$ satisfies the following deviation bound [288, Theorem 1]

$$\mathbb{P}\Big(\big|\widehat{\mu}^{\epsilon}_{\mathrm{TM}} - \mu\big| > 12\sigma\sqrt{2\eta} + 2\sigma\sqrt{\frac{\log(4/\delta)}{n}}\Big) \leqslant \delta. \tag{2.1.4}$$

Thus, the trimmed mean estimator is sub-Gaussian while also being robust to η-corruption and has optimal dependency $\sqrt{\eta}$ in the latter. Notice that, similarly to the MOM estimator, the trimmed mean depends on a parameter which must be set according to the desired confidence level δ. This dependency, which also occurs for robust M-estimators such as Catoni's mean, is in fact inevitable as no single estimator exists which works for all δ [110]. Returning to the trimmed mean $\widehat{\mu}^{\epsilon}_{\mathrm{TM}}$, the only non-trivial task for its computation is to determine the quantiles $X^{(\epsilon n/2)}$ and $X^{((1-\epsilon)n/2)}$ which can be done in linear time using the median of medians algorithm (see for instance [98, Chapter 9]) or simply the quickselect algorithm [189]. Thanks to the combination of these properties, we will see in the following chapters that the trimmed mean generally leads to the best compromise between strong robustness and computational speed for the solution of various learning problems in the robust setting.

The previous estimators provide fairly satisfactory solutions for robust scalar mean estimation. Unfortunately, we will see that this hardly extends to multidimensional mean estimation for which most algorithms either lack robustness or are computationally heavy. This represents an important obstacle to robust learning since the solution of problems of the form (2.0.1) crucially relies on accurate estimation of gradients as multidimensional means. A way around this issue is explored in Chapter 3 by combining robust scalar estimators with coordinate gradient descent (CGD) to solve linear learning problems.

2.1.2 Robust vector mean estimation

Before exploring robust estimators, we need to determine the best achievable concentration rate. In the multidimensional case, a broader choice of ways to measure the distance between an estimator $\widehat{\mu}$ and the true value μ is possible and will influence the type of obtainable bounds. Since the Euclidean distance is the most widely adopted one, we will focus on this case here. A natural choice of reference for concentration is to consider the asymptotically Gaussian rate stated by the central limit theorem. Therefore, our reference will be the deviation bound satisfied by the standard mean $\widehat{\mu}$ defined in (2.1.1) on an i.i.d Gaussian sample $X_1, \dots, X_n \sim \mathcal{N}(\mu, \Sigma)$ where $\mu \in \mathbb{R}^d$ for some $d > 1$ and $\Sigma \in \mathbb{R}^{d \times d}$ is a symmetric positive definite covariance matrix. In this setting, the following high probability bound can be derived using the Borell-TIS inequality [418,

50, 267]

$$\mathbb{P}\Big(\|\widehat{\mu}-\mu\| > \sqrt{\frac{\mathrm{Tr}(\Sigma)}{n}} + \sqrt{\frac{2\|\Sigma\|_{\mathrm{op}}\log(1/\delta)}{n}}\Big) \leqslant \delta,$$

where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm. An important feature of the above sub-Gaussian bound is that the deviation for different confidence levels δ is modulated by $\|\Sigma\|_{\mathrm{op}}$ which may be smaller than $\mathrm{Tr}(\Sigma)$ by a factor as much as d the space dimension.

Intractable estimators

Quite a few robust estimators are known [131, 289, 288, 84] which are not usable in practice because their computation has exponential complexity in the space dimension (intractable). A notable example is the *Tukey median* which is defined based on Tukey's Depth. For a point $u \in \mathbb{R}^d$, its Tukey depth relatively to a set of points $v_1, \ldots, v_n \in \mathbb{R}^d$ is defined as

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\big|\{1 \leqslant j \leqslant n : \langle w, u - v_j\rangle > 0\}\big|,$$

where $|\cdot|$ denotes the set cardinal. The Tukey median is then defined by maximizing this depth measure relatively to the sample set. This leads to a *maximum centrality* point similarly to the sub-Gaussian estimator of [289]. Note that the minimum with respect to $w \in \mathbb{R}^d$ in the definition can be restricted to unit vectors. Such optimization problems w.r.t. all possible space directions also appear in other robust vector mean estimators and are the reason for their intractability. The study carried out by [84] showed that Tukey's median is robust to Huber η-contamination if the true data distribution is Gaussian with identity covariance, in which case, it satisfies a deviation bound of the form

$$\mathbb{P}\Big(\|\widehat{\mu}_{\mathrm{Tukey}}-\mu\| > \sqrt{\frac{d}{n}} \vee \eta + \sqrt{\frac{\log(1/\delta)}{n}}\Big) \leqslant \delta.$$

These relatively restrictive assumptions are relaxed by the generalized trimmed mean estimator [288] which satisfies the following optimal deviation bound for η-corrupted data with finite covariance Σ :

$$\mathbb{P}\Big(\|\widehat{\mu}_{\mathrm{TM}}-\mu\| > C\Big(\sqrt{\|\Sigma\|_{\mathrm{op}}\eta} + \sqrt{\frac{\mathrm{Tr}(\Sigma)}{n}} + \sqrt{\frac{\|\Sigma\|_{\mathrm{op}}\log(1/\delta)}{n}}\Big)\Big) \leqslant \delta, \tag{2.1.5}$$

where C is an absolute constant. Note that dependence in the corruption rate worsens from η for Gaussian data to $\sqrt{\eta}$ under finite covariance. Unfortunately, the definition of the generalized trimmed mean is based on the computation of the quantiles of data vectors projected with respect to all possible directions in space. Consequently, a direct implementation would have complexity exponential in the dimension making the estimator intractable.

Polynomial complexity robust estimators

We now turn to computable estimators. A first naive idea is to use a robust scalar estimator separately for each coordinate. If the base estimator is sub-Gaussian, then for samples with finite covariance Σ, this leads to a bound of the form

$$\mathbb{P}\Big(\|\widehat{\mu}-\mu\| > C\sqrt{\frac{\mathrm{Tr}(\Sigma)\log(d/\delta)}{n}}\Big) \leqslant \delta,$$

where C is an absolute constant. Unfortunately, this new bound is clearly inferior since the confidence had to be replaced by δ/d for a union bound argument. More importantly, the fluctuations are modulated by $\mathrm{Tr}(\Sigma)$ rather than $\|\Sigma\|_{\mathrm{op}}$ implying a worse dimension dependence.

A slightly better solution is given by a multidimensional generalization of the MOM estimator called the geometric Median-Of-Means which was proposed by [315] (see also [196] for another generalization with similar properties). The latter computes block means and aggregates them using the following generalization of the median for vectors $v_1, \dots, v_K \in \mathbb{R}^d$:

$$\mathrm{median}(v_1, \dots, v_K) = \operatorname*{arg\,min}_{u \in \mathbb{R}^d} \sum_{i=1}^{K} \|u - v_i\|.$$

By setting the number of blocks as $K = \lceil 8 \log(1/\delta) \rceil$, the obtained estimator satisfies the slightly improved bound

$$\mathbb{P}\Big(\|\widehat{\mu} - \mu\| > C \sqrt{\frac{\mathrm{Tr}(\Sigma) \log(1/\delta)}{n}} \Big) \leqslant \delta,$$

with the additional perk of being robust to the presence of a number $\leqslant K/2$ of outliers in the sample. However, this estimator is still not sub-Gaussian due to the factor $\mathrm{Tr}(\Sigma)$ in front of $\log(1/\delta)$.

The first polynomial complexity sub-Gaussian vector mean estimator was proposed in [195] by designing an algorithm capable of certifying the centrality criterion of [289] using semidefinite programming (SDP). This concentration certification approach is also the basis for Sum-Of-Squares algorithms for robust mean estimation [244, 194] which also run in polynomial time. Progressive improvements in the literature eventually brought the complexity down to near linear time [92, 107]. These results initially considered little to no corruption and were later complemented by [268, 117] which introduced adversarial η-corruption and obtained performances approaching (2.1.5) (up to logarithmic factors) while also replacing SDP with lighter spectral methods.

Stability based robust vector mean estimation

The framework of [117] represents the state-of-the-art as of this writing and deserves a more elaborate discussion. They design algorithms which perform iterative filtering on the data sample and obtain an estimator $\widehat{\mu}_{\mathrm{DKP}}$ satisfying the following bound for an adversarially η-corrupted sample following a finite covariance distribution [117, Proposition 1.5]

$$\mathbb{P}\Big(\|\widehat{\mu}_{\mathrm{DKP}} - \mu\| > C \Big(\sqrt{\|\Sigma\|_{\mathrm{op}} \eta} + \sqrt{\frac{\mathrm{Tr}(\Sigma) \log(\mathrm{r}(\Sigma))}{n}} + \sqrt{\frac{\|\Sigma\|_{\mathrm{op}} \log(1/\delta)}{n}} \Big) \Big) \leqslant \delta, \qquad (2.1.6)$$

where $\mathrm{r}(\Sigma) = \mathrm{Tr}(\Sigma)/\|\Sigma\|_{\mathrm{op}}$ is the stable rank of Σ. The factor $\log(\mathrm{r}(\Sigma)) \leqslant \log(d)$ is the only sub-optimality in this bound and can actually be removed thanks to a Median-of-Means preprocessing step. However, the amount of corruption η the procedure can handle is then strongly restricted (see also Section 4.5.2). The framework of [117] hinges upon the following property.

Definition 2.1 (Stability)**.** *Fix $\epsilon \in (0, 1/2)$ and $\tau \geqslant \epsilon$. A finite set $S \subset \mathbb{R}^d$ is (ϵ, τ)-stable with respect to $\mu \in \mathbb{R}^d$ and σ^2 if for every $S' \subseteq S$ with $|S'| \geqslant (1 - \epsilon)|S|$ we have*

$$\|\mu_{S'} - \mu\| \leqslant \sigma\tau \quad \text{and} \quad \|\overline{\Sigma}_{S'} - \sigma^2 I_d\|_{\mathrm{op}} \leqslant \sigma^2 \tau^2 / \epsilon,$$

where $\mu_{S'}$ and $\overline{\Sigma}_{S'}$ are the mean and covariance of the set S' respectively.

Given this definition, the key mathematical claim motivating the stability based approach is the following.

Theorem 2.1 ([117, Theorem 1.4] informal)**.** *Let $\delta > 0, \eta > 0$ and $S = \{X_1, \dots, X_n\}$ be a set of i.i.d samples from a distribution over $\mathbb{R}^d$ with mean μ and covariance Σ. Assume $\epsilon := \Theta(\log(1/\delta)/n + \eta)$ is small enough. With probability at least $1 - \delta$, there exists a subset $S' \subseteq S$ such that $|S'| \geqslant (1-\epsilon)n$ and S' is $(2\epsilon, \tau)$-stable with respect to μ and $\|\Sigma\|_{\mathrm{op}}$ where $\tau = O\big(\sqrt{\mathrm{r}(\Sigma)\log \mathrm{r}(\Sigma)/n} + \sqrt{\eta} + \sqrt{\log(1/\delta)/n}\big)$.*

It follows from this result that, with high probability, an η-corrupted set of samples following a finite covariance distribution contains a (ϵ', τ')-stable subset where (ϵ', τ') are equivalent to (ϵ, τ) in the above statement up to constant factors. Consequently, computing an estimator satisfying (2.1.6) can be done by identifying such a subset and leveraging the first inequality of Definition 2.1. This requirement is actually relaxed to finding weights $w^* \in \mathbb{R}^n_+$ with $\sum_i w^*_i = 1$ over the set of samples in order to compute the estimator as

$$\widehat{\mu}_{\mathrm{DKP}} = \mu_{w^*} := \sum_i w^*_i X_i,$$

The task of determining w^* is executed by an iterative filtering algorithm which proceeds roughly as follows (see [116, 117] for details). Given ϵ and τ, start from uniform weights $w \in \mathbb{R}^n_+$ and loop over the steps:

- Compute the weighted mean and weighted covariance

$$\mu_w = \sum_i w_i X_i \quad \text{and} \quad \Sigma_w = \sum_i w_i (X_i - \mu_w)(X_i - \mu_w)^\top,$$

  with respect to the current weights.
- Compute the largest eigenvector v of Σ_w.
- Downweight the ϵ portion of samples with highest contribution to the variance along direction v.

The loop continues until the stopping criterion $\|\Sigma_w\|_{\mathrm{op}} \leqslant \|\Sigma\|_{\mathrm{op}}(1 + O(\tau^2/\epsilon))$ is met. The complexity of this algorithm is claimed to be polynomial but not precisely specified by the authors of [117]. In practice, the procedure involves several loops over the data with non negligible linear algebra operations at each turn making it significantly slower than the computation of the geometric MOM for example.

Robust vector mean estimation w.r.t a non-Euclidean metric

As mentioned above, the accuracy measure of a vector mean estimator depends on the norm chosen for this purpose. For an arbitrary norm, the first important question is to determine the corresponding optimal rate which will serve as a reference to judge a candidate estimator. This task was addressed by [285] who derived an entropy based lower bound on the optimal performance along with an (intractable) estimator to achieve it. The previous lower bound is shown to be tight up to a minor caveat which stems from the looseness of Sudakov's minoration inequality [267, 432] and may lead to a logarithmic gap in the Euclidean case. This gap was filled in [106] by replacing the entropy with the Gaussian width as statistical complexity measure. They also propose a convex optimization algorithm for the estimation on corrupted and heavy-tailed data.

In Chapter 4, we consider robust linear learning in the high-dimensional setting. We adopt an approach based on non-Euclidean optimization in order to minimize the impact of the high dimension. This entails that the gradient estimation error is measured with non-Euclidean norms and we take advantage of this to propose efficient robust learning procedures using adapted estimators.

2.1.3 Robustness in the streaming setting

Up to this point, we discussed robust estimation in the *batch* setting where one assumes free access to all data samples and no particular memory constraint. However, there are numerous contexts in which this luxury is not available and one must incrementally build an estimator using a single sample at a time without the possibility to go back to previous ones. This situation arises for example in streaming stochastic optimization [47, 49, 307] and online learning problems [183, 77, 347] where data is accessible as a sequence of individual samples.

We propose to discuss streaming robustness through a stochastic optimization problem with a similar objective to (2.0.1). We rewrite it here for clearness

$$\min_{\theta} \mathcal{L}(\theta) := \mathbb{E}_{\zeta}[\ell(\theta, \zeta)]. \tag{2.1.7}$$

We assume that the objective $\mathcal{L}$ satisfies at least the properties of convexity and smoothness and that it admits a minimum $\mathcal{L}^\star$ so that it can be optimized using a gradient method. Notice that basic mean estimation can be formulated as a particular case of this problem by setting $\zeta = X \in \mathbb{R}^d$ and $\ell(\theta, X) = \|\theta - X\|^2$. Given a sequence of random samples $\zeta_1, \dots, \zeta_T$, the previous objective can be optimized using stochastic gradient descent (SGD) with gradient samples $\nabla\ell(\theta, \zeta_t)$ (assuming differentiability of ℓ) through the iteration

$$\theta_{t+1} = \theta_t - \beta_t \nabla\ell(\theta_t, \zeta_t) \tag{2.1.8}$$

started at some $\theta_0 \in \mathbb{R}^d$ and with step sizes $\beta_t > 0$. The performance of an estimator $\widehat{\theta}_T$ after T iterations is measured either in terms of the excess risk $\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}^\star$ or in terms of the square distance $\|\widehat{\theta}_T - \theta^\star\|^2$ to an optimal $\theta^\star$ such that $\mathcal{L}(\theta^\star) = \mathcal{L}^\star$ when a unique one exists (this is the case for strongly convex functions for example).

Early guarantees for SGD

Early works derived guarantees in expectation on the optimality gap after T steps. For instance, [371] obtained the following result.

Theorem 2.2 ([371, Theorem 1])**.** *Assume the objective $\mathcal{L}$ is μ-strongly convex and L-smooth and that $\mathbb{E}[\|\nabla\ell(\theta_t, \zeta_t)\|^2] \leqslant G^2$ for all t. Let θ_T be the result of running Iteration* (2.1.8) *for T steps starting from $\theta_0 = 0$ and using step-sizes $\beta_t = (\mu t)^{-1}$, we have*

$$\mathbb{E}[\mathcal{L}(\theta_T) - \mathcal{L}^\star] \leqslant \frac{2LG^2}{\mu^2 T}. \tag{2.1.9}$$

Although they guarantee that a precise solution is reached in the average case, results in expectation give little confidence about a particular estimate $\widehat{\theta}$. Indeed, using a Markov inequality on Inequality (2.1.9) only yields a bound on $\mathcal{L}(\theta_T) - \mathcal{L}^\star$ which poorly scales as δ^{-1} for confidence δ.

Deriving bounds with milder fluctuations is enabled by assuming concentration properties for the random gradients. Specifically, assumptions generally pertain to the behavior of the gradient errors

$$\varepsilon_\zeta(\theta) := \nabla\ell(\theta, \zeta) - \nabla\mathcal{L}(\theta). \tag{2.1.10}$$

One of the earliest results of this kind appeared in [17] who used an L_p assumption on $\varepsilon_\zeta(\theta)$ (with $p > 1$) in order to show L_p convergence of $\mathcal{L}(\overline{\theta}_T) - \mathcal{L}^\star$ for constant step-size $\beta = O(1/p)$ where $\overline{\theta}_T = T^{-1}\sum_{t=0}^{T-1}\theta_t$ is the average iterate. This leads to an improved confidence bound of order $O(\delta^{-1/p})$ on the optimality gap $\mathcal{L}(\overline{\theta}_T) - \mathcal{L}^\star$ for confidence level δ (we neglect the other

involved constants for simplicity, see Theorem 2 and Corollary 1 of [17] for details).

High-confidence results for SGD

We are interested in high-confidence bounds on the optimality gap in terms of $\log(1/\delta)$ for confidence level δ. For example, such a result was presented in [179] who defined an estimator $\widehat{\theta}_T$ based on T steps of SGD and satisfying the following bound

$$\mathbb{P}\Big(\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}^\star \geqslant O\Big(\frac{1}{T} \cdot \frac{L \cdot \log(1/\delta) + L^2}{\mu}\Big)\Big) \leqslant \delta,$$

where the objective $\mathcal{L}$ is assumed L-Lipschitz and μ-strongly convex. However, the previous bound was derived under the strong condition of bounded gradient errors $\varepsilon_\zeta(\theta)$. Similar results were obtained for sub-Gaussian gradient errors in [338] but that remains a strong hypothesis. From a robust statistics' perspective, a first question is whether such high-confidence bounds can be obtained when the distribution of $\varepsilon_\zeta(\theta)$ is allowed to be heavy-tailed.

This was first achieved by [166] who considered an L-smooth convex objective and assumed uniformly bounded variance of the gradient error

$$\mathbb{E}[\|\varepsilon_\zeta(\theta)\|^2] \leqslant \sigma^2 < +\infty. \tag{2.1.11}$$

They used the following minibatch clipped SGD iteration

$$\theta_{t+1} = \theta_t - \beta_t \operatorname{clip}\big(\widehat{\nabla}_m \ell(\theta_t), \lambda_t\big) \quad \text{with} \quad \operatorname{clip}(v, \lambda) := \min\Big(1, \frac{\lambda}{\|v\|}\Big) \cdot v, \tag{2.1.12}$$

where $\beta_t > 0$ are step-sizes, $\lambda_t > 0$ are clipping thresholds and $\widehat{\nabla}_m \ell(\theta) = \frac{1}{m}\sum_{j=1}^m \nabla \ell(\theta, \zeta_j)$ is a minibatch gradient average of size m. This leads to the below statement.

Theorem 2.3 ([166, Theorem 3.1] informal)**.** *Let $\mathcal{L}$ be convex and L-smooth with unique minimum at $\theta^\star \in \mathbb{R}^d$ and grant* (2.1.11)*. Let $\theta_0 \in \mathbb{R}^d$ be an initial parameter such that $R_0 = \|\theta_0 - \theta^\star\|$ and $T \geqslant 1$ the horizon. For confidence level $\delta > 0$, assume that Iteration* (2.1.12) *is run with*

$$m = \Theta\Big(\frac{T\sigma^2}{R_0^2 L^2 \log(T/\delta)}\Big), \quad \lambda_t = \lambda = \Theta(LR_0) \quad \textit{and} \quad \beta_t = \beta = O\big((L\log(T/\delta))^{-1}\big),$$

then we have the following bound for $\overline{\theta}_T = \frac{1}{T}\sum_{t=0}^{T-1}$:

$$\mathbb{P}\Big(\mathcal{L}(\overline{\theta}_T) - \mathcal{L}(\theta^\star) \geqslant O\Big(\frac{LR_0^2 \log(T/\delta)}{T}\Big)\Big) \leqslant \delta. \tag{2.1.13}$$

The work of [166] was nonetheless preceded by [330] who obtained high-confidence guarantees for stochastic mirror descent (SMD) using a similar truncation approach. Still thanks to clipped SGD, [417] showed an equivalent result for strongly convex objectives without minibatching and using decreasing step-sizes $\beta_t = \Theta(1/t)$ and a fixed clipping threshold of order $\lambda = \Omega\big(\sqrt{T/\log(1/\delta)}\big)$. Apart from removing the need for minibatching, the main improvement brought by [417] is relaxing the uniform variance bound (2.1.11) to

$$\mathbb{E}[\|\varepsilon_\zeta(\theta)\|^2] \leqslant a\|\theta - \theta^\star\|^2 + b \quad \text{for some} \quad a, b > 0,$$

which is a much more reasonable assumption for common problems like linear regression. It is important to notice that [166] and [417] both use high clipping thresholds compared to the expected values. For [166], minibatch averages of size $m = \Omega(T)$ are used so that, near the

optimum, $\|\widehat{\nabla}_m \ell(\theta)\|$ has order $O(1/\sqrt{T})$ while the applied threshold has constant order. Analogously, [417] uses $m = 1$ so that $\|\widehat{\nabla}_m \ell(\theta)\|$ has constant order near the optimum and the clipping threshold is set to $\Omega(\sqrt{T})$. This high value is compensated by the decreasing step-size which has order $O(1/T)$ near the end of the iteration. In both cases, the final estimator $\widehat{\theta}$ satisfies $\|\widehat{\theta} - \theta^\star\| = O(1/\sqrt{T})$ which leads to an $O(1/T)$ rate as in (2.1.13) (up to a logarithmic factor) since L-smoothness implies

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) \leqslant \frac{L}{2}\|\theta - \theta^\star\|^2.$$

Subsequent work [389, 346] further weakened this requirement on $\varepsilon_\zeta(\theta)$ to the existence of a finite α-moment with $\alpha \in (1, 2]$. Note that in this case, the optimal rate degrades from $O(1/T)$ to $O(T^{-2(\alpha-1)/\alpha})$. Additional developments for non-smooth objectives were also recently brought by [277].

Corruption robustness for streaming data

As of this writing, few works have considered outlier robust procedures for streaming data i.e. when samples are received individually and sequentially so that algorithms are constrained to work with a single sample at a time without access to history. The main reference on the subject is [118] which deals with vector mean estimation from an adversarially corrupted stream of heavy-tailed samples. The proposed algorithm adapts the filtering scheme which appeared in [117] for the batch case by restricting it to minibatches of logarithmic size and introducing a number of optimizations to the necessary linear algebra operations to avoid an $O(d^2)$ complexity per iteration. Remarkably, the resulting algorithm attains the optimal robust rates with respect to the corruption rate η and applications of it are considered for strongly convex learning tasks like linear and logistic regression on streaming data. However, the study of the algorithm is purely theoretical and its implementation may be difficult due to its complex steps and number of unknown absolute constants on which it depends.

In Chapter 5, we propose a special clipping strategy for SGD making it robust both to heavy tails and corruptions. Our procedure is quite easy to implement and is supported by rigorous theoretical analysis confirming its robustness.

2.2 Robust Learning with Linear Models

In machine learning, the task is often to design a model which predicts an answer $Y \in \mathcal{Y}$ based on *features* $X \in \mathcal{X}$ as accurately as possible. We will consider the very common situation where the feature space is Euclidean $\mathcal{X} = \mathbb{R}^d$. A prediction model is a measurable function $\phi : \mathcal{X} \to \widehat{\mathcal{Y}}$ which, given an instance X, outputs $\phi(X)$ predicting the ground truth Y. The accuracy of a model can be measured in different ways. This is generally done using a loss function $\ell : \widehat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}_+$ which evaluates the gap between prediction and ground truth as $\ell(\phi(X), Y)$. The overall performance of ϕ is measured by its *risk* which corresponds to the expectation over the data distribution of (X, Y):

$$\mathcal{R}(\phi) := \mathbb{E}[\ell(\phi(X), Y)].$$

Depending on the nature of the labels Y which need to be predicted, we refer to the learning task at hand as a regression problem if $Y \in \mathcal{Y} = \mathbb{R}$ is a real value and as a classification problem if $Y \in \mathcal{Y} = \{\pm 1\}$ is a binary class (multi-class extensions can also be considered). The loss function ℓ is then chosen according to the learning task.

The predictor ϕ is generally selected among a class of functions Φ based on the observed performance on a training sample of instances $(X_1, Y_1), \ldots, (X_n, Y_n)$. This step aims to find a predictor minimizing the risk $\mathcal{R}(\phi)$ with good generalization properties i.e. good prediction

performance on new (test) data outside the training sample. Obviously, the size of the class Φ determines the attainable performance according to how well functions $\phi \in \Phi$ capture the dependency between X and Y. Note however that considering excessively complex models is a pitfall since it may complicate the task of finding the optimal predictor or lead to *overfitting*.

Indeed, it is very common to train prediction models by minimizing the *empirical risk* with respect to the training sample $(X_i, Y_i)_{i=1}^n$ defined as

$$\widehat{\mathcal{R}}_n(\phi) := \frac{1}{n}\sum_{i=1}^{n} \ell(\phi(X_i), Y_i). \tag{2.2.1}$$

However, $\widehat{\mathcal{R}}_n$ may not be a good surrogate for $\mathcal{R}$ and overfitting occurs when a predictor ϕ achieves high performance on training data as measured by $\widehat{\mathcal{R}}_n$ but fails to generalize on new test data suggesting a high value of the true risk $\mathcal{R}$ [329, 60]. This phenomenon is exacerbated by small sample counts n and excessively complex function classes Φ which tend to capture too much noise from the data. A solution to mitigate this problem is to add a regularization term to the empirical risk (2.2.1) which penalizes predictors according to their complexity and favors "simpler" ones [36].

Linear models correspond to setting Φ to be the class of linear functions which is a customary choice for many problems. Indeed, this model set often provides satisfactory predictive power while allowing to use off-the-shelf optimization methods to determine a good candidate $\phi^\star \in \Phi$. Each model can be represented by a vector $\theta \in \mathbb{R}^d$ such that $\phi(X) = \theta^\top X$ and finding a good predictor $\phi^\star$ is equivalent to determining the associated vector $\theta^\star$. By choosing a convex and smooth loss function ℓ, finding a low risk predictor boils down to the convex optimization problem

$$\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta) := \mathbb{E}[\ell(\theta^\top X, Y)], \tag{2.2.2}$$

which can be efficiently solved using a gradient method. Two of the most famous examples within this framework are:

- Least squares regression : which corresponds to learning to predict a real value Y by defining the objective using the square loss $\ell(y, z) = \frac{1}{2}(y - z)^2$.
- Logistic regression : in which a binary label $Y \in \{\pm 1\}$ is predicted by optimizing the objective $\mathcal{L}$ defined with the logistic loss function $\ell(y, z) = \log(1 + \exp(-yz))$.

Note that the linear prediction framework can be extended to more complex models by considering polynomial features in the variables of X or kernel methods [403, 245].

Nevertheless, the approach one should adopt in order to solve Problem (2.2.2) is not completely obvious since the data (X, Y) follow a probability distribution which is unknown in general and only accessible through a training sample $(X_i, Y_i)_{i=1}^n$. As mentioned above, a standard solution is to replace the true risk $\mathcal{L}$ by the empirical risk

$$\widehat{\mathcal{L}}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta^\top X_i, Y_i), \tag{2.2.3}$$

which is defined using the training sample and may be readily optimized to obtain an estimator $\widehat{\theta}_{\text{erm}}$ (empirical risk minimizer). In order to guarantee that the latter is a good optimizer for $\mathcal{L}$, it is necessary that $\widehat{\mathcal{L}}_n$ provides a good approximation of the true objective in the sense that the deviation $|\widehat{\mathcal{L}}_n(\theta) - \mathcal{L}(\theta)|$ satisfies a sub-Gaussian bound similar to (2.1.2). This can be shown to hold under strong assumptions on the distribution of the data samples (X, Y) such as sub-Gaussian concentration or boundedness [427, 298, 157]. However, these guarantees fail for

heavy-tailed or corrupted data which call for a more robust approach. For a regression problem, an early proposition was to replace the square loss by Huber's loss

$$\ell_H(y,z) = \begin{cases} \frac{1}{2}(y-z)^2 & \text{if} \quad |y-z| \leqslant 1 \\ |y-z| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{2.2.4}$$

which is able to minimize the effect of outliers thanks to its slower linear rather than quadratic rate for $|y-z| \gg 1$. Although this allows to handle heavy-tailed labels Y, it remains insufficient when assumptions on the covariates X are weakened. Another possibility is to use repeated robust scalar estimation on the objective $\mathcal{L}$ in order to optimize it [57]. However, this procedure turns out to be unstable and difficult to implement.

Given that gradient based methods provide the best approach to optimization tasks in general, a natural idea is to focus on estimating the gradient of $\mathcal{L}$ rather than the objective itself and perform gradient descent [364]. Unfortunately, this requires to employ robust vector mean estimation algorithms which, despite numerous efforts, remain computationally costly or unreliable in certain robust settings as discussed in Section 2.1.2.

2.2.1 Contribution: Robust Learning with Coordinate Gradient Descent

To enable robust learning without running into the difficulty of robust vector mean estimation, we propose in Chapter 3 to replace the conventional gradient descent iteration by coordinate gradient descent [341, 442] using a robust scalar estimator of the partial derivatives. For iterates $\theta^{(t)}$, this corresponds to the update

$$\theta_j^{(t+1)} = \begin{cases} \theta_j^{(t)} - \beta_j \widehat{g}_j(\theta^{(t)}) & \text{if } j = j_t \\ \theta_j^{(t)} & \text{otherwise,} \end{cases} \tag{2.2.5}$$

where $1 \leqslant j_t \leqslant d$, the β_js are step-sizes and $\widehat{g}_j(\theta^{(t)})$ is a robust estimator of $\frac{\partial \mathcal{L}(\theta^{(t)})}{\partial \theta_j}$. For such an estimator $\widehat{g}$ of the partial derivatives, we define its error vector $\epsilon(\delta)$ which, for a failure probability $\delta \in (0,1)$, satisfies for all $j \in [\![d]\!]$:

$$\mathbb{P}\Big[\sup_{\theta} \big|\widehat{g}_j(\theta) - g_j(\theta)\big| \leqslant \epsilon_j(\delta) \Big] \geqslant 1 - \delta.$$

With this definition, we have the following result for robust learning with CGD.

Theorem 2.4 (Theorem 3.1 informal)**.** *Assume the objective $\mathcal{L}(\theta)$ is λ-strongly convex and L_j-Lipschitz-smooth w.r.t. each coordinate θ_j. Let $\theta^{(T)}$ be the output of Iteration (2.2.5) with step-sizes $\beta_j = 1/L_j$, an initial iterate $\theta^{(0)}$, and coordinates j_t sampled according to the importance sampling distribution $p_j = L_j / \sum_{k \in [\![d]\!]} L_k$, we have*

$$\mathbb{E}\big[\mathcal{L}(\theta^{(T)})\big] - \mathcal{L}^\star \leqslant \big(\mathcal{L}(\theta^{(0)}) - \mathcal{L}^\star\big)\Big(1 - \frac{\lambda}{\sum_{j \in [\![d]\!]} L_j}\Big)^T + \frac{1}{2\lambda}\big\|\epsilon(\delta)\big\|_2^2,$$

with probability at least $1-\delta$, where the expectation is w.r.t. coordinate sampling of j_t.

The value of $\|\epsilon(\delta)\|_2^2$ in Theorem 2.4 depends on the gradient estimator used in Iteration (2.2.5). By plugging robust scalar mean estimators, we obtain robust procedures with minimal computational overhead compared to non robust gradient descent using the empirical risk (2.2.3). We also obtain similar results to Theorem 2.4 for various CGD coordinate sampling strategies. For a smooth but non-strongly convex objective, we show that the optimization

converges despite the potential errors by clipping the gradient estimate within the limits of the confidence bound it satisfies. We consider three estimators: Catoni's estimator, the Median-Of-

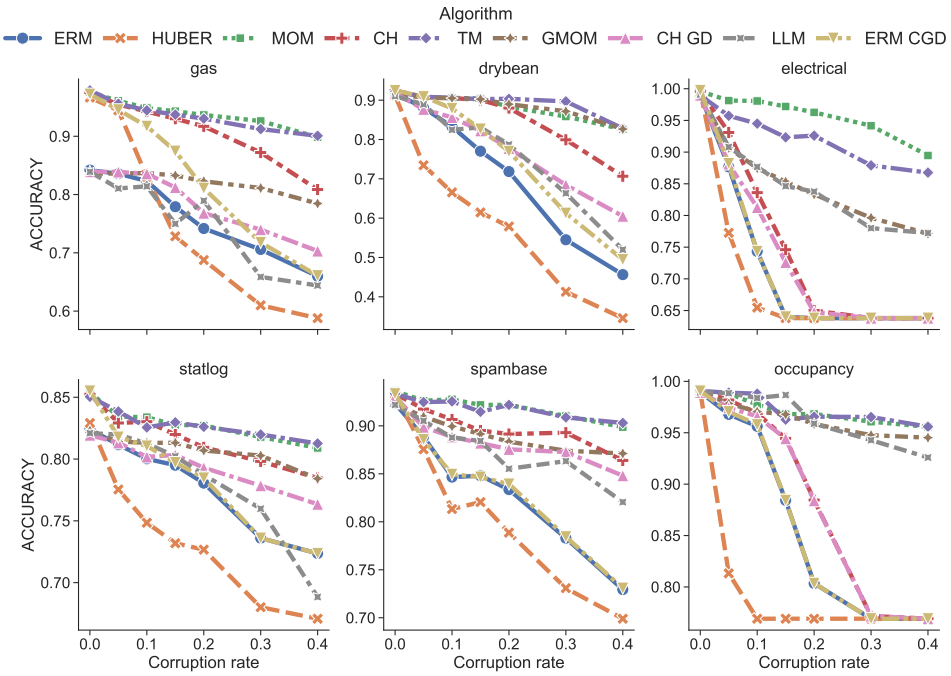

Figure 2.2: Test accuracy (y-axis) against the proportion of corrupted samples (x-axis) for six data sets and various algorithms. `MOM`, `CH` and `TM` represent CGD using MOM, Catoni's estimator and the trimmed mean respectively. `ERM` and `ERM CGD` represent gradient descent (GD) and CGD using the empirical mean. `HUBER` uses GD and the modified Huber loss [452] for classification. The remaining baselines are algorithms drawn from [262, 364, 191].

Means and the trimmed mean which we defined previously. These allow to handle corrupted and heavy-tailed gradients, including the infinite variance case. We write an efficient implementation of the corresponding learning algorithms and provide public access to the code through a Python library called `linlearn`[1].

Using our implementation, we carry out an extensive experimental comparison of robust CGD with other robust learning methods from the literature [262, 364, 191]. Figure 2.2 is drawn from Chapter 3 and shows the comparison in terms of accuracy on several classification datasets for increasing proportions of corrupted data. The results show that CGD based algorithms (especially `MOM` and `TM`) are robust even to high levels of corruption and preserve good performance in this case.

2.2.2 The high-dimensional case

Machine learning using linear models is sometimes confronted with situations where the data is represented in a high-dimensional space $X \in \mathcal{X} = \mathbb{R}^d$ with $d \gg 1$. Beyond computational considerations, the situation where the dimension is greater than the sample count ($d > n$) leads to over-determination which requires the model to be modified. To solve this issue, it is standard to add a *sparsity* assumption [66, 61, 425] which posits that the label Y can be predicted using only a subset of size $s < d$ of the available features. This is equivalent to assuming that the optimum $\theta^\star = \arg\min_\theta \mathcal{L}(\theta)$ is s-sparse i.e. all but s of its coordinates are equal to zero.

[1] `https://github.com/linlearn/linlearn`

Since this subset of non zero coordinates is not known a priori, the optimization of the objective (2.2.2) still needs to be carried out in high-dimensional space where conventional gradient descent would be an inefficient solution. More appropriate methods in this context are offered by *non-Euclidean* optimization schemes such as Mirror descent [339] and Dual averaging [343]. Indeed, for a function f and step-size β, conventional gradient descent performs the iteration

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

The latter implicitly considers the Euclidean metric on the parameter space so that it is isometric with its *dual* which is the gradient space. Indeed, the above iteration is equivalent to

$$\theta_{t+1} = \arg\min_{\theta} \Big\{ \langle \theta, \beta \nabla f(\theta_t) \rangle + \frac{1}{2} \|\theta - \theta_t\|^2 \Big\},$$

which uses the Euclidean norm on the parameter space. Mirror descent may be seen as a generalization of gradient descent which replaces the Euclidean metric with a *Bregman divergence* [339] $V(\theta, \theta')$ resulting in the iteration

$$\theta_{t+1} = \arg\min_{\theta} \Big\{ \langle \theta, \beta \nabla f(\theta_t) \rangle + V(\theta, \theta_t) \Big\}.$$

In this fashion, depending on the choice of V, a different metric is induced on the gradient space by duality. In particular, a judicious choice allows to make the optimization more efficient in the high-dimensional case.

2.2.3 Contribution: Robust high-dimensional learning

In Chapter 4, we tackle smooth learning problems with linear models in the high-dimensional case by using Mirror descent. We take inspiration from [223] to define a multistage Mirror descent based robust learning algorithm which we call AMMD (Approximate Multistage Mirror Descent). The latter can be used for different learning tasks including Vanilla sparse estimation, group-sparse estimation and low-rank matrix recovery by plugging an appropriate gradient estimator in each case.

Depending on the problem, different Bregman divergences are used for Mirror descent inducing different metrics on the gradient space. For Vanilla sparse estimation in particular, the parameter space is endowed with the ℓ_1 metric which induces the ℓ_∞ metric on the gradient space. In this case, the gradient can be estimated with near optimal statistical rate and minimal computational overhead simply using a componentwise trimmed mean. Indeed, we have the following statement.

Lemma 2.1. *Let $X_1, \dots, X_n$ be an i.i.d η-corrupted sample from a random variable $X \in \mathbb{R}^d$ such that each coordinate of X has variance smaller than $\overline{\sigma}^2 < +\infty$. Let $\delta > 0$ and $\widetilde{\mu}^{\epsilon}_{\mathrm{TM}}$ be the coordinatewise trimmed mean estimator of $\mu = \mathbb{E}[X]$ computed on $(X_i)_{i=1}^n$ with parameter $\epsilon = 8\eta + 24\frac{\log(4/\delta)}{n}$, we have*

$$\mathbb{P}\left(\left\| \widetilde{\mu}^{\epsilon}_{\mathrm{TM}} - \mu \right\|_\infty > 12\overline{\sigma}\sqrt{2\eta} + 2\overline{\sigma}\sqrt{\frac{\log(d) + \log(4/\delta)}{n}} \right) \leqslant \delta.$$

Lemma 2.1 follows from (2.1.4) with a union bound argument. This leads to an estimator $\widehat{\theta}$ for Vanilla sparse estimation which is robust to heavy tails and corruption and satisfies the

following estimation bound (see Chapter 4 Corollary 4.1 for more details)

$$\mathbb{P}\left(\left\|\widehat{\theta}-\theta^{\star}\right\|_2 > \frac{2^{-K/2}R}{\sqrt{\bar{s}}} + \frac{140\sqrt{2\bar{s}}\sigma_{\max}}{\kappa}\sqrt{4\eta + 6\frac{\log(4/\widetilde{\delta}) + \log(d)}{\widetilde{n}}}\right) \leqslant \delta,$$

where $\widetilde{\delta} = \delta/T$ and $\widetilde{n} = n/T$ with T the number of iterations, $\bar{s}$ is an upperbound on the sparsity s, K is the number of stages run by the algorithm and $\sigma_{\max}$ is a uniform bound on the gradient coordinate variances.

We also extend the analysis to non-smooth objectives by defining a similar algorithm to AMMD based on Dual averaging [343]. Further, we propose robust estimators achieving near optimal statistical rates to cover the group-sparse estimation and low-rank matrix recovery problems according to the emerging metrics. We implement these algorithms in the `linlearn` library and support the theoretical results by numerical experiments on real datasets which confirm the efficiency of the proposed algorithms compared to the existing baselines.

2.3 SGD Robustness and Concentration Properties

After considering robust learning methods in the batch setting, we now turn our attention to the same problem for streaming data by considering the stochastic optimization problem (2.1.7). Among the most relevant references are the articles of [166, 417], both of which tackle problem (2.1.7) with heavy-tailed gradient samples using clipped SGD (2.1.12) and obtain high-confidence deviation bounds on the final optimality gap. These works use the martingale framework to study the estimation error through the iteration and apply Freedman's concentration inequality [150] (also referred to as Bernstein's inequality for martingales) in order to derive their final concentration bounds.

Although SGD with heavy-tailed gradient samples received a fair amount of attention [166, 417, 389, 346], a solution able to handle corrupted data was still lacking. We consider this problem in Chapter 5 under a simple Huber-like corruption model in which, at each round, the received gradient sample is corrupted with probability $\eta < 1/2$ independently from other rounds. Simple intuition suggests that, if one were to run SGD and cap the contribution of each gradient sample low enough, then the optimization will not be led astray by corruptions since these only constitute a minority of the samples ($\eta < 1/2$). In other words, a clipped version of SGD is robust to our η-corruption model.

In order to avoid excessively slowing the optimization, the clipping threshold needs to be set in an adaptive and robust way. A choice fitting these criteria is provided, for instance, by the gradient samples' median norm which can be estimated in practice by tracking a short history of the gradient norms. In order to analyze this procedure, we shift from the martingale framework used by [166, 417] to the Markov chain framework allowing to side-step the combinatorial task of enumerating the possible sequences of corrupted and uncorrupted samples and their probabilities.

2.3.1 Contribution: Robust SGD to heavy-tailed and corrupted data

We carry out the theoretical analysis of the algorithm informally described above in Chapter 5. For this task, we employ tools from generic space Markov chain theory [313] to establish convergence in distribution and characterize the attained limit. As the analysis shows, the iteration is best defined using a quantile of the expected gradient norm as clipping threshold. Denoting $G(\theta, \zeta)$ a (random) gradient sample following the corrupted distribution and $\widetilde{G}(\theta, \zeta)$ a sample from the true distribution (i.e. such that $\mathbb{E}_{\zeta}[\widetilde{G}(\theta, \zeta)] = \nabla\mathcal{L}(\theta)$), this leads to the following itera-

tion which we call quantile clipped SGD (QC-SGD)

$$\theta_{t+1} = \theta_t - \alpha_{\theta_t}\beta G(\theta_t, \zeta_t) \quad \text{with} \quad \alpha_{\theta_t} = \min\left(1, \frac{Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)}{\|G(\theta_t, \zeta_t)\|}\right), \tag{2.3.1}$$

where $\beta > 0$ is a constant step size and α_{θ_t} is the clipping factor[2] with threshold defined by $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ the p-th quantile of $\|\widetilde{G}(\theta_t, \zeta_t)\|$ for $p \in (0, 1)$. Notice that this is reminiscent of the scalar trimmed mean estimator introduced in Section 2.1.1 which is also based on clipped values.

We show that the above iteration, seen as a Markov chain, converges to an invariant distribution which is concentrated around the optimum $\theta^\star = \arg\min \mathcal{L}(\theta)$ when the latter exists. This is the subject of the following statement drawn from Chapter 5.

Proposition 2.1 (Proposition 5.1 informal)**.** *Assume that $\mathcal{L}$ is Lipschitz-smooth and strongly convex, that the gradient samples $G(\theta, \zeta)$ are η-corrupted and that the gradient errors satisfy*

$$\mathbb{E}\big[\|\varepsilon_\zeta(\theta)\|^q|\theta\big]^{1/q} = \mathbb{E}\big[\|\widetilde{G}(\theta, \zeta) - \nabla\mathcal{L}(\theta)\|^q|\theta\big]^{1/q} \leqslant A_q\|\theta - \theta^\star\| + B_q,$$

for some $q > 1$. Then, for properly chosen step-size β and quantile index p, the QC-SGD Markov chain (2.3.1) *converges geometrically in Total Variation distance to an invariant distribution $\pi_{\beta,p}$ such that we have*

$$\mathbb{E}_{\theta\sim\pi_{\beta,p}}\big[\|\theta - \theta^\star\|^2\big] \leqslant 20\Big(\frac{\eta^{1-1/q}B_q}{\kappa}\Big)^2. \tag{2.3.2}$$

Thus, QC-SGD is robust both to corrupted and heavy-tailed gradients, including the infinite variance case ($q \in (1, 2)$). Moreover, by restricting Iteration (2.3.1) to a bounded set (via projection), we show that the limit distribution $\pi_{\beta,p}$ is sub-Gaussian if $\eta = 0$ (no corruption) and sub-exponential otherwise. Indeed, we argue that $\pi_{\beta,p}$ cannot be sub-Gaussian when corruption is present.

We also derive a similar robust convergence result for smooth and positive objectives $\mathcal{L}$, but at a slower sublinear speed and with the following weaker characterization of the limit distribution $\theta \sim \pi_{\beta,p}$ (see Chapter 5 Proposition 5.3)

$$\mathbb{E}_{\theta\sim\pi_{\beta,p}}\big[\big\|\nabla\mathcal{L}(\theta)\big\|^2\big] \leqslant O\big(\eta^{2-\frac{2}{q}}B_q^2\big). \tag{2.3.3}$$

Namely, the limit distribution is concentrated around a critical point. Note that convexity of $\mathcal{L}$ is not required for this result to hold.

The Markov chain Total Variation convergence statements of Chapter 5 (and Chapter 6) are based on ergodicity results from [313]. In order to prove geometric ergodicity (i.e. convergence at geometric speed), we show, for strongly convex objectives $\mathcal{L}$, that the Markov chain satisfies the following *geometric drift* condition (see [313, Chapter 15]) with respect to the function $V(\theta) = 1 + \|\theta - \theta^\star\|^2$:

$$\mathbb{E}[V(\theta_{t+1})|\theta_t] \leqslant \kappa V(\theta_t) + b \cdot \mathbf{1}_{\theta_t\in\mathcal{C}},$$

where $\kappa < 1$ is a contraction factor, $b < +\infty$ and $\mathcal{C}$ is a bounded neighborhood of $\theta^\star$. That is, the distance to the optimum roughly shrinks by a factor κ after each iteration of (2.3.1) on average for $\theta_t \notin \mathcal{C}$. For smooth objectives, ergodicity with sublinear convergence speed is established by showing the weaker drift property

$$\mathbb{E}[V(\theta_{t+1})|\theta_t] - V(\theta_t) \leqslant -1 + b \cdot \mathbf{1}_{\theta_t\in\mathcal{C}},$$

[2]This new notation for clipping will be more convenient in the sequel.

with V a scaling of the objective $\mathcal{L}$. Namely, $\mathcal{L}$ decreases by a constant amount after each iteration on average outside $\mathcal{C}$.

A key element to derive the concentration properties (2.3.2) and (2.3.3) of the invariant limit distribution in Chapter 5 is to define the gradient bias in Iteration (2.3.1) as

$$\alpha_\theta G(\theta, \zeta) - \overline{\alpha}_\theta \nabla \mathcal{L}(\theta),$$

where $\overline{\alpha}_\theta = \mathbb{E}[\alpha_\theta \mid \theta]$ is the expected value of the clipping factor (see Section 5.9 Lemma 5.2 for a more precise definition). This contrasts with the usual definition taking $\nabla \mathcal{L}(\theta)$ as reference rather than $\overline{\alpha}_\theta \nabla \mathcal{L}(\theta)$ and used in [166, 417]. The difference lies in the fact that [166, 417] assume unbiased gradient samples and use clipping thresholds well above the expected values so that they are rarely exceeded (see discussion following Theorem 2.3 above). In this case, clipping mainly serves to mitigate the effect of a heavy-tailed distribution. In contrast, our clipping strategy is additionally intended to filter out corruption which requires a lower threshold and, in turn, a redefinition of the bias for the analysis.

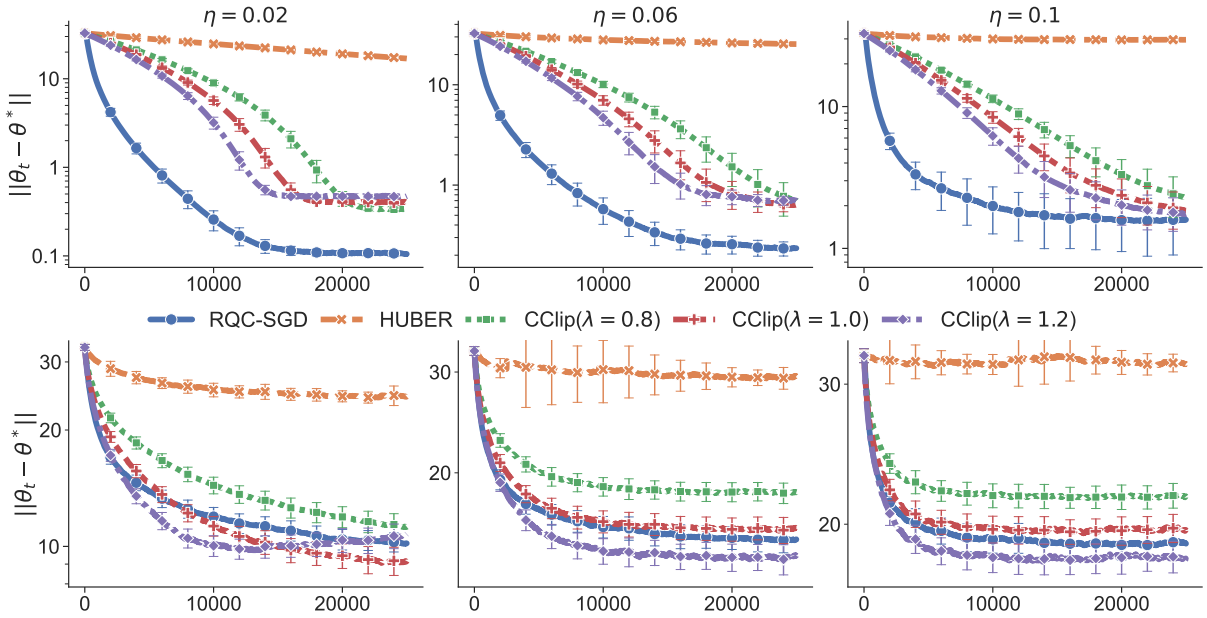

Figure 2.3: Evolution of $\|\theta_t - \theta^\star\|$ on the tasks of linear regression (top row) and logistic regression (bottom row) averaged over 100 runs at increasing corruption levels (error bars represent half the standard deviation). Estimators based on Huber's loss are strongly affected by data corruption. SGD with constant clipping thresholds is robust but slow to converge for linear regression and requires tuning for better final precision. RQC-SGD combines fast convergence with good final precision thanks to its adaptive clipping strategy.

A direct implementation of Iteration (2.3.1) is not possible due to the quantiles $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ being unknown in general. Nevertheless, we implement a variant called "rolling QC-SGD" (RQC-SGD) which replaces $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ by the p-th quantile of a history $(\|G(\theta_{t-s}, \zeta_{t-s})\|)_{s=0}^{S-1}$ of size $S \in \mathbb{N}^*$. We run numerical experiments which show strong robustness properties for RQC-SGD as suggested by theory. The experiments were run on synthetic data for vector mean estimation, linear regression and logistic regression.

Figure 2.3 is drawn from Chapter 5 and compares RQC-SGD to Huber's estimator [204] and clipped SGD with constant thresholds for linear and logistic regression with corrupted data. While Huber's estimator is not robust to corruption, this is true for clipped SGD with constant

thresholds. However, the latter lacks the adaptivity of RQC-SGD and may converge too slowly or yield imprecise final estimates.

2.3.2 Contribution: Convergence and concentration properties of SGD

The Markov chain tools we use in Chapter 5 also allow us to study the properties of conventional SGD with unbiased gradients outside the robust setting. We show in Chapter 6 that the standard SGD iteration (2.1.8) with constant step-size converges to an invariant limit distribution which inherits sub-Gaussian and sub-exponential concentration properties when these hold for the gradient distribution i.e. we have the following statement.

Proposition 2.2 (Proposition 6.2 informal)**.** *Assume that $\mathcal{L}$ is μ-strongly convex and that the gradient errors $G(\theta,\zeta)-\nabla\mathcal{L}(\theta)$ are sub-Gaussian (resp. sub-exponential) with constant K for all θ. Then standard SGD* (2.1.8) *run with constant step-size $\beta_t=\beta$ small enough converges to an invariant distribution π_β which is sub-Gaussian (resp. sub-exponential) with constant $O\big(K\sqrt{\beta/\mu}\big)$.*

Importantly, the sub-Gaussian (resp. sub-exponential) constant of the invariant distribution is modulated by the value of β allowing to obtain high-confidence bounds on the deviations of a final estimate $\widehat{\theta}_T$ after T iterations using a small step-size. Under slightly stronger conditions on the gradient samples' concentration properties, we show that the claims of Proposition 2.2 hold with constant K independent from the dimension which leads to sub-Gaussian concentration bounds for $\widehat{\theta}_T$.

We also derive high-confidence bounds for estimators $\widehat{\theta}$ defined as a tail average of the SGD iteration (Polyak-Ruppert averaging [358, 386]). We obtain the following non-asymptotic result.

Proposition 2.3 (Proposition 6.6 informal)**.** *Assume that $\mathcal{L}$ is Lipschitz-smooth and μ-strongly convex and that the gradient $\nabla\mathcal{L}(\theta)$ is linear. Assume also that the gradient errors $\varepsilon_\zeta(\theta)=G(\theta,\zeta)-\nabla\mathcal{L}(\theta)$ are sub-Gaussian with constant K and satisfy for all θ,θ' :*

$$\mathcal{W}_2^2\big(\mathcal{D}(\varepsilon_\zeta(\theta)),\mathcal{D}(\varepsilon_\zeta(\theta'))\big)\leqslant L_{\mathcal{W}}\|\theta-\theta'\|_2^2 \tag{2.3.4}$$

with $L_{\mathcal{W}}<+\infty$, where $\mathcal{D}(\varepsilon_\zeta(\theta))$ is the distribution of $\varepsilon_\zeta(\theta)$ and $\mathcal{W}_2$ is the Wasserstein-2 distance.

Let $(\theta_t)_{t\geqslant 0}$ be the SGD Markov chain started from $\theta_0\sim\nu$ with step-size β small enough. Then, there exist $\rho<1$ and $M<+\infty$ such that for $\delta>0$ and $n,n_0>0$:

$$\Big\|\frac{1}{n}\sum_{t=n_0+1}^{n_0+n}\theta_t-\theta^\star\Big\|\leqslant\sqrt{\frac{2}{n}\frac{1+\alpha}{1-\alpha}\Big(\alpha_{\mathcal{W}}^{n_0}\mathcal{W}_2^2(\nu,\pi_\beta)+\frac{\beta\sigma^2}{\mu}\Big)}+\frac{4K\sqrt{\beta/\mu}}{1-\alpha_{\mathcal{W}}}\sqrt{\frac{\log(1/\delta)}{n}} \tag{2.3.5}$$

with probability at least $1-\Upsilon(\nu,n_0)\delta$, where

$$\alpha=1-\beta\mu,\quad \alpha_{\mathcal{W}}=\sqrt{\alpha^2+\beta^2L_{\mathcal{W}}}\quad\text{and}\quad\Upsilon(\nu,n_0)=1+M\rho^{n_0}\Big\|\frac{d\nu}{d\pi_\beta}\Big\|_\infty.$$

The derivation of the statement above combines several properties of potential independent interest such as convergence in Total Variation and Wasserstein distance and a geometric decorrelation phenomenon between successive iterates θ_t when the gradient $\nabla\mathcal{L}$ is linear. In particular, we obtain convergence w.r.t. the Wasserstein metric thanks to the assumption of Inequality (2.3.4) which is a more general condition than previously known in the literature as we argue in Section 6.5. Finally, we discuss the application of these results for the examples of linear regression and logistic regression.

2.4 Optimal regret online logistic regression

In this section, we consider the online logistic regression problem which consists in predicting a sequence of binary labels $y_t \in \mathcal{Y} = \{\pm 1\}$ from Euclidean inputs $x_t \in \mathcal{X} = \mathbb{R}^d$ for $t \geqslant 1$ (we focus on the binary label case but an extension to the multi-class case is possible).

In the online setting, the instances $(x_t, y_t)_{t \geqslant 1}$ arrive sequentially and the *agent* makes a prediction $\widehat{y}_t$ at each turn which is evaluated by the logistic loss

$$\ell(\widehat{y}_t, y_t) = \log(1 + \exp(-\widehat{y}_t y_t)). \tag{2.4.1}$$

More precisely, the two following steps occur for each $t \geqslant 1$:

- The agent receives a sample x_t from the environment and uses it along with the history up to this point $H_t = \{(x_1, y_1), \dots, (x_{t-1}, y_{t-1})\}$ in order to make a prediction $\widehat{y}_t$ of y_t.
- Then, the environment reveals the value of the label y_t and the agent incurs the logistic loss $\ell(\widehat{y}_t, y_t)$.

The overall performance of the agent's predictions is evaluated through the notion of regret which is conventionally defined with respect to the best linear predictor in hindsight within a comparison class $\Theta \subset \mathbb{R}^d$. This corresponds to the definition below after n turns

$$R_n := \sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \min_{\theta \in \Theta} \sum_{t=1}^{n} \ell(\theta^\top x_t, y_t). \tag{2.4.2}$$

The classical setting for regret minimization considers bounded features $\|x_t\| \leqslant R$ and fixes the comparison class as the Euclidean ball of radius B i.e. $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\| \leqslant B\}$.

Pay attention to the fact that this online logistic regression task must not be confused with the robust learning problems we discussed previously since they differ in many points:

- The sequence (x_t, y_t) is purely arbitrary in this case and does not follow any probability distribution.
- The x_ts cannot be heavy-tailed since they are assumed bounded.
- Although the sequence (x_t, y_t) may be crafted by an adversary, no sample is considered fake or corrupted since they all count for the regret (2.4.2).

The quality of a learning strategy for regret minimization is measured by the asymptotic rate it achieves in terms of the number of rounds n. It is important to design strategies with at most sublinear regret $R_n = o(n)$ since, in the contrary case, a linear regret means that the agent fails to learn the task by incurring at least a constant loss per round on average.

Multiple regret minimization strategies were proposed in the literature with progressively improved regret bounds by leveraging the properties of the logistic loss (2.4.1). For instance, Online Gradient Descent (OGD) [460] achieves $O(BR\sqrt{n})$ regret using the convexity of ℓ and the Online Newton Step (ONS) [184] obtains an $O(de^{BR}\log(n))$ bound by taking advantage of strong convexity.

An important breakthrough was made by [149] who showed that logarithmic regret can be achieved without the exponential factor e^{BR} appearing in the bound satisfied by ONS. This is made possible using an *improper* algorithm which makes predictions as $\widehat{y}_t = \theta_t^\top x_t$ where the linear parameter θ_t is computed using the knowledge of x_t in addition to the history samples $(x_s, y_s)_{s=1}^{t-1}$. Thanks to a Bayesian prediction approach, [149] obtains an algorithm with regret bound $O(d\log(BRn))$ which remains unmatched as of this writing. However, this algorithm cannot be used in practice due to its prohibitive complexity of order $O((BR)^6 n^{12}(BRn + d)^{12})$.

Subsequent works [218, 4] designed more practical improper algorithms which achieve logarithmic regret at low computational costs of order $O(nd^2)$. However, all these algorithms have regret of order $O(dBR\log(n))$ which is suboptimal by a factor BR compared to [149].

In Appendix A, we discuss an efficient algorithm which may provide a solution bridging this gap. The latter is based on the Sample Minmax Predictor (SMP) which was presented in [325] and shown to reach the following excess risk upper bound on batch logistic regression with n samples (see definition (2.2.2) in Section 2.2 above)

$$\mathcal{L}(\widehat{\theta}_{\mathrm{SMP}}) - \min_{\theta\in\Theta}\mathcal{L}(\theta) \leqslant \frac{ed + B^2R^2}{n}. \tag{2.4.3}$$

SMP is a non Bayesian estimator which can be computed by solving two batch logistic regression problems making it far more efficient than Bayesian algorithms which require using costly MCMC integration methods as in [149]. Based on the bound (2.4.3) on the batch excess risk, an online version of SMP is likely to achieve regret $O((d + B^2R^2)\log(n))$ which would match the performance of [149] since it can be argued that $BR \lesssim \sqrt{d}$ (see [326, Remark 2]).

Online SMP (OSMP) is introduced in Appendix A by adapting ideas from [326] to the online setting. In addition, we present a preliminary regret analysis based on the strong convexity of the logistic loss in order to show logarithmic regret. Considering that it is still unclear how the target regret bound $O((d + B^2R^2)\log(n))$ can be proved for OSMP, we discuss the underlying difficulty in the analysis and how it may be overcome.

2.5 From random trees to WildWood

As mentioned before, most machine learning problems consist in choosing a good predictor $\phi : \mathcal{X} \to \widehat{\mathcal{Y}}$ among a class of candidate functions Φ based on sample data. We have seen that linear models correspond to setting Φ as the set of linear functions. In this section, we consider random forests which correspond to taking Φ as the class of piecewise constant functions. It is easy to see that this class has strong expressive power in that it allows to approximate (for example) any bounded continuous function up to arbitrary precision using a fine partition of the space $\mathcal{X}$.

Indeed, a piecewise constant function ϕ can be defined by a partition of the space into disjoint regions $\mathcal{X} = \cup_j C_j$ and values $v_j \in \mathcal{Y}$ taken by ϕ on each region so that we have $\phi(x) = \sum_j \mathbf{1}_{x\in C_j} v_j$. Contrary to linear functions, the class of piecewise constant functions has a non parametric character which excludes gradient based optimization methods for the training task. Instead, piecewise constant models are trained using a recursive *tree growth* method which we describe below.

The tree growth algorithm

We consider a Euclidean space $\mathcal{X} = \mathbb{R}^d$. Starting from a trivial partition $\mathcal{P}_0 = \{C_{\texttt{root}}\} = \{\mathbb{R}^d\}$ at $t = 0$, we execute the following steps:

1. Choose a cell $C_{\mathbf{v}} \in \mathcal{P}_t$ among the current partition.
2. Choose a coordinate (or *feature*) $j \in \{1, \dots, d\}$ and a threshold $s \in \mathbb{R}$.
3. Define subcells $C_{\mathbf{v}} = C_{\mathbf{v}0} \cup C_{\mathbf{v}1}$ where $C_{\mathbf{v}0} = \{x \in C_{\mathbf{v}} : x_j \leqslant s\}$ and $C_{\mathbf{v}1} = C_{\mathbf{v}} \setminus C_{\mathbf{v}0}$.
4. Update partition as $\mathcal{P}_{t+1} = (\mathcal{P}_t \setminus C_{\mathbf{v}}) \cup \{C_{\mathbf{v}0}, C_{\mathbf{v}1}\}$.

After one iteration, we can define the model prediction by setting constant values v_0, v_1 on each member of the partition $\mathcal{P}_1 = \{C_0, C_1\}$. To obtain a fine-grained model, the above steps can be

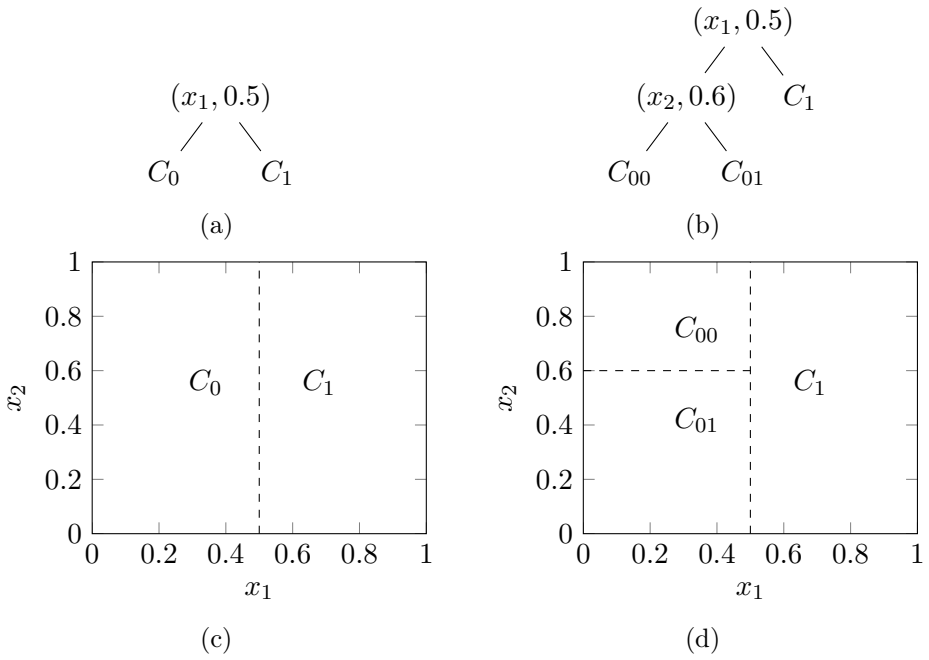

Figure 2.4: Illustration of two initial steps of the tree growth algorithm in dimension 2 where a first split is made along coordinate x_1 with threshold 0.5 yielding $\mathbb{R}^d = C_0 \cup C_1$ (left). The cell C_0 is then split again along coordinate x_2 with threshold 0.6 (right).

recursively repeated in order to sharpen the partition $\mathcal{P}_t$ until a satisfactory precision level is reached.

The space partition yielded by this process can be represented by a hierarchical tree structure. Figure 2.4 provides an example illustration of two initial steps of this algorithm for $d = 2$. The associated tree structure encodes the information about the splits defining the partition in its interior nodes (feature along which splits are made and their thresholds) while the leaves represent the regions (also called cells) constituting the partition.

Ideally, given a training sample $(X_i, Y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, the goal would be to find the best prediction model $\phi^\star$ of this kind which minimizes an objective like (2.2.1). However, the problem of determining the splits defining the tree structure of such a $\phi^\star$ turns out to be NP-Hard [256]. To avoid this difficulty, the tree growth algorithm is used to greedily train a proxy for the optimal predictor one split at a time.

Tree growth as a learning method

In order to obtain a good predictor for a dataset $(X_i, Y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ via tree growth, each split is made so as to maximize the homogeneity of subcells in terms of the labels Y_i of the samples X_i they contain. This is done by choosing split parameters j and s in item 2. above minimizing heuristic subcell impurity criteria. Given a function $\mathcal{I}$ measuring the impurity of a probability distribution and denoting δ_x the Dirac measure at x, this corresponds to the following optimization

$$\underset{s\in\mathbb{R}, j\in\{1,\dots,d\}}{\arg\min} \quad \mathcal{I}\big(\mathcal{D}_0\big) + \mathcal{I}\big(\mathcal{D}_1\big) \quad \text{where} \quad \mathcal{D}_k = \frac{1}{|\{i : X_i \in C_{\mathbf{v}k}\}|} \sum_{i : X_i \in C_{\mathbf{v}k}} \delta_{Y_i}. \tag{2.5.1}$$

For classification problems (the Y_is are discrete labels), a common choice for $\mathcal{I}$ is the entropy function whereas for regression problems (the Y_is are real values) $\mathcal{I}$ can be defined as the variance.

In this fashion, the space $\mathcal{X}$ is partitioned into neighborhoods containing samples X_i with similar labels Y_i allowing to define a high accuracy predictor [55, 423]. In addition to this heuristic, the procedure is randomized using *feature subsampling* which consists in limiting the candidate features for each split to a randomly sampled subset of the whole features i.e. $\{1, \ldots, d\}$ is replaced with a random subset in item 2. above and in (2.5.1).

The final structure is called a *binary decision tree* [370, 56] and enables an efficient implementation of the associated predictor by navigating the tree structure upon a query $X \in \mathcal{X}$. The term *random tree* [51, 53] is also often used and refers to the randomization of tree growth via feature subsampling.

Depending on the size of the training sample $(X_i, Y_i)_{i=1}^n$, the space partition can be refined by continuing the tree growth process and splitting leaf nodes in the tree structure until a stopping criterion is satisfied. In the extreme case, the process may go on until the partition is fine enough so that each leaf node only contains a single sample X_i of the training dataset. However, this often leads to overfitting and it is better to avoid pushing tree growth to this point for better performance.

Tree-based Ensemble methods

Tree-based decision models appeared as a machine learning method in the 1960s [322, 312, 370] before being popularized by the in-depth practical and theoretical study of [54]. Such models attracted interest thanks to their simplicity, intuitiveness and ease of use. However, their strong expressive power and the noisy nature of data in practice make them prone to overfitting which leads to poor predictive performance. This problem was addressed using *ensemble methods* which roughly consist in training multiple instances of a simple model and combining their predictions within an ensemble model to achieve better results. Two big families of ensemble models are worth mentioning:

- Boosting: presented in [392, 152], boosting consists in incrementally building an ensemble model by repeatedly adding simple models which are trained to correct the current ensemble's mistakes. Tree-based boosting uses shallow trees as base models and applies split finding criteria which are directly related to the optimization of an empirical objective (2.2.1). Popular implementations are available in a number of libraries such as XGBoost [87], LightGBM [231] or CatBoost [367].
- Bagging: or Bootstrap Aggregation [51] simultaneously trains models on bootstrap datasets and combines them into an overall ensemble model which makes predictions based on majority rules using the submodels' outputs. In tree-based bagging, trees are grown using impurity criteria and their depth is controlled by heuristics like a maximal depth limit or minimal sample count within cells to avoid overfitting.

We take a closer look at Bagging. The bootstrap datasets it uses are obtained by sampling n elements (with n the sample count) of the training set with replacement for each submodel. As a result, each submodel is trained on a subset of the training sample called *in-the-bag* samples while the remaining *out-of-bag* samples are ignored. This bootstrap method aims to create a degree of independence between the submodels and average the noise out of their predictions through the subsequent aggregation.

Tree-based bagging with extra feature subsampling corresponds to the famous Random Forest algorithm [188, 52], a reference implementation of which is provided in the Python `scikit-learn` library [355, 281]. In Chapter 7, we introduce a new algorithm called WildWood which improves upon the classical random forest algorithm by introducing an efficient within-tree regularization mechanism.

2.5.1 Contribution: WildWood: a new Random Forest algorithm

Let $\mathcal{T}$ be a binary tree obtained through the previous tree growth procedure on a dataset $(X_i, Y_i)_{i=1}^n$, and denote $\phi_{\mathcal{T}} : \mathcal{X} \to \widehat{\mathcal{Y}}$ its prediction function.

Recall that the performance of $\phi_{\mathcal{T}}$ is often negatively affected by $\mathcal{T}$ being grown to excessive depth since, beyond a certain precision scale, the associated space partition ceases to capture relevant information on the data distribution and starts to follow uninformative noise instead. A solution to this problem is to use a *pruning* (or subtree) $T \subset \mathcal{T}$ of the full tree $\mathcal{T}$ which drops part of the total set of splits defining it (for example, the left tree on Figure 2.4 is a subtree of the right one). For a well chosen subtree, this yields a better predictor ϕ_T based on a simpler space partition minimizing the effect of noise while preserving good predictive power. However, considering that the number of possible subtrees $T \subset \mathcal{T}$ is highly combinatorial, an algorithm which simply enumerates them all in order to determine the one with the best predictor ϕ_T would be excessively inefficient.

To bypass this difficulty, WildWood defines the predictor $\widehat{\phi}_{\mathcal{T}}$ of a tree $\mathcal{T}$ as the following exponentially weighted average

$$\widehat{\phi}_{\mathcal{T}}(x) = \frac{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T} \phi_T(x)}{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T}} \quad \text{with} \quad L_T = \sum_{i \in I_{\texttt{oob}}} \ell(\phi_T(X_i), Y_i) \quad \text{and} \quad \pi(T) = 2^{-\|T\|}, \tag{2.5.2}$$

where the sum is over all possible subtrees $T \subset \mathcal{T}$, L_T is the loss of ϕ_T on the out-of-bag sample $I_{\texttt{oob}}$ of $\mathcal{T}$, $\|T\|$ measures the complexity of T and $\eta > 0$ is a temperature parameter.

The exponential average (2.5.2) weighs subtrees $T \subset \mathcal{T}$ according to their predictive performance on the out-of-bag sample and their simplicity (shallower subtrees are favored). The computation of (2.5.2) is implemented thanks to an adaptation of the Context Tree Weighting algorithm [441, 440] which allows to avoid the direct enumeration of the subtrees $T \subset \mathcal{T}$ and maintain the same complexity as the original random forest algorithm.

The implementation of WildWood (which is publicly available on GitHub[3]) also includes the histogram splitting strategy used by boosting libraries [87, 231, 367]. The latter consists in quantizing the values of the features X_i in order to speed-up the solution of the split finding optimizations (2.5.1) allowing for faster model training.

The aggregation mechanism defined by (2.5.2) allows to obtain a model whose performance approaches that of the best subtree $T \subset \mathcal{T}$ in terms of prediction accuracy i.e. the following result holds.

Theorem 2.5 (Theorem 7.2 informal)**.** *Assume that the loss function ℓ is η-exp-concave. Then, the prediction function $\widehat{\phi}_{\mathcal{T}}$ given by* (2.5.2) *satisfies the oracle inequality*

$$\frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{\phi}_{\mathcal{T}}(X_i), Y_i) \leqslant \inf_{T \subset \mathcal{T}} \left\{ \frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\phi_T(X_i), Y_i) + \frac{\log 2}{\eta} \frac{\|T\|}{n_{\texttt{oob}} + 1} \right\},$$

where $\|T\|$ is the number of nodes in T minus its number of leaves that are also leaves of $\mathcal{T}$ and $n_{\texttt{oob}} = |I_{\texttt{oob}}|$.

We assess the performances of WildWood in practice and compare it to classical random forests and boosting algorithms through extensive numerical experiments which confirm its predictive gains on publicly available classification datasets. We also show that WildWood achieves good results with smaller lightweight models which train faster and result in simpler decision boundaries for classification problems and better interpretability.

[3] `https://github.com/pyensemble/wildwood.git`

This is illustrated by Figures 2.5 and 2.6 which are drawn from Chapter 7. The former shows that WildWood obtains better models using fewer trees than previous random forest algorithms while the latter illustrates the regularization effect of the aggregation mechanism on the decision function of each tree on a toy classification problem. Aggregation leads to simpler decision boundaries which minimize the effect of noise and avoid over-fitting.

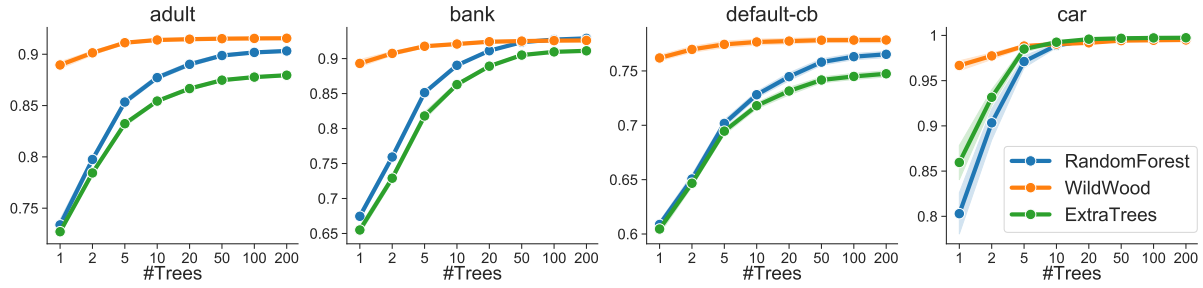

Figure 2.5: AUC scores for classification on a few datasets for different random forest algorithms with increasing numbers of trees.

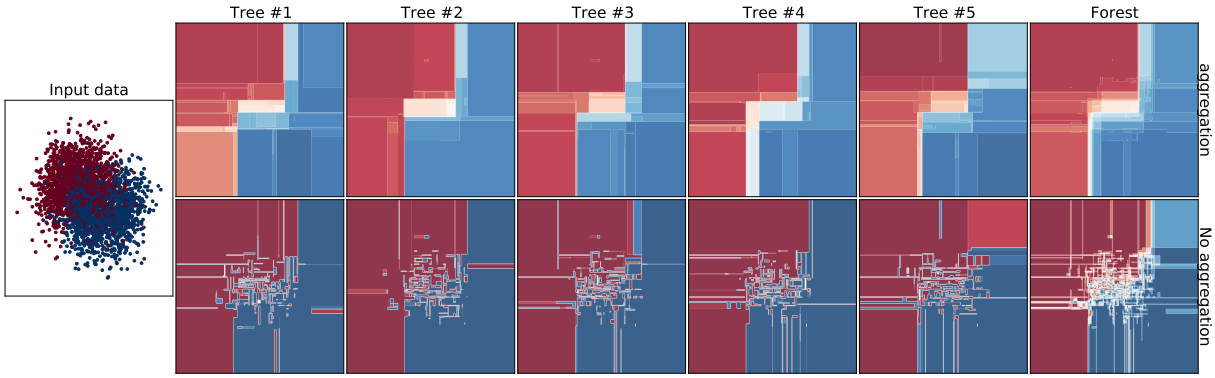

Figure 2.6: WildWood decision functions illustrated on a toy dataset (left) with subtrees aggregation (top) and without it (bottom). Subtrees aggregation improves trees predictions, as illustrated by smoother decision functions in the top compared with the bottom, improving overall predictions of the forest (last column).

2.6 List of articles

We provide below the list of research papers written during this thesis and their status.

- S. Gaïffas, I. Merad and Y. Yu. "WildWood: a new Random Forest algorithm". In : *IEEE Transactions on Information Theory* (2023).
- I. Merad and S. Gaïffas. "Robust supervised learning with coordinate gradient descent". Submitted.
- I. Merad and S. Gaïffas. "Robust methods for high-dimensional linear learning". In : *Journal of Machine Learning Research* (2023).
- I. Merad and S. Gaïffas. "Robust stochastic optimization via gradient quantile clipping". Submitted.
- I. Merad and S. Gaïffas. "Convergence and concentration properties of constant step-size SGD through Markov chains". Submitted.

Chapter 3

Robust Learning with Coordinate Gradient Descent

This chapter is based on the article [153] in collaboration with Stéphane Gaïffas.

Contents

Abstract

This paper considers the problem of supervised learning with linear methods when both features and labels can be corrupted, either in the form of heavy tailed data and/or corrupted rows. We introduce a combination of coordinate gradient descent as a learning algorithm together with robust estimators of the partial derivatives. This leads to robust statistical learning methods that have a numerical complexity *nearly identical* to non-robust ones based on empirical risk minimization. The main idea is simple: while robust learning with gradient descent requires the computational cost of robustly estimating the whole gradient to update all parameters, a parameter can be updated immediately using a robust estimator of a single partial derivative in coordinate gradient descent. We prove upper bounds on the generalization error of the algorithms derived from this idea, that control both the optimization and statistical errors with and without a strong convexity assumption of the risk. Finally, we propose an efficient implementation of this approach in a new `Python` library called `linlearn`, and demonstrate through extensive numerical experiments that our approach introduces a new interesting compromise between robustness, statistical performance and numerical efficiency for this problem.

3.1 Introduction

Outliers and heavy tailed data are a fundamental problem in supervised learning. As explained by [182], an outlier is a sample that differs from the data's "global picture". A rule-of-thumb is that a typical data set may contain between 1% and 10% of outliers [175], or even more than that depending on the considered application. For instance, the inherently complex and random nature of users' web browsing makes web-marketing data sets contain a significant proportion of outliers and have heavy-tailed distributions [171]. Statistical handling of outliers was already considered in the early 50's [125, 169] and motivated in the 70's the development of *robust statistics* [206, 205].

Setting. In this paper, we consider the problem of large-scale supervised learning, where we observe possibly corrupted samples $(X_i, Y_i)_{i=1}^n$ of a random variable $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with distribution P, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space and $\mathcal{Y} \subset \mathbb{R}$ is the set of label values. We focus on linear methods, where the learning task corresponds to finding an approximation of an optimal parameter

$$\theta^\star \in \operatorname*{arg\,min}_{\theta \in \Theta} R(\theta) \quad \text{where} \quad R(\theta) := \mathbb{E}\big[\ell(X^\top \theta, Y)\big], \tag{3.1.1}$$

where Θ is a convex compact subset of $\mathbb{R}^d$ with diameter Δ containing the origin and $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function satisfying the following. We denote $\ell'(z, y) := \partial \ell(z, y)/\partial z$.

Assumption 3.1. *The loss $z \mapsto \ell(z, y)$ is convex for any $y \in \mathcal{Y}$, differentiable and γ-smooth in the sense that $|\ell'(z, y) - \ell'(z', y)| \leqslant \gamma |z - z'|$ for all $z, z' \in \mathbb{R}$ and $y \in \mathcal{Y}$. Moreover, there exists $q \in [1, 2]$, which we will call the asymptotic polynomial degree, and positive constants $C_{\ell,1}, C_{\ell,2}, C'_{\ell,1}$ and $C'_{\ell,2}$ such that*

$$|\ell(z, y)| \leqslant C_{\ell,1} + C_{\ell,2}|z - y|^q \quad \text{and} \quad |\ell'(z, y)| \leqslant C'_{\ell,1} + C'_{\ell,2}|z - y|^{q-1}$$

for all $z \in \mathbb{R}$ and $y \in \mathcal{Y}$.

Note that Assumption 3.1 holds for the majority of loss functions used both for regression and classification, such as the square loss $\ell(z,y) = (z-y)^2/2$ with $q=2$ or the Huber loss [203] $\ell(z,y) = r_\tau(z-y)$ for $z,y \in \mathbb{R}$ with $\gamma = 1$ and $q = 1$, where $r_\tau(u) = \frac{1}{2}u^2\mathbf{1}_{|u|\leqslant\tau} + \tau(|u| - \frac{1}{2}\tau)\mathbf{1}_{|u|>\tau}$ with $\tau > 0$ and the logistic loss $\ell(z,y) = \log(1+e^{-yz})$ for $z \in \mathbb{R}$ and $y \in \{-1,1\}$ with $\gamma = 1/4$ and $q = 1$. We will see shortly that a smaller degree q associated to the loss entails looser requirements on the data distribution. If P were known, one could approximate $\theta^\star$ using a first-order optimization algorithm such as *gradient descent* (GD), using iterations of the form

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla R(\theta_t) \quad \text{with} \quad \nabla R(\theta) = \mathbb{E}[\ell'(X^\top\theta, Y)X] \tag{3.1.2}$$

for $t = 1, 2, \ldots$ where $\eta > 0$ is a learning rate.

Empirical risk minimization. With P unknown, most supervised learning algorithms rely on *empirical risk minimization* (ERM) [427, 157], which requires (a) the fact that samples are independent and with the same distribution P and (b) that P has sub-Gaussian tails, as explained below. Such assumptions are hardly ever met in practice, and entail implicitly that, for real-world applications, the construction of a training data set requires involved data preparation, such as outlier detection and removal, data normalization and other issues related to feature engineering [457, 247]. An *implicit*[1] ERM estimator of $\theta^\star$ is a minimizer of the empirical risk R_n given by

$$\widehat{\theta}_n^{\texttt{erm}} \in \operatorname*{arg\,min}_{\theta\in\Theta} R_n(\theta) \quad \text{where} \quad R_n(\theta) := \frac{1}{n}\sum_{i=1}^n \ell(X_i^\top\theta, Y_i), \tag{3.1.3}$$

for which one can prove sub-Gaussian deviation bounds under strong hypotheses such as boundedness of ℓ or sub-Gaussian concentration [298, 263]. In the general case, ERM leads to poor estimations of $\theta^\star$ whenever (a) and/or (b) are not met, corresponding to situations where (a) the data set contains outliers and (b) the data distribution has heavy tails. This fact motivated the theory of robust statistics [203, 205, 176, 175, 421]. The poor performance of ERM stems from the loose deviation bounds of the empirical mean estimator. Indeed, as explained by [73] for the estimation of the expectation of a real random variable, the Chebyshev inequality provably provides the best concentration bound for the empirical mean estimator in the general case, so that the error is $\Omega(1/\sqrt{n\delta})$ for a confidence $1-\delta$. Gradient Descent (GD) combined with ERM leads to an *explicit* algorithm using iterations (3.1.2) with gradients estimated by an average over the samples

$$\widehat{\nabla}^{\texttt{erm}} R(\theta) := \nabla R_n(\theta) = \frac{1}{n}\sum_{i=1}^n \ell'(X_i^\top\theta, Y_i)X_i, \tag{3.1.4}$$

which is, as explained above, a poor estimator of $\nabla R(\theta)$ beyond (a) and (b).

Robust gradient descent. A growing literature about robust GD estimators [364, 273, 192, 162] suggests to perform GD iterations with $\widehat{\nabla}^{\texttt{erm}} R(\theta)$ replaced by some robust estimator of $\nabla R(\theta)$. An implicit estimator is considered by [262], based on the minimization of a robust estimate of the risk objective using median-of-means. Robust estimators of $\nabla R(\theta)$ can be built using several approaches including geometric median-of-means [364]; robust coordinate-wise estimators [191] based on a modification of [73]; coordinate-wise median-of-means or trimmed means [273] or robust vector means through projection and truncation [364]. Other works achieve robustness by performing standard training on disjoint subsets of data and aggregating the re-

[1] By *implicit*, we mean defined as the arg min of some functional, as opposed to the *explicit* iterations of an optimization algorithm: an implicit estimator differs from the exact algorithm applied on the data, while an *explicit* algorithm does not.

sulting estimators into a robust one [316, 57]. We discuss such alternative methods in more details in Section 3.4 below.

These procedures based on GD require to run *costly* subroutines (at the exception of [262, 162]) that induce a considerable computational overhead compared to the non-robust approach based on ERM. The aim of this paper is to introduce *robust* and *explicit* learning algorithms, with performance guarantees under weak assumptions on $(X_i, Y_i)_{i=1}^n$, that have a computational cost *comparable* to the non-robust ERM approach. As explained in Section 3.2 below, the main idea is to combine *coordinate gradient descent* with robust estimators of the partial derivatives $\partial R(\theta)/\partial\theta_j$, that are scalar (univariate) functionals of the unknown distribution P.

We denote $|A|$ as the cardinality of a finite set A and use the notation $[\![k]\!] = \{1, \dots, k\}$ for any integer $k \in \mathbb{N} \setminus \{0\}$. We denote x^j as the j-th coordinate of a vector x. We will work under the following assumption.

Assumption 3.2. *The indices of the training samples $[\![n]\!]$ can be divided into two disjoint subsets $[\![n]\!] = \mathcal{I} \cup \mathcal{O}$ of* outliers $\mathcal{O}$ *and* inliers $\mathcal{I}$ *for which we assume the following*: (*a*) *we have $|\mathcal{I}| > |\mathcal{O}|$*; (*b*) *the pairs $(X_i, Y_i)_{i\in\mathcal{I}}$ are i.i.d with distribution P and the outliers $(X_i, Y_i)_{i\in\mathcal{O}}$ are arbitrary*; (*c*) *there is $\alpha \in (0, 1]$ such that*

$$\mathbb{E}\big[|X^j|^{\max(2,q(\alpha+1))}\big] < +\infty, \quad \mathbb{E}\big[|Y^{q-1}X^j|^{1+\alpha}\big] < +\infty \quad \textit{and} \quad \mathbb{E}\big[|Y|^q\big] < +\infty \tag{3.1.5}$$

for any $j \in [\![d]\!]$ where $q \in [1, 2]$ is the loss' asymptotic polynomial degree from Assumption 3.1.

Assumption 3.2 is purposely vague about $|\mathcal{I}|$ and $|\mathcal{O}|$ and the value of $\alpha \in (0, 1]$. Indeed, conditions on $|\mathcal{O}|$ and α will depend on the considered robust estimator of the partial derivatives, as explained in Section 3.3 below, including theoretical guarantees with $\alpha < 1$ and cases with $\mathbb{E}[Y^2] = +\infty$ (for the Huber loss for instance). The existence of a second moment for X is indispensable for the objective $R(\theta)$ to be Lipschitz-smooth, see Section 3.2.2 below.

Square loss. For the square loss we have $q = 2$ and $\mathbb{E}[Y^2] < +\infty$ is required for the risk $R(\theta)$ and its partial derivatives to be well-defined. Note that we have $\mathbb{E}[|\ell'(X^\top\theta, Y)X^j|^{1+\alpha}] = \mathbb{E}[|YX^j|^{1+\alpha}]$ for $\theta = 0 \in \Theta$, which makes (3.1.5) somewhat minimal in order to ensure the existence of the moment we need for the loss derivative for all $\theta \in \Theta$.

Huber loss. For the Huber loss, we have $q = 1$ and the only requirement on Y is $\mathbb{E}|Y| < +\infty$ and we have $\max(2, q(\alpha+1)) = 2$ ensuring that $\mathbb{E}[|X^j|^2] < +\infty$, a requirement for the Lipschitz-smoothness of $R(\theta)$, as detailed in Section 3.2.2.

Logistic loss. For the logistic loss we have $|Y| \leqslant 1$ and $q = 1$ so that the only assumption is once again $\mathbb{E}[|X^j|^2] < +\infty$.

Main contributions. This paper introduces a new interesting compromise between robustness, statistical performance and numerical efficiency for supervised learning with linear methods through the following main contributions:

- We introduce a new approach for robust supervised learning with linear methods by combining coordinate gradient descent (CGD) with robust estimators of the partial derivatives used in its iterations (Section 3.2). This novel and intuitive idea turns out to be very effective experimentally (see Section 3.6) and amenable to an in-depth theoretical analysis (see Section 3.2.2 for guarantees under strong convexity and Section 3.5 without it).
- We consider state-of-the-art robust estimators of the partial derivatives (Section 3.3) and provide theoretical guarantees for CGD combined with each of them. For some robust estimators, our analysis requires only weak moments (allowing $\mathbb{E}[Y^2] = +\infty$ in some cases)

together with strong corruption (large $|\mathcal{O}|$) which lets our results apply to very general settings with minimal assumptions compared to the relevant literature. We provide guarantees for several variants of CGD namely random uniform sampling, importance sampling and deterministic sampling of the coordinates (Section 3.2.2).

- We perform extensive numerical experiments, both for regression and classification on several data sets (Section 3.6). We compare many combinations of gradient descent, coordinate gradient descent and robust estimators of the gradients and partial derivatives. Some of these combinations correspond to state-of-the-art algorithms [260, 191, 364], and we also consider several additional baselines such as Huber regression [349], classification with the modified Huber loss [452], Least Absolute Deviation (LAD) [139] and RANSAC [148]. We carry out an in-depth experimental comparison of state-of-the-art robust methods for supervised linear learning both in terms of statistical precision and numerical complexity. We thereby demonstrate the outstanding performance of our methods on both aspects.

- All the algorithms studied and compared in the paper are made easily accessible in a few lines of code through a new `Python` library called `linlearn`, open-sourced under the BSD-3 License on `GitHub` and available here[2]. This library follows the API conventions of `scikit-learn` [355].

3.2 Robust coordinate gradient descent

CGD is well-known for its efficiency and fast convergence properties based on both theoretical and practical studies [341, 406, 159, 443] and is the de-facto standard optimization algorithm used in many machine learning libraries. In this paper, we suggest to use CGD with robust estimators $\widehat{g}_j(\theta)$ of the partial derivatives $g_j(\theta) := \partial R(\theta)/\partial\theta_j \in \mathbb{R}$ of the true risk given by Equation (3.1.1), several robust estimators $\widehat{g}_j(\theta)$ are described in Section 3.3 below.

3.2.1 Iterations

At iteration $t+1$, given the current iterate $\theta^{(t)}$, CGD proceeds as follows. It chooses a coordinate $j_t \in [\![d]\!]$ (several sampling mechanisms are possible, as explained below) and the parameter is updated using

$$\begin{cases} \theta_j^{(t+1)} \leftarrow \theta_j^{(t)} - \beta_j \widehat{g}_j(\theta^{(t)}) & \text{if } j = j_t \\ \theta_j^{(t+1)} \leftarrow \theta_j^{(t)} & \text{otherwise} \end{cases} \tag{3.2.1}$$

for all $j \in [\![d]\!]$, where $\beta_j > 0$ is a step-size for coordinate j. A *single* coordinate is updated at each iteration of CGD, and we will designate d iterations of CGD as a *cycle*. The CGD procedure is summarized in Algorithm 1 below, where we denote by $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ the features matrix with rows $X_1^\top, \dots, X_n^\top$ and where $\boldsymbol{X}_\bullet^j \in \mathbb{R}^n$ stands for its j-th column.

A simple choice for the distribution p is the uniform distribution over $[\![d]\!]$, but improved convergence rates can be achieved using importance sampling, as explained in Theorem 3.1 below, where the choice of the step-sizes $(\beta_j)_{j=1}^d$ is described as well. The partial derivatives estimators $(\widehat{g}_j(\cdot))_{j=1}^d$ described in Section 3.3 will determine the statistical error of this explicit learning procedure. Note that line 6 of Algorithm 1 uses the fact that

$$I^{(t+1)} = \boldsymbol{X}\theta^{(t+1)} = \sum_{j\neq j_t} \boldsymbol{X}_\bullet^j \theta_j^{(t+1)} + \boldsymbol{X}_\bullet^{j_t} \theta_{j_t}^{(t+1)}$$

[2] `https://github.com/linlearn/linlearn`

Algorithm 1 Robust coordinate gradient descent

1: **Inputs:** Learning rates $\beta_1, \dots, \beta_d > 0$; estimators $(\widehat{g}_j(\cdot))_{j=1}^d$ of the partial derivatives; initial parameter $\theta^{(0)}$; distribution $p = [p_1 \cdots p_d]$ over $[\![d]\!]$ and number of iterations T.
2: Compute $I^{(0)} \leftarrow \boldsymbol{X}\theta^{(0)}$
3: **for** $t = 0, \dots, T-1$ **do**
4: Sample a coordinate $j_t \in \{1, \dots, d\}$ with distribution p independently of $j_1, \dots, j_{t-1}$
5: Compute $\widehat{g}_{j_t}(\theta^{(t)})$ using $I^{(t)}$ and put $D^{(t)} \leftarrow -\beta_{j_t}\widehat{g}_{j_t}(\theta^{(t)})$
6: Update the inner products using $I^{(t+1)} \leftarrow I^{(t)} + \boldsymbol{X}_\bullet^{j_t} D^{(t)}$
7: Apply the update $\theta_{j_t}^{(t+1)} \leftarrow \theta_{j_t}^{(t)} + D^{(t)}$
8: **end for**
9: **return** The last iterate $\theta^{(T)}$

$$= \sum_{j \neq j_t} \boldsymbol{X}_\bullet^j \theta_j^{(t)} + \boldsymbol{X}_\bullet^{j_t}\big(\theta_{j_t}^{(t)} + D^{(t)}\big) = I^{(t)} + \boldsymbol{X}_\bullet^{j_t} D^{(t)}.$$

This computation has complexity $O(n)$, and we will see in Section 3.3 that the complexity of the considered robust estimators $\widehat{g}_{j_t}(\theta^{(t)})$ at line 5 is also $O(n)$, so that the overall complexity of one iteration of robust CGD is also $O(n)$. This makes the complexity of one cycle of robust CGD $O(nd)$, which corresponds to the complexity of *one iteration of GD using the non-robust estimator* $\widehat{\nabla}^{\mathtt{erm}} R(\theta)$, see Equation (3.1.4). A more precise study of these complexities is discussed in Section 3.3, see in particular Table 3.1. Moreover, we will see experimentally in Section 3.6 that our approach is indeed very competitive in terms of the compromise between computational cost and statistical accuracy, compared to all the considered baselines.

Comparison with robust gradient descent. Robust estimators of the expectation of a random vector (such as the geometric median by [316]) require to solve a d-dimensional optimization problem at each iteration step while, in the univariate case, a robust estimator of the expectation can be obtained at a cost comparable to that of an ordinary empirical average. Of course, one can combine such univariate estimators into a full gradient: this is considered for instance by [191, 192, 193, 273, 420], but this approach accumulates errors into the overall estimation of the gradient. This paper introduces an alternative method, where univariate estimators of the partial derivatives are used *immediately* to update the current iterate. We believe that this is the main benefit of using CGD in this context: even if our theoretical analysis hardly explains this, our understanding is that one iteration of CGD is impacted by the estimator error of a *single* partial derivative, that can be corrected straight away in the next iteration, while one iteration of GD is impacted by the accumulated estimation errors of the d partial derivatives, when using d univariate estimators for efficiency, instead of a computationally involved d-dimensional estimator (such as geometric median).

3.2.2 Theoretical guarantees under strong convexity

In this Section, we provide theoretical guarantees in the form of upper bounds on the risk $R(\theta^{(T)})$ (see Equation (3.1.1)) for the output $\theta^{(T)}$ of Algorithm 1. These upper bounds are generic with respect to the considered robust estimators $\big(\widehat{g}_j(\cdot)\big)_{j=1}^d$ and rely on the following definition.

Definition 3.1. *Let $\delta \in (0,1)$ be a failure probability. We say that a partial derivatives estimator $\widehat{g}$ has an* error vector $\epsilon(\delta) \in \mathbb{R}_+^d$ *if it satisfies*

$$\mathbb{P}\Big[\sup_{\theta \in \Theta} \big|\widehat{g}_j(\theta) - g_j(\theta)\big| \leqslant \epsilon_j(\delta)\Big] \geqslant 1 - \delta \tag{3.2.2}$$

for all $j \in [\![d]\!]$.

In Section 3.3 below, we specify a value of $\epsilon_j(\delta)$ for each considered robust estimator which will lead to upper bounds on the risk. Recall that $g_j(\theta) = \partial R(\theta)/\partial \theta_j$ and let us denote as e_j the j-th canonical basis vector of $\mathbb{R}^d$. We need the following extra assumptions on the optimization problem itself.

Assumption 3.3. *There exists* $\theta^\star \in \Theta$ *satisfying the stationary gradient condition* $\nabla R(\theta^\star) = 0$. *Moreover, we assume that there are Lipschitz constants* $L_j > 0$ *such that*

$$\big|g_j(\theta + he_j) - g_j(\theta)\big| \leqslant L_j|h|$$

for any $j \in [\![d]\!]$, $h \in \mathbb{R}$ *and* $\theta \in \Theta$ *such that* $\theta + he_j \in \Theta$. *We also consider* $L > 0$ *such that*

$$\big\|g(\theta + h) - g(\theta)\big\| \leqslant L\|h\|$$

for any $h \in \Theta$ *and* $\theta \in \Theta$ *such that* $\theta + h \in \Theta$. *We denote* $L_{\max} := \max_{j \in [\![d]\!]} L_j$ *and* $L_{\min} := \min_{j \in [\![d]\!]} L_j$.

Under Assumptions 3.1 and 3.2, we know that the Lipschitz constants $(L_j)_{j \in [\![d]\!]}$ and L do exist. Indeed, the Hessian matrix of the risk $R(\theta)$ is given by

$$\nabla^2 R(\theta) = \mathbb{E}\big[\ell''(\theta^\top X, Y) X X^\top\big],$$

where $\ell''(z, y) := \partial^2 \ell(z, y)/\partial z^2$, so that

$$L_j = \sup_{\theta \in \Theta} \mathbb{E}\big[\ell''(\theta^\top X, Y)(X^j)^2\big] \quad \text{and} \quad L = \sup_{\theta \in \Theta} \big\|\nabla^2 R(\theta)\big\|_{\mathrm{op}}, \tag{3.2.3}$$

where $\big\|H\big\|_{\mathrm{op}}$ stands for the operator norm of a matrix H. Assumption 3.1 entails $L_j \leqslant \gamma \mathbb{E}\big[(X^j)^2\big]$, which is finite because of Equation (3.1.5) from Assumption 3.2. In order to derive *linear* convergence rates for CGD, it is standard to require strong convexity [341, 442]. Here, we require strong convexity on the risk $R(\theta)$ itself, as described in the following.

Assumption 3.4. *We assume that the risk* R *given by Equation* (3.1.1) *is* λ*-strongly convex, namely that*

$$R(\theta_2) \geqslant R(\theta_1) + \langle \nabla R(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\lambda}{2}\|\theta_2 - \theta_1\|^2 \tag{3.2.4}$$

for any $\theta_1, \theta_2 \in \Theta$.

Assumption 3.4 is satisfied whenever $\lambda_{\min}\big(\nabla^2 R(\theta)\big) \geqslant \lambda$ for any $\theta \in \Theta$, where $\lambda_{\min}(H)$ stands for the smallest eigenvalue of a symmetric matrix H. For the least-squares loss, this translates into the condition $\lambda_{\min}\big(\mathbb{E}[XX^\top]\big) \geqslant \lambda$. Note that one can always make the risk λ-strongly convex by considering ridge penalization, namely by replacing $R(\theta)$ by $R(\theta) + \frac{\lambda}{2}\|\theta\|_2^2$, but we provide also guarantees without this Assumption in Section 3.5 below. The following Theorem provides an upper bound over the risk of Algorithm 1 whenever the estimators $\widehat{g}_j(\cdot)$ have an error vector $\epsilon(\delta)$, as defined in Definition 3.1. We introduce for short $R^\star = R(\theta^\star) = \min_{\theta \in \Theta} R(\theta)$.

Theorem 3.1. *Grant Assumptions 3.1, 3.3 and 3.4. Let* $\theta^{(T)}$ *be the output of Algorithm 1 with step-sizes* $\beta_j = 1/L_j$, *an initial iterate* $\theta^{(0)}$, *uniform coordinates sampling* $p_j = 1/d$ *and estimators of the partial derivatives with error vector* $\epsilon(\cdot)$. *Then, we have*

$$\mathbb{E}\big[R(\theta^{(T)})\big] - R^\star \leqslant \big(R(\theta^{(0)}) - R^\star\big)\Big(1 - \frac{\lambda}{L_{\max} d}\Big)^T + \frac{L_{\max}}{2\lambda L_{\min}}\big\|\epsilon(\delta)\big\|_2^2 \tag{3.2.5}$$

with probability at least $1-\delta$*, where the expectation is w.r.t. the sampling of the coordinates. Now, if Algorithm 1 is run as before, but with an* importance sampling *distribution* $p_j = L_j / \sum_{k \in [\![d]\!]} L_k$*, we have*

$$\mathbb{E}\big[R(\theta^{(T)})\big] - R^\star \leqslant \big(R(\theta^{(0)}) - R^\star\big)\Big(1 - \frac{\lambda}{\sum_{j \in [\![d]\!]} L_j}\Big)^T + \frac{1}{2\lambda}\big\|\epsilon(\delta)\big\|_2^2 \tag{3.2.6}$$

with probability at least $1 - \delta$*.*

The proof of Theorem 3.1 is given in Section 3.9. It adapts standard arguments for the analysis of CGD [341, 442] with inexact estimators of the partial derivatives. The statistical error $\|\epsilon(\delta)\|_2^2$ is studied in Section 3.3 for each considered robust estimator of the partial derivatives. Both (3.2.5) and (3.2.6) are upper bounds on the excess risk with exponentially vanishing optimization errors (called *linear* rate in optimization) and a constant statistical error. The optimization error term of (3.2.6), given by

$$\big(R(\theta^{(0)}) - R^\star\big)\Big(1 - \frac{\lambda}{\sum_{j \in [\![d]\!]} L_j}\Big)^T,$$

goes to 0 exponentially fast as the number of iterations T increases, with a contraction constant better than that of (3.2.5) since $\sum_{j \in [\![d]\!]} L_j \leqslant d L_{\max}$. This can be understood from the fact that importance sampling better exploits the knowledge of the Lipschitz constants L_j. Also, note that T is the number of iterations of CGD, so that $T = Cd$ where C is the number of CGD cycles. Therefore, defining $L' := \frac{1}{d}\sum_{j \in [\![d]\!]} L_j$, we have

$$\Big(1 - \frac{\lambda}{dL'}\Big)^{Cd} \leqslant \Big(1 - \frac{\lambda}{L'}\Big)^C,$$

for $d \geqslant 1$, which leads to a linear rate at least similar to the one of GD [58].

Theorem 3.1 proves an upper bound on the excess risk $R(\theta^{(T)}) - R^\star$ of the iterates of robust CGD directly, without using an intermediate upper bound on $\|\theta^{(T)} - \theta^\star\|_2^2$. This differs from the approaches used by [364, 191] that consider robust GD (while we introduce robust CGD here) to bound the excess risk of the iterates. This allows us to obtain a better contraction factor for the optimization error and a better constant in front of the statistical error. Note that we can derive also an upper bound on $\|\theta^{(T)} - \theta^\star\|_2^2$, see Theorem 3.4 in Section 3.9.

Note that the iterations considered in Algorithm 1 do not perform a projection in Θ. Indeed, one can show that $\|\theta^{(t)} - \theta^\star\|$ is also subject to a contraction and is therefore decreasing w.r.t. t. Thus, if $\theta^{(0)} = 0$, iterates $\theta^{(t)}$ naturally belong to the ℓ_2 ball of radius $2\|\theta^\star\|$.

Step-sizes. The step-sizes $\beta_j = 1/L_j$ are unknown, since they are functionals of the unknown distribution P. So, we provide, in Section 3.8.1, theoretical guarantees similar to that of Theorem 3.1 using step-sizes $\widehat{\beta}_j = 1/\widehat{L}_j$, where $\widehat{L}_j$ is a robust estimator of the upper bound $\overline{L}_j := \gamma \mathbb{E}\big[(X^j)^2\big] \geqslant L_j$ of the Lipschitz constant L_j.

A deterministic result. The previous Theorem 3.1 provides upper bounds on the expectation of the excess risk with respect to the sampling of the coordinates used in CGD. In Theorem 3.2 below, we provide an upper bound similar to the one from Theorem 3.1, but with a fully deterministic variant of CGD, where we replace line 4 of Algorithm 1 with a deterministic cycling through the coordinates.

Theorem 3.2. *Grant Assumptions 3.1, 3.3 and 3.4. Let* $\theta^{(T)}$ *be the output of Algorithm 1 with step-sizes* $\beta_j = 1/L_j$*, an initial iterate* $\theta^{(0)}$*, deterministic cycling over* $[\![d]\!]$ *such that*

$$\{j_{td+1}, j_{td+2}, \ldots, j_{(t+1)d-1}\} = [\![d]\!]$$

for any t and estimators of the partial derivatives with error vector $\epsilon(\cdot)$. Then, we have

$$R(\theta^{(T)}) - R^\star \leqslant \big(R(\theta^{(0)}) - R^\star\big)\big(1 - 2\lambda\kappa\big)^T + \frac{3}{8\lambda\kappa L_{\min}}\big\|\epsilon(\delta)\big\|_2^2$$

with probability at least $1-\delta$, where we introduced the constant

$$\kappa = \frac{1}{8L_{\max}(1 + d(L_{\max}/L_{\min}))}.$$

The proof of Theorem 3.2 is given in Section 3.9 and uses arguments from [25] and [271]. It provides an extra guarantee on the convergence of CGD, for a very general choice of coordinates cycling, at the cost of degraded constants compared to Theorem 3.1, both for the optimization and statistical error terms.

Note that, our convergence results are set under a Lipschitz-smoothness assumption (see also Theorem 3.3 for the non strongly convex case), this excludes problems with non-smooth regularization such as Lasso to which CGD has commonly been applied [443, 151, 447]. Although such applications remain beyond the scope of our theory, there is no reason to doubt that plugging robust estimators, such as those given in Section 3.3 below, into CGD applied to non-smooth problems would lead to improved statistical performance and robustness.

3.3 Robust estimators of the partial derivatives

We consider three estimators of the partial derivatives

$$g_j(\theta) = \frac{\partial R(\theta)}{\partial \theta_j} = \mathbb{E}\big[\ell'(X^\top\theta, Y)X^j\big]$$

that can be used within Algorithm 1: Median-of-Means in Section 3.3.1, Trimmed mean in Section 3.3.2 and an estimator that we will call "Catoni-Holland" in Section 3.3.3. We provide, for each estimator, a concentration inequality for the estimation of $g_j(\theta)$ for fixed θ under a weak moments assumption (Lemmas 3.2, 3.3 and 3.4). We derive also uniform versions of the bounds in each case (Propositions 3.1, 3.2, 3.3 and 3.4) which define the error vectors to be plugged into Theorems 3.1 and 3.2. We also discuss in details the numerical complexity of each estimator and explain that they all are, in their own way, an interpolation between the empirical mean and the median. We wrap up these results in Table 3.1 below.

	Optimal deviation bound	Robustness to outliers	Numerical complexity	Hyper-parameter
`ERM`	No	None	$O(n)$	None
`MOM`	Yes	Yes for $\lvert\mathcal{O}\rvert < K/2$	$O(n+K)$	$K \in [\![n]\!]$
`CH`	Yes	None	$O(n)$	Scale s
`TM`	Yes	Yes for $\lvert\mathcal{O}\rvert < n/8$	$O(n)$	Proportion $\epsilon \in [0, 1/2)$

Table 3.1: Properties of some robust estimators, where `ERM` = Empirical Risk Minimizer (ordinary mean), `MOM` = Median-of-Means, `CH` = Catoni-Holland and `TM` = Trimmed Mean. We recall that n = sample size and $|\mathcal{O}|$ = number of outliers. The parameters of each estimators are: the number of blocks K in `MOM`, a scale parameter $s > 0$ in `CH` and a proportion of samples ϵ in `TM`.

The deviation bound optimality in Table 3.1 is meant in terms of the dependence, up to a constant, on the sample size n, required confidence $\delta \in (0,1)$ and distribution variance[3]. An estimator's deviation bound is deemed optimal if it fits the lower bounds given by Theorems 1 and 3 in [284]. Let us introduce the centered moment of order $1+\alpha$ of the partial derivatives and its maximum over Θ, given by

$$m_{\alpha,j}(\theta) := \mathbb{E}\Big[\big|\ell'(X^\top\theta, Y)X^j - \mathbb{E}[\ell'(X^\top\theta, Y)X^j]\big|^{1+\alpha}\Big] \qquad \text{and} \qquad M_{\alpha,j} = \sup_{\theta\in\Theta} m_{\alpha,j}(\theta) \tag{3.3.1}$$

for $\alpha \in (0,1]$. Note that $m_{1,j}(\theta) = \mathbb{V}\big[\ell'(X^\top\theta, Y)X^j\big]$ and we know that $m_{\alpha,j}(\theta)$ exists, as explained in the next Lemma.

Lemma 3.1. *Under Assumptions 3.1 and 3.2 the risk $R(\theta)$ is well defined for all $\theta \in \Theta$ and we have*

$$\mathbb{E}\big[\big|\ell'(X^\top\theta, Y)X^j\big|^{1+\alpha}\big] < +\infty$$

for any $j \in [\![d]\!]$ and $\theta \in \Theta$.

The proof of Lemma 3.1 involves simple algebra and is provided in Section 3.9 below. Let us introduce

$$g_j^i(\theta) := \ell'(X_i^\top\theta, Y_i)X_i^j, \tag{3.3.2}$$

the sample $i \in [\![n]\!]$ partial derivative for coordinate $j \in [\![d]\!]$.

3.3.1 Median-of-Means

The Median-Of-Means (`MOM`) estimator is the median

$$\widehat{g}_j^{\texttt{MOM}}(\theta) := \text{median}\left(\widehat{g}_j^{(1)}(\theta), \ldots, \widehat{g}_j^{(K)}(\theta)\right) \tag{3.3.3}$$

of the block-wise empirical means

$$\widehat{g}_j^{(k)}(\theta) := \frac{1}{|B_k|}\sum_{i\in B_k} g_j^i(\theta) \tag{3.3.4}$$

within blocks $B_1, \ldots, B_K$ of roughly equal size that form a partition of $[\![n]\!]$ and that are sampled uniformly at random. This estimator depends on the choice of the number K of blocks used to compute it, which can be understood as an "interpolation" parameter between the ordinary mean ($K = 1$) and the median ($K = n$). It is robust to heavy-tailed data and a limited number of outliers as explained in the following lemma.

Lemma 3.2. *Grant Assumptions 3.1 and 3.2 with $\alpha \in (0,1]$. If $|\mathcal{O}| \leqslant K/12$, we have:*

$$\mathbb{P}\left[\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g(\theta)_j\big| > (24 m_{\alpha,j}(\theta))^{1/(1+\alpha)}\Big(\frac{K}{n}\Big)^{\alpha/(1+\alpha)}\right] \leqslant e^{-K/18}$$

for any fixed $j \in [\![d]\!]$ and $\theta \in \Theta$. If we fix a confidence level $\delta \in (0,1)$ and choose $K := \lceil 18\log(1/\delta)\rceil$, we have

$$\begin{aligned}\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g(\theta)_j\big| &\leqslant c_\alpha m_{\alpha,j}(\theta)^{1/(1+\alpha)}\Big(\frac{\log(1/\delta)}{n}\Big)^{\alpha/(1+\alpha)} \\ &\leqslant c_\alpha M_{\alpha,j}^{1/(1+\alpha)}\Big(\frac{\log(1/\delta)}{n}\Big)^{\alpha/(1+\alpha)}\end{aligned} \tag{3.3.5}$$

[3] or more generally the centered moment of order $1+\alpha$ for $\alpha \in (0,1]$, see below.

with a probability larger than $1-\delta$*, where* $c_\alpha := 2^{(3+\alpha)/(1+\alpha)}3^{(1+2\alpha)/(1+\alpha)}$.

The proof of Lemma 3.2 is given in Section 3.9 below and it adapts simple arguments from [284] and [262]. Compared to [284], it provides additional robustness with respect to $|\mathcal{O}| \geqslant 1$ outliers and compared to [262] it provides guarantees with weak moments $\alpha < 1$. An inspection of the proof of Lemma 3.2 shows that it holds also under the assumption $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$ for any $\varepsilon \in (0,1)$ with an increased constant $c_\alpha = 8 \times 3^{1/(1+\alpha)}/\varepsilon^{(1+2\alpha)/(1+\alpha)}$. This concentration bound is optimal under the $(1+\alpha)$-moment assumption (see Theorems 1 and 3 in [284]) and is sub-Gaussian when $\alpha = 1$ (finite variance). The next proposition provides a *uniform* deviation bound over Θ for $\widehat{g}_j^{\texttt{MOM}}(\theta)$.

Proposition 3.1. *Grant Assumptions 3.1 and 3.2 with* $\alpha \in (0,1]$ *and* $|\mathcal{O}| \leqslant K/12$*. We have*

$$\mathbb{P}\Big[\sup_{\theta\in\Theta} \big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big| \leqslant \epsilon_j^{\texttt{MOM}}(\delta)\Big] \geqslant 1-\delta$$

for any $j \in [\![d]\!]$*, with*

$$\epsilon_j^{\texttt{MOM}}(\delta) := c_\alpha\Big(M_{j,\alpha} + \frac{m_{L,\alpha}}{n^\alpha}\Big)^{1/(1+\alpha)}\Big(\frac{\log(d/\delta) + d\log(3\Delta n^{\alpha/(1+\alpha)}/2)}{n}\Big)^{\alpha/(1+\alpha)} + (\overline{L}+L_j)\Big(\frac{1}{n}\Big)^{\alpha/(1+\alpha)}$$

where $\overline{L} = \gamma\mathbb{E}\|X\|^2$*,* $m_{L,\alpha} = \mathbb{E}|\gamma\|X\|^2 - \overline{L}|^{1+\alpha}$ *and* $c_\alpha = 2^{(3+2\alpha)/(1+\alpha)}3^{(1+3\alpha)/(1+\alpha)}$.

The proof of Proposition 3.1 is given in Section 3.9 and uses methods similar to Lemma 3.2 with an ε-net argument. This defines the error vector $\epsilon^{\texttt{MOM}}(\delta)$ of the MOM estimator of the partial derivatives in the sense of Definition 3.1, that can be combined directly with the convergence results from Theorems 3.1 and 3.2 from Section 3.2. Since the optimization error decreases exponentially w.r.t. the number of iterations T in these theorems, while the estimator error $\|\epsilon(\delta)\|_2$ is fixed, one only needs $T = O(\|\epsilon(\delta)\|_2)$ to make both terms of the same order.

About uniform bounds. What is necessary to obtain a control of the excess risk of robust CGD is a control of the noise terms $|\widehat{g}_j(\theta^{(t)}) - g_j(\theta^{(t)})|$, where both iterates $\theta^{(t)}$ and estimators $\widehat{g}_j(\cdot)$ of the partial derivatives depend on the same data. This forbids the direct use of a deviation such as the one from Lemma 3.2 (and Lemmas 3.3 and 3.4 below) where θ must be deterministic. We use in this paper an approach based on uniform deviation bounds (Propositions 3.1, 3.3 and 3.4) in order to bypass this problem, similarly to [193] and many other papers using empirical process theory. This is of course pessimistic, since $\theta^{(t)}$ goes to $\theta^\star$ as t increases. Another approach considered in [364] is to split data into segments of size n/T and to compute the gradient estimator using a segment independent of the ones used to compute the current iterate. This approach departs strongly from what is actually done in practice, and leads to controls on the excess risk expressed with $\widetilde{\delta} = \delta/T$ and $\widetilde{n} = n/T$ instead of δ and n, hence a deterioration of the control of the excess risk. Our approach based on uniform deviations also suffers from a deterioration, due to the use of an ε-net argument, observed in Proposition 3.1 through the extra $d^{\alpha/(1+\alpha)}$ factor when compared to Lemma 3.2. Avoiding such deteriorations is an open difficult problem, either using uniform bounds or data splitting.

In addition to Proposition 3.1, we propose another uniform deviation bound for $\widehat{g}_j^{\texttt{MOM}}(\theta)$ using the Rademacher complexity, which is a fundamental tool in statistical learning theory and empirical process theory [267, 240, 22]. Let us introduce

$$\mathcal{R}_j(\Theta) = \mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i\in\mathcal{I}}\varepsilon_i g_j^i(\theta)\Big]$$

for $j \in [\![d]\!]$, where $(\varepsilon_i)_{i\in\mathcal{I}}$ are i.i.d Rademacher variables and where we recall that $\mathcal{I}$ contains the inliers indices (see Assumption 3.2).

Proposition 3.2. *Grant Assumptions 3.1 and 3.2 with $\alpha \in (0,1]$. If $|\mathcal{O}| \leqslant K/12$, we have*

$$\mathbb{P}\Big[\sup_{\theta\in\Theta} \big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big| \geqslant \max\Big(\Big(\frac{36M_{\alpha,j}}{(n/K)^\alpha}\Big)^{1/(1+\alpha)}, \frac{64\mathcal{R}_j(\Theta)}{n}\Big)\Big] \leqslant e^{-K/18}$$

for any $j \in [\![d]\!]$. If we fix a confidence level $\delta \in (0,1)$ and choose $K := \lceil 18\log(1/\delta)\rceil$, we have

$$\sup_{\theta\in\Theta} \big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g(\theta)_j\big| \leqslant \max\Big(c_\alpha M_{\alpha,j}^{1/(1+\alpha)}\Big(\frac{\log(d/\delta)}{n}\Big)^{\alpha/(1+\alpha)}, \frac{64\mathcal{R}_j(\Theta)}{n}\Big) \tag{3.3.6}$$

with a probability larger than $1-\delta$ for all $j \in [\![d]\!]$, where $c_\alpha := 2^{(2+\alpha)/(1+\alpha)}3^2$. Moreover, if $\mu_{X,j}^{2(1+\alpha)} := \mathbb{E}[(X^j)^{2(1+\alpha)}] < +\infty$ for all $j \in [\![d]\!]$ we have

$$\mathcal{R}_j(\Theta) \leqslant \gamma\Delta C_\alpha\Big(n\mu_{X,j}^{1+\alpha}\sum_{k\in[\![d]\!]}\mu_{X,k}^{1+\alpha}\Big)^{1/(1+\alpha)} = O((nd)^{1/(1+\alpha)}),$$

where C_α is a constant depending only on α.

The proof of Proposition 3.2 is given in Section 3.9 and borrows arguments from [262, 50]. For $\alpha = 1$, the bound (3.3.6) has order $O(\sqrt{d/n})$ similarly to Theorem 2 from [262], although we consider here a different quantity (Rademacher complexity of the partial derivatives, towards the study of the *explicit* robust CGD algorithm, while *implicit* algorithms are studied herein). Note also that we do not prove similar uniform bounds using the Rademacher complexity for the `TM` and `CH` algorithms considered below, an interesting open question.

Comparison with [364, 191]. A first distinction of our results compared to [364, 191] is the use and theoretical study of robust CGD instead of robust GD. A second distinction is that we work under $1+\alpha$ moments on the partial derivatives of the risk, while [364, 191] require $\alpha = 1$. Our setting is similar but more general than the one laid out in [191] since the latter does not consider the presence of outliers. Theorem 5 from [191] states linear convergence of the optimization error thanks to strong convexity similarly to our Theorem 3.1. Their management of the statistical error is quite similar and leads to the same rate. However, our bound involves the sum of the coordinatewise moments of the gradient thanks to Proposition 3.1, an improvement over the bound from [191] which is only stated in terms of a uniform bound on the coordinate variances. Another reference point is the heavy-tailed setting of [364], which deals with heavy-tails independently from the problem of corruption and requires $\alpha = 1$. More importantly, the approach considered in [364] relies on data-splitting, which departs significantly from what is done in practice, while we do not perform data-spitting but use uniform bounds, as discussed above.

Complexity of $\widehat{g}_j^{\texttt{MOM}}(\theta)$. The computation of $\widehat{g}_j^{\texttt{MOM}}(\theta)$ requires (a) to sample a permutation of $[\![n]\!]$ to sample the blocks $B_1, \dots, B_K$, (b) to compute averages within the blocks and (c) to compute the median of K numbers. Sampling a permutation of $[\![n]\!]$ has complexity $O(n)$ using the Fischer-Yates algorithm [238], and so does the computation of the averages, so that (a) and (b) have complexity $O(n)$. The computation of the median of K numbers can be done using the quickselect algorithm [189] with $O(K)$ average complexity, leading to a complexity $O(n+K) = O(n)$ since $K < n$.

3.3.2 Trimmed Mean estimator

The idea of the Trimmed Mean (`TM`) estimator is to exclude a proportion of data in the tails of their distribution to achieve robustness. We are aware of two variants: (1) one in which samples in the tails are removed, the remaining samples being used to compute an empirical mean and (2) another variant in which samples in the tails are clipped but not removed from the empirical mean. Variant (1) is robust to η-corruption[4] whenever the data distribution is sub-exponential [273] or sub-Gaussian [115, 114, 120]. Variant (2), also known as *Winsorized mean*, enjoys a sub-Gaussian deviation [284] for heavy-tailed distributions. Both robustness properties are shown simultaneously (sub-Gaussian deviations under a heavy-tails assumption and η-corruption) in [288] (see Theorem 1 therein). We consider below variant (2), which proceeds as follows.

First, the `TM` estimator splits $[\![n]\!] = [\![n/2]\!] \cup [\![n/2]\!]^{\complement}$ where $[\![n/2]\!]^{\complement} = [\![n]\!] \setminus [\![n/2]\!]$, assuming without loss of generality that n is even, and it computes the sample derivatives $g_j^i(\theta)$ given by (3.3.2) for all $i \in [\![n]\!]$. Then, given a proportion $\epsilon \in [0, 1/2)$, it computes the ϵ and $1-\epsilon$ quantiles of $(g_j^i(\theta))_{i\in[\![n/2]\!]}$ given by

$$q_\epsilon := g_j^{([\epsilon n/2])}(\theta) \quad \text{and} \quad q_{1-\epsilon} := g_j^{([(1-\epsilon)n/2])}(\theta),$$

where $g_j^{(1)}(\theta) \leqslant \cdots \leqslant g_j^{(n/2)}(\theta)$ is the order statistics of $(g_j^i(\theta))_{i\in[\![n/2]\!]}$ and where $[x]$ is the lower integer part of $x \in \mathbb{R}$. Finally, the estimator is computed as

$$\widehat{g}_j^{\mathtt{TM}}(\theta) = \frac{2}{n} \sum_{i\in[\![n/2]\!]^{\complement}} q_\epsilon \vee g_j^i(\theta) \wedge q_{1-\epsilon}, \tag{3.3.7}$$

where $a \wedge b := \min(a, b)$ and $a \vee b := \max(a, b)$, namely it is the average of the partial derivatives from samples in $[\![n/2]\!]^{\complement}$ clipped in the interval $[q_\epsilon, q_{1-\epsilon}]$. Note that $\widehat{g}_j^{\mathtt{TM}}(\theta)$ is also some form of "interpolation" between the average and the median through ϵ: it is the average of the partial derivatives for $\epsilon = 0$ and their median for $\epsilon = 1/2$. As explained in the next lemma, the `TM` estimator is robust both to a proportion of corrupted samples and heavy-tailed data.

Lemma 3.3. *Grant Assumptions 3.1 and 3.2 with $\alpha \in (0, 1]$ and assume that $|\mathcal{O}| \leqslant \eta n$ with $\eta < 1/8$. If we fix a confidence level $\delta \in (0, 1)$ and choose $\epsilon = 8\eta + 12\log(4/\delta)/n$, we have*

$$\begin{aligned}\left|\widehat{g}_j^{\mathtt{TM}}(\theta) - g_j(\theta)\right| &\leqslant 7m_{\alpha,j}(\theta)^{1/(1+\alpha)}\Big(4\eta + \frac{6\log(4/\delta)}{n}\Big)^{\alpha/(1+\alpha)} \\ &\leqslant 7M_{\alpha,j}^{1/(1+\alpha)}\Big(4\eta + \frac{6\log(4/\delta)}{n}\Big)^{\alpha/(1+\alpha)}\end{aligned}$$

with a probability larger than $1-\delta$.

The proof of Lemma 3.3 is given in Section 3.9 and extends Theorem 1 from [288] to $\alpha \in (0, 1]$ instead of $\alpha = 1$ only. It shows that the `TM` estimator has the remarkable quality of being simultaneously robust to heavy-tailed and a *fraction* of corrupted data, as opposed to `MOM` which is only robust to a limited *number* of outliers. Note that for the computation of the `TM` estimator, the splitting $[\![n]\!] = [\![n/2]\!] \cup [\![n/2]\!]^{\complement}$ is a technical theoretical requirement used to induce independence between $q_\epsilon, q_{1-\epsilon}$ and the sample partial derivatives $(g_j^i(\theta))_{i\in[\![n/2]\!]^{\complement}}$ involved in the average (3.3.7). Our implementation does not use this splitting.

[4] We call "η-corruption" the context where the outlier set $\mathcal{O}$ in Assumption 3.2 satisfies $|\mathcal{O}| = \eta n$ with $\eta \in [0, 1/2)$

Comparison with [364]. A comparison between Lemma 3.3 and the results from [364] pertaining to the corrupted setting is relevant here. We first point out that corruption in [364] is modeled as receiving data from the "η-contaminated" distribution $(1-\eta)P + \eta Q$ with Q an arbitrary distribution. On the other hand, Lemma 3.3 considers the more general η-corrupted setting where an η-proportion of the data is replaced by arbitrary outliers *after* sampling. In this case, Lemma 3.3 results in a statistical error with a dependence of order $\sqrt{\eta d}$ in the corruption (on the vector euclidean norm). On the other hand, Lemma 1 in [364] yields a better dependence of order $\sqrt{\eta \log d}$ in the corresponding case. Keep in mind, however, that Algorithm 2 from [364] which achieves this rate requires recursive SVD decompositions to compute a robust gradient making it computationally heavy and impractical for moderately high dimension. Additionally, the relevant results in [364] require a stronger moment assumption on the gradient and impose additional constraints on the corruption rate η. We also mention Algorithm 5 from [364] which yields an even better dependence on the dimension (see their Lemma 2), although it involves a computationally costly procedure as well. Besides, knowledge of the trace and operator norm of the covariance matrix of the estimated vector is required which makes the algorithm more difficult to use in practice.

Proposition 3.3. *Grant Assumptions 3.1 and 3.2 with $\alpha \in (0,1]$ and $|\mathcal{O}| \leqslant \eta n$. We have*

$$\mathbb{P}\Big[\sup_{\theta\in\Theta} \big|\widehat{g}_j^{\mathtt{TM}}(\theta) - g_j(\theta)\big| \leqslant \epsilon_j^{\mathtt{TM}}(\delta)\Big] \geqslant 1-\delta$$

for any $j \in [\![d]\!]$ with

$$\epsilon_j^{\mathtt{TM}}(\delta) := 28\Big(M_{j,\alpha} + \frac{m_{L,\alpha}}{n^{\alpha(1+\alpha)}}\Big)^{1/(1+\alpha)}\Big(2\eta + 3\frac{\log(4d/\delta) + d\log(3\Delta n^{\alpha/(1+\alpha)}/2)}{n}\Big)^{\alpha/(1+\alpha)} + \frac{\overline{L} + L_j}{n^{\alpha/(1+\alpha)}}$$

where $\overline{L}$ and $m_{L,\alpha}$ are as in Proposition 3.1.

The proof of Proposition 3.3 is given in Section 3.9 and uses an ε-net argument to obtain a uniform bound. Similarly to `MOM`, the resulting statistical error has optimal dependence on the $(1+\alpha)$-moments of the partial derivatives (3.3.1).

By plugging, the error vector $\epsilon^{\mathtt{TM}}(\delta)$ into Theorem 3.1, we obtain the following corollary which summarizes the best learning guarantees we obtain.

Corollary 3.1. *In the combined settings of Theorem 3.1 and Proposition 3.3, let $\widehat{\theta} := \theta^{(T)}$ denote the estimator obtained by running CGD with importance sampling using the `TM` estimator for T iterations where T will be specified shortly. Then, with probability at least $1-\delta$ we have*

$$\frac{\lambda}{2}\mathbb{E}\big\|\widehat{\theta} - \theta^\star\big\|^2 \leqslant \mathbb{E}[R(\widehat{\theta})] - R^\star \leqslant O\Big(\frac{1}{\lambda}\Big(\sum_{j\in[\![d]\!]} M_{j,\alpha}^{2/(1+\alpha)}\Big)\Big(\eta + \frac{\log(d/\delta) + d\log(n)}{n}\Big)^{2\alpha/(1+\alpha)}\Big), \quad (3.3.8)$$

where the expectations are w.r.t. the sampling of the coordinates. The above bound holds for a number of iterations T of order

$$T \geqslant \Omega\Bigg(\log\Bigg[\frac{\lambda(R(\theta^{(0)}) - R^\star)}{\sum_{j\in[\![d]\!]} M_{j,\alpha}^{2/(1+\alpha)}\Big(\eta + \frac{\log(d/\delta)+d\log(n)}{n}\Big)^{2\alpha/(1+\alpha)}}\Bigg] \Big/ \log\Big[\frac{1}{1-\lambda/\sum_{j\in[\![d]\!]} L_j}\Big]\Bigg).$$

The proof of Corollary 3.1 is given in Section 3.9 and is a straightforward combination of

Theorem 3.1 and Proposition 3.3 where a big O notation was used to make the bound more legible.

Complexity of $\widehat{g}_j^{\texttt{TM}}(\theta)$. The most demanding part for the computation of $\widehat{g}_j^{\texttt{TM}}(\theta)$ is the computation of q_ϵ and $q_{1-\epsilon}$. A naive idea is to sort all n values at an average cost $O(n \log n)$ with quicksort for example [189] and to simply retrieve the desired order statistics afterwards. Of course, better approaches are possible, including the median-of-medians algorithm (not to be confused with `MOM`), which remarkably manages to keep the cost of finding an order statistic with complexity $O(n)$ even in the worst case (see for instance Chapter 9 of [98]). However, the constant hidden in the previous big-O notations seriously impact performances in real-world implementations: we compared several implementations experimentally and concluded that a variant of the quickselect algorithm [189] was the fastest for this problem.

3.3.3 Catoni-Holland estimator

This estimator is a variation of the robust mean estimator by Catoni [73] introduced by Holland [191] for robust statistical learning, hence the name "Catoni-Holland", that we will denote $\widehat{g}_j^{\texttt{CH}}(\theta)$. It is defined as an M-estimator which consists in solving

$$\sum_{i=1}^{n} \psi\Big(\frac{g_j^i(\theta) - \zeta}{\widehat{s}_j(\theta)}\Big) = 0 \tag{3.3.9}$$

with respect to ζ, where ψ is an uneven function satisfying $\psi(0) = 0$, $\psi(x) \sim x$ when $x \sim 0$ and $\psi(x) = o(x)$ when $x \to +\infty$ and where $\widehat{s}_j(\theta) > 0$ is a scale estimator. An approximate solution can be found using the fixed-point iterations

$$\zeta_{k+1} = \zeta_k + \frac{\widehat{s}_j(\theta)}{n} \sum_{i=1}^{n} \psi\Big(\frac{g_j^i(\theta) - \zeta_k}{\widehat{s}_j(\theta)}\Big),$$

which can easily be shown to converge to the desired value thanks to the monotonicity and Lipschitz-property of ψ. Following [191], we use the function $\psi(x) = 2\arctan(\exp(x)) - \pi/2$, while functions satisfying $-\log(1 - x + x^2/2) \leqslant \psi(x) \leqslant \log(1 + x + x^2/2)$ are considered in [73]. As explained in [191], the scale estimator is given by

$$\widehat{s}_j(\theta) := \widehat{\sigma}_j(\theta) \sqrt{\frac{n}{2\log(4/\delta)}}, \tag{3.3.10}$$

for a confidence level $\delta \in (0, 1)$, where $\widehat{\sigma}_j(\theta)$ is an estimator of the standard deviation of the partial derivative $\sigma_j(\theta) := m_{1,j}(\theta)^{1/2} = \mathbb{V}[\ell'(X^\top\theta, Y)X^j]^{1/2}$, see (3.3.1). The estimator $\widehat{\sigma}_j(\theta)$ is defined through another M-estimator solution to

$$\sum_{i=1}^{n} \chi\Big(\frac{g_j^i(\theta) - \bar{g}_j(\theta)}{\sigma}\Big) = 0 \tag{3.3.11}$$

with respect to σ, where $\bar{g}_j(\theta) = \frac{1}{n}\sum_{i=1}^n g_j^i(\theta)$ and χ is an even function satisfying $\chi(0) < 0$ and $\chi(x) > 0$ as $x \to +\infty$. We use the same function as in [191] given by $\chi(u) = u^2/(1 + u^2) - c$ where c is such that $\mathbb{E}\chi(Z) = 0$ for Z a standard Gaussian random variable. To compute $\widehat{\sigma}_j(\theta)$

we use also fixed-point iterations

$$\sigma_{k+1} = \sigma_k\Big(1 - \frac{\chi(0)}{n}\sum_{i=1}^{n}\chi\Big(\frac{g_j^i(\theta) - \bar{g}_j(\theta)}{\sigma_k}\Big)\Big). \tag{3.3.12}$$

We refer to the supplementary material of [191] for further details on this procedure.

The CH estimator can be understood, once again, as an interpolation between the average and the median of the partial derivatives. Indeed, whenever s is large, the function $\psi(\cdot/s)$ is close to the sign function, which, if used in (3.3.9), leads to an M-estimator corresponding to the median [424]. For s small, $\psi(\cdot/s)$ is close to the identity, so that minimizing (3.3.9) leads to an ordinary average. As explained in the next lemma, this estimator is robust to heavy-tailed data (with $\alpha = 1$).

Lemma 3.4. *Grant Assumptions 3.1 and 3.2 with $\alpha = 1$ and assume that $\mathcal{O} = \emptyset$ (no outliers). For some failure probability $\delta > 0$, assume that we have, with probability at least $1 - \delta/2$, that $\sigma_j(\theta)/C' \leqslant \widehat{\sigma}_j(\theta) \leqslant C'\sigma_j(\theta)$ for some constant $C' > 1$. Then, we have*

$$\big|\widehat{g}_j^{\mathtt{CH}}(\theta) - g_j(\theta)\big| \leqslant C'\sigma_j(\theta)\sqrt{\frac{8\log(4/\delta)}{n}} \leqslant C'\Sigma_j\sqrt{\frac{8\log(4/\delta)}{n}}$$

with probability at least $1 - \delta$, where $\Sigma_j = M_{1,j} = \sup_{\theta\in\Theta}\sigma_j(\theta)$.

The proof of Lemma 3.4 is given in Section 3.9 and is an almost direct application of the deviation bound from [191]. If $C' \approx 1$, the deviation bound of $\widehat{g}_j^{\mathtt{CH}}(\cdot)$ is better than the ones given in Lemmas 3.2 and 3.3 with $\alpha = 1$. This stems from the fact that the analysis of Catoni's estimator [73] results in a deviation with the best possible constant [111]. However, contrary to MOM and TM, an estimator of the scale is necessary: it makes CH computationally much more demanding (see Figure 3.1 below), since it requires to perform two fixed-point iterations to approximate both $\widehat{\sigma}_j(\theta)$ and $\widehat{g}_j^{\mathtt{CH}}(\theta)$ and it requires Assumption 3.2 with $\alpha = 1$ so that $\sigma_j(\theta) < +\infty$. Moreover, there is no guaranteed robustness to outliers, a fact confirmed by the numerical experiments performed in Section 3.6 below.

Proposition 3.4. *Grant Assumptions 3.1 and 3.2 with $\alpha = 1$ and $\mathcal{O} = \emptyset$. Denote $\overline{L} = \mathbb{E}[\gamma\|X\|^2]$, $\sigma_L^2 = \mathbb{V}[\gamma\|X\|^2]$ and assume that for all $\theta, \widetilde{\theta} \in \Theta$ such that $\|\theta - \widetilde{\theta}\| \leqslant 1/\sqrt{n}$ we have*

$$\frac{1}{2}\sigma_j^2(\widetilde{\theta}) \leqslant \sigma_j^2(\theta) \leqslant 2\sigma_j^2(\widetilde{\theta}) \quad \text{and} \quad \frac{\sigma_j(\theta)}{\sigma_L} \geqslant \frac{1}{\sqrt{n}}.$$

Furthermore, assume that for all $\theta \in \Theta$, the variance estimator $\widehat{\sigma}_j(\theta)$ defined by (3.3.11) satisfies $\sigma_j(\theta)/C' \leqslant \widehat{\sigma}_j(\theta) \leqslant C'\sigma_j(\theta)$ for some constant $C' > 1$ with probability at least $1 - \delta/2$. Then, we have

$$\mathbb{P}\Big[\sup_{\theta\in\Theta}\big|\widehat{g}_j^{\mathtt{CH}}(\theta) - g_j(\theta)\big| \leqslant \epsilon_j^{\mathtt{CH}}(\delta)\Big] \geqslant 1 - \delta$$

for any $j \in [\![d]\!]$ with

$$\epsilon_j^{\mathtt{CH}}(\delta) := 4C'\Big(2\Sigma_j + \frac{\sigma_L}{\sqrt{n}}\Big)\sqrt{\frac{\log(4d/\delta) + d\log(3\Delta\sqrt{n}/2)}{n}} + \frac{\overline{L} + L_j}{\sqrt{n}}$$

where $\overline{L}$ is as in Proposition 3.1.

The proof of Proposition 3.4 is given in Section 3.9. It uses again an ε-net argument combined with a careful control of the variations of $\widehat{g}_j^{\mathtt{CH}}(\theta)$ with respect to θ. Compared with [191], we make

a different use of the CH estimator: while it is used therein to estimate the whole gradient $\nabla R(\theta)$ during the robust GD iterations, we use it here to estimate the partial derivatives $g_j(\theta)$ during iterations of robust CGD. The numerical experiments from Section 3.6 confirm, in particular, that our approach leads to a considerable speedup and improved statistical performances when compared to [191].

The statements of Lemma 3.4 and Proposition 3.4 require $\alpha = 1$, while a very recent extension of Catoni's bound [86] is available for $\alpha \in (0, 1)$. However, the necessity to estimate the centered $(1 + \alpha)$-moment subsists (standard-deviation for $\alpha = 1$). Although iteration (3.3.12) may be adapted to this case, theoretical guarantees for it do lack. Note that even for $\alpha = 1$, the statements of Lemma 3.4 and Proposition 3.4 require assumptions on $\sigma_j^2(\theta)$ and $\widehat{\sigma}_j(\theta)$: an extension to $\alpha \in (0, 1]$ would lead to a set of even more intricate assumptions.

Complexity of $\widehat{g}_j^{\texttt{CH}}(\theta)$. It is not straightforward to analyze the complexity of this estimator, since it involves fixed-point iterations with a number of iterations that can vary from one run to the other. However, each iteration has complexity $O(n)$ and we observe empirically that the number of iterations is of constant order (usually smaller than 10) independently from the required confidence. Therefore, the overall complexity remains in $O(n)$ as demonstrated also by Figure 3.1 below. The latter also shows that the numerical complexity of CH is larger than that of MOM and TM, which later impacts the overall training time.

3.3.4 A comparison of the numerical complexities

As explained above, all the considered estimators of the partial derivatives have a numerical complexity $O(n)$. However, they perform different computations and have very different running times in practice. So, in order to compare their actual computational complexities we perform the following experiment. We consider an increasing sample size n between 10^2 and 10^6 on a logarithmic scale and run all the estimators: MOM, TM, CH and ERM, which is the average of the per-sample partial derivatives $g_j^i(\theta)$. We fix their parameters so as to obtain deviation bounds with confidence $1 - \delta = 99\%$: this corresponds to 82 blocks for MOM, $\epsilon = 72/n$ for TM and $\delta = 0.01$ for CH, but the conclusion is similar with different combinations of parameters. We use random samples with student $t(2.1)$ distribution (a finite variance distribution but with heavy tails, although run times do not differ by much when using different distributions). This leads to the display proposed in Figure 3.1, where we display the averaged timings over 100 repetitions (together with standard-deviations).

We observe that the run times of the estimators increase with a similar slope (on a logarithmic scale) against the sample size, confirming the $O(n)$ complexities. However, their timings differ significantly. MOM and TM share similar timings (TM becomes faster than MOM for large samples) and are about 10 times slower than ERM. CH is the slowest of all and is roughly 50 times slower than ERM. This is of course related to the fact that CH requires to perform the fixed-point iterations each of which roughly costing $\Theta(n)$. In all cases, the estimators' complexities remain in $O(n)$ so that the complexity of a single iteration of robust CGD (see Algorithm 1) using either of them is $O(n)$, which is identical to the complexity of a non-robust ERM-based CGD. This means that Algorithm 1 achieves robustness at a limited cost, where the computational difference lies only in the constants in front of the big O notations.

3.4 Related works

The robust statistics field appeared in the 60s with the pioneering works of [421] and [203] and has received longstanding interest since then. Several works pursued the development of

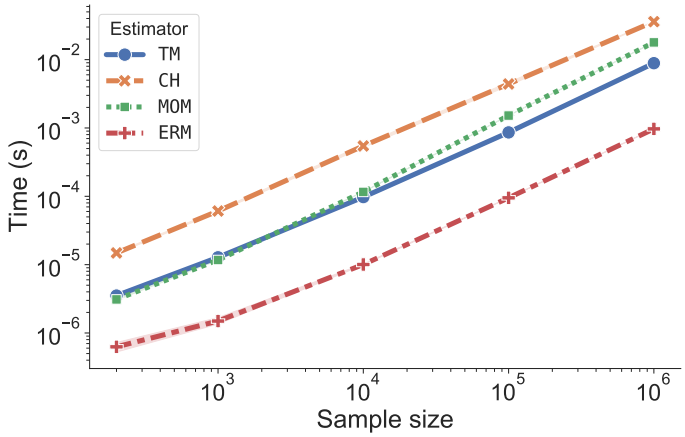

Figure 3.1: Average running time (y-axis) of all the considered estimators against an increasing sample size (x-axis). The run times increase with a similar slope (on a logarithmic scale), confirming $O(n)$ complexities, but differ significantly: `ERM` is of course the fastest, followed by `TM` and `MOM` (both are close) and finally `CH`, which is the slowest.

robust statistical methods including non-convex M-estimators [205], ℓ_1 tournaments [109, 132] and methods based on depth functions [85, 155, 319], the latter being difficult to use in practice because of their numerical complexity.

Renewed interest has manifested recently, related, on the one hand, to the increasing need for algorithms able to learn from large non-curated data sets and on the other hand, to the development of robust mean estimators with good theoretical guarantees under weak moment assumptions, including Median-of-Means (MOM) [339, 7, 217] and Catoni's estimator [73]. Under adversarial corruption [81], several statistical learning problems are studied in a robust setting, such as parameter estimation [251, 361, 317, 114, 288], regression [236, 274, 91, 31], classification [262, 237, 275], PCA [269, 69, 353] and most recently online learning [143].

In the heavy-tailed setting, a robust learning approach introduced in [57] proposes to optimize a robust estimator of the risk based on Catoni's mean estimator [73] resulting in an implicit estimator for which near-optimal guarantees are shown under weak assumptions on the data. However, the new risk may not be convex (even if the considered loss is), so that its minimization may be expensive and lead to an estimator unrelated to the one theoretically studied, potentially making the associated guarantees inapplicable. More recently, an explicit variant was proposed in [451] which applies Catoni's influence function to each term of the sum defining the empirical risk for linear regression. The associated optimum enjoys a sub-Gaussian bound on the excess risk, albeit with a slow rate since the ℓ_1 loss was used. A follow-up extended this result under weaker distribution assumptions [86]. The main drawback of this approach is that the unconventional use of the influence function introduces a considerable amount of bias which appears in the excess risk bounds.

Another way to obtain a robust estimator was proposed by [316, 197] and consists in computing standard ERMs on disjoint subsets of the data and aggregating them using a multi-dimensional MOM. This approach recently appeared in [190] as well with various aggregation strategies in order to perform robust distributed learning. Although the previous works use easily implementable aggregation procedures, the associated deviation bounds are sub-optimal (see for instance [284]). Moreover, dividing the data into multiple subsets makes the method impractical for small sample sizes and may introduce bias coming from the choice of such a subdivision.

In the setting where an η-proportion of the data consist of arbitrary outliers, a robust meta-

algorithm is introduced in [115], which repeatedly trains a given base learner and filters outliers based on an eccentricity score. The method reaches the target $\sigma\sqrt{\eta}$ error rate with σ the gradient standard deviation, although the requirement of multiple training rounds may be computationally expensive.

More recently, robust solutions to classification problems were proposed in [262] by using MOM to estimate the risk and computing gradients on trustworthy data subsets in order to perform descent. A variant was also proposed by the same authors in [260] where a pair of parameters is alternately optimized for a min-max objective. The resulting algorithm is efficient numerically, though it requires a vanishing step-size to converge due to the variance coming from gradient estimation. Moreover, the provided theoretical guarantees concern the optimum of the formulated problem but not the optimization algorithm put to use.

Several recent papers [364, 193, 192, 191, 89, 1] use a form of robust gradient descent, where learning is guided by various robust estimators of the true gradient $\nabla R(\theta)$. Two such estimators are proposed in [364]. The first one is a vector analog of MOM where the scalar median is replaced by the geometric median

$$\mathrm{GMed}(g_1, \dots, g_K) := \operatorname*{arg\,min}_{g \in \mathbb{R}^d} \sum_{j=1}^{K} \|g - g_j\|_2, \tag{3.4.1}$$

which can be computed using the algorithm given in [428]. This vector mean estimator enjoys improved concentration properties over the standard mean as shown in [316] although these remain sub-optimal (see also [284]). A line of works [289, 195, 92, 107, 288, 268, 117] specifically addresses the issue of devising efficient procedures with optimal deviation bounds.

Supervised learning with robustness to heavy-tails and a limited number of outliers is thus achieved but at a possibly high computational cost. The second algorithm called "Huber gradient estimator" is intended for Huber's ϵ-contamination setting. It uses recursive SVD decompositions followed by projections and truncations in order to filter out corruption. The method proves to be robust to data corruption but its computational cost becomes prohibitive as soon as the data has moderately large dimensionality.

Finally, the recent work of [356] proposed to perform robust regression by applying an initial filtering step on the data followed by regression using the robust Huber loss function. Remarkably, the resulting algorithm attains the optimal rates and is simultaneously robust to η-corruption and heavy tails. However, the theoretical guarantees only apply for linear regression and require assumptions which are rarely satisfied in practice such as isotropic covariance of the data.

Table 3.2 summarizes and compares the characteristics of a number of previously mentioned algorithms with ours. The statistical rate may be understood as the final excess risk or parameter error which are interchangeable up to a constant thanks to strong convexity. We have marked the complexities of some algorithms with a dagger (†) to signal the use of iteratively computed estimators with unpredictable iteration count. This indicates that a big constant is hidden by the big O notation. Note that the rows "GD-Huber gradient" and "GD-Geometric MOM" (drawn from [364]) have statistical rates in terms of $\widetilde{n} = n/T$ and $\widetilde{\delta} = \delta/T$ with T the optimization iterations count. This results from a sample splitting strategy yielding milder dimension dependence. However, one can check that the best choice of T degrades these bounds by a factor $\sim \log n$ roughly[5]. Finally, the statistical rate of "GD-implicit MOM" is marked with a double dagger (‡) because it is derived under the only assumption that the loss function is Lipschitz. Moreover, it only bounds the error on objective value estimation and does not directly apply to the estimate computed by the latter algorithm.

[5]Indeed, considering strong convexity, optimization converges linearly and the final bound is of the form $a\exp(-bT) + cT\log(T/\delta)/n$ for some $a, b, c > 0$ and one can see that $T \sim \log n$ is approximately optimal.

Algorithm	Statistical Performance	Iteration/cycle complexity	Robustness to corruption
CGD-MOM (`MOM`) This paper	$O\Big(\frac{d\log(d/\delta)+d^2\log(n)}{n}\Big)$	$O((n+K)d)$ (cycle)	Yes for $\lvert\mathcal{O}\rvert < K/2$
CGD-TM (`TM`) This paper	$O\Big(d\Big(\eta + \frac{\log(d/\delta)+d\log(n)}{n}\Big)\Big)$	$O(nd)$ (cycle)	Yes for $\lvert\mathcal{O}\rvert < n/8$
CGD-CH (`CH`) This paper	$O\Big(\frac{d\log(d/\delta)+d^2\log(n)}{n}\Big)$	$O(nd)\dagger$ (cycle)	None
GD-Geometric MOM [364] (`GMOM`)	$O\Big(\frac{d\log(1/\widetilde{\delta})}{\widetilde{n}}\Big)$	$O((n+K)d)\dagger$	Yes for $\lvert\mathcal{O}\rvert < K/2$
GD-Huber gradient (`HG`) [364]	$O\Big(\log(d)\Big(\eta + \Big(\frac{d\log(d)\log(\widetilde{n}/(d\widetilde{\delta}))}{\widetilde{n}}\Big)^{3/4} + d\sqrt{\frac{\eta\log(d)\log(d\log(d)/\widetilde{\delta})}{\widetilde{n}}}\Big)\Big)$	$O(nd^2+d^3)\dagger$	Yes for η-*contamination*
GD-implicit MOM [262] (`LLM`)	$O\Big(\sqrt{\frac{d+\log(1/\delta)}{n}}\Big)\ddagger$	$O(nd)$	Yes for $\lvert\mathcal{O}\rvert < K/4$
GD-CH (`CH GD`) [191]	$O\Big(\frac{d\log(d/\delta)+d^2\log(n)}{n}\Big)$	$O(nd)\dagger$	None

Table 3.2: Summary of the main characteristics of our proposed algorithms (using CGD) and the main competitors in the literature. The notations $\widetilde{n}$ and $\widetilde{\delta}$ stand for n/T and δ/T respectively with T the optimization horizon. All statistical rates are derived under a strong convexity assumption except for "GD-implicit MOM". For each algorithm, the combination of optimization method and gradient estimator is indicated and the associated code name used in the experimental section is given between parentheses. Cycle complexities are given for CGD algorithms for more relevant comparison.

3.5 Theoretical guarantee without strong convexity

In this section we provide an upper bound similar to that of Theorem 3.1, but without the strong convexity condition from Assumption 3.4. As explained in Theorem 3.3 below, without strong convexity, the optimization error shrinks at a slower sub-linear rate when compared to Theorem 3.1 (a well-known fact, see [58]). In order to ensure that robust CGD, which uses "noisy" partial derivatives, remains a descent algorithm, we assume that the parameter set can be written as a product $\Theta = \prod_{j\in[\![d]\!]}\Theta_j$ and replace the iterations (3.2.1) (corresponding to Line 5 in Algorithm 1) by

$$\begin{cases} \theta_j^{(t+1)} \leftarrow \mathrm{proj}_{\Theta_j}\left(\theta_j^{(t)} - \beta_j \tau_{\epsilon_j}\left(\widehat{g}_j(\theta^{(t)})\right)\right) & \text{if } j = j_t \\ \theta_j^{(t+1)} \leftarrow \theta_j^{(t)} & \text{otherwise,} \end{cases} \tag{3.5.1}$$

where proj_{Θ_j} is the projection onto Θ_j and τ_ϵ is the soft-thresholding operator given by $\tau_\epsilon(x) = \mathrm{sign}(x)(|x|-\epsilon)_+$ with $(x)_+ = \max(x,0)$. In Theorem 3.3 below we use $\epsilon_j = \epsilon_j(\delta)$, the j-th coordinate of the error vector from Definition 3.1, which is instantiated for each robust estimator in Section 3.3. Since it depends on the moment $m_{\alpha,j}$, it is not observable, so we propose in Lemma 3.6 from Section 3.8.1 an observable upper bound deviation for it based on `MOM`.

This use of soft-thresholding of the partial derivatives can be understood as a form of partial derivatives (or gradient) clipping. However, note that it is rather a theoretical artifact than something to use in practice (we never use τ_ϵ in our numerical experiments from Section 3.6

below). Indeed, the operator τ_ϵ naturally appears for the following simple reason: consider a convex L-smooth scalar function $f : \mathbb{R} \to \mathbb{R}$ with derivative $g(x) := f'(x)$. An iteration of gradient descent from x_0 uses an increment δ that minimizes the right-hand side of the following inequality:

$$f(x_0 + \delta) \leqslant Q(\delta, x_0) := f(x_0) + \delta g(x_0) + \frac{L}{2}\delta^2,$$

namely $\arg\min_\delta Q(\delta, x_0) = -g(x_0)/L$ leading to the iterate $x_0 - g(x_0)/L$ with ensured improvement of the objective. In our context, $g(x)$ is unknown and we use an estimator $\widehat{g}(x)$ satisfying $|\widehat{g}(x) - g(x)| \leqslant \epsilon$ with a large probability. Taking this uncertainty into account leads to the upper bound

$$f(x_0 + \delta) \leqslant \widetilde{Q}(\delta, x_0) := f(x_0) + \delta\widehat{g}(x_0) + \frac{L}{2}\delta^2 + \epsilon|\delta|,$$

and, after projection onto the parameter set, to the iteration (3.5.1) since $\arg\min_\delta \widetilde{Q}(\delta, x_0) = x_0 - \tau_\epsilon(\widehat{g}(x_0))/L$, with guaranteed decrease of the objective.

The clipping of partial derivatives is unnecessary in the strongly convex case since each iteration translates into a contraction of the excess risk, so that the degradations caused by the gradient errors remain controlled (see the proof of Theorem 3.1). No such contraction can be established without strong convexity, and clipping prevents gradient errors to accumulate uncontrollably.

Theorem 3.3. *Grant Assumptions 3.1 and 3.3 with $\Theta = \prod_{j\in[\![d]\!]} \Theta_j$. Let $\theta^{(T)}$ be the output of Algorithm 1 where we replace iterations* (3.2.1) *by* (3.5.1) *with step-sizes $\beta_j = 1/L_j$, an initial iterate $\theta^{(0)} \in \Theta$, uniform coordinates sampling $p_j = 1/d$ and estimators of the partial derivatives with error vector $\epsilon(\cdot)$. Then, we have with probability at least $1 - \delta$*

$$\mathbb{E}\big[R(\theta^{(T)})\big] - R^\star \leqslant \frac{d}{T+1}\bigg(\sum_{j\in[\![d]\!]} \frac{L_j}{2}\big(\theta_j^{(0)} - \theta_j^\star\big)^2 + R(\theta^{(0)})\bigg) + \frac{2\|\epsilon(\delta)\|_2}{T+1}\sum_{t=0}^{T}\|\theta^{(t)} - \theta^\star\|_2,$$

where the expectation is w.r.t the sampling of the coordinates. Moreover, we have

$$\big\|\theta^{(t)} - \theta^\star\big\|_2 \leqslant \|\theta^{(t-1)} - \theta^\star\|_2$$

with the same probability, for all $t \in [\![T]\!]$.

The proof of Theorem 3.3 is given in Section 3.9 and is based on the proof of Theorem 5 from [341] and Theorem 1 from [401] while managing noisy partial derivatives. The optimization error term vanishes at a sublinear $1/T$ rate and is initially of order $R(\theta^{(0)})$ plus the potential $\Phi(\theta) = \sum_{j=1}^d L_j(\theta_j - \theta_j^\star)^2/2$ which is instrumental in the proof. Notice that $\|\epsilon(\delta)\|_2$ appears without the square which translates into "slow" $1/\sqrt{n}$ rates instead of "fast" $1/n$ rates achieved in Section 3.2. This degradation is an unavoidable consequence of the loss of strong convexity of the risk [408].

3.6 Numerical Experiments

The theoretical results of Sections 3.2, 3.3 and 3.5 can be applied to multiple supervised linear learning problems, with guaranteed robustness to both heavy-tailed data and outliers. We perform experiments confirming these properties for several tasks (regression, binary classification and multi-class classification) on multiple data sets comparing with a number of baselines including the state-of-the-art.

3.6.1 Algorithms

We compare our methods with several baselines among the following set of algorithms. For all algorithms, we use, unless specified otherwise, the least-squares loss for regression, and the logistic loss for classification (both for binary and multiclass problems, using the multiclass logistic loss). All considered algorithms can be used easily in a few lines of `Python` code with our library called `linlearn`, open-sourced under the BSD-3 License on `GitHub` and available here: `https://github.com/linlearn/linlearn`. This library follows the API conventions of `scikit-learn` [355].

CGD algorithms: `MOM`, `CH`, `TM` and `CGD ERM`. The `MOM`, `CH` and `TM` algorithms are the variants of robust CGD (Algorithm 1) respectively based on the median-of-means, trimmed mean and Catoni-Holland estimators introduced in Section 3.3. We also include `CGD ERM` which is CGD using a standard mean as estimator.

GD algorithms: `ERM`, `LLM`, `HG`, `GMOM`, `CH GD` and `Oracle`. These are all GD algorithms using different estimators of the gradient. `ERM` uses a non-robust gradient based on a simple mean. `LLM` corresponds to Algorithm 1 from [262]. It uses a MOM estimation of the risk and performs GD using gradients computed as the mean of the sample gradients from the block corresponding to the median of the risk. `HG` is Algorithm 2 from [364], called Huber Gradient Estimator, which uses recursive SVD decompositions and truncations to compute a robust gradient. `GMOM` is Algorithm 3 from [364], which estimates gradients using a geometric MOM (based on the geometric median). `CH GD` is the robust GD algorithm from [191], which uses gradients computed as coordinate-wise `CH` estimators. We consider also `Oracle`, which is GD performed with "oracle" gradients, namely the gradient of the unobserved true risk (only available for linear regression experiments using simulated data).

Extra algorithms: `RANSAC`, `HUBER` and `LAD`. We also include the following algorithms. For regression, we consider `RANSAC` [148], using the implementation available in the `scikit-learn` library [355]. `HUBER` stands for ERM learning with the modified Huber loss [452] for classification and Huber loss [349] for regression. `LAD` is ERM learning using the least absolute deviation loss [139], namely regression using the mean absolute error instead of least-squares.

3.6.2 Regression on simulated data

We consider the following simulation setting for linear regression with the square loss. We generate features $X \in \mathbb{R}^d$ with $d = 5$ with a non-isotropic Gaussian distribution with covariance matrix Σ and labels $Y = X^\top \theta^\star + \xi$ for a fixed $\theta^\star \in \mathbb{R}^d$ and simulated noise ξ. Since all distributions are known in this setting, we can compute the true risk and true gradients (used in `Oracle`).

We consider the following settings: (a) ξ is centered Gaussian; (b) ξ is Student with $\nu = 2.1$ degrees of freedom (heavy-tailed noise). In the remaining settings (c), (d), (e) and (f), ξ is as in (b) but 1% of the data is replaced by outliers as follows. For case (*c*), $X \in \mathbb{R}^5$ is replaced by a constant equal to $\lambda_{\max}(\Sigma)$ (largest eigenvalue of Σ) and labels are replaced by $2y_{\max}$ with $y_{\max} = \max_{i \in \mathcal{I}} |y_i|$; for (d) we do the same as (c) and multiply labels by -1 with probability $1/2$; for (e) we sample $X = 10\lambda_{\max}(\Sigma)v + Z$ where $v \in \mathbb{R}^5$ is a fixed unit vector and Z is a standard Gaussian vector and labels are i.i.d. Bernoulli random variables; finally for (f) we sample $X = 10\lambda_{\max}(\Sigma)V$ where V is uniform on the unit sphere and labels $y = y_{\max} \times (\varepsilon + U)$ where ε is a Rademacher variable and U is uniform in $[-1/5, 1/5]$.

For this experiment, we fix the parameters of the robust partial derivative estimators using the confidence level $\delta = 0.01$ and the number of outliers for `MOM` and `TM`. We report, for all algorithms and settings (a)-(f), the evolution of the square loss (y-axis) along the iterations (x-axis, corresponding to cycles for CGD and iterations for GD). The results are averaged over 30 repetitions.

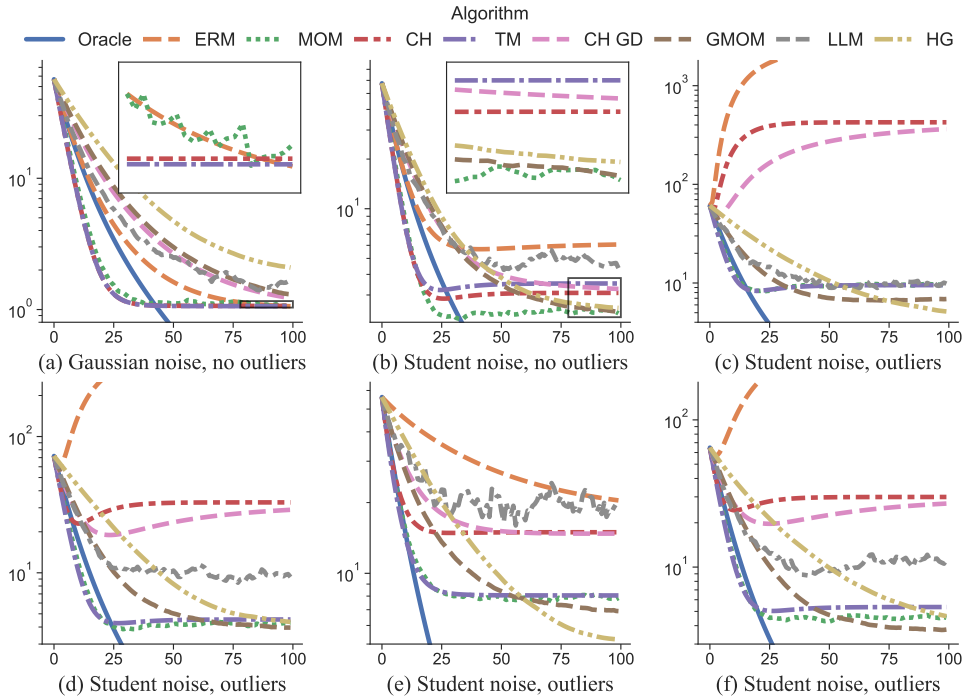

Figure 3.2: Excess-risk for the square loss (y-axis) against iterations (x-axis) for all the considered algorithms in the simulation settings (a)-(c) (top row) and (d)-(f) (bottom row). We zoom-in the last iterations for settings (a) and (b) to improve readability.

We observe that CGD-based algorithms generally converge faster than GD-based ones, independently of the quality of the optimum found. For setting (a), the final performance of all algorithms is roughly similar to that of `ERM` (as expected since the data are neither heavy-tailed nor corrupted) except for `LLM` and `HG` that converge slowly. For setting (b), the performance of `ERM` degrades visibly. Among robust methods, a slight advantage is observed when a robust vector mean is used as opposed to coordinatewise estimators. Though, `MOM` seems to be an exception to this rule. Different behaviours manifest in settings (c)-(f). We observe that `ERM` and Catoni-Holland estimators (`CH` and `CH GD`) are generally the most sensitive to outliers, especially in setting (c) where corrupted samples are introduced in a single-direction. The best final solutions are often found by `GMOM` and `HG`. This is not surprising since the latter is designed to handle corrupted data and the former is far from its breakdown point with only 1% corruption. Nonetheless, we also observe that `MOM` and `TM` consistently show comparable performance. In particular, for settings (d) and (f), corrupted samples are introduced in multiple directions and the performance gap between `GMOM`/`HG` and `MOM`/`TM` is small. Note that `MOM`/`TM` always converge faster. Finally, while `LLM` seems robust to heavy tails and outliers, its use of a median mini-batch and vanishing steps makes it unstable and often prevents it from converging to a good minimum compared to other algorithms.

3.6.3 Classification on real data sets

We consider classification tasks (binary and multiclass) on several data sets from the UCI Machine Learning Repository [136]. We use the logistic loss for binary and multiclass classification problems. For k-class problems with $k > 2$, the parameter θ is a $d \times k$ matrix and CGD is performed block-wise along the class axis. In this case, a CGD cycle performs again d iterations, one for each feature coordinate, each time updating the k associated model weights (a form of block coordinate gradient descent, see [39] for arguments in favor of this approach).

For each data set, we corrupt an increasing random fraction of samples with uninformative outliers or heavy-tailed noise. Each algorithm is hyper-optimized using cross-validation over an appropriate grid of hyper-parameters, see Section 3.8.2 for further details. Subsequently, we train each algorithm with optimal hyper-parameters 10 times over to account for the methods' randomness (most procedures appear to be quite stable across runs) and we finally report in Figure 3.3 the median accuracy obtained on a 15% test-set (y-axis) for each data set, corruption level (x-axis) and algorithm.

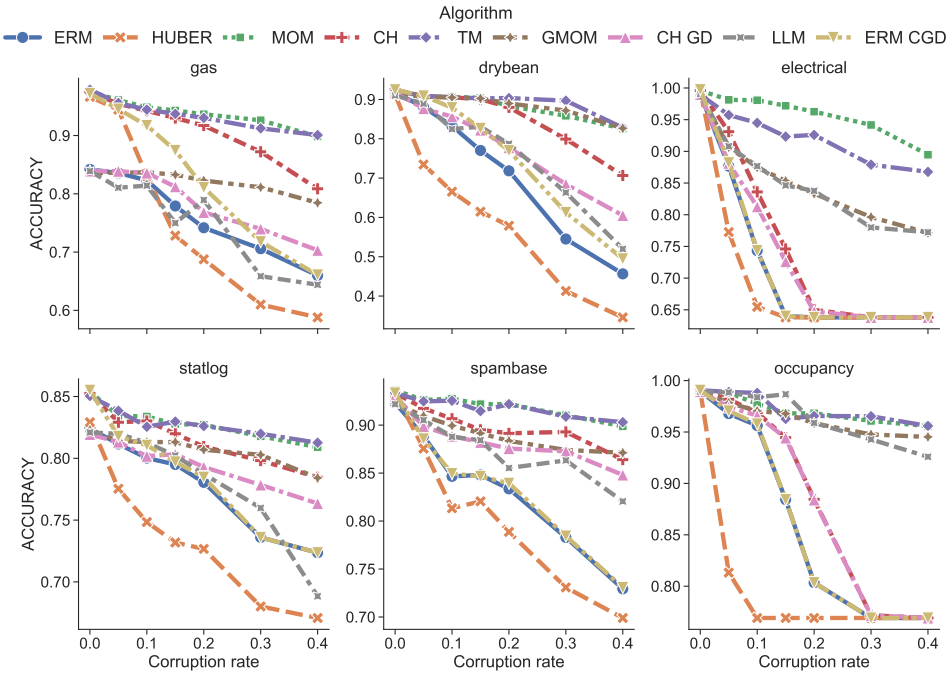

Figure 3.3: Test accuracy (y-axis) against the proportion of corrupted samples (x-axis) for six data sets and the considered algorithms.

First, we note that better optima can sometimes be found using CGD over GD as can be seen by comparing `ERM` and `ERM CGD` on the *gas* and *statlog* data sets with zero corruption. This is also apparent when corruption is present through the fact that `CH` often outperforms `CH GD`. Unsurprisingly, the accuracy of algorithms deteriorates with increasing corruption. In particular, fast degradations occur for `ERM` and `HUBER` which are not intended to handle corrupted covariates. The best performances are generally achieved by `TM` and `MOM` which only lose a minimal fraction of their accuracy to corruption. Although `CH` has no theoretical guarantees against corruption, we see that it is fairly robust on many data sets, especially at low corruption rates. However, its performance inevitably degrades beyond 20% corruption. The most competitive baseline is `GMOM` which manages to match the performance of `TM` and `MOM` on certain instances but seems to generally lag behind as a GD based algorithm. Finally, `LLM` fails to provide a competitive baseline in most cases and suffers from unsteady performance across data sets.

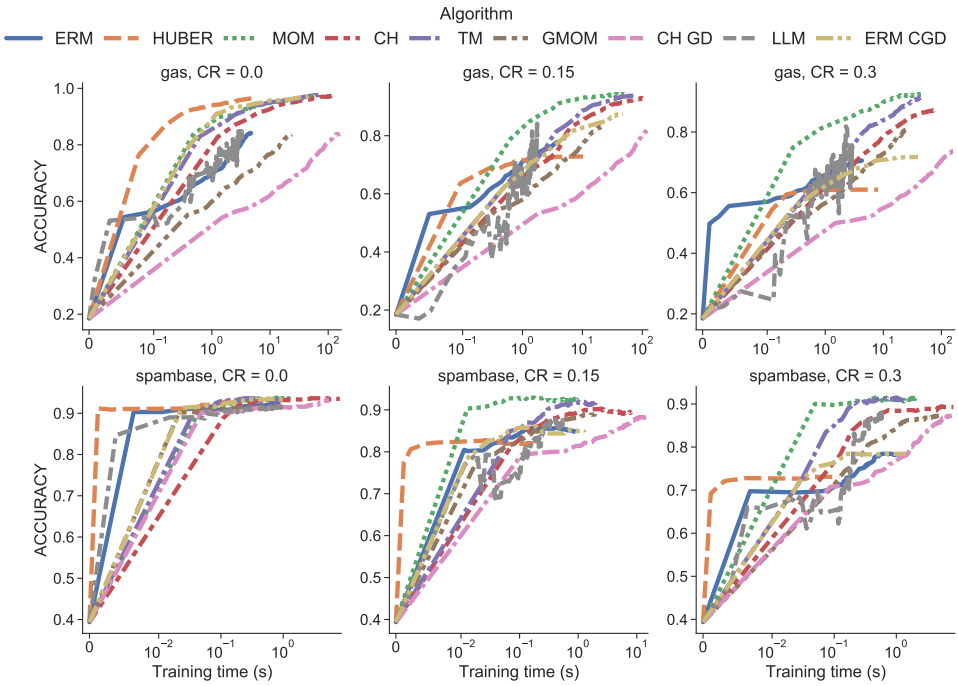

Figure 3.4: Test accuracy (y-axis) against computation time (x-axis) along training iterations on two data sets (rows) for 0% corruption (first column), 15% corruption (middle column) and 30% corruption (last column). Note the log scale on the x-axis.

In order to illustrate the computational performance of each method, we report in Figure 3.4 the test accuracy (y-axis) against the training time (x-axis) along iterations of each algorithm for two data sets (rows) and 0%, 15% and 30% corruption (resp. first, middle and last column). In all situations, standard methods such as `ERM` and `HUBER` run the fastest due to the absence of computational overhead. However, they only reach poor quality optima when data is corrupted as opposed to robust methods. The results of Figure 3.4 concur with Figure 3.1 showing `MOM` to be the fastest and `CH` the slowest CGD algorithm. We also observe that `MOM` and `TM` are clear favourites in terms of final performance and convergence speed, especially when corruption is present. Unsurprisingly, we observe that the combination of GD with the Catoni-Holland estimator in `CH GD` results in the slowest method in most cases. In comparison, `GMOM` is a faster alternative whose speed varies between data sets. This may be explained by the varying number of features and distribution of the data sets affecting the vector median computations. Finally, we see that although `LLM` is among the fastest methods (as seen for 0% corruption), its iteration lacks stability and is visibly affected by corruption.

3.6.4 Regression on real data sets

We consider the same experimental setting (data corruption and hyper-optimization of algorithms) as in Section 3.6.3 on regression data sets from the UCI Machine Learning Database, see Section 3.8.2 for details. We use the square loss for training and the mean squared error (MSE) as test metric, except for `HUBER`, `RANSAC` and `LAD` which are trained differently. We report the results in Figures 3.5 and 3.6. Figure 3.5 shows the test MSE (y-axis) against the corruption rate (x-axis) for several data sets and algorithms while Figure 3.6 displays the test MSE against the training time analogously to Figure 3.4. Note that only final performance and total training time are shown for `RANSAC`, `HUBER` and `LAD` on Figure 3.6. This is because they were run using

`scikit-learn`'s implementation which does not give access to training history. We observe on

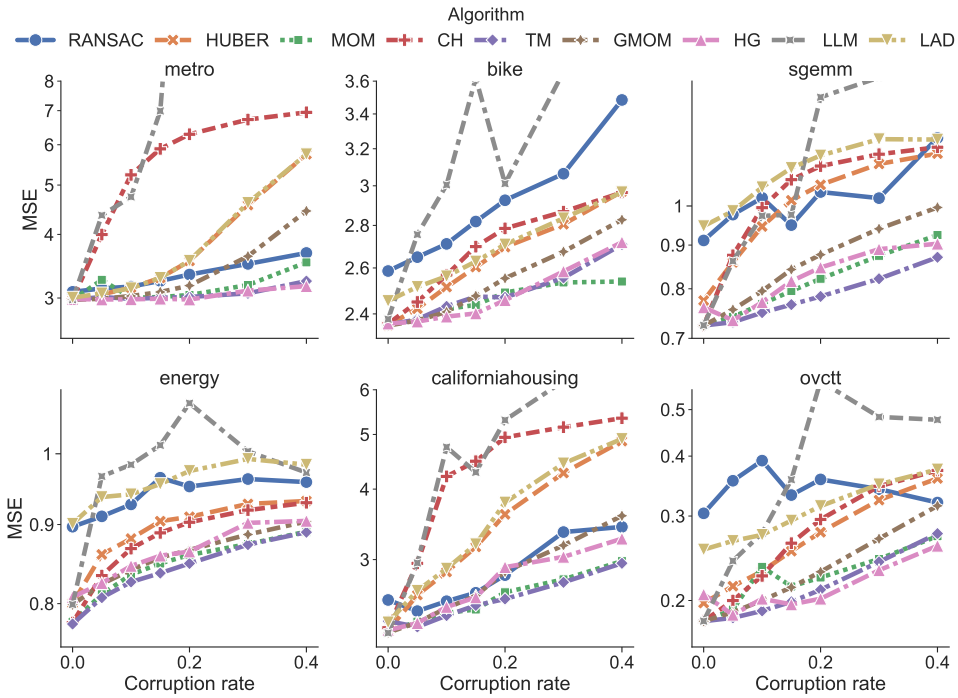

Figure 3.5: Mean squared error (y-axis) against the proportion of corrupted samples (x-axis) for six data sets and the considered algorithms.

Figure 3.5 that `HUBER` and `LAD` often achieve similar performance as they both optimize ℓ_1 objectives. However, `HUBER` finds more precise optima in many cases at low corruption rates. This may be attributed to the quadratic nature of its loss function around zero. Poor performance at low corruption levels is also observed for `RANSAC` in most cases. The latter appears to be somewhat resilient to corruption, suffering limited performance degradation at increasing levels. However, like `LLM`, `RANSAC` displays fluctuating and non competitive results. While `CH` turns out to be especially fragile to corruption on the regression task, the other CGD algorithms `TM` and `MOM` generally secure the best scores. Close competition and sometimes improved performance is shown by `HG` and `GMOM` which prove to effectively filter out corruption, although `GMOM` seems less reliable at higher levels. Furthermore, Figure 3.6 shows that the robustness of `HG` and `GMOM` comes at a significantly higher computational price, especially for `HG` whose running time is slower by orders of magnitude on some data sets and outright prohibitive on others.

As for the remaining algorithms, Figure 3.6 again shows fast convergence for CGD methods with good final performances for `MOM` and `TM`. The iteration of `LLM` is similarly swift but severely destabilized by corruption. Finally, `LAD`, `HUBER` and `RANSAC` sometimes offer short runtimes but lack robustness to corruption.

Our numerical experiments confirm that robust CGD algorithms (`TM` and `MOM`) offer a good compromise between statistical accuracy, robustness and computational cost.

3.7 Conclusion

In this paper, we introduce new robust algorithms for supervised learning by combining CGD with several robust partial derivative estimators. We derive convergence results for several variants of CGD with noisy partial derivatives and prove deviation bounds for all the considered estimators

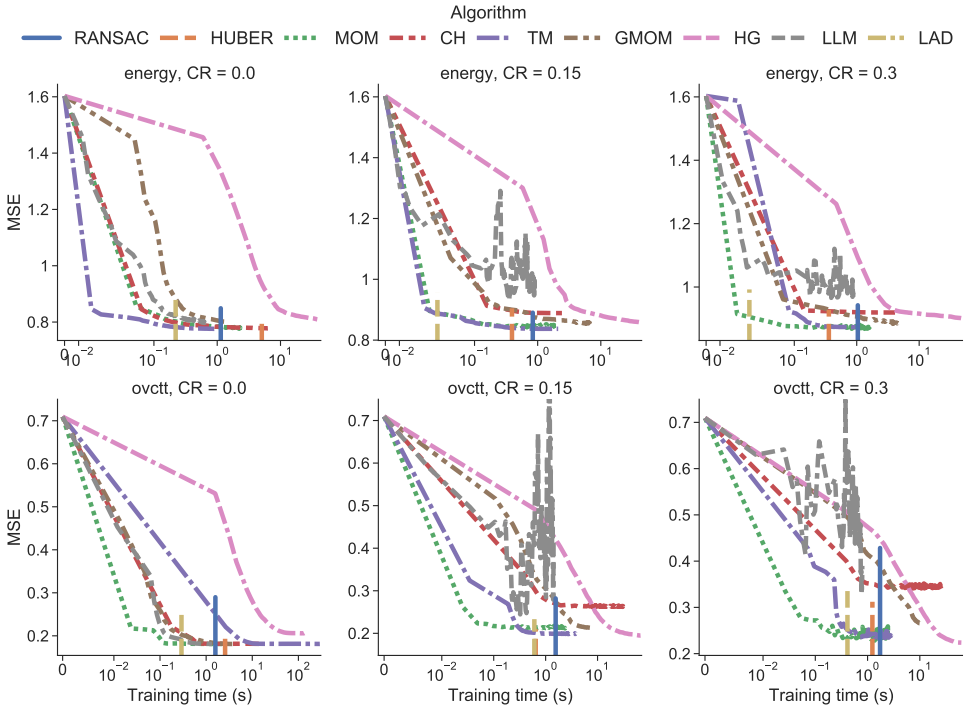

Figure 3.6: Mean squared error (y-axis) against computation time (x-axis) along training iterations on two data sets (rows) for 0% corruption (first column), 15% corruption (middle column) and 30% corruption (last column).

under minimal moment assumptions, including cases with infinite variance and the presence of arbitrary outliers (except for the `CH` estimator). This leads to very robust learning algorithms, with a numerical cost comparable to that of non-robust approaches based on empirical risk minimization, since it lets us bypass the need of a robust *vector mean* and allows to update model weights immediately using a robust estimator of a *single partial derivative* only. This is substantiated by our numerical experiments which illustrate the good compromise offered by our approach between statistical accuracy, robustness and computational cost. Perspectives include robust learning algorithms in high dimension, achieving sparsity-aware generalization bounds, which is beyond the scope of this paper, since it would require different algorithms based on methods such as mirror descent with an appropriately chosen divergence, see for instance [401, 223].

3.8 Supplementary theoretical results and details on experiments

3.8.1 Theoretical supplements

The Lipschitz constants L_j are unknown

The step-sizes $(\beta_j)_{j\in[\![d]\!]}$ used in Theorems 3.1 and 3.2 are given by $\beta_j = 1/L_j$, where the Lipschitz constants L_j are defined by (3.2.3). This makes them non-observable, since they depend on the unknown distribution of the non-corrupted features P_{X_i} for $i \in \mathcal{I}$. We cannot use line-search [10] here, since it requires to evaluate the objective $R(\theta)$, which is unknown as well. In order to provide theoretical guarantees similar to that of Theorem 3.1 without knowing $(L_j)_{j=1}^d$, we use

the following approach. First, we use the upper bound

$$U_j := \gamma \mathbb{E}\big[(X^j)^2\big] \geqslant L_j, \tag{3.8.1}$$

which holds under Assumption 3.1 and estimate $\mathbb{E}[(X^j)^2]$ to build a robust estimator of U_j. In order to obtain an observable upper bound and to control its deviation with a large probability, we introduce the following condition.

Definition 3.2. *We say that a real random variable Z satisfies the L^ζ-L^ξ condition with constant $C \geqslant 1$ whenever it satisfies*

$$\big(\mathbb{E}\big[|Z - \mathbb{E}Z|^\zeta\big]\big)^{1/\zeta} \leqslant C\big(\mathbb{E}\big[|Z - \mathbb{E}Z|^\xi\big]\big)^{1/\xi}. \tag{3.8.2}$$

Using this condition, we can use the `MOM` estimator to obtain a high probability upper bound on $\mathbb{E}[(X^j)^2]$ as stated in the following lemma.

Lemma 3.5. *Grant Assumption 3.2 with $\alpha \in (0, 1]$ and suppose that for all $j \in [\![d]\!]$, the variable $(X^j)^2$ satisfies the $L^{(1+\alpha)}$-L^1 condition with a known constant C. For any fixed $j \in [\![d]\!]$, let $\widehat{\sigma}_j^2$ be the* `MOM` *estimator of $\mathbb{E}[(X^j)^2]$ with K blocks. If $|\mathcal{O}| \leqslant K/12$, we have*

$$\mathbb{P}\Big[\Big(1 - 12^{1/(1+\alpha)} C\Big(\frac{K}{n}\Big)^{\alpha/(1+\alpha)}\Big)^{-1} \widehat{\sigma}_j^2 \leqslant \mathbb{E}[(X^j)^2]\Big] \leqslant \exp(-K/18).$$

If we fix a confidence level $\delta \in (0, 1)$ and choose $K := \lceil 18 \log(1/\delta) \rceil$, we have

$$\Big(1 - 216^{1/(1+\alpha)} C\Big(\frac{\log(1/\delta)}{n}\Big)^{\alpha/(1+\alpha)}\Big)^{-1} \widehat{\sigma}_j^2 > \mathbb{E}[(X^j)^2]$$

with a probability larger than $1 - \delta$.

The proof of Lemma 3.5 is given in Section 3.9. Denoting $\widehat{U}_j$ the upper bounds it provides on $\mathbb{E}[(X^j)^2]$, we can readily bound the Lipschitz constants as $L_j \leqslant \gamma \widehat{U}_j$ which leads to the following statement.

Corollary 3.2. *Grant the same assumptions as in Theorem 3.1 and Proposition 3.1. Suppose additionally that for all $j \in [\![d]\!]$, the variable $(X^j)^2$ satisfies the $L^{(1+\alpha)}$-L^1 condition with a known constant C and fix $\delta \in (0, 1)$. Let $\theta^{(T)}$ be the output of Algorithm 1 with step-sizes $\widehat{\beta}_j = 1/\overline{L}_j$ where $\overline{L}_j := \gamma \widehat{U}_j$ and $\widehat{U}_j$ are the upper bounds from Lemma 3.5 with confidence $\delta/2d$, an initial iterate $\theta^{(0)}$, importance sampling distribution $p_j = \overline{L}_j / \sum_{k \in [\![d]\!]} \overline{L}_k$ and estimators of the partial derivatives with error vector $\epsilon(\cdot)$. Then, we have*

$$\mathbb{E}\big[R(\theta^{(T)})\big] - R^\star \leqslant (R(\theta^{(0)}) - R^\star)\Big(1 - \frac{\lambda}{\sum_{j \in [\![d]\!]} \overline{L}_j}\Big)^T + \frac{1}{2\lambda}\big\|\epsilon(\delta/2)\big\|_2^2 \tag{3.8.3}$$

with probability at least $1 - \delta$.

The proof of Corollary 3.2 is given in Section 3.9. It is a direct consequence of Theorem 3.1 and Lemma 3.5 and shows that an upper bound similar to that of Theorem 3.1 can be achieved with *observable* step-sizes. One may argue that the $L^{(1+\alpha)}$-L^1 condition simply bypasses the difficulty of deriving an observable upper bound by arbitrarily assuming that a ratio of moments is observed. However, we point out that a hypothesis of this nature is indispensable to obtain bounds such as the one above (alternatively, consider a real random variable with an infinitesimal mass drifting towards infinity). In fact, the $L^{(1+\alpha)}$-L^1 condition is much weaker than the requirement of boundedness (with known range) common to most known empirical bounds [302, 14, 320].

Observable upper bound for the moment $m_{\alpha,j}$

Since the moment $m_{\alpha,j}$, it is not observable, so we propose in Lemma 3.6 below an observable upper bound deviation for it based on MOM. Let us introduce now a robust estimator $\widehat{m}^{\texttt{MOM}}_{\alpha,j}(\theta)$ of the unknown moment $m_{\alpha,j}(\theta)$ using the following "two-step" MOM procedure. First, we compute $\widehat{g}^{\texttt{MOM}}_j(\theta)$, the MOM estimator of $g_j(\theta)$ with K blocks given by (3.3.3). Then, we compute again a MOM estimator on $|g^i_j(\theta) - \widehat{g}^{\texttt{MOM}}_j(\theta)|^{1+\alpha}$ for $i \in [\![n]\!]$, namely

$$\widehat{m}^{\texttt{MOM}}_{\alpha,j}(\theta) := \text{median}\big(\widehat{m}^{(1)}_{\alpha,j}(\theta), \dots, \widehat{m}^{(K)}_{\alpha,j}(\theta)\big), \tag{3.8.4}$$

where

$$\widehat{m}^{(k)}_{\alpha,j}(\theta) := \frac{1}{|B_k|} \sum_{i \in B_k} \big|g^i_j(\theta) - \widehat{g}^{\texttt{MOM}}_j(\theta)\big|^{1+\alpha},$$

using uniformly sampled blocks $B_1, \dots, B_K$ of equal size that form a partition of $[\![n]\!]$.

Lemma 3.6. *Grant Assumptions 3.1 and 3.2 with $\alpha \in (0, 1]$ and suppose that for all $j \in [\![d]\!]$ and $\theta \in \Theta$ the partial derivatives $\ell'(X^\top\theta, Y)X^j$ satisfy the $L^{(1+\alpha)^2}$-$L^{(1+\alpha)}$ condition with known constant C for any $j \in [\![d]\!]$ (see Definition 3.2). Then, if $|\mathcal{O}| \leqslant K/12$, we have*

$$\mathbb{P}\big[\widehat{m}^{\texttt{MOM}}_{\alpha,j}(\theta) \leqslant (1-\kappa) m_{\alpha,j}(\theta)\big] \leqslant 2\exp(-K/18)$$

where $\kappa = \epsilon + 24(1+\alpha)\big(\frac{(1+\epsilon)K}{n}\big)^{\alpha/(1+\alpha)}$ and $\epsilon = (24(1 + C^{(1+\alpha)^2}))^{1/(1+\alpha)}\big(\frac{K}{n}\big)^{\alpha/(1+\alpha)}$.

The proof of Lemma 3.6 is given in Section 3.9.

3.8.2 Experimental details

We provide in this section supplementary information about the numerical experiments conducted in Section 3.6.

Data sets

The main characteristics of the data sets used from the UCI repository are given in Table 3.3 and their direct URLs are given in Table 3.4.

Data set	# Samples	# Features	# Categorical	# Classes
statlog	6,435	36	0	6
spambase	4,601	57	0	2
electrical	10,000	13	0	2
occupancy [64]	20,560	5	0	2
gas [431]	13,910	128	0	6
drybean [239]	13,611	16	0	7
energy [65]	19,735	27	0	-
bike [146]	17,379	10	5	-
metro	48,204	6	1	-
sgemm [20]	241,600	14	0	-
ovctt	68,784	20	2	-
californiahousing	20,640	8	0	-

Table 3.3: Main characteristics of the data sets used in experiments, including number of samples, number of features, number of categorical features and number of classes.

Data set	URL
statlog	https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29
spambase	https://archive.ics.uci.edu/ml/datasets/spambase
electrical	https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+
occupancy	https://archivehttps://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+
gas	https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset
drybean	https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset
energy	https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction
bike	https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset
metro	https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume
sgemm	https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance
ovctt	https://archive.ics.uci.edu/ml/datasets/Online+Video+Characteristics+and+Transcoding+Time+Dataset
californiahousing	loaded from scikitlearn.datasets

Table 3.4: The URLs of all the data sets used in the paper, giving direct download links and supplementary details.

Data corruption

For a given corruption rate η, we obtain a corrupted version of a data set by replacing an η-fraction of its samples with uninformative elements. For a data set of size n we choose $\mathcal{O} \subset [\![n]\!]$ which satisfies $|\mathcal{O}| = \eta n$ up to integer rounding. The corruption is applied prior to any prepro-

cessing except in the regression case where label scaling is applied before. The affected subset is chosen uniformly at random. Since many data sets contain both continuous and categorical data features, we distinguish two different corruption mechanisms which we apply depending on their nature. The labels are corrupted as continuous or categorical values when the task is respectively regression or classification. Denote $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{n\times(d+1)}$ the data matrix with the vector of labels added to its columns. Let $\widetilde{J} \subset [\![d+1]\!]$ denote the index of continuous columns, we compute $\widehat{\mu}_j$ and $\widehat{\sigma}_j$ their empirical means and standard deviations respectively for $j \in \widetilde{J}$. We also sample a random unit vector u of size $|\widetilde{J}|$.

- For categorical feature columns, for each corrupted index $i \in \mathcal{O}$, we replace $\boldsymbol{X}_{i,j}$ with a uniformly sampled value among $\{\boldsymbol{X}_{\bullet,j}\}$ i.e. among the possible modalities of the categorical feature in question.
- For continuous features, for each corrupted index $i \in \mathcal{O}$, we replace $\boldsymbol{X}_{i,\widetilde{J}}$ with equal probability with one of the following possibilities:
  - a vector ξ sampled coordinatewise according to $\xi_j = r_j + 5\widehat{\sigma}_j\nu$ where r_j is a value randomly picked in the column $\boldsymbol{X}_{\bullet,j}$ and ν is a sample from the Student distribution with 2.1 degrees of freedom.
  - a vector ξ sampled coordinatewise according to $\xi_j = \widehat{\mu}_j + 5\widehat{\sigma}_j u_j + z$ where z is a standard gaussian.
  - a vector ξ sampled according to $\xi = \widehat{\mu} + 5\widehat{\sigma} \otimes w$ where w is a uniformly sampled unit vector.

Preprocessing

We apply a minimal amount of preprocessing to the data before applying the considered learning algorithms. More precisely, categorical features are one-hot encoded while centering and standard scaling is applied to the continuous features.

Parameter hyper-optimization

We use the `hyperopt` library to find optimal hyper-parameters for all algorithms. For each data set, the available samples are split into training, validation and test sets with proportions $70\%, 15\%, 15\%$. Whenever corruption is applied, it is restricted to the training set. We run 50 rounds of hyper-parameter optimization which are trained on the training set and evaluated on the validation set. Then, we report results on the test set for all hyper-optimized algorithms. For each algorithm, the hyper-parameters are tried out using the following sampling mechanism (the one we specify to `hyperopt`):

- `MOM`, `GMOM`, `LLM`: we optimize the number of blocks K used for the median-of-means computations. This is done through a `block_size` $= K/n$ hyper-parameter chosen with log-uniform distribution over $[10^{-5}, 0.2]$
- `CH` and `CH GD`: we optimize the confidence δ used to define the `CH` estimator's scale parameter (see Equation (3.3.10)) chosen with log-uniform distribution over $[e^{-10}, 1]$
- `TM`, `HG`: we optimize the percentage used for trimming uniformly in $[10^{-5}, 0.3]$
- `RANSAC`: we optimize the value of the `min_samples` parameter in the scikit-learn implementation, chosen as $4 + m$ with m an integer chosen uniformly in $[\![100]\!]$
- `HUBER`: we optimize the `epsilon` parameter in the scikit-learn implementation chosen uniformly in $[1.0, 2.5]$

3.9 Proofs

3.9.1 Proofs for Section 3.2

Proof of Theorem 3.1

This proof follows, with minor modifications, the proof of Theorem 1 from [442]. Using Definition 3.1 , we obtain

$$\mathbb{P}[\mathcal{E}] \geqslant 1-\delta \quad \text{where} \quad \mathcal{E} := \big\{\forall j \in [\![d]\!], \quad \forall t \in [T], \quad \big|\widehat{g}_j(\theta^{(t)}) - g_j(\theta^{(t)})\big| \leqslant \epsilon_j(\delta)\big\}. \tag{3.9.1}$$

Let us recall that e_j stands for the j-th canonical basis of $\mathbb{R}^d$ and that, as described in Algorithm 1, we have

$$\theta^{(t+1)} = \theta^{(t)} - \beta_{j_t}\widehat{g}_t e_{j_t},$$

where we use the notations $\widehat{g}_t = \widehat{g}_{j_t}(\theta^{(t)})$ and $g_t = g_{j_t}(\theta^{(t)})$ and where we recall that $j_1, \dots, j_t$ is a i.i.d sequence with distribution p. We introduce also $\epsilon_j := \epsilon_j(\delta)$. Using Assumption 3.3, we obtain

$$\begin{aligned}
R(\theta^{(t+1)}) &= R\big(\theta^{(t)} - \beta_{j_t}\widehat{g}_t e_{j_t}\big) \\
&\leqslant R(\theta^{(t)}) - \big\langle g(\theta^{(t)}), \beta_{j_t}\widehat{g}_t e_{j_t}\big\rangle + \frac{L_{j_t}}{2}\beta_{j_t}^2\widehat{g}_t^2 \\
&= R(\theta^{(t)}) - \beta_{j_t}g_t^2 - \beta_{j_t}g_t(\widehat{g}_t - g_t) + \frac{L_{j_t}\beta_{j_t}^2}{2}\big(g_t^2 + (\widehat{g}_t - g_t)^2 + 2g_t(\widehat{g}_t - g_t)\big) \\
&= R(\theta^{(t)}) - \beta_{j_t}g_t(1 - L_{j_t}\beta_{j_t})(\widehat{g}_t - g_t) - \beta_{j_t}\Big(1 - \frac{L_{j_t}\beta_{j_t}}{2}\Big)g_t^2 + \frac{L_{j_t}\beta_{j_t}^2}{2}(\widehat{g}_t - g_t)^2 \\
&= R(\theta^{(t)}) - \frac{1}{2L_{j_t}}g_t^2 + \frac{1}{2L_{j_t}}(\widehat{g}_t - g_t)^2 \\
&\leqslant R(\theta^{(t)}) - \frac{1}{2L_{j_t}}g_t^2 + \frac{\epsilon_{j_t}^2}{2L_{j_t}}
\end{aligned} \tag{3.9.2}$$

on the event $\mathcal{E}$, where we used the choice $\beta_{j_t} = 1/L_{j_t}$ and the fact that $|\widehat{g}_t - g_t| \leqslant \epsilon_{j_t}$ on $\mathcal{E}$.

Since $j_1, \dots, j_t$ is a i.i.d sequence with distribution p, we have for any $(j_1, \dots, j_{t-1})$-measurable and integrable function φ that

$$\mathbb{E}_{t-1}\big[\varphi(j_t)\big] = \sum_{j \in [\![d]\!]} \varphi(j)p_j,$$

where we denote for short the conditional expectation $\mathbb{E}_{t-1}[\cdot] = \mathbb{E}_{t-1}[\cdot|j_1, \dots, j_{t-1}]$. So, taking $\mathbb{E}_{t-1}[\cdot]$ on both sides of (3.9.2) leads, whenever $p_j = L_j / \sum_{k=1}^d L_k$, to

$$\mathbb{E}_{t-1}\big[R(\theta^{(t+1)})\big] \leqslant R(\theta^{(t)}) - \frac{1}{2\sum_k L_k}\big\|g(\theta^{(t)})\big\|^2 + \frac{1}{2\sum_k L_k}\Xi,$$

where we introduced $\Xi := \|\epsilon(\delta)\|_2^2$, while it leads to

$$\mathbb{E}_{t-1}\big[R(\theta^{(t+1)})\big] \leqslant R(\theta^{(t)}) - \frac{1}{2L_{\max}d}\big\|g(\theta^{(t)})\big\|^2 + \frac{1}{2dL_{\min}}\Xi$$

whenever $p_j = 1/d$, simply using $L_{\min} \leqslant L_j \leqslant L_{\max}$. In order to treat both cases simultaneously, consider $\bar{L} = \sum_{k=1} L_k$ and $\bar{\epsilon} = \Xi/(2\sum_k L_k)$ whenever $p_j = L_j / \sum_{k=1}^d L_k$ and $\bar{L} = dL_{\max}$ and

$\bar{\epsilon}/(2dL_{\min})$ whenever $p_j = 1/d$ and continue from the inequality

$$\mathbb{E}_{t-1}\big[R(\theta^{(t+1)})\big] \leqslant R(\theta^{(t)}) - \frac{1}{2\bar{L}}\big\|g(\theta^{(t)})\big\|^2 + \bar{\epsilon}.$$

Introducing $\phi_t := \mathbb{E}\big[R(\theta^{(t)})\big] - R^\star$ and taking the expectation w.r.t. all $j_1, \ldots, j_t$ we obtain

$$\phi_{t+1} \leqslant \phi_t - \frac{1}{2\bar{L}}\mathbb{E}\big\|g(\theta^{(t)})\big\|^2 + \bar{\epsilon}. \tag{3.9.3}$$

Using Inequality (3.2.4) with $\theta_1 = \theta^{(t)}$ gives

$$R(\theta_2) \geqslant R(\theta^{(t)}) + \big\langle \nabla R(\theta^{(t)}), \theta_2 - \theta^{(t)} \big\rangle + \frac{\lambda}{2}\big\|\theta_2 - \theta^{(t)}\big\|^2$$

for any $\theta_2 \in \mathbb{R}^d$, so that by minimizing both sides with respect to θ_2 leads to

$$R^\star \geqslant R(\theta^{(t)}) - \frac{1}{2\lambda}\big\|g(\theta^{(t)})\big\|^2$$

namely

$$\phi_t \leqslant \frac{1}{2\lambda}\mathbb{E}\big\|g(\theta^{(t)})\big\|^2,$$

by taking the expectation on both sides. Together with (3.9.3) this leads to the following approximate contraction property:

$$\phi_{t+1} \leqslant \phi_t\Big(1 - \frac{\lambda}{\bar{L}}\Big) + \bar{\epsilon},$$

and by iterating $t = 1, \ldots, T$ to

$$\phi_T \leqslant \phi_0\Big(1 - \frac{\lambda}{\bar{L}}\Big)^T + \frac{\bar{\epsilon}\bar{L}}{\lambda},$$

which allows to conclude the Proof of Theorem 3.1. □

Proof of Theorem 3.2

This proof reuses ideas from [271] and [25] and adapts them to our context where the gradient coordinates are replaced with high confidence approximations. Without loss of generality, we initially assume that the coordinates are cycled upon in the natural order. We condition on the event (3.9.1) which holds with probability $\geqslant 1 - \delta$ as in the proof of Theorem 3.1 and denote $\epsilon_j = \epsilon_j(\delta)$ and $\epsilon_{Euc} = \|\epsilon(\delta)\|$.

Let the iterations be denoted as $\theta^{(t)}$ for $t = 0, \ldots, T$ and $\theta_{i+1}^{(t)} = \theta_i^{(t)} - \beta_{i+1}\widehat{g}(\theta_i^{(t)})_{i+1}e_{i+1}$ for $i = 0, \ldots, d-1$ with $\beta_i = 1/L_i$, $\theta_0^{(t)} = \theta^{(t)}$ and $\theta_d^{(t)} = \theta^{(t+1)}$. With these notations we have

$$R(\theta^{(t)}) - R(\theta^{(t+1)}) = \sum_{i=0}^{d-1} R(\theta_i^{(t)}) - R(\theta_{i+1}^{(t)}).$$

Similarly to (3.9.2) in the proof of Theorem 3.1 we find:

$$R(\theta_i^{(t)}) - R(\theta_{i+1}^{(t)}) \geqslant \frac{1}{2L_{i+1}}\big(g(\theta_i^{(t)})_{i+1}^2 - \epsilon_{i+1}^2\big),$$

leading to

$$R(\theta^{(t)}) - R(\theta^{(t+1)}) \geqslant \sum_{i=0}^{d-1} \frac{1}{2L_{i+1}} g(\theta_i^{(t)})_{i+1}^2 - \frac{1}{2L_{\min}} \sum_{i=0}^{d-1} \epsilon_{i+1}^2. \tag{3.9.4}$$

The following aims to find a relationship between $\sum_{i=0}^{d-1} \frac{1}{2L_{i+1}} g(\theta_i^{(t)})_{i+1}^2$ and $\|g(\theta^{(t)})\|_2^2$ which we do by comparing coordinates. For the first step in a cycle we have $g(\theta^{(t)})_1 = g(\theta_0^{(t)})_1$ because $\theta^{(t)} = \theta_0^{(t)}$. Let $j \in \{1, \dots, d-1\}$, by the Mean Value Theorem, there exists $\gamma_j^{(t)} \in \mathbb{R}^d$ such that we have:

$$\begin{aligned}
&g(\theta^{(t)})_{j+1} = g(\theta^{(t)})_{j+1} - g(\theta_j^{(t)})_{j+1} + g(\theta_j^{(t)})_{j+1} \\
&= \left(\nabla g_{j+1}(\gamma_j^{(t)})\right)^\top \left(\theta^{(t)} - \theta_j^{(t)}\right) + g(\theta_j^{(t)})_{j+1} \\
&= \left[\frac{\partial R(\gamma_j^{(t)})}{\partial_{j+1}\partial_1}, \dots, \frac{\partial R(\gamma_j^{(t)})}{\partial_{j+1}\partial_j}, 0, \dots, 0\right] \left[(\theta^{(t)} - \theta_j^{(t)})_1, \dots, (\theta^{(t)} - \theta_j^{(t)})_j, 0, \dots, 0\right]^\top \\
&\quad + g(\theta_j^{(t)})_{j+1} \\
&= [H_{j+1,1}, \dots, H_{j+1,j}, 0, \dots, 0] \left[\frac{\widehat{g}_1(\theta_0^{(t)})}{L_1}, \dots, \frac{\widehat{g}_j(\theta_{j-1}^{(t)})}{L_j}, 0, \dots, 0\right]^\top + g(\theta_j^{(t)})_{j+1} \\
&= [H_{j+1,1}, \dots, H_{j+1,j}, 0, \dots, 0] \left[\frac{g_1(\theta_0^{(t)}) + \delta_1^{(t)}}{L_1}, \dots, \frac{g_j(\theta_{j-1}^{(t)}) + \delta_j^{(t)}}{L_j}, 0, \dots, 0\right]^\top \\
&\quad + g(\theta_j^{(t)})_{j+1} \\
&= \underbrace{\left[\frac{H_{j+1,1}}{\sqrt{L_1}}, \dots, \frac{H_{j+1,j}}{\sqrt{L_j}}, \sqrt{L_{j+1}}, 0, \dots, 0\right]}_{\widetilde{h}_{j+1}^\top} \underbrace{\left[\frac{g_1(\theta_0^{(t)})}{\sqrt{L_1}}, \dots, \frac{g_d(\theta_{d-1}^{(t)})}{\sqrt{L_d}}\right]^\top}_{\widetilde{g}_t} \\
&\quad + \underbrace{[H_{j+1,1}, \dots, H_{j+1,j}, 0, \dots, 0]}_{h_{j+1}^\top} \left[\frac{\delta_1^{(t)}}{L_1}, \dots, \frac{\delta_d^{(t)}}{L_d}\right]^\top \\
&= \widetilde{h}_{j+1}\widetilde{g}_t + h_{j+1}A^{-1}\delta^{(t)},
\end{aligned}$$

where we introduced the following quantities: $A \in \mathbb{R}^d$ equal to $A = \operatorname{diag}(L_j)_{j=1}^d$, the vector $\delta^{(t)} \in \mathbb{R}^d$ is such that $\delta_j^{(t)} = \widehat{g}(\theta_{j-1}^{(t)})_j - g(\theta_{j-1}^{(t)})_j$ which satisfies $|\delta_j^{(t)}| \leqslant \epsilon_j$, the matrix $H = (h_1, \dots, h_d)^\top$ and $\widetilde{H} = A^{1/2} + HA^{-1/2} = (\widetilde{h}_1, \dots, \widetilde{h}_d)^\top$. In the case $j = 0$ the vector $h_{j+1} = h_1$ is simply zero. This allows us to obtain the following estimation:

$$\begin{aligned}
\left\|g(\theta^{(t)})\right\|^2 &= \sum_{j=1}^d g(\theta^{(t)})_j^2 = \sum_{j=1}^d (\widetilde{h}_j^\top \widetilde{g}_t + h_j^\top A^{-1}\delta^{(t)})^2 \\
&\leqslant \sum_{j=1}^d 2(\widetilde{h}_j^\top \widetilde{g}_t)^2 + 2(h_j^\top A^{-1}\delta^{(t)})^2 = 2\left\|\widetilde{H}\widetilde{g}_t\right\|^2 + 2\left\|HA^{-1}\delta^{(t)}\right\|^2 \\
&\leqslant 2\left\|\widetilde{H}\right\|^2 \left\|\widetilde{g}_t\right\|^2 + \frac{2}{L_{\min}^2}\|H\|^2\epsilon_{Euc}^2 \\
&= 2\|\widetilde{H}\|^2 \sum_{i=0}^{d-1} \frac{1}{L_{i+1}} g(\theta_i^{(t)})_{i+1}^2 + \frac{2}{L_{\min}^2}\|H\|^2\epsilon_{Euc}^2.
\end{aligned} \tag{3.9.5}$$

We can bound the spectral norm $\|\widetilde{H}\|$ as follows:

$$\|\widetilde{H}\|^2 = \|A^{1/2} + HA^{-1/2}\|^2 \leqslant 2\|A^{1/2}{}^2 + 2\|HA^{-1/2}\|^2 \leqslant 2\Big(L_{\max} + \frac{\|H\|^2}{L_{\min}}\Big).$$

For $\|H\|$, we use the coordinate-wise Lipschitz-smoothness in order to find

$$\|H\|^2 \leqslant \|H\|_F^2 = \sum_{j=1}^{d} \|h_j\|^2 \leqslant \sum_{j=1}^{d} \big\|\nabla g_j(\gamma_{j-1}^{(t)})\big\|^2 \leqslant \sum_{j=1}^{d} L_j^2 \leqslant dL_{\max}^2.$$

Combining the previous inequality with (3.9.4) and (3.9.5), we find:

$$\begin{aligned}
&R(\theta^{(t)}) - R(\theta^{(t+1)}) \\
&\geqslant \frac{1}{8L_{\max}(1 + d\frac{L_{\max}}{L_{\min}})}\big\|g(\theta^{(t)})\big\|^2 - \frac{\epsilon_{Euc}^2}{2}\Big((\frac{1}{L_{\min}} + \frac{d\Big(\frac{L_{\max}}{L_{\min}}\Big)^2}{2L_{\max}(1 + d\frac{L_{\max}}{L_{\min}})}\Big) \\
&\geqslant \frac{1}{8L_{\max}(1 + d\frac{L_{\max}}{L_{\min}})}\big\|g(\theta^{(t)})\big\|^2 - \frac{\epsilon_{Euc}^2}{2}\Big(\frac{1}{L_{\min}} + \frac{1}{2L_{\min}}\frac{dL_{\max}/L_{\min}}{1 + d\frac{L_{\max}}{L_{\min}}}\Big) \\
&\geqslant \underbrace{\frac{1}{8L_{\max}(1 + d\frac{L_{\max}}{L_{\min}})}}_{=:\kappa}\big\|g(\theta^{(t)})\big\|^2 - \frac{3}{4L_{\min}}\epsilon_{Euc}^2,
\end{aligned}$$

where the last step uses that $\frac{dL_{\max}/L_{\min}}{1+d\frac{L_{\max}}{L_{\min}}} \leqslant 1$. Using λ-strong convexity by choosing $\theta_1 = \theta^{(t)}$ in inequality (3.2.4) and minimizing both sides w.r.t. θ_2 we obtain:

$$R(\theta^{(t)}) - R^\star \leqslant \frac{1}{2\lambda}\|g(\theta^{(t)})\|^2,$$

which combined with the previous inequality yields the contraction inequality:

$$R(\theta^{(t+1)}) - R^\star \leqslant (R(\theta^{(t)}) - R^\star)(1 - 2\lambda\kappa) + \frac{3}{4L_{\min}}\epsilon_{Euc}^2,$$

and after T iterations we have:

$$R(\theta^{(T)}) - R^\star \leqslant (R(\theta^{(0)}) - R^\star)(1 - 2\lambda\kappa)^T + \frac{3\epsilon_{Euc}^2}{8L_{\min}\lambda\kappa},$$

which concludes the proof of Theorem 3.2. To see that the proof still holds for any choice of coordinates satisfying the conditions in the main claim, notice that the computations leading up to Inequality (3.9.5) work all the same if one were to apply a permutation to the coordinates beforehand.

Convergence of the parameter error

We state and prove a result about the linear convergence of the parameter under strong convexity.

Theorem 3.4. *Grant Assumptions 3.1, 3.3 and 3.4. Let $\theta^{(T)}$ be the output of Algorithm 1 with constant step-size $\beta = \frac{2}{\lambda+L}$, an initial iterate $\theta^{(0)}$, uniform coordinates sampling $p_j = 1/d$ and*

estimators of the partial derivatives with error vector $\epsilon(\cdot)$. Then, we have

$$\mathbb{E}\big\|\theta^{(T)}-\theta^\star\big\|_2 \leqslant \big\|\theta^{(0)}-\theta^\star\big\|_2\Big(1-\frac{2\beta\lambda L}{d(\lambda+L)}\Big)^T+\frac{\sqrt{d}(\lambda+L)}{\lambda L}\big\|\epsilon(\delta)\big\|_2 \tag{3.9.6}$$

with probability at least $1-\delta$, where the expectation is w.r.t. the sampling of the coordinates.

Proof. As in the proof of Theorem 3.1, let $(\widehat{g}_j(\theta))_{j=1}^d$ be the estimators used and introduce the notations

$$\widehat{g}_t=\widehat{g}_{j_t}(\theta^{(t)}) \quad \text{and} \quad g_t=g_{j_t}(\theta^{(t)}).$$

We also condition on the event (3.9.1) which holds with probability $1-\delta$ and use the notations $\epsilon_{Euc}=\|\epsilon(\delta)\|_2$ and $\epsilon_j=\epsilon_j(\delta)$. We denote $\|\cdot\|_{L_2}$ the L_2-norm w.r.t. the distribution over j_t i.e. for a random variable ξ we have $\|\xi\|_{L_2}=\sqrt{\mathbb{E}_{j_t}\|\xi\|^2}$. We compute:

$$\big\|\theta^{(t+1)}-\theta^\star\big\|_{L_2}=\big\|\theta^{(t)}-\beta_{j_t}\widehat{g}_te_{j_t}-\theta^\star\big\|_{L_2}\leqslant\big\|\theta^{(t)}-\beta_{j_t}g_te_{j_t}-\theta^\star\big\|_{L_2}+\big\|\beta_{j_t}(\widehat{g}_t-g_t)\big\|_{L_2}. \tag{3.9.7}$$

We first treat the first term of (3.9.7), in the case of uniform sampling with equal step-sizes $\beta_j=\beta$ we have:

$$\big\|\theta^{(t)}-\beta g_te_{j_t}-\theta^\star\big\|^2=\big\|\theta^{(t)}-\theta^\star\big\|^2+\beta^2g_t^2-2\beta\big\langle g_te_{j_t},\theta^{(t)}-\theta^\star\big\rangle.$$

By taking the expectation w.r.t. the random coordinate j_t we find:

$$\begin{aligned}
&\big\|\theta^{(t)}-\beta g_te_{j_t}-\theta^\star\big\|_{L_2}^2=\mathbb{E}\big\|\theta^{(t)}-\beta g_te_{j_t}-\theta^\star\big\|^2\\
&=\mathbb{E}\big\|\theta^{(t)}-\theta^\star\big\|^2+\frac{\beta^2}{d}\mathbb{E}\big\|g(\theta^{(t)})\big\|^2-2\frac{\beta}{d}\mathbb{E}\big\langle g(\theta^{(t)}),\theta^{(t)}-\theta^\star\big\rangle\\
&=\mathbb{E}\|\theta^{(t)}-\theta^\star\|^2+\Big(\frac{\beta}{d}\Big)^2\mathbb{E}\|g(\theta^{(t)})\|^2-2\frac{\beta}{d}\mathbb{E}\big\langle g(\theta^{(t)}),\theta^{(t)}-\theta^\star\big\rangle+\frac{\beta^2}{d}\mathbb{E}\big\|g(\theta^{(t)})\big\|^2\Big(1-\frac{1}{d}\Big)\\
&\leqslant\mathbb{E}\big\|\theta^{(t)}-\theta^\star\big\|^2\Big(1-\frac{2\beta\lambda L}{d(\lambda+L)}\Big)+\frac{\beta}{d}\Big(\frac{\beta}{d}-\frac{2}{\lambda+L}\Big)\mathbb{E}\big\|g(\theta^{(t)})\big\|^2+\frac{\beta^2}{d}\mathbb{E}\big\|g(\theta^{(t)})\big\|^2\Big(1-\frac{1}{d}\Big)\\
&=\mathbb{E}\big\|\theta^{(t)}-\theta^\star\big\|^2\Big(1-\frac{2\beta\lambda L}{d(\lambda+L)}\Big)+\frac{\beta}{d}\Big(\beta-\frac{2}{\lambda+L}\Big)\mathbb{E}\big\|g(\theta^{(t)})\big\|^2\\
&\leqslant\mathbb{E}\big\|\theta^{(t)}-\theta^\star\big\|^2\underbrace{\Big(1-\frac{2\beta\lambda L}{d(\lambda+L)}\Big)}_{=:\kappa^2}.
\end{aligned}$$

The first inequality is obtained by applying inequality (2.1.24) from [342] (see also [58] Lemma 3.11) and the second one is due to the choice of β. We can bound the second term as follows:

$$\big\|\widehat{g}_t-g_t\big\|_{L_2}^2=\mathbb{E}_{j_t}\big|\widehat{g}_t-g_t\big|^2=\frac{1}{d}\sum_{j=1}^d\big|\widehat{g}_j(\theta^{(t)})-g_j(\theta^{(t)})\big|^2\leqslant\frac{\epsilon_{Euc}^2}{d}.$$

Combining the latter with the former bound, we obtain the approximate contraction:

$$\big\|\theta^{(t+1)}-\theta^\star\big\|_{L_2}\leqslant\kappa\big\|\theta^{(t)}-\theta^\star\big\|_{L_2}+\frac{\beta\epsilon_{Euc}}{\sqrt{d}}.$$

By iterating this argument on T rounds we find that:

$$\big\|\theta^{(T)}-\theta^\star\big\|_{L_2}\leqslant\kappa^T\big\|\theta^{(0)}-\theta^\star\big\|_{L_2}+\frac{\beta\epsilon_{Euc}}{\sqrt{d}(1-\kappa)}.$$

Finally, the following inequality yields the result in the case of uniform sampling:

$$\frac{1}{1-\kappa} \leqslant \frac{1+\sqrt{1-\frac{2\beta\lambda L}{d(\lambda+L)}}}{\frac{2\beta\lambda L}{d(\lambda+L)}} \leqslant \frac{d(\lambda+L)}{\beta\lambda L}.$$

□

3.9.2 Proofs for Section 3.3

Proof of Lemma 3.1

Let $\theta \in \Theta$, using Assumption 3.1 we have:

$$|\ell(\theta^\top X, Y)| \leqslant C_{\ell,1} + C_{\ell,2}|\theta^\top X - Y|^q \leqslant C_{\ell,1} + 2^{q-1}C_{\ell,2}(|\theta^\top X|^q + |Y|^q).$$

Taking the expectation and using Assumption 3.2 shows that the risk $R(\theta)$ is well defined (recall that $q \leqslant 2$). Next, since $1 \leqslant q \leqslant 2$, simple algebra gives

$$\begin{aligned}
&\big|\ell'(\theta^\top X, Y)X_j\big|^{1+\alpha} \\
&\leqslant \big|\big(C'_{\ell,1} + C'_{\ell,2}|\theta^\top X - Y|^{q-1}\big)X^j\big|^{1+\alpha} \\
&\leqslant 2^\alpha\big(\big|C'_{\ell,1}X^j\big|^{1+\alpha} + (C'_{\ell,2}(|(\theta^\top X)^{q-1}X^j| + |Y^{q-1}X^j|))^{1+\alpha}\big) \\
&\leqslant 2^\alpha\Big(\big|C'_{\ell,1}X^j\big|^{1+\alpha} + \Big(C'_{\ell,2}\Big(\sum_{k=1}^d |\theta_k|^{q-1}|(X^k)^{q-1}X^j| + |Y^{q-1}X^j|\Big)\Big)^{1+\alpha}\Big) \\
&\leqslant 2^\alpha\Big(\big|C'_{\ell,1}X^j\big|^{1+\alpha} \\
&+ 2^\alpha(C'_{\ell,2})^{1+\alpha}\Big(d^\alpha\sum_{k=1}^d |\theta_k|^{(q-1)(1+\alpha)}|(X^k)^{q-1}X^j|^{1+\alpha} + |Y^{q-1}X^j|^{1+\alpha}\Big)\Big).
\end{aligned}$$

Given Assumption 3.2, it is straightforward that $\mathbb{E}|X^j\big|^{1+\alpha} < \infty$ and $\mathbb{E}|Y^{q-1}X^j|^{1+\alpha} < \infty$. Moreover, using a Hölder inequality with exponents $a = \frac{q(1+\alpha)}{(q-1)(1+\alpha)}$ and $b = q$ (the case $q = 1$ is trivial) we find:

$$\mathbb{E}\big|(X^k)^{q-1}X^j\big|^{1+\alpha} \leqslant \big(\mathbb{E}\big|X^k\big|^{q(1+\alpha)}\big)^{1/a}\big(\mathbb{E}\big|X^j\big|^{q(1+\alpha)}\big)^{1/b},$$

which is finite under Assumption 3.2. This concludes the proof of Lemma 3.1.

Proof of Lemma 3.2

This proof follows a standard argument from [284, 162] in which we use a Lemma from [59] in order to control the $(1+\alpha)$-moment of the block means instead of their variance. Indeed, we know from Lemma 3.1 that under Assumptions 3.1 and 3.2, the gradient coordinates have finite $(1+\alpha)$-moments, namely $\mathbb{E}[|\ell'(X^\top\theta, Y)X_j|^{1+\alpha}] < +\infty$ for any $j \in [\![d]\!]$. Recall that $(\widehat{g}_j^{(k)}(\theta))_{k\in[\![K]\!]}$ stands for the block-wise empirical mean given by Equation (3.3.4) and introduce the set of non-corrupted block indices given by $\mathcal{K} = \{k \in [\![K]\!] \ : \ B_k \cap \mathcal{O} = \emptyset\}$. We will initially assume that the number of outliers satisfies $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$ for some $0 < \varepsilon < 1$. Note that since samples are i.i.d in B_k for $k \in \mathcal{K}$, we have $\mathbb{E}\big[\widehat{g}_j^{(k)}(\theta)\big] = g_j(\theta)$. We use the following Lemma from [59].

Lemma 3.7 (Lemma 3 from [59]). *Let $Z, Z_1, \dots, Z_n$ be a i.i.d sequence with $m_\alpha = \mathbb{E}[|Z - \mathbb{E}Z|^{1+\alpha}] < +\infty$ for some $\alpha \in (0,1]$ and put $\bar{Z}_n = \frac{1}{n}\sum_{i\in[\![n]\!]} Z_i$. Then, we have*

$$\bar{Z}_n \leqslant \mathbb{E}Z + \Big(\frac{3m_\alpha}{\delta n^\alpha}\Big)^{1/(1+\alpha)}$$

for any $\delta \in (0,1)$, with a probability $1-\delta$.

Lemma 3.7 entails that

$$\big|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)\big| \leqslant \Big(\frac{3m_{j,\alpha}(\theta)}{\delta'(n/K)^\alpha}\Big)^{1/(1+\alpha)} =: \eta_{j,\alpha,\delta'}(\theta)$$

with probability larger than $1-2\delta'$, for each $k \in \mathcal{K}$, since we have n/K samples in block B_k. Now, recalling that $\widehat{g}_j(\theta)$ is the median (see (3.3.3)), we can upper bound its failure probability as follows:

$$\begin{aligned}
\mathbb{P}\Big[\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big| \geqslant \eta_{j,\alpha,\delta'}(\theta)\Big] &\\
&\leqslant \mathbb{P}\Big[\sum_{k\in[\![K]\!]} \mathbf{1}\Big\{\big|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)\big| \geqslant \eta_{j,\alpha,\delta'}(\theta)\Big\} > K/2\Big] \\
&\leqslant \mathbb{P}\Big[\sum_{k\in\mathcal{K}} \mathbf{1}\Big\{\big|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)\big| \geqslant \eta_{j,\alpha,\delta'}(\theta)\Big\} > K/2 - |\mathcal{O}|\Big],
\end{aligned}$$

since at most $|\mathcal{O}|$ blocks contain one outlier. Since the blocks B_k are disjoint and contain i.i.d samples for $k \in \mathcal{K}$, we know that

$$\sum_{k\in\mathcal{K}} \mathbf{1}\Big\{\big|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)\big| \geqslant \eta_{j,\alpha,\delta'}(\theta)\Big\}$$

follows a binomial distribution $\mathrm{Bin}(|\mathcal{K}|, p)$ with $p \leqslant 2\delta'$. Using the fact that $\mathrm{Bin}(|\mathcal{K}|, p)$ is stochastically dominated by $\mathrm{Bin}(|\mathcal{K}|, 2\delta')$ and that $\mathbb{E}[\mathrm{Bin}(|\mathcal{K}|, 2\delta')] = 2\delta'|\mathcal{K}|$, we obtain, if $S \sim \mathrm{Bin}(|\mathcal{K}|, 2\delta')$, that

$$\begin{aligned}
\mathbb{P}\Big[\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big| \geqslant \eta_{j,\alpha,\delta'}(\theta)\Big] &\leqslant \mathbb{P}\big[S > K/2 - |\mathcal{O}|\big] \\
&= \mathbb{P}\big[S - \mathbb{E}S > K/2 - |\mathcal{O}| - 2\delta'|\mathcal{K}|\big] \\
&\leqslant \mathbb{P}\big[S - \mathbb{E}S > K(\varepsilon - 4\delta')/2\big] \\
&\leqslant \exp\big(-K(\varepsilon - 4\delta')^2/2\big),
\end{aligned}$$

where we used the fact that $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$ and $|\mathcal{K}| \leqslant K$ for the second inequality and the Hoeffding inequality for the last. This concludes the proof of Lemma 3.2 for the choice $\varepsilon = 5/6$ and $\delta' = 1/8$.

Proof of Proposition 3.1

Step 1. First, we fix $\theta \in \Theta$ and try to bound $\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big|$ in terms of quantities only depending on $\widetilde{\theta}$ which is the closest point to θ in an ε-net. Recall that Δ is the diameter of the parameter set Θ and let $\varepsilon > 0$ be a positive number. There exists an ε-net covering Θ with cardinality no more than $(3\Delta/2\varepsilon)^d$ i.e. a set N_ε such that for all $\theta \in \Theta$ there exists $\widetilde{\theta} \in N_\varepsilon$ such that $\|\widetilde{\theta} - \theta\| \leqslant \varepsilon$. Consider a fixed $\theta \in \Theta$ and $j \in [\![d]\!]$, we wish to bound the quantity $\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big|$. Using the ε-net N_ε, there exists $\widetilde{\theta}$ such that $\|\widetilde{\theta} - \theta\| \leqslant \varepsilon$ which we can use as

follows:

$$\begin{aligned}\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)\big| &\leqslant \big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\widetilde{\theta})\big| + \big|g_j(\widetilde{\theta}) - g_j(\theta)\big| \\ &\leqslant \big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\widetilde{\theta})\big| + L_j\varepsilon, \end{aligned} \tag{3.9.8}$$

where we used the gradient's coordinate Lipschitz constant to bound the second term. We now focus on the second term. Introducing the notation $g_j^i(\theta) = \ell'(\theta^\top X_i, Y_i)X_i^j$, we have

$$g_j^i(\theta) = \ell'(\widetilde{\theta}^\top X_i, Y_i)X_i^j + \underbrace{(\ell'(\theta^\top X_i, Y_i) - \ell'(\widetilde{\theta}^\top X_i, Y_i))X_i^j}_{=:\Delta_i}.$$

Let $(B_k)_{k\in[\![K]\!]}$ be the blocks used to compute the `MOM` estimator and associated block means $\widehat{g}_j^{(k)}(\theta)$ and $\widehat{g}_j^{(k)}(\widetilde{\theta})$. Notice that the `MOM` estimator is *monotonous* non decreasing w.r.t. to each of the entries $g_j^i(\theta)$ when the others are fixed. Without loss of generality, assume that $\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\widetilde{\theta}) \geqslant 0$ then we have:

$$\big|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\widetilde{\theta})\big| \leqslant \big|\breve{g}_j^{\texttt{MOM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big|, \tag{3.9.9}$$

where $\breve{g}_j^{\texttt{MOM}}(\widetilde{\theta})$ is the `MOM` estimator obtained using the entries $\ell'\big(\widetilde{\theta}^\top X_i, Y_i\big)X_i^j + \varepsilon\gamma\|X_i\|^2 = g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2$ instead of $g_j^i(\theta)$. Note that $\breve{g}_j^{\texttt{MOM}}(\widetilde{\theta})$ no longer depends on θ except through the fact that $\widetilde{\theta}$ is chosen in N_ε so that $\big\|\widetilde{\theta} - \theta\big\| \leqslant \varepsilon$. Indeed, using the Lipschitz smoothness of the loss function and a Cauchy-Schwarz inequality we find that:

$$|\Delta_i| \leqslant \gamma\|\theta - \widetilde{\theta}\| \cdot \|X_i\| \cdot |X_i^j| \leqslant \varepsilon\gamma\|X_i\|^2.$$

Step 2. We now use the concentration property of `MOM` to bound the quantity which is in terms of $\widetilde{\theta}$. The samples $(g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)_{i\in[\![n]\!]}$ are independent and distributed according to the random variable $\ell'(\widetilde{\theta}^\top X, Y)X^j + \varepsilon\gamma\|X\|^2$. Denote $\overline{L} = \gamma\mathbb{E}\|X\|^2$ and for $k \in [\![K]\!]$ let $\widehat{g}_j^{(k)}(\widetilde{\theta}) = \frac{K}{n}\sum_{i\in B_k} g_j^i(\widetilde{\theta})$ and $\widehat{L}^{(k)} = \frac{K}{n}\sum_{i\in B_k}\gamma\|X_i\|^2$. We use Lemma 3.7 for each of these pairs of means to obtain that with probability at least $1 - \delta'/2$:

$$\big|\widehat{g}_j^{(k)}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| \leqslant \Big(\frac{6m_{j,\alpha}(\widetilde{\theta})}{\delta'(n/K)^\alpha}\Big)^{1/(1+\alpha)} =: \eta_{j,\alpha,\delta'/2}(\widetilde{\theta}),$$

and with probability at least $1 - \delta'/2$

$$\big|\widehat{L}^{(k)} - \overline{L}\big| \leqslant \Big(\frac{6m_{L,\alpha}}{\delta'(n/K)^\alpha}\Big)^{1/(1+\alpha)} =: \eta_{L,\alpha,\delta'/2},$$

where $m_{L,\alpha} = \mathbb{E}|\gamma\|X\|^2 - \overline{L}|^{1+\alpha}$. Hence for all $k \in [\![K]\!]$

$$\begin{aligned}&\mathbb{P}\big(\big|\widehat{g}_j^{(k)}(\widetilde{\theta}) + \varepsilon\widehat{L}^{(k)} - g_j(\widetilde{\theta})\big| > \eta_{j,\alpha,\delta'/2}(\widetilde{\theta}) + \varepsilon(\overline{L} + \eta_{L,\alpha,\delta'/2})\big) \\ &\quad\leqslant \mathbb{P}\big(\big|\widehat{g}_j^{(k)}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| > \eta_{j,\alpha,\delta'/2}(\widetilde{\theta})\big) + \mathbb{P}\big(\big|\widehat{L}^{(k)} - \overline{L}\big| > \eta_{L,\alpha,\delta'/2}\big) \\ &\quad\leqslant \delta'/2 + \delta'/2 = \delta'.\end{aligned}$$

Now defining the Bernoulli variables

$$U_k := \mathbf{1}\Big\{\big|\widehat{g}_j^{(k)}(\widetilde{\theta}) + \delta\widehat{L}^{(k)} - g_j(\widetilde{\theta})\big| > \eta_{j,\alpha,\delta'/2}(\widetilde{\theta}) + \varepsilon\big(\overline{L} + \eta_{L,\alpha,\delta'/2}\big)\Big\},$$

we have just seen they have success probability $\leqslant \delta'$, moreover

$$\mathbb{P}\Big[|\widetilde{g}_j^{\texttt{MOM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})| \geqslant \eta_{j,\alpha,\delta'/2}(\widetilde{\theta}) + \varepsilon(\overline{L} + \eta_{L,\alpha,\delta'/2})\Big] \leqslant \mathbb{P}\Big[\sum_{k\in[\![K]\!]} U_k > K/2\Big]$$
$$\leqslant \mathbb{P}\Big[\sum_{k\in\mathcal{K}} U_k > K/2 - |\mathcal{O}|\Big],$$

since at most $|\mathcal{O}|$ blocks contain one outlier. Since the blocks B_k are disjoint and contain i.i.d samples for $k \in \mathcal{K}$, we know that $\sum_{k\in\mathcal{K}} U_k$ follows a binomial distribution $\mathrm{Bin}(|\mathcal{K}|, p)$ with $p \leqslant \delta'$. Using the fact that $\mathrm{Bin}(|\mathcal{K}|, p)$ is stochastically dominated by $\mathrm{Bin}(|\mathcal{K}|, \delta')$ and that $\mathbb{E}[\mathrm{Bin}(|\mathcal{K}|, \delta')] = \delta'|\mathcal{K}|$, we obtain, if $S \sim \mathrm{Bin}(|\mathcal{K}|, \delta')$, that

$$\begin{aligned}\mathbb{P}\Big[|\widetilde{g}_j^{\texttt{MOM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})| \geqslant \eta_{j,\alpha,\delta'/2}(\widetilde{\theta}) + \varepsilon\big(\overline{L} + \eta_{L,\alpha,\delta'/2}\big)\Big] &\leqslant \mathbb{P}\big[S > K/2 - |\mathcal{O}|\big] \\ &= \mathbb{P}\big[S - \mathbb{E}S > K/2 - |\mathcal{O}| - \delta'|\mathcal{K}|\big] \\ &\leqslant \mathbb{P}\big[S - \mathbb{E}S > K(\varepsilon' - 2\delta')/2\big] \\ &\leqslant \exp\big(-K(\varepsilon' - 2\delta')^2/2\big),\end{aligned}$$

where we used the condition $|\mathcal{O}| \leqslant (1-\varepsilon')K/2$ and $|\mathcal{K}| \leqslant K$ for the second inequality and the Hoeffding inequality for the last. To conclude, we choose $\varepsilon' = 5/6$ and $\delta' = 1/4$ and combine (3.9.8), (3.9.9) and the last inequality in which we take $K = \lceil 18\log(1/\delta)\rceil$ and use a union bound argument to obtain that with probability at least $1-\delta$ for all $j \in [\![d]\!]$

$$|\widetilde{g}_j^{\texttt{MOM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})| \leqslant \big((24m_{j,\alpha}(\widetilde{\theta}))^{1/(1+\alpha)} + \varepsilon(24m_{L,\alpha})^{1/(1+\alpha)}\big)\Big(\frac{18\log(d/\delta)}{n}\Big)^{\alpha/(1+\alpha)} + \varepsilon\overline{L}. \quad (3.9.10)$$

Step 3. We use the ε-net to obtain a uniform bound. For $\theta \in \Theta$ denote $\widetilde{\theta}(\theta) \in N_\varepsilon$ the closest point in N_ε satisfying in particular $\|\widetilde{\theta}(\theta) - \theta\| \leqslant \varepsilon$, we write, following previous arguments

$$\begin{aligned}\sup_{\theta\in\Theta} |\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)| &\leqslant \sup_{\theta\in\Theta} |\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\widetilde{\theta}(\theta))| + |g_j(\widetilde{\theta}(\theta)) - g_j(\theta)| \\ &\leqslant \sup_{\theta\in\Theta} |\widetilde{g}_j^{\texttt{MOM}}(\widetilde{\theta}(\theta)) - g_j(\widetilde{\theta}(\theta))| + \varepsilon L_j \\ &= \max_{\widetilde{\theta}\in N_\varepsilon} |\widetilde{g}_j^{\texttt{MOM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})| + \varepsilon L_j.\end{aligned}$$

Here, we make a union bound argument over $\widetilde{\theta} \in N_\varepsilon$ for the inequality (3.9.10) and choose $\varepsilon = n^{-\alpha/(1+\alpha)}$ to obtain the final result concluding the proof of Proposition 3.1.

Proof of Proposition 3.2

This proof reuses arguments from the proof of Theorem 2 in [262]. We wish to bound $|\widehat{g}_j^{\texttt{MOM}}(\theta) - g_j(\theta)|$ with high probability and uniformly on $\theta \in \Theta$. Fix $\theta \in \Theta$ and $j \in [\![d]\!]$, we have $\widehat{g}_j^{\texttt{MOM}}(\theta) = \mathrm{median}\big(\widehat{g}_j^{(1)}(\theta), \dots, \widehat{g}_j^{(K)}(\theta)\big)$ with $\widehat{g}_j^{(k)}(\theta) = \frac{K}{n}\sum_{i\in B_k} g_j^i(\theta)$ where the blocks $B_1, \dots, B_K$ constitute a partition of $[\![n]\!]$.

Define the function $\phi(t) = (t-1)\mathbf{1}_{1\leqslant t\leqslant 2} + \mathbf{1}_{t>2}$, let $\mathcal{K} = \{k \in [\![K]\!],\ B_k \cap \mathcal{O} = \emptyset\}$ and $\mathcal{J} = \bigcup_{k\in\mathcal{K}} B_k$. Thanks to the inequality $\phi(t) \geqslant \mathbf{1}_{t\geqslant 2}$, we have:

$$\sup_{\theta\in\Theta} \sum_{k=1}^{K} \mathbf{1}\Big\{|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)| > x\Big\} \leqslant \sup_{\theta\in\Theta} \sum_{k\in\mathcal{K}} \mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta) - g_j(\theta)|/x\big)\big] + |\mathcal{O}|$$

$$+\sup_{\theta\in\Theta}\sum_{k\in\mathcal{K}}\Big(\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)-\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\Big).$$

Besides, the inequality $\phi(t)\leqslant \mathbf{1}_{t\geqslant 1}$, an application of Markov's inequality and Lemma 3.7 yield:

$$\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\leqslant \mathbb{P}\big(|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|\geqslant x/2\big)\leqslant \frac{3m_{\alpha,j}(\theta)}{(x/2)^{1+\alpha}(n/K)^{\alpha}}.$$

Therefore, recalling that we defined $M_{\alpha,j}:=\sup_{\theta\in\Theta} m_{\alpha,j}(\theta)$ we have

$$\sup_{\theta\in\Theta}\sum_{k=1}^{K}\mathbf{1}\Big\{|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|>x\Big\}\leqslant K\Big(\frac{3M_{\alpha,j}}{(x/2)^{1+\alpha}(n/K)^{\alpha}}+\frac{|\mathcal{O}|}{K}$$
$$+\sup_{\theta\in\Theta}\frac{1}{K}\Big(\sum_{k\in\mathcal{K}}\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)-\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\Big)\Big).$$

Now since for all t we have $0\leqslant\phi(t)\leqslant 1$, McDiarmid's inequality says with probability $\geqslant 1-\exp(-2y^2K)$ that:

$$\sup_{\theta\in\Theta}\frac{1}{K}\Big(\sum_{k\in\mathcal{K}}\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)-\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\Big)\leqslant$$
$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\frac{1}{K}\Big(\sum_{k\in\mathcal{K}}\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)-\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\Big)\Big]+y.$$

Using a simple symmetrization argument (see for instance Lemma 11.4 in [50]) we find:

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\frac{1}{K}\Big(\sum_{k\in\mathcal{K}}\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)-\mathbb{E}\big[\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\big]\Big)\Big]\leqslant$$
$$2\mathbb{E}\Big[\sup_{\theta\in\Theta}\frac{1}{K}\sum_{k\in\mathcal{K}}\varepsilon_k\phi\big(2|\widehat{g}^{(k)}(\theta)-g(\theta)|/x\big)\Big],$$

where the ε_ks are independent Rademacher variables. Since ϕ is 1-Lipschitz and satisfies $\phi(0)=0$ we can use the contraction principle (see Theorem 11.6 in [50]) followed by another symmetrization step to find

$$2\mathbb{E}\Big[\sup_{\theta\in\Theta}\frac{1}{K}\sum_{k\in\mathcal{K}}\varepsilon_k\phi\big(2|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\big)\Big]\leqslant 4\mathbb{E}\Big[\sup_{\theta\in\Theta}\frac{1}{K}\sum_{k\in\mathcal{K}}\varepsilon_k|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|/x\Big]$$
$$\leqslant\frac{8}{xn}\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i\in\mathcal{J}}\varepsilon_i g_j^i(\theta)\Big]\leqslant\frac{8\mathcal{R}_j(\Theta)}{xn}.$$

Taking $|\mathcal{O}|\leqslant(1-\varepsilon)K/2$, we found that with probability $\geqslant 1-\exp(-2y^2K)$

$$\sup_{\theta\in\Theta}\sum_{k=1}^{K}\mathbf{1}\Big\{|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)|>x\Big\}\leqslant K\Big(\frac{3M_{\alpha,j}}{(x/2)^{1+\alpha}(n/K)^{\alpha}}+\frac{|\mathcal{O}|}{K}+\frac{8\mathcal{R}_j(\Theta)}{xn}\Big).$$

Now by choosing $y=1/4-|\mathcal{O}|/K$ and $x=\max\Big(\Big(\frac{36M_{\alpha,j}}{(n/K)^{\alpha}}\Big)^{1/(1+\alpha)},\frac{64\mathcal{R}_j(\Theta)}{n}\Big)$, we obtain the

deviation bound:

$$\begin{aligned}\mathbb{P}\Big(\sup_{\theta\in\Theta}\big|\widehat{g}_j^{\texttt{MOM}}(\theta)-g_j(\theta)\big|\geqslant\max\Big(\Big(\frac{36M_{\alpha,j}}{(n/K)^\alpha}\Big)^{1/(1+\alpha)},\frac{64\mathcal{R}_j(\Theta)}{n}\Big)\Big)&\\ \leqslant\mathbb{P}\Big(\sup_{\theta\in\Theta}\sum_{k=1}^{K}\mathbf{1}\Big\{\big|\widehat{g}_j^{(k)}(\theta)-g_j(\theta)\big|>x\Big\}>K/2\Big)&\\ \leqslant\exp(-2(\varepsilon-1/2)^2K/4)&\\ \leqslant\exp(-K/18),&\end{aligned}$$

where the last inequality comes from the choice $\varepsilon=5/6$. A simple union bound argument lets the previous inequality hold for all $j\in[\![d]\!]$ with high probability.

Finally, assuming that X^j has finite fourth moment for all $j\in[\![d]\!]$, we can control the Rademacher complexity. In this part, we assume without loss of generality that $\mathcal{I}=[\![n]\!]$, we first write

$$\begin{aligned}\mathcal{R}_j(\Theta)&=\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n}\varepsilon_i\ell'(\theta^\top X_i,Y_i)X_i^j\Big]\\&=\mathbb{E}\Big[\sum_{i=1}^{n}\varepsilon_i\ell'(0,Y_i)X_i^j+\sup_{\theta\in\Theta}\sum_{i=1}^{n}\varepsilon_i(\ell'(\theta^\top X_i,Y_i)-\ell'(0,Y_i))X_i^j\Big].\end{aligned}$$

Denote $\phi_i(t)=(\ell'(t,Y_i)-\ell'(0,Y_i))X_i^j$ and notice that $\mathbb{E}\big[\sum_{i=1}^n\varepsilon_i\ell'(0,Y_i)X_i^j\big]=0$. Notice also that $\phi_i(0)=0$ and ϕ_i is $\gamma|X_i^j|$-Lipschitz for all i. We use a variant of the contraction principle adapted to our case in which functions with different Lipschitz constants appear. We use Lemma 11.7 from [50] and adapt the proof of their Theorem 11.6 to make the following estimations:

$$\begin{aligned}&\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n}\varepsilon_i\phi_i(\theta^\top X_i)\Big]\\&=\mathbb{E}\Big[\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n-1}\varepsilon_i\phi_i(\theta^\top X_i)+\varepsilon_n\phi_n(\theta^\top X_n)\Big|(\varepsilon_i)_{i=1}^{n-1},(X_i,Y_i)_{i\in[\![n]\!]}\Big]\Big]\\&\leqslant\mathbb{E}\Big[\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n-1}\varepsilon_i\phi_i(\theta^\top X_i)+\varepsilon_n\gamma|X_n^j|\theta^\top X_n\Big|(\varepsilon_i)_{i=1}^{n-1},(X_i,Y_i)_{i\in[\![n]\!]}\Big]\Big]\\&=\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n-1}\varepsilon_i\phi_i(\theta^\top X_i)+\varepsilon_n\gamma|X_n^j|\theta^\top X_n\Big].\end{aligned}$$

By iterating the previous argument n times we find:

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n}\varepsilon_i\phi_i(\theta^\top X_i)\Big]\leqslant\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n-1}\varepsilon_i\gamma|X_i^j|\theta^\top X_i\Big].$$

Now recalling that the diameter of Θ is Δ, we use Lemma 3.8 below with $p=1$ to bound the previous quantity as:

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\sum_{i=1}^{n}\varepsilon_i\gamma|X_i^j|\theta^\top X_i\Big]=\gamma\mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\langle\theta,\sum_{i=1}^{n}\varepsilon_iX_i|X_i^j|\Big\rangle\Big]$$

$$
\begin{aligned}
&\leqslant \gamma\Delta\mathbb{E}\Big[\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\varepsilon_i X_i|X_i^j|\Big\|_1\Big|(X_i)_{i\in[\![n]\!]}\Big]\Big]\\
&\leqslant \gamma\Delta C_\alpha\mathbb{E}\Big[\sum_{i=1}^{n}\|X_i\|^{1+\alpha}|X_i^j|^{1+\alpha}\Big]^{1/(1+\alpha)}\\
&\leqslant \gamma\Delta C_\alpha\Big(n\mathbb{E}\big[(X^j)^{2(1+\alpha)}\big]^{1/2}\sum_{k\in[\![d]\!]}\mathbb{E}\big[(X^k)^{2(1+\alpha)}\big]^{1/2}\Big)^{1/(1+\alpha)},
\end{aligned}
$$

where we used a Cauchy-Schwarz inequality in the last step, which concludes the proof of Proposition 3.2. □

Lemma 3.8 (Khintchine inequality variant)**.** *Let $\alpha\in(0,1]$ and $(x_i)_{i\in[\![n]\!]}$ be real numbers with $n\in\mathbb{N}$ and $p>0$ and $(\varepsilon_i)_{i\in[\![n]\!]}$ be i.i.d Rademacher random variables then we have the inequality:*

$$
\mathbb{E}\Big[\Big|\sum_{i=1}^{n}\varepsilon_i x_i\Big|^p\Big]^{1/p}\leqslant B_{p,\alpha}\Big(\sum_{i=1}^{n}|x_i|^{1+\alpha}\Big)^{1/(1+\alpha)}
$$

with the constant $B_{p,\alpha}:=2p\Big(\frac{1+\alpha}{\alpha}\Big)^{\alpha p/(1+\alpha)-1}\Gamma\Big(\frac{\alpha p}{1+\alpha}\Big)$. Moreover, for $p=1$ the constant $B_{1,\alpha}$ is bounded for any $\alpha\geqslant 0$.

Proof. This proof is a generalization of Lemma 4.1 from [267] and uses similar methods. For all $\lambda>0$ we have:

$$
\begin{aligned}
\mathbb{E}\exp\Big(\lambda\sum_i\varepsilon_i x_i\Big)&=\prod_i\mathbb{E}\exp(\lambda\varepsilon_i x_i)=\prod_i\cosh(\lambda x_i)\\
&\leqslant\prod_i\exp\Big(\frac{|\lambda x_i|^{1+\alpha}}{1+\alpha}\Big)=\exp\Big(\sum_i\frac{|\lambda x_i|^{1+\alpha}}{1+\alpha}\Big),
\end{aligned}
$$

where we used the inequality $\cosh(u)\leqslant\exp\Big(\frac{|u|^{1+\alpha}}{1+\alpha}\Big)$ valid for all $u\in\mathbb{R}$ which can be quickly proven. Since both functions are even, fix $u>0$ and define $f_u(\alpha)=\exp\Big(\frac{|u|^{1+\alpha}}{1+\alpha}\Big)-\cosh(u)$, we can show that f_u is monotonous on $[0,1]$ separately for $u\in(0,\sqrt{e})$ and $(e,+\infty)$ and notice that $f_u(0)$ and $f_u(1)$ are both non-negative for all $u>0$ thanks to the famous inequality $\cosh(u)\leqslant e^{u^2/2}$. Therefore, the inequality holds for $u\in(0,\sqrt{e})$ and $(e,+\infty)$. Finally, for $u\in(\sqrt{e},e)$, the function $f_u(\alpha)$ reaches a minimum at $f_u(1/\log(u)-1)=u^e-\cosh(u)$ and by taking logarithms we have $u^e\geqslant\cosh(u)\iff\log(1+e^{2u})\leqslant u+\log(2)+e\log(u)$ but since the derivatives verify $\frac{2}{1+e^{-2u}}\leqslant 2\leqslant 1+e/u$ for $u\in(\sqrt{e},e)$ and $e^{e/2}\geqslant\cosh(\sqrt{e})$ the desired inequality follows by integration.

By homogeneity, we can focus on the case $\big(\sum_{i=1}^n|x_i|^{1+\alpha}\big)^{1/(1+\alpha)}=1$, we compute:

$$
\begin{aligned}
\mathbb{E}\Big|\sum_i\varepsilon_i x_i\Big|^p&=\int_0^{+\infty}\mathbb{P}\Big(\Big|\sum_i\varepsilon_i x_i\Big|^p>t\Big)dt\\
&\leqslant 2\int_0^{+\infty}\exp\Big(\frac{\lambda^{1+\alpha}}{1+\alpha}-\lambda t^{1/p}\Big)dt\\
&=2\int_0^{+\infty}\exp\Big(-\frac{\alpha}{1+\alpha}u^{(1+\alpha)/\alpha}\Big)du^p\\
&=2p\Big(\frac{1+\alpha}{\alpha}\Big)^{\alpha p/(1+\alpha)-1}\Gamma\Big(\frac{\alpha p}{1+\alpha}\Big)=B_{p,\alpha}^p,
\end{aligned}
$$

where we used the previous inequality and chose $\lambda = (t^{1/p})^{1/\alpha}$ in the last step. This proves the main inequality. Finally, it is easy to see that $B_{1,\alpha}$ is bounded for high values of α while for $\alpha \sim 0$ it is consequence of the fact that $\Gamma(x) \sim 1/x$ near 0 and the limit $x^x \to 0$ when $x \to 0^+$. □

Proof of Lemma 3.3

As previously, Lemma 3.1 along with Assumptions 3.1 and 3.2 guarantee that the gradient coordinates have finite $(1+\alpha)$-moments. From here, Lemma 3.3 is a direct application of Lemma 3.9 stated and proved below. In the following lemma, for any sequence $(z_i)_{i=1}^N$ of real numbers, $(z_i^*)_{i=1}^N$ denotes a non-decreasing reordering of it.

Lemma 3.9. *Let $\widetilde{X}_1, \dots, \widetilde{X}_N, \widetilde{Y}_1, \dots, \widetilde{Y}_N$ denote an η-corrupted i.i.d sample with rate η from a random variable X with expectation $\mu = \mathbb{E}X$ and with finite $1+\gamma$ centered moment $\mathbb{E}|X-\mu|^{1+\gamma} = M < \infty$ for some $0 < \gamma \leqslant 1$. Denote $\widehat{\mu}$ the ϵ-trimmed mean estimator computed as $\widehat{\mu} = \frac{1}{N}\sum_{i=1}^N \phi_{\alpha,\beta}(\widetilde{X}_i)$ with $\phi_{\alpha,\beta}(x) = \max(\alpha, \min(x,\beta))$ and the thresholds $\alpha = \widetilde{Y}^*_{\epsilon N}$ and $\beta = \widetilde{Y}^*_{(1-\epsilon)N}$. Let $1 > \delta \geqslant e^{-N}/4$, taking $\epsilon = 8\eta + 12\frac{\log(4/\delta)}{n}$, we have*

$$|\widehat{\mu} - \mu| \leqslant 7M^{\frac{1}{1+\gamma}}(\epsilon/2)^{\frac{\gamma}{1+\gamma}} \tag{3.9.11}$$

with probability at least $1-\delta$.

Proof. This proof goes along the lines of the proof of Theorem 1 from [288] with the main difference that only the $(1+\gamma)$-moment is used instead of the variance. Denote X the random variable whose expectation $\mu = \mathbb{E}X$ is to be estimated and $\overline{X} = X - \mu$. Let $X_1, \dots, X_N, Y_1, \dots, Y_N$ the original uncorrupted i.i.d. sample from X and let $\widetilde{X}_1, \dots, \widetilde{X}_N, \widetilde{Y}_1, \dots, \widetilde{Y}_N$ denote the corrupted sample with rate η. We define the following quantity which will intervene in the proof:

$$\overline{\mathcal{E}}(\epsilon, X) := \max\Big\{\mathbb{E}\big[|\overline{X} - Q_{\epsilon/2}(\overline{X})|\mathbf{1}_{\overline{X} \leqslant Q_{\epsilon/2}(\overline{X})}\big], \mathbb{E}\big[|\overline{X} - Q_{1-\epsilon/2}(\overline{X})|\mathbf{1}_{\overline{X} \geqslant Q_{1-\epsilon/2}(\overline{X})}\big]\Big\}. \tag{3.9.12}$$

Step 1. We first derive confidence bounds on the truncation thresholds. Define the random variable $U = \mathbf{1}_{\overline{X} \geqslant Q_{1-2\epsilon}(\overline{X})}$. Its standard deviation satisfies $\sigma_U \leqslant \mathbb{P}^{1/2}(\overline{X} \geqslant Q_{1-2\epsilon}(\overline{X})) = \sqrt{2\epsilon}$. By applying Bernstein's inequality we find with probability $\geqslant 1 - \exp(-\epsilon N/12)$ that:

$$\big|\big\{i \,:\, Y_i \geqslant \mu + Q_{1-2\epsilon}(\overline{X})\big\}\big| \geqslant 3\epsilon N/2,$$

a similar argument with $U = \mathbf{1}_{\overline{X} > Q_{1-\epsilon/2}(\overline{X})}$ yields with probability $\geqslant 1 - \exp(-\epsilon N/12)$ that:

$$\big|\big\{i \,:\, Y_i \leqslant \mu + Q_{1-\epsilon/2}(\overline{X})\big\}\big| \geqslant (1 - (3/4)\epsilon)N,$$

and similarly with probability $\geqslant 1 - \exp(-\epsilon N/12)$ we have:

$$\big|\big\{i \,:\, Y_i \leqslant \mu + Q_{2\epsilon}(\overline{X})\big|\big\}\big| \geqslant 3\epsilon N/2,$$

and with probability $\geqslant 1 - \exp(-\epsilon N/12)$:

$$\big|\big\{i \,:\, Y_i \geqslant \mu + Q_{\epsilon/2}(\overline{X})\big\}\big| \geqslant (1 - (3/4)\epsilon)N,$$

so that with probability $\geqslant 1 - 4\exp(-\epsilon N/12) \geqslant 1 - \delta/2$ the four previous inequalities hold simultaneously. We call this event E which only depends on the variables $Y_1, \dots, Y_N$. Since $\eta \leqslant \epsilon/8$, if $2\eta N$ samples are corrupted we still have:

$$\big|\big\{i \,:\, \widetilde{Y}_i \geqslant \mu + Q_{1-2\epsilon}(\overline{X})\big|\big\}\big| \geqslant ((3/2)\epsilon - 2\eta)N \geqslant \epsilon N$$

and

$$\left|\left\{i \,:\, \widetilde{Y}_i \leqslant \mu + Q_{1-\epsilon/2}(\overline{X})\right\}\right| \geqslant (1-(3/4)\epsilon - 2\eta)N \geqslant (1-\epsilon)N$$

consequently, the two following bounds hold

$$Q_{1-2\epsilon}(\overline{X}) \leqslant \widetilde{Y}^*_{(1-\epsilon)N} - \mu \leqslant Q_{1-\epsilon/2}(\overline{X})$$

and similarly

$$Q_{\epsilon/2}(\overline{X}) \leqslant \widetilde{Y}^*_{\epsilon N} - \mu \leqslant Q_{2\epsilon}(\overline{X}).$$

This provides guarantees on the truncation levels used which are $\alpha = \widetilde{Y}^*_{\epsilon N}$ and $\beta = \widetilde{Y}^*_{(1-\epsilon)N}$.

Step 2. We first bound the deviation $\left|\frac{1}{N}\sum_{i=1}^N \phi_{\alpha,\beta}(X_i) - \mu\right|$ in the absence of corruption. W e write:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^N \phi_{\alpha,\beta}(X_i) &\leqslant \frac{1}{N}\sum_{i=1}^N \phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X_i) = \mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)\big] \\ &\quad + \frac{1}{N}\sum_{i=1}^N \Big(\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X_i) - \mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)\big]\Big). \qquad (3.9.13)\end{aligned}$$

The first term is dominated by:

$$\begin{aligned}&\mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)\big] = \mathbb{E}\big[\phi_{Q_{2\epsilon}(X),Q_{1-\epsilon/2}(X)}(X)\big] \\ &\quad = \mathbb{E}\big[Q_{2\epsilon}(X)\mathbf{1}_{X\leqslant Q_{2\epsilon}(X)} + X\mathbf{1}_{Q_{2\epsilon}(X)<X<Q_{1-\epsilon/2}(X)} + Q_{1-\epsilon/2}(X)\mathbf{1}_{X\geqslant Q_{1-\epsilon/2}(X)}\big] \\ &\quad = \mu + \mathbb{E}\big[(Q_{2\epsilon}(X)-X)\mathbf{1}_{X\leqslant Q_{2\epsilon}(X)} + (Q_{1-\epsilon/2}(X)-X)\mathbf{1}_{X\geqslant Q_{1-\epsilon/2}(X)}\big] \\ &\quad \leqslant \mu + \mathbb{E}\big[(Q_{2\epsilon}(X)-X)\mathbf{1}_{X\leqslant Q_{2\epsilon}(X)}\big] = \mu + \mathbb{E}\big[(Q_{2\epsilon}(\overline{X})-\overline{X})\mathbf{1}_{\overline{X}\leqslant Q_{2\epsilon}(\overline{X})}\big] \\ &\quad \leqslant \mu + \overline{\mathcal{E}}(4\epsilon, X),\end{aligned}$$

and lower bounded by:

$$\begin{aligned}&\mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)\big] \\ &\quad = \mu + \mathbb{E}\big[(Q_{2\epsilon}(X)-X)\mathbf{1}_{X\leqslant Q_{2\epsilon}(X)} + (Q_{1-\epsilon/2}(X)-X)\mathbf{1}_{X\geqslant Q_{1-\epsilon/2}(X)}\big] \\ &\quad \geqslant \mu + \mathbb{E}\big[(Q_{1-\epsilon/2}(X)-X)\mathbf{1}_{X\geqslant Q_{1-\epsilon/2}(X)}\big] = \mu + \mathbb{E}\big[(Q_{1-\epsilon/2}(\overline{X})-\overline{X})\mathbf{1}_{\overline{X}\geqslant Q_{1-\epsilon/2}(\overline{X})}\big] \\ &\quad \geqslant \mu - \overline{\mathcal{E}}(\epsilon, X).\end{aligned}$$

The sum in (3.9.13) above has terms upper bounded by $Q_{1-\epsilon/2}(\overline{X}) + \overline{\mathcal{E}}(\epsilon, X)$. We need to work with the knowledge that $\mathbb{E}|\overline{X}|^{1+\gamma} = M < \infty$ in order to bound their variance:

$$\begin{aligned}&\mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X) - \mathbb{E}[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)]\big]^2 \\ &\quad \leqslant \mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X) - \mu\big]^2 = \mathbb{E}\big[\phi_{Q_{2\epsilon}(\overline{X}),Q_{1-\epsilon/2}(\overline{X})}(\overline{X})^2\big] \\ &\quad = \mathbb{E}\big[Q_{2\epsilon}(\overline{X})\mathbf{1}_{\overline{X}\leqslant Q_{2\epsilon}(\overline{X})} + \overline{X}\mathbf{1}_{Q_{2\epsilon}(\overline{X})<\overline{X}<Q_{1-\epsilon/2}(\overline{X})} + Q_{1-\epsilon/2}(\overline{X})\mathbf{1}_{\overline{X}\geqslant Q_{1-\epsilon/2}(\overline{X})}\big]^2 \\ &\quad = \mathbb{E}\big[Q_{2\epsilon}(\overline{X})^2\mathbf{1}_{\overline{X}\leqslant Q_{2\epsilon}(\overline{X})} + \overline{X}^2\mathbf{1}_{Q_{2\epsilon}(\overline{X})<\overline{X}<Q_{1-\epsilon/2}(\overline{X})} + Q_{1-\epsilon/2}(\overline{X})^2\mathbf{1}_{\overline{X}\geqslant Q_{1-\epsilon/2}(\overline{X})}\big].\end{aligned}$$

To control the three terms in the previous expression we mimic the proof of Chebyshev's inequality to obtain that, when $Q_{2\epsilon}(\overline{X}) < 0$:

$$2\epsilon = \mathbb{P}\big(\overline{X} \leqslant Q_{2\epsilon}(\overline{X})\big) \leqslant \mathbb{P}\big(|\overline{X}|^{1+\gamma} \geqslant |Q_{2\epsilon}(\overline{X})|^{1+\gamma}\big) \leqslant \frac{M}{|Q_{2\epsilon}(\overline{X})|^{1+\gamma}}, \tag{3.9.14}$$

analogously, when $Q_{1-\epsilon/2}(\overline{X}) > 0$ we have:

$$\epsilon/2 = \mathbb{P}\big(\overline{X} \geqslant Q_{1-\epsilon/2}(\overline{X})\big) \leqslant \mathbb{P}\big(|\overline{X}|^{1+\gamma} \geqslant |Q_{1-\epsilon/2}(\overline{X})|^{1+\gamma}\big) \leqslant \frac{M}{|Q_{1-\epsilon/2}(\overline{X})|^{1+\gamma}}, \tag{3.9.15}$$

from (3.9.14), we deduce that

$$\mathbb{E}\big[Q_{2\epsilon}(\overline{X})^2 \mathbf{1}_{\overline{X} \leqslant Q_{2\epsilon}(\overline{X})}\big] = 2\epsilon Q_{2\epsilon}(\overline{X})^2 \leqslant 2\epsilon\Big(\frac{M}{2\epsilon}\Big)^{\frac{2}{1+\gamma}} \leqslant 2\epsilon\Big(\frac{2M}{\epsilon}\Big)^{2/(1+\gamma)},$$

and from (3.9.15) we find

$$\mathbb{E}\big[Q_{1-\epsilon/2}(\overline{X})^2 \mathbf{1}_{\overline{X} \geqslant Q_{1-\epsilon/2}(\overline{X})}\big] = Q_{1-\epsilon/2}(\overline{X})^2 \epsilon/2 \leqslant 2\epsilon\Big(\frac{2M}{\epsilon}\Big)^{2/(1+\gamma)}.$$

In the pathological case where we have $Q_{2\epsilon}(\overline{X}) \geqslant 0$ we use that $Q_{2\epsilon}(\overline{X}) \leqslant Q_{1-\epsilon/2}(\overline{X})$ (for $\epsilon \leqslant 2/5$) we deduce $|Q_{2\epsilon}(\overline{X})| \leqslant |Q_{1-\epsilon/2}(\overline{X})|$ and hence we still have

$$\mathbb{E}\big[Q_{2\epsilon}(\overline{X})^2 \mathbf{1}_{\overline{X} \leqslant Q_{2\epsilon}(\overline{X})}\big] \leqslant 2\epsilon Q_{1-\epsilon/2}(\overline{X})^2 \leqslant 2\epsilon\Big(\frac{2M}{\epsilon}\Big)^{2/(1+\gamma)}.$$

The case $Q_{1-\epsilon/2}(\overline{X}) \leqslant 0$ is similarly handled. Moreover, a simple calculation yields

$$\mathbb{E}\big[\overline{X}^2 \mathbf{1}_{Q_{2\epsilon}(\overline{X}) \leqslant \overline{X} \leqslant Q_{1-\epsilon/2}(\overline{X})}\big] \leqslant M \max\big\{|Q_{2\epsilon}(\overline{X})|, |Q_{1-\epsilon/2}(\overline{X})|\big\}^{1-\gamma} \leqslant 2\epsilon\Big(\frac{2M}{\epsilon}\Big)^{2/(1+\gamma)}.$$

All in all, we have shown the inequality:

$$\mathbb{E}\big[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X) - \mathbb{E}[\phi_{\mu+Q_{2\epsilon}(\overline{X}),\mu+Q_{1-\epsilon/2}(\overline{X})}(X)]\big]^2 \leqslant 6\epsilon\Big(\frac{2M}{\epsilon}\Big)^{2/(1+\gamma)},$$

which we now use to apply Bernstein's inequality on the sum in (3.9.13) to find, conditionally on $Y_1, \ldots, Y_n$, with probability at least $1 - \delta/4$:

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N} \phi_{\alpha,\beta}(X_i) \\
&\leqslant \mu + \overline{\mathcal{E}}(4\epsilon, X) + \sqrt{\frac{6\epsilon \log(4/\delta)}{N}}\Big(\frac{2M}{\epsilon}\Big)^{1/(1+\gamma)} + \frac{\log(4/\delta)}{3N}(Q_{1-\epsilon/2}(\overline{X}) + \overline{\mathcal{E}}(\epsilon, X)) \\
&\leqslant \mu + 2\overline{\mathcal{E}}(4\epsilon, X) + \sqrt{\frac{6\epsilon \log(4/\delta)}{N}}\Big(\frac{2M}{\epsilon}\Big)^{1/(1+\gamma)} + \frac{\log(4/\delta)}{3N} Q_{1-\epsilon/2}(\overline{X}) \\
&\leqslant \mu + 2\overline{\mathcal{E}}(4\epsilon, X) + (3/2) M^{1/(1+\gamma)} (\epsilon/2)^{\gamma/(1+\gamma)},
\end{aligned}$$

where we used (3.9.15), the fact that $\frac{\log(4/\delta)}{N} \leqslant \epsilon/12$ and the assumption that $\delta \geqslant e^{-N}/4$. Using the same argument on the lower tail, we obtain, on the event E, that with probability at least

$1-\delta/2$

$$\Big|\frac{1}{N}\sum_{i=1}^{N}\phi_{\alpha,\beta}(X_i)-\mu\Big| \leqslant 2\overline{\mathcal{E}}(4\epsilon,X)+(3/2)M^{\frac{1}{1+\gamma}}(\epsilon/2)^{\gamma/(1+\gamma)}.$$

Step 3. Now we show that $\Big|\frac{1}{N}\sum_{i=1}^{N}\phi_{\alpha,\beta}(X_i)-\frac{1}{N}\sum_{i=1}^{N}\phi_{\alpha,\beta}(\widetilde{X}_i)\Big|$ is of the same order as the previous bounds. There are at most $2\eta N$ indices such that $X_i\neq\widetilde{X}_i$ and for such differences we have the bound:

$$\big|\phi_{\alpha,\beta}(X_i)-\phi_{\alpha,\beta}(\widetilde{X}_i)\big| \leqslant |Q_{\epsilon/2}(\overline{X})|+|Q_{1-\epsilon/2}(\overline{X})|,$$

and since we have $\eta\leqslant\epsilon/8$ then

$$\begin{aligned}\Big|\frac{1}{N}\sum_{i=1}^{N}\phi_{\alpha,\beta}(X_i)-\frac{1}{N}\sum_{i=1}^{N}\phi_{\alpha,\beta}(\widetilde{X}_i)\Big| &\leqslant 2\eta\big(|Q_{\epsilon/2}(\overline{X})|+|Q_{1-\epsilon/2}(\overline{X})|\big)\\ &\leqslant \frac{\epsilon}{2}\max\big\{|Q_{\epsilon/2}(\overline{X})|,|Q_{1-\epsilon/2}(\overline{X})|\big\}\\ &\leqslant M^{1/(1+\gamma)}(\epsilon/2)^{\gamma/(1+\gamma)},\end{aligned}$$

where the last step follows from (3.9.14) and (3.9.15). Finally, using similar arguments along with Hölder's inequality, we show that:

$$\begin{aligned}\mathbb{E}\big[|\overline{X}-Q_{\epsilon/2}(\overline{X})|\mathbf{1}_{\overline{X}\leqslant Q_{\epsilon/2}(\overline{X})}\big] &\leqslant \mathbb{E}\big[|\overline{X}|\mathbf{1}_{\overline{X}\leqslant Q_{\epsilon/2}(\overline{X})}\big]+\mathbb{E}\big[|Q_{\epsilon/2}(\overline{X})|\mathbf{1}_{\overline{X}\leqslant Q_{\epsilon/2}(\overline{X})}\big]\\ &\leqslant M^{1/(1+\gamma)}(\epsilon/2)^{\gamma/(1+\gamma)}+|Q_{\epsilon/2}(\overline{X})|(\epsilon/2)\\ &\leqslant 2M^{1/(1+\gamma)}(\epsilon/2)^{\gamma/(1+\gamma)},\end{aligned}$$

and a similar computation for $\mathbb{E}\big[|\overline{X}-Q_{1-\epsilon/2}(\overline{X})|\mathbf{1}_{\overline{X}\geqslant Q_{1-\epsilon/2}(\overline{X})}\big]$ leads to

$$\overline{\mathcal{E}}(4\epsilon,X)\leqslant 2M^{1/(1+\gamma)}(2\epsilon)^{\gamma/(1+\gamma)}.$$

This completes the proof of Lemma 3.9. □

Proof of Proposition 3.3

Step 1. Notice that the `TM` estimator is also a monotonous non decreasing function of each of its entries when the others are fixed. This allows us to replicate Step 1 of the proof of Proposition 3.1. We define an ε-net N_ε on the set Θ, fix $\theta\in\Theta$ and let $\widetilde{\theta}$ be the closest point in N_ε. We obtain, for all $j\in[\![d]\!]$, the inequalities:

$$\begin{aligned}\big|\widehat{g}_j^{\mathtt{TM}}(\theta)-g_j(\theta)\big| &\leqslant \big|\widehat{g}_j^{\mathtt{TM}}(\theta)-g_j(\widetilde{\theta})\big|+\big|g_j(\widetilde{\theta})-g_j(\theta)\big|\\ &\leqslant \big|\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta})-g_j(\widetilde{\theta})\big|+\varepsilon L_j,\end{aligned}\tag{3.9.16}$$

where $\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta})$ is the `TM` estimator obtained for the entries $\ell'\big(\widetilde{\theta}^\top X_i,Y_i\big)X_i^j+\varepsilon\gamma\|X_i\|^2=g_j^i(\widetilde{\theta})+\varepsilon\gamma\|X_i\|^2$.

Step 2. We use the concentration property of the `TM` estimator to bound the previous quantity which is in terms of $\widetilde{\theta}$. The terms $\big(g_j^i(\widetilde{\theta})+\varepsilon\gamma\|X_i\|^2\big)_{i\in[\![n]\!]}$ are independent and distributed according to $Z:=\ell'\big(\widetilde{\theta}^\top X,Y\big)X^j+\gamma\varepsilon\|X\|^2$. Obviously we have $\mathbb{E}\ell'\big(\theta^\top X,Y\big)X^j=g_j(\theta)$. Furthermore, let $\overline{L}=\mathbb{E}\gamma\|X\|^2$, so that $\mathbb{E}\big[g_j^i(\widetilde{\theta})+\varepsilon\gamma\|X_i\|^2\big]=g_j(\theta)+\varepsilon\overline{L}$. We will apply Lemma 3.9 for $\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta})$.

Before we do so, we need to compute the centered $(1+\alpha)$-moment of Z. Let $m_{j,\alpha}(\widetilde{\theta})$ and $m_{L,\alpha}$ be the centered $(1+\alpha)$-moments of $\ell'(\theta^\top X, Y)X^j$ and $\gamma\|X\|^2$ respectively, we have:

$$\mathbb{E}\big|Z - \mathbb{E}Z\big|^{1+\alpha} \leqslant 2^\alpha\big(m_{j,\alpha}(\theta) + \varepsilon^{1+\alpha} m_{L,\alpha}\big).$$

Now applying Lemma 3.9 we find with probability no less than $1-\delta$

$$\big|\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta}) - g_j(\widetilde{\theta}) - \varepsilon\overline{L}\big| \leqslant 7\big(m_{j,\alpha}(\widetilde{\theta}) + \varepsilon^{1+\alpha} m_{L,\alpha}\big)^{1/(1+\alpha)}(2\epsilon)^{\alpha/(1+\alpha)},$$

with $\epsilon_\delta = 8\eta + 12\frac{\log(4/\delta)}{n}$. By combining with (3.9.16) and using a union bound argument, we deduce that with the same probability, we have for all $j \in [\![d]\!]$

$$\big|\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| \leqslant 7\big(m_{j,\alpha}(\widetilde{\theta}) + \varepsilon^{(1+\alpha)^2} m_{L,\alpha}\big)^{1/(1+\alpha)}(4\epsilon_{d\delta})^{\alpha/(1+\alpha)} + \varepsilon\overline{L}. \tag{3.9.17}$$

Step 3. We use the ε-net to obtain a uniform bound. We proceed similarly as in the proof of Proposition 3.1. For $\theta \in \Theta$ denote $\widetilde{\theta}(\theta) \in N_\varepsilon$ the closest point in N_ε satisfying in particular $\|\widetilde{\theta}(\theta) - \theta\| \leqslant \varepsilon$, we write, following previous arguments

$$\begin{aligned}
\sup_{\theta\in\Theta}\big|\widehat{g}_j^{\mathtt{TM}}(\theta) - g_j(\theta)\big| &\leqslant \sup_{\theta\in\Theta}\big|\widehat{g}_j^{\mathtt{TM}}(\theta) - g_j(\widetilde{\theta}(\theta))\big| + \big|g_j(\widetilde{\theta}(\theta)) - g_j(\theta)\big| \\
&\leqslant \sup_{\theta\in\Theta}\big|\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta}(\theta)) - g_j(\widetilde{\theta}(\theta))\big| + \varepsilon L_j \\
&= \max_{\widetilde{\theta}\in N_\varepsilon}\big|\breve{g}_j^{\mathtt{TM}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| + \varepsilon L_j.
\end{aligned}$$

Taking union bound over $\widetilde{\theta} \in N_\varepsilon$ for the inequality (3.9.17) and choosing $\varepsilon = n^{-\alpha/(1+\alpha)}$ concludes the proof of Proposition 3.3.

Proof of Corollary 3.1

We first write the result of Proposition 3.3 with a big O notation. This tells us that with probability at least $1-\delta$ for all $\theta \in \Theta$, for all $j \in [\![d]\!]$ we have :

$$\big|\epsilon_j^{\mathtt{TM}}(\delta)\big| \leqslant O\bigg(M_{j,\alpha}^{1/(1+\alpha)}\Big(\frac{\log(d/\delta) + d\log(n)}{n}\Big)^{\alpha/(1+\alpha)}\bigg)$$

It only remains to apply Theorem 3.1 with importance sampling. The main result corresponds to having the second term (the statistical error) dominate the bound given by Theorem 3.1. This happens as soon as the number of iterations T is high enough so that

$$\big(R(\theta^{(0)}) - R^\star\big)\Big(1 - \frac{\lambda}{\sum_{j\in[\![d]\!]} L_j}\Big)^T \leqslant \frac{\big\|\epsilon^{\mathtt{TM}}(\delta)\big\|_2^2}{2\lambda}.$$

From here, it is straightforward to check that the stated number of iterations suffices.

Proof of Lemma 3.4

Similarly to the proof of Lemma 3.2, the assumptions, this time taken with $\alpha = 1$, imply that the gradient has a second moment so that the existence of $\sigma_j^2 = \mathbb{V}(g_j(\theta))$ is guaranteed. We

apply Lemma 1 from [191] with $\delta/2$ to obtain:

$$\frac{1}{2}|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\theta)| \leqslant \frac{C\sigma_j^2}{s} + \frac{s\log(4\delta^{-1})}{n}$$

with probability at least $1-\delta/2$, where C is a constant such that we have:

$$-\log(1-u+Cu^2) \leqslant \psi(u) \leqslant \log(1+u+Cu^2),$$

and one can easily check that our choice of ψ, the Gudermannian function, satisfies the previous inequality for $C = 1/2$. This, along with the choice of scale s according to (3.3.10) and our assumption on $\widehat{\sigma}_j$ yields the announced deviation bound by a simple union bound argument.

Proof of Proposition 3.4

In this proof, for a scale $s > 0$ and a set of real numbers $(x_i)_{i\in[\![n]\!]}$, we let $\bar{x} = \frac{1}{n}\sum_{i\in[\![n]\!]} x_i$ be their mean and define the function $\zeta_s\big((x_i)_{i\in[\![n]\!]}\big)$ as the unique x satisfying

$$\sum_{i\in[\![n]\!]} \psi\Big(\frac{x-\bar{x}}{s}\Big) = 0.$$

Since the function ψ is increasing the previous equation has a unique solution. Moreover, for fixed scale s, the function $\zeta_s\big((x_i)_{i\in[\![n]\!]}\big)$ is monotonous non decreasing w.r.t. each x_i when the others are fixed.

Step 1. We proceed similarly as in the proof of Proposition 3.1 except that we only use the monotonicity of the `CH` estimator with fixed scale. Let N_ε be an ε-net for Θ with $\varepsilon = 1/\sqrt{n}$. We have $|N_\varepsilon| \leqslant (3\Delta/2\varepsilon)^d$ with Δ the diameter of Θ. Fix a coordinate $j \in [\![d]\!]$, a point $\theta \in \Theta$ and let $\widetilde{\theta}$ be the closest point to it in the ε-net. We wish to bound the difference

$$\begin{aligned}\big|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\theta)\big| &\leqslant \big|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\widetilde{\theta})\big| + \big|g_j(\widetilde{\theta}) - g_j(\theta)\big| \\ &\leqslant \big|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\widetilde{\theta})\big| + \varepsilon L_j,\end{aligned}$$

where we have the `CH` estimator $\widehat{g}_j^{\texttt{CH}}(\theta) = \zeta_{s(\theta)}\big((g_j^i(\theta))_{i\in[\![n]\!]}\big)$ with scale $s(\theta)$ computed according to (3.3.10) and (3.3.11). Assume, without loss of generality that $\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\widetilde{\theta}) \geqslant 0$. Using the non-decreasing property of the `CH` estimator at a fixed scale, we find that

$$\begin{aligned}\big|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\widetilde{\theta})\big| &= \big|\zeta_{s(\theta)}\big((g_j^i(\theta))_{i\in[\![n]\!]}\big) - g_j(\widetilde{\theta})\big| \\ &\leqslant \big|\zeta_{s(\theta)}\big((g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)_{i\in[\![n]\!]}\big) - g_j(\widetilde{\theta})\big|.\end{aligned}$$

Indeed, one has

$$\begin{aligned}g_j^i(\theta) &= g_j^i(\widetilde{\theta}) + \big(g_j^i(\theta) - g_j^i(\widetilde{\theta})\big) \\ &\leqslant g_j^i(\widetilde{\theta}) + \gamma\|\widetilde{\theta} - \theta\| \cdot \|X_i\| \cdot |X_i^j| \\ &\leqslant g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2.\end{aligned}$$

We introduce the notation $\breve{g}_j^{\texttt{CH}}(\widetilde{\theta}) := \zeta_{s(\theta)}\big((g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)_{i\in[\![n]\!]}\big)$ so that:

$$\big|\widehat{g}_j^{\texttt{CH}}(\theta) - g_j(\widetilde{\theta})\big| \leqslant \big|\breve{g}_j^{\texttt{CH}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big|.$$

Step 2. We now use the concentration property of CH to bound the previous quantity which is in terms of $\widetilde{\theta}$. We apply Lemma 1 from [191] with $\delta/2$ and scale $s(\theta)$ to the samples $(g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)_{i\in[\![n]\!]}$ which are independent and distributed according to the random variable $\ell'\big(\widetilde{\theta}^\top X, Y\big)X^j + \varepsilon\gamma\|X\|^2$ with expectation $g_j(\widetilde{\theta}) + \varepsilon\overline{L}$. Using our assumptions on $\sigma_L, \sigma_j(\theta), \sigma_j(\widetilde{\theta}), \widehat{\sigma}_j(\theta)$ and the definition of the scale $s(\theta)$ according to (3.3.10) we find:

$$\begin{aligned}
\frac{1}{2}\big|\breve{g}_j^{\mathtt{CH}}(\widetilde{\theta}) - g_j(\widetilde{\theta}) - \varepsilon\overline{L}\big| &= \frac{1}{2}\big|\zeta_{s(\theta)}\big((g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)_{i\in[\![n]\!]}\big) - g_j(\widetilde{\theta}) - \varepsilon\overline{L}\big| \\
&\leqslant \frac{C\mathbb{V}(g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)}{s(\theta)} + \frac{s(\theta)\log(4/\delta)}{n} \\
&\leqslant \frac{CC'\mathbb{V}(g_j^i(\widetilde{\theta}) + \varepsilon\gamma\|X_i\|^2)}{\sigma_j(\theta)}\sqrt{\frac{2\log(4/\delta)}{n}} + C'\sigma_j(\theta)\sqrt{\frac{2\log(4/\delta)}{n}} \\
&\leqslant \frac{CC'2(\sigma_j^2(\widetilde{\theta}) + \varepsilon^2\sigma_L^2)}{\sigma_j(\theta)}\sqrt{\frac{2\log(4/\delta)}{n}} + C'\sigma_j(\theta)\sqrt{\frac{2\log(4/\delta)}{n}} \\
&\leqslant CC'2\big(\sqrt{2}\sigma_j(\widetilde{\theta}) + \varepsilon\sigma_L\big)\sqrt{\frac{2\log(4/\delta)}{n}} + 2C'\sigma_j(\widetilde{\theta})\sqrt{\frac{\log(4/\delta)}{n}} \\
&\leqslant 4C'\sigma_j(\widetilde{\theta})\sqrt{\frac{\log(4/\delta)}{n}} + 2C'\varepsilon\sigma_L\sqrt{\frac{\log(4/\delta)}{n}} \\
&\leqslant 2C'(2\sigma_j(\widetilde{\theta}) + \varepsilon\sigma_L)\sqrt{\frac{\log(4/\delta)}{n}}.
\end{aligned}$$

A simple union bound yields that for all $j \in [\![d]\!]$

$$\big|\breve{g}_j^{\mathtt{CH}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| \leqslant 4C'(2\sigma_j(\widetilde{\theta}) + \varepsilon\sigma_L)\sqrt{\frac{\log(4d/\delta)}{n}} + \varepsilon\overline{L}. \tag{3.9.18}$$

Step 3. We use the ε-net to obtain a uniform bound. We proceed similarly to the proof of Proposition 3.1. For $\theta \in \Theta$ denote $\widetilde{\theta}(\theta) \in N_\varepsilon$ the closest point in N_ε satisfying in particular $\|\widetilde{\theta}(\theta) - \theta\| \leqslant \varepsilon$, we write, following previous arguments

$$\begin{aligned}
\sup_{\theta\in\Theta}\big|\widehat{g}_j^{\mathtt{CH}}(\theta) - g_j(\theta)\big| &\leqslant \sup_{\theta\in\Theta}\big|\widehat{g}_j^{\mathtt{CH}}(\theta) - g_j(\widetilde{\theta}(\theta))\big| + \big|g_j(\widetilde{\theta}(\theta)) - g_j(\theta)\big| \\
&\leqslant \sup_{\theta\in\Theta}\big|\breve{g}_j^{\mathtt{CH}}(\widetilde{\theta}(\theta)) - g_j(\widetilde{\theta}(\theta))\big| + \varepsilon L_j \\
&= \max_{\widetilde{\theta}\in N_\varepsilon}\big|\breve{g}_j^{\mathtt{CH}}(\widetilde{\theta}) - g_j(\widetilde{\theta})\big| + \varepsilon L_j.
\end{aligned}$$

Taking union bound over $\widetilde{\theta} \in N_\varepsilon$ for the inequality (3.9.18) and using the choice $\varepsilon = 1/\sqrt{n}$ concludes the proof of Proposition 3.4.

Proof of Corollary 3.2

Under the assumptions made, the constants $(L_j)_{j\in[\![d]\!]}$ are estimated using the MOM estimator and we obtain the bounds $(\overline{L}_j)_{j\in[\![d]\!]}$ which hold with probability at least $1 - \delta/2$ by a union bound argument. The rest of the proof is the same as that of Theorem 3.1 using a failure probability $\delta/2$ instead of δ and replacing the constants $(L_j)_{j\in[\![d]\!]}$ by their upperbounds accordingly. The result then follows after a simple union bound argument.

Proof of Lemma 3.5

Let $B_1, \dots, B_K$ be the blocks used for the estimation so that $B_1 \cup \cdots \cup B_K = [\![n]\!]$ and $B_{k_1} \cap B_{k_2} = \emptyset$ for $k_1 \neq k_2$. Let $\mathcal{K}$ denote the uncorrupted block indices $\mathcal{K} = \{k \in [\![K]\!] \text{ such that } B_k \cap \mathcal{O} = \emptyset\}$ and assume $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$. For $k \in [\![K]\!]$ let $\widehat{\sigma}_k^2 = \frac{K}{n}\sum_{i\in B_k} X_i^2$ be the block means computed by MOM. Denote $N = n/K$, by using (a slight generalization of) Lemma 3.7 and the $L^{(1+\alpha)}$-L^1 condition satisfied by X^2 with a known constant C, we obtain that with probability at least $1-\delta$ we have

$$|\widehat{\sigma}_k^2 - \sigma^2| \leqslant \Big(\frac{3\mathbb{E}|X^2-\sigma^2|^{1+\alpha}}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}} \leqslant \Big(\frac{3}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}} C\mathbb{E}|X^2-\sigma^2| \leqslant \Big(\frac{3}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}} C\sigma^2,$$

which implies the inequality

$$\sigma^2 \leqslant \Big(1 - C\Big(\frac{3}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}}\Big)^{-1} \widehat{\sigma}_k^2.$$

Define the Bernoulli random variables $U_k = \mathbf{1}\Big\{\sigma^2 > \Big(1 - C\big(\frac{3}{\delta N^\alpha}\big)^{\frac{1}{1+\alpha}}\Big)^{-1}\widehat{\sigma}_k^2\Big\}$ for $k \in [\![K]\!]$ which have success probability $\leqslant \delta$. Denote $S = \sum_k U_k$, we can bound the failure probability of the estimator as follows:

$$\begin{aligned}
\mathbb{P}\Big(\Big(1 - C\Big(\frac{3}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}}\Big)^{-1}\widehat{\sigma}^2 < \sigma^2\Big) &\leqslant \mathbb{P}\big[S > K/2 - |\mathcal{O}|\big] \\
&= \mathbb{P}\big[S - \mathbb{E}S > K/2 - |\mathcal{O}| - \delta|\mathcal{K}|\big] \\
&\leqslant \mathbb{P}\big[S - \mathbb{E}S > K(\varepsilon - 2\delta)/2\big] \\
&\leqslant \exp\big(-K(\varepsilon-2\delta)^2/2\big),
\end{aligned}$$

where we used the fact that $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$ and $|\mathcal{K}| \leqslant K$ for the second inequality and Hoeffding's inequality for the last. The proof is finished by taking $\varepsilon = 5/6$ and $\delta = 1/4$.

Proof of Lemma 3.6

Lemma 3.6 is a direct consequence of the following result.

Lemma 3.10. *Let $X_1, \dots, X_n$ an i.i.d sample of a random variable X with expectation $\mathbb{E}X = \mu$ and $(1+\alpha)$-moment $\mathbb{E}|X-\mu|^{1+\alpha} = m_\alpha < \infty$. Assume that the variable X satisfies the $L^{(1+\alpha)^2}$-$L^{(1+\alpha)}$ condition with constant $C > 1$. Let $\widehat{\mu}$ be the median-of-means estimate of μ with K blocks and $\widehat{m}_\alpha$ a similarly obtained estimate of m_α from the samples $(|X_i - \widehat{\mu}|^{1+\alpha})_{i\in[\![n]\!]}$. Then, with probability at least $1 - 2\exp(-K/18)$ we have*

$$\widehat{m}_\alpha \geqslant (1-\kappa)m_\alpha,$$

with $\kappa = \epsilon + 24(1+\alpha)\Big(\frac{1+\epsilon}{n/K}\Big)^{\frac{\alpha}{1+\alpha}}$ and $\epsilon = \Big(\frac{3\times 2^{2+\alpha}(1+C^{(1+\alpha)^2})}{(n/K)^\alpha}\Big)^{\frac{1}{1+\alpha}}$.

Proof. Let $\widehat{\mu}$ be the MOM estimate of μ with K blocks, using Lemma 3.2, we have with probability at least $1 - \exp(-K/18)$,

$$|\mu - \widehat{\mu}| > (24m_\alpha)^{\frac{1}{1+\alpha}}\Big(\frac{K}{n}\Big)^{\frac{\alpha}{1+\alpha}}. \tag{3.9.19}$$

Let $\widehat{m}_\alpha$ be the MOM estimate of m_α obtained from the samples $\big(|X_i - \widehat{\mu}|^{1+\alpha}\big)_{i\in[\![n]\!]}$. Denote

$B_1, \dots, B_K$ the blocks we use, we have:

$$\widehat{m}_\alpha = \text{median}\left(\frac{K}{n}\sum_{i\in B_j}|X_i - \widehat{\mu}|^{1+\alpha}\right)_{j\in[\![K]\!]}$$

for any $i \in [\![n]\!]$. Let $N = n/K$, using the convexity of the function $f(x) = |x|^{1+\alpha}$ we find that:

$$\begin{aligned}
\frac{1}{N}\sum_{i\in B_j}\big|X_i - \widehat{\mu}\big|^{1+\alpha} &= \frac{1}{N}\sum_{i\in B_j}\big|(X_i-\mu)+(\mu-\widehat{\mu})\big|^{1+\alpha} \\
&\geqslant \frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{1+\alpha} + \frac{1}{N}(1+\alpha)\sum_{i\in B_j}|X_i-\mu|^{\alpha}\text{sign}(X_i-\mu)(\mu-\widehat{\mu}) \\
&\geqslant \frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{1+\alpha} - (1+\alpha)|\mu-\widehat{\mu}|\Big[\frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{\alpha}\Big] \\
&\geqslant \frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{1+\alpha} - (1+\alpha)|\mu-\widehat{\mu}|\Big[\frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{1+\alpha}\Big]^{\frac{\alpha}{1+\alpha}}, \qquad (3.9.20)
\end{aligned}$$

where the last step uses Jensen's inequality. Using Lemma 3.7 we have, for $\delta > 0$, the concentration bound

$$\mathbb{P}\Big(\Big|\frac{1}{N}\sum_{i\in B_j}\big|X_i-\mu\big|^{1+\alpha} - m_\alpha\Big| > \Big(\frac{3\mathbb{E}\big||X-\mu|^{1+\alpha}-m_\alpha\big|^{1+\alpha}}{\delta N^\alpha}\Big)^{\frac{1}{1+\alpha}}\Big) \leqslant \delta$$

which, using that X satisfies the $L^{(1+\alpha)^2}$-$L^{(1+\alpha)}$ condition, translates to

$$\begin{aligned}
\mathbb{P}\Big(\Big|\frac{1}{N}\sum_{i\in B_j}\big|X_i-\mu\big|^{1+\alpha} - m_\alpha\Big| > \epsilon\Big) &\leqslant \frac{3\mathbb{E}\big||X-\mu|^{1+\alpha}-m_\alpha\big|^{1+\alpha}}{\epsilon^{1+\alpha}N^\alpha} \\
&\leqslant \frac{3\times 2^\alpha\big(\mathbb{E}|X-\mu|^{(1+\alpha)^2} + m_\alpha^{1+\alpha}\big)}{\epsilon^{1+\alpha}N^\alpha} \\
&\leqslant \frac{3\times 2^\alpha m_\alpha^{1+\alpha}\big(1+C^{(1+\alpha)^2}\big)}{\epsilon^{1+\alpha}N^\alpha}.
\end{aligned}$$

Replacing ϵ with ϵm_α we find

$$\mathbb{P}\Big(\Big|\frac{1}{N}\sum_{i\in B_j}|X_i-\mu|^{1+\alpha} - m_\alpha\Big| > \epsilon m_\alpha\Big) \leqslant \frac{3\times 2^\alpha\big(1+C^{(1+\alpha)^2}\big)}{N^\alpha\epsilon^{1+\alpha}}.$$

Now conditioning on the event (3.9.19) and using the previous bound with $\epsilon = \Big(\frac{3\times 2^\alpha\left(1+C^{(1+\alpha)^2}\right)}{N^\alpha\delta}\Big)^{\frac{1}{1+\alpha}}$ in (3.9.20), we obtain that

$$\begin{aligned}
&\mathbb{P}\Big(\frac{1}{N}\sum_{i\in B_j}\big|X_i-\widehat{\mu}\big|^{1+\alpha} \leqslant (1-\epsilon)m_\alpha - (1+\alpha)\Big(\frac{24m_\alpha}{N^\alpha}\Big)^{\frac{1}{1+\alpha}}((1+\epsilon)m_\alpha)^{\frac{\alpha}{1+\alpha}}\Big) \leqslant \delta \\
&\implies \mathbb{P}\Big(\frac{1}{N}\sum_{i\in B_j}\big|X_i-\widehat{\mu}\big|^{1+\alpha} \leqslant \underbrace{\Big(1-\epsilon-24(1+\alpha)\Big(\frac{1+\epsilon}{N}\Big)^{\frac{\alpha}{1+\alpha}}\Big)}_{=:(1-\kappa)}m_\alpha\Big) \leqslant \delta.
\end{aligned}$$

Now define U_j as the indicator variable of the event in the last probability. We have just seen

it has success rate less than δ. We can use the MOM trick, assuming the number of outliers satisfies $|\mathcal{O}| \leqslant K(1-\varepsilon)/2$ for $\varepsilon \in (0,1)$, we have for $S = \sum_j U_j$

$$\begin{aligned}\mathbb{P}(\widehat{m}_\alpha \leqslant (1-\kappa)m_\alpha) &\leqslant \mathbb{P}(S > K/2 - |\mathcal{O}|) \\ &= \mathbb{P}\big[S - \mathbb{E}S > K/2 - |\mathcal{O}| - \delta|\mathcal{K}|\big] \\ &\leqslant \mathbb{P}\big[S - \mathbb{E}S > K(\varepsilon - 2\delta)/2\big] \\ &\leqslant \exp\big(-K(\varepsilon-2\delta)^2/2\big).\end{aligned}$$

Taking $\varepsilon = 5/6$ and $\delta = 1/4$ yields that the previous probability is $\leqslant \exp(-K/18)$. Finally, recall that we conditioned on the event where the deviation $|\mu - \widehat{\mu}|$ is bounded as previously stated and that this event holds with $\geqslant 1 - \exp(-K/18)$. Taking this conditioning into account and using a union bound argument leads to the fact that the bound

$$\widehat{m}_\alpha \geqslant (1-\kappa)m_\alpha$$

holds with probability at least $1 - 2\exp(-K/18)$. □

3.9.3 Proof of Theorem 3.3

This proof is inspired from Theorem 5 in [341] and Theorem 1 in [401] while keeping track of the degradations caused by the errors on the gradient coordinates.

We condition on the event (3.9.1) and denote $\epsilon_j = \epsilon_j(\delta)$ and $\epsilon_{Euc} = \|\epsilon(\delta)\|_2$. We define for all $\theta \in \Theta$

$$\begin{aligned}u_j(\theta) &= \operatorname*{arg\,min}_{\vartheta \in \Theta_j} \widehat{g}_j(\theta)(\vartheta - \theta_j) + \frac{L_j}{2}(\vartheta - \theta_j)^2 + \epsilon_j|\vartheta - \theta_j| \\ &= \mathrm{proj}_{\Theta_j}\big(\theta_j - \beta_j \tau_{\epsilon_j}\big(\widehat{g}_j(\theta)\big)\big)\end{aligned}$$

and denote $\theta^{(t)}$ the optimization iterates for $t = 0, \dots, T$ and j_t the random coordinate sampled at step t and let $\widehat{g}_t = \widehat{g}_{j_t}(\theta^{(t)})$ for brevity. We have that $u_{j_t}(\theta^{(t)})$ satisfies the following optimality condition

$$\forall \vartheta \in \Theta_{j_t} \quad \big(\widehat{g}_t + L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big) + \epsilon_{j_t}\rho_t\big)\big(\vartheta - u_{j_t}(\theta^{(t)})\big) \geqslant 0,$$

where $\rho_t = \mathrm{sign}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)$. Using this condition for $\vartheta = \theta^{(t)}_{j_t}$ and the coordinate-wise Lipschitz smoothness property of R we find

$$\begin{aligned}R(\theta^{(t+1)}) &\leqslant R(\theta^{(t)}) + g_{j_t}(\theta^{(t)})\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big) + \frac{L_{j_t}}{2}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 \\ &\leqslant R(\theta^{(t)}) + (\widehat{g}_t + \epsilon_{j_t}\rho_t)\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big) + \frac{L_{j_t}}{2}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 \qquad (3.9.21) \\ &\leqslant R(\theta^{(t)}) - \frac{L_{j_t}}{2}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2. \qquad (3.9.22)\end{aligned}$$

Defining the potential $\Phi(\theta) = \sum_{j=1}^d L_j(\theta_j - \theta^\star_j)^2$, we have:

$$\begin{aligned}\Phi(\theta^{(t+1)}) &= \Phi(\theta^{(t)}) + 2L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)\big(\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big) + L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 \\ &= \Phi(\theta^{(t)}) + 2L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)\big(u_{j_t}(\theta^{(t)}) - \theta^\star_{j_t}\big) - L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 \\ &\leqslant \Phi(\theta^{(t)}) - 2(\widehat{g}_t + \epsilon_{j_t}\rho_t)\big(u_{j_t}(\theta^{(t)}) - \theta^\star_{j_t}\big) - L_{j_t}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 \\ &= \Phi(\theta^{(t)}) + 2(\widehat{g}_t + \epsilon_{j_t}\rho_t)\big(\theta^\star_{j_t} - \theta^{(t)}_{j_t}\big) - 2\Big((\widehat{g}_t + \epsilon_{j_t}\rho_t)\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)\end{aligned}$$

$$
\begin{aligned}
&+ \frac{L_{j_t}}{2}\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2\Big) \\
&\leqslant \Phi(\theta^{(t)}) + 2(\widehat{g}_t + \epsilon_{j_t}\rho_t)\big(\theta^\star_{j_t} - \theta^{(t)}_{j_t}\big) + 2\big(R(\theta^{(t)}) - R(\theta^{(t+1)})\big) \\
&\leqslant \Phi(\theta^{(t)}) + 2g_{j_t}(\theta^{(t)})\big(\theta^\star_{j_t} - \theta^{(t)}_{j_t}\big) + 2\big(R(\theta^{(t)}) - R(\theta^{(t+1)})\big) + 4\epsilon_{j_t}\big|\theta^\star_{j_t} - \theta^{(t)}_{j_t}\big|,
\end{aligned}
$$

where the first inequality uses the optimality condition with $\vartheta = \theta^\star_{j_t}$ and the second one uses (3.9.21). Now, defining $\Psi(\theta) = \frac{1}{2}\Phi(\theta) + R(\theta)$, taking the expectation w.r.t. j_t and using the convexity of R and a Cauchy-Schwarz inequality, we find

$$
\mathbb{E}\big[\Psi(\theta^{(t)}) - \Psi(\theta^{(t+1)})\big] \geqslant \frac{1}{d}\big(R(\theta^{(t)}) - R(\theta^\star) - 2\epsilon_{Euc}\big\|\theta^{(t)} - \theta^\star\big\|_2\big).
$$

Recall that according to (3.9.22), we have $R(\theta^{(t+1)}) \leqslant R(\theta^{(t)})$, summing over $t = 0, \dots, T$ we find:

$$
\begin{aligned}
\mathbb{E}\Big[\frac{T+1}{d}\big(R(\theta^{(T)}) - R(\theta^\star)\big)\Big] &\leqslant \mathbb{E}\Big[\frac{1}{d}\sum_{t=0}^{T}\big(R(\theta^{(t)}) - R(\theta^\star)\big)\Big] \\
&\leqslant \sum_{t=0}^{T}\Big(\mathbb{E}\big[\Psi(\theta^{(t)}) - \Psi(\theta^{(t+1)})\big] + \frac{2\epsilon_{Euc}}{d}\mathbb{E}\big[\big\|\theta^{(t)} - \theta^\star\big\|_2\big]\Big) \\
&= \mathbb{E}\big[\Psi(\theta^{(0)}) - \Psi(\theta^{(t+1)})\big] + \frac{2\epsilon_{Euc}}{d}\sum_{t=0}^{T}\mathbb{E}\big[\big\|\theta^{(t)} - \theta^\star\big\|_2\big] \\
&\leqslant \Psi(\theta^{(0)}) + \frac{2\epsilon_{Euc}}{d}\sum_{t=0}^{T}\mathbb{E}\big[\big\|\theta^{(t)} - \theta^\star\big\|_2\big],
\end{aligned}
$$

which yields the result after multiplying by $\frac{d}{T+1}$. To finish, we show that conditionally on any choice of j_t we have $\|\theta^{(t+1)} - \theta^\star\|_2 \leqslant \|\theta^{(t)} - \theta^\star\|_2$. Indeed a straightforward computation yields

$$
\big\|\theta^{(t+1)} - \theta^\star\big\|_2^2 = \big\|\theta^{(t)} - \theta^\star\big\|_2^2 + \big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)^2 + 2\big(u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t}\big)\big(\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big).
$$

We need to show that $\delta_t^2 \leqslant -2\delta_t\big(\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big)$ with $\delta_t = (u_{j_t}(\theta^{(t)}) - \theta^{(t)}_{j_t})$. Notice that δ_t always has the opposite sign of $g_{j_t}(\theta^{(t)})$ (thanks to the thresholding) so by convexity of R along the coordinate j_t we have $\delta_t\big(\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big) \leqslant 0$ and so it is down to showing $|\delta_t| \leqslant 2\big|\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big|$ which can be seen from

$$
|\delta_t| \leqslant \frac{\big|g_{j_t}(\theta^{(t)})\big|}{L_{j_t}} = \frac{\big|g_{j_t}(\theta^{(t)}) - g_{j_t}(\theta^\star)\big|}{L_{j_t}} \leqslant \big|\theta^{(t)}_{j_t} - \theta^\star_{j_t}\big|,
$$

which concludes the proof of Theorem 3.3.

Chapter 4

Robust High-Dimensional Learning

This chapter is based on the article [310] in collaboration with Stéphane Gaïffas.

Contents

Abstract

We propose statistically robust and computationally efficient linear learning methods in the high-dimensional batch setting, where the number of features d may exceed the sample size n. We employ, in a generic learning setting, two algorithms depending on whether the considered loss function is gradient-Lipschitz or not. Then, we instantiate our framework on several applications including vanilla sparse, group-sparse and low-rank matrix recovery. This leads, for each application, to efficient and robust learning algorithms, that reach near-optimal estimation rates under heavy-tailed distributions and the presence of outliers. For vanilla s-sparsity, we are able to reach the $s\log(d)/n$ rate under heavy-tails and η-corruption, at a computational cost comparable to that of non-robust analogs. We provide an efficient implementation of our algorithms in an open-source `Python` library called `linlearn`, by means of which we carry out numerical experiments which confirm our theoretical findings together with a comparison to other recent approaches proposed in the literature.

4.1 Introduction

Learning from heavy tailed or corrupted data is a long pursued challenge in statistics receiving considerable attention in literature [203, 176, 201, 364, 114, 13] and gaining an additional degree of complexity in the high-dimensional setting [286, 259, 101, 19, 273, 274, 145]. Sparsity inducing penalization techniques [127, 60] are the go-to approach for high-dimensional data and have found many applications in modern statistics [414, 60, 127, 181]. A clear favourite is the Least Absolute Shrinkage and Selection Operator (LASSO) [414]. Theoretical studies have shown that under the so-called Restricted Eigenvalue (RE) condition, the latter achieves a nearly-optimal estimation rate [35, 60, 337]. Further, a rich literature extensively studies the oracle performances of LASSO in various contexts and conditions [61, 279, 448, 454, 455, 461, 425, 264, 27]. Other penalization techniques induce different sparsity patterns or lead to different statistical guarantees, such as, to cite but a few, SLOPE [44, 410] which is adaptive to the unknown sparsity and leads to the optimal estimation rate, OSCAR [45] which induces feature grouping or group-ℓ_1 penalization [447, 198] which induces block-sparsity. Other approaches include for instance Iterative Hard Thresholding (IHT) [40, 41, 212, 405, 211] whose properties are studied under the Restricted Isometry Property (RIP). Another close problem is low-rank matrix recovery, involving the nuclear norm as a low-rank inducing convex penalization [241, 68, 67, 378, 335, 336]. The high-dimensional statistical inference methods cited above are, however, not robust: theoretical guarantees are derived under light-tails (generally sub-Gaussian) and the i.i.d assumption. Unfortunately, these assumptions fail to hold in general, for instance, it is known that financial and biological data often displays heavy-tailed behaviour [145] and outliers or corruption are not uncommon when handling massive amounts of data which are tedious to thoroughly clean. Moreover, the majority of the previous references focus on the oracle performance of estimators as opposed to providing guarantees for explicit algorithms to compute them. A natural question is therefore : *can one build alternatives to such high-dimensional estimators that are robust to heavy tails and outliers, that are computationally efficient and achieve rates similar to their non-robust counterparts ?* Recent advances about robust mean estimation [73, 288, 117, 107, 268] gave a strong impulse in the field of robust learning [262, 364, 191, 114, 91, 18], including the high-dimensional setting [273, 274, 19] which led to significant progress towards a positive answer to this question.

However, to the best of our knowledge, the solutions proposed until now are all suboptimal in one way or another. The shortcomings either lie in the obtained statistical rate: which is sometimes significantly far from optimal, or in the robustness: most works consider heavy tailed and corrupted data separately and only very limited amounts of corruption, or in computational complexity: some corruption-filtering algorithms are too heavy and do not scale to real world applications.

In this paper, we propose explicit algorithms to solve multiple sparse estimation problems with high performances in all previous aspects. In particular, our algorithm for vanilla sparse estimation enjoys a nearly optimal statistical rate (up to a logarithmic factor), is simultaneously robust to heavy tails and strong corruption (when a fraction of the data is corrupted) and has a comparable computational complexity to a non robust method.

4.1.1 Main contributions.

This paper combines non-Euclidean optimization algorithms and robust mean estimators of the gradient into explicit algorithms in order to achieve the following main contributions.

- We propose a framework for robust high-dimensional linear learning in the batch setting using two linearly converging stage-wise algorithms for high-dimensional optimization based on Mirror Descent and Dual Averaging. These may be applied for smooth and non-smooth objectives respectively so that most common loss functions are covered. The previous algorithms may be plugged with an appropriate gradient estimator to obtain explicit robust algorithms for solving a variety of problems.

- The central application of our framework is an algorithm for "vanilla" s-sparse estimation reaching the nearly optimal $s\log(d)/n$ statistical rate in the batch setting by combining stage-wise Mirror Descent (Section 4.3) with a simple trimmed mean estimator of the gradient. This algorithm is simultaneously robust to heavy-tailed distributions and η-corruption of the data[1]. This improves over previous literature which considered the two issues separately or required a very restricted value of the corruption rate η.

- In addition to vanilla sparsity, we instantiate our procedures for group sparse estimation and low-rank matrix recovery, in which different metrics on the parameter space are induced and used to measure the statistical error on the gradient (Section 4.5). For heavy-tailed data and η-corruption, the gradient estimator we propose for vanilla sparsity enjoys an optimal statistical rate with respect to the induced metric while the one proposed for group sparsity is nearly optimal up to a logarithmic factor. Moreover, for heavy tailed data and a limited number of outliers[2], our proposed gradient estimator for low-rank matrix recovery is nearly optimal. Thus, our solutions to each of these problems are the most robust yet in the literature.

- Our algorithms offer a good compromise between robustness and computational efficiency with the only source of overhead coming from the robust gradient estimation component. In particular, for vanilla sparse estimation, this overhead is minimal so that the asymptotic complexity of our procedure is *equivalent* to that of non-robust counterparts. This is in contrast with previous works requiring costly sub-procedures to filter out corruption.

[1]We say that data is η-corrupted for some number $0 < \eta < 1/2$ if an η fraction of the samples is replaced by arbitrary (and potentially adversarial) outliers after data generation.

[2]For low-rank matrix recovery, our estimator is based on Median-Of-Means so that the number of tolerated outliers is up to $K/2$ where K is the number of blocks used for estimation (see Section 4.5.3)

- We validate our results through numerical experiments using synthetic data for regression and real data sets for classification (Section 4.6). Our experiments confirm our mathematical results and compare our algorithms to concurrent baselines from literature.

- All algorithms introduced in this paper as well as the main baselines from literature we use for comparisons are implemented and easily accessible in a few lines of code through our `Python` library called `linlearn`, open-sourced under the BSD-3 License on `GitHub` and available here[3].

4.1.2 Related Works

The general problem of robust linear learning was addressed by [153] where the performance of coordinate gradient descent using various estimators was studied and experimentally evaluated. Several other works [364, 356, 262, 191] deal with this problem, however, they do not consider the high-dimensional setting.

The early work of [3] focuses on vanilla sparse recovery in the stochastic optimization setting and uses a multistage annealed LASSO algorithm where the penalty shrinks progressively. The method reaches the nearly optimal $s\log(d)/n$ rate, however, it is not robust since the data is assumed i.i.d sub-Gaussian. The subsequent work of [399] extends this framework to other problems such as additive sparse and low-rank matrix decomposition by changing the optimization algorithm but the sub-Gaussian assumption remains necessary.

More recently, [223] proposed a stochastic optimization mirror descent algorithm which computes multiple solutions on disjoint subsets of the data and aggregates them with a Median-Of-Means type procedure. The final solution achieves the rate $s\log(d)/n$ under s-vanilla sparsity with sub-Gaussian deviation and an application to low-rank matrix recovery is also developed. This aggregation method can handle some but not all heavy-tailed data. For instance, if the data follows a Pareto(α) distribution then the analysis yields a statistical error with a factor of order $d^{1/\alpha}$ which is not acceptable in a high-dimensional setting. Moreover, the presence of outliers is not considered so that the given bounds do not measure the impact of corruption. Nevertheless, the combination of the restarted mirror descent optimization procedure proposed in [223, Algorithm 1] with the proper robust gradient estimators yields a fast and highly robust learning algorithm in the batch setting which we present in Section 4.3. We also exploit the same core ideas in Section 4.4 to extend our framework to a wider range of objective functions.

Other works consider high-dimensional linear learning methods that are robust to corrupted data. An outlier robust method for mean and covariance estimation in the sparse η-contaminated[4] high-dimensional setting is proposed by [19] along with theoretical guarantees. By extension, these also apply to several problems of interest such as sparse linear estimation or sparse GLMs. The idea is to use an SDP relaxation of sparse PCA [100] in order to adapt the filtering approach from [114], which relies on the covariance matrix to detect outliers, to the high-dimensional setting. However, the need to solve SDP problems makes the algorithm computationally costly. In addition, the true data distribution is assumed Gaussian and the considered η-contamination framework is weaker than η-corruption which allows for adversarial outliers.

The previous ideas were picked up again in the later work of [274] who proposes a robust variant of IHT for sparse regression on η-corrupted data. Unfortunately, these results suffer from several shortcomings since data needs to be Gaussian with a known or sparse covariance matrix. Moreover, the gradient estimation subroutine, which is reminiscent of [19], is computationally heavy since it requires solving SDP problems as well and a number of samples scaling as s^2

[3] `https://github.com/linlearn/linlearn`

[4] η-contamination refers to the case where the data is sampled from a mixed distribution $(1-\eta)P + \eta Q$ where P is the true data distribution and Q is an arbitrary one.

instead of s is required. This seems to come from the fact that sparse gradients need to be estimated, in which case the s^2 dependence is unavoidable based on an oracle lower bound for such an estimation [119].

Recently, [101] derived oracle bounds for a robust estimator in the linear model with Gaussian design and a number of adversarially contaminated labels. Although optimal rates in terms of the corruption are achieved, this setting excludes corruption of the covariates and does not apply for heavy-tailed data distributions. [318] similarly consider the linear model and derive oracle bounds for a robustified SLOPE objective which is adaptive to the sparsity level. Remarkably, they achieve the optimal $s\log(ed/s)/n$ dependence in the heavy-tailed corrupted setting. However, corruption is restricted to the labels and the dependence of the result thereon significantly degrades if the covariates or the noise are not sub-Gaussian. In contrast, the very recent work of [390] considers sparse estimation with heavy-tailed and η-corrupted data and derives a nearly optimal estimation bound using an algorithm which filters the data before running an ℓ_1-penalized robust Huber regression which corresponds to a similar approach to [356] where the non sparse case was treated. Although the $s\log(d)/n$ rate is achieved with optimal robustness, this claim only applies for regression under the linear model with some restrictive assumptions such as zero mean covariates. In addition, little attention is granted to the practical aspect and no experiments are carried out. A later extension [391] improves the statistical rate to $s\log(d/s)/n$ for sub-Gaussian covariates and, if the data covariance is known as well, better dependence on the corruption rate is obtained. Robust high-dimensional linear regression algorithms were recently surveyed by [147] with particular attention payed to methods based on dimension reduction, shrinkage and combinations thereof.

Finally, [273] proposes an IHT algorithm using robust coordinatewise gradient estimators. These results cover the heavy-tailed and η-corrupted settings separately thanks to Median-Of-Means [7, 217, 339] and Trimmed mean [88, 445] estimators respectively. However, the corruption rate η is restricted to be of order at most $O(1/(\sqrt{s}\log(nd)))$ and the question of elaborating an algorithm which is simultaneously robust to both corruption and heavy tails is left open.

We summarize the settings and results of the previously mentioned works, along with ours on vanilla sparse estimation, in Table 4.1 which focuses on robust papers with explicit algorithms.

4.1.3 Agenda

The remainder of this document is structured as follows : Section 4.2 lays out the setting including the definition of the objective and our assumptions on the data. Sections 4.3 and 4.4 define optimization algorithms based on Mirror Descent and Dual Averaging addressing the cases of smooth and non-smooth losses respectively. Both Sections state convergence results for their respective algorithms. Section 4.5 considers instantiations of our general setting to vanilla sparse, group sparse and low-rank matrix estimation for a general loss. In each case, the norm $\|\cdot\|$ and dual norm $\|\cdot\|_*$ are instantiated and a robust and efficient gradient estimator is proposed so that, combined with the results of Sections 4.3 and 4.4, we obtain solutions with nearly optimal statistical rates (up to logarithmic terms) in each case. Finally, Section 4.6 presents numerical experiments on synthetic and real data sets which demonstrate the performance of our proposed methods and compare them with baselines from recent literature.

4.2 Setting, Notation and Assumptions

We consider supervised learning from a data set $(X_i, Y_i)_{i=1}^n$ from which the majority is distributed as a random variable $(X, Y) \in \mathcal{X}\times\mathcal{Y}$ where the covariate space $\mathcal{X}$ is a high-dimensional Euclidean space and the label space $\mathcal{Y}$ is $\mathbb{R}$ or a finite set. The remaining minority of the data are called

Method	Statistical rate	Iteration Complexity	Data dist. and corruption	Loss
AMMD This paper Section 4.3	$O\Big(\sqrt{s}\sqrt{\eta+\frac{\log(d/\delta)}{n}}\Big)$ with $\mathbb{P}\geqslant 1-\delta$	$O(nd)$	L_4 covariates, L_2 labels, η-corruption.	Lip. smooth, QM.
AMDA This paper Section 4.4	$O\Big(\sqrt{s}\sqrt{\eta+\frac{\log(d/\delta)}{n}}\Big)$ with $\mathbb{P}\geqslant 1-\delta$	$O(nd)$	L_4 covariates, L_2 labels, η-corruption.	Lipschitz, PLM.
(Balakrishnan et al., 2017) [19]	$O\Big(\|\theta^\star\|_2\big(\eta\log(1/\eta)+ s\sqrt{\log(d/\delta)/n}\big)\Big)$ with $\mathbb{P}\geqslant 1-\delta$	$\Omega(nd^2+d^3)$	Gaussian, η-contamination.	LSQ, GLMs, Logit.
(Liu et al., 2020) [274]	$O\big(\eta\vee s\sqrt{\log(d/\delta)/n}\big)$ with $\mathbb{P}\geqslant 1-\delta$	$\Omega(nd^2+d^3)$	Gaussian, η-corruption.	LSQ.
(Liu et al., 2019) (MOM) [273]	$O\Big(\sqrt{s\log(d)/n}\Big)$ with $\mathbb{P}\geqslant 1-d^{-2}$	$O(nd)$	L_4 covariates, linear/logit model.	LSQ, Logit.
(Liu et al., 2019) (TMean) [273]	$O\Big(\eta\sqrt{s}\log(nd) +\sqrt{s\log(d)/n}\Big)$ with $\mathbb{P}\geqslant 1-d^{-2}$	$O(nd\log(n))$	sub-Gaussian, η-corruption.	LSQ, Logit.
(Juditsky et al. 2020) [223]	$O\big(\sqrt{s\log(d)\log(1/\delta)/n}\big)$ with $\mathbb{P}\geqslant 1-\delta$	$O(d)$ (stochastic optim.)	L_2 gradient.	Lip. smooth, QM.
(Sasai, 2022) [390]	$O\big(\sqrt{\eta}+\sqrt{s\log(d/\delta)/n}\big)$ with $\mathbb{P}\geqslant 1-\delta$	Preliminary $\Omega(nd^2+d^3)$ then $O(nd)$	zero-mean & L_8 covariates, indep. noise, $n\gtrsim s^2\log(d/\delta)$.	Penalized Huber.

Table 4.1: Summary of the main hypotheses and results of our proposed algorithms and related works in the literature on vanilla sparse estimation. The statistical rate column gives the derived error bound on $\|\widehat{\theta}-\theta^\star\|_2$ between the estimated and true parameter and the associated confidence. In the "Data distribution and corruption" column, rows with no reference to corruption correspond to methods which do not consider it. In the "Loss" column, the following abbreviations are used : QM = quadratic minorization (Assumption 4.4), PLM = pseudo-linear minorization (Assumption 4.6), LSQ = least squares, GLM = generalized linear model, Logit = Logistic, Lip. smooth = Lipschitz smooth (gradient Lipschitz).

outliers and may be completely arbitrary or even adversarial. Given a loss function $\ell : \widehat{\mathcal{Y}}\times\mathcal{Y}\to\mathbb{R}$ where $\widehat{\mathcal{Y}}$ is the prediction space, our goal is to optimize the unobserved objective

$$\mathcal{L}(\theta)=\mathbb{E}[\ell(\langle\theta,X\rangle,Y)] \tag{4.2.1}$$

over a convex set of parameters Θ, where the expectation is taken w.r.t. the joint distribution of (X,Y). Given an optimum $\theta^\star=\arg\min_{\theta\in\Theta}\mathcal{L}(\theta)$ (assumed unique), we are also interested in controlling the estimation error $\|\theta-\theta^\star\|$ where $\|\cdot\|$ is a norm on Θ, which will be defined according to each specific problem (see Section 4.5 below). Moreover, we assume that the optimal parameter is sparse according to an abstract sparsity measure $S:\Theta\to\mathbb{N}$.

Assumption 4.1. *The optimal solution $\theta^\star$ is s-sparse for some integer s smaller than the problem dimension i.e. $S(\theta^\star)\leqslant s$. Additionally, for any s-sparse vector $\theta\in\Theta$ we have the inequality*

$\|\theta\| \leqslant \sqrt{s}\|\theta\|_2$ *and an upper bound* $\bar{s} \geqslant s$ *on the sparsity is known.*

The precise notion of sparsity will be determined later in Section 4.5 through the sparsity measure S depending on the application at hand. The simplest case corresponds to the conventional notion of vector sparsity where $\mathcal{X} = \mathbb{R}^d$ for some large d, $\Theta \subset \mathbb{R}^d$ and the sparsity measure $S(\theta) = \sum_{j \in [\![d]\!]} \mathbf{1}_{\theta_j \neq 0}$ counts the number of non-zero coordinates. However, as we intend to also cover other forms of sparsity later, we do not fix this setting right away. Note that the required knowledge of $\bar{s} \geqslant s$ in Assumption 4.1 is common in the thresholding based sparse learning literature [40, 41, 212, 211, 223]. Adaptive methods to unknown sparsity exist [44, 27, 410] although designing a robust version thereof is beyond the scope of this work.

Since the objective (4.2.1) is not observed due to the distribution of the data being unknown, statistical approximation will be necessary in order to recover an approximation of $\theta^\star$. Instead of estimating the objective itself (which is of limited use for optimization), we will rather compute estimates of the gradient

$$g(\theta) := \nabla_\theta \mathcal{L}(\theta) \tag{4.2.2}$$

in order to run gradient based optimization procedures. Note that, since we consider the high-dimensional setting, a standard gradient descent approach is excluded since it would incur an error strongly depending on the problem dimension. In order to avoid this, one must use a non Euclidean optimization method as is customary for high-dimensional problems [223, 3].

As commonly stated in the robust statistics literature [73, 288, 284], estimating an expectation using a conventional empirical mean only yields values with far from optimal deviation properties in the general case. Several estimators have been proposed which enjoy sub-Gaussian deviations from the true mean and robustness to corruption. Notable examples in the univariate case are the median-of-means (MOM) estimator [7, 217, 339], Catoni's estimator [73] and the trimmed mean [288]. However, in the multivariate case (estimating the mean of a random vector), the optimal sub-Gaussian estimation rate cannot be obtained by a straightforward extension of the previous methods and a line of works [289, 195, 92, 107, 288, 268] has pursued elaborating efficient algorithms to achieve it. Most recently, [117] managed to show that stability based estimators enjoy sub-Gaussian deviations while being robust to corruption of a fraction of the data. However, it is important to remember that all the works we just mentioned measure the estimation error using the Euclidean norm while many other choices are possible which may require the estimation algorithm to be adapted in order to achieve optimal deviations with respect to the chosen norm. This aspect was studied in [285] who gave a norm-dependent formula for the optimal deviation and an algorithm to achieve it, although the latter has exponential complexity and does not consider the presence of outliers.

This is an important aspect to keep in mind in our high-dimensional setting since we will be measuring the statistical error on the gradient using the dual norm $\|\cdot\|_*$ of $\|\cdot\|$ which will never be the Euclidean one.

$$\|v\|_* = \sup_{\|x\| \leqslant 1} \langle v, x \rangle \tag{4.2.3}$$

Of course, apart from the way it is measured, the quality of the estimations one can obtain also crucially depends on the assumptions made on the data. We formally state ours here. We denote $|A|$ as the cardinality of a finite set A and use the notation $[\![k]\!] = \{1, \dots, k\}$ for any integer $k \in \mathbb{N} \setminus \{0\}$.

Assumption 4.2. *The indices of the training samples* $[\![n]\!]$ *can be divided into two disjoint subsets* $[\![n]\!] = \mathcal{I} \cup \mathcal{O}$ *of* outliers $\mathcal{O}$ *and* inliers $\mathcal{I}$ *for which we assume the following*: (*a*) *we have* $|\mathcal{I}| > |\mathcal{O}|$; (*b*) *the pairs* $(X_i, Y_i)_{i \in \mathcal{I}}$ *are i.i.d with distribution* P *and the outliers* $(X_i, Y_i)_{i \in \mathcal{O}}$ *are*

arbitrary; (*c*) *the distribution* P *is such that*:

$$\mathbb{E}\big[\|X\|_2^4\big] < +\infty, \quad \mathbb{E}\big[\|YX\|_2^2\big] < +\infty \quad \textit{and} \quad \mathbb{E}\big[|Y|^2\big] < +\infty. \tag{4.2.4}$$

Moreover, the loss function ℓ *admits constants* $C_{\ell,1}, C_{\ell,2}, C'_{\ell,1}, C'_{\ell,2} > 0$ *such that for all* $z, y \in \widehat{\mathcal{Y}} \times \mathcal{Y}$:

$$|\ell(z,y)| \leqslant C_{\ell,1} + C_{\ell,2}|z-y|^2 \quad \textit{and} \quad |\ell'(z,y)| \leqslant C'_{\ell,1} + C'_{\ell,2}|z-y|,$$

where ℓ' *is the derivative of* ℓ *in its first argument.*

The above hypotheses are sufficient so that the objective function and its gradient exist for any parameter θ and the gradient admits a second moment. The distribution P is allowed to be heavy-tailed and the conditions on ℓ do not go far beyond limiting it to a quadratic behavior and are satisfied by common loss functions for regression and classification[5]. Note that depending on the loss function used and the moment requirements of gradient estimation, the previous moments assumption can be weakened as in [153, Assumption 2] for instance. However, we stick to this version in this work for simplicity. Depending on the gradient estimator, the number of outliers $|\mathcal{O}|$ will be bounded in the subsequent statements either by a constant fraction ηn (η-corruption) of the sample for some $0 < \eta < 1/2$, or by a constant number.

In the two following sections, we will assume that we have a gradient estimator $\widehat{g}$ at our disposal such that for all $\theta \in \Theta$ we have $\widehat{g}(\theta) = g(\theta) + \epsilon(\theta)$ where $\epsilon(\theta)$ represents the random estimation error of the gradient at θ. Same as for the norm $\|\cdot\|$, this estimator will be precisely defined for each individual application in Section 4.5 in such a way that the error $\|\widehat{g}(\theta) - g(\theta)\|_* = \|\epsilon(\theta)\|_*$ is (nearly) optimally controlled.

In the sequel, we interchangeably use the terms *statistical rate*, *estimation rate*, *deviation rate* or simply *rate* to designate the statistical dependence of the bounds we obtain on the excess risk $\mathcal{L}(\widehat{\theta}) - \mathcal{L}(\theta^\star)$ and the parameter error $\|\widehat{\theta} - \theta^\star\|_2$ for a general estimator $\widehat{\theta}$. This may lead to some confusion since the excess risk is only comparable to the square error $\|\widehat{\theta} - \theta^\star\|_2^2$ up to a constant factor. However, the reader should be able to distinguish the two situations based on context. Note that the previous terms should not be confused with *optimization rate* and *corruption rate*.

4.3 The Smooth Case with Mirror Descent

In this section we will assume the loss ℓ is smooth, formally :

Assumption 4.3. *For any* $y \in \mathcal{Y}$, *the loss* $z \mapsto \ell(z,y)$ *is convex, differentiable and* γ*-smooth meaning that*

$$|\ell'(z,y) - \ell'(z',y)| \leqslant \gamma|z - z'| \tag{4.3.1}$$

for some $\gamma > 0$ *and all* $z, z' \in \widehat{\mathcal{Y}}$, *where the derivative is taken w.r.t. the first argument.*

The above assumption is stated in all generality for z, z' belonging to the prediction space $\widehat{\mathcal{Y}}$. For regression or binary classification tasks, we will have $\widehat{\mathcal{Y}} = \mathbb{R}$ and $\ell'(\cdot, y) \in \mathbb{R}$ so that the absolute values are enough to interpret the required inequality. Nonetheless, it can also be extended for K-way multiclass classification where $\widehat{\mathcal{Y}} = \mathbb{R}^K$ and $\ell'(\cdot, y) \in \mathbb{R}^K$, in which case the absolute values on both sides of the above inequality should be interpreted as Euclidean norms.

We also make the following quadratic growth assumption [332, Definition 4].

Assumption 4.4. *Let* $\theta^\star \in \Theta$ *be the optimum of the objective* $\mathcal{L}$ *and* $\|\cdot\|_2$ *the usual Euclidean norm. There exists a constant* $\kappa > 0$ *such that for all* $\theta \in \Theta$:

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) \geqslant \kappa\|\theta - \theta^\star\|_2^2. \tag{4.3.2}$$

[5]Including the square loss, the absolute loss, Huber's loss, the logistic loss and the Hinge loss.

Assumption 4.4 is similar to but weaker than strong convexity because it only requires the quadratic minorization to hold around the optimum $\theta^\star$ whereas a strongly convex function is minorized by a quadratic function at every point. In the linear regression setting with samples $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$, it is easy to see that the above condition holds as soon as the data follows a distribution with non-singular covariance $\Sigma = \mathbb{E}XX^\top$. A more general setting where this condition holds is also given in [223, Section 3.1]. In comparison, the commonly used restricted eigenvalue or compatibility conditions [35, 72] roughly require the empirical covariance $\widehat{\Sigma} = \frac{1}{n}\sum_{i\leqslant n} X_i X_i^\top$ to satisfy $\|\widehat{\Sigma}v\|_2 \geqslant \kappa\|v\|_2$ for all approximately s-sparse vectors $v \in \mathbb{R}^d$. This was shown to hold for covariates following some well known distributions (e.g. Gaussian with non singular covariance) with a sufficient sample count n [372]. However, this is clearly more constraining than Assumption 4.4. Some variants of the compatibility condition are formulated in terms of the population covariance Σ [158, 60] but these serve as a basis to show oracle inequalities for LASSO rather than the study of an implementable algorithm.

As a consequence of Assumptions 4.2 and 4.3, the objective gradient is L-Lipschitz continuous for some constant $L > 0$, meaning that we have :

$$\|g(\theta) - g(\theta')\|_* \leqslant L\|\theta - \theta'\| \quad \forall \theta, \theta' \in \Theta. \tag{4.3.3}$$

This property is necessary to establish the convergence of the Mirror Descent algorithm [339] proposed in this section. Since we adopt a multistage mirror descent procedure as done in [223], our framework is also similar to theirs.

Definition 4.1. *A function $\omega : \Theta \to \mathbb{R}$ is a distance generating function if it is a real convex function over Θ which satisfies :*

1. *ω is continuously differentiable and strongly convex w.r.t. the norm $\|\cdot\|$ i.e.*

$$\langle \nabla\omega(\theta) - \nabla\omega(\theta'), \theta - \theta'\rangle \geqslant \|\theta - \theta'\|^2.$$

2. *We have $\omega(\theta) \geqslant \omega(0) = 0$ for all $\theta \in \Theta$.*

3. *There exists a constant $\nu > 0$ called the quadratic growth constant such that we have :*

$$\omega(\theta) \leqslant \nu\|\theta\|^2 \quad \forall \theta \in \Theta. \tag{4.3.4}$$

We shall see, for individual applications, that one needs to choose ω in such a way that it is strongly convex and the constant ν has only a light dependence on the dimension. For a reference point $\theta_0 \in \Theta$, we define $\omega_{\theta_0}(\theta) = \omega(\theta - \theta_0)$ and the associated Bregman divergence :

$$V_{\theta_0}(\theta, \theta') = \omega_{\theta_0}(\theta) - \omega_{\theta_0}(\theta') - \langle \nabla\omega_{\theta_0}(\theta'), \theta - \theta'\rangle.$$

Given a step size $\beta > 0$ and a dual vector $u \in \Theta^*$, we define the following proximal mapping :

$$\begin{aligned}\mathrm{prox}_\beta(u, \theta; \theta_0, \Theta) &:= \arg\min_{\theta' \in \Theta}\{\langle \beta u, \theta'\rangle + V_{\theta_0}(\theta', \theta)\} \\ &= \arg\min_{\theta' \in \Theta}\{\langle \beta u - \nabla\omega_{\theta_0}(\theta), \theta'\rangle + \omega_{\theta_0}(\theta')\}.\end{aligned}$$

The previous operator yields the next iterate of Mirror Descent for previous iterate θ, gradient u and step size β with Bregman divergence defined according to the reference point θ_0. Ideally, we would plug $g(\theta)$ as gradient u but since the true gradient is not observed, we replace it with the estimator $\widehat{g}(\cdot)$. All in all, given an initial parameter θ_0 we obtain the following iteration for

Mirror Descent :

$$\theta_{t+1} = \mathrm{prox}_\beta(\widehat{g}(\theta_t), \theta_t; \theta_0, \Theta), \tag{4.3.5}$$

with a step size β to be defined later according to problem parameters. The previous proximal operator can be computed in closed form in each of the applications we consider in Section 4.5, see Section 4.8.3 for details. We state the convergence properties of the above iteration in the following proposition.

Proposition 4.1. *Grant Assumptions 4.2 and 4.3 so that the objective $\mathcal{L}$ is L-Lipschitz-smooth for some $L > 0$. Let mirror descent be run with constant step size $\beta \leqslant 1/L$ starting from $\theta_0 \in \Theta$ with $\Theta = B_{\|\cdot\|}(\theta_0, R)$ for some radius $R > 0$. Let $\theta_1, \dots, \theta_T$ denote the resulting iterates and $\widehat{\theta}_T = \sum_{t=1}^T \theta_t/T$, then the following inequality holds :*

$$\begin{aligned}\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}(\theta^\star) &\leqslant \frac{1}{T}\Big(\frac{1}{\beta} V_{\theta_0}(\theta^\star, \theta_0) + \sum_{t=0}^{T-1} \langle \epsilon_t, \theta^\star - \theta_{t+1}\rangle\Big) \\ &\leqslant \frac{\nu R^2}{\beta T} + 2\bar{\epsilon} R\end{aligned}$$

where $\epsilon_t = \widehat{g}(\theta_t) - g(\theta_t)$ and $\bar{\epsilon} = \max_{t=0\dots T-1} \|\epsilon_t\|_*$.

Proposition 4.1 is proven in Section 4.8.1 based on [223, Proposition 2.1] and quantifies the progress of mirror descent on the objective value while measuring the impact of the gradient errors. The original version in [223] considers a stochastic optimization problem in which a new sample arrives at each iteration providing an unbiased estimate of the gradient so that it is possible to obtain a bound with optimal quadratic dependence on the statistical error. Though the above result is suboptimal in this respect, we will show in the sequel that an optimal statistical rate can still be achieved using a multistage procedure.

Notice that the previous statement only provides guarantees for the average $\widehat{\theta}_T = \sum_{t=1}^T \theta_t/T$. While this is commonplace for online settings, we intuitively expect the last iterate θ_T to be the best estimate of $\theta^\star$ in our batch setting where all the data is available from the beginning.

In order to address this issue, we define a *corrected* proximal operator given an upper bound $\bar{\epsilon}$ on the statistical error :

$$\widehat{\mathrm{prox}}_\beta(u, \theta; \theta_0, \Theta) = \operatorname*{arg\,min}_{\theta' \in \Theta} \{\langle \beta u, \theta'\rangle + \beta\bar{\epsilon}\|\theta' - \theta\| + V_{\theta_0}(\theta', \theta)\}. \tag{4.3.6}$$

For this new operator, the following statement applies.

Proposition 4.2. *In the setting of Proposition 4.1, let mirror descent be similarly run with constant step size $\beta \leqslant 1/L$ starting from $\theta_0 \in \Theta$ with $\Theta = B_{\|\cdot\|}(\theta_0, R)$ for some radius $R > 0$. Let $\theta_1, \dots, \theta_T$ denote the resulting iterates obtained through $\theta_{t+1} = \widehat{\mathrm{prox}}_\beta(\widehat{g}(\theta_t), \theta_t; \theta_0, \Theta)$ then the following inequality holds :*

$$\begin{aligned}\mathcal{L}(\theta_T) - \mathcal{L}(\theta^\star) &\leqslant \frac{1}{T}\Big(\frac{1}{\beta} V_{\theta_0}(\theta^\star, \theta_0) + 2\sum_{t=0}^{T-1} \langle \epsilon_t, \theta^\star - \theta_{t+1}\rangle\Big) \\ &\leqslant \frac{\nu R^2}{\beta T} + 4\bar{\epsilon} R,\end{aligned}$$

where $\bar{\epsilon} = \max_{t=0\dots T-1} \|\epsilon_t\|_*$.

The proof of Proposition 4.2 is given in Section 4.8.1 and mainly differs from that of Proposition 4.1 in that the introduced correction allows to show a monotonous decrease of the objective

i.e. $\mathcal{L}(\theta_{t+1}) \leqslant \mathcal{L}(\theta_t)$ letting us draw the conclusion on the last iterate. Nevertheless, we suspect that the correction is not really needed for this bound to hold on θ_T and consider it rather as an artifact of our proof.

Propositions 4.1 and 4.2 only state a linear dependence of the final excess risk on the statistical error $\bar{\epsilon}$ which leads to a suboptimal statistical rate of $1/\sqrt{n}$. However, the optimal rate of $1/n$ can be achieved by leveraging the sparsity condition on $\theta^\star$ (Assumption 4.1) and the quadratic growth condition (Assumption 4.4) upon running *multiple stages* of Mirror Descent [226, 223]. The idea is that by factoring these two assumptions in, for T big enough in Proposition 4.2 and given $\bar{s} \geqslant s$, it can be shown that the closest $\bar{s}$-sparse element $\text{sparse}_{\bar{s}}(\theta_T)$ to the last iterate θ_T is such that $\Theta' = B_{\|\cdot\|}(\text{sparse}_{\bar{s}}(\theta_T), R') \ni \theta^\star$ with $R' < R$. Therefore, $\text{sparse}_{\bar{s}}(\theta_T)$ can serve as the starting point of a new stage of mirror descent on the smaller domain Θ'. By repeating this trick multiple times, we obtain the following multistage mirror descent algorithm.

Algorithm : Approximate Multistage Mirror Descent (AMMD)

- *Initialization*: Initial parameter $\theta^{(0)}$ and $R > 0$ such that $\theta^\star \in \Theta := B_{\|\cdot\|}(\theta_0, R)$.
  Number of stages $K > 0$. Step size $\beta \leqslant 1/L$. Quadratic minorization constant κ.
  High probability upperbound $\bar{\epsilon}$ on the error $\|\widehat{g}(\theta) - g(\theta)\|_*$.
  Upperbound $\bar{s}$ on the sparsity s.
- Set $R_0 = R$.
- Loop over stages $k = 1 \dots K$:
  - Set $\theta_0^{(k)} = \theta^{(k-1)}$ and $\Theta_k = B_{\|\cdot\|}(\theta_0^{(k)}, R_{k-1})$.
  - Run iteration
    $$\theta_{t+1}^{(k)} = \widehat{\text{prox}}_\beta\big(\widehat{g}(\theta_t^{(k)}), \theta_t^{(k)}; \theta_0^{(k)}, \Theta_k\big),$$
    for T_k steps with $T_k = \left\lceil \frac{\nu R_{k-1}}{\beta \bar{\epsilon}} \right\rceil$.
  - Set $\theta^{(k)} = \text{sparse}_{\bar{s}}(\widetilde{\theta}^{(k)})$ where $\widetilde{\theta}^{(k)} = \theta_{T_k}^{(k)}$.
  - Set $R_k = \frac{1}{2}\big(R_{k-1} + \frac{40\bar{s}\bar{\epsilon}}{\kappa}\big)$.
- *Output*: The final stage estimate $\theta^{(K)}$.

The AMMD algorithm borrows ideas from [226, 227, 223] aiming to achieve linear convergence using mirror descent. The main trick lies in the fact that performing multiple stages of mirror descent allows to repeatedly restrict the parameter space into a ball of radius R_k which shrinks geometrically with each stage. In this work, we find that the radius R_k evolves following a special contraction as a result of the statistical error being factored in. Note that, although a few instructions of AMMD are stated in terms of unknown quantities, the procedure may be simplified to get around this difficulty with satisfactory results, see Section 4.6 for details. We show that the above procedure allows to improve the result of Proposition 4.2 to achieve a fast statistical rate. The following statement expresses the theoretical properties of AMMD.

Theorem 4.1. *Grant Assumptions 4.1, 4.2, 4.3 and 4.4. Let $L > 0$ denote the Lipschitz smoothness constant for the objective $\mathcal{L}$. Assume approximate Mirror Descent is run with step size $\beta \leqslant 1/L$ starting from $\theta_0 \in \Theta$ such that $\theta^\star \in B_{\|\cdot\|}(\theta_0, R)$ for some $R > 0$ and using a gradient estimator $\widehat{g}$ with error upperbound $\bar{\epsilon}$ as in Proposition 4.3.5, then after K stages we have the inequalities :*

$$\big\|\theta^{(K)} - \theta^\star\big\| \leqslant \sqrt{2\bar{s}}\big\|\theta^{(K)} - \theta^\star\big\|_2 \leqslant 2\sqrt{2\bar{s}}\big\|\widetilde{\theta}^{(K)} - \theta^\star\big\|_2 \leqslant 2^{-(K-1)/2}R + \frac{40\bar{s}\bar{\epsilon}}{\kappa},$$

$$\mathcal{L}(\widetilde{\theta}^{(K)}) - \mathcal{L}(\theta^\star) \leqslant 10\bar{\epsilon}\Big(2^{-K}R + \frac{40\bar{s}\bar{\epsilon}}{\kappa}\Big).$$

Moreover, the corresponding number of necessary iterations is bounded by :

$$T = \sum_{k=1}^{K} T_k \leqslant \frac{2R\nu}{\beta\bar{\epsilon}} + K\Big(1 + \frac{40\nu\bar{s}}{\kappa\beta}\Big).$$

Theorem 4.1 is proven in Section 4.8.1 and may be compared to [223, Theorem 2.1]. Both statements bound the risk in terms of the objective and parameter error by the sum of an exponentially vanishing optimization error and a statistical error term. Note that the exponential optimization rate in the number of stages K also holds in the number of iterations since successive stages contain a geometrically decreasing number of them. Theorem 4.1 expresses the statistical error in terms of a bound $\bar{\epsilon}$ and will thus lead to a high confidence statement when combined with a bound on $\bar{\epsilon}$ (see Section 4.5). In contrast, Theorem 2.1 of [223] is a result in expectation which is later used to derive a high confidence bound on an aggregated estimate.

One can see that Theorem 4.1 exhibits a dependence in $\bar{\epsilon}^2$ of the excess risk upperbound so that the suboptimal statistical rate in Propositions 4.1 and 4.2 is improved into a fast rate as announced. This is accomplished by shrinking the size of the considered parameter set through the stages until it reaches the scale of the statistical error, yielding an optimal rate. This shrinkage is achieved thanks to the choice of stage-length $T_k = \Omega(R_{k-1})$ in AMMD leading to a bound in terms of R rather than R^2 in Proposition 4.2. Combined with Assumption 4.4, this implies that a square root function is applied to R_k after each stage, see the proof for further details. We can now turn to the case of a non smooth loss function ℓ.

4.4 The Non Smooth Case with Dual Averaging

In the previous section, we saw how sparse estimation can be performed using the Mirror Descent algorithm to optimize a smooth objective with a non Euclidean metric on the parameter space. The smoothness property is necessary for these results to hold so that many loss functions not satisfying it are left uncovered. Therefore, we propose to use another algorithm for non smooth objectives. The alternative is the Dual Averaging algorithm [343] which was already used for non smooth sparse estimation in [3] for instance. Since the original algorithm requires to average the iterates to obtain a parameter with provable convergence properties, we instead use a variant [340] for which such properties apply for individual iterates.

The smoothness condition in Assumption 4.3 is no longer required but we still need to replace it with a Lipschitz property :

Assumption 4.5. *There exists a positive constant $M > 0$ such that the objective $\mathcal{L}$ is M-Lipschitz w.r.t. the norm $\|\cdot\|$ i.e. for all $\theta, \theta' \in \Theta$ it holds that* :

$$\mathcal{L}(\theta) - \mathcal{L}(\theta') \leqslant M\|\theta - \theta'\|.$$

We also replace Assumption 4.4 by the following weaker assumption :

Assumption 4.6. *There exist positive constants $\kappa, \lambda > 0$ such that the following inequality holds* :

$$\mathcal{L}(\theta) - \mathcal{L}(\theta^\star) \geqslant \frac{\kappa\|\theta - \theta^\star\|_2^2}{\lambda + \|\theta - \theta^\star\|_2}.$$

We introduce this (to our knowledge) previously unknown assumption in the literature which we call the *pseudo-linear* growth assumption in order to better suit the setting of this section.

Indeed, few non-smooth loss functions, if any, result in quadratically growing objectives as Assumption 4.4 requires. Note that the lower bound of Assumption 4.6 is linear away from the optimum i.e. for big $\|\theta - \theta^\star\|_2$ and behaves quadratically around it. This assumption is also weaker than a linear lower bound proportional to $\|\theta - \theta^\star\|_2$ because of its quadratic behaviour around the optimum. We will show that, for κ big enough, this minorization suffices to obtain linear convergence to a solution with fast statistical rate.

Analogously to Mirror Descent's distance generating function ω, we let $\omega : \Theta \to \mathbb{R}^+$ be the *prox-function*. We choose to denote it similarly since it plays an analogous role for Dual Averaging and has the same properties as those listed in Definition 4.1.

Let $(a_t)_{t\geqslant 0}$ be a sequence of step sizes and $(\gamma_t)_{t\geqslant 0}$ a non decreasing sequence of positive scaling coefficients. The DA procedure is defined, given an initial $\theta_0 \in \Theta$, by the following scheme :

$$s_t = \frac{1}{A_t}\sum_{i=0}^{t} a_i \widehat{g}_i \quad \text{with} \quad \widehat{g}_i = \widehat{g}(\theta_i) \text{ and } g_i = g(\theta_i) \quad \forall i = 0, \dots, T.$$

$$A_t = \sum_{i=0}^{t} a_i \quad \text{and} \quad \theta_t^+ = \arg\min_{\theta\in\Theta} A_t\langle s_t, \theta\rangle + \gamma_t \omega(\theta).$$

$$\theta_{t+1} = (1-\tau_t)\theta_t + \tau_t \theta_t^+ \qquad \text{where} \qquad \tau_t = \frac{a_{t+1}}{A_{t+1}}.$$

Proposition 4.3. *Grant Assumption 4.5, let Dual Averaging be run following the above scheme, let $R > 0$ such that $\Theta \subseteq B_{\|\cdot\|}(\theta_0, R)$ and denote $\bar{\epsilon} = \max_i \|\epsilon_i\|_*$, we have the following inequality :*

$$A_t(\mathcal{L}(\theta_t) - \mathcal{L}(\theta^\star)) + \frac{\gamma_t}{2}\|\theta_t^+ - \theta^\star\|^2 \leqslant \gamma_t \omega(\theta^\star) + \sum_{i=0}^{t} \frac{a_i^2}{2\gamma_{i-1}}\|g_i\|_*^2 + 4A_t R\bar{\epsilon}.$$

In particular, by choosing $a_i = 1$ and $\gamma_i = \sqrt{i+1}$ for all i we get :

$$\mathcal{L}(\theta_t) - \mathcal{L}(\theta^\star) \leqslant \frac{1}{\sqrt{t}}\big(\omega(\theta^\star) + M^2\big) + 4R\bar{\epsilon}.$$

The proof of Proposition 4.3 is given in Section 4.8.2 and is inspired from [340, Theorem 3.1]. In this result, we manage to obtain a statement in terms of the individual iterates θ_t thanks to the running average performed in the above scheme whereas the initial study of dual averaging focused on the *average* of the iterates [343]. Notice that, due to Assumption 4.3 being dropped, the convergence speed degrades to $1/\sqrt{t}$ as opposed to $1/t$ previously. This convergence speed is the fastest possible and cannot be improved with a different choice of a_i and γ_i. Most importantly, Proposition 4.3 quantifies the impact of the errors on the gradients on the quality of the optimisation result and shows that it remains controlled in this case too.

As in the previous section, the statistical rate we initially obtain is suboptimal and a multistage procedure is needed to improve it. The idea is the same as in Section 4.3 and consists in sparsifying the final iterate in Proposition 4.3 and using it as the initial point of a new optimization stage which takes place on a narrower domain. We make the resulting algorithm explicit below.

Algorithm : Approximate Multistage Dual Averaging (AMDA)

- *Initialization*: Initial parameter θ_0 and $R > 0$ such that $\theta^\star \in \Theta := B_{\|\cdot\|}(\theta_0, R)$.

  Pseudolinear minorization constants κ, λ.

  High probability upperbound $\bar{\epsilon}$ on the error $\|\widehat{g}(\theta) - g(\theta)\|_*$.

Upperbound $\bar{s}$ on the sparsity s.

- Set $R_0 = R$ and $\tau = \frac{10\sqrt{8\bar{s}\bar{\epsilon}}}{\kappa}$ and $R^\star = \frac{80\lambda\bar{s}\bar{\epsilon}}{\kappa}$.
- Set $k = 0$ and the per stage number of iterations $T' = \left\lceil \left(\frac{\nu+M^2}{\bar{\epsilon}}\right)^2 \right\rceil$
- For $k = 1, \dots, K$:
  - Set $\theta_0^{(k)} = \theta^{(k-1)}$ and $\Theta_k = B_{\|\cdot\|}(\theta_0^{(k)}, R_{k-1})$.
  - Run Dual averaging with prox-function $\omega_{\theta^{(k-1)}}$ and steps $a_i = R_{k-1}$ for T' iterations.
  - Set $\theta^{(k)} = \mathrm{sparse}_{\bar{s}}(\widetilde{\theta}^{(k)})$ where $\widetilde{\theta}^{(k)} := \theta_{T'}^{(k)}$.
  - Set $R_k = \max\left(\tau R_{k-1}, \frac{1}{2}(R_{k-1} + R^\star)\right)$.
- *Output*: The final stage estimate $\theta^{(K)}$.

Similar to AMMD, the AMDA algorithm runs multiple optimisation stages through which the parameter space is repeatedly restricted allowing to obtain similar benefits regarding convergence speed and statistical performance. However, it is worth noting that these improvements are obtained under much milder conditions here since the objective may not even be smooth and is only required to satisfy the pseudo-linear growth condition of Assumption 4.6 whereas smoothness and quadratic minorization or strong convexity were indispensable in previous works [226, 227, 223].

As previously mentioned for AMMD, a simplified version of AMDA can be implemented which does not require knowledge of all the involved quantities, see Section 4.6 for details. We state the convergence guarantees for the above algorithm in the following Theorem.

Theorem 4.2. *Grant Assumptions 4.1, 4.2, 4.5, 4.6 and assume that* $\tau = \frac{10\sqrt{8\bar{s}\bar{\epsilon}}}{\kappa} < 1$. *At the end of each stage* $k \geqslant 1$, *we have* :

$$\left\|\theta^{(k)} - \theta^\star\right\| \leqslant \sqrt{2\bar{s}}\left\|\theta^{(k)} - \theta^\star\right\|_2 \leqslant 2\sqrt{2\bar{s}}\left\|\widetilde{\theta}^{(k)} - \theta^\star\right\|_2 \leqslant R_k, \tag{4.4.1}$$

$$\mathcal{L}(\widetilde{\theta}^{(k)}) - \mathcal{L}(\theta^\star) \leqslant 5\bar{\epsilon}R_{k-1}. \tag{4.4.2}$$

Moreover, the total number of necessary iterations before $R_k \leqslant 2R^\star = \frac{160\lambda\bar{s}\bar{\epsilon}}{\kappa}$ *is at most*

$$\log(R_0/R^\star)\Big(\frac{1}{\log(1/\tau)} + \frac{1}{\log(2)}\Big)\Big(\Big(\frac{\nu+M^2}{\bar{\epsilon}}\Big)^2 + 1\Big).$$

The proof of Theorem 4.2 is given in Section 4.8.2 and shows that the optimization stages go through two phases: an initial *linear* phase corresponding to the linear regime of the lower bound given by Assumption 4.6 and a later *quadratic* phase during which the quadratic regime takes over. The success of the linear phase relies on the condition $\tau < 1$ which can be rewritten as $\bar{\epsilon} \leqslant O(\bar{s}\kappa)$ where the factor $\bar{s}$ is a byproduct of measuring the parameter error with the norm $\|\cdot\|$ while Assumption 4.6 is stated with the Euclidean one. In the linear regime, κ acts as a lower bound for the gradient norm so that the condition ensures that the error is smaller than the actual gradient allowing the optimisation to make progress. Theorem 4.2 states that convergence to the optimum occurs at geometrical speed through the stages of AMDA despite the absence of strong convexity. This is achieved thanks to the choice of step $a_i = R_{k-1}$ in AMDA which leads to a bound in terms of R rather than R^2 emerging from the term $\omega(\theta^\star)$ in Proposition 4.3, see the proof for more details.

Remark. In recent work, [225] created a common framework for the study of the Mirror Descent and Dual Averaging algorithms which they recover as special cases of a generic Unified Mirror Descent procedure. However, the distinction between the two remains necessary since they address smooth and non-smooth objectives respectively and, in each case, the attainable within-stage convergence speed differs from $1/t$ to $1/\sqrt{t}$ as seen in Propositions 4.2 and 4.3. This is reflected in Theorems 4.1 and 4.2 which display a dependence of the necessary number of iterations in $1/\bar{\epsilon}$ in the gradient error for Mirror Descent as opposed to $1/\bar{\epsilon}^2$ for Dual Averaging.

4.5 Applications

We now consider a few problems which may be solved using the previous optimization procedures. As said earlier, we have omitted to quantify the gradient errors $\|\epsilon\|_*$ until now. This is because the definition of the dual norm $\|\cdot\|_*$ is problem dependent. In the next subsections, we consider a few instances and propose adapted gradient estimators for them. In each case, the existence of a second moment for the gradient random variable $G(\theta) := \ell'(\langle\theta, X\rangle, Y)X$ is required. This follows from the next Lemma proven in Section 4.8.3 based on Assumption 4.2.

Lemma 4.1. *Under Assumption 4.2 the objective $\mathcal{L}(\theta)$ is well defined for all $\theta \in \Theta$ and we have*

$$\mathbb{E}\big[\big\|G(\theta)\big\|_*^2\big] = \mathbb{E}\big[\big\|\ell'(X^\top\theta, Y)X\big\|_*^2\big] < +\infty.$$

In what follows, we will assume that, at each step of the optimization algorithm, the estimation of the gradient is performed with a new batch of data. For example, if the available data set contains n samples then it needs to be divided into T disjoint splits in order to make T optimization steps. This is necessary in order to guarantee that the gradient samples used for estimation at each step t are independent from θ_t, the (random) current parameter which depends on the data used before.

This trick was previously used for example in [364] for the same reasons. A possible alternative is to use an ϵ-net argument or Rademacher complexity in order to obtain uniform deviation bounds on gradient estimation over a compact parameter set Θ. However, this entails extra dependence on the dimension in the resulting deviation bound which we cannot afford in the high-dimensional setting. For these reasons, we prefer to use data splitting in this work and regard it more like a proof artifact rather than a true practical constraint. Note that we do not implement it later in our experimental section.

4.5.1 Vanilla sparse estimation

In this section, we consider the problem of optimizing an objective $\mathcal{L}(\theta) = \mathbb{E}[\ell(\langle\theta, X\rangle, Y)]$ where the covariate space $\mathcal{X}$ is simply $\mathbb{R}^d$ and the labels are either real numbers $\mathcal{Y} = \mathbb{R}$ (regression) or binary labels (binary classification). In this case, the parameter space is a subset $\Theta \subset \mathbb{R}^d$, the sparsity of a parameter $\theta \in \Theta$ is measured as its number of nonzero entries $S(\theta) = \sum_{j\in[\![d]\!]} \mathbf{1}_{\theta_j\neq 0}$, and $\|\cdot\|$ is defined as the ℓ_1 norm $\|\cdot\| = \|\cdot\|_1$ so that $\|\cdot\|_* = \|\cdot\|_\infty$. We define the distance generating function ω as :

$$\omega(\theta) = \frac{1}{2}e\log(d)d^{(p-1)(2-p)/p}\|\theta\|_p^2 \quad \text{with} \quad p = 1 + \frac{1}{\log(d)}.$$

One can check that the above definition satisfies the requirements of Definition 4.1. In particular, it is strongly convex w.r.t. $\|\cdot\|$ and quadratically growing with constant $\nu = \frac{1}{2}e^2\log(d)$ (see [344, Theorem 2.1]). In others words, conditions 1 and 3 of Definition 4.1 are reconciled with $\nu = O(\log d)$. For the sake of achieving this compromise, the previous choice of distance generating

function is common in the high-dimensional learning literature using mirror descent [3, 138, 401, 339, 226, 344] up to slight variations in p and the multiplying factor.

We consider Assumption 4.2 on the data with a constant fraction of outliers $|\mathcal{O}| \leqslant \eta n$ for some $\eta < 1/2$ (η-corruption) so that the gradient samples $g^i(\theta) := \ell'(\theta^\top X_i, Y_i)X_i$ may be both heavy-tailed and corrupted as well. We propose to compute $\widehat{g}(\theta)$ as the coordinatewise trimmed mean of the sample gradients i.e.

$$\widehat{g}_j(\theta) = \mathtt{TM}_\alpha\big(g_j^1(\theta), \dots, g_j^n(\theta)\big), \tag{4.5.1}$$

where, assuming without loss of generality that n is even, the trimmed mean estimator with parameter α for a sample $x_1, \dots, x_n \in \mathbb{R}$ is defined as follows

$$\mathtt{TM}_\alpha(x_1, \dots, x_n) = \frac{2}{n} \sum_{i=n/2+1}^{n} q_\alpha \vee x_i \wedge q_{1-\alpha},$$

where we denoted $a \wedge b := \min(a, b)$ and $a \vee b := \max(a, b)$ and used the quantiles $q_\alpha := x^{([\alpha n/2])}$ and $q_{1-\alpha} = x^{([(1-\alpha)n/2])}$ with $x^{(1)} \leqslant \cdots \leqslant x^{(n/2)}$ the order statistics of $(x_i)_{i \in [\![n/2]\!]}$ and where $[\cdot]$ denotes the integer part.

The main hurdle to compute the trimmed mean estimator is to find the two previous quantiles. A naive approach for this task would be to sort all the values leading to an $O(n \log(n))$ complexity. However, this can be brought down to $O(n)$ using the median-of-medians algorithm (see for instance [98, Chapter 9]) so that the whole procedure runs in linear time.

We now give the deviation bound satisfied by the estimator (4.5.1). We denote x^j as the j-th coordinate of a vector x.

Lemma 4.2. *Grant Assumption 4.2 with a fraction of outliers $|\mathcal{O}| \leqslant \eta n$ with $\eta < 1/8$. Fix $\theta \in \Theta$, let $\sigma_j^2 = \mathrm{Var}(\ell'(\theta^\top X, Y)X^j)$ for $j \in [\![d]\!]$ be the gradient coordinate variances and let $1 > \delta > e^{-n/2}/4$ be a failure probability and consider the coordinatewise trimmed mean estimator* (4.5.1) *with parameter $\alpha = 8\eta + 12\frac{\log(4/\delta)}{n}$. Denoting $\sigma_{\max}^2 = \max_j \sigma_j^2$, we have with probability at least $1 - \delta$:*

$$\big\|\widehat{g}(\theta) - g(\theta)\big\|_\infty \leqslant 7\sigma_{\max}\sqrt{4\eta + 6\frac{\log(4/\delta) + \log(d)}{n}}. \tag{4.5.2}$$

Proof. This is an almost immediate application of Lemma 3.9 (see also [288, Theorem 1]). By an immediate application of the latter, we obtain for each $j \in [\![d]\!]$ that with probability at least $1 - \delta/d$ we have :

$$\big|\widehat{g}_j(\theta) - g_j(\theta)\big| \leqslant 7\sigma_j\sqrt{4\eta + 6\frac{\log(4/\delta) + \log(d)}{n}}. \tag{4.5.3}$$

Hence, the lemma follows by a simple union bound argument. □

For the sake of simplicity, this deviation bound is only stated for a square integrable gradient which yields a $\sqrt{\eta}$ dependence in the corruption rate. More generally, for a random variable admitting a finite moment of order k, one can derive a bound in terms of $\eta^{1-1/k}$ which reflects a milder dependence for greater k, see [153, Lemma 9] for the bound in question.

In a way, the fact that the gradient error is measured with the infinity norm in this setting is a "stroke of luck" since the optimal dependence in the dimension for the statistical error becomes achievable using only a univariate estimator. This is in contrast with situations where multivariate robust estimators need to be used for which the combination of efficiency, sub-Gaussianity and robustness to η-corruption is hard to come by.

Based on Lemma 4.2 we obtain a gradient error of order $\bar{\epsilon} = O(\sqrt{\log(d)/n})$. Plugging this deviation rate into Theorem 4.1 yields the optimal $s\log(d)/n$ rate for vanilla sparse estimation. The same applies for Theorem 4.2 provided the condition $\tau < 1$ holds.

Corollary 4.1. *In the context of Theorem 4.1 and Lemma 4.2, let the AMMD algorithm be run starting from $\theta_0 \in \Theta = B_{\|\cdot\|}(\theta_0, R)$ using the coordinatewise trimmed mean estimator with sample splitting i.e. at each iteration a different batch of size $\widetilde{n} = n/T$ is used for gradient estimation with confidence $\widetilde{\delta} = \delta/T$ where T is the total number of iterations. Let K be the number of stages and $\widehat{\theta}$ the obtained estimator. Denote $\sigma_{\max}^2 = \sup_{\theta\in\Theta}\max_{j\in[\![d]\!]}\mathrm{Var}(\ell'(\theta^\top X, Y)X^j)$, with probability at least $1-\delta$, the latter satisfies :*

$$\left\|\widehat{\theta} - \theta^\star\right\|_2 \leqslant \frac{2^{-K/2}R}{\sqrt{\bar{s}}} + \frac{140\sqrt{2\bar{s}}\sigma_{\max}}{\kappa}\sqrt{4\eta + 6\frac{\log(4/\widetilde{\delta}) + \log(d)}{\widetilde{n}}}.$$

Proof. The result is easily obtained by combining Theorem 4.1 and Lemma 4.2 with a union bound argument over all iterations T in order to bound $\bar{\epsilon} = \max_{t=0,\dots,T-1}\|\epsilon_t\|_*$ as defined in Proposition 4.2. □

In the above upper bound, the optimisation error vanishes exponentially with the number of stages K so that the final error can be attributed in large part to the second statistical error term. The latter achieves the nearly optimal $\sqrt{s\log(d)/n}$ rate and combines robustness to heavy tails and η-corruption. Moreover, this statement holds for a *generic* loss function satisfying the assumptions of Section 4.3. These can be further weakened to those given in Section 4.4 by using Dual Averaging while preserving the same statistical rate. The statement of a result in this weaker setting is postponed to Section 4.8.3 in order to avoid excessive repetition. To our knowledge, this is the first result with such properties for vanilla sparse estimation whereas previous results from the literature either focused on specific learning problems with a fixed loss function [101, 390] or isolated the issues of robustness by assuming the data to be either heavy tailed or Gaussian and corrupted [19, 273, 274, 223]. Furthermore, we stress that this error bound is achieved at a comparable computational cost to that of a standard non robust algorithm since, as mentioned earlier, the robust trimmed mean estimator can be computed in linear time.

A possible room for improvement is to try to remove the $\bar{s}$ factor multiplying the corruption rate η. The only works we are aware of achieving this are [274, 390] but both involve costly data-filtering steps. We suspect this may be an inevitable price to pay for such an improvement.

4.5.2 Group sparse estimation

In the group sparse case, we again consider the covariate space $\mathcal{X} = \mathbb{R}^d$ where the coordinates $[\![d]\!]$ are arranged into groups $G_1, \dots, G_{N_G}$ which form a partition of the coordinates $[\![d]\!]$ and sparsity is measured in terms of these groups i.e. $S(\theta) = \sum_{j\in[\![N_G]\!]}\mathbf{1}_{\theta_{G_j}\neq 0}$ and we assume the optimal $\theta^\star = \arg\min_\theta \mathcal{L}(\theta)$ satisfies $S(\theta^\star) \leqslant s$. The norm $\|\cdot\|$ is set to be the ℓ_1/ℓ_2 norm : $\|\theta\|_{1,2} = \sum_{j\in[\![N_G]\!]}\|\theta_{G_j}\|_2$ and the dual norm is the analogous ℓ_∞/ℓ_2 norm $\|\theta\|_{\infty,2} = \max_{j\in[\![N_G]\!]}\|\theta_{G_j}\|_2$. The label set $\mathcal{Y}$ may be equal to $\mathbb{R}$ (regression) or a finite set (binary or multiclass classification).

For simplicity, we assume that the groups are of equal size m so that $d = mN_G$. This is for example the case when trying to solve a d-dimensional linear multiclass classification with K classes by estimating a parameter $\theta \in \mathbb{R}^{d\times K}$ and predicting $\arg\max_j(\theta^\top X)_j$ for a datapoint $X \in \mathbb{R}^d$. In this case, it makes sense to consider the rows $(\theta_{i,:})_{i\in[\![d]\!]}$ as groups which are collectively determined to be zero or not depending on the importance of feature i. For simplicity, we restrict ourselves to this setting until the end of this section with no loss of generality.

Analogously to the vanilla sparse case, following [344], the distance generating function (or prox-function) ω may be chosen in this case as:

$$\omega(\theta) = \frac{1}{2} e \log(d) d^{(p-1)(2-p)/p} \Big(\sum_{i=1}^{d} \|\theta_{i,:}\|_2^p \Big)^{2/p} \quad \text{with} \quad p = 1 + \frac{1}{\log(d)}.$$

We assume the data corresponds to Assumption 4.2 with η-corruption (i.e. $|\mathcal{O}| \leqslant \eta n$). Analogously to the vanilla case, we propose to estimate the gradient *groupwise* i.e. one group of coordinates at a time. For this task, a multivariate, sub-Gaussian and corruption-resilient estimation algorithm is needed. We suggest to use the estimator advocated in [117] for this purpose which remarkably combines these qualities. We refer to it as the DKP estimator and restate its deviation bound here for the sake of completeness.

Proposition 4.4 ([117, Proposition 1.5])**.** *Let T be an η-corrupted set of n samples from a distribution P in $\mathbb{R}^d$ with mean μ and covariance Σ. Let $\eta' = \Theta(\log(1/\delta)/n + \eta) \leqslant c$ be given, for a constant $c > 0$. Then any stability-based algorithm on input T and η', efficiently computes $\widehat{\mu}$ such that with probability at least $1 - \delta$, we have :*

$$\big\|\widehat{\mu} - \mu\big\|_2 = O\bigg(\sqrt{\frac{\mathrm{Tr}(\Sigma) \log \mathrm{r}(\Sigma)}{n}} + \sqrt{\|\Sigma\|_{\mathrm{op}} \eta} + \sqrt{\frac{\|\Sigma\|_{\mathrm{op}} \log(1/\delta)}{n}} \bigg), \tag{4.5.4}$$

where $\mathrm{r}(\Sigma) = \mathrm{Tr}(\Sigma)/\|\Sigma\|_{\mathrm{op}}$ is the stable rank of Σ.

The above bound is almost optimal up to the $\sqrt{\log \mathrm{r}(\Sigma)}$ factor which is at most $\sqrt{\log(d)}$. Note that we are also aware that [117, Proposition 1.6] states that, by adding a Median-Of-Means preprocessing step, stability based algorithms can achieve the optimal deviation. Nevertheless, the number k of block means required needs to be such that $k \geqslant 100\eta n$ so that the corruption rate η is strongly restricted because necessarily $n \geqslant k$. Therefore, we prefer to stick with the result above.

An algorithm with the statistical performance stated in Proposition 4.4 is given, for instance, in [117, Appendix A.2], we provide it in Section 4.8.3 for the sake of completeness. Now, we can reuse the previous section's trick by estimating the gradients *blockwise* this time to obtain the following lemma:

Lemma 4.3. *Grant Assumption 4.2 with a fraction of outliers $|\mathcal{O}| \leqslant \eta n$. Fix $\theta \in \Theta$ and denote $G_1(\theta), \dots, G_n(\theta)$ the gradient samples distributed according to $G(\theta) \in \mathbb{R}^{d \times K}$ (except for the outliers). Let $\Sigma_j = \mathrm{Var}(G(\theta)_{j,:}) \in \mathbb{R}^{K \times K}$ be the gradient block variances. Consider the groupwise estimator $\widehat{g}(\theta)$ defined such that $\widehat{g}(\theta)_{j,:}$ is the DKP estimator applied to $G(\theta)_{j,:}$. Then we have with probability at least $1 - \delta$:*

$$\big\|\widehat{g}(\theta) - g(\theta)\big\|_{\infty,2} \leqslant O\bigg(\max_j \sqrt{\frac{\mathrm{Tr}(\Sigma_j) \log \mathrm{r}(\Sigma_j)}{n}} + \sqrt{\|\Sigma_j\|_{\mathrm{op}}} \bigg(\sqrt{\eta} + \sqrt{\frac{\log(1/\delta) + \log(d)}{n}} \bigg) \bigg) \tag{4.5.5}$$

Proof. Inequality (4.5.5) is straightforward to obtain using Proposition 4.4 and a union bound argument on $j \in [\![d]\!]$. □

One can easily see that, in the absence of corruption, the above deviation bound scales as $\widetilde{O}\Big(\sqrt{\frac{K}{n}} + \sqrt{\frac{\log(d)}{n}} \Big)$. Combined with the sparsity assumption given above, plugging this estimation, which applies for the gradient error $\|\epsilon_t\|_*$, into Theorems 4.1 or 4.2 yields near optimal (up to a logarithmic factor) estimation rates for the group-sparse estimation problem [337, 280]. The following corollary formalizes this statement.

Corollary 4.2. *In the context of Theorem 4.1 and Lemma 4.3, let the AMMD algorithm be run starting from $\theta_0 \in \Theta = B_{\|\cdot\|}(\theta_0, R)$ and using the blockwise DKP estimator with sample splitting i.e. at each iteration a different batch of size $\widetilde{n} = n/T$ is used for gradient estimation with confidence $\widetilde{\delta} = \delta/T$ where T is the total number of iterations. Let N be the number of stages and $\widehat{\theta}$ the obtained estimator. With probability at least $1-\delta$, the latter satisfies :*

$$\begin{aligned}\left\|\widehat{\theta} - \theta^\star\right\|_2 &\leqslant \frac{2^{-N/2}R}{\sqrt{\bar{s}}} + \frac{\sqrt{\bar{s}}}{\kappa} O\Bigg(\sup_{\theta\in\Theta} \max_{j\in[\![d]\!]} \sqrt{\frac{\mathrm{Tr}(\Sigma_{\theta,j}) \log \mathrm{r}(\Sigma_{\theta,j})}{\widetilde{n}}} + \\ &\qquad \sqrt{\|\Sigma_{\theta,j}\|_{\mathrm{op}}} \Bigg(\sqrt{\eta} + \sqrt{\frac{\log(1/\widetilde{\delta}) + \log(d)}{\widetilde{n}}} \Bigg) \Bigg) \\ &\leqslant \frac{2^{-N/2}R}{\sqrt{\bar{s}}} + \frac{\sqrt{\bar{s}}}{\kappa} \widetilde{O}\Bigg(\sqrt{\frac{K}{\widetilde{n}}} + \Bigg(\sqrt{\eta} + \sqrt{\frac{\log(1/\widetilde{\delta}) + \log(d)}{\widetilde{n}}} \Bigg) \Bigg),\end{aligned}$$

where $\Sigma_{\theta,j} = \mathrm{Var}(G(\theta)_{j,:})$.

As before, the stated bound reflects a linearly converging optimisation and displays a statistical rate nearly matching the optimal rate for group-sparse estimation [337, 280] up to logarithmic factors. In addition, robustness to heavy tails and η-corruption likely makes this result the first of its kind for group-sparse estimation since all robust works we are aware of focus on vanilla sparsity.

4.5.3 Low-rank matrix recovery

We also consider the variant of the problem where the covariates belong to a matrix space $\mathcal{X} = \mathbb{R}^{p\times q}$ in which case the objective $\mathcal{L}(\theta) = \mathbb{E}[\ell(\langle\theta, X\rangle, Y)]$ needs to be optimized over $\Theta \subset \mathbb{R}^{p\times q}$. In this setting, $\langle\cdot,\cdot\rangle$ refers to the Frobenius scalar product between matrices

$$\langle a, b\rangle = \mathrm{Tr}(a^\top b).$$

Without loss of generality, we assume that $p \geqslant q$ and sparsity is meant as the number of non zero singular values i.e. for a matrix $A \in \mathbb{R}^{p\times q}$, denoting $\sigma(A) = (\varsigma_j(A))_{j\in[\![q]\!]}$ the set of its singular values we define $S(A) = \sum_{j\in[\![q]\!]} \mathbf{1}_{\varsigma_j(A)\neq 0}$. We set $\|\cdot\|$ to be the nuclear norm $\|A\| = \|\sigma(A)\|_1$ and the associated dual norm is the operator norm $\|\cdot\|_* = \|\cdot\|_{\mathrm{op}}$.

On the optimization side, an appropriate distance generating function (resp. prox-function) needs to be defined for this setting before Mirror Descent (resp. Dual Averaging) can be run. Based on previous literature (see [344, Theorem 2.3] and [223]), we know that the following choice satisfies the requirements of Definition 4.1 :

$$\omega(\theta) = 2e\log(2q)\Big(\sum_{j=1}^{q} \varsigma_j(\theta)^{1+r}\Big)^{2/(1+r)} \quad \text{with} \quad r = 1/(12\log(2q)).$$

This yields a corresponding quadratic growth parameter $\nu = O(\log(q))$. In order to fully define our optimization algorithm for this problem, it remains to specify a robust estimator for the gradient. This turns out to be a challenging question since the estimated value is matricial and the operator norm $\|\cdot\|_* = \|\cdot\|_{\mathrm{op}}$ emerging in this case is a fairly exotic choice to measure statistical error.

In order to achieve a nearly optimal statistical rate we define a new estimator called "CM-MOM" (short for Catoni Minsker Median-Of-Means) which combines methods from [317] for sub-Gaussian matrix mean estimation and ideas from [316, 197] in order to apply a Median-Of-

Means approach for multivariate estimation granting robustness to outliers provided these are limited in number. We now define this estimator in detail. Let ψ be a function defined as

$$\psi(x) = \log(1 + |x| + x^2/2)$$

We consider a restricted version of Assumption 4.2 in which the number of outliers is limited as[6] $|\mathcal{O}| \leqslant K/12$ where K is an integer such that $K < n$. Provided a sample of matrices $A_1, \dots, A_n \in \mathbb{R}^{p\times q}$ and a scale parameter $\chi > 0$, the CM-MOM estimator proceeds as follows :

- Split the sample into K disjoint blocks $B_1, \dots, B_K$ of equal size $m = n/K$.
- Compute the dilated block means $\xi^{(j)}$ for $j = 1, \dots, K$ as

$$\xi^{(j)} = \frac{1}{\chi m} \sum_{i\in B_j} \psi(\chi \widetilde{A}_i) \in \mathbb{R}^{(p+q)\times(p+q)},$$

  where the dilation $\widetilde{A}$ of matrix $A \in \mathbb{R}^{p\times q}$ is defined as $\widetilde{A} = \begin{pmatrix} 0 & A \\ A^\top & 0 \end{pmatrix} \in \mathbb{R}^{(p+q)\times(p+q)}$ which is symmetric and the function ψ is applied to a symmetric matrix $S \in \mathbb{R}^{d\times d}$ by applying it to its eigenvalues i.e. let $S = UDU^\top$ be its eigendecomposition with $D = \mathrm{diag}((\lambda_j)_{j\in[\![d]\!]})$ then $\psi(S) = U\psi(D)U^\top = U\mathrm{diag}((\psi(\lambda_j))_{j\in[\![d]\!]})U^\top$.
- Extract the block means $\widehat{\mu}_j \in \mathbb{R}^{p\times q}$ such that $\xi^{(j)} = \begin{pmatrix} \xi_{11}^{(j)} & \widehat{\mu}_j \\ \widehat{\mu}_j^\top & \xi_{22}^{(j)} \end{pmatrix}$.
- Compute the pairwise distances $r_{jl} = \|\widehat{\mu}_j - \widehat{\mu}_l\|_{\mathrm{op}}$ for $j, l \in [\![K]\!]$.
- Compute the vectors $r^{(j)} \in \mathbb{R}^K$ for $j \in [\![K]\!]$ where $r^{(j)}$ is the increasingly sorted version of $r_{j:}$.
- return $\widehat{\mu}_{\widehat{i}}$ where $\widehat{i} \in \arg\min_i r^{(i)}_{K/2}$.

One may guess that the choice of the scale parameter χ plays an important role to guarantee the quality of the estimate. This aspect is inherited from Catoni's original estimator for the mean of a heavy-tailed real random variable [73] from which Minsker's estimator [317], which we use to estimate the block means, is inspired. The following statement gives the optimal value for χ and the associated deviation bound satisfied by CM-MOM.

Proposition 4.5 (CM-MOM)**.** *Let $A_1, \dots, A_n \in \mathbb{R}^{p\times q}$ be an i.i.d sample following a random variable A with expectation $\mu = \mathbb{E}A$ such that a subset of indices $\mathcal{O} \subset [\![n]\!]$ are outliers and finite variance*

$$v(A) = \max\left(\left\|\mathbb{E}(A-\mu)(A-\mu)^\top\right\|_{\mathrm{op}}, \left\|\mathbb{E}(A-\mu)^\top(A-\mu)\right\|_{\mathrm{op}}\right) < \infty.$$

Let $\delta > 0$ be a failure probability and take $K = \lceil 18\log(1/\delta)\rceil < n$ blocks, we assume $n = mK$. Let $\widehat{\mu}$ be the CM-MOM estimate as defined above with scale parameter

$$\chi = \sqrt{\frac{2m\log(8(p+q))}{v(A)}}.$$

[6] In fact, one may allow up to $|\mathcal{O}| \leqslant K/2$ outliers at the price of worse constants in the resulting deviation bound. See the proof of Proposition 4.5

Assume we have $|\mathcal{O}| \leqslant K/12$ *then with probability at least* $1-\delta$ *we have :*

$$\left\|\widehat{\mu}-\mu\right\|_{\mathrm{op}} \leqslant 18\sqrt{\frac{v(A)\log(8(p+q))\log(1/\delta)}{n}}.$$

Proposition 4.5 is proven in Section 4.8.3 and enjoys a deviation rate which scales optimally, up to logarithmic factors, as $\sqrt{p+q}$ in the dimension [432]. This dependence is hidden by the factor $\sqrt{v(A)}$ which scales in that order (see for instance [416]). Although the dependence of the optimal scale χ on the unknown value of $v(A)$ constitutes an obstacle, previous experience using Catoni-based estimators [73, 191, 153] has shown that the choice is lenient and good results are obtained as long as a value of the correct scale is used. Possible improvements for Proposition 4.5 are to derive a bound with an additive instead of multiplicative term $\log(1/\delta)$ or supporting η-corruption. However, we are not aware of a more robust solution for matrix mean estimation in the general case than the above result.

Now that we have an adapted gradient estimation procedure, we can proceed to combine its deviation bound with our optimization theorems in order to obtain guarantees on learning performance.

Corollary 4.3. *In the context of Theorem 4.1 and Proposition 4.5, let the AMMD algorithm be run starting from* $\theta_0 \in \Theta = B_{\|\cdot\|}(\theta_0, R)$ *and using the CM-MOM estimator with sample splitting i.e. at each iteration a different batch of size* $\widetilde{n} = n/T$ *is used for gradient estimation with confidence* $\widetilde{\delta} = \delta/T$ *where* T *is the total number of iterations. Assume that each batch contains no more than* $K/12$ *outliers. Let* N *be the number of stages and* $\widehat{\theta}$ *the obtained estimator. With probability at least* $1-\delta$*, the latter satisfies :*

$$\left\|\widehat{\theta}-\theta^{\star}\right\|_2 \leqslant \frac{2^{-N/2}R}{\sqrt{\bar{s}}} + \sup_{\theta\in\Theta}\frac{360\sqrt{\bar{s}}}{\kappa}\sqrt{\frac{2v(G(\theta))\log(8(p+q))\log(1/\widetilde{\delta})}{\widetilde{n}}},$$

where $\|\cdot\|_2$ *denotes the Frobenius norm.*

Corollary 4.3 matches the optimal performance bounds given in classical literature for low-rank matrix recovery [241, 378, 67, 335] up to logarithmic factors. The previous statement is most similar to [223, Proposition 3.3] except that it applies for more general learning tasks and under much lighter data assumption.

4.6 Implementation and Numerical Experiments

In this section, we demonstrate the performance of the proposed algorithms on synthetic and real data. Before we proceed, we prefer to point out that our implementation does not *exactly* correspond to the previously given pseudo-codes. Indeed, as the reader may have noticed previously, certain instructions of AMMD and AMDA require the knowledge of quantities which are not available in practice and, even in a controlled setting, the estimation of some quantities (such as the maximum gradient error $\bar{\epsilon}$) may be overly conservative which generally impedes the proper convergence of the optimization. We list the main divergences of our implementation from the theoretically studied procedures of AMMD and AMDA given before :

1. For AMMD, we only use the conventional prox operator rather than the corrected $\widehat{\mathrm{prox}}$ operator defined in Section 4.3.

2. For both AMMD and AMDA, the whole data set is used at each step to compute a gradient estimate and no data-splitting is performed.

3. The radii R_k are taken constant equal to a fixed $R > 0$.

4. The stage lengths (T_k for AMMD and T' for AMDA) are fixed as constants.

5. The number of stages is determined through a maximum number of iterations but the algorithm stops after the last whole stage.

6. The within-stage step-sizes a_i in AMDA are fixed to a small constant (smaller than R) for more stability.

The constant stage-lengths are fixed using the following heuristic: run the MD/DA iteration while tracking the evolution of the empirical objective on a validation subset of the data[7] and set the stage-length as the number of steps before a plateau is reached. Reaching a plateau indicates that the current reference point $\theta^{(k)}$ has become too constraining for the optimization and more progress can be made after updating it.

The simplifications brought by points 1. and 2. are related to likely proof artifacts. Indeed as pointed out in Section 4.3 the use of the corrected $\widehat{\text{prox}}$ operator rather than simply prox is chiefly meant to ensure objective monotonicity while the data splitting ensures the gradient deviation bounds are usable in the proofs at each iteration.

Points 3. and 4. are due to the fact that the values of the stage-lengths T_k/T' and the radii R_k used in AMMD and AMDA are based on conservative estimates from the theoretical analysis making them unfit for practical implementation. Moreover, the said estimates use constants which cannot be identified in an arbitrary setting. For example, for least squares regression, the quadratic minorization constant κ depends on the data distribution which is unknown in general.

Finally, point 5. is commonplace for batch learning where one may simply iterate until convergence and point 6. follows the wisdom that smaller step-sizes ensure more stability.

Despite these differences, the numerical experiments we present below demonstrate that our implementations perform on par with the associated theoretical results.

Clearly, optimisation using Mirror Descent should be preferred over Dual Averaging due to its faster convergence speed. This is conditioned by the smoothness of the objective $\mathcal{L}$ which holds, for example, when the loss function ℓ is smooth. If ℓ is not smooth but the data distribution contains no atoms, one may still use Mirror Descent since it is reasonable to expect the objective $\mathcal{L}$ to be smoothed by the expectation (4.2.1). Note however that such an objective is likely not to satisfy Assumption 4.4 making Theorem 4.1 inapplicable. Still, in this case, if the weaker Assumption 4.6 holds, the expected performance is as stated in Theorem 4.2 with an improved number of required iterations of order $1/\bar{\epsilon}$ instead of $1/\bar{\epsilon}^2$ due to faster within-stage optimisation.

4.6.1 Synthetic sparse linear regression

We first test our algorithms on the classic problem of linear regression. We generate n covariates $X_i \in \mathbb{R}^d$ following a non-isotropic distribution with covariance matrix Σ and labels $Y_i = X_i^\top \theta^\star + \xi_i$ for a fixed s-sparse $\theta^\star \in \mathbb{R}^d$ and simulated noise entries ξ_i. The covariance matrix Σ is diagonal with entries drawn uniformly at random in $[1, 10]$.

We use the least-squares loss $\ell(z, y) = \frac{1}{2}(z - y)^2$ in this experiment and the problem parameters are $n = 500, d = 5000, s = 40$ and a sparsity upper bound $\bar{s} = 50$ is given to the algorithms instead of the real value. The noise variables ξ_i always follow a Pareto distribution with parameter $\alpha = 2.05$. Apart from that we consider three settings :

(a) The gaussian setting : the covariates follow a gaussian distribution.

[7] To remain consistent with a robust approach, the objective is estimated using a trimmed mean here as well.

(b) The heavytailed setting : the covariates are generated from a multivariate Student distribution with $\nu = 4.1$ degrees of freedom.

(c) The corrupted setting : the covariates follow the same Student distribution and 5% of the data ((X_i, Y_i) pairs) are corrupted.

We run various algorithms:

- AMMD using the trimmed mean estimator (`AMMD`).
- AMDA using the trimmed mean estimator (`AMDA`).
- The iterative thresholding procedure defined in [273] using the MOM estimator (`LLC_MOM`).
- The iterative thresholding procedure defined in [273] using the trimmed-mean estimator (`LLC_TM`).
- Lasso with CGD solver and the trimmed mean estimator as implemented in [153] (`Lasso_CGD_TM`).
- Lasso with CGD solver as implemented in Scikit Learn [354] (`Lasso_CGD`).

Another possible baseline is the algorithm proposed in [274]. Nevertheless, we do not include it here because it relies on the outlier removal algorithm inspired from [19]. The latter requires to run an SDP subroutine making it excessively slow as soon as the dimension is greater than a few hundreds.

Note that the "trimmed mean" estimator used in [273] is different from ours since they simply exclude the entries below and above a pair of empirical data quantiles. On the other hand, the estimator we define in Section 4.5.1 simply replaces the extreme values by the exceeded threshold before computing an average. This is also called a "Winsorized mean" and enjoys better statistical properties.

The algorithms using Lasso [414] optimize an ℓ_1 regularized objective. The regularization is weighted by a factor $2\sigma\sqrt{\frac{2\log(d)}{n}}$ where $\sigma^2 = \mathrm{Var}(\xi)$ is the noise variance. The previous regularization weight is known to ensure optimal statistical performance, see for instance [35]. The

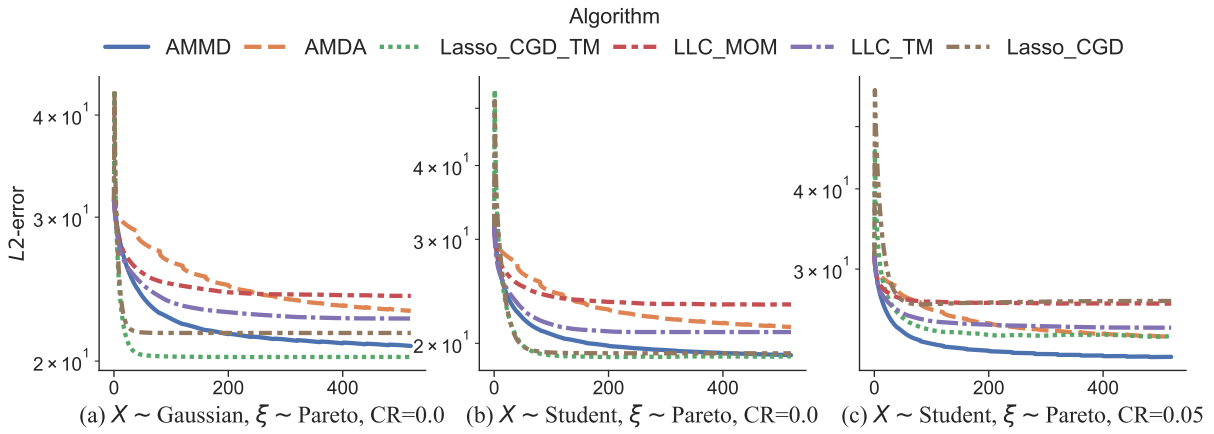

Figure 4.1: L2 error $\|\theta_t - \theta^\star\|_2$ (y-axis) against iterations (x-axis) for all the considered algorithms in the simulation settings.

experiment is repeated 30 times and the results are averaged. We do not display any confidence intervals for better readability. Figure 4.1 displays the results. We observe that Lasso based

methods quickly reach good optima in general and that the version using the robust trimmed mean estimator is sometimes superior in the presence of heavy tails and corruption in particular. `AMMD` reaches nearly equivalent optima, albeit significantly slower than Lasso methods as seen for settings (a) and (b). However, it is somehow more robust to corruption as seen on setting (c). Unfortunately, the `AMDA` algorithm struggles to closely approximate the original parameter. We mainly attribute this to slow convergence in settings (a) and (b). Nonetheless, `AMDA` is among the most robust algorithms to corruption as seen on setting (c). The remaining iterative thresholding based methods `LLC_MOM` and `LLC_TM` seem to generally stop at suboptimal optima. The Median-Of-Means variant `LLC_MOM` is barely more robust than `Lasso_CGD` in the corrupted setting (c). The `LLC_TM` variant is better but still inferior to `AMMD`. This reflects the superiority of the (Winsorized) trimmed mean used by `AMMD` and `Lasso_CGD_TM` to the conventional trimmed mean in `LLC_TM`.

4.6.2 Sparse classification on real data

We also carry out experiments on real high dimensional binary classification data sets. These are referred to as `gina` and `bioresponse` and were both downloaded from `openml.org`. We run `AMMD`, `AMDA`, `LLC_MOM` and `LLC_TM` with similar sparsity upperbounds and various levels of corruption and track the objective value, defined using the Logistic loss $\ell(z, y) = \log(1 + e^{-zy})$ (with $y \in \mathcal{Y} = \{\pm 1\}$), for each of them. The results are displayed on Figure 4.2 (average over 10 runs). In the non corrupted case, we see that all algorithms reach approximately equivalent optima whereas they display different levels of resilience when corruption is present. In particular, `LLC_MOM` is unsurprisingly the most vulnerable since it is based on Median-Of-Means which is not robust to η-corruption. The rest of the algorithms cope better thanks to the use of trimmed mean estimators, although `LLC_TM` seems to be a little less robust which is probably due to the previously mentioned difference in its gradient estimator. Finally, Figure 4.2 also shows that `AMMD` and `AMDA` (respectively using Mirror Descent and Dual Averaging) tend to reach generally better final optima despite converging a bit slower than the other algorithms. They also prove to be more stable, even when high step sizes are used.

4.7 Conclusion

In this work, we address the problem of robust supervised linear learning in the high-dimensional setting. In order to cover both smooth and non-smooth loss functions, we propose two optimisation algorithms which enjoy linear convergence speeds with only a mild dependence on the dimension. We combine these algorithms with various robust mean estimators, each of them tailored for a specific variant of the sparse estimation problem. We show that the said estimators are robust to heavy-tailed and corrupted data and allow to reach the optimal statistical rates for their respective instances of sparse estimation problems. Furthermore, their computation is efficient which favorably reflects on the computational cost of the overall procedure. We also confirm our theoretical results through numerical experiments where we evaluate our algorithms in terms of speed, robustness and performance of the final estimates. Finally, we compare our performances with the most relevant concurrent works and discuss the main differences. Perspectives for future work include considering other types of sparsity, divising an algorithm capable of reaching the optimal $s \log(d/s)/n$ rate for vanilla sparsity or considering problems beyond recovery of a single parameter such as, for example, additive sparse and low-rank matrix decomposition.

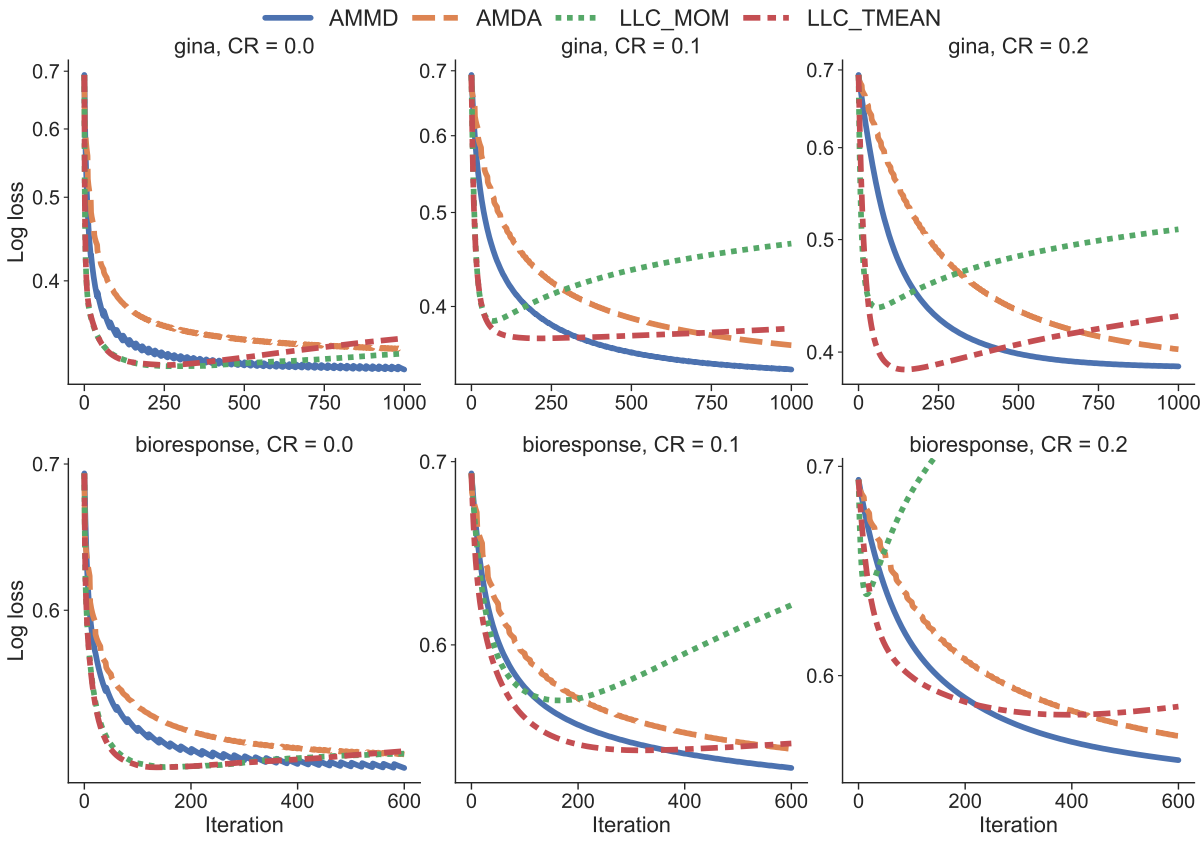

Figure 4.2: Log loss (y-axis) along training iterations (x-axis) on two data sets (rows) for 0% corruption (first column), 10% corruption (middle column) and 20% corruption (last column).

4.8 Proofs

4.8.1 Proofs for Section 4.3

Proof of Proposition 4.1

We use the abbreviations $\widehat{g}_t = \widehat{g}(\theta_t)$ and $g_t = g(\theta_t)$. Let $\phi \in \Theta$ be any parameter, we first write the optimality condition of the proximal operator defining each step $\theta_{t+1} = \text{prox}_\beta(\widehat{g}_t, \theta_t; \theta_0, \Theta)$. Using the convexity and smoothness properties of the objective $\mathcal{L}$, we find that :

$$\begin{aligned}\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\phi) &= \mathcal{L}(\theta_{t+1}) - \mathcal{L}(\theta_t) + \mathcal{L}(\theta_t) - \mathcal{L}(\phi) \\ &\leqslant \langle g(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 + \langle g(\theta_t), \theta_t - \phi \rangle \\ &= \langle g_t, \theta_{t+1} - \phi \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2. \end{aligned} \tag{4.8.1}$$

We have $\widehat{g}_t = g_t + \epsilon_t$ and the optimality condition says that for all $\phi \in \Theta$ we have the inequality :

$$\langle \beta \widehat{g}_t, \phi - \theta_{t+1} \rangle + \langle \nabla \omega_{\theta_0}(\theta_{t+1}) - \nabla \omega_{\theta_0}(\theta_t), \phi - \theta_{t+1} \rangle \geqslant 0.$$

Plugging this into (4.8.1) we get :

$$\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\phi) \leqslant \frac{1}{\beta}\langle \nabla \omega_{\theta_0}(\theta_{t+1}) - \nabla \omega_{\theta_0}(\theta_t), \phi - \theta_{t+1} \rangle + \langle \epsilon_t, \phi - \theta_{t+1} \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2$$

$$
\begin{aligned}
&= \frac{1}{\beta}(V_{\theta_0}(\phi, \theta_t) - V_{\theta_0}(\theta_{t+1}, \theta_t) - V_{\theta_0}(\phi, \theta_{t+1})) + \langle \epsilon_t, \phi - \theta_{t+1} \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 \\
&\leqslant \frac{1}{\beta}(V_{\theta_0}(\phi, \theta_t) - V_{\theta_0}(\phi, \theta_{t+1})) + \langle \epsilon_t, \phi - \theta_{t+1} \rangle,
\end{aligned}
$$

where the last step is due to the choice $\beta \leqslant 1/L$ and the strong convexity of V and the second step follows from the remarkable identity :

$$
\langle \nabla_\theta V_{\theta_0}(\theta, \theta'), z - \theta \rangle = V_{\theta_0}(z, \theta') - V_{\theta_0}(\theta, \theta') - V_{\theta_0}(z, \theta) \quad \text{for all } z, \theta, \theta', \theta_0 \in \mathbb{R}^d.
$$

It suffices to multiply the previous inequality by β, sum it for $t = 0 \dots, T-1$ and use the convexity of $\mathcal{L}$ to find that $\widehat{\theta}_T = \sum_{t=1}^T \theta_t / T$ satisfies :

$$
\mathcal{L}(\widehat{\theta}_T) - \mathcal{L}(\phi) \leqslant \frac{V_{\theta_0}(\phi, \theta_0) - V_{\theta_0}(\phi, \theta_T)}{\beta T} + \frac{1}{T}\sum_{t=0}^{T-1} \langle \epsilon_t, \phi - \theta_{t+1} \rangle.
$$

Then, it only remains to choose $\phi = \theta^\star$ to finish the proof.

Proof of Proposition 4.2

We proceed similarly to the previous Proposition. As previously we have :

$$
\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\phi) \leqslant \langle g_t, \theta_{t+1} - \phi \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2,
$$

where $\widehat{g}_t = g_t + \epsilon_t$. Let $\phi \in \Theta$, the optimality condition of $\theta_{t+1} = \widehat{\mathrm{prox}}_\beta(\widehat{g}_t, \theta_t; \theta_0, \Theta)$ reads :

$$
\langle \beta(\widehat{g}_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t)), \phi - \theta_{t+1} \rangle + \langle \nabla \omega_{\theta_0}(\theta_{t+1}) - \nabla \omega_{\theta_0}(\theta_t), \phi - \theta_{t+1} \rangle \geqslant 0,
$$

where $\partial_{\|\cdot\|}(\theta)$ is any subgradient of $\|\cdot\|$ at θ. Plugging this into (4.8.1) we get

$$
\begin{aligned}
\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\phi) &\leqslant \frac{1}{\beta}\langle \nabla \omega_{\theta_0}(\theta_{t+1}) - \nabla \omega_{\theta_0}(\theta_t), \phi - \theta_{t+1} \rangle + \langle \epsilon_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t), \phi - \theta_{t+1} \rangle \\
&\quad + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 \\
&= \frac{1}{\beta}(V_{\theta_0}(\phi, \theta_t) - V_{\theta_0}(\theta_{t+1}, \theta_t) - V_{\theta_0}(\phi, \theta_{t+1})) \\
&\quad + \langle \epsilon_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t), \phi - \theta_{t+1} \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 \\
&\leqslant \frac{1}{\beta}(V_{\theta_0}(\phi, \theta_t) - V_{\theta_0}(\phi, \theta_{t+1})) + \langle \epsilon_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t), \phi - \theta_{t+1} \rangle, \qquad (4.8.2)
\end{aligned}
$$

where the last step is due to the choice $\beta \leqslant 1/L$ and the strong convexity of V and the second step follows from the remarkable identity :

$$
\langle \nabla_\theta V_{\theta_0}(\theta, \theta'), z - \theta \rangle = V_{\theta_0}(z, \theta') - V_{\theta_0}(\theta, \theta') - V_{\theta_0}(z, \theta) \quad \text{for all } z, \theta, \theta', \theta_0 \in \mathbb{R}^d.
$$

Notice that since $\|\epsilon_t\|_* \leqslant \bar{\epsilon}$ for all t and using the identity $\langle \partial_{\|\cdot\|}(\theta), \theta \rangle = \|\theta\|$ which holds for any norm $\|\cdot\|$ we find :

$$
\langle \epsilon_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t), \theta_t - \theta_{t+1} \rangle = \langle \epsilon_t, \theta_t - \theta_{t+1} \rangle - \bar{\epsilon}\|\theta_{t+1} - \theta_t\| \leqslant 0,
$$

so that by taking $\phi = \theta_t$ in (4.8.2) we find that :

$$\mathcal{L}(\theta_{t+1}) \leqslant \mathcal{L}(\theta_t),$$

i.e. $\mathcal{L}(\theta_t)$ is monotonously decreasing with t. Using this observation, it suffices to average (4.8.2) over $t = 0 \dots, T-1$ to find that :

$$\mathcal{L}(\theta_T) - \mathcal{L}(\phi) \leqslant \frac{V_{\theta_0}(\phi, \theta_0) - V_{\theta_0}(\phi, \theta_T)}{\beta T} + \frac{1}{T}\sum_{t=0}^{T-1} \langle \epsilon_t + \bar{\epsilon}\partial_{\|\cdot\|}(\theta_{t+1} - \theta_t), \phi - \theta_{t+1} \rangle.$$

From here, the final bound is straight forward to derive by replacing $\phi = \theta^\star$ and using the fact that $\|\partial_{\|\cdot\|}(\theta)\|_* \leqslant 1$.

Proof of Theorem 4.1

The following Lemma is needed for this proof and that of Theorem 4.2.

Lemma 4.4 ([223, Lemma A.1])**.** *Let $\theta^\star \in \Theta$ be s-sparse, $\theta \in \Theta$, and let $\theta_s = \mathrm{sparse}(\theta) \in \arg\min\{\|\mu - \theta\| : \mu \in \Theta$ s-sparse$\}$. We have :*

$$\|\theta_s - \theta^\star\| \leqslant \sqrt{2s}\|\theta_s - \theta^\star\|_2 \leqslant 2\sqrt{2s}\|\theta - \theta^\star\|_2.$$

We would like to show by induction that $\|\theta^{(k)} - \theta^\star\| \leqslant R_k$ for $k \geqslant 0$. In the base case $k = 0$, we have $\|\theta^{(0)} - \theta^\star\| \leqslant R = R_0$. For a phase $k+1 \geqslant 1$ of the approximate Mirror Descent algorithm, assuming the property holds for k, by applying Lemma 4.4 and Proposition 4.2 we find :

$$\mathcal{L}(\widetilde{\theta}^{(k+1)}) - \mathcal{L}(\theta^\star) \leqslant \frac{\nu R_k^2}{\beta T_{k+1}} + 4\bar{\epsilon}R_k \leqslant 5\bar{\epsilon}R_k, \tag{4.8.3}$$

where the last inequality uses that that $T_{k+1} = \left\lceil \frac{\nu R_k}{\beta \bar{\epsilon}} \right\rceil$. Using the quadratic growth hypothesis (Assumption 4.4) leads to :

$$\left\|\theta^{(k+1)} - \theta^\star\right\|^2 \leqslant 2\bar{s}\left\|\theta^{(k+1)} - \theta^\star\right\|_2^2 \leqslant 8\bar{s}\left\|\widetilde{\theta}^{(k+1)} - \theta^\star\right\|_2^2 \leqslant \frac{40\bar{s}\bar{\epsilon}R_k}{\kappa}. \tag{4.8.4}$$

We have just obtained the bound $\left\|\theta^{(k+1)} - \theta^\star\right\| =: \widehat{R}_{k+1} \leqslant h(R_k) := \sqrt{\frac{40\bar{s}\bar{\epsilon}R_k}{\kappa}}$. It is easy to check that $h(r)$ has a unique fixed point $R^\star := \frac{40\bar{s}\bar{\epsilon}}{\kappa}$ and that for $r \geqslant R^\star$ we have $h'(r) \leqslant 1/2$. Assuming that the former bound holds for $r = R_k$ (otherwise there is nothing to prove) we find :

$$\begin{aligned} \widehat{R}_{k+1} - R^\star \leqslant h(R_k) - h(R^\star) &\leqslant \frac{1}{2}(R_k - R^\star) \\ \implies \widehat{R}_{k+1} &\leqslant \frac{1}{2}(R_k + R^\star) = R_{k+1}, \end{aligned}$$

this finishes the induction argument. By unrolling the recursive definition of R_k, we obtain that, for all $k \geqslant 1$:

$$R_k \leqslant 2^{-k}R_0 + R^\star = 2^{-k}R_0 + \frac{40\bar{s}\bar{\epsilon}}{\kappa}.$$

The main bound of the Theorem then follows by plugging the above inequality with $k = K - 1$ into (4.8.4) and using the fact that $R_0 \geqslant R^\star$ and the standard inequality $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ which holds for all $a, b \geqslant 0$. The bound on the objective is obtained similarly.

Let us compute T, the total number of iterations necessary for this bound to hold. Given

the minimum number of iterations T_k necessary for stage k we have :

$$T = \sum_{k=1}^{K} T_k \leqslant \sum_{k=1}^{K} \Big(\frac{\nu R_{k-1}}{\beta \bar{\epsilon}} + 1 \Big) \leqslant K + \frac{\nu}{\beta \bar{\epsilon}} \sum_{k=0}^{K-1} (2^{-k} R_0 + R^\star)$$
$$\leqslant \frac{2R_0 \nu}{\beta \bar{\epsilon}} + K \Big(1 + \frac{40 \nu \bar{s}}{\kappa \beta} \Big).$$

This completes the proof.

4.8.2 Proofs for Section 4.4

Proof of Proposition 4.3

For a sequence of iterates $(\theta_t)_{t=0\dots T}$ we introduce the notations :

$$\ell_t(\theta) = \sum_{i=0}^{t} a_i (\mathcal{L}(\theta_i) + \langle \widehat{g}_i, \theta - \theta_i \rangle) \quad \text{and} \quad \psi_t^* = \min_{\theta \in \Theta} \ell_t(\theta) + \gamma_t \omega(\theta).$$

We will show the following inequality by induction :

$$A_t f(\theta_t) \leqslant \psi_t^* + \widehat{B}_t,$$

where we define $\widehat{B}_t = \sum_{i=0}^{t} \frac{a_i^2}{2\gamma_{i-1}} \|g_i\|_*^2 + 2a_i R \|\epsilon_i\|_*$ with the convention $\gamma_{-1} = \gamma_0$. Assume it holds for $t \geqslant 0$, since $\gamma_{t+1} \geqslant \gamma_t$ we have :

$$\begin{aligned} \psi_{t+1}^* &= \min_{\theta \in \Theta} \ell_t(\theta) + a_{t+1}(\mathcal{L}(\theta_{t+1}) + \langle \widehat{g}_{t+1}, \theta - \theta_{t+1} \rangle) + \gamma_{t+1} \omega(\theta) \\ &\geqslant \min_{\theta \in \Theta} \ell_t(\theta) + a_{t+1}(\mathcal{L}(\theta_{t+1}) + \langle \widehat{g}_{t+1}, \theta - \theta_{t+1} \rangle) + \gamma_t \omega(\theta). \end{aligned} \tag{4.8.5}$$

Note that, by definition, θ_t^+ realizes the minimum $\psi_t^* = \min_{\theta \in \Theta} \ell_t(\theta) + \gamma_t \omega(\theta) = \ell_t(\theta_t^+) + \gamma_t \omega(\theta_t^+)$, so that for all $\theta \in \Theta$ we have

$$\langle \nabla \ell_t(\theta_t^+) + \gamma_t \nabla \omega(\theta_t^+), \theta - \theta_t^+ \rangle \geqslant 0. \tag{4.8.6}$$

Also, using the convexity of ℓ_t and the strong convexity of $\omega(\cdot)$ we have :

$$\begin{aligned} \ell_t(\theta) + \gamma_t \omega(\theta) \geqslant & \ell_t(\theta_t^+) + \langle \nabla \ell_t(\theta_t^+), \theta - \theta_t^+ \rangle + \\ & \gamma_t \big(\omega(\theta_t^+) + \langle \nabla \omega(\theta_t^+), \theta - \theta_t^+ \rangle + \frac{1}{2} \|\theta - \theta_t^+\|^2 \big). \end{aligned} \tag{4.8.7}$$

By combining Inequalities (4.8.5), (4.8.6) and (4.8.7), we find that :

$$\psi_{t+1}^* \geqslant \min_{\theta \in \Theta} \psi_t^* + \frac{\gamma_t}{2} \|\theta - \theta_t^+\|^2 + a_{t+1}(\mathcal{L}(\theta_{t+1}) + \langle \widehat{g}_{t+1}, \theta - \theta_{t+1} \rangle).$$

Now, using the induction hypothesis $A_t f(\theta_t) \leqslant \psi_t^* + \widehat{B}_t$, we compute that :

$$\begin{aligned} \psi_{t+1}^* &\geqslant \min_{\theta \in \Theta} A_t f(\theta_t) - \widehat{B}_t + \frac{\gamma_t}{2} \|\theta - \theta_t^+\|^2 + a_{t+1}(\mathcal{L}(\theta_{t+1}) + \langle \widehat{g}_{t+1}, \theta - \theta_{t+1} \rangle) \\ &\geqslant \min_{\theta \in \Theta} A_t (f(\theta_{t+1}) + \langle g_{t+1}, \theta_t - \theta_{t+1} \rangle) - \widehat{B}_t + \frac{\gamma_t}{2} \|\theta - \theta_t^+\|^2 + a_{t+1}(\mathcal{L}(\theta_{t+1}) + \\ & \langle g_{t+1}, \theta - \theta_{t+1} \rangle) - 2R a_{t+1} \epsilon_{t+1} \end{aligned}$$

$$
\begin{aligned}
&\geqslant \min_{\theta\in\Theta} A_{t+1}f(\theta_{t+1}) - \widehat{B}_t + \frac{\gamma_t}{2}\|\theta-\theta_t^+\|^2 + a_{t+1}\langle g_{t+1}, \theta-\theta_t^+\rangle - 2Ra_{t+1}\epsilon_{t+1}\\
&\geqslant \min_{\theta\in\Theta} A_{t+1}f(\theta_{t+1}) - \widehat{B}_t - \frac{a_{t+1}^2}{2\gamma_t}\|\widehat{g}_t\|_*^2 - 2Ra_{t+1}\|\epsilon_{t+1}\|_*\\
&= \min_{\theta\in\Theta} A_{t+1}f(\theta_{t+1}) - \widehat{B}_{t+1},
\end{aligned}
$$

where the penultimate inequality uses that $A_{t+1}\theta_{t+1} = A_t\theta_t + a_{t+1}\theta_t^+$. It only remains to check the base case :

$$
\begin{aligned}
\psi_0^* &= \min_{\theta\in\Theta} a_0(\mathcal{L}(\theta_0) + \langle \widehat{g}_0, \theta-\theta_0\rangle) + \frac{\gamma_0}{2}\omega(\theta)\\
&\geqslant \min_{\theta\in\Theta} a_0(\mathcal{L}(\theta_0) + \langle g_0, \theta-\theta_0\rangle) + \frac{\gamma_0}{2}\omega(\theta) + \min_{\theta\in\Theta} a_0\langle \epsilon_0, \theta-\theta_0\rangle\\
&\geqslant A_0\mathcal{L}(\theta_0) - \frac{a_0^2}{2\gamma_{-1}}\|g_0\|_*^2 - 2a_0R\|\epsilon_0\|_* = A_0\mathcal{L}(\theta_0) - \widehat{B}_0,
\end{aligned}
$$

which completes the induction. Now we can compute :

$$
\begin{aligned}
A_t\langle s_t, \theta^\star\rangle + \gamma_t\omega(\theta^\star) &\geqslant A_t\langle s_t, \theta_t^+\rangle + \gamma_t\omega(\theta_t^+) + \frac{\gamma_t}{2}\|\theta_t^+ - \theta^\star\|^2\\
&\geqslant \psi_t^* - \sum_{i=0}^t a_i(\mathcal{L}(\theta_i) - \langle \widehat{g}_i, \theta_i\rangle) + \frac{\gamma_t}{2}\|\theta_t^+ - \theta^\star\|^2\\
&\geqslant A_t(\mathcal{L}(\theta_t) - \mathcal{L}(\theta^\star) + \langle s_t, \theta^\star\rangle) - \widehat{B}_t +\\
&\sum_{i=0}^t a_i(\mathcal{L}(\theta^\star) - \mathcal{L}(\theta_i) - \langle g_i, \theta^\star - \theta_i\rangle) - \sum_{i=0}^t a_i\langle \epsilon_i, \theta^\star - \theta_i\rangle + \frac{\gamma_t}{2}\|\theta_t^+ - \theta^\star\|^2,
\end{aligned}
$$

which, by rearranging and using the convexity of $\mathcal{L}$, leads to :

$$
\begin{aligned}
A_t(\mathcal{L}(\theta_t) - \mathcal{L}(\theta^\star)) + \frac{\gamma_t}{2}\|\theta_t^+ - \theta^\star\|^2 &\leqslant \gamma_t\omega(\theta^\star) + \widehat{B}_t + \sum_{i=0}^t a_i\langle \epsilon_i, \theta^\star - \theta_i\rangle\\
&\leqslant \gamma_t\omega(\theta^\star) + \sum_{i=0}^t \frac{a_i^2}{2\gamma_{i-1}}\|g_i\|_*^2 + 4A_tR\bar{\epsilon}.
\end{aligned}
$$

For the particular case $a_t = 1$ and $\gamma_t = \sqrt{t+1}$ we have $A_t = t+1$. Moreover, by summing the inequality $\frac{1}{\sqrt{i+1}} \leqslant \int_i^{i+1} \frac{du}{\sqrt{u}}$ we get that $\sum_{i=1}^t \frac{1}{\sqrt{i}} \leqslant 2\sqrt{t} - 1$. Using this estimate and the Lipschitz property of the objective quickly yields the last part of the Theorem.

Proof of Theorem 4.2

We will show by induction that the inequality $\|\theta^{(k)} - \theta^\star\| \leqslant R_k$ holds for all $k \geqslant 0$. The case $k = 0$ holds by assumption. Note that based on Proposition 4.3 with the choice $a_i = R$ and $\gamma_i = \sqrt{i+1}$ and using the quadratic growth bound (4.3.4) for ω we get :

$$
\mathcal{L}(\theta_{T'}) - \mathcal{L}(\theta^\star) \leqslant \frac{(\nu + M^2)R}{\sqrt{T'}} + 4R\bar{\epsilon} \leqslant 5\bar{\epsilon}R, \tag{4.8.8}
$$

where the last step follows from the choice of T'. At the same time, at the end of stage $k+1$, we have the alternative :

- Either $\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2 \leqslant \lambda$: then using Lemma 4.4 we find :

$$\begin{aligned}\big\|\theta^{(k+1)} - \theta^\star\big\|^2 &= \big\|\operatorname{sparse}_{\bar{s}}(\widetilde{\theta}^{(k+1)}) - \theta^\star\big\|^2 \leqslant 8\bar{s}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2^2 \\ &\leqslant \frac{16\lambda\bar{s}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2^2}{\lambda + \big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2} \leqslant \frac{16\lambda\bar{s}}{\kappa}\big(\mathcal{L}(\widetilde{\theta}^{(k+1)}) - \mathcal{L}(\theta^\star)\big) \\ &\leqslant \frac{80\lambda\bar{s}\bar{\epsilon}R_k}{\kappa} = R^\star R_k, \end{aligned} \tag{4.8.9}$$

  where the last inequality is an application of (4.8.8) since $\widetilde{\theta}^{(k+1)} = \theta_{T'}^{(k+1)}$.

- Or we have $\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2 > \lambda$: then only a linear regime holds and using Lemma 4.4 we get :

$$\begin{aligned}\big\|\theta^{(k+1)} - \theta^\star\big\| &\leqslant \sqrt{2\bar{s}}\big\|\theta^{(k+1)} - \theta^\star\big\|_2 \leqslant 2\sqrt{2\bar{s}}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2 \\ &\leqslant \frac{4\sqrt{2\bar{s}}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2^2/\lambda}{1 + \big\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2/\lambda} = \frac{4\sqrt{2\bar{s}}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2^2}{\lambda + \big\|\widetilde{\theta}^{(k+1)} - \theta^\star\big\|_2} \\ &\leqslant \frac{4\sqrt{2\bar{s}}}{\kappa}\big(\mathcal{L}(\widetilde{\theta}^{(k+1)}) - \mathcal{L}(\theta^\star)\big) \leqslant \frac{20\bar{\epsilon}R_k\sqrt{2\bar{s}}}{\kappa} = \tau R_k, \end{aligned} \tag{4.8.10}$$

  where we used the inequality $1 \leqslant 2x/(1+x)$ valid for all $x \geqslant 1$ on the quantity $\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2/\lambda$.

Similarly to the proof of Theorem 4.1, inequality (4.8.9) implies that $\big\|\theta^{(k+1)} - \theta^\star\big\| \leqslant \frac{1}{2}(R_k + R^\star)$ so that we obtained :

$$\big\|\theta^{(k+1)} - \theta^\star\big\| \leqslant \sqrt{2\bar{s}}\big\|\theta^{(k+1)} - \theta^\star\|_2 \leqslant 2\sqrt{2\bar{s}}\big\|\widetilde{\theta}^{(k+1)} - \theta^\star\|_2 \leqslant \max\big(\tau R_k, \frac{1}{2}(R_k + R^\star)\big) = R_{k+1},$$

which finishes the induction to show (4.4.1). Inequality (4.4.2) then follows using (4.8.8). Note that, since we assume $\tau < 1$ and $R_0 \geqslant R^\star$, the sequence $(R_k)_{k\geqslant 0}$ is decreasing. We now distinguish two phases :

- The linear phase : if $\tau R_0 > \frac{1}{2}(R_0 + R^\star)$ (which implies $\tau > 1/2$ since $R_0 \geqslant R^\star$) then while $\tau R_k > \frac{1}{2}(R_k + R^\star)$ we have $R_{k+1} = \tau R_k = \tau^{k+1} R_0$ and the number of stages necessary to reverse the previous inequality is :

$$\log\Big(\frac{(2\tau - 1)R_0}{R^\star}\Big)\Big/\log(1/\tau) \leqslant \log\Big(\frac{R_0}{R^\star}\Big)\Big/\log(1/\tau).$$

- The quadratic phase : let $K_1 \geqslant 0$ be the first stage index such that we have $\tau R_k \leqslant \frac{1}{2}(R_k + R^\star)$ and so $R_{k+1} = \frac{1}{2}(R_k + R^\star)$ and by iteration $R_{l+K_1} \leqslant 2^{-l} R_{K_1} + R^\star$. In all cases $R_{K_1} \leqslant R_0$ so the number of necessary stages is :

$$\frac{\log(R_{K_1}/R^\star)}{\log(2)} \leqslant \frac{\log(R_0/R^\star)}{\log(2)}.$$

We have shown that the overall number of necessary stages is at most $\log(R_0/R^\star)\Big(\frac{1}{\log(2)} + \frac{1}{\log(1/\tau)}\Big)$. The Theorem's final claim then follows since the number of per-stage iterations is constant equal to $T' = \Big\lceil\Big(\frac{\nu + M^2}{\bar{\epsilon}}\Big)^2\Big\rceil$.

4.8.3 Proofs for Section 4.5

Proof of Lemma 4.1

Let $\theta \in \Theta$, using Assumption 4.2 we have:

$$|\ell(\theta^\top X, Y)| \leqslant C_{\ell,1} + C_{\ell,2}|\theta^\top X - Y|^2 \leqslant C_{\ell,1} + 2C_{\ell,2}(|\theta^\top X|^2 + |Y|^2).$$

Taking the expectation and using Assumption 4.2 again shows that the objective $\mathcal{L}(\theta)$ is well defined. Next, for all $j \in [\![d]\!]$, simple algebra gives:

$$\begin{aligned}
\big|\ell'(\theta^\top X, Y)X_j\big|^2 &\leqslant \big|\big(C'_{\ell,1} + C'_{\ell,2}|\theta^\top X - Y|\big)X^j\big|^2 \\
&\leqslant 2\big(\big|C'_{\ell,1}X^j\big|^2 + (C'_{\ell,2}(|(\theta^\top X)X^j| + |YX^j|))^2\big) \\
&\leqslant 2\Big(\big|C'_{\ell,1}X^j\big|^2 + \Big(C'_{\ell,2}\Big(\sum_{k=1}^d |\theta_k||(X^k)X^j| + |YX^j|\Big)\Big)^2\Big) \\
&\leqslant 2\Big(\big|C'_{\ell,1}X^j\big|^2 + 2(C'_{\ell,2})^2\Big(d\sum_{k=1}^d |\theta_k|^2|(X^k)X^j|^2 + |YX^j|^2\Big)\Big).
\end{aligned}$$

Recall that we assume $\mathbb{E}|X^j\big|^2 < \infty$ and $\mathbb{E}|YX^j|^2 < \infty$, moreover, using a Cauchy Schwarz inequality, we find:

$$\mathbb{E}\big|(X^k)X^j\big|^2 \leqslant \sqrt{\mathbb{E}\big|X^k\big|^4\mathbb{E}\big|X^j\big|^4},$$

which is also assumed finite. This concludes the proof of Lemma 4.1.

Dual averaging for vanilla sparse estimation

Corollary 4.4. *In the context of Theorem 4.2 and Lemma 4.2, let the AMDA algorithm be run starting from $\theta_0 \in \Theta = B_{\|\cdot\|}(\theta_0, R)$ using the coordinatewise trimmed mean estimator with sample splitting i.e. at each iteration a different batch of size $\widetilde{n} = n/T$ is used for gradient estimation with confidence $\widetilde{\delta} = \delta/T$ where T is the total number of iterations. Let K be the number of stages and $\widehat{\theta}$ the obtained estimator. Denote $\sigma^2_{\max} = \sup_{\theta\in\Theta}\max_{j\in[\![d]\!]}\mathrm{Var}(\ell'(\theta^\top X, Y)X^j)$, with probability at least $1-\delta$, the latter satisfies :*

$$\big\|\widehat{\theta} - \theta^\star\big\|_2 \leqslant \tau^{K\wedge K_1}2^{(K_1-K)\wedge 0}\frac{R}{\sqrt{2\bar{s}}} + \frac{280\lambda\sqrt{2\bar{s}}\sigma_{\max}}{\kappa}\sqrt{4\eta + 6\frac{\log(4/\widetilde{\delta}) + \log(d)}{\widetilde{n}}}.$$

with K_1 an integer such that $K_1 \leq \log\Big(\frac{\kappa R}{80\lambda\bar{s}\bar{\epsilon}}\Big)\Big/\log(1/\tau)$ with $\tau = \frac{10\sqrt{8\bar{s}\bar{\epsilon}}}{\kappa} < 1$ by assumption and

$$\bar{\epsilon} = 7\sigma_{\max}\sqrt{4\eta + 6\frac{\log(4/\delta) + \log(d)}{n}}.$$

Proof. By the proof of Theorem 4.2, we have $R = R_0$ and K_1 is defined as the first stage index k such that we have $\tau R_k \leqslant \frac{1}{2}(R_k + R^\star)$ implying that $R_{k+1} = \frac{1}{2}(R_k + R^\star)$ and hence $R_{l+K_1} \leqslant 2^{-l}R_{K_1} + R^\star$ for $l \geqslant 0$. We also had $K_1 \leq \log\Big(\frac{R}{R^\star}\Big)\Big/\log(1/\tau)$ with $R^\star = \frac{80\lambda\bar{s}\bar{\epsilon}}{\kappa}$. Using Theorem 4.2, it follows that:

$$\sqrt{2\bar{s}}\big\|\theta^{(k)} - \theta^\star\big\|_2 \leqslant R_k \leqslant 2^{(K_1-k)\wedge 0}\tau^{k\wedge K_1}R + R^\star,$$

whence the result is easily obtained by plugging the value of $R^\star$ and using Lemma 4.2 with a union bound argument over all iterations T in order to bound $\bar{\epsilon} = \max_i \|\epsilon_i\|_*$ as defined in Proposition 4.3. □

Stability based robust mean estimator

The following robust mean estimation algorithm is drawn from [117] and allows to compute an estimator with the properties stated in Proposition 4.4. Let T be the set of sample vectors whose mean we wish to estimate. Define a weight function $w : T \to \mathbb{R}_+$ initialized as $w(x) = 1/|T|$ for all $x \in T$. Given a fraction $\epsilon < 1/2$, repeat the following steps :

- Compute $\mu(w) = \frac{1}{\|w\|_1} \sum_{x \in T} w(x)x$.
- Compute $\Sigma(w) = \frac{1}{\|w\|_1} \sum_{x \in T} w(x)(x - \mu(w))(x - \mu(w))^\top$.
- Compute v the largest eigenvector of $\Sigma(w)$.
- Define $g(x)$ for $x \in T$ as $g(x) = |v \cdot (x - \mu(w))|^2$.
- Find the largest t such that $\sum_{x \in T:\, g(x) \geqslant t} w(x) \geqslant \epsilon$.
- Define $f(x) = \begin{cases} g(x) & \text{if } g(x) \geqslant t \\ 0 & \text{otherwise.} \end{cases}$
- Let m be the largest value of $f(x)$ for any $x \in T$ with $w(x) \neq 0$.
- Set $w(x)$ to $w(x)(1 - f(x)/m)$ for all $x \in T$.

The loop is repeated until $\|w\|_1 < 1 - 2\epsilon$, at which point $\mu(w)$ is returned.

Closed form computation of the prox operator

In both Sections 4.3 and 4.4 the optimization methods are defined using a prox operator which involves solving an optimization problem of the following form:

$$\arg\min_{\|\theta\| \leqslant R} \langle u, \theta \rangle + \omega(\theta)$$

for some $R > 0$ and $u \in \Theta^*$.

Vanilla/Group-sparse case. We consider the group sparse case where $\Theta \subset \mathbb{R}^{d \times K}$, the groups are the rows of $\theta \in \Theta$ and we use the norm $\|\cdot\| = \|\cdot\|_{1,2}$ and the usual scalar product $\langle u, \theta \rangle = u^\top \theta$. We use the prox function ω defined as $\omega(\theta) = C\|\theta\|_{p,2}^2 = C\big(\sum_{i=1}^d \|\theta_{i,:}\|_2^p\big)^{2/p}$ with $p = 1 + 1/\log(d)$ and $\theta_{i,:}$ the i-th row of θ. Note that, for $K = 1$ we retrieve the usual linear learning setting. The setting $K > 1$ can be used, for example, for multiclass classification.

In order to obtain a closed form solution, we start by writing the Lagrangian :

$$\mathcal{L}(\theta) = \langle u, \theta \rangle + \omega(\theta) + \lambda(\|\theta\| - R), \tag{4.8.11}$$

where we introduce the multiplier $\lambda \geqslant 0$. We initially assume the latter known and try to find a critical point for $\mathcal{L}$ that is $\theta \in \mathbb{R}^{d \times K}$ such that :

$$\partial \mathcal{L}(\theta) = u + \nabla \omega(\theta) + \partial \|\cdot\|_{p,2}(\theta) \ni 0,$$

where we denoted $\partial\|\cdot\|_{1,2}(\theta)$ the subdifferential of the norm $\|\cdot\|_{1,2}$ since the latter is not differentiable whenever $\theta_{i,:}=0$ for some $i\in[\![d]\!]$.

Defining the function $h_\alpha(\theta):\mathbb{R}^{d\times K}\to\mathbb{R}^{d\times K}$ such that $h_\alpha(\theta)_{i,j}=\frac{\theta_{i,j}}{\|\theta_{i,:}\|_2^\alpha}$ for θ such that $\theta_{i,:}\neq 0$ for all i, we can write for such θ :

$$\nabla\omega(\theta)=2C\|\theta\|_{p,2}^{2-p}h_{2-p}(\theta)\quad\text{ and }\quad\partial\|\cdot\|_{1,2}(\theta)=h_1(\theta).$$

When $\theta_{i,:}=0$ for some i, a subgradient of $\|\cdot\|_{1,2}$ can be obtained by using this definition and plugging any subunit vector for index i. Assuming that $\theta_{i,:}\neq 0$ for all $i\in[\![d]\!]$, a critical point of the Lagrangian must satisfy for all i,j :

$$u_{i,j}+\theta_{i,j}\Big(2C\Big(\frac{\|\theta\|_{p,2}}{\|\theta_{i,:}\|_2}\Big)^{2-p}+\frac{\lambda}{\|\theta_{i,:}\|_2}\Big)=0.$$

From here, a quick computation yields for all i that :

$$\|\theta_{i,:}\|_2=\frac{\|u_{i,:}\|_2}{\Big(2C\Big(\frac{\|\theta\|_{p,2}}{\|\theta_{i,:}\|_2}\Big)^{2-p}+\frac{\lambda}{\|\theta_{i,:}\|_2}\Big)}\quad\text{and hence}\quad 2C\|\theta\|_{p,2}^{2-p}\|\theta_{i,:}\|_2^{p-1}=\|u_{i,:}\|_2-\lambda.$$

Notice that this equality cannot hold when $\|u_{i,:}\|_2<\lambda$, in this case, we deduce that $\theta_{i,:}=0$ which satisfies the critical point condition. This leads to the relation :

$$\theta_{i,j}=\frac{-\beta_i u_{i,j}}{2C\Big(\frac{\|\theta\|_{p,2}}{\|\theta_{i,:}\|_2}\Big)^{2-p}+\frac{\lambda}{\|\theta_{i,:}\|_2}}\qquad\text{with }\beta_i=\mathbf{1}_{\|u_{i,:}\|_2>\lambda}.$$

From here, it is easy to figure out that $\|\theta\|_{p,2}=\Big(\sum_{i=1}^d\beta_i\Big(\frac{\|u_{i,:}\|_2-\lambda}{2C}\Big)^{p/(p-1)}\Big)^{(p-1)/p}$. All computations are now possible knowing λ.

To find the latter's value, we plug the formula we have for θ into the constraint $\|\theta\|_{1,2}\leqslant R$. After a few manipulations, we find the constraint is satisfied for λ such that :

$$\frac{1}{2C}\Big(\sum_{i=1}^d\beta_i\big(\|u_{i,:}\|_2-\lambda\big)^{1/(p-1)}\Big)\Big(\sum_{i=1}^d\beta_i\big(\|u_{i,:}\|_2-\lambda\big)^{p/(p-1)}\Big)^{(p-2)/p}\leqslant R,$$

(recall that the β_is also depend on λ). It only remains to choose the smallest $\lambda\geqslant 0$ such that the above inequality holds.

Low-rank matrix case. In the low-rank matrix case the parameter space is $\Theta\subset\mathbb{R}^{p\times q}$ and the norm $\|\cdot\|$ represents the nuclear norm $\|\theta\|=\|\sigma(\theta)\|_1$ with $\sigma(\theta)$. The d.-g.f./proximal function is the scaled squared p-Schatten norm $\omega(\theta)=C\|\sigma(\theta)\|_p^2$ with $p=1+\frac{1}{12\log(q)}=1+r$ and the scalar product is defined as $\langle u,v\rangle=\mathrm{Tr}(u^\top v)$.

We introduce the notation $h_r(\theta)=\|\sigma(\theta)\|_{1+r}$. Letting $UDV^\top$ denote an SVD of $\theta\in\mathbb{R}^{p\times q}$, the gradient of h_r at θ for $r>0$ is given by $\nabla h_r(\theta)=U\Big(\frac{D}{h_r(\theta)}\Big)^r V^\top$. The nuclear norm h_0 is not differentiable but a subgradient is given by $\partial h_0(\theta)=UV^\top+W$ for any W such that $\|W\|_{\mathrm{op}}\leqslant 1$ and $UU^\top W=0$ and $WVV^\top=0$ (see [439]).

In order to define the prox operator in this setting we need to solve problem (4.8.11) again which amounts to finding a critical point $\theta=UDV^\top$ such that :

$$u+2Ch_r(\theta)^{1-r}UD^rV^\top+\lambda UV^\top\ni 0.$$

We define θ by choosing U and V such that $u = UD_uV^\top$ is an SVD of u. Thanks to this choice, it only remains to choose $D = \mathrm{diag}(\sigma(\theta))$ properly in order to have :

$$\sigma(u) + 2C\|\sigma(\theta)\|_{1+r}^{1-r}\sigma(\theta)^r + \lambda(\mathbf{1}_{\sigma(\theta)\neq 0} + w) = 0,$$

where the power $\sigma(\theta)^r$ is computed coordinatewise, $\mathbf{1}_{\sigma(\theta)\neq 0}$ is the indicator vector of non zero coordinates of $\sigma(\theta)$ and for some vector w such that $|w_j| \leqslant 1$ for all j supported on the coordinates where $\sigma(\theta) = 0$.

The problem then becomes analogous to finding the proximal operator for Vanilla sparsity and after some computations we find that the solution is given by :

$$\sigma(\theta) = -\frac{\mathcal{T}_\lambda(u)^{1/r}}{2C\|\mathcal{T}_\lambda(u)\|_{(1+r)/r}^{(1-r)/r}},$$

where the soft threshold operator is defined by $\mathcal{T}_\lambda(u)_j = \mathrm{sign}(u_j)\max(0, |u_j| - \lambda)$ and λ is the smallest real number such that :

$$\frac{1}{2C}\|\mathcal{T}_\lambda(u)\|_{1/r}^{1/r} \cdot \|\mathcal{T}_\lambda(u)\|_{(r-1)/r}^{(r+1)/r} \leqslant R.$$

Proof of Proposition 4.5

Let $B_1, \dots, B_K$ be a partition of $[\![n]\!]$ into disjoint equal sized blocks. We assume that the number of outliers is $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$ where $0 < \varepsilon < 1$ will be fixed later. Let $\mathcal{K} = \{k \in [\![K]\!] : B_k \cap \mathcal{O} = \emptyset\}$ be the set of outlier-free blocks. Denote the block means for $j \in [\![K]\!]$ as

$$\xi^{(j)} = \frac{1}{m\chi}\sum_{i\in B_j}\psi(\chi\widetilde{A}_i),$$

where χ is temporarily chosen as $\chi = \sqrt{\frac{2m\log(2(p+q)/\delta')}{v(A)}}$ for some $0 < \delta' < 1/2$. By applying Corollary 3.1 from [317], we obtain that with probability at least $1 - \delta'$

$$\left\|\widehat{\mu}^{(k)} - \mu\right\|_{\mathrm{op}} \leqslant \sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}}. \tag{4.8.12}$$

Now let $r_{jl} = \|\widehat{\mu}^{(j)} - \widehat{\mu}^{(l)}\|_{\mathrm{op}}$, denote $r^{(j)}$ the increasingly sorted version of $r_{j:}$ and let $\widehat{j} \in \arg\min_j r^{(j)}_{\lceil K/2\rceil}$ so that $\widehat{\mu} = \widehat{\mu}^{(\widehat{j})}$.

Define the events $E_j = \left\{\left\|\widehat{\mu}^{(j)} - \mu\right\|_{\mathrm{op}} \leqslant \sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}}\right\}$ and assume that we have $\sum_{k=1}^K \mathbf{1}_{E_k} > K/2$ i.e. over half of the block means satisfy Inequality (4.8.12) simultaneously. Then there exists $j' \in [\![K]\!]$ such that the block $\widehat{j}$ satisfies

$$r^{(\widehat{j})}_{K/2} = \left\|\widehat{\mu}^{(\widehat{j})} - \widehat{\mu}^{(j')}\right\|_{\mathrm{op}} \leqslant \left\|\widehat{\mu}^{(\widehat{j})} - \mu\right\|_{\mathrm{op}} + \left\|\widehat{\mu}^{(j')} - \mu\right\|_{\mathrm{op}} \leqslant 2\sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}}.$$

Moreover, among the $K/2$ block means closest to $\widehat{\mu}^{(\widehat{j})}$, at least one of them $\widehat{\mu}^{(j'')}$ satisfies $\|\widehat{\mu}^{(j'')} - \mu\|_{\mathrm{op}} \leqslant \sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}}$ thus we find :

$$\left\|\widehat{\mu} - \mu\right\|_{\mathrm{op}} = \left\|\widehat{\mu}^{(\widehat{j})} - \mu\right\|_{\mathrm{op}} \leqslant \left\|\widehat{\mu}^{(\widehat{j})} - \widehat{\mu}^{(j'')}\right\|_{\mathrm{op}} + \left\|\widehat{\mu}^{(j'')} - \mu\right\|_{\mathrm{op}}$$

$$\leqslant r^{(\widehat{j})}_{K/2} + \sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}} \leqslant 3\sqrt{\frac{2v(A)\log(2(p+q)/\delta'))}{m}}.$$

Finally, let us show that $\sum_{k=1}^{K} \mathbf{1}_{E_k} > K/2$ happens with high probability. Observe that for $j \in \mathcal{K}$ the variables $\mathbf{1}_{\overline{E}_j}$ are stochastically dominated by Bernoulli variables with parameter δ' so that their sum is stochastically dominated by a Binomial random variable $S := \mathrm{Bin}(|\mathcal{K}|, \delta')$. We compute :

$$\begin{aligned}
\mathbb{P}\Big(\sum_{k=1}^{K} \mathbf{1}_{E_k} < K/2\Big) &= \mathbb{P}\Big(\sum_{k=1}^{K} \mathbf{1}_{\overline{E}_k} > K/2\Big) \\
&\leqslant \mathbb{P}\Big(|\mathcal{O}| + \sum_{k\in\mathcal{K}} \mathbf{1}_{\overline{E}_k} > K/2\Big) \\
&\leqslant \mathbb{P}(S - \mathbb{E}S > K/2 - |\mathcal{O}| - \delta'|\mathcal{K}|) \\
&\leqslant \mathbb{P}(S - \mathbb{E}S > K(\varepsilon - 2\delta')/2) \\
&\leqslant \exp(-K(\varepsilon - 2\delta')^2/2)
\end{aligned}$$

where we used that $|\mathcal{O}| \leqslant (1-\varepsilon)K/2$, $|\mathcal{K}| \leqslant K$ and Hoeffding's inequality at the end. Choosing $\varepsilon = 5/6$, $\delta' = 1/4$ and recalling the choice of K and that $m = n/K$ we finish the proof.

Chapter 5

Robust SGD via Gradient Quantile Clipping

This chapter is based on joint work with Stéphane Gaïffas.

Contents

Abstract

We introduce a clipping strategy for Stochastic Gradient Descent (SGD) which uses quantiles of the gradient norm as clipping thresholds. We prove that this new strategy provides a robust and efficient optimization algorithm for smooth objectives (convex or non-convex), that tolerates heavy-tailed samples (including infinite variance) and a fraction of outliers in the data stream akin to Huber contamination. Our mathematical analysis leverages the connection between constant step size SGD and Markov chains and handles the bias introduced by clipping in an original way. For strongly convex objectives, we prove that the iteration converges to a concentrated distribution and derive high probability bounds on the final estimation error. In the non-convex case, we prove that the limit distribution is localized on a neighborhood with low gradient. We propose an implementation of this algorithm using rolling quantiles which leads to a highly efficient optimization procedure with strong robustness properties, as confirmed by our numerical experiments.

5.1 Introduction

Stochastic gradient descent (SGD) [374] is the core optimization algorithm at the origin of most stochastic optimization procedures [235, 103, 222]. SGD and its variants are ubiquitously employed in machine learning in order to train most models [249, 28, 255, 400, 48, 294]. The convergence properties of SGD are therefore subjects of major interest. The first guarantees [338, 165] hold under strong statistical assumptions which require data to follow light-tailed sub-Gaussian distributions and provide error bounds in expectation. With the recent resurgence of interest for robust statistics [197, 114, 258, 363], variants of SGD based on clipping are shown to be robust to heavy-tailed gradients [166, 417], where the gradient samples are only required to have a finite variance. The latter requirement has been further weakened to the existence of a q-th moment for some $q > 1$ in [389, 346]. In this paper, we go further and show that another variant of clipped SGD with proper thresholds is robust both to heavy tails *and* outliers in the data stream.

Robust statistics appeared in the 60s with the pioneering works of Huber, Tukey and others [422, 200, 206, 383, 174]. More recently, the field found new momentum thanks to a series of works about robust scalar mean estimation [73, 6, 216, 287] and the more challenging multidimensional case [195, 76, 289, 315, 92, 107, 268, 117]. These paved the way to the elaboration of a host of robust learning algorithms [191, 363, 258, 274, 356] which have to date overwhelmingly focused on the batch learning setting. We consider the setting of streaming stochastic optimization [47, 49, 307], which raises an additional difficulty coming from the fact that algorithms can see each sample only once and must operate under an $\mathcal{O}(d)$ memory and complexity constraint for d-dimensional optimization problems. A limited number of papers [417, 330, 118] propose theoretical guarantees for robust algorithms learning from streaming data.

This work introduces such an algorithm that learns from data on the fly and is robust both to heavy tails and outliers, with minimal computational overhead and sound theoretical guarantees.

We consider the problem of minimizing a smooth objective

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) := \mathbb{E}_\zeta[\ell(\theta, \zeta)] \tag{5.1.1}$$

using observations $G(\theta, \zeta_t)$ of the unknown gradient $\nabla\mathcal{L}(\theta)$, based on samples $(\zeta_t)_{t \geqslant 0}$ received sequentially that include corruptions with probability $\eta < 1/2$. Formulation (5.1.1) is common to numerous machine learning problems where ℓ is a loss function evaluating the fit of a model

with parameters θ on a sample ζ, the expectation $\mathbb{E}$ is w.r.t the unknown uncorrupted sample distribution.

We introduce quantile-clipped SGD (QC-SGD) which uses the iteration

$$\theta_{t+1} = \theta_t - \alpha_{\theta_t}\beta G(\theta_t, \zeta_t) \quad \text{with} \quad \alpha_{\theta_t} = \min\Big(1, \frac{\tau_{\theta_t}}{\|G(\theta_t, \zeta_t)\|}\Big), \tag{5.1.2}$$

where $\beta > 0$ is a constant step size and α_{θ_t} is the clipping factor with threshold chosen as the p-th quantile $\tau_{\theta_t} = Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ with $\widetilde{G}(\theta_t, \zeta_t)$ an uncorrupted sample of $\nabla\mathcal{L}(\theta_t)$ and $p \in (0, 1)$ (details will follow). Quantiles are a natural choice of clipping threshold which allows to handle heavy tails [382, 38] and corrupted data. For instance, the trimmed mean offers a robust and computationally efficient estimator of a scalar expectation [287]. Since the quantile $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ is non-observable, we introduce a method based on rolling quantiles in Section 5.5 which keeps the procedure $\mathcal{O}(d)$ both memory and complexity-wise.

Contributions. Our main contributions are as follows:

- For small enough η and well-chosen p, we show that, whenever the optimization objective is smooth and strongly convex, QC-SGD converges *geometrically* to a limit distribution such that the deviation around the optimum achieves the *optimal* dependence on η.
- In the non-corrupted case $\eta = 0$ and with a strongly convex objective, we prove that a coordinated choice of β and p ensures that the limit distribution is sub-Gaussian with constant of order $\mathcal{O}(\sqrt{\beta})$. In the corrupted case $\eta > 0$, the limit distribution is sub-exponential.
- For a smooth objective (non-convex) whose gradient satisfies an identifiability condition, we prove that the total variation distance between QC-SGD iterates and its limit distribution vanishes sub-linearly. In this case, the limit distribution is such that the deviation of the objective gradient is optimally controlled in terms of η.
- Finally, we provide experiments to demonstrate that QC-SGD can be easily implemented by estimating $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ with rolling quantiles. In particular, we show that the iteration is indeed robust to heavy tails and corruption on synthetic stochastic optimization tasks.

Our theoretical results are derived thanks to a modelling through Markov chains and hold under an L_q assumption on the gradient distribution with $q > 1$.

Related works. Convergence in distribution of the Markov chain generated by constant step size SGD, relatively to the Wasserstein metric, was first established in [122]. Another geometric convergence result was derived in [446] for non-convex, non-smooth, but quadratically growing objectives, where a convergence statement relatively to a weighted total variation distance is given and a CLT is established. These papers do not consider robustness to heavy tails or outliers. Early works proposed stochastic optimization and parameter estimation algorithms which are robust to a wide class of noise of distributions [297, 359, 360, 365, 409, 83, 82, 331], where asymptotic convergence guarantees are stated for large sample sizes. Initial evidence of the robustness of clipped SGD to heavy tails was given by [450] who obtained results in expectation. Subsequent works derived high-confidence sub-Gaussian performance bounds under a finite variance assumption [166, 417] and later under an L_q assumption [389, 346] with $q > 1$.

Robust versions of Stochastic Mirror Descent (SMD) are introduced in [330, 224]. For a proper choice of the mirror map, SMD is shown to handle infinite variance gradients without any explicit clipping [437]. Finally, [118] studies heavy-tailed and outlier robust streaming estimation algorithms of the expectation and covariance. On this basis, robust algorithms for linear and

logistic regression are derived. However, the involved filtering procedure is hard to implement in practice and no numerical evaluation of the considered approach is proposed.

Agenda. In Section 5.2 we set notations, state the assumptions required by our theoretical results and provide some necessary background on continuous state Markov chains. In Section 5.3, we state our results for strongly convex objectives including geometric ergodicity of QC-SGD (Theorem 5.1), characterizations of the limit distribution and deviation bounds on the final estimate. In Section 5.4, we remove the convexity assumption and obtain a weaker ergodicity result (Theorem 5.2) and characterize the limit distribution in terms of the deviations of the objective gradient. Finally, we present a rolling quantile procedure in Section 5.5 and demonstrate its performance through a few numerical experiments on synthetic data.

5.2 Preliminaries

The model parameter space is $\mathbb{R}^d$ endowed with the Euclidean norm $\|\cdot\|$, $\mathcal{B}(\mathbb{R}^d)$ is the Borel σ-algebra of $\mathbb{R}^d$ and we denote by $\mathcal{M}_1(\mathbb{R}^d)$ the set of probability measures over $\mathbb{R}^d$. We assume throughout the paper that the objective $\mathcal{L}$ is smooth.

Assumption 5.1. *The objective $\mathcal{L}$ is L-Lipschitz-smooth, namely*

$$\mathcal{L}(\theta') \leqslant \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta\rangle + \frac{L}{2}\|\theta - \theta'\|^2$$

with $L < +\infty$ for all $\theta, \theta' \in \mathbb{R}^d$.

The results from Section 5.3 below use the following

Assumption 5.2. *The objective $\mathcal{L}$ is μ-strongly convex, namely*

$$\mathcal{L}(\theta') \geqslant \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta\rangle + \frac{\mu}{2}\|\theta - \theta'\|^2$$

with $\mu > 0$ for all $\theta, \theta' \in \mathbb{R}^d$.

An immediate consequence of Assumption 5.2 is the existence of a unique minimizer $\theta^\star = \arg\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta)$. The next assumption formalizes our corruption model.

Assumption 5.3 (η-corruption)**.** *The gradients $(G(\theta_t, \zeta_t))_{t\geqslant 0}$ used in Iteration (5.1.2) are sampled as $G(\theta_t, \zeta_t) = U_t\breve{G}(\theta_t) + (1 - U_t)\widetilde{G}(\theta_t, \zeta_t)$ where U_t are i.i.d Bernoulli random variables with parameter $\eta < 1/2$, $\breve{G}(\theta_t) \sim \mathcal{D}_{\mathcal{O}}(\theta_t)$ with $\mathcal{D}_{\mathcal{O}}(\theta_t)$ an arbitrary distribution and $\widetilde{G}(\theta_t, \zeta_t) \sim \mathcal{D}_{\mathcal{I}}(\theta_t)$ follows the true gradient distribution and is independent from the past given θ_t.*

Assumption 5.3 is an online analog of the Huber contamination model [199, 200] where corruptions occur with probability η and where the distribution of corrupted samples is not fixed and may depend on the current iterate θ_t. The next assumption requires the true gradient distribution to be unbiased and diffuse.

Assumption 5.4. *For all θ, non-corrupted gradient samples $\widetilde{G}(\theta, \zeta) \sim \mathcal{D}_{\mathcal{I}}(\theta)$ are such that*

$$\widetilde{G}(\theta, \zeta) = \nabla\mathcal{L}(\theta) + \varepsilon_\theta, \tag{5.2.1}$$

where ε_θ is a centered noise $\mathbb{E}[\varepsilon_\theta|\theta] = 0$ with distribution $\delta\nu_{\theta,1} + (1 - \delta)\nu_{\theta,2}$ where $\delta > 0$ and $\nu_{\theta,1}, \nu_{\theta,2}$ are distributions over $\mathbb{R}^d$ such that $\nu_{\theta,1}$ admits a density h_θ w.r.t. the Lebesgue measure satisfying

$$\inf_{\|\omega\|\leqslant R} h_\theta(\omega) > \varkappa(R) > 0$$

for all $R > 0$, where $\varkappa(\cdot)$ is independent of θ.

Assumption 5.4 imposes a weak constraint, since it is satisfied, for example, as soon as the noise ε_θ admits a density w.r.t. Lebesgue's measure. Our last assumption formalizes the requirement of a finite moment for the gradient error.

Assumption 5.5. *There is $q > 1$ such that for $\widetilde{G}(\theta, \zeta) \sim \mathcal{D}_{\mathcal{I}}(\theta)$, we have*

$$\mathbb{E}\big[\|\varepsilon_\theta\|^q \mid \theta\big]^{1/q} = \mathbb{E}\big[\big\|\widetilde{G}(\theta,\zeta) - \nabla\mathcal{L}(\theta)\big\|^q \mid \theta\big]^{1/q} \leqslant A_q\|\theta - \theta^\star\| + B_q \tag{5.2.2}$$

for all $\theta \in \mathbb{R}^d$, where $A_q, B_q > 0$. When $\mathcal{L}$ is not strongly convex, we further assume that $A_q = 0$.

The bound (5.2.2) captures the case of arbitrarily high noise magnitude through the dependence on $\|\theta - \theta^\star\|$. This is consistent with common strongly convex optimization problems such as least squares regression. For non-strongly convex $\mathcal{L}$, we require $A_q = 0$ since $\theta^\star$ may not exist.

Definition 5.1. *If X is a real random variable, we say that X is K-sub-Gaussian for $K > 0$ if*

$$\mathbb{E}\exp(\lambda^2 X^2) \leqslant e^{\lambda^2 K^2} \quad \textit{for} \quad |\lambda| \leqslant 1/K. \tag{5.2.3}$$

We say that X is K-sub-exponential for $K > 0$ if

$$\mathbb{E}\exp(\lambda|X|) \leqslant \exp(\lambda K) \quad \textit{for all} \quad 0 \leqslant \lambda \leqslant 1/K. \tag{5.2.4}$$

The convergence results presented in this paper use the following formalism of continuous state Markov chains. Given a step size $\beta > 0$ and a quantile $p \in (0,1)$, we denote by $P_{\beta,p}$ the Markov transition kernel governing the Markov chain $(\theta_t)_{t\geqslant 0}$ generated by QC-SGD, so that

$$\mathbb{P}(\theta_{t+1} \in A \,|\, \theta_t) = P_{\beta,p}(\theta_t, A)$$

for $t \geqslant 0$ and $A \in \mathcal{B}(\mathbb{R}^d)$. The transition kernel $P_{\beta,p}$ acts on probability distributions $\nu \in \mathcal{M}_1(\mathbb{R}^d)$ through the mapping $\nu \to \nu P_{\beta,p}$ which is defined, for all $A \in \mathcal{B}(\mathbb{R}^d)$, by $\nu P_{\beta,p}(A) = \int_A P_{\beta,p}(\theta, A) d\nu(\theta) = \mathbb{P}(\theta_{t+1} \in A \,|\, \theta_t \sim \nu)$. For $n \geqslant 1$, we similarly define the multi-step transition kernel $P^n_{\beta,p}$ which is such that $P^n_{\beta,p}(\theta_t, A) = \mathbb{P}(\theta_{t+n} \in A \,|\, \theta_t)$ and acts on probability distributions $\nu \in \mathcal{M}_1(\mathbb{R}^d)$ through $\nu P^n_{\beta,p} = (\nu P_{\beta,p})P^{n-1}_{\beta,p}$. Finally, we define the total variation (TV) norm of a signed measure ν as

$$2\|\nu\|_{\mathrm{TV}} = \sup_{f:|f|\leqslant 1} \int f(\theta)\nu(d\theta) = \sup_{A\in\mathcal{B}(\mathbb{R}^d)} \nu(A) - \inf_{A\in\mathcal{B}(\mathbb{R}^d)} \nu(A).$$

In particular, we recover the TV *distance* between $\nu_1, \nu_2 \in \mathcal{M}_1(\mathbb{R}^d)$ as $d_{\mathrm{TV}}(\nu_1, \nu_2) = \|\nu_1 - \nu_2\|_{\mathrm{TV}}$.

5.3 Strongly Convex Objectives

We are ready to state our convergence result for the stochastic optimization of a strongly convex objective using QC-SGD with η-corrupted samples.

Theorem 5.1 (Geometric ergodicity)**.** *Let Assumptions 5.1-5.5 hold and assume there is a quantile $p \in [\eta, 1-\eta]$ such that*

$$\kappa := (1-\eta)p\mu - \eta L - (1-p)^{-\frac{1}{q}} A_q(1 - p(1-\eta)) > 0. \tag{5.3.1}$$

Then, for a step size β satisfying

$$\beta < \frac{1}{4}\frac{\kappa}{\mu^2 + 6L^2 + 16\eta^{-\frac{2}{q}}A_q^2}, \tag{5.3.2}$$

the Markov chain $(\theta_t)_{t\geqslant 0}$ generated by QC-SGD with parameters β and p converges geometrically to a unique invariant measure $\pi_{\beta,p}$: for any initial $\theta_0 \in \mathbb{R}^d$, there is $\rho < 1$ and $M < \infty$ such that after T iterations

$$\left\|\delta_{\theta_0} P_{\beta,p}^T - \pi_{\beta,p}\right\|_{\mathrm{TV}} \leqslant M\rho^T\left(1 + \|\theta_0 - \theta^\star\|^2\right),$$

where δ_{θ_0} is the Dirac measure located at θ_0.

The proof of Theorem 5.1 is given in Section 5.9.1 and relies on the geometric ergodicity result of [313, Chapter 15] for Markov chains with a geometric drift property. A similar result for quadratically growing objectives was established by [446] and convergence w.r.t. Wasserstein's metric was shown in [122] assuming uniform gradient co-coercivity. However, robustness was not considered in these works. The restriction $p \in [\eta, 1-\eta]$ comes from the consideration that other quantiles are not estimable in the event of η-corruption. Condition (5.3.1) is best interpreted for the choice $p = 1 - \eta$ in which case it translates into $\eta^{1-1/q} \leqslant \mathcal{O}(\mu/(L + A_q))$ implying that it is verified for η small enough within a limit fixed by the problem conditioning. A similar condition with $q = 2$ appears in [118, Theorem E.9] which uses a finite variance assumption.

The constants M and ρ controlling the geometric convergence speed in Theorem 5.1 depend on the parameters β, p and the initial θ_0. Among choices fulfilling the convergence conditions, it is straightforward that greater step size β and θ_0 closer to $\theta^\star$ lead to faster convergence. However, the dependence in p is more intricate and should be evaluated through the resulting value of κ. We provide a more detailed discussion about the value of ρ in Section 5.8.

The choice $p = 1 - \eta$ appears to be ideal since it leads to optimal deviation of the invariant distribution around the optimum $\theta^\star$ which is the essence of our next statement.

Proposition 5.1. *Assume the same as in Theorem 5.1 and condition* (5.3.1) *with the choice $p = 1 - \eta$. For step size β satisfying* (5.3.2)*, $q \geqslant 2$, and additionally*:

$$\beta \leqslant \eta^{2-2/q}/\kappa, \tag{5.3.3}$$

for $\theta \sim \pi_{\beta,1-\eta}$, we have the following upper bound:

$$\mathbb{E}\|\theta - \theta^\star\|^2 \leqslant \left(\frac{6\eta^{1-1/q}B_q}{\kappa}\right)^2.$$

Proposition 5.1 is proven in Section 5.9.1. An analogous result holds for $q \in (1, 2)$ but requires a different proof and can be found in Section 5.9.1. Proposition 5.1 may be compared to [446, Theorem 3.1] which shows that the asymptotic estimation error can be reduced arbitrarily using a small step size. However, this is impossible in our case since we consider corrupted gradients. The performance of Proposition 5.1 is best discussed in the specific context of linear regression where gradients are given as $G(\theta, (X, Y)) = X(X^\top\theta - Y)$ for samples $X, Y \in \mathbb{R}^d \times \mathbb{R}$ such that $Y = X^\top\theta^\star + \epsilon$ with ϵ a centered noise. In this case, a finite moment of order k for the data implies order $k/2$ for the gradient corresponding to an $\eta^{1-2/k}$ rate in Proposition 5.1. Since Assumption 5.5 does not include independence of the noise ϵ from X, this corresponds to the negatively correlated moments assumption of [18] being unsatisfied. Consequently, Proposition 5.1 is information-theoretically optimal in η based on [18, Corollary 4.2]. Nonetheless, the poor dimension dependence through B_q may still be improved. If the gradient is sub-Gaussian with constant K, we would have $B_q \lesssim K\sqrt{q}$ for $q \geqslant 1$ (see [432] for a reference), in which case,

the choice $q = \log(1/\eta)$ recovers the optimal rate in $\eta\sqrt{\log(1/\eta)}$ for the Gaussian case.

We now turn to showing strong concentration properties for the invariant distribution $\pi_{\beta,p}$. For this purpose, we restrict the optimization to a bounded and convex set $\Theta \subset \mathbb{R}^d$ and replace Iteration (5.1.2) by the projected iteration

$$\theta_{t+1} = \Pi_\Theta\big(\theta_t - \alpha_{\theta_t}\beta G(\theta_t, \zeta_t)\big), \tag{5.3.4}$$

where Π_Θ is the projection onto Θ. Assuming that the latter contains the optimum $\theta^\star \in \Theta$, one can check that the previous results continue to hold thanks to the inequality

$$\|\Pi_\Theta(\theta) - \theta^\star\| = \|\Pi_\Theta(\theta) - \Pi_\Theta(\theta^\star)\| \leqslant \|\theta - \theta^\star\|,$$

which results from the convexity of Θ. The restriction of the optimization to a bounded set allows us to uniformly bound the clipping threshold τ_θ, which is indispensable for the following result.

Proposition 5.2. *In the setting of Theorem 5.1, consider projected QC-SGD* (5.3.4) *and let* $\overline{\tau} = \sup_{\theta\in\Theta} \tau_\theta, D = \operatorname{diam}(\Theta)$ *the diameter of* Θ *and* $\overline{B}_q = A_q D + B_q$.

- *Consider the non-corrupted case* $\eta = 0$ *and set the quantile* p *such that* $p \geqslant 1 - (\beta\mu)^{\frac{q}{2(q-1)}}$. *Then, for* $\theta \sim \pi_{\beta,p}$, *the variable* $\|\theta - \theta^\star\|$ *is sub-Gaussian in the sense of Definition 5.1 with constant*
$$K = 4\sqrt{\frac{2\beta(\overline{B}_q^2 + \overline{\tau}^2)}{p\mu}}.$$
- *Consider the corrupted case* $\eta > 0$, *and set the quantile* $p \in [\eta, 1-\eta]$ *such that Inequality* (5.3.1) *holds. Then, for* $\theta \sim \pi_{\beta,p}$, *the variable* $\|\theta - \theta^\star\|$ *is sub-exponential in the sense of Definition 5.1 with constant*
$$K = \frac{7\overline{\tau} + (1-p)^{1-1/q}\overline{B}_q}{p\mu}.$$

The proof can be found in Section 5.9.1. The strong concentration properties given by Proposition 5.2 for the invariant distribution appear to be new. Still, the previous result remains asymptotic in nature. High confidence deviation bounds for an iterate θ_t can be derived by leveraging the convergence in Total Variation distance given by Theorem 5.1 leading to the following result.

Corollary 5.1. *In the setting of Proposition 5.2, in the absence of corruption* $\eta = 0$, *after* T *iterations, for* $\delta > 0$, *we have*

$$\mathbb{P}\bigg(\big\|\theta_T - \theta^\star\big\| > 4\sqrt{\overline{B}_q^2 + \overline{\tau}^2}\sqrt{\frac{2\beta\log(e/\delta)}{p\mu}}\bigg) \leqslant \delta + \rho^T M\big(1 + \|\theta_0 - \theta^\star\|^2\big).$$

Choosing a smaller step size β in Corollary 5.1 allows to improve the deviation bound. However, this comes at the cost of weaker confidence because of slower convergence due to a greater ρ. See Section 5.8 for a discussion including a possible compromise. Corollary 5.1 may be compared to the results of [166, 417, 389, 346] which correspond to $\beta \approx 1/T$ and have a similar dependence on the dimension through the gradient variance. Although their approach is also based on gradient clipping, they use different thresholds and proof methods. In the presence of corruption, the invariant distribution is not sub-Gaussian. This can be seen by considering the

following toy Markov chain:

$$X_{t+1} = \begin{cases} \alpha X_t + \xi & \text{w.p.} \quad 1-\eta \\ X_t + \tau & \text{w.p.} \quad \eta \end{cases}$$

where $\alpha < 1, \tau > 0$ are constants and ξ is a positive random noise. Using similar methods to the proof of Theorem 5.1, one can show that $(X_t)_{t \geqslant 0}$ converges (for any initial X_0) to an invariant distribution whose moments can be shown to grow linearly, indicating a sub-exponential distribution and excluding a sub-Gaussian one. We provide additional details for the underlying argument in Section 5.9.1. For the corrupted case, the sub-exponential property stated in Proposition 5.2 holds with a constant K of order $\overline{\tau}/\mu$, which is not satisfactory and leaves little room for improvement due to the inevitable bias introduced by corruption. Therefore, we propose the following procedure in order to obtain a high confidence estimate, similarly to Corollary 5.1.

Algorithm 2 Aggregation of cycling iterates

Input: Step size $\beta > 0$, quantile index $p \in (0,1)$, initial parameter $\theta_0 \in \Theta$, horizon T and number of concurrent iterates $N \geqslant 1$.
Optimize multiple parameters $\theta_t^{(1)}, \dots, \theta_t^{(N)}$ starting from a common $\theta_0 = \theta_0^{(n)}$ for $n \in [\![N]\!] =: \{1, \dots, N\}$ and T steps $t = 0, \dots, T$ using the following cycling iteration:

$$\theta_{t+1}^{(n)} = \begin{cases} \theta_t^{(n)} - \alpha_{\theta_t^{(n)}} \beta G(\theta_t^{(n)}, \zeta_t) & \text{if} \quad t \equiv n-1 \bmod N, \\ \theta_t^{(n)} & \text{otherwise.} \end{cases} \tag{5.3.5}$$

Compute $r_{ij} = \|\theta_T^{(i)} - \theta_T^{(j)}\|$ for $i, j \in [\![N]\!]$
For $j \in [\![N]\!]$, let $r^{(j)} \in \mathbb{R}_+^N$ be the vector $r_{j,:} := [r_{j,1}, \dots r_{j,N}]$ sorted in non decreasing order.
Define the aggregated estimator as

$$\widehat{\theta} = \theta_T^{(\widehat{i})} \quad \text{with} \quad \widehat{i} = \operatorname*{arg\,min}_{i \in [\![N]\!]} r_{N/2}^{(i)}.$$

return $\widehat{\theta}$

Algorithm 2 uses ideas from [197] (see also [315, 224]) and combines a collection of *weak* estimators (only satisfying L_2 bounds) into a strong one with sub-exponential deviation. The aggregated estimator $\widehat{\theta}$ satisfies the high probability bound given in the next result.

Corollary 5.2. *Assume the same as in Theorem 5.1 and Proposition 5.1. Consider $\widehat{\theta}$ given by Algorithm 2, with the assumption that the gradient sample sets used for each $\left(\theta_T^{(n)}\right)_{n \in [\![N]\!]}$ in Equation (5.3.5) are independent. For $\delta > 0$, if $N \geqslant 16 \log(1/\delta)$ and T satisfies*

$$T \geqslant N \log(15M(1 + \|\theta_0 - \theta^\star\|^2)) / \log(1/\rho),$$

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

$$\left\|\widehat{\theta} - \theta^\star\right\| \leqslant \frac{27 \eta^{1 - \frac{1}{q}} \overline{B}_q}{\kappa}. \tag{5.3.6}$$

We obtain a high confidence version of the bound in expectation previously stated in Proposition 5.1. As argued before, the above bound depends optimally on η. Similar bounds to (5.3.6) are obtained for $q = 2$ in [118] for streaming mean estimation, linear and logistic regression.

Their results enjoy better dimension dependence but are less general than ours. In addition, the implementation of the associated algorithm is not straightforward whereas our method is quite easy to use (see Section 5.5).

5.4 Smooth Objectives

In this section, we drop Assumption 5.2 and consider the optimization of possibly non-convex objectives. Consequently, the existence of a unique optimum $\theta^\star$ and the quadratic growth of the objective are no longer guaranteed. This motivates us to use a uniform version of Assumption 5.5 with $A_q = 0$ since the gradient is no longer assumed coercive and its deviation moments can be taken as bounded. In this context, we obtain the following weaker (compared to Theorem 5.1) ergodicity result for QC-SGD.

Theorem 5.2 (Ergodicity)**.** *Let Assumptions 5.1, 5.3, 5.4 and 5.5 hold with $A_q = 0$ (uniformly bounded moments) and positive objective $\mathcal{L}$. Let $(\theta_t)_{t\geqslant 0}$ be the Markov chain generated by QC-SGD with step size β and quantile $p \in [\eta, 1-\eta]$. Assume that p and β are such that $3p(1-\eta)/4 > L\beta + \eta$ and that the subset of $\mathbb{R}^d$ given by*

$$\Big\{\theta \in \mathbb{R}^d : \frac{1}{2}\big\|\nabla\mathcal{L}(\theta)\big\|^2 \leqslant \frac{B_q^2\big((1-p)^{-\frac{2}{q}}(L\beta + 2\eta^2) + 2\eta^{2-\frac{2}{q}}\big)}{p(1-\eta)(3p(1-\eta)/4 - L\beta - \eta)}\Big\} \tag{5.4.1}$$

is bounded. Then, for any initial $\theta_0 \in \mathbb{R}^d$, there exists $M < +\infty$ such that after T iterations

$$\big\|\delta_{\theta_0} P_{\beta,p}^T - \pi_{\beta,p}\big\|_{\mathrm{TV}} \leqslant \frac{M}{T}, \tag{5.4.2}$$

where $\pi_{\beta,p}$ is a unique invariant measure and where δ_{θ_0} is the Dirac measure located at θ_0.

The proof is given in Section 5.9.1 and uses ergodicity results from [313, Chapter 13]. Theorem 5.2 provides convergence conditions for an SGD Markov chain on a smooth objective in a robust setting. We are unaware of anterior results of this kind in the literature. Condition (5.4.1) requires that the true gradient exceeds the estimation error at least outside of a bounded set. If this does not hold, the gradient would be dominated by the estimation error, leaving no hope for the iteration to converge. Observe that, for no corruption ($\eta = 0$), the condition is always fulfilled for some β and p. Note also that without strong convexity (Assumption 5.2), convergence occurs at a slower sublinear rate which is consistent with the optimization rate expected for a smooth objective (see [58, Theorem 3.3]).

As previously, we complement Theorem 5.1 with a characterization of the invariant distribution.

Proposition 5.3. *Under the conditions of Theorem 5.2, assume that the choice $p = 1 - \eta$ is such that the set* (5.4.1) *is bounded. For step size $\beta \leqslant \eta^2/L$, the stationary measure $\theta \sim \pi_{\beta,1-\eta}$ satisfies*

$$\mathbb{E}\big\|\nabla\mathcal{L}(\theta)\big\|^2 \leqslant \frac{5\eta^{2-\frac{2}{q}} B_q^2}{p(1-\eta)\big(3p(1-\eta)/4 - L\beta - \eta\big)}. \tag{5.4.3}$$

The statement of Proposition 5.3 is clearly less informative than Propositions 5.1 and 5.2 since it only pertains to the gradient rather than, for example, the excess risk. This is due to the weaker assumptions that do not allow to relate these quantities. Still, the purpose remains to find a critical point and is achieved up to $\mathcal{O}(\eta^{1-1/q})$ precision according to this result. Due to corruption, the estimation error on the gradient cannot be reduced beyond $\Omega(\eta^{1-1/q})$ [362, 194,

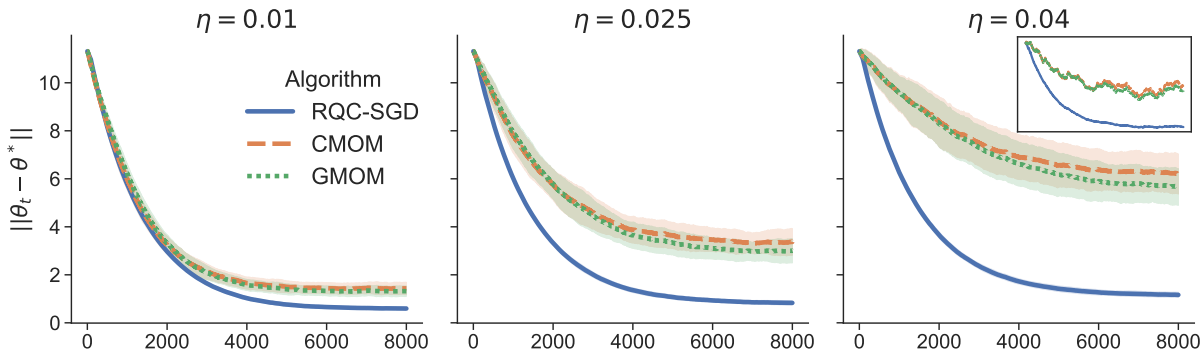

Figure 5.1: Evolution of $\|\theta_t - \theta^\star\|$ (y-axis) against iteration t (x-axis) for the expectation estimation task, averaged over 100 runs at different corruption levels η (bands widths correspond to the standard deviation of the 100 runs). For $\eta = 0.04$, the evolution on a single run is also displayed. We observe good performance for RQC-SGD for increasing η while CMOM and GMOM are more sensitive.

116]. Therefore, one may draw a parallel with a corrupted mean estimation task, in which case, the previous rate is, in fact, information-theoretically optimal.

5.5 Implementation and Numerical Experiments

The use of the generally unknown quantile $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ in QC-SGD constitutes the main obstacle to its implementation. For strongly convex objectives, one may use a proxy such as $a\|\theta_t - \theta_{\text{ref}}\| + b$ with positive a, b and $\theta_{\text{ref}} \in \mathbb{R}^d$ an approximation of $\theta^\star$ serving as reference point. This choice is consistent with Assumptions 5.1 and 5.5, see Lemma 5.2 in Section 5.9. In the non-strongly convex case, a constant threshold can be used since the gradient is a priori uniformly bounded, implying the same for the quantiles of its deviations. In practice, we propose a simpler and more direct approach: we use a rolling quantile procedure, described in Algorithm 3. The latter stores the values $(\|G(\theta_{t-j}, \zeta_{t-j})\|)_{1 \leqslant j \leqslant S}$ in a buffer of size $S \in \mathbb{N}^*$ and replaces $Q_p(\|\widetilde{G}(\theta_t, \zeta_t)\|)$ in QC-SGD by an estimate $\widehat{Q}_p$ which is the $\lfloor pS \rfloor$-th order statistic in the buffer. Note that only the norms of previous gradients are stored in the buffer, limiting the memory overhead to $\mathcal{O}(S)$. The computational cost of $\widehat{Q}_p$ can also be kept to $\mathcal{O}(S)$ per iteration thanks to a bookkeeping procedure (see Section 5.7).

We implement this procedure for a few tasks and compare its performance with relevant baselines. We do not include a comparison with [118] whose procedure has no implementation we are aware of and is difficult to use in practice due to its dependence on a number of unknown constants. All our experiments consider an infinite horizon, dimension $d = 128$, and a constant step size for all methods.

Expectation estimation. We estimate the expectation of a random vector X by minimizing the objective $\mathcal{L}(\theta) = \frac{1}{2}\|\theta - \theta^\star\|^2$ with $\theta^\star = \mathbb{E}[X]$ using a stream of both corrupted and heavy-tailed samples, see Section 5.7 for details. We run RQC-SGD (Algorithm 3) and compare it to an online version of geometric and coordinate-wise Median-Of-Means (GMOM and CMOM) [70, 71] which use block sample means to minimize an L_1 objective (see Section 5.7). Although these estimators are a priori not robust to η-corruption, we ensure that their estimates are meaningful by limiting η to 4% and using blocks of 10 samples. Thus, blocks are corrupted with probability $< 1/2$ so that the majority contains only true samples. Figure 5.1 displays the evolution of

$\|\theta_t - \theta^\star\|$ for each method averaged over 100 runs for increasing η and constant step size. We also display a single run for $\eta = 0.04$. We observe that RQC-SGD is only weakly affected by the increasing corruption whereas the performance of GMOM and CMOM quickly degrades with η, leading to unstable estimates.

Linear regression. We consider least-squares linear regression and compare RQC-SGD with Huber's estimator [204] and clipped SGD (designated as CClip(λ)) with three clipping levels $\lambda\sigma_{\max}\sqrt{d}$ for $\lambda \in \{0.8, 1.0, 1.2\}$ where $\sigma_{\max}$ is a fixed data scaling factor. These thresholds provide a rough estimate of the gradient norm. We generate covariates X and labels Y both heavy-tailed and corrupted. Corruption in the data stream is generated according to Assumption 5.3 with outliers represented either by aberrant values or *fake* samples $Y = X^\top\theta_{\text{fake}} + \epsilon$ using a false parameter θ_{fake}, see Section 5.7 for further details on data generation and fine tuning of the Huber parameter. All methods are run with constant step size and averaged results over 100 runs are displayed on Figure 5.2 (top row).

Algorithm 3 Quantile clipped SGD

Input: Step size $\beta > 0$, quantile index $p \in (0, 1)$, initial parameter $\theta_0 \in \mathbb{R}^d$, positive τ_{unif}, buffer B of size S and horizon T.
Fill B with $S - 1$ values equal to τ_{unif}.
for $t = 0 \dots T - 1$ **do**
  Draw a sample $G(\theta_t, \zeta_t)$ and add $\|G(\theta_t, \zeta_t)\|$ to B.
  $\widehat{Q}_p \leftarrow \lfloor pS \rfloor$ rank element of B.
  $\theta_{t+1} \leftarrow \theta_t - \beta\text{clip}\big(G(\theta_t, \zeta_t), \widehat{Q}_p\big)$
  Delete the oldest value in B.
end for
return θ_T

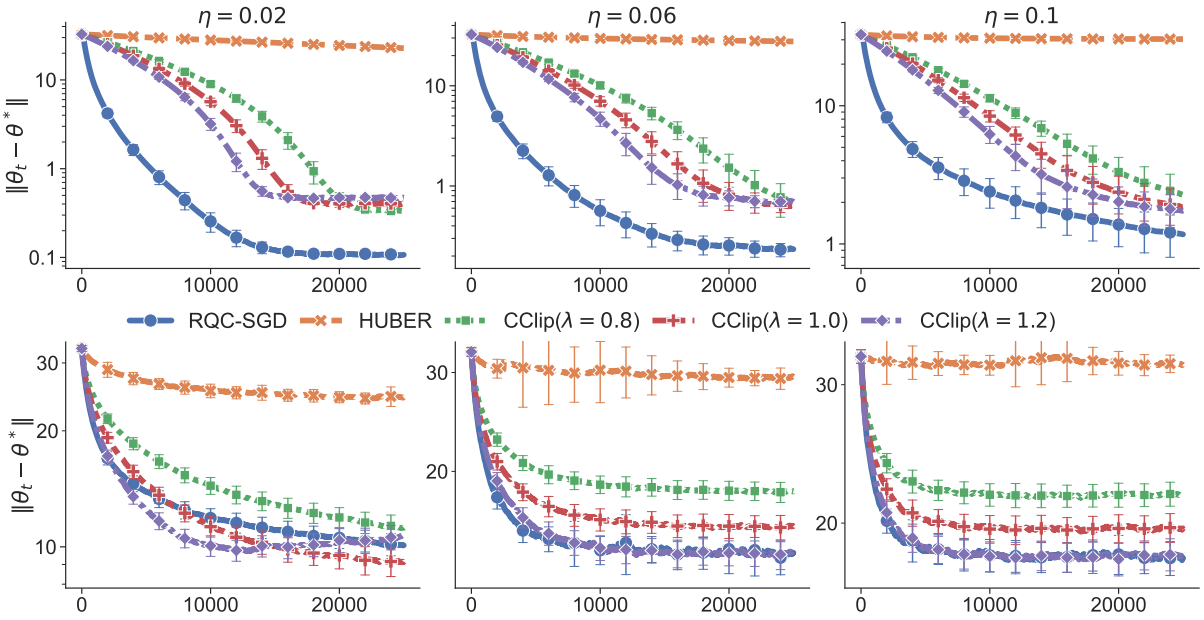

Figure 5.2: Evolution of $\|\theta_t - \theta^\star\|$ on the tasks of linear regression (top row) and logistic regression (bottom row) averaged over 100 runs at increasing corruption levels (error bars represent half the standard deviation). Estimators based on Huber's loss are strongly affected by data corruption. SGD with constant clipping thresholds is robust but slow to converge for linear regression and requires tuning for better final precision. RQC-SGD combines fast convergence with good final precision thanks to its adaptive clipping strategy.

As anticipated, Huber's loss function is not robust to corrupted covariates. In contrast, using gradient clipping allows convergence to meaningful estimates. Although this holds true for a constant threshold, Figure 5.2 shows it may considerably slow the convergence if started away

from the optimum. In addition, the clipping level also affects the final estimation precision and requires tuning. Both of the previous issues are well addressed by RQC-SGD whose adaptive clipping level allows fast progress of the optimization and accurate convergence towards a small neighborhood of the optimum.

Logistic regression. Finally, we test the same methods on logistic regression. Huber's baseline is represented by the modified Huber loss (also known as quadratic SVM [453]). We generate data similarly to the previous task except for the labels which follow $Y \sim \text{Bernoulli}(\sigma(X^\top \theta^\star))$ with σ the sigmoid function. Corrupted labels are either uninformative, flipped or obtained with a fake θ_{fake} (see details in Section 5.7). Results are displayed on the bottom row of Figure 5.2.

As previously, Huber's estimator performs poorly with corruption. However, constant clipping appears to be better suited when the gradient is bounded, so that the optimization is less affected by its underestimation. We observe, nonetheless, that a higher clipping level may lead to poor convergence properties, even at a low corruption rate. Note also that the constant levels we use are based on prior knowledge about the data distribution and would have to be fine tuned in practice. Meanwhile, the latter issue is well addressed by quantile clipping. Finally, notice that no algorithm truly approaches the true solution for this task. This reflects the difficulty of improving upon Proposition 5.3 which only states convergence to a neighborhood where the objective gradient is comparable to the estimation error in magnitude.

5.6 Conclusion

We introduced a new clipping strategy for SGD and proved that it defines a stochastic optimization procedure which is robust to both heavy tails and outliers in the data stream. We also provided an efficient rolling quantile procedure to implement it and demonstrated its performance through numerical experiments on synthetic data. Future research directions include improving the dimension dependence in our bounds, possibly by using sample rejection rules or by considering stochastic mirror descent [339, 24] clipped with respect to a non Euclidean norm. This may also procure robustness to higher corruption rates. Another interesting research track is the precise quantification of the geometric convergence speed of the Markov chain generated by constant step size SGD on a strongly convex objective.

5.7 Experimental details

As previously mentioned, the dimension is set to $d = 128$ in all our experiments. We also set $\sigma_{\min} = 1$ and $\sigma_{\max} = 5$ as minimum and maximum scaling factors. For all tasks and algorithms, the optimization starts from $\theta_0 = 0$.

Bookkeeping in RQC-SGD The buffer in Algorithm 3 stores values in sorted order along with their "ages". The most recent and oldest values have ages 0 and $S-1$ respectively. At each iteration, a new gradient is received, all ages are incremented and the oldest value is replaced by the new one with age 0. The latter is then sorted using one iteration of insertion sort. The estimate $\widehat{Q}_p$ is retrieved at each iteration as the value at position $\lfloor pS \rfloor$.

5.7.1 Mean estimation

Data generation We compute a matrix $\Sigma = (AA^\top + A^\top A)/2$ where $A \in \mathbb{R}^{d\times d}$ is a random matrix with i.i.d centered Gaussian entries with variance $1/d$ sampled once and for all. We generate true samples as $X = \mathbf{1} + \Sigma V$ where V is a vector of i.i.d symmetrized Pareto random variables with parameter 2 and $\mathbf{1} \in \mathbb{R}^d$ denotes the vector with all entries equal to 1.

We draw corrupted samples as $\check{X} = 10\check{V} - 100 \times \mathbf{1}$ where $\check{V}$ is a vector of i.i.d symmetrized Pareto variables with parameter 1.5. We use step size $\beta = 10^{-3}$.

GMOM and CMOM The geometric and coordinatewise Median-Of-Means estimators (GMOM and CMOM) optimize the following objectives respectively:

$$\mathbb{E}\|\theta - \overline{X}_{N_b}\|_2 \quad \text{and} \quad \mathbb{E}\left\|\theta - \overline{X}_{N_b}\right\|_1,$$

where $\overline{X}_{N_b}$ is the average of N_b independent copies of X. The block size is set to $N_b = 10$ in the whole experiment. The above objectives are optimized by computing samples of $\overline{X}_{N_b}$ in a streaming fashion so that one step is made for each N_b samples. In order to compensate for this inefficiency we multiply the step size by N_b for both GMOM and CMOM. For GMOM, we additionally multiply the step size by $\sqrt{d}$ in order to compensate the normalization included in the gradient formula.

RQC-SGD For mean estimation, we implement RQC-SGD (Algorithm 3) with buffer size $S = 100, p = 0.2$ and $\tau_{\text{unif}} = 10$.

5.7.2 Linear regression

Data generation We choose the true parameter $\theta^\star$ by independently sampling its coordinate uniformly in the interval $[-5, +5]$. The true covariates are sampled as $X = \Sigma V$ where Σ is a diagonal matrix with entries sampled uniformly in the interval $[\sigma_{\min}, \sigma_{\max}]$ (once and for all) and V is a vector of i.i.d symmetrized Pareto random variables with parameter 2. The labels are sampled as $Y = X^\top\theta^\star + \epsilon$ where ϵ is a symmetrized Pareto random variable with parameter 2.

The corrupted samples are obtained according to one of the following possibilities with equal probability:

- $X = 1000(\max_i \Sigma_{ii})v + W$ where v is a fixed unit vector and W is a standard Gaussian vector and $Y \sim \text{Bernoulli}(1/2)$.
- $X = 1000(\max_i \Sigma_{ii})V$ with V a unit norm random vector with uniform distribution and $Y = 1000(Z + U)$ where Z is a random sign and U is uniform over $[-1/5, 1/5]$.

- $X = 10V$ with V a random vector with i.i.d entries following a symmetrized Log-normal distribution and $Y = X^\top \theta_{\text{fake}} + \epsilon$ with θ_{fake} a fake parameter drawn similarly to $\theta^\star$ once and for all and ϵ a standard Gaussian variable.

We use step size $\beta = 10^{-3}$.

Huber parameter In order to tune the parameter δ of Huber's loss function, we proceed as follows:

– For each corruption level η, we consider 10 candidate values $\delta_j = 10^{j/2-5}$ for $0 \leqslant j < 10$.

– For each candidate δ_j, we train 250 estimators $\left(\widehat{\theta}^{(i)}_{\delta_j}\right)_{i \in [\![250]\!]}$ using 1000 samples each.

– We choose $\widehat{j}$ for which the average $\frac{1}{250} \sum_{i \in [\![250]\!]} \left\| \widehat{\theta}^{(i)}_{\delta_j} - \theta^\star \right\|$ is minimal and use $\widehat{\delta} = \delta_{\widehat{j}}$ as parameter.

RQC-SGD For linear regression, we run RQC-SGD with buffer size $S = 100$ and $\tau_{\text{unif}} = 10$. The quantile value was set to $p = 0.1$ for $\eta \in \{0.02, 0.06\}$ and $p = 0.05$ for $\eta = 0.1$.

5.7.3 Logistic regression

Data generation The true parameter $\theta^\star$ and covariates X are chosen similarly to linear regression. Given X, the label Y is set to $+1$ with probability $\sigma(X^\top \theta^\star)$ where σ is the sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$ and to -1 otherwise.

The corrupted covariates are determined similarly to linear regression while the labels are set as follows in each respective case:

- Y is set to $+1$ or -1 with equal probability.
- $Y = -\text{sign}(X^\top \theta^\star)$.
- $Y = \text{sign}(X^\top \theta_{\text{fake}})$ with θ_{fake} a fake parameter drawn similarly to $\theta^\star$ once and for all.

We use step size $\beta = 6 \times 10^{-3}$.

Huber parameter The same procedure is used to tune the parameter of the modified Huber loss as for linear regression.

RQC-SGD For logistic regression, we run RQC-SGD with buffer size $S = 100$ and $\tau_{\text{unif}} = 10$. The quantile value was set to $p = 1 - \eta - 0.1$ for $\eta = 0.02$ and $p = 1 - \eta - 0.05$ otherwise.

5.8 Geometric convergence speed and relation to step size

The geometric Markov chain convergence stated in Theorem 5.1 occurs at a speed determined by the contraction factor ρ which mainly depends on the step size β and quantile p defining the iteration. Therefore, an explicit formulation of this dependency is necessary to precisely quantify the convergence speed. This question is lightly touched upon in [446] whose Proposition 2.1 is an analogous SGD ergodicity result. Like Theorem 5.1, the latter relies on the Markov chain theory presented in [313]. It is argued in [446] that a vanishing step size $\beta \to 0$ causes ρ to be close to one, leading to slow convergence but with smaller bias. However, these considerations remain asymptotic and do not quite address the convergence speed issue.

More generally, the precise estimation of the factor ρ goes back to the evaluation of the convergence speed of a Markov chain satisfying a geometric drift property. Near optimal results exist for chains with particular properties such as stochastic order [291, 377], reversibility [215] or special assumptions on the renewal distribution [29]. Unfortunately, such properties do not hold for SGD. Let $(\theta_t)_{t\geqslant 0}$ be a Markov chain satisfying the drift property:

$$\Delta V(\theta) \leqslant \begin{cases} (1-\lambda)V(\theta) & \text{for} \quad \theta \notin \mathcal{C} \\ b & \text{for} \quad \theta \in \mathcal{C} \end{cases}$$

with $\lambda \in (0,1), b < +\infty, V$ a real function such that $V(\theta) \geqslant 1$ for all θ and $\mathcal{C}$ a (bounded) small set (see [313, Chapter 5]). Then, based on the available literature [23, 26], $(\theta_t)_{t\geqslant 0}$ converges as in Theorem 5.1 with $\rho \approx 1-\lambda^3$, the latter estimation being unimprovable without further information on $(\theta_t)_{t\geqslant 0}$ (see the discussion following Theorem 3.2 in [23]). For the specific setting of Theorem 5.1 (and more generally for SGD by setting $p=1$), this only yields an excessively pessimistic estimate

$$\rho \approx 1-(p\beta\mu)^3 \tag{5.8.1}$$

whereas it is reasonable to conjecture that $\rho \approx 1-p\beta\mu$. The suboptimality of (5.8.1) is felt in the uncorrupted case in Proposition 5.2 and Corollary 5.1 where one is tempted to set β of order $1/T$, with T the horizon, reducing the bias to $\mathcal{O}(1/\sqrt{T})$. However, this results in an unacceptable sample cost of order T^3 before convergence occurs. On the other hand, assuming the estimate $\rho \approx 1-p\beta\mu$ holds, using a step size of order $\log(T)/T$ allows to combine fast convergence and near optimal statistical performance. Finally, note that in the corrupted case, the optimal statistical rate is $\mathcal{O}(\eta^{1-1/q})$ so that striking such a compromise is unnecessary.

5.9 Proofs

5.9.1 Preliminary lemmas

Lemma 5.1. *Grant Assumptions 5.1 and 5.2. For any $\theta, \theta' \in \mathbb{R}^d$ and $\beta \leqslant \frac{2}{\mu+L}$ we have :*

$$\left\|\theta - \beta\nabla\mathcal{L}(\theta) - (\theta' - \beta\nabla\mathcal{L}(\theta'))\right\|^2 \leqslant (1-\beta\mu)^2\|\theta-\theta'\|^2 \tag{5.9.1}$$

Proof. For $\beta \leqslant \frac{2}{\mu+L}$, we have:

$$\begin{aligned}
&\left\|\theta - \beta\nabla\mathcal{L}(\theta) - (\theta' - \beta\nabla\mathcal{L}(\theta'))\right\|^2 \\
&\qquad = \|\theta-\theta'\|^2 - 2\beta\langle\theta-\theta', \nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\rangle + \beta^2\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\|^2 \\
&\qquad \leqslant (1-\beta^2\mu L)\|\theta-\theta'\|^2 - \beta(2-\beta(\mu+L))\langle\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta-\theta'\rangle) \\
&\qquad \leqslant (1-\beta^2\mu L)\|\theta-\theta'\|^2 - \beta(2-\beta(\mu+L))\mu\|\theta-\theta'\|^2 \\
&\qquad = (1-\beta^2\mu L - 2\beta\mu + \beta^2\mu(\mu+L))\|\theta-\theta'\|^2 \\
&\qquad = (1-\beta\mu)^2\|\theta-\theta'\|^2,
\end{aligned}$$

where we used the inequalities :

$$\left\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\right\|^2 \leqslant (\mu+L)\langle\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta-\theta'\rangle - \mu L\|\theta-\theta'\|^2 \tag{5.9.2}$$

$$\mu\|\theta-\theta'\|^2 \leqslant \langle\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta-\theta'\rangle, \tag{5.9.3}$$

valid for all θ, θ'. Inequality (5.9.2) is stated, for example, in [342, Theoerem 2.1.12] (see also

[58, Lemma 3.11] and (5.9.3) is just a characterization of strong convexity (see for instance [342, Theorem 2.1.9]). □

In the sequel we will write gradient samples $G(\theta,\zeta)$ and $\widetilde{G}(\theta,\zeta)$ simply as $G(\theta)$ and $\widetilde{G}(\theta)$ respectively in order to lighten notation. The following lemma will be needed in the proof of Theorem 5.1.

Lemma 5.2. *Let Assumptions 5.4 and 5.5 hold. Let $\theta \in \mathbb{R}^d$ be fixed and let $\widetilde{G}(\theta) \sim \mathcal{D}_{\mathcal{I}}(\theta)$ be a non corrupted gradient sample. Choosing the clipping threshold as $\tau_\theta = Q_p(\|\widetilde{G}(\theta)\|)$ for some $p \in (0,1)$ and denoting $\alpha_\theta = \min\Big(1, \frac{\tau_\theta}{\|G(\theta)\|}\Big)$ the clipping factor and its average $\overline{\alpha}_\theta = \mathbb{E}\big[\alpha_\theta | \theta, G(\theta) = \widetilde{G}(\theta) \sim \mathcal{D}_{\mathcal{I}}(\theta)\big]$ we have*:

$$\big\|\mathbb{E}[\alpha_\theta \widetilde{G}(\theta)] - \overline{\alpha}_\theta \nabla\mathcal{L}(\theta)\big\| \leqslant (1-p)^{1-1/q}\big(A_q\|\theta-\theta^\star\| + B_q\big), \tag{5.9.4}$$

$$\begin{aligned}\tau_\theta &\leqslant \big\|\nabla\mathcal{L}(\theta)\big\| + Q_p(\|\varepsilon_\theta\|)\\ &\leqslant \big\|\nabla\mathcal{L}(\theta)\big\| + (1-p)^{-1/q}\big(A_q\|\theta-\theta^\star\| + B_q\big).\end{aligned} \tag{5.9.5}$$

If Assumption 5.5 holds with $q \geqslant 2$ then we also have

$$\mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 \leqslant \big(A_q\|\theta-\theta^\star\| + B_q\big)^2 + 5(1-p)\tau_\theta^2. \tag{5.9.6}$$

Proof. We condition on the event that the sample $G(\theta)$ is not corrupted i.e. $G(\theta) = \widetilde{G}(\theta) \sim \mathcal{D}_{\mathcal{I}}(\theta)$. Noticing that $\mathbf{1}_{\|\widetilde{G}(\theta)\|\leqslant\tau_\theta} = 1 - \mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}$ and using the equality $\mathbb{E}[\widetilde{G}(\theta)] = \nabla\mathcal{L}(\theta)$, we find :

$$\begin{aligned}\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta) &= \mathbb{E}\big[(\alpha_\theta - \overline{\alpha}_\theta)\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\big]\\
&= \mathbb{E}\big[(1-\overline{\alpha}_\theta)\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|\leqslant\tau_\theta}\big] + \mathbb{E}\big[(\alpha_\theta - \overline{\alpha}_\theta)\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big]\\
&= (\overline{\alpha}_\theta - 1)\mathbb{E}\big[\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big] - \overline{\alpha}_\theta\mathbb{E}\big[\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big]\\
&\quad + \mathbb{E}\big[\big(\tau_\theta/\|\widetilde{G}(\theta)\|\big)\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big]\\
&= -\mathbb{E}\big[\big(1 - \tau_\theta/\|\widetilde{G}(\theta)\|\big)\big(\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big)\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big].\end{aligned}$$

Using our choice of τ_θ and Hölder's inequality, we find :

$$\begin{aligned}\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\| &\leqslant \mathbb{E}\big[\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big|1 - \tau_\theta/\|\widetilde{G}(\theta)\|\big|\big\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big\|\big]\\
&\leqslant \mathbb{E}\big[\mathbf{1}_{\|\widetilde{G}(\theta)\|>\tau_\theta}\big\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big\|\big]\\
&\leqslant (1-p)^{1-1/q}\mathbb{E}\big[\big\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big\|^q\big]^{1/q}\\
&\leqslant (1-p)^{1-1/q}\big(A_q\|\theta-\theta^\star\| + B_q\big),\end{aligned}$$

which proves (5.9.4). To show (5.9.5), we first write the inequality:

$$\tau_\theta = Q_p(\|\widetilde{G}(\theta)\|) = Q_p(\|\nabla\mathcal{L}(\theta) + \varepsilon_\theta\|) \leqslant \|\nabla\mathcal{L}(\theta)\| + Q_p(\|\varepsilon_\theta\|),$$

which holds since $\|\widetilde{G}(\theta)\|$ is a positive random variable. Further, using Assumption 5.5, we have:

$$1 - p = \mathbb{P}\big(\|\varepsilon_\theta\| > Q_p(\|\varepsilon_\theta\|)\big) \leqslant \frac{\mathbb{E}[\|\varepsilon_\theta\|^q]}{Q_p(\|\varepsilon_\theta\|)^q} \leqslant \Big(\frac{A_q\|\theta-\theta^\star\| + B_q}{Q_p(\|\varepsilon_\theta\|)}\Big)^q.$$

It only remains to take the q-th root and plug the obtained bound on $Q_p(\|\varepsilon_\theta\|)$ back above to

obtain (5.9.5).

To show (5.9.6), we define the event $\mathcal{E} = \{\|\widetilde{G}(\theta)\| \leqslant \tau_\theta\}$ and denote $\overline{\mathcal{E}}$ its complement such that $\mathbb{P}(\mathcal{E}) = p = 1 - \mathbb{P}(\overline{\mathcal{E}})$. We write

$$\begin{aligned}
\mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 &= p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2|\mathcal{E}\big] \\
&\quad + (1-p)\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2|\overline{\mathcal{E}}\big] \\
&\leqslant p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2|\mathcal{E}\big] + 4(1-p)\tau_\theta^2 \\
&= p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] \\
&\quad + p\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}] - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 + 4(1-p)\tau_\theta^2 \\
&= p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] \\
&\quad + p\big\|(1-p)\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}] - (1-p)\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\overline{\mathcal{E}}]\big\|^2 + 4(1-p)\tau_\theta^2 \\
&\leqslant p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] + 4p(1-p)^2\tau_\theta^2 + 4(1-p)\tau_\theta^2 \\
&\leqslant p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] + 5(1-p)\tau_\theta^2
\end{aligned}$$

where we used that $\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] = p\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}] + (1-p)\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\overline{\mathcal{E}}]$ and $p(1-p) \leqslant 1/4$. In addition, we have

$$\begin{aligned}
p\mathbb{E}\big[\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] &= p\mathbb{E}\big[\big\|\widetilde{G}(\theta) - \mathbb{E}[\widetilde{G}(\theta)|\mathcal{E}]\big\|^2|\mathcal{E}\big] \leqslant \mathbb{E}\big[\big\|\widetilde{G}(\theta) - \mathbb{E}[\widetilde{G}(\theta)]\big\|^2\big] \\
&\leqslant \mathbb{E}\big[\big\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\big\|^q\big]^{2/q} \leqslant \big(A_q\|\theta - \theta^\star\| + B_q\big)^2.
\end{aligned}$$

The first inequality is obtained by applying Lemma 5.3 to each coordinate of $\widetilde{G}(\theta)$ while conditioning on θ. The second one uses Jensen's inequality and the third results from Assumption 5.5. □

Lemma 5.3. *Let Y be a real random variable and $\mathcal{E}$ an event, then we have the inequality*

$$\mathbb{P}(\mathcal{E})\mathbb{E}[(Y - \mathbb{E}[Y|\mathcal{E}])^2|\mathcal{E}] \leqslant \mathbb{E}(Y - \mathbb{E}[Y])^2.$$

Proof. Define the conditional variance of a real random variable Y w.r.t. another variable Y as $\mathrm{Var}(Y|X) = \mathbb{E}[(Y - \mathbb{E}[Y|X])^2|X]$.

By Eve's law (see for instance [37, Theorem 9.5.4]) we have the identity

$$\mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y|X)] + \mathrm{Var}(\mathbb{E}[Y|X]),$$

which entails the inequality $\mathrm{Var}(Y) \geqslant \mathbb{E}[\mathrm{Var}(Y|X)]$. Applying the latter with $X = \mathbf{1}_\mathcal{E}$ yields

$$\mathbb{E}(Y - \mathbb{E}[Y])^2 = \mathrm{Var}(Y) \geqslant \mathbb{P}(\mathcal{E})\mathbb{E}[(Y - \mathbb{E}[Y|\mathcal{E}])^2|\mathcal{E}] + (1 - \mathbb{P}(\mathcal{E}))\mathbb{E}[(Y - \mathbb{E}[Y|\overline{\mathcal{E}}])^2|\overline{\mathcal{E}}],$$

which implies the result. □

We show the geometric ergodicity of the SGD Markov chain $(\theta_t)_{t\geqslant 0}$ by relying on [313, Theorem 15.0.1]. We will show that the following function :

$$V(\theta) := 1 + \|\theta - \theta^\star\|^2,$$

satisfies a *geometric drift* property. We define the action of the transition kernel $P_{\beta,p}$ on real integrable functions f over $\mathbb{R}^d$ through:

$$P_{\beta,p}f(\theta) = \mathbb{E}\big[f(\theta - \alpha_\theta\beta G(\theta))\big].$$

We also define the variation operator:

$$\Delta f(\theta) := P_{\beta,p} f(\theta) - f(\theta).$$

In many of our proofs, we will make use of the following adjustable bound.

Fact 1. *For any real numbers a, b and positive ϵ, we have the inequality*

$$2ab \leqslant a^2\epsilon + b^2/\epsilon.$$

Proof of Theorem 5.1

First, we define $\underline{\tau} = \inf_\theta \tau_\theta$. Note that Assumption 5.4 excludes the existence of θ such that $\widetilde{G}(\theta) = 0$ almost surely, therefore, we have $\underline{\tau} > 0$.

Thanks to Assumption 5.4 and conditioning on $\theta_t = \theta \in \mathbb{R}^d$ for $t \geqslant 0$, the distribution of θ_{t+1} has a strictly positive density at least on a ball of radius $\beta\underline{\tau}$ around θ_t. This implies that the chain is aperiodic since $P_{\beta,p}(\theta_t, W_{\theta_t}) > 0$ for any neighborhood W_{θ_t} contained in the previous ball. Moreover, by induction, the distribution of θ_{t+m} has positive density at least on a ball of radius $m\beta\underline{\tau}$ around θ_t. Thus for m high enough we have $P^m_{\beta,p}(\theta_t, A) > 0$ for any set A with non zero Lebesgue measure. It follows that the Markov chain is irreducible w.r.t. Lebesgue's measure and is thus ψ-irreducible (see [313, Chapter 4]).

For fixed θ, noticing that condition (5.3.2) implies $\beta < \frac{2}{\mu+L}$, using Lemma 5.1 and denoting α_θ and $\overline{\alpha}_\theta$ as in Lemma 5.2, we find:

$$\begin{aligned}
P_{\beta,p}\|\theta - \theta^\star\|^2 &= \mathbb{E}\|\theta - \beta\alpha_\theta G(\theta) - \theta^\star\|^2 \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\|\theta - \alpha_\theta\beta\widetilde{G}(\theta) - \theta^\star\|^2\big] \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - \\
&\quad 2\beta\langle\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star, \alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \beta^2\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2\big] \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\| \\
&\quad + \beta\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)^2 + \beta^2\tau_\theta^2\big],
\end{aligned}$$

where the last step uses that

$$\begin{aligned}
\mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2 &= \mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 + \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2 \\
&= \mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta)\big\|^2 - \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 + \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2 \\
&\leqslant \tau_\theta^2 + \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2.
\end{aligned}$$

Using Lemmas 5.1 and 5.2 and Assumption 5.5, we arrive at:

$$\begin{aligned}
P_{\beta,p}\|\theta - \theta^\star\|^2 &\leqslant \big(\eta + (1-\eta)(1 - \overline{\alpha}_\theta\beta\mu)^2\big)\mathbb{E}\|\theta - \theta^\star\|^2 + \\
&\quad 2\beta\mathbb{E}\big[\|\theta - \theta^\star\|\big(\eta\tau_\theta + (1-\eta)(1 - \overline{\alpha}_\theta\beta\mu)\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)\big] + \\
&\quad \beta^2\big(\tau_\theta^2 + (1-\eta)\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2\big) \\
&\leqslant \mathbb{E}\big[\mathfrak{A}\|\theta - \theta^\star\|^2 + 2\beta\mathfrak{B}\|\theta - \theta^\star\|\big] + \beta^2\mathfrak{C} \\
&\leqslant (\mathfrak{A} + \beta\kappa)\mathbb{E}\big[\|\theta - \theta^\star\|^2\big] + \frac{\beta\mathfrak{B}^2}{\kappa} + \beta^2\mathfrak{C},
\end{aligned} \tag{5.9.7}$$

where the last step uses Fact 1 and we defined the quantities:

$$\begin{aligned}
\mathfrak{A} &= 1 - 2\beta\big((1-\eta)\overline{\alpha}_\theta\mu - \eta L - (1-p)^{-\frac{1}{q}}A_q\big(\eta + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)(1-p)\big)\big) + \\
&\quad \beta^2\big((1-\eta)(\overline{\alpha}_\theta\mu)^2 + \big(L + (1-p)^{-\frac{1}{q}}A_q\big)^2 + (1-\eta)(1-p)^{2-\frac{2}{q}}A_q^2\big) \\
&\leqslant 1 - 2\beta\big((1-\eta)p\mu - \eta L - (1-p)^{-\frac{1}{q}}A_q\big(1-p(1-\eta)\big)\big) + \beta^2\big(\mu^2 + 2L^2 + 4(1-p)^{-\frac{2}{q}}A_q^2\big), \\
\mathfrak{B} &= (1-p)^{-\frac{1}{q}}B_q\big(\eta + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)(1-p) + \beta\big(L + (1-p)^{-\frac{1}{q}}A_q(1+(1-\eta)(1-p)^2)\big)\big) \\
&\leqslant (1-p)^{-\frac{1}{q}}B_q\big(\eta + (1-p) + \beta(L + 2(1-p)^{-\frac{1}{q}}A_q)\big), \\
\mathfrak{C} &= (1-p)^{-\frac{2}{q}}B_q^2 + (1-\eta)(1-p)^{2-\frac{2}{q}}B_q^2 \\
&\leqslant 2(1-p)^{-\frac{2}{q}}B_q^2.
\end{aligned}$$

Thanks to our choice of β, we get that $\mathfrak{A}+\beta\kappa < 1$. It is now easy to check that $V(\theta) = 1+\|\theta-\theta^\star\|^2$ satisfies the contraction

$$P_{\beta,p}V(\theta) \leqslant \underbrace{(\mathfrak{A}+\beta\kappa)}_{=:\widetilde{\lambda}<1}V(\theta) + \underbrace{1 - (\mathfrak{A}+\beta\kappa) + \frac{\beta\mathfrak{B}^2}{\kappa} + \mathfrak{C}}_{=:\widetilde{b}}.$$

We now define the set $\mathcal{C} = \big\{\theta \in \mathbb{R}^d\,,\, V(\theta) \leqslant 2\widetilde{b}/(1-\widetilde{\lambda})\big\}$ for which we have:

$$\Delta V(\theta) \leqslant -\frac{1-\widetilde{\lambda}}{2}V(\theta) + \widetilde{b}\mathbf{1}_{\theta\in\mathcal{C}}. \tag{5.9.8}$$

For such $\mathcal{C}$, let $\Delta_{\mathcal{C}} = \operatorname{diam}(\mathcal{C})$ be its diameter and set $m_{\mathcal{C}} = \left\lceil \frac{\Delta_{\mathcal{C}}}{\beta\underline{\tau}} \right\rceil$. As previously mentioned in the beginning of the proof, conditioning on $\theta_t = \theta$, the distribution of θ_{t+m} admits a positive density at least over a ball of radius $m\beta\underline{\tau}$ around θ i.e. there exists $h_\theta^{+m}(\omega) \geqslant 0$ satisfying

$$h_\theta^{+m}(\omega) > 0 \quad \text{for} \quad \|\theta-\omega\| < m\beta\underline{\tau} \quad \text{and} \quad P_{\beta,p}^m(\theta, A) \geqslant \int_A h_\theta^{+m}(\omega)d\omega \quad \text{for} \quad A \in \mathcal{B}(\mathbb{R}^d).$$

We then let $\underline{h}_{\mathcal{C}}(\omega) = \inf_{\theta\in\mathcal{C}} h_\theta^{+m_{\mathcal{C}}}(\omega)$ and define the measure $\nu_{\mathcal{C}}$ by:

$$\nu_{\mathcal{C}}(A) = \int_{A\cap\mathcal{C}} \underline{h}_{\mathcal{C}}(\theta)d\theta.$$

The above measure is non trivial since our choice of $m_{\mathcal{C}}$ ensures that $\underline{h}_{\mathcal{C}}$ defines a non zero density at least on $\mathcal{C}$. It follows that for all $\theta_0 \in \mathcal{C}$, we have the following minorization property:

$$P_{\beta,p}^{m_{\mathcal{C}}}(\theta_0, A) \geqslant \nu_{\mathcal{C}}(A) \quad \text{for all} \quad A \in \mathcal{B}(\mathbb{R}^d).$$

This implies that the set $\mathcal{C}$ is $(m_{\mathcal{C}}, \nu_{\mathcal{C}})$-*small* and, thanks to [313, Proposition 5.5.3], also a *petite* set (see definitions in [313, Chapter 5]).

We have shown that the Markov chain (θ_t) satisfies condition (iii) of [313, Theorem 15.0.1]. By the latter result, it follows that it admits a unique invariant probability measure $\pi_{\beta,p}$ and there exist $r > 1$ and $M < \infty$ such that:

$$\sum_{t\geqslant 0} r^t \big\|P_{\beta,p}(\theta_0,\cdot) - \pi_{\beta,p}\big\|_V \leqslant MV(\theta_0). \tag{5.9.9}$$

Taking $\rho = r^{-1}$ concludes the proof.

Proof of Proposition 5.1

We consider the case $q \geqslant 2$. Since the distribution $\pi_{\beta,p}$ is invariant by the transition kernel $P_{\beta,p}$, we can deduce that for $\theta \sim \pi_{\beta,p}$, we have:

$$\begin{aligned}
\mathbb{E}\|\theta - \theta^\star\|^2 &= \mathbb{E}\|\theta - \beta\alpha_\theta G(\theta) - \theta^\star\|^2 \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\|\theta - \alpha_\theta\beta\widetilde{G}(\theta) - \theta^\star\|^2\big] \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - \\
&\quad 2\beta\langle\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star, \alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \beta^2\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2\big] \\
&\leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^2 + (1-\eta)\mathbb{E}\big[\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\| \\
&\quad + \beta\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)^2 + \beta^2\big(A_q\|\theta - \theta^\star\| + B_q\big)^2 + 5\beta^2(1-p)\tau_\theta^2\big],
\end{aligned}$$

where the last step uses that $\mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2 = \mathbb{E}\big\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\|^2 + \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2$ and Lemma 5.2. Using Lemmas 5.1 and 5.2 and Assumption 5.5 and grouping the terms by powers of $\|\theta - \theta^\star\|$, we arrive at

$$\begin{aligned}
\mathbb{E}\|\theta - \theta^\star\|^2 &\leqslant \big(\eta + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)^2\big)\mathbb{E}\|\theta - \theta^\star\|^2 + \\
&\quad 2\beta\mathbb{E}\big[\|\theta - \theta^\star\|\big(\eta\tau_\theta + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)\big] + \\
&\quad \beta^2\big(\eta\tau_\theta^2 + (1-\eta)\big(\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2 + (A_q\|\theta - \theta^\star\| + B_q)^2 + \\
&\quad 5(1-p)\tau_\theta^2\big)\big) \qquad (5.9.10) \\
&\leqslant \mathbb{E}\big[\mathfrak{A}\|\theta - \theta^\star\|^2 + 2\beta\mathfrak{B}\|\theta - \theta^\star\|\big] + \beta^2\mathfrak{C} \\
&\leqslant (\mathfrak{A} + \beta\kappa)\mathbb{E}\big[\|\theta - \theta^\star\|^2\big] + \frac{\beta\mathfrak{B}^2}{\kappa} + \beta^2\mathfrak{C}, \qquad (5.9.11)
\end{aligned}$$

where we used Fact 1 and defined the quantities $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$ which may be bounded as follows

$$\begin{aligned}
\mathfrak{A} &= \eta + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)^2 + 2\beta(\eta(L + A_q(1-p)^{-1/q}) + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)(1-p)^{-1/q}A_q) \\
&\quad + \beta^2((\eta + 5(1-p))(L + (1-p)^{-1/q}A_q)^2 + 4A_q^2) \\
&\leqslant 1 - 2\beta\big((1-\eta)\overline{\alpha}_\theta\mu - \eta(L + A_q(1-p)^{-1/q}) - (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)(1-p)^{1-1/q}A_q\big) + \\
&\quad \beta^2\big((1-\eta)(\overline{\alpha}_\theta\mu)^2 + (\eta + 5(1-p))\big(L + (1-p)^{-\frac{1}{q}}A_q\big)^2 + 4A_q^2\big) \\
&\leqslant 1 - 2\beta\big((1-\eta)p\mu - \eta L - (1-p)^{-\frac{1}{q}}A_q\big(1 - p(1-\eta)\big)\big) + \beta^2\big(\mu^2 + 12\eta L^2 + 16A_q^2\big) \\
&= 1 - 2\beta\kappa + \beta^2\big(\mu^2 + 12\eta L^2 + 16A_q^2\big), \\
\mathfrak{B} &= \big(\eta(1-p)^{-1/q} + (1-\eta)(1-\overline{\alpha}_\theta\beta\mu)(1-p)^{1-1/q}\big)B_q + \\
&\quad \beta\big((\eta + 5(1-p))(L + A_q(1-p)^{-1/q})(1-p)^{-1/q}B_q\big) \\
&\leqslant (1-p)^{-1/q}B_q(\eta + (1-p) + \beta(\eta + 5(1-p))(L + A_q\eta^{-1/q})) \\
&\leqslant 2\eta^{1-1/q}B_q(1 + 3\beta(L + A_q\eta^{-1/q})), \\
\mathfrak{C} &= \big(4 + (\eta + 5(1-p))(1-p)^{-2/q}\big)B_q^2 \leqslant 10B_q^2.
\end{aligned}$$

The above bounds use the simple properties $p \leqslant \overline{\alpha}_\theta \leqslant 1$, $0 \leqslant \eta \leqslant 1$, $1 - \overline{\alpha}_\theta\beta\mu \leqslant 1$, $q \geqslant 2$, $(a+b)^2 \leqslant 2a^2 + 2b^2$ for all a, b and the choice $p = 1 - \eta$.

Hence, we have that:

$$\mathbb{E}\|\theta - \theta^\star\|^2 \leqslant \frac{\beta\mathfrak{B}^2}{\kappa(1-\mathfrak{A}-\beta\kappa)} + \frac{\beta^2\mathfrak{C}}{1-\mathfrak{A}-\beta\kappa}. \tag{5.9.12}$$

Using (5.3.2), we find that

$$1 - \mathfrak{A} - \beta\kappa \geqslant \beta\kappa - \beta^2(\mu^2 + 12\eta L^2 + 16A_q^2) \geqslant 3\beta\kappa/4, \tag{5.9.13}$$

Moreover, using (5.3.2) and (5.3.3), we have

$$\begin{aligned}(1 + 3\beta(L + A_q\eta^{-1/q}))^2 &\leqslant 2 + 18\beta^2(L + A_q\eta^{-1/q})^2 \leqslant 2 + 36\beta^2(L^2 + 2A_q^2\eta^{-2/q}) \\ &\leqslant 2 + 3\kappa\beta/2 \leqslant 2 + 3\eta^{2-2/q}/2 \leqslant 4\end{aligned}$$

Putting everything together into (5.9.12) and using (5.3.3) once more, we find

$$\mathbb{E}\|\theta - \theta^\star\|^2 \leqslant \frac{16\eta^{2-2/q}B_q^2}{3\kappa^2/4} + \frac{10\beta B_q^2}{3\kappa/4} \leqslant \frac{26\eta^{2-2/q}B_q^2}{3\kappa^2/4} \leqslant \frac{35\eta^{2-2/q}B_q^2}{\kappa^2}$$

which implies the result.

The case $q \in (1, 2)$

A similar result to Proposition 5.1 holds for the case $q \in (1, 2)$ but requires a different proof and is given below.

Proposition 5.4. *Let Assumptions 5.1-5.5 hold with $q \in (1, 2)$ and assume that*

$$\kappa' := (1-\eta)p\mu - q\eta L - 2q\eta^{1-1/q}A_q > 0. \tag{5.9.14}$$

Let QC-SGD be run with quantile $p = 1 - \eta$ and step size satisfying

$$\beta \leqslant \frac{\eta}{\kappa'} \wedge \Big(\frac{\kappa'}{86(L + A_q)^q}\Big)^{\frac{1}{q-1}}, \tag{5.9.15}$$

then the generated Markov chain $(\theta_t)_t$ converges geometrically to a unique invariant measure $\pi_{\beta,1-\eta}$ as in Theorem 5.1. In addition, for $\theta \sim \pi_{\beta,1-\eta}$, we have

$$\mathbb{E}\|\theta - \theta^\star\|^q \leqslant 128\Big(\frac{\eta^{1-1/q}B_q}{\kappa'}\Big)^q.$$

Proof. One can see that conditions (5.3.1) and (5.3.2) of Theorem 5.1 with $p = 1-\eta$ are implied by the assumptions. Therefore, the geometric convergence of the Markov chain follows.

In this proof, we will use the following inequalities valid for all positive x, y and ε and $q \in (1, 2)$

$$(x + y)^q \leqslant 2^{q-1}(x^q + y^q), \tag{5.9.16}$$

$$(x + y)^q \leqslant x^q + qx^{q-1}y + y^q, \tag{5.9.17}$$

(a consequence of the inequality $(1 + a)^q \leqslant 1 + qa + a^q$ for $a > 0$),

$$(x + y)^{q-1} \leqslant x^{q-1} + y^{q-1}, \tag{5.9.18}$$

and

$$xy \leqslant \frac{(x\varepsilon)^q}{q} + \frac{q-1}{q}(y/\varepsilon)^{q/(q-1)} \tag{5.9.19}$$

which is a consequence of Young's inequality applied to the pair $x\varepsilon$ and y/ε with exponent q and its conjugate. We first write

$$\mathbb{E}\|\theta - \theta^\star\|^q = \mathbb{E}\|\theta - \beta\alpha_\theta G(\theta) - \theta^\star\|^q \leqslant \eta\mathbb{E}\big(\|\theta - \theta^\star\| + \beta\tau_\theta\big)^q + (1-\eta)\mathbb{E}\|\theta - \alpha_\theta\beta\widetilde{G}(\theta) - \theta^\star\|^q$$

Defining the notation $\mathbb{E}_\theta[\cdot] = \mathbb{E}[\cdot|\theta]$, we have

$$\begin{aligned}
&\mathbb{E}\|\theta - \alpha_\theta\beta\widetilde{G}(\theta) - \theta^\star\|^q = \mathbb{E}\big[\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - \\
&\quad 2\beta\langle\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star, \alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \beta^2\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2\big)^{q/2}\big] \\
&\quad \leqslant \mathbb{E}\big[\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - 2\beta\langle\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star, \alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \\
&\quad \beta^2(\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^2 + 2\langle\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)], \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \\
&\quad \|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2)\big)^{q/2}\big] \\
&\quad \leqslant \mathbb{E}\big[\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - 2\beta\langle\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star, \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\rangle + \\
&\quad \beta^2(\mathbb{E}_\theta\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^2 + \|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2)\big)^{q/2}\big] \\
&\quad \leqslant \mathbb{E}\big(\|\theta - \overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta) - \theta^\star\| + \beta\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|\big)^q + \beta^q\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q,
\end{aligned}$$

where we conditioned on θ and used Jensen's inequality for $\mathbb{E}_\theta$ then a Cauchy-Schwarz inequality and (5.9.18). We focus on the last term. Defining the event $\mathcal{E} = \{\|\widetilde{G}(\theta)\| \leqslant \tau_\theta\}$ such that $\mathbb{P}(\mathcal{E}) = p$, we have

$$\begin{aligned}
\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q &= (1-p)\mathbb{E}[\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q|\overline{\mathcal{E}}] \\
&\quad + p\mathbb{E}[\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q|\mathcal{E}] \\
&\leqslant (1-p)(2\tau_\theta)^q + p\mathbb{E}[\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q|\mathcal{E}].
\end{aligned}$$

In addition, using (5.9.18) twice, we find

$$\begin{aligned}
\mathbb{E}[\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q|\mathcal{E}] &\leqslant 2^{q-1}\big(\mathbb{E}[\|\alpha_\theta\widetilde{G}(\theta) - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^q|\mathcal{E}] + \\
&\quad \|\overline{\alpha}_\theta\nabla\mathcal{L}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q\big) \\
&\leqslant 2^{q-1}\big(2^{q-1}\mathbb{E}[\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\|^q|\mathcal{E}] + 2^{q-1}(1-\overline{\alpha}_\theta)^q\|\nabla\mathcal{L}(\theta)\|^q + \\
&\quad \|\overline{\alpha}_\theta\nabla\mathcal{L}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q\big).
\end{aligned}$$

Therefore, from the two previous displays and using Assumption 5.1, Lemma 5.2 and the inequality $p\mathbb{E}[\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\|^q|\mathcal{E}] \leqslant \mathbb{E}[\|\widetilde{G}(\theta) - \nabla\mathcal{L}(\theta)\|^q]$, we get

$$\begin{aligned}
\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q &\leqslant (1-p)(2\tau_\theta)^q + 2^{2q-2}p(1-\overline{\alpha}_\theta)^q L^q\|\theta - \theta^\star\|^q \\
&\quad + (2^{2q-2} + 2^{q-1}p(1-p)^{q-1})(A_q\|\theta - \theta^\star\| + B_q)^q
\end{aligned}$$

We also have

$$\begin{aligned}
(1-p)\tau_\theta^q &\leqslant (1-p)((L + (1-p)^{-1/q}A_q)\|\theta - \theta^\star\| + (1-p)^{-1/q}B_q)^q \\
&\leqslant ((L + A_q)\|\theta - \theta^\star\| + B_q)^q.
\end{aligned}$$

Plugging into the previous display and simplifying leads to

$$\begin{aligned}\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta) - \mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q &\leqslant 2^{2q-2}((L+A_q)\|\theta-\theta^\star\|+B_q)^q + 2^{2q-2}p(1-\overline{\alpha}_\theta)^q(L\|\theta-\theta^\star\|)^q \\ &\quad + (2^{2q-2}+2^{q-1}p(1-p)^{q-1})(A_q\|\theta-\theta^\star\|+B_q)^q \\ &\leqslant 2^{2q-1}((L+A_q)\|\theta-\theta^\star\|+B_q)^q + (2^{2q-2}+2^{q-1}p(1-p)^{q-1})(A_q\|\theta-\theta^\star\|+B_q)^q \\ &\leqslant 14((L+A_q)\|\theta-\theta^\star\|+B_q)^q. \end{aligned} \tag{5.9.20}$$

Using these inequalities along with Lemma 5.2 to bound $\mathbb{E}\|\theta-\theta^\star\|^q$, we find that

$$\begin{aligned}\mathbb{E}\|\theta-\theta^\star\|^q &\leqslant \eta\mathbb{E}\big(\|\theta-\theta^\star\|+\beta\tau_\theta\big)^q + (1-\eta)\beta^q\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta)-\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q \\ &\quad + (1-\eta)\mathbb{E}\big(\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\|+\beta\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|\big)^q \\ &\leqslant \eta\mathbb{E}\big(\|\theta-\theta^\star\|^q+\beta^q\tau_\theta^q+q\beta\|\theta-\theta^\star\|^{q-1}\tau_\theta\big) + (1-\eta)\mathbb{E}\big((1-\overline{\alpha}_\theta\beta\mu)^q\|\theta-\theta^\star\|^q \\ &\quad + \beta^q\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^q + q\beta\|\theta-\theta^\star\|^{q-1}\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|\big) \\ &\quad + (1-\eta)\beta^q\mathbb{E}\|\alpha_\theta\widetilde{G}(\theta)-\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q \\ &\leqslant \mathbb{E}\big[(1-(1-\eta)\overline{\alpha}_\theta\beta\mu)\|\theta-\theta^\star\|^q\big] + q\beta\mathbb{E}\big[\|\theta-\theta^\star\|^{q-1}\big(\eta\tau_\theta \\ &\quad + (1-\eta)\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|\big) + \beta^q\big(\eta\tau_\theta^q \\ &\quad + (1-\eta)\big(\|\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^q + \|\alpha_\theta\widetilde{G}(\theta)-\mathbb{E}_\theta[\alpha_\theta\widetilde{G}(\theta)]\|^q\big)\big)\big] \\ &\leqslant (1-\beta((1-\eta)p\mu - q\eta(L+(1-p)^{-1/q}A_q) - q(1-\eta)(1-p)^{1-1/q}A_q) \\ &\quad + \beta^q 2^{q-1}(\eta(L+(1-p)^{-1/q}A_q)^q + (1-\eta)14(L+A_q)^q \\ &\quad + (1-\eta)(1-p)^{q-1}A_q^q))\mathbb{E}\|\theta-\theta^\star\|^q + q\beta\mathbb{E}\|\theta-\theta^\star\|^{q-1}\big((1-\eta)(1-p)^{1-1/q}B_q \\ &\quad + \eta(1-p)^{-1/q}B_q\big) + \beta^q 2^{q-1}\big(\eta(1-p)^{-1}B_q^q + (1-\eta)(1-p)^{q-1}B_q^q + 14B_q^q\big) \\ &\leqslant \big(1-\beta\kappa' + \beta^q 2^{q-1}(16)(L+A_q)^q\big)\mathbb{E}\|\theta-\theta^\star\|^q + 2q\beta\eta^{1-1/q}B_q\mathbb{E}\|\theta-\theta^\star\|^{q-1} \\ &\quad + \beta^q 2^{q-1}(16)B_q^q, \end{aligned}$$

where the second inequality uses Lemma 5.1 and (5.9.17) twice, the third inequality rearranges according to powers of $\|\theta-\theta^\star\|$, the fourth one applies Lemma 5.2 and (5.9.20) and the last inequalities simplifies the factors. Applying (5.9.19), we have for all $\varepsilon > 0$ that

$$\eta^{1-1/q}B_q\mathbb{E}\|\theta-\theta^\star\|^{q-1} \leqslant \frac{\varepsilon^{\frac{q}{q-1}}\mathbb{E}\|\theta-\theta^\star\|^q}{q/(q-1)} + \frac{(\eta^{1-1/q}B_q/\varepsilon)^q}{q}$$

and we choose $\varepsilon = \Big(\frac{\kappa'}{8(q-1)}\Big)^{\frac{q-1}{q}}$. Plugging back above and using condition (5.9.15) on β, we find

$$\mathbb{E}\|\theta-\theta^\star\|^q \leqslant \big(1-(3/4)\beta\kappa' + 32\beta^q(L+A_q)^q\big)\mathbb{E}\|\theta-\theta^\star\|^q + 2\beta(\eta^{1-1/q}B_q/\varepsilon)^q + 32\beta^q B_q^q$$

Now using the condition $\beta \leqslant \eta/\kappa'$ from (5.9.15) and rearranging the inequality, we finally arrive at

$$\begin{aligned}&\mathbb{E}\|\theta-\theta^\star\|^q \leqslant (1-3\beta\kappa'/8)\mathbb{E}\|\theta-\theta^\star\|^q + 2\beta(\eta^{1-1/q}B_q)^q\Big(\frac{\kappa'}{8(q-1)}\Big)^{1-q} + 32\beta^q B_q^q \\ \implies\ &\mathbb{E}\|\theta-\theta^\star\|^q \leqslant \frac{8}{3}\Big(\frac{\eta^{1-1/q}B_q}{\kappa'}\Big)^q\Big(2(8(q-1))^{q-1}+32\Big) \\ &\qquad\qquad \leqslant 128\Big(\frac{\eta^{1-1/q}B_q}{\kappa'}\Big)^q, \end{aligned}$$

which is the desired result. □

Proof of Proposition 5.2

We use the invariance of $\pi_{\beta,p}$ by the transition kernel $P_{\beta,p}$. For real λ, this implies the equality

$$\mathbb{E}\exp\big(\lambda^2\|\theta-\theta^\star\|^2\big)=\mathbb{E}\exp\big(\lambda^2\|\theta-\alpha_\theta\beta G(\theta)-\theta^\star\|^2\big). \tag{5.9.21}$$

We then write:

$$\begin{aligned}\big\|\theta-\alpha_\theta\beta G(\theta)-\theta^\star\big\|^2=\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|^2+\beta^2\big\|\alpha_\theta G(\theta)-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2\\ -2\beta\big\langle\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star,\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]+\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\rangle.\end{aligned}$$

We also have the inequality

$$\big\|\alpha_\theta G(\theta)-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2\leqslant 2\big\|\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]\big\|^2+2\big\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2.$$

Conditioning upon θ, we have that $\|\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]\|\leqslant 2\tau_\theta\leqslant 2\overline{\tau}$. Moreover, the vector $\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]$ is centered, therefore it is sub-Gaussian with constant $2\overline{\tau}$ (see [432, Proposition 2.5.2]). Still conditioning on θ, using these two properties and a Cauchy-Schwarz inequality, we find:

$$\begin{aligned}\mathbb{E}\exp\big(\lambda^2\big(2\beta^2\big\|\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]\big\|^2-2\beta\big\langle\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star,\alpha_\theta G(\theta)-\mathbb{E}[\alpha_\theta G(\theta)]\big\rangle\big)\big)\\ \leqslant\exp\big(8\lambda^2\beta^2\overline{\tau}^2+16\lambda^4\beta^2\overline{\tau}^2\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|^2\big).\end{aligned}$$

Putting everything together in (5.9.21), we get:

$$\begin{aligned}\mathbb{E}\exp(\lambda^2\|\theta-\theta^\star\|^2)&\leqslant\mathbb{E}\exp\big(\lambda^2\big((1+16\lambda^2\beta^2\overline{\tau}^2)\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|^2+8\beta^2\overline{\tau}^2\\ &\quad+2\beta^2\big\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2+2\beta\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|\big\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)\big)\\ &\leqslant\mathbb{E}\exp\big(\lambda^2\big((1+16\lambda^2\beta^2\overline{\tau}^2+\epsilon)\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|^2+8\beta^2\overline{\tau}^2\\ &\quad+2\beta^2(1+1/(2\epsilon))\big\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|^2\big)\big),\end{aligned}$$

where we used Fact 1. Now, recalling that $\overline{\alpha}_\theta\geqslant p$, we set $\epsilon=\overline{\alpha}_\theta\beta\mu/2$ and restrict λ to $\lambda\leqslant(4\overline{\tau}\sqrt{2\beta/p\mu})^{-1}$ to find:

$$\begin{aligned}\mathbb{E}\exp\big(\lambda^2\|\theta-\theta^\star\|^2\big)&\leqslant\mathbb{E}\exp\big(\lambda^2\big((1+16\lambda^2\beta^2\overline{\tau}^2+\epsilon)\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\|^2+8\beta^2\overline{\tau}^2\\ &\quad+2\beta^2(1+1/(2\epsilon))\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\|^2\big)\big)\\ &\leqslant\mathbb{E}\exp\Big(\lambda^2\Big((1+\overline{\alpha}_\theta\beta\mu)\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\|^2+8\beta^2\overline{\tau}^2+\frac{4\beta}{\overline{\alpha}_\theta\mu}\big\|\mathbb{E}[\alpha_\theta G(\theta)]-\overline{\alpha}\nabla\mathcal{L}(\theta)\big\|^2\Big)\Big)\\ &\leqslant\mathbb{E}\exp\Big(\lambda^2\Big((1+\overline{\alpha}_\theta\beta\mu)(1-\overline{\alpha}_\theta\beta\mu)^2\|\theta-\theta^\star\|^2+8\beta^2\overline{\tau}^2+\frac{4\beta}{\overline{\alpha}_\theta\mu}\big((1-p)^{1-1/q}\overline{B}_q\big)^2\Big)\Big)\\ &\leqslant\mathbb{E}\exp\big(\lambda^2\big((1-\overline{\alpha}_\theta\beta\mu)(1-(\overline{\alpha}_\theta\beta\mu)^2)\|\theta-\theta^\star\|^2+4\beta^2(2\overline{\tau}^2+\overline{B}_q^2/p)\big)\big),\end{aligned}$$

where we used Lemma 5.1, inequality (5.9.4) from Lemma 5.2 (recall that $\eta=0$ in this context) and the imposed bound relating β and p.

Finally, using Jensen's inequality and the fact that $\overline{\alpha}_\theta\geqslant p\geqslant 1/2$ we get:

$$\mathbb{E}\exp\big(\lambda^2\|\theta-\theta^\star\|^2\big)\leqslant\exp\left(\frac{8\lambda^2\beta^2\big(\overline{\tau}^2+\overline{B}_q^2\big)}{\overline{\alpha}\beta\mu+(\overline{\alpha}\beta\mu)^2-(\overline{\alpha}\beta\mu)^3}\right)\leqslant\exp\left(\frac{8\lambda^2\beta\big(\overline{\tau}^2+\overline{B}_q^2\big)}{p\mu}\right),$$

which concludes the first part of the proof. We now consider the corrupted case $\eta > 0$. Let $\lambda > 0$ and write :

$$\begin{aligned}
\mathbb{E}\big[\exp\big(\lambda\|\theta-\theta^\star\|\big)\big] &= \mathbb{E}\big[\exp\big(\lambda\|\theta-\alpha_\theta\beta G(\theta)-\theta^\star\|\big)\big] \\
&\leqslant \eta\mathbb{E}\big[\exp\big(\lambda\|\theta-\alpha_\theta\beta\breve{G}(\theta)-\theta^\star\|\big)\big] + (1-\eta)\mathbb{E}\big[\exp\big(\lambda\|\theta-\alpha_\theta\beta\widetilde{G}(\theta)-\theta^\star\|\big)\big] \\
&\leqslant \eta\mathbb{E}\big[\exp\big(\lambda(\|\theta-\theta^\star\|+\beta\tau_\theta)\big)\big] + (1-\eta)\mathbb{E}\big[\exp\big(\lambda\big\|\theta-\overline{\alpha}_\theta\beta\nabla\mathcal{L}(\theta)-\theta^\star\big\| \\
&+ \lambda\beta\big(\big\|\alpha_\theta\widetilde{G}(\theta)-\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\| + \big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]-\overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big)\big)\big] \\
&\leqslant \eta\mathbb{E}\big[\exp\big(\lambda\|\theta-\theta^\star\|\big)\big]e^{\lambda\beta\overline{\tau}} \\
&+ (1-\eta)\mathbb{E}\big[\exp\big(\lambda(1-p\beta\mu)\|\theta-\theta^\star\|\big)\big]\exp\big(\lambda\beta(2\overline{\tau}+(1-p)^{1-1/q}\overline{B}_q)\big),
\end{aligned}$$

where we used Lemma 5.1, the inequality $\alpha_\theta \geqslant p$, the inequality $\big\|\alpha_\theta\widetilde{G}(\theta)-\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)]\big\| \leqslant 2\tau_\theta \leqslant 2\overline{\tau}$ and (5.9.4) from Lemma 5.2. Using Hölder's inequality, this leads to :

$$\mathbb{E}\big[\exp\big(\lambda\|\theta-\theta^\star\|\big)\big] \leqslant \Big(\frac{1-\eta}{1-\eta e^{\lambda\beta\overline{\tau}}}\Big)^{1/(p\beta\mu)}\exp\big(\lambda(2\overline{\tau}+(1-p)^{1-1/q}\overline{B}_q)/(p\mu)\big)$$

Now we use the inequality $\log\big(\frac{1-\eta}{1-\eta e^{\lambda\beta\overline{\tau}}}\big) \leqslant \frac{\beta\overline{\tau}\lambda/\log(1/\eta)^2}{1-\beta\overline{\tau}\lambda/\log(1/\eta)}$ valid for $\lambda \geqslant 0$ which leads to:

$$\Big(\frac{1-\eta}{1-\eta e^{\lambda\beta\overline{\tau}}}\Big)^{1/(p\beta\mu)} \leqslant \exp\Big(\frac{2\lambda\overline{\tau}}{p\mu\log(1/\eta)^2}\Big) \quad \text{for} \quad 0 \leqslant \lambda \leqslant \frac{\log(1/\eta)}{2\beta\overline{\tau}}.$$

Using that $\eta < 1/2$, we find that for $0 \leqslant \lambda \leqslant \frac{\log(1/\eta)}{2\beta\overline{\tau}}$, the following inequality holds :

$$\begin{aligned}
\mathbb{E}[\exp(\lambda\|\theta-\theta^\star\|)] &\leqslant \exp\Big(\frac{\lambda}{p\mu}\Big(2\overline{\tau}\Big(1+\frac{1}{\log(1/\eta)^2}\Big)+(1-p)^{1-1/q}\overline{B}_q\Big)\Big) \\
&\leqslant \exp\Big(\frac{\lambda}{p\mu}\big(7\overline{\tau}+(1-p)^{1-1/q}\overline{B}_q\big)\Big).
\end{aligned}$$

Noticing that $\beta \leqslant \frac{1}{\mu}$ allows to finish the proof.

Unimprovability of the sub-exponential property for $\eta > 0$

We consider the Markov chain

$$X_{t+1} = \begin{cases} \alpha X_t + \xi & \text{w.p.} \quad 1-\eta \\ X_t + \tau & \text{w.p.} \quad \eta \end{cases}$$

Assuming that the distribution of the noise ξ has a density, one can show that the chain is aperiodic and satisfies a minorization property as in the proof of Theorem 5.1 (see Section 5.9.1).

Defining $V(x) = 1+x$, we can show that V verifies a geometric drift property similar to (5.9.8). Consequently, Theorem 15.0.1 of [313] applies to the chain $(X_t)_{t\geqslant 0}$ and implies that it converges geometrically to a limit distribution π analogously to the claim of Theorem 5.1.

We denote $M_k^k = \mathbb{E}|X|^k$ the absolute moments of X for $k \geqslant 1$ and show that $M_k = \Omega(k)$ (we merely provide a sketch and do not attempt to explicitly compute the involved constants). For $X \sim \pi$ following the invariant measure, using the recursion defining X_t and the positivity of ξ, it is easy to establish the inequality for $k \geqslant 1$

$$(1-\eta)(1-\alpha^k)M_k^k \geqslant \eta\sum_{j=1}^{k}\binom{k}{j}\tau^j M_{k-j}^{k-j}$$

where one may use the convention that $M_0 = 1$. We now postulate the induction hypothesis $M_j \geqslant Cj$ up to $j = k-1$ for some $k > 1$ and $C > 0$. Using Stirling's formula, we find:

$$\begin{aligned}\frac{(1-\eta)(1-\alpha^k)}{\eta} M_k^k &\geqslant \sum_{j=1}^{k} \binom{k}{j} \tau^j \big(C(k-j)\big)^{k-j} = \sum_{j=1}^{k} \frac{k!}{j!(k-j)!} \tau^j \big(C(k-j)\big)^{k-j} \\ &\gtrsim \sum_{j=1}^{k} \sqrt{\frac{k}{j(k-j)}} \frac{k^k \tau^j \big(C(k-j)\big)^{k-j}}{j^j (k-j)^{k-j}} = \sum_{j=1}^{k} \sqrt{\frac{k}{j(k-j)}} \Big(\frac{\tau}{j}\Big)^j k^k C^{k-j} \\ &\geqslant \frac{\tau}{C}(Ck)^k\end{aligned}$$

where $\gtrsim$ denotes an inequality up to a universal constant and we took the term $j = 1$ in the last step. It is only left to set C small enough such that $\frac{\tau\eta}{C(1-\eta)(1-\alpha)} \geqslant 1$ in order to finish the induction. It follows that $M_k = \Omega(k)$ implying that π may be sub-exponential but cannot be sub-Gaussian since that would require $M_k = \mathcal{O}(\sqrt{k})$ (see [432, Chapter 2] for a reference).

Proof of Corollary 5.1

We need the following lemma.

Lemma 5.4. *Let X be a real sub-Gaussian random variable with constant K then, with probability at least δ, we have :*

$$|X| \leqslant K\sqrt{\log(e/\delta)}$$

Proof. Using Chernoff's method, we find for $t > 0$ and $\lambda > 0$:

$$\begin{aligned}\mathbb{P}\big(|X| > t\big) = \mathbb{P}\big(\lambda^2 X^2 > \lambda^2 t^2\big) &= \mathbb{P}\big(\exp(\lambda^2 X^2) > \exp(\lambda^2 t^2)\big) \\ &\leqslant \mathbb{E}\exp(\lambda^2 X^2) e^{-\lambda^2 t^2} \leqslant \exp\big(\lambda^2(K^2 - t^2)\big).\end{aligned}$$

Choosing $\lambda = 1/K$, we have $\exp\big(1-(t/K)^2\big) \leqslant \delta \iff t \geqslant K\sqrt{\log(e/\delta)}$ and the result follows. □

By Theorem 5.1, the Markov chain $(\theta_t)_{t\geqslant 0}$ is geometrically converging to the invariant distribution $\pi_{\beta,p}$ w.r.t. the Total Variation distance so that for any event $\mathcal{E} \in \mathcal{B}(\mathbb{R}^d)$, we have:

$$\big|\mathbb{P}\big(\theta_T \in \mathcal{E}\big) - \mathbb{P}_{\theta\sim\pi_{\beta,p}}\big(\theta \in \mathcal{E}\big)\big| \leqslant M\rho^T V(\theta_0). \tag{5.9.22}$$

Proposition 5.2 states that, in the absence of corruption, for $\theta \sim \pi_{\beta,p}$, the variable $\|\theta - \theta^\star\|$ is sub-Gaussian with constant $K = 4\sqrt{\frac{2\beta(B_q^2+\bar{\tau}^2)}{p\mu}}$. It is only left to combine this conclusion with Lemma 5.4 in order to obtain the claimed bound.

Proof of Corollary 5.2

We assume without loss of generality that T is a multiple of N. Note that according to the assumptions, the estimators $\theta_T^{(n)}$ for $n \in [\![N]\!]$ are independent and for each n. For positive $\epsilon < 1$ define the events $E_n := \big\{\|\theta_T^{(n)} - \theta^\star\| \leqslant \frac{\eta^{1-\frac{1}{q}} B_q \sqrt{20}}{\kappa\epsilon}\big\}$. We first assume that $\sum_{n=1}^N \mathbf{1}_{E_n} > N/2$ then there exists $i' \in [\![N]\!]$ such that

$$r_{N/2}^{(\widehat{i})} = \big\|\theta_T^{(\widehat{i})} - \theta_T^{(i')}\big\| \leqslant \big\|\theta_T^{(\widehat{i})} - \theta^\star\big\| + \big\|\theta_T^{(i')} - \theta^\star\big\| \leqslant 2\frac{\eta^{1-\frac{1}{q}} B_q \sqrt{20}}{\kappa\epsilon}.$$

Moreover, among the $N/2$ estimators closest to $\theta_T^{(\widehat{i})}$, at least one of them $\theta_T^{(i'')}$ satisfies $\|\theta_T^{(i'')} - \theta^\star\| \leqslant \frac{\eta^{1-\frac{1}{q}} B_q \sqrt{20}}{\kappa\epsilon}$ thus we find :

$$\begin{aligned}\big\|\widehat{\theta} - \theta^\star\big\| = \big\|\theta_T^{(\widehat{i})} - \theta^\star\big\| &\leqslant \big\|\theta_T^{(\widehat{i})} - \theta_T^{(i'')}\big\| + \big\|\theta_T^{(i'')} - \theta^\star\big\| \\ &\leqslant r_{N/2}^{(\widehat{i})} + \frac{\eta^{1-\frac{1}{q}} B_q \sqrt{20}}{\kappa\epsilon} \leqslant 3\frac{\eta^{1-\frac{1}{q}} B_q \sqrt{20}}{\kappa\epsilon}. \end{aligned} \tag{5.9.23}$$

Notice that setting $\epsilon = 1/2$ immediately yields (5.3.6). We now show that $\sum_{n=1}^N \mathbf{1}_{E_n} > N/2$ happens with high probability. Thanks to Theorem 5.1 and Proposition 5.1, we have:

$$\mathbb{P}(\overline{E}_n) \leqslant \epsilon^2 + \underbrace{\rho^{T/N}\big(1 + \|\theta_0 - \theta^\star\|^2\big)}_{\epsilon'}.$$

Consequently, the variables $\mathbf{1}_{\overline{E}_n}$ are stochastically dominated by Bernoulli variables with parameter $\epsilon^2 + \epsilon'$ so that their sum is stochastically dominated by a Binomial random variable $S := \mathrm{Bin}(N, \epsilon^2 + \epsilon')$. We compute :

$$\begin{aligned}\mathbb{P}\Big(\sum_{n=1}^N \mathbf{1}_{E_n} < N/2\Big) = \mathbb{P}\Big(\sum_{n=1}^N \mathbf{1}_{\overline{E}_n} > N/2\Big) &\leqslant \mathbb{P}\big(S - \mathbb{E}S > N/2 - (\epsilon^2 + \epsilon')N\big) \\ &\leqslant \exp\big(-2N(1/2 - \epsilon^2 - \epsilon')^2\big) \\ &\leqslant \exp\big(-2N(1/2 - 1/4 - M\rho^{T/N}(1 + \|\theta_0 - \theta^\star\|^2))^2\big) \\ &\leqslant \exp\big(-2N(1/4 - 1/15)^2\big) \leqslant \exp\big(-\log(1/\delta)\big) = \delta\end{aligned}$$

where we used Hoeffding's inequality, the choice $\epsilon = 1/2$ and the fact that our condition on T implies $M\rho^{T/N}(1 + \|\theta_0 - \theta^\star\|^2))^2 \leqslant 1/15$. The last inequalities result from our condition on N.

Proof of Theorem 5.2

As previously done in the proof of Theorem 5.1, we show that the Markov chain is aperiodic. We will now show that it satisfies a drift property. Let $\theta \in \Theta$ be fixed, we have:

$$\begin{aligned}\mathbb{E}\big[\mathcal{L}\big(\theta - \alpha_\theta \beta G(\theta)\big)\big] - \mathcal{L}(\theta) &\overset{①}{\leqslant} \mathbb{E}\Big[\eta\Big(-\beta\big\langle\nabla\mathcal{L}(\theta), \alpha_\theta\check{G}(\theta)\big\rangle + \frac{L\beta^2}{2}\big\|\alpha_\theta\check{G}(\theta)\big\|^2\Big) + \\ &\qquad (1-\eta)\Big(-\beta\big\langle\nabla\mathcal{L}(\theta), \alpha_\theta\widetilde{G}(\theta)\big\rangle + \frac{L\beta^2}{2}\big\|\alpha_\theta\widetilde{G}(\theta)\big\|^2\Big)\Big] \\ &\overset{②}{\leqslant} \frac{L\beta^2\tau_\theta^2}{2} - \eta\beta\big\langle\nabla\mathcal{L}(\theta), \mathbb{E}[\alpha_\theta\check{G}(\theta)]\big\rangle - \beta(1-\eta)\overline{\alpha}_\theta\big\|\nabla\mathcal{L}(\theta)\big\|^2 \\ &\qquad - \beta(1-\eta)\big\langle\nabla\mathcal{L}(\theta), \mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\rangle \\ &\overset{③}{\leqslant} -\beta(1-\eta)\overline{\alpha}_\theta\|\nabla\mathcal{L}(\theta)\|^2 + \frac{L\beta^2\tau_\theta^2}{2} + \eta\beta\tau_\theta\|\nabla\mathcal{L}(\theta)\| \\ &\qquad + \beta(1-\eta)\|\nabla\mathcal{L}(\theta)\|\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\| \\ &\overset{④}{\leqslant} -\beta\|\nabla\mathcal{L}(\theta)\|^2\big((1-\eta)\overline{\alpha}_\theta - L\beta - \eta\big) + L\beta^2 Q_p(\|\varepsilon_\theta\|)^2 \\ &\qquad + \beta\|\nabla\mathcal{L}(\theta)\|\big(\eta Q_p(\|\varepsilon_\theta\|) + (1-\eta)\big\|\mathbb{E}[\alpha_\theta\widetilde{G}(\theta)] - \overline{\alpha}_\theta\nabla\mathcal{L}(\theta)\big\|\big) \\ &\overset{⑤}{\leqslant} -\beta\|\nabla\mathcal{L}(\theta)\|^2\big((1-\eta)\overline{\alpha}_\theta - L\beta - \eta - \epsilon/2\big) + L\beta^2 Q_p(\|\varepsilon_\theta\|)^2\end{aligned}$$

$$
\begin{aligned}
&+ \frac{\beta}{2\epsilon}\big(\eta Q_p(\|\varepsilon_\theta\|) + (1-\eta)\big\|\mathbb{E}[\alpha_\theta \widetilde{G}(\theta)] - \overline{\alpha}_\theta \nabla\mathcal{L}(\theta)\big\|\big)^2 \\
&\overset{⑥}{\leqslant} -\beta\|\nabla\mathcal{L}(\theta)\|^2\big(3p(1-\eta)/4 - L\beta - \eta\big) + L\beta^2(1-p)^{-\frac{2}{q}}B_q^2 \\
&\quad + \frac{\beta B_q^2\big(\eta(1-p)^{-\frac{1}{q}} + (1-\eta)\eta^{1-\frac{1}{q}}\big)^2}{p(1-\eta)} \\
&\leqslant -\beta\|\nabla\mathcal{L}(\theta)\|^2\big(3p(1-\eta)/4 - L\beta - \eta\big) \\
&\quad + \frac{\beta B_q^2\big((1-p)^{-\frac{2}{q}}(L\beta + 2\eta^2) + 2\eta^{2-\frac{2}{q}}\big)}{p(1-\eta)},
\end{aligned} \tag{5.9.24}
$$

where ① uses Assumptions 5.1 and 5.3, ② and ③ use that $\|\alpha_\theta\widetilde{G}(\theta)\|, \|\alpha_\theta\breve{G}(\theta)\| \leqslant \tau_\theta$, ④ uses that $\tau_\theta \leqslant \|\nabla\mathcal{L}(\theta)\| + Q_p(\|\varepsilon_\theta\|)$ (see Lemma 5.2), ⑤ uses Fact 1 and ⑥ uses that $\overline{\alpha}_\theta \geqslant p$, the choice $\epsilon = p(1-\eta)/2$ and Lemma 5.2. By assumption, we have $3p(1-\eta)/4 - L\beta - \eta > 0$.

Define the quantity $\xi = \frac{B_q^2\left((1-p)^{-\frac{2}{q}}(L\beta+2\eta^2)+2\eta^{2-\frac{2}{q}}\right)}{p(1-\eta)}$ and the set $\mathcal{C} := \Big\{\theta \in \mathbb{R}^d : \frac{1}{2}\|\nabla\mathcal{L}(\theta)\|^2 \leqslant \frac{\xi}{3p(1-\eta)/4 - L\beta - \eta}\Big\}$. By assumption, $\mathcal{C}$ is bounded and it is clear that the right hand side in (5.9.24) is negative outside $\mathcal{C}$. Define the function $V(\theta) = \mathcal{L}(\theta)/(\beta\xi)$, which is positive and satisfies:

$$
\Delta V(\theta) \leqslant -1 + 2 \cdot \mathbf{1}_{\theta\in\mathcal{C}}. \tag{5.9.25}
$$

In addition, we show similarly to Theorem 5.1 that the set $\mathcal{C}$ is *small* and, therefore, also *petite* according to the definitions of [313, Chapter 5]). Since V is everywhere finite and bounded on $\mathcal{C}$ (because the latter is compact), the conditions of [313, Theorem 11.3.4] are fulfilled implying that the chain is Harris recurrent.

We have shown that the Markov chain verifies the fourth condition of [313, Theorem 13.0.1]. This allows us to conclude that the Markov chain is ergodic i.e. we have for any initial measure λ that $\|\lambda P^t - \pi_{\beta,p}\|_{\mathrm{TV}} \to 0$ and the following sum is finite

$$
\sum_t \big\|\lambda P_{\beta,p}^t - \pi_{\beta,p}\big\|_{\mathrm{TV}} < \infty.
$$

In addition, by [313, Proposition 13.3.2] the terms in the above sum are non-increasing which implies that $\big\|\lambda P_{\beta,p}^t - \pi_{\beta,p}\big\|_{\mathrm{TV}} = \mathcal{O}(t^{-1})$ and the result follows.

Proof of Proposition 5.3

By Theorem 5.2, the assumptions imply that the Markov chain $(\theta_t)_{t\geqslant 0}$ converges to an invariant distribution $\pi_{\beta,p}$. For $\theta \sim \pi_{\beta,1-\eta}$, by invariance of $\pi_{\beta,1-\eta}$, we have that the variables $\mathcal{L}(\theta - \alpha_\theta\beta G(\theta))$ and $\mathcal{L}(\theta)$ are identically distributed. Taking the expectation w.r.t. θ, this implies the identity $\mathbb{E}[\mathcal{L}(\theta - \alpha_\theta\beta G(\theta))] = \mathbb{E}[\mathcal{L}(\theta)]$. Plugging into Inequality (5.9.24) from the proof of Theorem 5.2, we find

$$
\begin{aligned}
\mathbb{E}\big[\|\nabla\mathcal{L}(\theta)\|^2\big] &\leqslant \frac{B_q^2\big((1-p)^{-\frac{2}{q}}(L\beta + 2\eta^2) + 2\eta^{2-\frac{2}{q}}\big)}{p(1-\eta)\big(3p(1-\eta)/4 - L\beta - \eta\big)} \\
&\leqslant \frac{B_q^2\big(3(1-p)^{-\frac{2}{q}}\eta^2 + 2\eta^{2-\frac{2}{q}}\big)}{p(1-\eta)\big(3p(1-\eta)/4 - L\beta - \eta\big)} \leqslant \frac{5B_q^2\eta^{2-\frac{2}{q}}}{p(1-\eta)\big(3p(1-\eta)/4 - L\beta - \eta\big)}
\end{aligned}
$$

where we used the choices $\beta \leqslant \frac{\eta^2}{L}$ and $p = 1-\eta$.

Chapter 6

Convergence and Concentration Properties of SGD

This chapter is based on the article [311] in collaboration with Stéphane Gaïffas.

Contents

Abstract

We consider the optimization of a smooth and strongly convex objective using constant step-size stochastic gradient descent (SGD) and study its properties through the prism of Markov chains. We show that, for unbiased gradient estimates with mildly controlled variance, the iteration converges to an invariant distribution in total variation distance. We also establish this convergence in Wasserstein-2 distance in a more general setting compared to previous work. Thanks to the invariance property of the limit distribution, our analysis shows that the latter *inherits* sub-Gaussian or sub-exponential concentration properties when these hold true for the gradient. This allows the derivation of high-confidence bounds for the final estimate. Finally, under such conditions in the linear case, we obtain a dimension-free deviation bound for the Polyak-Ruppert average of a tail sequence. All our results are non-asymptotic and their consequences are discussed through a few applications.

6.1 Introduction

We consider the following stochastic optimization problem

$$\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta) := \mathbb{E}_\zeta[\ell(\theta,\zeta)], \tag{6.1.1}$$

where $\mathcal{L}$ is a smooth strongly convex objective only accessible through unbiased random gradient samples $G(\theta,\zeta) = \nabla\ell(\theta,\zeta)$ which may be queried at any parameter value $\theta \in \mathbb{R}^d$. Given an initial point θ_0 and a step-size β, problem (6.1.1) is commonly solved using the well-known stochastic gradient descent (SGD) algorithm defined by the iteration

$$\theta_{t+1} = \theta_t - \beta G(\theta_t,\zeta_t), \quad \text{for} \quad t \geqslant 0. \tag{6.1.2}$$

We study the convergence properties of the Markov chain $(\theta_t)_{t\geqslant 0}$ generated by the above iteration as well as the concentration properties satisfied by a derived estimator $\widehat{\theta}$ of the global optimum $\theta^\star = \arg\min_\theta \mathcal{L}(\theta)$ based on the concentration of the gradient samples $G(\theta_t,\zeta_t)$.

Problem (6.1.1) is the common formulation for a large fraction of statistical learning problems where the objective $\mathcal{L}(\theta)$ is defined as the expectation of a loss function ℓ over a random variable ζ following an unknown distribution of samples. In a practical setting, the random gradients $G(\theta_t,\zeta_t)$ are computed using a dataset of independent and identically distributed samples $(\zeta_i)_{i=1}^n$. The SGD algorithm is employed to solve (6.1.1) in two situations. Either the samples $(\zeta_i)_{i=1}^n$ are available offline but in such a great amount that using the whole dataset at each gradient step incurs an excessive computational load, therefore, individual samples or small batches are used at each iteration instead. Or, the samples ζ_i are received individually in an online fashion and optimization must be run using one instance at a time. Our framework covers both cases provided that each iteration uses new data which is independent from the past.

Thanks to its simplicity and efficiency, the SGD algorithm is widely adopted as the go-to approach for stochastic optimization problems in general. Since its first appearance in the seminal work of [374] the theoretical properties of SGD have been investigated in a series of pioneering works [96, 388, 144]. A notable milestone in these theoretical developments was the discovery of Polyak-Ruppert averaging [386, 358] which allows to reduce the impact of noise and improve the convergence rate for certain cases of interest. The subject benefited from a growing attention with the advent of complex machine learning models such as neural networks and a rich literature has appeared to address the surfacing questions about SGD and its numerous variants and use cases [402, 16, 324, 46, 334].

Although the basic definition of the SGD iteration (6.1.2) is quite simple, a great number of variations are possible by playing on various aspects among which the choice of step-size is critical. Early work [374] suggested a decaying step-size of order t^{-1} but this leads to poor dependence on problem conditioning [16, 144] while other step-size schedules with slower decay of order $t^{-\alpha}$ with $\alpha \in (1/2, 1]$ combined with averaging achieve better practical and theoretical performance [386, 358, 324]. In this work, we consider constant step-size SGD which is also a commonly adopted choice due to its usually fast convergence [393, 430, 293].

6.1.1 Main contributions

This paper studies constant step-size SGD as a Markov chain and makes the following contributions.

- We state two convergence results of the Markov chain to an invariant distribution. The first ergodicity theorem states convergence in total variation distance and the second one in terms of the Wasserstein-2 distance. While similar results exist in the literature, our version for the Wasserstein convergence mode holds in a more general setting than previous results.
- We show that sub-Gaussian and sub-exponential concentration of the gradient samples imply the same property for the invariant limit distribution with Ψ_1/Ψ_2 constant proportional to the step-size. Thanks to this property, we obtain high-confidence deviation bounds on the final SGD iterate.
- Provided a slightly stronger concentration assumption on the gradient samples, we show similar but dimension-free high-confidence bounds on the last SGD iterate.
- Finally, for the special case of a linear gradient, we obtain a high-confidence dimension-free bound for the Polyak-Ruppert average of a tail sequence of the SGD iterates. This is achieved, in part, thanks to a more generic concentration result which holds for any Lipschitz function applied to a stationary sequence.

All our results are non-asymptotic.

6.1.2 Related works

SGD as a Markov chain. A fairly limited portion of the SGD literature adopts the Markov chain approach. Among the earliest, [357] studied the iteration in question for vanishing step-size, while [17] considers constant step-size averaged SGD for non-strongly-convex smooth objectives and shows L_p convergence of the excess risk for all $p \geqslant 1$. Although their analysis does not use Markov chain theory, they discuss properties of the invariant distribution which the iteration converges to, including a few properties we state in this paper. However, they do not derive high-confidence estimation bounds as we do. More recently, convergence in Wasserstein distance was established by [123] for constant step-size SGD applied to a strongly convex and smooth objective, albeit under a co-coercivity condition which is hard to establish in the nonlinear case. Further, an expansion of the asymptotic moments of averaged SGD is provided in [123] and the Richardson-Romberg extrapolation strategy is studied which allows to reduce the estimation error on the global optimum. Most recently, [446] studied SGD run on a non-convex, non-smooth but quadratically growing objective. Under such weakened conditions, they show that the generated Markov chain is geometrically ergodic (see [313]) and proceed to establish a CLT for the generated Markov iterates. They also state results controlling the bias of the limit distribution under additional assumptions such as convexity, L_4 control of the gradient noise and a generalized Łojasiewicz condition [229].

High probability bounds. In addition to establishing the convergence of SGD in expectation, the works of [371, 17, 16] make the further step of stating high-confidence bounds on the final error. Still, sub-Gaussian concentration only holds under strong bounded gradient assumptions. High-confidence deviation results are also stated in [164] where an accelerated stochastic optimization method for strongly convex composite objectives is studied. However, the bounds are sub-exponential while the gradient is assumed to be sub-Gaussian.

In [228], high probability bounds are proved for the PEGASOS algorithm using Freedman's inequality for martingales [150]. A generalization of the said inequality was used by [179, 178] to prove such bounds for SGD in the non-smooth strongly convex case. Most recently, for a careful choice of step-size, [210] obtained high-confidence results on the last SGD iterate. Unfortunately, both previous works require a deterministic bound to hold over the gradient or its noise which strongly constrains their applicability. Finally, a high probability analysis of Delayed AdaGrad with momentum was presented by [270] in the smooth non-convex setting.

Note that certain recent works design *robust* variants of SGD achieving sub-Gaussian deviation bounds on the last iterate with only a second moment assumption on the gradient [417, 166]. Similar results were later obtained under even weaker gradient moment assumptions [389, 346]. However, in this work, we focus on the *classical* SGD algorithm and the properties inherited by its iterates from the gradient samples.

Polyak-Ruppert averaging. The averaging procedure introduced by [358, 386] was also studied by [172, 104] who proved asymptotic convergence properties. Non-asymptotic results and additional developments appeared in the works of [324, 121, 124, 209, 208, 253] with particular attention to least-squares, logistic regression and kernel-based methods. In particular, non-asymptotic results of convergence in expectation were obtained for averaged SGD in [124, 253, 345, 371]. Among such results, some demonstrate the advantages of special averaging schemes [402, 250]. Finally, some relatively recent works obtained high probability concentration bounds for Polyak Ruppert averaging with and without sub-Gaussian assumptions on the data [323, 278].

6.1.3 Paper organization

Section 6.2 lays out the basic setting and assumptions necessary for SGD convergence. Section 6.3 states our first SGD ergodicity result. In Section 6.4, we first state a basic result on the invariant measure's expectation, bias and variance and proceed to derive concentration properties based on analogous assumptions on the gradient. Section 6.5 presents an additional convergence result in Wasserstein distance. In Section 6.6, we give deviation bounds on the final SGD iterate which follow from preceding results. We also formulate our high-confidence bound on a tail Polyak-Ruppert average for the linear case. Finally, we discuss a few applications in Section 6.7 and conclude.

6.2 Setting and notations

Let Θ denote either a convex subset of $\mathbb{R}^d$ or $\mathbb{R}^d$ itself depending on context. We refer to the Borel σ-algebra of $\mathbb{R}^d$ as $\mathcal{B}(\mathbb{R}^d)$. For any random variable X, we denote $\mathcal{D}(X)$ its distribution. We refer to the space of square-integrable measures on $\mathbb{R}^d$ as $\mathcal{P}_2(\mathbb{R}^d)$. We denote $\mathcal{M}_1(\mathbb{R}^d)$ the set of probability measures over $\mathbb{R}^d$. For real numbers a and b, we denote $\min(a,b) = a \wedge b$ and $\max(a,b) = a \vee b$. We denote $\mathrm{Lip}(\mathcal{X})$ the set of 1-Lipschitz functions $h : \mathcal{X} \to \mathbb{R}$. For $p \in \mathbb{N}^*$, we denote $\|X\|_{L_p} = (\mathbb{E}|X|^p)^{1/p}$ the L_p norm of a random variable X.

In the entirety of this work, we assume that $\mathcal{L}$ satisfies

Assumption 6.1. *There exist positive constants* $0 < \mu \leqslant L < +\infty$ *such that*

$$\frac{\mu}{2}\|\theta - \theta'\|^2 \leqslant \mathcal{L}(\theta) - \mathcal{L}(\theta') - \langle \nabla\mathcal{L}(\theta'), \theta - \theta'\rangle \leqslant \frac{L}{2}\|\theta - \theta'\|^2$$

for all $\theta, \theta' \in \mathbb{R}^d$, *i.e.* $\mathcal{L}$ *is* L *gradient-Lipschitz and* μ*-strongly convex.*

As an immediate consequence, $\mathcal{L}$ admits a unique minimum $\theta^\star$ which is a critical point:

$$\theta^\star = \arg\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta) \quad \text{such that} \quad \nabla\mathcal{L}(\theta^\star) = 0. \tag{6.2.1}$$

For an initial $\theta_0 \in \Theta$, step-size $\beta > 0$ and all $t \geqslant 0$, we recall the basic SGD iteration

$$\theta_{t+1} = \theta_t - \beta G(\theta_t, \zeta_t). \tag{6.2.2}$$

In this work, we consider constant step-size SGD so that β is fixed along the iteration. We require some basic assumptions on the samples $G(\theta_t, \zeta_t)$ in order to prove the convergence of SGD. Namely, $G(\theta_t, \zeta_t)$ needs to be an unbiased estimator of the true gradient $\nabla\mathcal{L}(\theta_t)$ with controlled variance as we formally state in

Assumption 6.2. *Given any parameter* $\theta \in \Theta$, *the random gradient sample* $G(\theta, \zeta)$ *can be written as*

$$G(\theta, \zeta) = \nabla\mathcal{L}(\theta) + \varepsilon_\zeta(\theta), \tag{6.2.3}$$

where $\varepsilon_\zeta(\theta)$ *is a centered noise i.e.* $\mathbb{E}\big[\varepsilon_\zeta(\theta)|\theta\big] = 0$ *whose distribution can be written as* $\mathcal{D}(\varepsilon_\zeta(\theta)) = \delta\nu_{\theta,1} + (1-\delta)\nu_{\theta,2}$ *with* $\delta > 0$ *and* $\nu_{\theta,1}, \nu_{\theta,2}$ *two probability distributions over* $\mathbb{R}^d$ *such that* $\nu_{\theta,1}$ *admits a density* $h(\theta, \cdot)$ *w.r.t. Lebesgue's measure satisfying*:

$$\inf_{\omega\in S} h(\theta, \omega) > 0 \quad \textit{for all } \theta \textit{ and compact } S \subset \mathbb{R}^d.$$

Moreover, there are positive constants L_σ *and* σ^2 *such that for all* θ *we have*

$$\mathbb{E}\big[\|\varepsilon_\zeta(\theta)\|^2 \mid \theta\big] = \mathbb{E}\big[\|G(\theta, \zeta) - \nabla\mathcal{L}(\theta)\|^2 \mid \theta\big] \leqslant L_\sigma\|\theta - \theta^\star\|^2 + \sigma^2. \tag{6.2.4}$$

The additional assumptions on the distribution of the noise $\varepsilon_\zeta(\theta)$ are needed in order to establish the ergodicity of the resulting Markov chain $(\theta_t)_{t\geqslant 0}$ (Theorem 6.1 below) by ensuring that the associated transition kernel satisfies a *minorization* property implying that the chain will sufficiently explore the state space, see [313] for more details. Note also that these requirements are fairly mild: they only require that the noise distribution admits a diffuse component.

6.3 Markov Chain and Geometric Ergodicity

Before stating the convergence result for the SGD Markov chain we introduce some further useful notation. For a given step-size $\beta > 0$, we will denote P_β the Markov transition kernel governing the Markov chain $(\theta_t)_{t\geqslant 0}$ generated by iteration (6.2.2), so that for any $t \geqslant 0$ and $A \in \mathcal{B}(\mathbb{R}^d)$ we have:

$$\mathbb{P}(\theta_{t+1} \in A \mid \theta_t) = P_\beta(\theta_t, A).$$

The transition kernel P_β acts on probability distributions $\nu \in \mathcal{M}_1(\mathbb{R}^d)$ through the mapping $\nu \to \nu P_\beta$ which is defined, for all $A \in \mathcal{B}(\mathbb{R}^d)$, by $\nu P_\beta(A) = \int_A P_\beta(\theta, A)d\nu(\theta) = \mathbb{P}(\theta_{t+1} \in A \mid \theta_t \sim \nu)$. For $n \geqslant 1$, we similarly define the multi-step transition kernel P_β^n which is such that $P_\beta^n(\theta_t, A) = \mathbb{P}(\theta_{t+n} \in A \mid \theta_t)$ and acts on probability distributions $\nu \in \mathcal{M}_1(\mathbb{R}^d)$ through

$\nu P_\beta^n = (\nu P_\beta) P_\beta^{n-1}$. Finally, we define the total-variation norm of a signed measure ν as

$$2\|\nu\|_{\mathrm{TV}} = \sup_{f:|f|\leqslant 1} \int f(\theta)\nu(d\theta) = \sup_{A\in\mathcal{B}(\mathbb{R}^d)} \nu(A) - \inf_{A\in\mathcal{B}(\mathbb{R}^d)} \nu(A).$$

In particular, we recover the total-variation *distance* between two probability distributions $\nu_1, \nu_2 \in \mathcal{M}_1(\mathbb{R}^d)$ as $d_{\mathrm{TV}}(\nu_1,\nu_2) = \|\nu_1 - \nu_2\|_{\mathrm{TV}}$. We are now ready to state the geometric ergodicity result for the SGD Markov chain.

Theorem 6.1. *Under Assumptions 6.1 and 6.2, the Markov chain $(\theta_t)_{t\geqslant 0}$ defined by iteration* (6.2.2) *with step-size*

$$\beta < \frac{2\mu}{\mu^2 + (\mu L \vee L_\sigma)}$$

converges geometrically to a unique invariant measure π_β. Namely, for any initial $\theta_0 \in \mathbb{R}^d$, there exist $\rho < 1$ and $M < +\infty$ such that

$$\left\|\delta_{\theta_0} P^n - \pi_\beta\right\|_{\mathrm{TV}} \leqslant M\rho^n\big(1 + \|\theta_0 - \theta^\star\|^2\big), \tag{6.3.1}$$

where δ_{θ_0} is the Dirac measure located at θ_0.

The proof of Theorem 6.1 is given in Section 6.9.2 and is based on [313, Theorem 15.0.1] and a *drift* condition in terms of a Lyapunov function. A similar method was previously used in [446] to establish the convergence of SGD for non-convex, non-smooth objectives with quadratic growth. However, the focus in [446] is on proving a central limit theorem for the Markov sequence $(\theta_t)_{t\geqslant 0}$ and bounding the invariant distribution's bias under certain regularity conditions. In contrast, we focus on proving concentration properties of the invariant distribution π_β which allow us to obtain non-asymptotic deviation bounds on the estimation of the optimum $\theta^\star$.

One of the main questions which arise after the statement of Theorem 6.1 is regarding the speed of the claimed convergence. It is clear that the contraction factor ρ and the constant M mainly depend on the initial state θ_0 and the step-size β, however, a precise quantification is lacking.

The issue is closely related to renewal theory and can be traced back to Kendall's theorem [232] and more generally concerns Markov chains satisfying a drift property. A rich literature [381, 375, 291, 380, 398, 377, 135, 23, 314, 215, 26] deals with the matter. Near optimal results were obtained for stochastically ordered Markov Processes [291, 377, 292, 156]. Other examples especially amenable to such analysis include reversible Markov chains [113, 112, 215, 376] and chains satisfying special assumptions on their renewal distribution [29, 290, 233]. However, the SGD Markov chain does not satisfy such criteria. An estimation of ρ may be obtained using results based on renewal theory and Kendall's theorem [232, 23, 26]. However, the resulting estimations are notoriously pessimistic [368]. Indeed, let $\alpha := 1 - \beta\mu$ be the contraction factor in the absence of gradient noise (i.e. simple gradient descent) so that we have

$$\|\theta_t - \beta\nabla\mathcal{L}(\theta_t) - \theta^\star\| \leqslant \alpha\|\theta_t - \theta^\star\| \quad \text{for all} \quad t \geqslant 0,$$

then the worst-case bound on ρ obtained thanks to [23, 26] is such that $1 - \rho \sim (\beta\mu)^3$ which is far worse than the intuitive expectation that $\rho \approx \alpha$. It is unclear whether the previous estimation can be improved through a careful study of the renewal properties of the SGD Markov chain or if a different approach such as the study of the spectral properties of the transition kernel P_β is more appropriate. Nevertheless, we will see in Section 6.5 below that ρ can be estimated close to α under additional conditions by leveraging Wasserstein convergence.

6.4 Invariant Distribution Properties

This section states that concentration properties of the random gradient samples used for SGD *transfer* to the invariant distribution π_β towards which iteration (6.2.2) converges as stated in Theorem 6.1. We begin with a basic statement which holds without additional assumptions and bounds the variance of π_β and its bias w.r.t. the true optimum $\theta^\star$.

Proposition 6.1. *In the setting of Theorem 6.1, let π_β be the invariant measure and $\bar{\theta}_\beta := \mathbb{E}_{\theta\sim\pi_\beta}[\theta]$ be its expectation. For $\theta \sim \pi_\beta$, the invariant distribution π_β satisfies the following properties.*

(*a*) $\mathbb{E}_{\theta\sim\pi_\beta}\big[\nabla\mathcal{L}(\theta)\big] = 0$. *In particular, if the gradient $\nabla\mathcal{L}$ is linear (see Assumption 6.6 below) then we have $\bar{\theta}_\beta = \theta^\star$.*

(*b*) *The variance and the bias are bounded as follows*:

$$\mathrm{Var}_{\pi_\beta}(\theta) \leqslant \frac{\beta\sigma^2}{2\mu - \beta(\mu^2 + L_\sigma)} \quad \text{and} \quad \big\|\bar{\theta}_\beta - \theta^\star\big\| \leqslant \sqrt{\mathrm{Var}_{\pi_\beta}(\theta)}.$$

Proposition 6.1 is proven in Section 6.9.2 and relies on the invariance property of π_β and the contraction property of the optimisation iteration. The second part of this statement reflects the known-fact that the iterates have an asymptotic magnitude of $\sqrt{\beta}$ [357, 333].

Before stating further results, we need to define sub-Gaussian and sub-exponential concentration properties for real random variables. Among the many known equivalent characterizations, we only introduce those required for the proofs of our results, see [432, Chapter 2] for other characterizations.

Definition 6.1. *Let X be a real random variable. We say that X is K-sub-Gaussian for some $K > 0$ whenever*

(*i*) *we have*

$$\mathbb{E}\exp(\lambda^2 X^2) \leqslant e^{\lambda^2 K^2} \quad \text{for} \quad 0 \leqslant \lambda \leqslant 1/K, \tag{6.4.1}$$

which we will denote $X \in \widetilde{\Psi}_2(K)$,

(*ii*) *or we have*

$$\mathbb{E}\exp(\lambda X) \leqslant \exp(\lambda^2 K^2) \quad \text{for all} \quad \lambda \in \mathbb{R}, \tag{6.4.2}$$

which we will denote $X \in \Psi_2(K)$.

Definition 6.2. *Let X be a real random variable. We say that X is sub-exponential if one of the two following conditions holds.*

(*i*) *There exists $K_1 > 0$ such that*

$$\|X\|_{L_p} \leqslant K_1 p \quad \text{for all} \quad p \geqslant 1, \tag{6.4.3}$$

in which case we write $X \in \widetilde{\Psi}_1(K_1)$.

(*ii*) *There exists K_2 such that*

$$\mathbb{E}\exp(\lambda X) \leqslant \exp(\lambda^2 K_2^2) \quad \text{for all} \quad |\lambda| \leqslant 1/K_2, \tag{6.4.4}$$

in which case we write $X \in \Psi_1(K_2)$.

Note that, for a centered variable X, the two characterizations of Definition 6.1 are equivalent with the same constant K. Analogously, for centered X, we have that $X \in \Psi_1(K)$ entails $X \in \widetilde{\Psi}_1(2eK)$ and $X \in \widetilde{\Psi}_1(K)$ entails $X \in \Psi_1(2eK)$. Namely, the two characterizations of Definition 6.2 imply each other but with worse constants (see [432, Section 2] for a proof). Since the constants in Definition 6.2 degrade by switching between the two properties, we will specify which property is meant in each subsequent statement in order to minimize these degradations.

We first formulate a sub-Gaussian/sub-exponential concentration assumption on the norms of the gradient errors.

Assumption 6.3. *There exists $\overline{K} < +\infty$ such that one of the following holds*:

(*a*) *For all $\theta \in \Theta$, the gradient error satisfies $\|\varepsilon_\zeta(\theta)\| \in \widetilde{\Psi}_2(\overline{K})$.*

(*b*) *For all $\theta \in \Theta$, the gradient error satisfies $\|\varepsilon_\zeta(\theta)\| \in \widetilde{\Psi}_1(\overline{K})$.*

In combination with Assumptions 6.1 and 6.2, the above pair of assumptions imply the following concentration properties for π_β.

Proposition 6.2. *In the setting of Theorem 6.1, for $\theta \sim \pi_\beta$, the invariant distribution π_β satisfies the following properties*:

(*a*) *If Assumption 6.3 (a) holds then $\|\theta - \theta^\star\| \in \widetilde{\Psi}_2\big(\overline{K}\sqrt{8\beta/\mu}\big)$.*

(*b*) *If Assumption 6.3 (b) holds and if $\beta \leqslant (2\mu)^{-1}$ then $\|\theta - \theta^\star\| \in \widetilde{\Psi}_1\big(2\overline{K}\sqrt{\beta/\mu}\big)$.*

The proof of Proposition 6.2 is given in Section 6.9.2. The most important aspect of this statement is that the quantity $\|\theta - \theta^\star\|$ is sub-Gaussian/sub-exponential with a constant *depending* on the step-size β. Indeed, it is fairly easy to show, for example, that $\|\theta - \theta^\star\| \in \widetilde{\Psi}_2(\overline{K}/\mu)$ under Assumption 6.3 (a), however, this constant is too pessimistic since it fails to take advantage of a small step-size which leads to stronger concentration. The improved constants above are obtained by carefully leveraging the unbiased property of the gradient error (see Assumption 6.2). Previous characterizations of π_β obtained bounds on the bias w.r.t. $\theta^\star$ [446] and moment expansions of $\widehat{\theta} - \theta^\star$ for $\widehat{\theta} \sim \pi_\beta$ or $\widehat{\theta}$ equal to a Polyak-Ruppert average [123], however, the sub-exponential and sub-Gaussian characterizations of Proposition 6.2 appear to be new.

Note that the constant $\overline{K}$ obtained from Assumption 6.3 and appearing in Proposition 6.2 may hide a dependence on the dimension in $\sqrt{d}$ since it is related to the Euclidean norm $\|\varepsilon_\zeta(\theta)\|$ of the gradient noise. In this respect, Proposition 6.2 resembles the results of [301] where similar hypotheses to Assumption 6.3 were used entailing the same dimension dependence. In order to avoid this shortcoming, one needs a stronger assumption which is stated along with the associated results further below. Note also that Assumption 6.3 considerably strengthens Assumption 6.2 by requiring that $\|\varepsilon_\zeta(\theta)\|$ admits a finite exponential moment. In addition, the involved bound is uniform w.r.t. θ. However, under a non-uniform finite p-moment assumption, it is still possible to show the following.

Lemma 6.1. *Grant Assumptions 6.1 and 6.2 and assume that there is $K, \underline{K} > 0$ and $p \in \mathbb{N}^*$ such that conditionally on any θ we have*

$$\big\|\|\varepsilon_\zeta(\theta)\|\big\|_{L_p} \leqslant K\|\theta - \theta^\star\| + \underline{K}, \tag{6.4.5}$$

then for step-size β as in Theorem 6.1 and satisfying the additional condition $\beta \leqslant \frac{\mu}{j(\mu^2+K^2)}$ with $j \leqslant p$, the Markov chain $(\theta_t)_{t\geqslant 0}$ converges to an invariant distribution π_β with at least j finite moments.

Lemma 6.1 is proved in Section 6.9.2 and shows that π_β can have as many finite moments as the gradient, provided that the step-size is small enough. This implies that even weaker concentration properties transfer to the invariant distribution. Note that a non-uniform sub-exponential (resp. sub-Gaussian) assumption would correspond to condition (6.4.5) with K replaced by Kp (resp. $K\sqrt{p}$) in which case the condition on β becomes at least $\beta \leqslant \mathcal{O}\big(\mu/(jK^2p)\big)$. This suggests that, for arbitrary p, $\pi_\beta \in L_p$ may only hold in the limit $\beta \to 0$. Lemma 6.1 shares a similarity with [17, Theorem 2] which states an L_p convergence result under a resembling condition on the step-size. However, [17, Theorem 2] uses a uniform L_p condition on the gradient error whereas we allow the upperbound to depend on $\|\theta - \theta^\star\|$ in Inequality (6.4.5).

We now introduce a stronger analog to Assumption 6.3, which will enable later the proof of dimension-free deviation bounds.

Assumption 6.4. *There is $K < +\infty$ such that one of the following holds*:

(*a*) *For all $\theta \in \Theta$ and all $f \in \mathrm{Lip}(\mathbb{R}^d)$, we have $f(G(\theta,\zeta)) - \mathbb{E}f(G(\theta,\zeta)) \in \Psi_2(K)$.*

(*b*) *For all $\theta \in \Theta$ and all $f \in \mathrm{Lip}(\mathbb{R}^d)$, we have $f(G(\theta,\zeta)) - \mathbb{E}f(G(\theta,\zeta)) \in \Psi_1(K)$.*

As announced, the subtle difference with Assumption 6.3 is that the involved constants are, a priori, independent from the dimension. The so-called Bobkov-Götze theorem [42] states that Assumption 6.4 (a) is equivalent to the fact that $\nu_\theta := \mathcal{D}(G(\theta,\zeta))$ satisfies the following *Transportation-Information* inequality

$$\mathcal{W}_1\big(\nu,\nu_\theta\big) \leqslant \sqrt{2K^2 D(\nu\|\nu_\theta)} \quad \text{for all} \quad \nu \in \mathcal{M}_1(\mathbb{R}^d), \tag{6.4.6}$$

where $\mathcal{W}_1$ and $D(\cdot\|\cdot)$ are the Wasserstein-1 distance [434] (see definition below) and the Kullback-Leibler divergence [248] between probability measures respectively. An analogous equivalence may be established for the sub-exponential case of Assumption 6.4 (b) (for instance, by adapting the proof given in [426, Theorem 4.8]).

By restricting the functions f in Assumption 6.4 to be linear, we recover the assumption that the vector $G(\theta,\zeta)$ is sub-Gaussian/sub-exponential. An interesting question is then whether this weaker property implies Assumption 6.4 with a dimension independent constant. To the best of our knowledge of the current literature, this is only known to hold for Gaussian vectors (see for instance [426, Theorem 3.25]). In fact, Talagrand's well-known transport inequality states that Gaussian vectors satisfy Inequality (6.4.6) for the $\mathcal{W}_2$ distance rather than $\mathcal{W}_1$, which is an even stronger property. Since Inequality (6.4.6) involves two very different forms of distance between probability measures, a direct intuitive understanding of its meaning is elusive. However, the above inequality is related to a host of properties used to describe the concentration of measure phenomenon including Poincaré inequalities [43, 167], logarithmic Sobolev inequalities [42, 266] and modified logarithmic Sobolev inequalities [160, 21] to mention only a few. A broad and comprehensive survey on transport inequalities and their consequences on concentration and deviation inequalities is available in [168].

Using the previous assumption, we can show that the invariant distribution inherits similar properties.

Proposition 6.3. *Grant Assumptions 6.1 and 6.2 and let π_β be the invariant distribution obtained for step-size β as in Theorem 6.1. For $\theta \sim \pi_\beta$, the invariant distribution π_β satisfies the following properties.*

(*a*) *If Assumption 6.4 (a) holds then $f(\theta) - \mathbb{E}f(\theta) \in \Psi_2\big(K\sqrt{\beta/\mu}\big)$ for all $f \in \mathrm{Lip}(\mathbb{R}^d)$.*

(*b*) *If Assumption 6.4 (b) holds then $f(\theta) - \mathbb{E}f(\theta) \in \Psi_1\big(K\sqrt{\beta/\mu}\big)$ for all $f \in \mathrm{Lip}(\mathbb{R}^d)$.*

Proposition 6.3 is proven in Section 6.9.2 and will be used in Section 6.6 to derive dimension-free deviation bounds. Note that the Ψ_1/Ψ_2 constants in Proposition 6.3 also display the crucial $\sqrt{\beta/\mu}$ dependence as in Proposition 6.2 and without further degradation. Before proceeding to the statement of such results we explore another convergence mode of the SGD Markov chain.

6.5 Wasserstein Convergence

This section complements Theorem 6.1 with an additional convergence result w.r.t. the Wasserstein metric. We recall that, for $p \geqslant 1$ and two distributions $\varpi, \nu \in \mathcal{M}_1(\mathbb{R}^d)$, the Wasserstein-$p$ distance is defined by

$$\mathcal{W}_p^p(\varpi, \nu) = \inf_{\xi \in \Pi(\varpi,\nu)} \mathbb{E}_{X,Y\sim\xi}\|X - Y\|^p,$$

where $\Pi(\varpi, \nu)$ is the set of all couplings of ϖ and ν i.e. distributions over $\mathbb{R}^d \times \mathbb{R}^d$ with first and second marginals equal to ϖ and ν respectively.

In order to show that the SGD iteration converges w.r.t. the Wasserstein-2 distance, we require the following assumption.

Assumption 6.5. *There is $L_{\mathcal{W}} < +\infty$ such that for all θ, θ', the gradient noise distributions $\mathcal{D}(\varepsilon_\zeta(\theta))$ and $\mathcal{D}(\varepsilon_\zeta(\theta'))$ at θ and θ' satisfy*

$$\mathcal{W}_2^2\big(\mathcal{D}(\varepsilon_\zeta(\theta)), \mathcal{D}(\varepsilon_\zeta(\theta'))\big) \leqslant L_{\mathcal{W}}\|\theta - \theta'\|^2.$$

In words, we assume that the change in the gradient noise distribution measured with the $\mathcal{W}_2$ metric is controlled by the change in the parameter θ. This assumption is discussed below and allows to obtain the following result.

Proposition 6.4. *Grant Assumptions 6.1, 6.2 and 6.5. Let $\nu_1, \nu_2 \in \mathcal{P}_2(\mathbb{R}^d)$ be two initial distributions and let β be a step-size such that*

$$\beta < \frac{2\mu}{\mu^2 + (L_{\mathcal{W}} \vee \mu L)},$$

then we have the contraction

$$\mathcal{W}_2^2(\nu_1 P_\beta, \nu_2 P_\beta) \leqslant \big((1 - \beta\mu)^2 + \beta^2 L_{\mathcal{W}}\big)\mathcal{W}_2^2(\nu_1, \nu_2).$$

Consequently, for such a β and any initial $\theta_0 \sim \nu \in \mathcal{P}_2(\mathbb{R}^d)$, the Markov chain generated by iteration (6.2.2) *converges to a unique stationary measure π_β in $\mathcal{W}_2$ distance.*

The proof of Proposition 6.4 is given in Section 6.9.3. The intuition behind it is that, if the Markov chain evolves according to a locally similar dynamic when started from different points then, for small enough step-size, the contraction phenomenon coming from the optimization will prevail so that trajectories associated to different initializations join even before convergence. A similar result was previously stated in [123, Proposition 2] for smooth and strongly convex functions also. In [123], Wasserstein convergence is shown under the assumption that every random gradient $G(\theta, \zeta)$ be almost surely co-coercive with fixed constant. Nonetheless, the proof also works when this property holds only in expectation (see [123, Assumption A7]), which translates to the following inequality for all θ, θ' :

$$L\langle \nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta - \theta'\rangle \geqslant \mathbb{E}\big[\|G(\theta, \zeta) - G(\theta', \zeta)\|^2\big]. \tag{6.5.1}$$

For the sake of illustration, we consider the simple example of least-squares linear regression in

which, given a sample $\zeta = (X, \xi) \in \mathbb{R}^d \times \mathbb{R}$, a random gradient is computed as

$$G(\theta, \zeta) = XX^\top \theta - XY \quad \text{with} \quad Y = X^\top \theta^\star + \xi,$$

where ξ is an independent centered noise. In this particular case, Inequality (6.5.1) can be verified as long as X has a bounded fourth moment. Regarding Assumption 6.5, we have $\varepsilon_\zeta(\theta) = G(\theta, \zeta) - \nabla\mathcal{L}(\theta) = (XX^\top - \mathbb{E}XX^\top)(\theta - \theta^\star) - X\xi$ and it is easy to couple the distributions of $\varepsilon_\zeta(\theta')$ and $\varepsilon_{\zeta'}(\theta')$ by defining them with the same variables $\zeta = \zeta' = (X, \xi)$ so that we find

$$\mathcal{W}_2^2\big(\mathcal{D}(\varepsilon_\zeta(\theta)), \mathcal{D}(\varepsilon_{\zeta'}(\theta'))\big) \leqslant \mathbb{E}\|\varepsilon_\zeta(\theta) - \varepsilon_{\zeta'}(\theta')\|^2 \leqslant \mathbb{E}\|XX^\top - \mathbb{E}XX^\top\|_2^2 \|\theta - \theta'\|^2,$$

where $\|\cdot\|_2$ is the operator norm. We then recover the bounded fourth moment condition on X.

More generally, we can consider an objective $\mathcal{L}$ defined by a linear learning task such that $\mathcal{L}(\theta) = \mathbb{E}_{(X,Y)}[\ell(X^\top\theta, Y)]$ for a convex loss ℓ and random samples and labels (X, Y) so that the gradient samples are $G(\theta, (X, Y)) = X\ell'(X^\top\theta, Y)$ with ℓ' the derivative in the first argument. By a similar computation as above, it is possible to verify Assumption 6.5 as soon as ℓ is smooth in its first argument and X has a finite fourth moment. On the other hand, it is unclear how to establish (6.5.1) in this setting which makes Assumption 6.5 more generic.

In the same vein as Assumption 6.5, it is possible to introduce a regularity condition on the transition kernel P_β in terms of the TV distance which allows to obtain the following result.

Proposition 6.5. *Let the assumptions of Proposition 6.4 hold and further assume that*:

- *For all $\theta \in \mathbb{R}^d$ the probability measure $P_\beta(\theta, \cdot)$ admits a density $p_\beta(\theta, \omega)$ w.r.t. Lebesgue's measure.*
- *There exists $A < \infty$ such that for all $\theta, \theta' \in \mathbb{R}^d$*

$$\|P_\beta(\theta, \cdot) - P_\beta(\theta', \cdot)\|_{\mathrm{TV}} = \frac{1}{2}\int_{\mathbb{R}^d} |p_\beta(\theta, \omega) - p_\beta(\theta', \omega)| d\omega \leqslant A\|\theta - \theta'\|. \tag{6.5.2}$$

Then the ergodicity result of Theorem 6.1 holds with $\rho \leqslant \sqrt{(1 - \beta\mu)^2 + \beta^2 L_{\mathcal{W}}}$.

Proof. Using [295, Theorem 12] (see also [295, Lemma 13]), the assumptions imply that for all $\varpi, \nu \in \mathcal{M}_1(\mathbb{R}^d)$ we have:

$$\|\varpi P_\beta - \nu P_\beta\|_{\mathrm{TV}} \leqslant A\mathcal{W}_1(\varpi, \nu).$$

It then only remains to use Proposition 6.4 with $\varpi = \delta_{\theta_0} P_\beta^{n-1}$ and $\nu = \pi_\beta = \pi_\beta P_\beta$ along with the inequality:

$$\mathcal{W}_1(\varpi, \nu) \leqslant \sqrt{\mathcal{W}_2^2(\varpi, \nu)}$$

valid for all $\varpi, \nu \in \mathcal{M}_1(\mathbb{R}^d)$. □

Proposition 6.5 uses the "Wasserstein-to-TV" method [369, 295] in order to derive convergence in TV distance from Proposition 6.4 which leads to an explicit estimate of ρ in Theorem 6.1. Note however that although the necessary condition (6.5.2) is quite intuitive, its verification is not straightforward even for a toy example.

6.6 Confidence bounds

Using the convergence and concentration results formulated in the previous sections for the invariant distribution of the SGD Markov chain, we are ready to state confidence bounds on the

estimation of the optimal $\theta^\star$. Recall that by Proposition 6.1, the invariant distribution π_β may not be centered around $\theta^\star$ unless the gradient is linear, which is a particular case. In general, the expectation of π_β may not be equal to $\theta^\star$ but the bias is controlled by the step-size β. Therefore, two possibilities are available for the final estimator:

- The last iterate θ_T: with T the optimization horizon. In which case a small step size is appropriate.
- A tail average $\frac{1}{n}\sum_{j=n_0+1}^{n_0+n}\theta_j$: in which case the step size may be chosen reasonably large within the convergence conditions.

6.6.1 Final iterate concentration bounds

When the expectation of the invariant measure $\overline{\theta}_\beta$ differs from the true optimum $\theta^\star$, one may choose a small step-size β in order to obtain a precise estimator of $\theta^\star$ through the final iterate θ_T of iteration (6.2.2). In the event where Assumption 6.3 holds, a quick application of Proposition 6.2 combined with Theorem 6.1 leads to the following first deviation bounds.

Corollary 6.1. *In the setting of Proposition 6.2, for a step-size β and horizon T, we have the following high-confidence bounds*:

(*a*) *Under Assumption 6.3 (a), for $\delta > 0$, we have*

$$\mathbb{P}\Big(\big\|\theta_T - \theta^\star\big\| > \overline{K}\sqrt{8\beta\log(e/\delta)/\mu}\Big) \leqslant \delta + \rho^T M\big(1 + \|\theta_0 - \theta^\star\|^2\big).$$

(*b*) *Under Assumption 6.3 (b), for $\delta > 0$, we have*

$$\mathbb{P}\Big(\big\|\theta_T - \theta^\star\big\| > 4e\overline{K}\log(2/\delta)\sqrt{\beta/\mu}\Big) \leqslant \delta + \rho^T M\big(1 + \|\theta_0 - \theta^\star\|^2\big),$$

where ρ and M are the constants obtained from Theorem 6.1.

The proof of Corollary 6.1 is given in Section 6.9.3. As discussed earlier, the constants $\overline{K}$ drawn from Assumption 6.3 may have a poor dependence on the dimension in $\sqrt{d}$ which leaves room for improvement in the above bounds. This can be achieved when the requirements of Assumption 6.4 are met leading to the following *dimension-free* deviation bounds.

Corollary 6.2. *In the setting of Proposition 6.3, for a step-size β such that $\beta \leqslant \frac{\mu}{\mu^2+L_\sigma}$, we have the following high-confidence bounds*:

(*a*) *Under Assumption 6.4 (a), for $\delta > 0$, we have*

$$\mathbb{P}\Bigg(\big\|\theta_n - \theta^\star\big\| > \sqrt{\frac{\beta\sigma^2}{\mu}} + 2K\sqrt{\frac{\beta\log(1/\delta)}{\mu}}\Bigg) \leqslant \delta + \rho^n M\big(1 + \|\theta_0 - \theta^\star\|^2\big). \tag{6.6.1}$$

(*b*) *Under Assumption 6.4 (b), for $\delta > 0$, we have*

$$\mathbb{P}\Bigg(\big\|\theta_n - \theta^\star\big\| > \sqrt{\frac{\beta\sigma^2}{\mu}} + 2K\bigg(\sqrt{\frac{\beta\log(1/\delta)}{\mu}} \ \vee \ \beta\log(1/\delta)\bigg)\Bigg) \leqslant \delta + \rho^n M\big(1 + \|\theta_0 - \theta^\star\|^2\big), \tag{6.6.2}$$

where ρ and M are the constants obtained from Theorem 6.1.

Corollary 6.2 is proven in Section 6.9.3 and is a consequence of Propositions 6.1, 6.3 and Theorem 6.1. Inequality (6.6.1) is an example of a sub-Gaussian deviation bound since the uncertainty term has no dependence on the dimension [284].

The bounds given in Corollaries 6.1 and 6.2 can be made more explicit by granting the conditions of Proposition 6.5 which yields $\rho \approx 1 - \beta\mu$. In this case, it is possible to set $\beta = \frac{\log(T)}{\mu T}$ leading to a nearly optimal statistical rate in T albeit with a sub-optimal dependence in the problem conditioning [16, 17].

6.6.2 Polyak-Ruppert averaging

In this part, we consider the case where the step-size β is chosen as a constant order value satisfying the convergence criteria required in our previous results. Our goal is to obtain a high-confidence bound for the Polyak-Ruppert average $\frac{1}{n}\sum_{j=n_0+1}^{n_0+n}\theta_j$ computed after a burn-in period of n_0 iterations. This raises two challenges, the first of which is that, even for a very long burn-in period n_0, the stationary regime is never reached in theory so that one cannot immediately use the concentration properties of π_β. The second challenge comes from the lack of independence of the Markov chain iterates. This prevents the adoption of certain approaches such as the entropy method as done in [301] for example.

Notice that, unless the gradient is linear, there is little hope to estimate $\theta^\star$ using the Polyak-Ruppert average since it is bound to approach $\mathbb{E}_{\theta\sim\pi_\beta}[\theta] = \overline{\theta}_\beta$ which may differ from $\theta^\star$ by up to $\sigma\sqrt{\beta/\mu}$ in the non linear case. Nevertheless, the following initial statement holds without this assumption.

Theorem 6.2. *Grant Assumptions 6.1, 6.2, 6.4 (a) and 6.5. Let $f : \Theta^n \to \mathbb{R}$ be a 1-Lipschitz function in each of its parameters and $\vec{\theta} := (\theta_1, \dots, \theta_n)$ be a sequence of SGD iterates with step size $\beta < \frac{2\mu}{\mu^2 + (\mu L \vee L_{\mathcal{W}})}$ started from stationarity i.e. such that $\theta_1 \sim \pi_\beta$. Then we have*

$$f(\vec{\theta}) - \mathbb{E}f(\vec{\theta}) \in \Psi_2\big(KC_{\mathcal{W}}\sqrt{\beta/\mu + (n-1)\beta^2}\big),$$

where $C_{\mathcal{W}} = \big(1 - \sqrt{(1-\beta\mu)^2 + \beta^2 L_{\mathcal{W}}}\big)^{-1}$. If Assumption 6.4 (a) is replaced by Assumption 6.4 (b) then

$$f(\vec{\theta}) - \mathbb{E}f(\vec{\theta}) \in \Psi_1\big(KC_{\mathcal{W}}\sqrt{\beta/\mu + (n-1)\beta^2}\big).$$

The proof of Theorem 6.2 is given in Section 6.9.3 and employs a hybrid martingale transportation method (see [50, 305, 137, 95] for a reference) leveraging the $\mathcal{W}_2$ convergence established in Proposition 6.4 in combination with [242, Theorem 4.3].

Theorem 6.2 may be used in a variety of ways by plugging different choices of the function f. For instance, one may choose $f(\vec{\theta}) = \sum_i g(\theta_i)$ for any $g \in \mathrm{Lip}(\mathbb{R}^d)$. Instead, in what follows, we focus on the choice

$$f(\vec{\theta}) = \Big\| \sum_{i=n_0+1}^{n_0+n} \theta_i - n\theta^\star \Big\|.$$

Before we proceed, we formalize the gradient linearity assumption.

Assumption 6.6. *The gradient $\nabla\mathcal{L}$ is linear i.e. for all $\theta \in \Theta$ it is equal to $\nabla\mathcal{L}(\theta) = \Sigma(\theta - \theta^\star)$ for some symmetric positive definite matrix $\Sigma \in \mathbb{R}^{d\times d}$.*

Note that the positive definiteness of Σ in Assumption 6.6 is a consequence of strong convexity while its symmetry is a result of the Hessian $\nabla^2\mathcal{L}$ being constant in this case and therefore continuous.We are now ready to state our non-asymptotic deviation bound for Polyak-Ruppert averaging.

Proposition 6.6. *Grant Assumptions 6.1, 6.2, 6.4 (a), 6.5 and 6.6. Let $(\theta_t)_{t\geqslant 0}$ be the Markov sequence obtained by running SGD with step-size*

$$\beta < \frac{2\mu}{\mu^2 + (\mu L \vee L_{\mathcal{W}})} \wedge \frac{\mu}{\mu^2 + L_\sigma}$$

and initial distribution $\theta_0 \sim \nu$. Then there exist $\rho < 1$ and $M < \infty$ such that

$$\begin{aligned}\Big\|\frac{1}{n}\sum_{t=n_0+1}^{n_0+n} \theta_t - \theta^\star\Big\| \leqslant \sqrt{\frac{2}{n}\frac{1+\alpha}{1-\alpha}\Big(\alpha_{\mathcal{W}}^{n_0}\mathcal{W}_2^2(\nu,\pi_\beta) + \frac{\beta\sigma^2}{\mu}\Big)} \\ + \frac{2K\sqrt{\beta/\mu}}{1-\alpha_{\mathcal{W}}}\sqrt{\beta\mu + \frac{1}{n}}\sqrt{\frac{\log(1/\delta)}{n}}\end{aligned} \tag{6.6.3}$$

for $\delta > 0$ and $n, n_0 > 0$ with probability at least $1 - \Upsilon(\nu, n_0)\delta$, where

$$\alpha = 1 - \beta\mu, \quad \alpha_{\mathcal{W}} = \sqrt{\alpha^2 + \beta^2 L_{\mathcal{W}}} \quad \textit{and} \quad \Upsilon(\nu, n_0) = 1 + M\rho^{n_0}\Big\|\frac{d\nu}{d\pi_\beta}\Big\|_\infty.$$

If Assumption 6.4 (a) is replaced by Assumption 6.4 (b) then

$$\begin{aligned}\Big\|\frac{1}{n}\sum_{t=n_0+1}^{n_0+n} \theta_t - \theta^\star\Big\| \leqslant \sqrt{\frac{2}{n}\frac{1+\alpha}{1-\alpha}\Big(\alpha_{\mathcal{W}}^{n_0}\mathcal{W}_2^2(\nu,\pi_\beta) + \frac{\beta\sigma^2}{\mu}\Big)} \\ + \frac{2K\sqrt{\beta/\mu}}{1-\alpha_{\mathcal{W}}}\Big(\sqrt{\beta\mu + \frac{1}{n}}\sqrt{\frac{\log(1/\delta)}{n}} \vee \frac{\log(1/\delta)}{n}\Big)\end{aligned} \tag{6.6.4}$$

holds with the same probability.

The proof of Proposition 6.6 is given in Section 6.9.3 and takes advantage of the convergence both in total-variation distance and in the $\mathcal{W}_2$ metric. Note that the given bounds are also dimension-free thanks to Assumption 6.4. It is possible to derive a weaker result using only Assumption 6.3 but we omit it to avoid repetition. The variance terms in the upperbounds of (6.6.3) and (6.6.4) (those independent of δ) are controlled thanks to a geometric decorrelation phenomenon which can be shown for the Markov chain iterates under Assumption 6.6 (see Lemma 6.6 in the Appendix). This phenomenon becomes weaker for smaller step-size β, therefore, it only makes sense to apply Proposition 6.6 with β of constant order to avoid excessive correlation between the averaged samples. Finally, the lack of stationarity of the involved Markov samples is tackled by taking advantage of a spectral gap property satisfied by the transition kernel P_β under the conditions of Theorem 6.1 (see [243]).

Proposition 6.6 may be compared to the works of [323] and [278]. The former derives a similar high probability bound for linear stochastic approximation under a generalized sub-Gaussianity assumption and uncorrelated noise. The latter considers a weaker finite L_p moment assumption on the SGD data and uses mini-batching to obtain Nagaev type concentration bounds with provably optimal dependence in the confidence level. However, the results of [323, 278] both lack the dimension-free property of Proposition 6.6.

6.7 Applications

We discuss the consequences of our results for two common use-cases of SGD.

6.7.1 Linear regression

Linear regression is one of the most popular and most used standard models. The aim is to predict a real variable Y based on a random vector $X \in \mathbb{R}^d$ according to the linear model

$$Y = X^\top \theta^\star + \epsilon$$

where $\theta^\star$ is an unknown parameter and ϵ a centered noise. The estimation of $\theta^\star$ may be carried out by minimizing the least-squares objective $\mathcal{L}(\theta) := \frac{1}{2}\mathbb{E}\big(X^\top\theta - Y\big)^2$ with respect to $\theta \in \mathbb{R}^d$. This may be done by running SGD with the random gradient $G(\theta,(X,Y)) = X(X^\top\theta - Y)$.

Provided the previous gradient admits a finite second moment, Theorem 6.1 and Proposition 6.4 apply and guarantee the convergence of the SGD Markov chain in total-variation and $\mathcal{W}_2$ distance. If the covariates X and the noise ϵ are both Gaussian then the gradient $G(\theta,(X,Y))$ is sub-exponential. However, note that Assumption 6.3 or (b) are not immediately satisfied since the associated Ψ_1 constant may be unbounded for arbitrarily high values of $\|\theta - \theta^\star\|$. This problem can be remedied thanks to the following lemma.

Lemma 6.2. *Let Assumption 6.1 hold and assume that gradient errors write $\varepsilon_{\zeta_t}(\theta_t) = \Xi_t(\theta_t - \theta^\star) + \xi_t$ where the pairs $(\Xi_t,\xi_t)_{t\geqslant 0}$ are i.i.d in $\mathbb{R}^{d\times d}\times\mathbb{R}^d$ with Ξ_t symmetric and such that for all $u\in\mathbb{R}^d, \|u\| = 1$ we have $\langle u,\Xi_t u\rangle \in \Psi_1(K_\Xi)$ and $\langle u,\xi_t\rangle \in \Psi_1(K_\xi)$ for $K_\Xi, K_\xi > 0$. Assume the following minibatch SGD iteration is run starting from θ_0 such that $\|\theta_0 - \theta^\star\| \leqslant C$ for some $C > 0$ for a finite horizon T*

$$\theta_{t+1} = \theta_t - \beta\overline{G}_N(\theta_t) \quad \textit{with} \quad \overline{G}_N(\theta_t) = \frac{1}{N}\sum_{i=1}^N G(\theta_t,\zeta_{tN+i})$$

with N the minibatch size. For a confidence level $\delta > 0$, assume that N and β satisfy

$$\frac{N}{\log(4T/\delta)+3d} \geqslant 1 \vee \Big(\frac{6}{\mu}\big(3K_\Xi \vee 4K_\xi/C\big)\Big)^2 \quad \textit{and} \quad \beta \leqslant \frac{\mu N}{54K_\Xi^2(\log(4T/\delta)+3d)} \wedge \frac{2}{\mu+L}.$$

Then, with probability at least $1-\delta$, we have $\max_{0\leqslant s\leqslant T}\|\theta_s - \theta^\star\| \leqslant C$.

Lemma 6.2 is proven in Section 6.9.3 and guarantees that, using a small step size and minibatching to reduce the gradient variance, with high probability, the iteration does not stray from the vicinity of the optimum during a finite horizon. This shows that the uniform aspect of Assumptions 6.3 and 6.4 does not prevent the application of the results given in the previous sections. Note that although Lemma 6.2 requires that $N = \Omega(d)$, the constant C is arbitrary and may be taken dimension-free, for instance, by starting the iteration from a preliminary estimator $\theta_0 = \widehat{\theta}$.

For the example of linear regression with sub-Gaussian samples $(X_t,Y_t)_t$, Lemma 6.2 applies with $\Xi_t = X_tX_t^\top - \mathbb{E}[X_tX_t^\top]$ and $\xi_t = -\epsilon_t X_t$. Thus, for finite horizon, one may consider the event where the bound of Lemma 6.2 holds to apply results from Sections 6.4 and 6.6.

Alternatively, it is also possible to restrict the optimization to a convex and bounded subset $\Theta \subset \mathbb{R}^d$ such that $\theta^\star \in \Theta$. By letting $\Pi_\Theta(\cdot)$ be the projection onto Θ and replacing iteration (6.2.2) with

$$\theta_{t+1} = \Pi_\Theta\big(\theta_t - \beta G(\theta_t,\zeta_t)\big), \tag{6.7.1}$$

we obtain a Markov chain to which Proposition 6.2 (b) applies and leads to the deviation bound (6.6.2). Indeed, it is easy to verify that these results still hold for iteration (6.7.1) thanks to the inequality

$$\big\|\Pi_\Theta\big(\theta - \beta G(\theta,\zeta)\big) - \theta^\star\big\| \leqslant \big\|\theta - \beta G(\theta,\zeta) - \theta^\star\big\|,$$

valid for all $\theta \in \mathbb{R}^d$ since $\theta^\star \in \Theta$ which is convex. However, by considering the projected iteration (6.7.1), Proposition 6.1 (a) may no longer hold so that $\overline{\theta}_\beta \neq \theta^\star$ making Proposition 6.6 no longer applicable.

6.7.2 Logistic regression

Logistic regression corresponds to the model

$$1 - \mathbb{P}(Y = -1|X) = \mathbb{P}(Y = +1|X) = \sigma(X^\top \theta^\star),$$

where σ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. For a parameter θ and a sample X, the predicted probability is $\mathbb{P}(Y = +1|X) = \sigma(X^\top \theta)$ and the model is trained using the log-loss $\ell(\theta, (X, Y)) = -\log(\sigma(YX^\top \theta))$ which yields the objective $\mathcal{L}(\theta) = \mathbb{E}_{(X,Y)}\ell(\theta, (X, Y))$.

In order to ensure the objective is strongly-convex, it is necessary to restrict the parameter θ to a bounded convex set Θ. This is commonly done by setting $\Theta = \{\theta \in \mathbb{R}^d\,,\ \|\theta\| \leqslant R\}$ for some radius $R > 0$ [185, 16, 325].

In this case, the projected iteration (6.7.1) may be used. In this setting, one may easily check that the gradient is sub-Gaussian/sub-exponential as soon as the covariates X satisfy one or the other of these properties. Therefore, the results of Propositions 6.2 and 6.3 apply in this context as well.

6.8 Conclusion and Discussion

The Markov chain point of view for SGD is very useful since it allows to draw conclusions and establish a number of characterizations for the invariant limit distribution. Convergence of the SGD Markov chain holds under fairly weak conditions [313]. As evidenced by our results, this opens doors for a better characterization of the limit distribution when the associated optimization iteration progresses at *geometric* speed, for instance, when strong convexity holds. The precise determination of the speed of convergence in distribution constitutes a particular difficulty which more generally concerns Markov chains with a geometric drift property. However, this difficulty may be circumvented for SGD by leveraging Wasserstein convergence provided a regularity condition on the noise distribution and transition kernel. Obtaining such properties from generic assumptions on the gradient distribution represents an interesting perspective. Finally, despite being quite productive, the Markov chain study of SGD remains limited to the constant step-size setting. This excludes the combination of a decreasing step-size with averaging which is known for its better dependence on problem conditioning [16, 17].

6.9 Proofs

6.9.1 Preliminary lemmas

Lemma 6.3. *Let X be a real random variable such that $X \in \widetilde{\Psi}_2(K)$ then, with probability at least $1-\delta$, we have*

$$|X| \leqslant K\sqrt{\log(e/\delta)}.$$

Proof. Using Chernoff's method, we find for $t>0$ and $\lambda>0$

$$\begin{aligned}\mathbb{P}\big(|X|>t\big) &= \mathbb{P}\big(\lambda^2X^2>\lambda^2t^2\big) = \mathbb{P}\big(\exp(\lambda^2X^2)>\exp(\lambda^2t^2)\big)\\ &\leqslant \mathbb{E}\exp(\lambda^2X^2)e^{-\lambda^2t^2} \leqslant \exp\big(\lambda^2(K^2-t^2)\big).\end{aligned}$$

Choosing $\lambda=1/K$, we have $\exp\big(1-(t/K)^2\big)\leqslant\delta \iff t\geqslant K\sqrt{\log(e/\delta)}$ and the result follows. □

Lemma 6.4. *Let X be a real random variable such that $X \in \widetilde{\Psi}_1(K)$ then, with probability at least $1-\delta$, we have*

$$|X| \leqslant 2eK\log(2/\delta).$$

Proof. Using Stirling's approximation, we find for $|\lambda|<(eK)^{-1}$:

$$\begin{aligned}\mathbb{E}\exp(\lambda|X|) &= \sum_{p\geqslant 0}\frac{\lambda^p\mathbb{E}|X|^p}{p!} \leqslant 1+\sum_{p\geqslant 1}\frac{(\lambda Kp)^p}{p!}\\ &\leqslant 1+\sum_{p\geqslant 1}\frac{(\lambda eK)^p}{\sqrt{2\pi p}} \leqslant 1+\frac{1}{\sqrt{2\pi}}\frac{\lambda eK}{1-\lambda eK} \leqslant \exp\Big(\frac{1}{\sqrt{2\pi}}\frac{\lambda eK}{1-\lambda eK}\Big),\end{aligned}$$

where we used the inequality $1+x\leqslant e^x$ in the last step. For $t>0$, using Chernoff's method and choosing $\lambda=(2eK)^{-1}$, we find:

$$\begin{aligned}\mathbb{P}\big(|X|>t\big) &= \mathbb{P}\big(\lambda|X|>\lambda t\big) = \mathbb{P}\big(\exp(\lambda|X|)>\exp(\lambda t)\big)\\ &\leqslant \mathbb{E}\exp(\lambda|X|)e^{-\lambda t} \leqslant \exp\Big(\frac{1}{\sqrt{2\pi}}-\frac{t}{2eK}\Big).\end{aligned}$$

It only remains to choose $t=2eK\log(2/\delta)$ to obtain the desired bound. □

The following fundamental lemma will be often used in our proofs.

Lemma 6.5. *Grant Assumption 6.1. For any $\theta,\theta'\in\mathbb{R}^d$ and $\beta\leqslant\frac{2}{\mu+L}$ we have*

$$\|\theta-\beta\nabla\mathcal{L}(\theta)-(\theta'-\beta\nabla\mathcal{L}(\theta'))\|^2 \leqslant (1-\beta\mu)^2\|\theta-\theta'\|^2. \tag{6.9.1}$$

Proof. For $\beta\leqslant\frac{2}{\mu+L}$, we have

$$\begin{aligned}&\|\theta-\beta\nabla\mathcal{L}(\theta)-(\theta'-\beta\nabla\mathcal{L}(\theta'))\|^2\\ &\qquad= \|\theta-\theta'\|^2-2\beta\langle\theta-\theta',\nabla\mathcal{L}(\theta)-\nabla\mathcal{L}(\theta')\rangle+\beta^2\|\nabla\mathcal{L}(\theta)-\nabla\mathcal{L}(\theta')\|^2\\ &\qquad\leqslant (1-\beta^2\mu L)\|\theta-\theta'\|^2-\beta(2-\beta(\mu+L))\langle\nabla\mathcal{L}(\theta)-\nabla\mathcal{L}(\theta'),\theta-\theta'\rangle)\\ &\qquad\leqslant (1-\beta^2\mu L)\|\theta-\theta'\|^2-\beta(2-\beta(\mu+L))\mu\|\theta-\theta'\|^2\\ &\qquad= (1-\beta^2\mu L-2\beta\mu+\beta^2\mu(\mu+L))\|\theta-\theta'\|^2\\ &\qquad= (1-\beta\mu)^2\|\theta-\theta'\|^2,\end{aligned}$$

where we used the inequalities

$$\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\|^2 \leqslant (\mu + L)\langle\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta - \theta'\rangle - \mu L\|\theta - \theta'\|^2 \tag{6.9.2}$$

$$\mu\|\theta - \theta'\|^2 \leqslant \langle\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta'), \theta - \theta'\rangle, \tag{6.9.3}$$

valid for all θ, θ'. Equation (6.9.2) is stated, for example, in [342, Theoerem 2.1.12] (see also [58, Lemma 3.11] and (6.9.3) is just a characterization of strong convexity (see for instance [342, Theorem 2.1.9]). □

6.9.2 Proof of geometric ergodicity and invariant properties

In the remaining part of this document, we make the dependences on ζ in the gradient samples and errors implicit and write $G(\theta)$ and $\varepsilon(\theta)$ instead of $G(\theta, \zeta)$ and $\varepsilon_\zeta(\theta)$ respectively.

We show the geometric ergodicity of the SGD Markov chain $(\theta_t)_{t\geqslant 0}$ by relying on [313, Theorem 15.0.1]. We will show that the following function:

$$V(\theta) := 1 + \|\theta - \theta^\star\|^2,$$

is a *drift* function for this Markov chain. We define the action of the transition kernel P on integrable functions f through

$$Pf(\theta) = \mathbb{E}f(\theta - \beta G(\theta)).$$

We also define the variation operator

$$\Delta f(\theta) := Pf(\theta) - f(\theta).$$

Proof of Theorem 6.1

First, we establish that the Markov chain is aperiodic. Indeed, by Assumption 6.2, for all θ, the gradient is distributed according to an everywhere positive density, therefore, for all $\theta \in S \subset \mathbb{R}^d$ with S a set with non zero Lebesgue measure we have $P(\theta, S) > 0$. This implies that the greatest possible period for the chain is 1 which makes it aperiodic.

We also show that the Markov chain is ψ-irreducible (see [313, Chapter 4]). For any initial θ_0, its successor reads:

$$\theta_{+1} = \theta_0 - \beta(\nabla\mathcal{L}(\theta_0) + \varepsilon(\theta_0))$$

Given Assumption 6.2, the distribution of $\varepsilon(\theta_0)$ is minorized by $\delta\nu_{\theta_0,1}$ where $\nu_{\theta_0,1}$ is a probability distribution which admits an everywhere positive density $h(\theta_0, \cdot)$. Consequently, for all $A \in \mathcal{B}(\mathbb{R}^d)$ with non zero Lebesgue measure, we have the following minorization:

$$\mathbb{P}(\theta_{+1} \in A|\theta_0) = P(\theta_0, A) \geqslant \frac{\delta}{\beta^d}\int_A h\Big(\theta_0, \frac{\theta - \theta_0}{\beta} - \nabla\mathcal{L}(\theta_0)\Big)d\theta > 0.$$

It follows that the Markov chain is irreducible w.r.t. Lebesgue's measure and is thus ψ-irreducible.

For fixed θ, and step-size $\beta < \frac{2}{\mu+L}$, using Lemma 6.5 we find:

$$\begin{aligned}
P\|\theta - \theta^\star\|^2 &= \mathbb{E}\|\theta - \beta G(\theta) - \theta^\star\|^2 \\
&= \mathbb{E}\big[\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle + \beta^2\|\varepsilon(\theta)\|^2\big] \\
&\leqslant (1 - \beta\mu)^2\|\theta - \theta^\star\|^2 + \beta^2\mathbb{E}\|\varepsilon(\theta)\|^2 \\
&\leqslant (1 - \beta\mu)^2\|\theta - \theta^\star\|^2 + \beta^2\big(L_\sigma\|\theta - \theta^\star\|^2 + \sigma^2\big)
\end{aligned}$$

$$= \big((1-\beta\mu)^2 + \beta^2 L_\sigma\big)\|\theta - \theta^\star\|^2 + \beta^2\sigma^2$$

The previous inequality yields a contraction for step-size satisfying $0 < \beta < \frac{2\mu}{\mu^2 + L_\sigma}$ and, as a consequence we have:

$$PV(\theta) \leqslant \underbrace{\big((1-\beta\mu)^2 + \beta^2 L_\sigma\big)}_{=:\widetilde{\lambda}} V(\theta) + \underbrace{\beta^2\sigma^2 + \big(1 - \big((1-\beta\mu)^2 + \beta^2 L_\sigma\big)\big)}_{=:\widetilde{b}}$$

We now define the set $\mathcal{C} = \big\{\theta \in \mathbb{R}^d\,,\, V(\theta) \leqslant 2\widetilde{b}/(1-\widetilde{\lambda})\big\}$ which satisfies:

$$\Delta V(\theta) \leqslant -\frac{1-\widetilde{\lambda}}{2} V(\theta) + \widetilde{b}\mathbf{1}_{\theta\in\mathcal{C}}.$$

For such $\mathcal{C}$, let $\underline{h}(\theta) = \inf_{\theta_0\in\mathcal{C}} h\Big(\theta_0, \frac{\theta-\theta_0}{\beta} - \nabla\mathcal{L}(\theta_0)\Big)$ and define the probability measure $\nu_{\mathcal{C}}$ by

$$\nu_{\mathcal{C}}(A) = \frac{\int_{A\cap\mathcal{C}} \underline{h}(\theta)d\theta}{\int_{\mathcal{C}} \underline{h}(\theta')d\theta'} \quad \text{for all} \quad A \in \mathcal{B}(\mathbb{R}^d).$$

It follows that for all $\theta_0 \in \mathcal{C}$, we have the following minorization property:

$$P(\theta_0, A) \geqslant \xi\nu_{\mathcal{C}}(A) \quad \text{for all} \quad A \in \mathcal{B}(\mathbb{R}^d),$$

where $\xi = \delta\int_{\mathcal{C}} \underline{h}(\theta)d\theta > 0$. In words, the set $\mathcal{C}$ is a *small* set and, thanks to [313, Proposition 5.5.3], also a *petite* set (see definitions in [313, Chapter 5]).

We have shown that the Markov chain (θ_t) satisfies condition (iii) of [313, Theorem 15.0.1]. By the latter result, it follows that it admits an invariant probability measure π_β and there exist $r > 1$ and $M < \infty$ such that:

$$\sum_{t\geqslant 0} r^t \|P(\theta_0, \cdot) - \pi_\beta\|_V \leqslant MV(\theta_0). \tag{6.9.4}$$

Taking $\rho = r^{-1}$ allows to conclude the proof.

Proof of Proposition 6.1

To prove (a), let $\theta \sim \pi_\beta$ and simply compute

$$\mathbb{E}[\theta] = \mathbb{E}\big[\theta - \beta G(\theta)\big] = \mathbb{E}\big[\theta - \beta\nabla\mathcal{L}(\theta)\big] = \mathbb{E}[\theta] - \beta\mathbb{E}[\nabla\mathcal{L}(\theta)]$$

since we know that $\mathbb{E}[\theta] < \infty$ (this follows from (6.9.4) in the proof of Theorem 6.1), this implies the first part of the claim. If we further assume the gradient to be linear, we have in addition

$$\mathbb{E}\nabla\mathcal{L}(\theta) = \nabla\mathcal{L}\big(\mathbb{E}\theta\big) = \nabla\mathcal{L}(\bar{\theta}) = 0,$$

and the conclusion follows since $\theta^\star$ is the unique critical point.

To prove (b), let $\theta \sim \pi_\beta$ which implies that $\theta - \beta G(\theta) \sim \pi_\beta$ as well. We compute:

$$\begin{aligned}
\mathbb{E}\|\theta - \theta^\star\|^2 &= \mathbb{E}\|\theta - \beta G(\theta) - \theta^\star\|^2 \\
&= \mathbb{E}\big[\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 + \beta^2\|\varepsilon(\theta)\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle\big] \\
&= \mathbb{E}\big[\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 + \beta^2\mathbb{E}\|\varepsilon(\theta)\|^2\big] \\
&\leqslant \big((1-\beta\mu)^2 + \beta^2 L_\sigma\big)\mathbb{E}\|\theta - \theta^\star\|^2 + \beta^2\sigma^2
\end{aligned}$$

which implies

$$\mathbb{E}\|\theta - \theta^\star\|^2 \leqslant \frac{\beta^2\sigma^2}{1-(1-\beta\mu)^2-\beta^2 L_\sigma} = \frac{\beta\sigma^2}{2\mu - \beta(\mu^2+L_\sigma)}.$$

Moreover, by Jensen and Cauchy-Schwarz inequalities, we have

$$\|\bar{\theta}_\beta - \theta^\star\| = \|\mathbb{E}\theta - \theta^\star\| \leqslant \mathbb{E}\|\theta-\theta^\star\| \leqslant \sqrt{\mathbb{E}\|\theta-\theta^\star\|^2}.$$

Proof of Proposition 6.2

Let us prove (a), we again use the fact that θ and $\theta - \beta G(\theta)$ have the same distribution when $\theta \sim \pi_\beta$:

$$\begin{aligned}\mathbb{E}\exp(\lambda^2\|\theta-\theta^\star\|^2) &= \mathbb{E}\exp(\lambda^2\|\theta-\beta G(\theta)-\theta^\star\|^2)\\ &= \mathbb{E}\exp\big(\lambda^2(\|\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star\|^2 - 2\beta\langle\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star,\varepsilon(\theta)\rangle + \beta^2\|\varepsilon(\theta)\|^2)\big)\end{aligned}$$

Since we assume that $\|\varepsilon(\theta)\| \in \widetilde{\Psi}_2(\overline{K})$, it is easy to check that for all $u \in \mathbb{R}^d$ with unit norm, $\langle u, \varepsilon(\theta)\rangle \in \Psi_2(\overline{K})$ because $\varepsilon(\theta)$ is centered. Therefore, conditioning on θ, we have

$$\begin{aligned}&\mathbb{E}\big[\exp(\lambda^2(-2\beta\langle\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star,\varepsilon(\theta)\rangle + \beta^2\|\varepsilon(\theta)\|^2))|\theta\big]\\ &\quad\leqslant \mathbb{E}\big[\exp(-(2\lambda)^2\beta\langle\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star,\varepsilon(\theta)\rangle|\theta\big]^{1/2}\mathbb{E}\big[\exp(2(\beta\lambda)^2\|\varepsilon(\theta)\|^2)|\theta\big]^{1/2}\\ &\quad\leqslant \exp(8\lambda^4\beta^2\|\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star\|^2\overline{K}^2 + \lambda^2\beta^2\overline{K}^2)\\ &\quad\leqslant \exp(8\lambda^4\beta^2(1-\beta\mu)^2\|\theta-\theta^\star\|^2\overline{K}^2 + \lambda^2\beta^2\overline{K}^2),\end{aligned}$$

where the last line uses that $\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\| \leqslant (1-\beta\mu)\|\theta-\theta^\star\|$. The previous inequality holds for $|\lambda| \leqslant \big(\sqrt{2}\beta\overline{K}\big)^{-1}$. We now restrict λ so that $|\lambda| \leqslant \big(\overline{K}\sqrt{8\beta/\mu}\big)^{-1}$ and use Jensen's inequality to obtain

$$\begin{aligned}\mathbb{E}\exp\big(\lambda^2\|\theta-\theta^\star\|^2\big) &\leqslant \mathbb{E}\exp\big(\lambda^2(1-\beta\mu)^2\|\theta-\theta^\star\|^2(1+8\lambda^2\beta^2\overline{K}^2)\big)\exp\big(\lambda^2\beta^2\overline{K}^2\big)\\ &= \mathbb{E}\exp\big(\lambda^2(1-\beta\mu)^2\|\theta-\theta^\star\|^2(1+\beta\mu)\big)\exp\big(\lambda^2\beta^2\overline{K}^2\big)\\ &= \mathbb{E}\exp\big(\lambda^2(1-\beta\mu)(1-(\beta\mu)^2)\|\theta-\theta^\star\|^2\big)\exp\big(\lambda^2\beta^2\overline{K}^2\big)\\ &\leqslant \Big[\mathbb{E}\exp\big(\lambda^2\|\theta-\theta^\star\|^2\big)\Big]^{(1-\beta\mu)(1-(\beta\mu)^2)}\exp\big(\lambda^2\beta^2\overline{K}^2\big)\end{aligned}$$

and we arrive to the conclusion that

$$\begin{aligned}\mathbb{E}\exp(\lambda^2\|\theta-\theta^\star\|^2) &\leqslant \exp\Big(\frac{\lambda^2\beta^2\overline{K}^2}{\beta\mu+(\beta\mu)^2-(\beta\mu)^3}\Big) = \exp\Big(\frac{\lambda^2\beta\overline{K}^2}{\mu+\beta\mu^2-\beta^2\mu^3}\Big)\\ &\leqslant \exp\Big(\frac{\lambda^2\beta\overline{K}^2}{\mu}\Big)\end{aligned}$$

where in the end we use that $\beta\mu^2 \geqslant \beta^2\mu^3$ since $\beta \leqslant 1/\mu$.

The task of proving (b) is more delicate. By assumption, we know that for all θ, the gradient error $\varepsilon(\theta)$ satisfies

$$\big\|\|\varepsilon(\theta)\|\big\|_{L_p} \leqslant \overline{K}p \quad \text{for} \quad p \geqslant 1. \tag{6.9.5}$$

We denote $M_p^p = \mathbb{E}\|\theta-\theta^\star\|^p$. For $p=2$, we have through a similar computation to that for

proving (b)

$$M_2^2 \leqslant \frac{\beta(2\overline{K})^2}{\mu(2-\beta\mu)} \leqslant \frac{\beta(2\overline{K})^2}{\mu},$$

which immediately entails $M_1 \leqslant 2\overline{K}\sqrt{\beta/\mu}$. We will show by induction that

$$M_p \leqslant K_\pi p \quad \text{for all} \quad p \geqslant 1, \tag{6.9.6}$$

with $K_\pi = C\overline{K}\sqrt{\beta/\mu}$ for some $C \geqslant 2$. For $p \geqslant 2$, we assume (6.9.6) holds up to $2p-2$ and consider M_{2p}, we compute

$$\begin{aligned}
M_{2p}^{2p} &= \mathbb{E}\|\theta - \theta^\star\|^{2p} = \mathbb{E}\|\theta - \beta G(\theta) - \theta^\star\|^{2p} \\
&= \mathbb{E}\big(\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle + \beta^2\|\varepsilon(\theta)\|^2\big)^p \\
&\leqslant \mathbb{E}\sum_{k=0}^{p}\binom{p}{k}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2k}\big(\beta^2\|\varepsilon(\theta)\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle\big)^{p-k} \\
&\leqslant (1-\beta\mu)^{2p}M_{2p}^{2p} + p\mathbb{E}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2p-2}(\beta\|\varepsilon(\theta)\|)^2 \\
&\quad + \mathbb{E}\sum_{k=0}^{p-2}\binom{p}{k}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2k}\big(\beta^2\|\varepsilon(\theta)\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle\big)^{p-k},
\end{aligned}$$

where we isolated the two last terms of the sum in the last step and used that $\varepsilon(\theta)$ is centered conditionally on θ. Further, we have

$$\begin{aligned}
&\mathbb{E}\sum_{k=0}^{p-2}\binom{p}{k}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2k}\big(\beta^2\|\varepsilon(\theta)\|^2 - 2\beta\langle\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star, \varepsilon(\theta)\rangle\big)^{p-k} \\
&\leqslant \mathbb{E}\sum_{k=0}^{p-2}\binom{p}{k}\|\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star\|^{2k}\sum_{j=0}^{p-k}\binom{p-k}{j}(\beta\|\varepsilon(\theta)\|)^{2j}(2\beta\|\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star\|\|\varepsilon(\theta)\|)^{p-k-j} \\
&= \sum_{k=0}^{p-2}\sum_{j=0}^{p-k}\binom{p}{k}\binom{p-k}{j}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{p+k-j}(\beta\|\varepsilon(\theta)\|)^{p-k+j}2^{p-k-j}.
\end{aligned}$$

Now consider an index $l = p-k+j$, note that $2p-l = p+k-j$ and we have $2 \leqslant l \leqslant 2p$. We compute the sum $\sum_{k=0}^{p-2}\sum_{j=0}^{p-k}\binom{p}{k}\binom{p-k}{j}2^{p-k-j}$ for a fixed value of l :

$$\begin{aligned}
\sum_{\substack{k=0,j=0\\p-k+j=l}}^{k=p-2,j=p-k}\binom{p}{k}\binom{p-k}{j}2^{p-k-j} &= \sum_{k=0}^{p-2}\binom{p}{k}\binom{p-k}{l-(p-k)}2^{2(p-k)-l}\mathbf{1}_{p-k\leqslant l\leqslant 2(p-k)} \qquad (6.9.7)\\
&= \sum_{k=0\vee(p-l)}^{(p-2)\wedge(p-\lceil l/2\rceil)}\binom{p}{k}\binom{p-k}{l-(p-k)}2^{2(p-k)-l} = \sum_{k=0}^{(\lfloor l/2\rfloor)\wedge(l-2)}\binom{p}{k,k+p-l,l-2k}2^{l-2k}
\end{aligned}$$

where $\binom{p}{k,k+p-l,l-2k} = \frac{p!}{k!(k+p-l)!(l-2k)!}$ is the *trinomial* coefficient. Similarly we find that:

$$\sum_{\substack{k=0,j=0\\p-k+j=l}}^{k=p,j=p-k}\binom{p}{k}\binom{p-k}{j}2^{p-k-j} = \sum_{k=0}^{\lfloor l/2\rfloor}\binom{p}{k,k+p-l,l-2k}2^{l-2k}.$$

In what follows, we set the convention that $\binom{p}{k,k+p-l,l-2k} = 0$ whenever $k \wedge (k+p-l) \wedge (l-2k) < 0$ which allows us to sum over all integer values without specifying the limits. For some variable x, we multiply by x^l, sum over l and perform the change of variable $l \to l+2k$ to find

$$\begin{aligned}\sum_l \sum_{k=0}^{\lfloor l/2 \rfloor} \binom{p}{k, k+p-l, l-2k} 2^{l-2k} x^l &= \sum_{l,k} \binom{p}{k, k+p-l, l-2k} 2^{l-2k} x^l \\ &= \sum_{l,k} \binom{p}{k, l, p-l-k} 2^l x^{l+2k} = \sum_{l,k} \binom{p}{k, l, p-l-k} (2x)^l (x^2)^k \\ &= (x^2+2x+1)^p = (x+1)^{2p} = \sum_{l=0}^{2p} \binom{2p}{l} x^l.\end{aligned}$$

By identification of the terms in the sum over l, this yields the equality

$$\sum_{k=0}^{\lfloor l/2 \rfloor} \binom{p}{k, k+p-l, l-2k} 2^{l-2k} = \binom{2p}{l}$$

Plugging back into (6.9.7) and paying attention to the missing terms in the original sum, we find, for $2 \leqslant l \leqslant 2p$:

$$\sum_{\substack{k=0,j=0 \\ p-k+j=l}}^{k=p-2,j=p-k} \binom{p}{k} \binom{p-k}{j} 2^{p-k-j} = \binom{2p}{l} - p\mathbf{1}_{l=2}.$$

Plugging back in the original sum, we find

$$\begin{aligned}(1-(1-\beta\mu)^{2p}) M_{2p}^{2p} &\leqslant p\mathbb{E}\|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2p-2} (\beta\|\varepsilon(\theta)\|)^2 \\ &\quad + \mathbb{E}\sum_{l=2}^{2p} \Big(\binom{2p}{l} - p\mathbf{1}_{l=2} \Big) \|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2p-l} (\beta\|\varepsilon(\theta)\|)^l \\ &= \mathbb{E}\sum_{l=2}^{2p} \binom{2p}{l} \|\theta - \beta\nabla\mathcal{L}(\theta) - \theta^\star\|^{2p-l} (\beta\|\varepsilon(\theta)\|)^l \qquad (6.9.8) \\ &\overset{①}{\leqslant} \sum_{l=2}^{2p} \binom{2p}{l} ((1-\beta\mu) M_{2p-l})^{2p-l} (\beta l \overline{K})^l \\ &\overset{②}{\leqslant} (\beta 2p\overline{K})^{2p} + \sum_{l=2}^{2p-1} \binom{2p}{l} ((1-\beta\mu)(2p-l)K_\pi)^{2p-l} (\beta l \overline{K})^l \\ &\overset{③}{\leqslant} ((2p-1)K_\pi)^{2p} \Big(\frac{2p}{2p-1} \Big)^{2p} \Big[(\beta\overline{K}/K_\pi)^{2p} + \frac{e^{\frac{1}{24p}}}{\sqrt{2\pi}} \sum_{l=2}^{2p-1} \sqrt{\frac{2p}{l(2p-l)}} (1-\beta\mu)^{2p-l} (\beta\overline{K}/K_\pi)^l \Big] \\ &\overset{④}{\leqslant} ((2p-1)K_\pi)^{2p} \Big(\frac{2p}{2p-1} \Big)^{2p} \Big[(\beta\overline{K}/K_\pi)^{2p} + \frac{e^{\frac{1}{24p}}}{\sqrt{4\pi}} (2p-2) \sqrt{\frac{2p}{2p-2}} (1-\beta\mu)^{2p-2} (\beta\overline{K}/K_\pi)^2 \Big],\end{aligned}$$

where ① uses (6.9.5) and Lemma 6.5, ② uses our induction hypothesis, ③ uses Stirling's approximation and ④ uses that $1 - \beta\mu > \beta\overline{K}/K_\pi$. We now use the following inequalities for

$p \geqslant 2$:

$$1-(1-\beta\mu)^{2p} = \beta\mu \sum_{i=0}^{2p-1} (1-\beta\mu)^i \geqslant 2p\beta\mu(1-\beta\mu)^{2p-1} \qquad \beta \leqslant 1/(2\mu)$$
$$\Big(\frac{2p}{2p-1}\Big)^{2p-1} = \Big(1+\frac{1}{2p-1}\Big)^{2p-1} \leqslant e \quad \text{and} \quad \frac{\sqrt{2p(2p-2)}}{2p-1} \leqslant 1,$$

in addition to the choice $K_\pi = C\overline{K}\sqrt{\beta\mu}$ with $C = 2$ to find

$$M_{2p}^{2p} \leqslant ((2p-1)K_\pi)^{2p}\Big(\frac{e}{1-\beta\mu}\Big)\Big[\Big(\frac{\sqrt{\beta\mu}}{1-\beta\mu}\Big)^{2p-2}\frac{C^{-2p}}{2p-1} + \frac{e^{\frac{1}{48}}C^{-2}}{\sqrt{4\pi}}\Big]$$
$$\leqslant ((2p-1)K_\pi)^{2p}(2e)\Big[\frac{1}{6}\Big(\frac{1}{\sqrt{2}}\Big)^{2p} + \frac{e^{\frac{1}{48}}}{8\sqrt{\pi}}\Big] \leqslant ((2p-1)K_\pi)^{2p}$$

which yields the desired bound (6.9.6) for M_{2p} as well as M_{2p-1} through $M_{2p-1} \leqslant M_{2p}$. This finishes the induction.

Proof of Lemma 6.1

Without loss of generality, we consider moments of even order. For $j \geqslant 1$, denoting $M_{2j}^{2j} = \mathbb{E}\|\theta-\theta^\star\|^{2j}$ and starting from Equation (6.9.8) which was obtained in the proof of Proposition 6.2 and using Lemma 6.5 and our assumption on $\|\varepsilon(\theta)\|$ yields

$$\big(1-(1-\beta\mu)^{2j}\big)M_{2j}^{2j} \leqslant \mathbb{E}\sum_{l=2}^{2j}\binom{2j}{l}\|\theta-\beta\nabla\mathcal{L}(\theta)-\theta^\star\|^{2j-l}(\beta\|\varepsilon(\theta)\|)^l$$
$$\leqslant \mathbb{E}\sum_{l=2}^{2j}\binom{2j}{l}\big((1-\beta\mu)\|\theta-\theta^\star\|\big)^{2j-l}\beta^l\big(K\|\theta-\theta^\star\|+\underline{K}\big)^l$$
$$\leqslant \mathbb{E}\sum_{l=2}^{2j}\binom{2j}{l}\big((1-\beta\mu)\|\theta-\theta^\star\|\big)^{2j-l}\beta^l\sum_{k=0}^{l}\binom{l}{k}\big(K\|\theta-\theta^\star\|\big)^{l-k}\underline{K}^k$$
$$\leqslant \sum_{l=2}^{2j}\binom{2j}{l}(1-\beta\mu)^{2j-l}\beta^l\Big(K^l M_{2j}^{2j} + \sum_{k=1}^{l}\binom{l}{k}K^{l-k}\underline{K}^k M_{2j-k}^{2p-k}\Big).$$

By sorting out the factors of M_{2j}^{2j} and rearranging the terms, we find

$$\Big(1-(1-\beta\mu)^{2j} - \sum_{l=2}^{2j}\binom{2j}{l}(1-\beta\mu)^{2j-l}(\beta K)^l\Big)M_{2j}^{2j} \leqslant$$
$$\sum_{l=2}^{2j}\binom{2j}{l}(1-\beta\mu)^{2j-l}\beta^l\sum_{k=1}^{l}\binom{l}{k}K^{l-k}\underline{K}^k M_{2j-k}^{2j-k}.$$

Assuming that $M_i < \infty$ for $i < 2j$, the above inequality would allow us to show that $M_{2j} < \infty$ as well provided that the factor of M_{2j}^{2j} on the LHS is positive. We now use the inequalities

$0 \leqslant (1-x)^k - (1-kx) \leqslant k(k-1)x^2/2$ valid for $x \geqslant 0$ and $k \in \mathbb{N}^*$ to find

$$\begin{aligned}1-(1-\beta\mu)^{2j} - \sum_{l=2}^{2j}\binom{2j}{l}(1-\beta\mu)^{2j-l}(\beta K)^l &= 1-(1-\beta(\mu-K))^{2j} + 2j\beta K(1-\beta\mu)^{2j-1}\\ &\geqslant 2j\beta\mu - 2j(2j-1)\beta^2((\mu-K)^2/2+\mu K)\\ &= 2j\beta(\mu-\beta(2j-1)(\mu^2+K^2)/2) \geqslant 0,\end{aligned}$$

where the last inequality follows from the bound we imposed on β. Therefore, we can deduce that $M_{2j} < \infty$. Since a similar argument works for $M_i < \infty$ with $i < 2p$ with a weaker condition on β, the result follows.

Proof of Proposition 6.3

We now prove (a). Let $\theta, \theta' \sim \pi_\beta$ be independent and define the *gradient step* function g_β as

$$g_\beta(\vartheta) = \vartheta - \beta\nabla\mathcal{L}(\vartheta) \quad \text{for} \quad \vartheta \in \mathbb{R}^d.$$

Note that, by Lemma 6.5, g_β is $(1-\beta\mu)$-Lipschitz. Under Assumption 6.4 (a) and using the invariance of π_β, we have for all $\lambda \in \mathbb{R}$:

$$\begin{aligned}&\sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\big(\lambda(f(\theta)-\mathbb{E}f(\theta'))\big)\\ &\quad= \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\big(\lambda(f(\theta-\beta G(\theta))-\mathbb{E}f(\theta'-\beta G(\theta')))\big)\\ &\quad= \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\Big(\lambda\big(f((\theta-\beta\nabla\mathcal{L}(\theta))-\beta\varepsilon(\theta)) - \mathbb{E}f((\theta'-\beta\nabla\mathcal{L}(\theta'))-\beta\varepsilon(\theta'))\big)\Big)\\ &\quad= \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\Big(\lambda\big(f(g_\beta(\theta)) - \mathbb{E}f(g_\beta(\theta'))\big)\\ &\qquad+\lambda\big(f(g_\beta(\theta)-\beta\varepsilon(\theta)) - f(g_\beta(\theta)) - \mathbb{E}[f(g_\beta(\theta')-\beta\varepsilon(\theta')) - f(g_\beta(\theta'))]\big)\Big).\end{aligned}$$

Conditioning on θ, it is clear that

$$\phi\big(G(\theta)\big) := f\big(g_\beta(\theta)-\beta(G(\theta)-\nabla\mathcal{L}(\theta))\big) - f\big(g_\beta(\theta)\big) = f\big(g_\beta(\theta)-\beta\varepsilon(\theta)\big) - f\big(g_\beta(\theta)\big)$$

is a β-Lipschitz function of $G(\theta)$. In addition, $f(g_\beta(\theta))$ is a $(1-\beta\mu)$-Lipschitz function of θ, therefore by reparametrizing the space of Lipschitz functions and using Jensen's inequality, we find

$$\begin{aligned}&\sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\big(\lambda(f(\theta)-\mathbb{E}f(\theta'))\big)\\ &\qquad\leqslant \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\Big(\lambda\big(f(g_\beta(\theta)) - \mathbb{E}f(g_\beta(\theta'))\big)\Big)\exp(\lambda^2\beta^2K^2)\\ &\qquad\leqslant \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\Big(\lambda(1-\beta\mu)\big(f(\theta) - \mathbb{E}f(\theta')\big)\Big)\exp(\lambda^2\beta^2K^2)\\ &\qquad\leqslant \Big(\sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\big(\lambda\big(f(\theta) - \mathbb{E}f(\theta')\big)\big)\Big)^{1-\beta\mu}\exp(\lambda^2\beta^2K^2)\\ &\quad\implies \sup_{f\in\mathrm{Lip}(\mathbb{R}^d)} \mathbb{E}\exp\big(\lambda(f(\theta)-\mathbb{E}f(\theta'))\big) \leqslant \exp(\lambda^2\beta K/\mu)\end{aligned}$$

The proof of (b) is analogous except for the fact that the above inequalities only hold for $|\lambda| \leqslant (\beta K)^{-1}$ when $f(G(\theta))$ is K-sub-exponential for all $f \in \mathrm{Lip}(\mathbb{R}^d)$. The rest of the proof is unchanged and since $\big(K\sqrt{\beta/\mu}\big)^{-1} < (\beta K)^{-1}$, we similarly obtain the sub-exponential property.

6.9.3 Proof of Wasserstein convergence and high-confidence bounds

Proof of Proposition 6.4

Let $\theta_1 \sim \nu_1$ and $\theta_2 \sim \nu_2$ be random variables such that $\mathcal{W}_2^2(\nu_1, \nu_2) = \mathbb{E}\big[\|\theta_1 - \theta_2\|^2\big]$. Such a pair of variables exists by [434, Theorem 4.1].

We consider the set of couplings of the distributions $\nu_1 P$ and $\nu_2 P$ through the random variables $G(\theta_1)$ and $G(\theta_2)$ such that

$$\theta_1 - \beta G(\theta_1) \sim \nu_1 P \quad \text{and} \quad \theta_2 - \beta G(\theta_2) \sim \nu_2 P.$$

Recall also that by Assumption 6.2, for $j = 1, 2$, conditionally on θ_j, we have

$$G(\theta_j) = \nabla \mathcal{L}(\theta_j) + \varepsilon(\theta_j) \quad \text{with} \quad \mathbb{E}[\varepsilon(\theta_j)|\theta_j] = 0. \tag{6.9.9}$$

Taking the infimum over such variables $G(\theta_j)$, we compute

$$\begin{aligned}
\mathcal{W}_2^2(\nu_1 P, \nu_2 P) &= \inf_{G(\theta_j)} \mathbb{E}\big\|\theta_1 - \beta G(\theta_1) - (\theta_2 - \beta G(\theta_2))\big\|^2 \\
&= \inf_{G(\theta_j)} \mathbb{E}\big[\big\|\theta_1 - \beta \nabla \mathcal{L}(\theta_1) - (\theta_2 - \beta \nabla \mathcal{L}(\theta_2))\big\|^2 \\
&\quad - 2\beta \langle \theta_1 - \beta \nabla \mathcal{L}(\theta_1) - (\theta_2 - \beta \nabla \mathcal{L}(\theta_2)), \varepsilon(\theta_1) - \varepsilon(\theta_2) \rangle \\
&\quad + \beta^2 \|\varepsilon(\theta_1) - \varepsilon(\theta_2)\|^2\big] \\
&\overset{①}{=} \mathbb{E}\big[\big\|\theta_1 - \beta \nabla \mathcal{L}(\theta_1) - (\theta_2 - \beta \nabla \mathcal{L}(\theta_2))\big\|^2 \\
&\quad + \beta^2 \inf_{G(\theta_j)} \mathbb{E}\big[\|\varepsilon(\theta_1) - \varepsilon(\theta_2)\|^2 | \theta_1, \theta_2\big]\big] \\
&\overset{②}{\leqslant} \mathbb{E}\big[(1 - \beta\mu)^2 \|\theta_1 - \theta_2\|^2 + \beta^2 \mathcal{W}_2^2\big(\mathcal{D}(\varepsilon(\theta_1)), \mathcal{D}(\varepsilon(\theta_2))\big)\big] \\
&\overset{③}{\leqslant} \mathbb{E}\big[\big((1 - \beta\mu)^2 + \beta^2 L_{\mathcal{W}}\big) \|\theta_1 - \theta_2\|^2\big] \\
&= \big((1 - \beta\mu)^2 + \beta^2 L_{\mathcal{W}}\big) \mathcal{W}_2^2(\nu_1, \nu_2),
\end{aligned}$$

where ① is obtained by conditioning on θ_1, θ_2 and using (6.9.9), ② uses Lemma 6.5 and ③ uses Assumption 6.5.

Since $\beta < \frac{2\mu}{\mu^2 + L_{\mathcal{W}}}$ by assumption, the obtained inequality shows that the mapping $\nu \to \nu P$ is a contraction in the space $\mathcal{P}_2(\mathbb{R}^d)$ endowed with the $\mathcal{W}_2$ metric which is complete and separable by [434, Theorem 6.18]. Consequently, by Banach's fixed-point theorem, the previous mapping admits a unique fixed point $\pi_\beta \in \mathcal{P}_2(\mathbb{R}^d)$ i.e. such that $\pi_\beta P = \pi_\beta$. Moreover, for any initial measure $\xi_0 \in \mathcal{P}_2(\mathbb{R}^d)$, the sequence $(\xi_n)_{n \in \mathbb{N}}$ defined by $\xi_n = \xi_0 P^n$ converges to π_β w.r.t. the $\mathcal{W}_2$ metric.

Proof of Corollary 6.1

By Theorem 6.1, we know that the Markov chain $(\theta_t)_{t \geqslant 0}$ is geometrically converging to the invariant distribution π_β w.r.t. the total-variation distance so that for any event $\mathcal{E} \in \mathcal{B}(\mathbb{R}^d)$, we

have

$$\big|\mathbb{P}\big(\theta_n \in \mathcal{E}\big) - \mathbb{P}_{\theta\sim\pi}\big(\theta \in \mathcal{E}\big)\big| \leqslant M\rho^n V(\theta_0). \tag{6.9.10}$$

By combining the above inequality with the conclusions of Proposition 6.2 and using Lemmas 6.3 and 6.4 in the sub-Gaussian and sub-exponential cases respectively we obtain the claimed bounds.

Proof of Corollary 6.2

Similarly to the proof of Corollary 6.1, inequality (6.9.10) holds by Theorem 6.1 for any event $\mathcal{E} \in \mathcal{B}(\mathbb{R}^d)$. For $\theta \sim \pi_\beta$ stationary, let $f(\theta) = \|\theta - \theta^\star\|$ and denote $\Delta = f(\theta) - \mathbb{E}f(\theta)$. Using Proposition 6.3 (a) and Chernoff's method for $t > 0$ and $\lambda > 0$, we have

$$\mathbb{P}(\Delta > t) = \mathbb{P}(e^{\lambda\Delta} > e^{\lambda t}) \leqslant \mathbb{E}\exp\big(\lambda\Delta - \lambda t\big) \leqslant \exp\big(\lambda^2 K^2\beta/\mu - \lambda t\big). \tag{6.9.11}$$

After minimizing over λ, we get for $\delta > 0$, with probability at least $1-\delta$, the following inequality holds:

$$\Delta \leqslant 2K\sqrt{\beta\log(1/\delta)/\mu}. \tag{6.9.12}$$

Additionally, by Proposition 6.1 (b) we have

$$\mathbb{E}f(\theta) = \mathbb{E}\|\theta - \theta^\star\| \leqslant \sqrt{\frac{\beta\sigma^2}{2\mu - \beta(\mu^2 + L_\sigma)}}. \tag{6.9.13}$$

Taking β such that $\beta \leqslant \frac{\mu}{\mu^2+L_\sigma}$ and combining (6.9.12) with (6.9.13) yields (6.6.1).

To obtain (6.6.2), we proceed similarly using Proposition 6.3 (b) this time. Applying the constraint $|\lambda| \leqslant (\beta K)^{-1}$ (see proof of Proposition 6.3) into the optimization of (6.9.11) yields

$$\mathbb{P}(\Delta > t) \leqslant \begin{cases} \exp\left(\frac{-t^2}{4\beta K^2/\mu}\right) & \text{if} \quad t \leqslant 2K/\mu \\ \exp\left(\frac{-t}{2\beta K}\right) & \text{otherwise.} \end{cases} \tag{6.9.14}$$

Expressing t in terms of the failure probability δ and combining with (6.9.13) as before finishes the proof.

Lemma 6.6. *Grant Assumption 6.1, 6.2, 6.5 and 6.6. Let the Markov chain $(\theta_t)_{t\geqslant}$ be initialized with $\theta_0 \sim \nu$ and β be chosen as in Proposition 6.4. The sequence of SGD iterates $\theta_0, \dots, \theta_n$ satisfies for $0 \leqslant i, j \leqslant n$:*

$$\mathbb{E}\langle\theta_i - \theta^\star, \theta_j - \theta^\star\rangle \leqslant 2(1-\beta\mu)^{|i-j|}\big(\big((1-\beta\mu) + \beta^2 L_{\mathcal{W}}\big)^i \mathcal{W}_2^2(\nu,\pi) + \mathrm{Var}_\pi(\theta)\big).$$

Proof. We assume without loss of generality that $i \leqslant j$. Since the gradient is linear it commutes with the expectation. Therefore, by conditioning over θ_{j-1} and later over θ_{j-1} we find

$$\begin{aligned}
\mathbb{E}\langle\theta_j - \theta^\star, \theta_i - \theta^\star\rangle &= \mathbb{E}\langle\theta_{j-1} - \beta G(\theta_{j-1}) - \theta^\star, \theta_i - \theta^\star\rangle \\
&= \mathbb{E}\langle\theta_{j-1} - \beta\nabla\mathcal{L}(\theta_{j-1}) - \theta^\star, \theta_i - \theta^\star\rangle \\
&= \mathbb{E}\langle\theta_{j-2} - \beta G(\theta_{j-2}) - \beta\nabla\mathcal{L}(\theta_{j-2} - \beta G(\theta_{j-2})) - \theta^\star, \theta_i - \theta^\star\rangle \\
&= \mathbb{E}\langle\theta_{j-2} - \beta\nabla\mathcal{L}(\theta_{j-2}) - \beta\nabla\mathcal{L}(\theta_{j-2} - \beta\nabla\mathcal{L}(\theta_{j-2})) - \theta^\star, \theta_i - \theta^\star\rangle.
\end{aligned}$$

It is clear that the previous steps can be repeated for the remaining indices $j-3, j-4, \dots$ down to i at which point the following identity is reached

$$\mathbb{E}\langle\theta_j - \theta^\star, \theta_i - \theta^\star\rangle = \mathbb{E}\langle\breve{\theta}_j - \theta^\star, \theta_i - \theta^\star\rangle,$$

where $\breve{\theta}_j$ is recursively defined by $\breve{\theta}_i = \theta_i$ and $\breve{\theta}_k = \breve{\theta}_{k-1} - \beta\nabla\mathcal{L}(\breve{\theta}_{k-1})$ for $i < k \leqslant j$.

Using Cauchy-Schwarz and iterating the inequality $\|\breve{\theta}_k - \theta^\star\| \leqslant (1-\beta\mu)\|\breve{\theta}_{k-1} - \theta^\star\|$ yields that

$$\mathbb{E}\langle\theta_j - \theta^\star, \theta_i - \theta^\star\rangle \leqslant (1-\beta\mu)^{j-i}\mathbb{E}\|\theta_i - \theta^\star\|^2.$$

Now, by [434, Theorem 4.1], there exists a random variable $\widetilde{\theta} \sim \pi_\beta$ such that the coupling $(\theta_i, \widetilde{\theta})$ satisfies

$$\mathbb{E}\|\theta_i - \widetilde{\theta}\|^2 = \mathcal{W}_2^2(\mathcal{D}(\theta_i), \pi) = \mathcal{W}_2^2(\nu P^i, \pi) \leqslant \big((1-\beta\mu) + \beta^2 L_{\mathcal{W}}\big)^i \mathcal{W}_2^2(\nu, \pi),$$

where the inequality comes from Proposition 6.4. It then only remains to write

$$\mathbb{E}\|\theta_i - \theta^\star\|^2 \leqslant 2\big(\mathbb{E}\|\theta_i - \widetilde{\theta}\|^2 + \mathbb{E}\|\widetilde{\theta} - \theta^\star\|^2\big) \leqslant 2\big((1-\beta\mu) + \beta^2 L_{\mathcal{W}}\big)^i \mathcal{W}_2^2(\nu, \pi) + 2\,\mathrm{Var}_\pi(\theta),$$

which implies the result. The case $i \geqslant j$ is handled similarly. □

Lemma 6.7. *Let $A \in \mathbb{R}^{n\times n}$ be a matrix with positive entries such that there exists $C > 0$ and $0 < \alpha < 1$ such that*

$$A_{ij} \leqslant C\alpha^{|j-i|} \quad \text{for} \quad 1 \leqslant i, j \leqslant n,$$

then we have

$$\sum_{i,j} A_{ij} \leqslant C\Big(n + \frac{2\alpha}{1-\alpha}\Big(n - \Big(\frac{1-\alpha^n}{1-\alpha}\Big)\Big)\Big).$$

Proof. Straightforward computations yield

$$\sum_{i,j} A_{ij} = \sum_{i=1}^n A_{ii} + 2\sum_{i<j} A_{ij} \leqslant nC + 2\sum_{i<j} A_{ij},$$

and we have

$$\begin{aligned}\sum_{i=1}^n \sum_{j=i+1}^n A_{ij} &\leqslant C\sum_{i=1}^n \sum_{j=i+1}^n \alpha^{j-i} = C\alpha\sum_{i=1}^n \frac{1-\alpha^{n-i}}{1-\alpha} \\ &= \frac{C\alpha}{1-\alpha}\Big(n - \sum_{i=1}^n \alpha^{n-i}\Big) = \frac{C\alpha}{1-\alpha}\Big(n - \frac{1-\alpha^n}{1-\alpha}\Big).\end{aligned}$$

□

Proof of Theorem 6.2

We introduce the notations $\theta_{[i]} = (\theta_1, \theta_2, \ldots, \theta_i)$ and $\theta_{[k,l]} = (\theta_k, \theta_{k+1}, \ldots, \theta_l)$ and define for $1 \leqslant i \leqslant n$ the variables

$$M^{(i)} := \mathbb{E}\big[f(\vec{\theta})|\theta_{[i]}\big] - \mathbb{E}\big[f(\vec{\theta})|\theta_{[i-1]}\big] \quad \text{so that} \quad f(\vec{\theta}) - \mathbb{E}[f(\vec{\theta})] = \sum_{i=1}^n M^{(i)}.$$

Notice that, if we condition on $\theta_{[i-1]}$ then $M^{(i)}$ only depends on θ_i. We consider $M^{(i)}$ as a function of θ_i and compute its Lipschitz constant. We temporarily consider θ_i and θ_i' as two fixed deterministic vectors of $\mathbb{R}^d$ and $\theta_{i+1}, \theta_{i+2}, \ldots$ and $\theta_{i+1}', \theta_{i+2}', \ldots$ are the SGD trajectories obtained from them i.e. for $j > i$:

$$\theta_j = \theta_{j-1} - \beta G(\theta_{j-1}) \quad \text{and} \quad \theta_j' = \theta_{j-1}' - \beta G(\theta_{j-1}').$$

In the following, we use the Lipschitz property of f and the Kantorovich-Rubinstein dual representation of the $\mathcal{W}_1$ metric

$$\mathcal{W}_1(\nu_1,\nu_2) = \sup_{h\in \mathrm{Lip}(\mathbb{R}^d)} \int h d\nu_1 - \int h d\nu_2,$$

in order to find

$$\begin{aligned}
\big|M^{(i)}(\theta_i) - M^{(i)}(\theta_i')\big| &= \Big|\mathbb{E}\big[f(\vec{\theta})|\theta_{[i]}\big] - \mathbb{E}\big[f(\theta_{[i-1]},\theta'_{[i,n]})|\theta_i',\theta_{[i-1]}\big]\Big| \\
&= \Big|\sum_{j=i}^{n} \mathbb{E}\Big[f\big(\theta_{[i-1]},\theta'_{[i,j-1]},\theta_{[j,n]}\big) - f\big(\theta_{[i-1]},\theta'_{[i,j]},\theta_{[j+1,n]}\big)|\theta_{[i-1]}\Big]\Big| \\
&\leqslant \sum_{j=i}^{n} \mathcal{W}_1\big(\mathcal{D}(\theta_j),\mathcal{D}(\theta_j')\big).
\end{aligned}$$

Using Proposition 6.4 we have

$$\begin{aligned}
\mathcal{W}_1\big(\mathcal{D}(\theta_j),\mathcal{D}(\theta_j')\big) &\leqslant \mathcal{W}_2\big(\mathcal{D}(\theta_j),\mathcal{D}(\theta_j')\big) = \mathcal{W}_2\big(\mathcal{D}(\theta_{j-1})P,\mathcal{D}(\theta'_{j-1})P\big) \\
&\leqslant \underbrace{\sqrt{(1-\beta\mu)^2+\beta^2 L_{\mathcal{W}}}}_{=:\alpha_{\mathcal{W}}(\beta,\mu)} \mathcal{W}_2\big(\mathcal{D}(\theta_{j-1}),\mathcal{D}(\theta'_{j-1})\big) \\
&\leqslant \dots \\
&\leqslant \alpha_{\mathcal{W}}(\beta,\mu)^{j-i}\mathcal{W}_2\big(\mathcal{D}(\theta_i),\mathcal{D}(\theta_i')\big) = \alpha_{\mathcal{W}}(\beta,\mu)^{j-i}\big\|\theta_i-\theta_i'\big\|,
\end{aligned}$$

where the last equality follows from θ_i and θ_i' being deterministic. Provided that $\beta < \frac{2\mu}{\mu^2+L_{\mathcal{W}}}$ we have $\alpha_{\mathcal{W}}(\beta,\mu) < 1$ so that $\mathcal{W}_1(\theta_j,\theta_j') \leqslant \alpha_{\mathcal{W}}(\beta,\mu)^{j-i}\|\theta_i-\theta_i'\|$ for $i \leqslant j \leqslant n$. By summing over j, we find that the $M^{(i)}$s are $(1-\alpha_{\mathcal{W}}(\beta,\mu))^{-1}$-Lipschitz

$$\big|M^{(i)}(\theta_i) - M^{(i)}(\theta_i')\big| \leqslant \frac{\|\theta_i-\theta_i'\|}{1-\alpha_{\mathcal{W}}(\beta,\mu)}.$$

In what follows we denote $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot|\theta_{[k]}]$ to lighten notation and let $C_{\mathcal{W}} := (1-\alpha_{\mathcal{W}}(\beta,\mu))^{-1}$. Let $\lambda \in \mathbb{R}$, by conditioning on $\theta_{[n-1]}$, we have

$$\begin{aligned}
\mathbb{E}\exp\big(\lambda(f(\vec{\theta}) - \mathbb{E}f(\vec{\theta}))\big) &= \mathbb{E}\exp\Big(\lambda\sum_{i=1}^{n} M^{(i)}\Big) = \mathbb{E}\Big[\mathbb{E}\Big[\exp\Big(\lambda\sum_{i=1}^{n} M^{(i)}\Big)|\theta_{[n-1]}\Big]\Big] \\
&= \mathbb{E}\Big[\exp\Big(\lambda\sum_{i=1}^{n-1} M^{(i)}\Big)\mathbb{E}\big[\exp(\lambda M^{(n)})|\theta_{[n-1]}\big]\Big].
\end{aligned}$$

Recall that conditionally on $\theta_{[n-1]}$, we have that $M^{(n)}$ is a function of $\theta_n = \theta_{n-1} - \beta G(\theta_{n-1})$ so that $M^{(n)}$ is a $\beta C_{\mathcal{W}}$-Lipschitz function of $G(\theta_{n-1})$ which satisfies Assumption 6.4 (a) and thus

$$\mathbb{E}\big[\exp(\lambda M^{(n)})|\theta_{[n-1]}\big] \leqslant \exp\big(\lambda^2\beta^2 C_{\mathcal{W}}^2 K^2\big).$$

By repeating this argument $n-1$ times, we arrive at

$$\begin{aligned}
\mathbb{E}\exp\big(\lambda(f(\vec{\theta}) - \mathbb{E}f(\vec{\theta}))\big) &\leqslant \mathbb{E}\big[\exp\big(\lambda M^{(1)}\big)\big]\exp\big((n-1)\lambda^2\beta^2 C_{\mathcal{W}}^2 K^2\big) \\
&\leqslant \exp\big(\lambda^2 C_{\mathcal{W}}^2 K^2\beta/\mu + (n-1)\lambda^2\beta^2 C_{\mathcal{W}}^2 K^2\big),
\end{aligned}$$

where the last inequality uses that $\theta_1 \sim \pi_\beta$ which is $K\sqrt{\beta/\mu}$-sub-Gaussian by Proposition 6.3 (a).

The proof in the sub-exponential case is completely analogous using Assumption 6.4 (b) and the result of Proposition 6.3 (b) with the main difference that the obtained inequalities only hold for $|\lambda| \leqslant \big(C_{\mathcal{W}} K\sqrt{\beta/\mu}\big)^{-1} \wedge \big(\beta C_{\mathcal{W}} K\big)^{-1} = \big(C_{\mathcal{W}} K\sqrt{\beta/\mu}\big)^{-1}$ because $\beta < \mu^{-1}$.

Proof of Proposition 6.6

For $j \geqslant 0$, we introduce the notation

$$\Delta_j := \Big\| \sum_{t=j+1}^{j+n} \theta_t - n\theta^\star \Big\| - \mathbb{E}\Big\| \sum_{t=j+1}^{j+n} \theta_t - n\theta^\star \Big\|.$$

We are interested in obtaining a high probability bound on the quantity Δ_{n_0}. We write $\mathbb{E}_\nu$ for the expectation when the Markov chain is started with distribution ν

$$\begin{aligned}\mathbb{E}_\nu\big[\exp\big(\lambda\Delta_{n_0}\big)\big] &= \mathbb{E}_{\nu P^{n_0}}\big[\exp\big(\lambda\Delta_1\big)\big] = \mathbb{E}_\pi\Big[\frac{d(\nu P^{n_0})}{d\pi}\exp\big(\lambda\Delta_1\big)\Big] \\ &\leqslant \Big\|\frac{d(\nu P^{n_0})}{d\pi}\Big\|_{\pi,\infty}\mathbb{E}_\pi\big[\exp\big(\lambda\Delta_1\big)\big].\end{aligned}$$

The last expectation can be bounded using Theorem 6.2. As for the factor coming from the measure change, we write

$$\Big\|\frac{d(\nu P^{n_0})}{d\pi}\Big\|_{\pi,\infty} \leqslant \Big\|\frac{d(\nu P^{n_0} - \pi)}{d\pi}\Big\|_{\pi,\infty} + 1.$$

For any function $F : \mathbb{R}^d \to \mathbb{R}$, we define the norm $\|F\|_V = \sup_{\vartheta\in\mathbb{R}^d} \frac{|F(\vartheta)|}{V(\vartheta)}$ and its induced operator norm $|||Q|||_V = \sup_F \frac{\|QF\|_V}{\|F\|_V}$, where V is the function defined in Section 6.9.2. We also denote $\odot$ the pointwise product between functions.

$$\begin{aligned}\Big\|\frac{d(\nu P^{n_0} - \pi)}{d\pi}\Big\|_{\pi,\infty} &= \Big\|\frac{d(\nu(P^{n_0} - \mathbf{1}\otimes\pi))}{d\pi}\Big\|_{\pi,\infty} = \Big\|(P^{n_0} - \mathbf{1}\otimes\pi)^*\frac{d\nu}{d\pi}\Big\|_{\pi,\infty} \\ &= \Big\|(P^{n_0} - \mathbf{1}\otimes\pi)^*\frac{d\nu}{d\pi}\odot V\odot\frac{1}{V}\Big\|_{\pi,\infty} \\ &\leqslant \Big\|(P^{n_0} - \mathbf{1}\otimes\pi)^*\frac{d\nu}{d\pi}\odot V\Big\|_V \\ &\leqslant |||(P^{n_0} - \mathbf{1}\otimes\pi)^*|||_V\Big\|\frac{d\nu}{d\pi}\odot V\Big\|_V = |||P^{n_0} - \mathbf{1}\otimes\pi|||_V\Big\|\frac{d\nu}{d\pi}\Big\|_\infty.\end{aligned}$$

The outer product $\mathbf{1}\otimes\pi_\beta$ denotes the kernel such that $\mathbf{1}\otimes\pi(\vartheta, A) = \pi(A)$ for all ϑ and $A \in \mathcal{B}(\mathbb{R}^d)$. By the proof of Theorem 6.1 and [243, Proposition 1.1] (see also Equation (4)) the kernel P has a spectral gap in the Banach space L^V_∞ of functions with finite norm $\|\cdot\|_V$ and, therefore, there exist $\rho < 1$ and $M < \infty$ such that

$$|||P^{n_0} - \mathbf{1}\otimes\pi|||_V \leqslant M\rho^{n_0},$$

which leads to

$$\Big\|\frac{d(\nu P^{n_0})}{d\pi}\Big\|_{\pi,\infty} \leqslant 1 + M\rho^{n_0}\Big\|\frac{d\nu}{d\pi}\Big\|_\infty = \Upsilon(\nu, n_0).$$

Using Theorem 6.2 in the sub-Gaussian case, denoting $\breve{K} = KC_{\mathcal{W}}\sqrt{\beta/\mu + (n-1)\beta^2}$, we find

$$\mathbb{E}_\nu\Big[\exp\big(\lambda\Delta_{n_0}\big)\Big] \leqslant \Upsilon(\nu, n_0)\exp(\lambda^2\breve{K}^2).$$

Using Chernoff's method for a random variable $X \in \Psi_2(\breve{K})$ and $t > 0$ and $\lambda > 0$, we have

$$\mathbb{P}_\nu(\Delta_{n_0} > t) = \mathbb{P}_\nu(e^{\lambda\Delta_{n_0}} > e^{\lambda t}) \leqslant \mathbb{E}_\nu \exp\big(\lambda\Delta_{n_0} - \lambda t\big) \leqslant \Upsilon(\nu, n_0)\exp\big(\lambda^2\breve{K}^2 - \lambda t\big).$$

After minimizing over λ, we get for $\delta > 0$, with probability at least $1 - \Upsilon(\nu, n_0)\delta$, the following inequality holds

$$\Delta_{n_0} \leqslant 2\breve{K}\sqrt{\log(1/\delta)}. \tag{6.9.15}$$

In the sub-exponential case (under Assumption 6.4 (b)), taking the constraint $|\lambda| \leqslant \big(C_{\mathcal{W}}K\sqrt{\beta/\mu}\big)^{-1}$ into account (see the proof of Theorem 6.2), we get that

$$\mathbb{P}(\Delta_{n_0} > t) \leqslant \begin{cases} \Upsilon(\nu, n_0)\exp\left(\frac{-t^2}{4\breve{K}^2}\right) & \text{if} \quad t \leqslant \frac{2\breve{K}^2}{C_{\mathcal{W}}K\sqrt{\beta/\mu}} \\ \Upsilon(\nu, n_0)\exp\left(\frac{-t}{2C_{\mathcal{W}}K\sqrt{\beta/\mu}}\right) & \text{otherwise.} \end{cases}$$

So that with probability at least $1 - \Upsilon(\nu, n_0)\delta$:

$$\Delta_{n_0} \leqslant \max\big(2\breve{K}\sqrt{\log(1/\delta)}, 2C_{\mathcal{W}}K\sqrt{\beta/\mu}\log(1/\delta)\big). \tag{6.9.16}$$

It then only remains to bound the expectation $\mathbb{E}\Big\|\sum_{t=n_0+1}^{n_0+n}\theta_t - n\theta^\star\Big\|$, which can be done as follows

$$\Big(\mathbb{E}\Big\|\sum_{t=n_0+1}^{n_0+n}\theta_t - n\theta^\star\Big\|\Big)^2 \leqslant \mathbb{E}\Big\|\sum_{t=n_0+1}^{n_0+n}(\theta_t - \theta^\star)\Big\|^2 = \sum_{i=n_0+1}^{n_0+n}\sum_{j=n_0+1}^{n_0+n}\mathbb{E}\langle\theta_i - \theta^\star, \theta_j - \theta^\star\rangle.$$

Using Lemmas 6.6 and 6.7 we find that

$$\Big(\mathbb{E}\Big\|\sum_{t=n_0+1}^{n_0+n}\theta_t - \theta^\star\Big\|\Big)^2 \leqslant 2n\frac{1+\alpha}{1-\alpha}\Big(\alpha_{\mathcal{W}}^{n_0}\mathcal{W}_2^2(\nu, \pi) + \mathrm{Var}_\pi(\theta)\Big),$$

where $\alpha = 1 - \beta\mu$ and $\alpha_{\mathcal{W}} = \sqrt{\alpha^2 + \beta^2 L_{\mathcal{W}}}$. Moreover, since $\beta < \frac{\mu}{\mu^2 + L_\sigma}$, by Proposition 6.1 we have

$$\mathrm{Var}_\pi(\theta) \leqslant \frac{\beta\sigma^2}{\mu}.$$

Plugging into Inequalities (6.9.15) and (6.9.16) and dividing by n finishes the proof.

Proof of Lemma 6.2

Denote $\Xi_t^{(N)} = \frac{1}{N}\sum_{i=1}^N \Xi_{tN+i}$ and $\xi_t^{(N)} = \frac{1}{N}\sum_{i=1}^N \xi_{tN+i}$. By Lemma 6.8, we have the following concentration inequalities for all $0 \leqslant t < T$:

$$\mathbb{P}\Big(\|\Xi_t^{(N)}\|_2 > 3K_\Xi\Big(\frac{\log(4T/\delta) + 3d}{N} \vee \sqrt{\frac{\log(4T/\delta) + 3d}{N}}\Big)\Big) \leqslant \delta/(2T)$$
$$\mathbb{P}\Big(\|\xi_t^{(N)}\| > 4K_\xi\Big(\frac{\log(4T/\delta) + 2d}{N} \vee \sqrt{\frac{\log(4T/\delta) + 2d}{N}}\Big)\Big) \leqslant \delta/(2T).$$

We will show by induction over $0 \leqslant t \leqslant T$ that we have with probability at least $1 - t\delta/T$ that

$$\max_{0 \leqslant s \leqslant t} \|\theta_s - \theta^\star\| \leqslant C. \tag{6.9.17}$$

The case $t = 0$ holds by assumption. Further, assuming the property at rank t and conditioning on θ_t we have with probability at least $1 - \delta/T$:

$$\begin{aligned}
\|\theta_{t+1} - \theta^\star\|^2 &= \|\theta_t - \beta\nabla\mathcal{L}(\theta_t) - \beta(\Xi_t^{(N)}(\theta_t - \theta^\star) + \xi_t^{(N)}) - \theta^\star\|^2 \\
&= \|\theta_t - \beta\nabla\mathcal{L}(\theta_t) - \theta^\star\|^2 - 2\beta\langle\theta_t - \beta\nabla\mathcal{L}(\theta_t) - \theta^\star, \Xi_t^{(N)}(\theta_t - \theta^\star) + \xi_t^{(N)}\rangle \\
&\quad + \beta^2\|\Xi_t^{(N)}(\theta_t - \theta^\star) + \xi_t^{(N)}\|^2 \\
&\overset{①}{\leqslant} (1-\beta\mu)^2\|\theta_t - \theta^\star\|^2 + 2\beta(1-\beta\mu)\|\theta_t - \theta^\star\|(\|\Xi_t^{(N)}(\theta_t - \theta^\star)\| + \|\xi_t^{(N)}\|) \\
&\quad + 2\beta^2\|\Xi_t^{(N)}(\theta_t - \theta^\star)\|^2 + 2\beta^2\|\xi_t^{(N)}\|^2 \\
&\leqslant \big[(1-\beta\mu)^2 + 2\beta(1-\beta\mu)\|\Xi_t^{(N)}\|_2 + 2\beta^2\|\Xi_t^{(N)}\|_2^2\big]\|\theta_t - \theta^\star\|^2 \\
&\quad + 2\beta(1-\beta\mu)\|\theta_t - \theta^\star\|\|\xi_t^{(N)}\| + 2\beta^2\|\xi_t^{(N)}\|^2 \\
&\overset{②}{\leqslant} \big[(1-\beta\mu)^2(1+\epsilon) + 2\beta(1-\beta\mu)\|\Xi_t^{(N)}\|_2 + 2\beta^2\|\Xi_t^{(N)}\|_2^2\big]\|\theta_t - \theta^\star\|^2 \\
&\quad + \beta^2(2 + 1/\epsilon)\|\xi_t^{(N)}\|^2 \\
&\overset{③}{\leqslant} \big[(1-\beta\mu) + 2\beta\|\Xi_t^{(N)}\|_2 + 2\beta^2\|\Xi_t^{(N)}\|_2^2\big]\|\theta_t - \theta^\star\|^2 + 3\frac{\beta}{\mu}\|\xi_t^{(N)}\|^2 \\
&\overset{④}{\leqslant} \big[(1-\beta\mu) + \beta\mu/3 + \beta\mu/3\big]C^2 + \beta\mu C^2/3 \leqslant C^2,
\end{aligned}$$

where ① uses Lemma 6.5 and the Cauchy-Schwarz inequality, ② uses the inequality $2ab \leqslant a^2\epsilon + b^2/\epsilon$ valid for all $\epsilon > 0$ and ③ sets the choice $\epsilon = \beta\mu$ and uses that $\beta \leqslant 1/\mu$. Finally ④ uses the high probability bounds stated above and the conditions on N and β.

Using a union bound argument, we obtain (6.9.17) for $t+1$ with probability at least $1 - (t+1)\delta/T$. The induction argument is completed and implies the result for $t = T$.

Lemma 6.8. *Let $\Xi_1, \dots, \Xi_N \in \mathbb{R}^{d\times d}$ be random matrices and $\xi_1, \dots, \xi_N \in \mathbb{R}^d$ random vectors as in Lemma 6.2. Then denoting $\overline{\Xi} = \frac{1}{N}\sum_{i=1}^N \Xi_i$ and $\overline{\xi} = \frac{1}{N}\sum_{i=1}^N \xi_i$, we have the high probability bounds*

$$\mathbb{P}\Big(\|\overline{\Xi}\|_2 > 3K_\Xi\phi\Big(\frac{\log(2/\delta) + 3d}{N}\Big)\Big) \leqslant \delta, \tag{6.9.18}$$

$$\mathbb{P}\Big(\|\overline{\xi}\| > 4K_\xi\phi\Big(\frac{\log(2/\delta) + 2d}{N}\Big)\Big) \leqslant \delta, \tag{6.9.19}$$

where $\phi(x) = \max(x, \sqrt{x})$.

Proof. We first prove (6.9.18). Denote $S^{d-1} = \{u \in \mathbb{R}^d \ : \ \|u\| = 1\}$ and let $u \in S^{d-1}$ and $|\lambda| \leqslant N/K_\Xi$, we have

$$\mathbb{E}\exp(\lambda\langle u, \overline{\Xi}u\rangle) = \prod_{i=1}^N \mathbb{E}\exp(\lambda\langle u, \Xi_i u\rangle/N) \leqslant \prod_{i=1}^N \exp(\lambda^2 K_\Xi^2/N^2) = \exp(\lambda^2 K_\Xi^2/N),$$

so that for all $u \in S^{d-1}$ we have $\langle u, \overline{\Xi}u\rangle \in \Psi_1(K_\Xi/\sqrt{N})$.

Let Ω_ϵ be an ϵ-net of S^{d-1}. By [433, Lemma 5.2], there exists an ϵ-net such that $|\Omega_\epsilon| \leqslant$

$(1+2/\epsilon)^d$ and for all $u \in S^{d-1}$ there exists $v \in \Omega_\epsilon$ such that $\|u-v\| \leqslant \epsilon$. We write

$$\langle u, \overline{\Xi} u\rangle = \langle v, \overline{\Xi} v\rangle + 2\langle u-v, \overline{\Xi} v\rangle + \langle u-v, \overline{\Xi}(u-v)\rangle,$$

which allows us to deduce that

$$\|\overline{\Xi}\|_2 = \sup_{u\in S^{d-1}} |\langle u, \overline{\Xi} u\rangle| \leqslant \sup_{v\in\Omega_\epsilon} |\langle v, \overline{\Xi} v\rangle| + (2\epsilon+\epsilon^2)\|\overline{\Xi}\|_2 \implies \|\overline{\Xi}\|_2 \leqslant \frac{\sup_{v\in\Omega_\epsilon} |\langle v, \overline{\Xi} v\rangle|}{1-2\epsilon-\epsilon^2}.$$

Let $v \in \Omega_\epsilon$, using Chernoff's method and the sub-exponential property of $\overline{\Xi}$ (see also the proof of Corollary 6.2), we find for $t > 0$:

$$\mathbb{P}(|\langle v, \overline{\Xi} v\rangle| > t) \leqslant \begin{cases} 2\exp(-Nt^2/(4K_\Xi^2)) & \text{if} \quad t \leqslant 2K_\Xi \\ 2\exp(-Nt/(2K_\Xi)) & \text{otherwise.} \end{cases}$$

Reformulating in terms of a failure probability δ, we find that

$$\mathbb{P}\Big(|\langle v, \overline{\Xi} v\rangle| > 2K_\Xi \phi\Big(\frac{\log(2/\delta)}{N}\Big)\Big) \leqslant \delta.$$

Replacing δ with $\delta/(1+2/\epsilon)^d$ and using a union bound argument over Ω_ϵ we find

$$\mathbb{P}\Big(\sup_{v\in\Omega_\epsilon} |\langle v, \overline{\Xi} v\rangle| > 2K_\Xi \phi\Big(\frac{\log(2/\delta) + d\log(1+2/\epsilon)}{N}\Big)\Big) \leqslant \delta.$$

It only remains to set $\epsilon = 1/8$ and plug back into the inequality $\|\overline{\Xi}\|_2 \leqslant \frac{\sup_{v\in\Omega_\epsilon} |\langle v, \overline{\Xi} v\rangle|}{1-2\epsilon-\epsilon^2}$ in order to obtain (6.9.18).

To prove (6.9.19), we proceed similarly and first obtain for all $u \in S^{d-1}$ and $|\lambda| \leqslant N/K_\xi$:

$$\mathbb{E}\exp(\langle u, \overline{\xi}\rangle) \leqslant \exp(\lambda^2 K_\xi^2/N).$$

For $u \in S^{d-1}$ and $v \in \Omega_\epsilon$ such that $\|u-v\| \leqslant \epsilon$, we write $\langle u, \overline{\xi}\rangle = \langle v, \overline{\xi}\rangle + \langle u-v, \overline{\xi}\rangle$ which yields the inequality

$$\|\overline{\xi}\| \leqslant \frac{\sup_{v\in\Omega_\epsilon} |\langle v, \overline{\xi}\rangle|}{1-\epsilon}.$$

As before, the sub-exponential property of $\overline{\xi}$ yields

$$\mathbb{P}\Big(|\langle v, \overline{\xi}\rangle| > 2K_\xi \phi\Big(\frac{\log(2/\delta)}{N}\Big)\Big) \leqslant \delta,$$

and using another union bound argument over Ω_ϵ we find

$$\mathbb{P}\Big(\sup_{v\in\Omega_\epsilon} |\langle v, \overline{\xi}\rangle| > 2K_\xi \phi\Big(\frac{\log(2/\delta) + d\log(1+2/\epsilon)}{N}\Big)\Big) \leqslant \delta.$$

It only remains to set $\epsilon = 1/2$ to finish the proof of (6.9.19). □

Chapter 7

WildWood: a New Random Forest Algorithm

This chapter is based on the article [154] in collaboration with Stéphane Gaïffas and Yiyang Yu.

Contents

Abstract

We introduce WildWood (WW), a new ensemble algorithm for supervised learning of Random Forest (RF) type. While standard RF algorithms use bootstrap out-of-bag samples to compute out-of-bag scores, WW uses these samples to produce improved predictions given by an aggregation of the predictions of all possible subtrees of each fully grown tree in the forest. This is achieved by aggregation with exponential weights computed over out-of-bag samples, that are computed exactly and very efficiently thanks to an algorithm called context tree weighting. This improvement, combined with a histogram strategy to accelerate split finding, makes WW fast and competitive compared with other well-established ensemble methods, such as standard RF and extreme gradient boosting algorithms.

7.1 Introduction

This paper introduces WildWood (WW), a new ensemble method of Random Forest (RF) type [52]. The main contributions of the paper and the main advantages of WW are as follows. Firstly, we use out-of-bag samples (trees in a RF use different bootstrapped samples) very differently than what is done in standard RF [281, 34]. Indeed, WW uses these samples to compute an aggregation of the predictions of all possible subtrees of each tree in the forest, using aggregation with exponential weights [75]. This leads to much improved predictions: while only leaves contribute to the predictions of a tree in standard RF, the full tree structure contributes to predictions in WW. An illustration of this effect is given in Figure 7.1 on a toy binary classification example, where we can observe that subtrees aggregation leads to improved and regularized decision functions for each individual tree and for the forest. We further illustrate

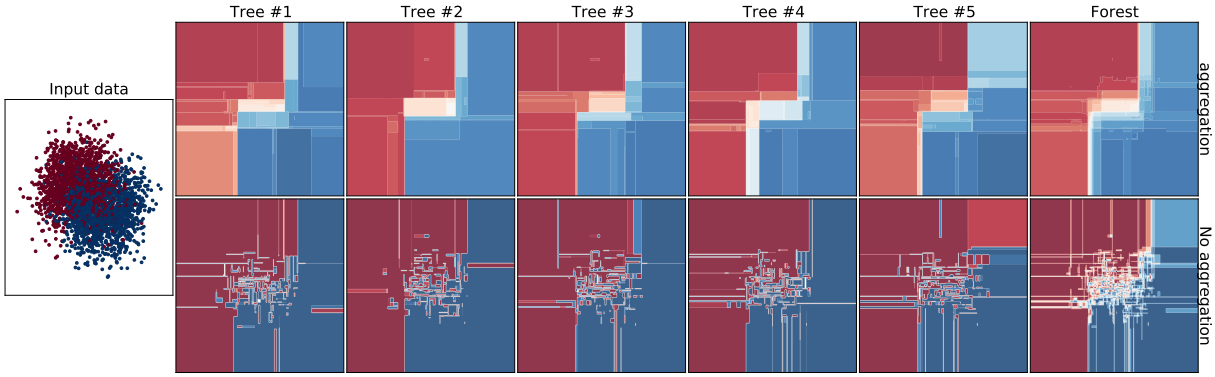

Figure 7.1: WW decision functions illustrated on a toy dataset (left) with subtrees aggregation (top) and without it (bottom). Subtrees aggregation improves trees predictions, as illustrated by smoother decision functions in the top compared with the bottom, improving overall predictions of the forest (last column).

in Figure 7.2 that each tree becomes a stronger learner, and that excellent performance can be achieved even when WW uses few trees, granting better interpretability of the solutions as a side benefit. Indeed, reducing the number of trees is a known way to obtain explainable models [459, 221]. A remarkable aspect of WW is that this improvement comes only at a small computational cost, thanks to a technique called "context tree weighting", used in lossless compression or online learning to aggregate all subtrees of a given tree [441, 440, 187, 75, 327]. Also, the predictions of WW do not rely on MCMC approximations required with Bayesian variants of RF [94, 105, 93, 412], which is a clear distinction from such methods. Secondly, WW uses feature binning ("his-

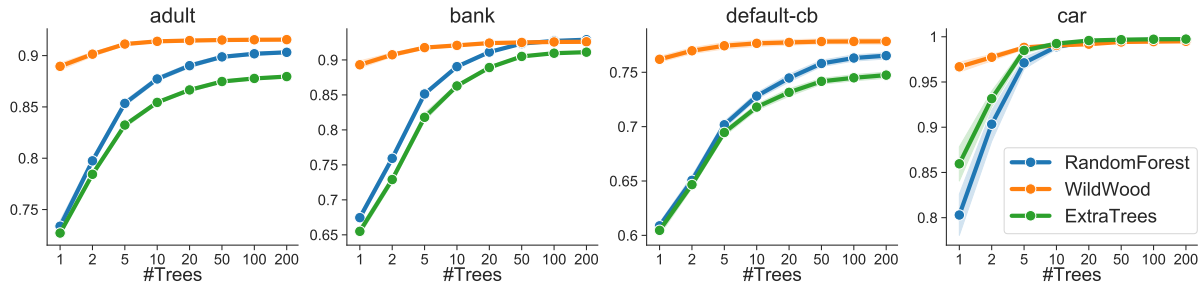

Figure 7.2: Mean test AUC and standard-deviations (y-axis) using 10 train/test splits for WW and `scikit-learn`'s implementations of RF [281] and Extra Trees [163], using default hyperparameters, on several datasets. Thanks to subtrees aggregation, WW improves these baselines, even with few trees (x-axis is the number of trees).

togram" strategy), similarly to extreme gradient boosting (EGB) libraries such as XGBoost [87], LightGBM [231] and CatBoost [367, 134]. This strategy helps accelerate computations in WW compared with standard RF algorithms, that typically require to sort features locally in nodes and try a larger number of splits [281]. This combination of subtrees aggregation and of the histogram strategy makes WW comparable with state-of-the-art implementations of EGB libraries, as illustrated in Figure 7.3. Moreover, WW supports optimal split finding for categorical features

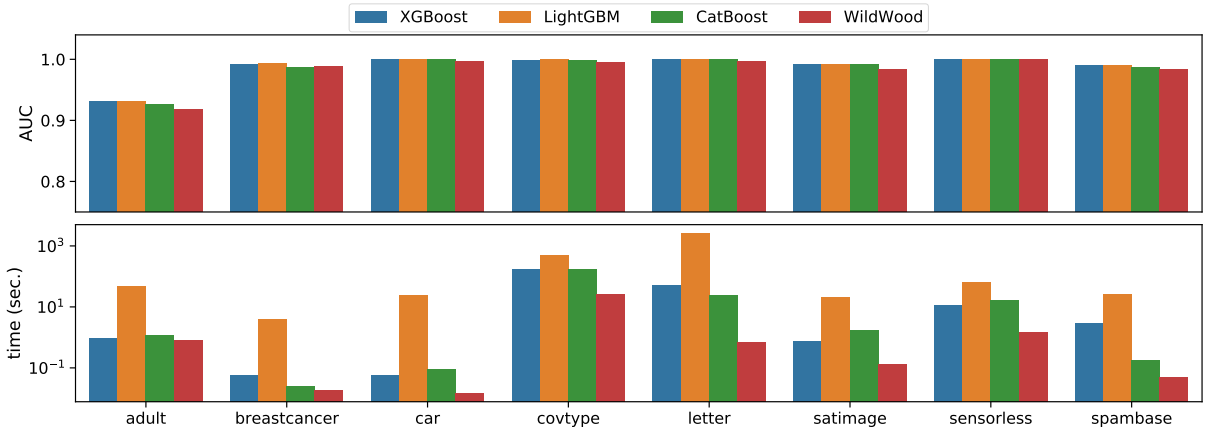

Figure 7.3: Test AUC (top) and training time (bottom) of WW compared with very popular EGB libraries (after hyperoptimization of all algorithms, see Section 7.4 for details). WW's performance, which uses only 10 trees in this display, is only slightly below such strong baselines, but is faster (training times are on a logarithmic scale) on the considered datasets.

and missing values, with no need for particular pre-processing (such as one-hot encoding, see [87] or target encoding, see [367, 134]). Finally, WW is supported by some theoretical evidence, since we prove that for a general loss function, the subtrees aggregation considered in WW leads indeed to a performance close to that of the best subtree.

Related works. Tree based decision models appeared as a supervised learning tool in the 1960s [322, 312, 370]. Their use was promoted by the in-depth practical and theoretical study of [54]. The key idea of forming tree ensembles led to the well-known Random Forest algorithm [188, 52] which became one of the most popular supervised learning algorithms thanks to its ease of use, robustness to hyperparameters [34, 366] and applicability to a wide range of domains. Recent examples include predictive medicine [411, 5], intrusion detection [90], car safety [413], differential privacy [352] and COVID-19 [404] to cite but a few.

The RF algorithm sparked considerable interest and benefited from a rich literature of methodological developments [163, 99, 12]. In particular, RF was adapted for various tasks including the computation of prediction intervals [385, 449, 63], online learning [252, 327] and quantile regression [308]. A theoretical framework was progressively built around random forests to study their statistical properties such as consistency [33, 32, 397], control over bias and variance [9, 161] and feature importance assessment [282, 230]. These works often consider purely random trees (splits are not optimized using training data) which are more amenable to theoretical analysis. More recent works introduced "honest" trees which are built using disjoint samples from those used for prediction [11]. Honest forests were considered and proven consistent for causal inference tasks by [438].

By essence, decision trees are universal approximators which lends them strong expressive power but also makes them prone to overfitting, hence the need for regularization. This was addressed in early works using pruning methods such as CCP, REP or MEP [370, 56]. Although they are fairly effective at reducing tree overfitting, these methods are mostly based on heuristics so that little is known about their theoretical properties. A form of soft-pruning was proposed by [62] and referred to as *tree smoothing*. The latter efficiently computes predictions as approximate Bayesian posteriors over the set of possible prunings, however, the associated complexity is of the order of the tree-size, which makes the computation of predictions slow. An alternative form of regularization called "shrinkage" was proposed in [180, 2] and consists in averaging predictions between leaf nodes and their ancestors. Although a parallel can be drawn between such methods and kernel Ridge regression, their theoretical support remains limited and does not allow to claim improved performance. Recent studies suggest that tree depth limitation and randomness of the tree building process bring an implicit regularization effect which improves performance on low signal-to-noise ratio datasets [309, 458].

In [327], an improvement of Mondrian Forests [252] is introduced for online learning, using subtrees aggregation with exponential weights, which is particularly convenient in the online learning setting. However, [327] considers only the online setting, with purely random trees, leading to poor performances compared with realistic decision trees.

Extreme boosting algorithms are another family of tree ensemble methods. XGBoost [87] provides an extremely popular scalable tree boosting system which has been widely adopted in industry. LightGBM [231] employed the "histogram strategy" for faster split finding, together with clever downsampling and features grouping algorithms in order to achieve high performance in reduced computation times. CatBoost [366] is another boosting library which pays particular attention to categorical features using target encoding, while addressing the potential bias issues associated to such an encoding.

In WW, we combine several ideas from the literature into a novel high performing RF algorithm. We use a subtrees aggregation mechanism similar to [327] for batch learning in a different way: we exploit the bootstrap, one of the key ingredients of RF, which provides in-the-bag and out-of-bag samples, to perform aggregation with exponential weights leading to considerably improved predictions. In addition, we employ the histogram strategy [231] in order to grow decision trees much more efficiently. Finally, we implement an improved split finding method along categorical features which makes for better predictions.

Limitations. Our implementation of WW is still evolving and is not yet at the level of maturity of state-of-the-art EGB libraries such as [87, 231, 366]. It does not outperform such strong baselines, but proposes an improvement of RF algorithms, and gives an interesting balance between performance and computational efficiency.

7.2 WildWood: a new Random Forest algorithm

We consider batch supervised learning, where data comes as a set of i.i.d training samples (x_i, y_i) for $i = 1, \dots, n$ with vectors of numerical or categorical features $x_i \in \mathcal{X} \subset \mathbb{R}^d$ and $y_i \in \mathcal{Y}$. Our aim is to design a RF predictor $\widehat{g}(\,\cdot\,; \mathbf{\Pi}) = \frac{1}{M}\sum_{m=1}^{M} \widehat{f}(\,\cdot\,; \Pi_m) : \mathcal{X} \to \widehat{\mathcal{Y}}$ computed from training samples, where $\widehat{\mathcal{Y}}$ is the prediction space. Such a RF computes the average of M randomized trees predictions $\widehat{f}(\,\cdot\,; \Pi_m)$ following the principle of bagging [51, 350], with $\mathbf{\Pi} = (\Pi_1, \dots, \Pi_M)$ where $\Pi_1, \dots, \Pi_M$ are i.i.d realizations of a random variable corresponding to bootstrap and feature subsampling (see Section 7.2.1 below). Each tree is trained independently of each other, in parallel. In what follows we describe only the construction of a single tree and omit from now on the dependence on $m = 1, \dots, M$.

Feature binning. The split finding strategy described in Section 7.2.2 below works on binned features. While this technique is of common practice in EGB libraries [87, 231, 367], we are not aware of an implementation of it for RF. Note that binning may affect predictive performance by making split thresholds less accurate. However, the experimental performance from EGB libraries (see also Section 7.4) suggests that this effect is minor. The input $n \times d$ matrix $\boldsymbol{X}$ of features is transformed into another same-size matrix of "binned" features denoted $\boldsymbol{X}^{\mathrm{bin}}$. To each input feature $j = 1, \dots, d$ is associated a set $B_j = \{1, \dots, b_j\}$ of bins, where $b_j \leqslant b_{\max}$ with $b_{\max}$ a hyperparameter corresponding to the maximum number of bins a feature can use (default is $b_{\max} = 256$ similarly to [231], so that a single byte can be used for entries of $\boldsymbol{X}^{\mathrm{bin}}$). When a feature is continuous, it is binned into $b_{\max}$ bins using inter-quantile intervals. If it is categorical, each modality is mapped to a bin whenever $b_{\max}$ is larger than its number of modalities, otherwise sparsest modalities end up binned together. If a feature j contains missing values, its rightmost bin in B_j is used to encode them (in such case, later in split, we do not loop only left to right (along bin order), but right to left as well, in order to compare splits that put missing values on the left or on the right). After binning, each column satisfies $\boldsymbol{X}^{\mathrm{bin}}_{\bullet,j} \in B_j^n$.

7.2.1 Random decision trees

Let $C = \prod_{j=1}^{d} B_j$ be the binned feature space. A random decision tree is a pair $(\mathcal{T}, \Sigma)$, where $\mathcal{T}$ is a finite ordered binary tree and Σ contains information about each node in $\mathcal{T}$, such as split information. The tree is random and its source of randomness Π comes from the bootstrap and feature subsampling as explained below.

Finite ordered binary trees. A finite ordered binary tree $\mathcal{T}$ is represented as a finite subset of the set $\{0,1\}^* = \bigcup_{n \geqslant 0} \{0,1\}^n$ of all finite words on $\{0,1\}$. The set $\{0,1\}^*$ is endowed with a tree structure (and called the complete binary tree): the empty word `root` is the root, and for any $\mathbf{v} \in \{0,1\}^*$, the left (resp. right) child of $\mathbf{v}$ is $\mathbf{v}0$ (resp. $\mathbf{v}1$). We denote by $\mathrm{intnodes}(\mathcal{T}) = \{\mathbf{v} \in \mathcal{T} : \mathbf{v}0, \mathbf{v}1 \in \mathcal{T}\}$ the set of its interior nodes and by $\mathrm{leaves}(\mathcal{T}) = \{\mathbf{v} \in \mathcal{T} : \mathbf{v}0, \mathbf{v}1 \notin \mathcal{T}\}$ the set of its leaves, both sets are disjoint and the set of all nodes is $\mathrm{nodes}(\mathcal{T}) = \mathrm{intnodes}(\mathcal{T}) \cup \mathrm{leaves}(\mathcal{T})$.

Splits and cells. The split $\sigma_{\mathbf{v}} = (j_{\mathbf{v}}, t_{\mathbf{v}}) \in \Sigma$ of each $\mathbf{v} \in \mathrm{intnodes}(\mathcal{T})$ is characterized by its dimension $j_{\mathbf{v}} \in \{1, \dots, d\}$ and a non-empty subset of bins $t_{\mathbf{v}} \subsetneq \{1, \dots, b_{j_{\mathbf{v}}}\}$. We associate to each $\mathbf{v} \in \mathcal{T}$ a cell $C_{\mathbf{v}} \subseteq C$ which is defined recursively: $C_{\mathtt{root}} = C$ and for each $\mathbf{v} \in \mathrm{intnodes}(\mathcal{T})$ we define

$$C_{\mathbf{v}0} := \{x \in C_{\mathbf{v}} : x_{j_{\mathbf{v}}} \in t_{\mathbf{v}}\} \quad \text{and} \quad C_{\mathbf{v}1} := C_{\mathbf{v}} \setminus C_{\mathbf{v}0}.$$

When $j_{\mathbf{v}}$ corresponds to a continuous feature, bins have a natural order so that $t_{\mathbf{v}}$ is encoded by a bin threshold $s_{\mathbf{v}} \in B_{j_{\mathbf{v}}}$ as $t_{\mathbf{v}} = \{1, 2, \dots, s_{\mathbf{v}}\}$; while for a categorical split, the set $t_{\mathbf{v}}$ may be any non-trivial subset of $B_{j_{\mathbf{v}}}$. By construction, $(C_{\mathbf{v}})_{\mathbf{v} \in \text{leaves}(\mathcal{T})}$ is a partition of C.

Bootstrap and feature subsampling. Let $I = \{1, \dots, n\}$ be the training samples indices. The randomization Π of the tree uses bootstrap: it samples uniformly at random, with replacement, elements of I corresponding to in-the-bag (`itb`) samples. If we denote as $I_{\texttt{itb}}$ the indices of unique `itb` samples, we can define the indices of out-of-bag (`oob`) samples as $I_{\texttt{oob}} = I \setminus I_{\texttt{itb}}$. A standard argument shows that $\mathbb{P}[i \in I_{\texttt{itb}}] = 1 - (1 - 1/n)^n \to 1 - e^{-1} \approx 0.632$ as $n \to +\infty$, known as the 0.632 rule [141]. The randomization Π uses also feature subsampling: each time we need to find a split, we do not try all the features $\{1, \dots, d\}$ but only a subset of them of size $d_{\max}$, chosen uniformly at random. This follows what standard RF algorithms do [52, 34, 281], with the default $d_{\max} = \sqrt{d}$.

7.2.2 Split finding on histograms

For K-class classification, when looking for a split for some node $\mathbf{v}$, we compute the node's "histogram" $\text{hist}_{\mathbf{v}}[j, b, k] = \sum_{i \in I_{\texttt{itb}} : x_i \in C_{\mathbf{v}}} \mathbf{1}_{x_{i,j}=b, y_i=k}$ for each sampled feature j, each bin b and label class k seen in the node's samples (actually weighted counts to handle bootstrapping and sample weights). Of course, one has $\text{hist}_{\mathbf{v}} = \text{hist}_{\mathbf{v}0} + \text{hist}_{\mathbf{v}1}$, so that we don't need to compute two histograms for siblings $\mathbf{v}0$ and $\mathbf{v}1$, but only a single one. Then, we loop over the set of non-constant (in the node) sampled features $\{j : \#\{b : \sum_k \text{hist}_{\mathbf{v}}[j, b, k] \geqslant 1\} \geqslant 2\}$ and over the set of non-empty bins $\{b : \sum_k \text{hist}_{\mathbf{v}}[j, b, k] \geqslant 1\}$ to find a split, by comparing standard impurity criteria computed on the histogram's statistics, such as gini or entropy for classification and variance for regression.

Bin order and categorical features. The order of the bins used in the loop depends on the type of the feature. If it is continuous, we use the natural order of bins. If it is categorical and the task is binary classification (labels in $\{0, 1\}$) we use the bin order that sorts $\text{hist}_{\mathbf{v}}[j, b, 1] / \sum_{k=0,1} \text{hist}_{\mathbf{v}}[j, b, k]$ with respect to b, namely the proportion of labels 1 in each bin. This allows to find the optimal split with complexity $O(b_j \log b_j)$, see Theorem 9.6 in [54], the logarithm coming from the sorting operation, while there are $2^{b_j - 1} - 1$ possible splits. This trick is used by EGB libraries as well, using an order of gradient/hessian statistics of the loss considered [87, 231, 367]. For K-class classification with $K > 2$, we consider two strategies: (1) one-versus-rest, where we train MK trees instead of M, each tree trained with a binary one-versus-rest label, so that trees can find optimal categorical splits and (2) heuristic, where we train M trees and where split finding uses K loops over bin orders that sort $\text{hist}_{\mathbf{v}}[j, b, k] / \sum_{k'} \text{hist}_{\mathbf{v}}[j, b, k']$ (w.r.t b) for $k = 0, \dots, K - 1$. The former strategy generally yields better predictions but at a higher computational cost, while the latter is more efficient but also more prone to overfitting.

Split requirements. Nodes must hold at least one `itb` and one `oob` sample to apply aggregation with exponential weights, see Section 7.2.3 below. A split is discarded if it leads to children with less than $n_{\text{min-leaf}}$ `itb` or `oob` samples and we do not split a node with less than $n_{\text{min-split}}$ `itb` or `oob` samples. These hyperparameters only weakly impact WW's performances and sticking to default values ($n_{\text{min-leaf}} = 1$ and $n_{\text{min-split}} = 2$, following `scikit-learn`'s [281, 355]) is usually enough (see Section 7.7.3 below).

Related works on categorical splits. In [97], an interesting characterization of an optimal categorical split for multiclass classification is introduced, but no efficient algorithm is, to the

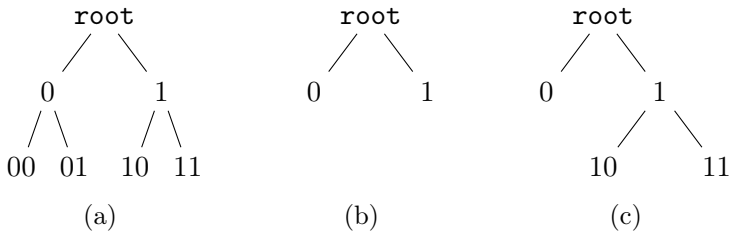

Figure 7.4: Example of a simple tree $\mathcal{T}$ show in (a) and two subtrees $T_1, T_2 \subset \mathcal{T}$ rooted at `root` shown in (b) and (c).

best of our understanding, available for it. A heuristic algorithm is proposed therein, but it requires to compute, for each split, the top principal component of the covariance matrix of the conditional distribution of labels given bins, which is computationally too demanding for an RF algorithm intended for large datasets. Regularized target encoding is shown in [351] to perform best when compared with many alternative categorical encoding methods. Catboost [367] uses target encoding, which replaces feature modalities by label statistics, so that a natural bin order can be used for split finding. To avoid overfitting on uninformative categorical features, a debiasing technique uses random permutations of samples and computes the target statistic of each element based only on its predecessors in the permutation. However, for multiclass classification, target encoding is influenced by the arbitrarily chosen ordinal encoding of the labels. LightGBM [231] uses a one-versus-rest strategy, which is also one of the approaches used in WW for categorical splits on multiclass tasks. For categorical splits, where bin order depends on labels statistics, WW does not use debiasing as in [367], since aggregation with exponential weights computed on `oob` samples allows to deal with overfitting.

Tree growth stopping. We do not split a node and make it a leaf if it contains less than $n_{\text{min-split}}$ `itb` or `oob` samples. The same applies when a node's impurity is not larger than a threshold ε ($\varepsilon = 0$ by default). When only leaves or non-splittable nodes remain, the growth of the tree is stopped. Trees grow in a depth-first fashion so that childs $\mathbf{v}0$ and $\mathbf{v}1$ have memory indexes larger than their parent $\mathbf{v}$ (as required by Algorithm 4 below). In practice, trees are grown to full depth in WildWood (i.e. we take $n_{\text{min-split}} = 2$) since we can only benefit from additional splits and do not need to worry about overfitting since it is prevented by the subtrees aggregation we present now.

7.2.3 Prediction function: aggregation with exponential weights

Given a tree $\mathcal{T}$ grown as described in Sections 7.2.1 and 7.2.2, its prediction function is an aggregation of the predictions given by all possible subtrees rooted at `root`, denoted $\{T : T \subset \mathcal{T}\}$. Figure 7.4 provides an example of a tree and two such subtrees. While $\mathcal{T}$ is grown using `itb` samples, we use `oob` samples to perform aggregation with exponential weights, with a branching process prior over subtrees. Thus, the aggregation mechanism gives more importance to shallow subtrees with a good predictive `oob` performance. The balance between shallowness and prediction quality is modulated through a temperature parameter denoted η in Equation (7.2.5) below.

Node and subtree prediction. We define $\mathbf{v}_T(x) \in \text{leaves}(T)$ as the leaf of T containing $x \in C$. The prediction of a node $\mathbf{v} \in \text{nodes}(\mathcal{T})$ and of a subtree $T \subset \mathcal{T}$ is given by

$$\widehat{y}_{\mathbf{v}} = h((y_i)_{i \in I_{\texttt{itb}} \,:\, x_i \in C_{\mathbf{v}}}) \quad \text{and} \quad \widehat{y}_T(x) = \widehat{y}_{\mathbf{v}_T(x)}, \tag{7.2.1}$$

where $h : \cup_{n \geqslant 0} \mathcal{Y}^n \to \widehat{\mathcal{Y}}$ is a generic "forecaster" used in each cell and where a subtree prediction is that of its leaf containing x. A standard choice for regression ($\mathcal{Y} = \widehat{\mathcal{Y}} = \mathbb{R}$) is the empirical mean forecaster

$$\widehat{y}_{\mathbf{v}} = \frac{1}{n_{\mathbf{v}}} \sum_{i \in I_{\texttt{itb}} \,:\, x_i \in C_{\mathbf{v}}} y_i, \tag{7.2.2}$$

where $n_{\mathbf{v}} = |\{i \in I_{\texttt{itb}} \;:\; x_i \in C_{\mathbf{v}}\}|$. For K-class classification with $\mathcal{Y} = \{1, \dots, K\}$ and $\widehat{\mathcal{Y}} = \mathcal{P}(\mathcal{Y})$, the set of probability distributions over $\mathcal{Y}$, a standard choice is a Bayes predictive posterior with a prior on $\mathcal{P}(\mathcal{Y})$ equal to the Dirichlet distribution $\mathsf{Dir}(\alpha, \dots, \alpha)$, namely the *Jeffreys prior* on the multinomial model $\mathcal{P}(\mathcal{Y})$, which leads to

$$\widehat{y}_{\mathbf{v}}(k) = \frac{n_{\mathbf{v}}(k) + \alpha}{n_{\mathbf{v}} + \alpha K}, \tag{7.2.3}$$

for any $k \in \mathcal{Y}$, where $n_{\mathbf{v}}(k) = |\{i \in I_{\texttt{itb}} : x_i \in C_{\mathbf{v}}, y_i = k\}|$. By default, WW uses $\alpha = 1/2$ (the *Krichevsky-Trofimov* forecaster [415]), but one can perfectly use any $\alpha > 0$, so that all the coordinates of $\widehat{y}_{\mathbf{v}}$ are positive. This is motivated by the fact that WW uses as default the log loss to assess `oob` performance for classification, which requires an arbitrarily chosen clipping value for zero probabilities. Different choices of α only weakly impact WW's performance, as illustrated in Section 7.7.3. We use `oob` samples to define the cumulative losses of the predictions of all $T \subset \mathcal{T}$

$$L_T = \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{y}_T(x_i), y_i), \tag{7.2.4}$$

where $\ell : \widehat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}^+$ is a loss function. For regression problems, a default choice is the quadratic loss $\ell(\widehat{y}, y) = (\widehat{y} - y)^2$ while for multiclass classification, a default is the log-loss $\ell(\widehat{y}, y) = -\log \widehat{y}(y)$, where $\widehat{y}(y) \in (0, 1]$ when using (7.2.3), but other loss choices are of course possible.

Prediction function. Let $x \in C$. The prediction function $\widehat{f}$ of a tree $\mathcal{T}$ in WW is given by

$$\widehat{f}(x) = \frac{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T} \widehat{y}_T(x)}{\sum_{T \subset \mathcal{T}} \pi(T) e^{-\eta L_T}} \quad \text{with} \quad \pi(T) = 2^{-\|T\|}, \tag{7.2.5}$$

where the sum is over all subtrees T of $\mathcal{T}$ rooted at `root`, where $\eta > 0$ is a temperature parameter and $\|T\|$ is the number of nodes in T minus its number of leaves that are also leaves of $\mathcal{T}$. A default choice for the temperature is $\eta = 1$ for the log-loss (see in particular Corollary 7.1 in Section 7.3 below), but it can also be tuned through hyperoptimization, although we do not observe strong performance gains, see Section 7.7.3 below.

The prediction function (7.2.5) corresponds to an exponentially weighted average over the predictors $\widehat{y}_T(x)$ for $T \subset \mathcal{T}$ rooted at `root` based on `oob` performance and the branching process prior π with branching probability 1/2. The latter being expressly chosen for its fitness within this paradigm. This definition falls within the framework of PAC-Bayesian theory [304, 303, 74] with theoretical guarantee of the oracle inequality for $\widehat{f}$ on `oob` (see Theorem 7.2 below). This aggregation procedure can be understood as a *non-greedy way to prune trees*: the weights depend not only on the quality of one single split but also on the performance of each subsequent split.

Naively computing $\widehat{f}$ from Equation (7.2.5) is computationally and memory-wise infeasible for a large $\mathcal{T}$, since it involves a sum over all $T \subset \mathcal{T}$ rooted at `root` and requires one weight for each T. Indeed, the number of subtrees of a minimal tree that separates n points is exponential in the number of nodes, and hence *exponential in* n. However, it turns out that one can compute $\widehat{f}$ exactly and very efficiently thanks to the prior choice π together with an adaptation of *context*

tree weighting [441, 440, 187, 75].

Theorem 7.1. *The prediction function* (7.2.5) *can be written as* $\widehat{f}(x) = \widehat{f}_{\texttt{root}}(x)$*, where* $\widehat{f}_{\texttt{root}}(x)$ *satisfies the recursion*

$$\widehat{f}_{\mathbf{v}}(x) = \frac{1}{2}\frac{w_{\mathbf{v}}}{w^{\mathrm{den}}_{\mathbf{v}}}\widehat{y}_{\mathbf{v}} + \Big(1 - \frac{1}{2}\frac{w_{\mathbf{v}}}{w^{\mathrm{den}}_{\mathbf{v}}}\Big)\widehat{f}_{\mathbf{v}a}(x) \tag{7.2.6}$$

for $\mathbf{v}, \mathbf{v}a \in \texttt{path}(x)$ $(a \in \{0,1\})$ *the path in* $\mathcal{T}$ *going from* `root` *to* $\mathbf{v}_{\mathcal{T}}(x)$*, where* $w_{\mathbf{v}} := \exp(-\eta L_{\mathbf{v}})$ *with* $L_{\mathbf{v}} := \sum_{i \in I_{\texttt{oob}} : x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_{\mathbf{v}}, y_i)$ *and where* $w^{\mathrm{den}}_{\mathbf{v}}$ *are weights satisfying the recursion*

$$w^{\mathrm{den}}_{\mathbf{v}} = \begin{cases} w_{\mathbf{v}} & \text{if } \mathbf{v} \in \mathrm{leaves}(\mathcal{T}), \\ \frac{1}{2}w_{\mathbf{v}} + \frac{1}{2}w^{\mathrm{den}}_{\mathbf{v}0}w^{\mathrm{den}}_{\mathbf{v}1} & \text{otherwise.} \end{cases} \tag{7.2.7}$$

The proof of Theorem 7.1 is given in Section 7.6.1 below, a consequence of this Theorem being a very efficient method to compute $\widehat{f}(x)$ described in Algorithms 4 and 5 below.

Algorithm 4 computes the weights $w^{\mathrm{den}}_{\mathbf{v}}$ using the fact that trees in WW are grown in a depth-first fashion, so that we can loop *once*, leading to a $O(|\mathrm{nodes}(\mathcal{T})|)$ complexity in time and in memory usage, over nodes from a data structure that respects the parenthood order. Direct computations can lead to numerical over- or under-flows (many products of exponentially small or large numbers are involved), so Algorithm 4 works recursively over the logarithms of the weights (line 6 uses a log-sum-exp function that can be made overflow-proof).

Algorithm 4 Computation of $\log(w^{\mathrm{den}}_{\mathbf{v}})$ for all $\mathbf{v} \in \mathrm{nodes}(\mathcal{T})$.

1: **Inputs:** $\mathcal{T}$, $\eta > 0$ and losses $L_{\mathbf{v}}$ for all $\mathbf{v} \in \mathrm{nodes}(\mathcal{T})$. Nodes from $\mathrm{nodes}(\mathcal{T})$ are stored in a data structure `nodes` that respects parenthood order: for any $\mathbf{v} = \texttt{nodes}[i_{\mathbf{v}}] \in \mathrm{intnodes}(\mathcal{T})$ and children $\mathbf{v}a = \texttt{nodes}[i_{\mathbf{v}a}]$ for $a \in \{0,1\}$, we have $i_{\mathbf{v}a} > i_{\mathbf{v}}$.
2: **for** $\mathbf{v} \in \mathrm{reversed}(\texttt{nodes})$ **do**
3: **if** $\mathbf{v}$ is a leaf **then**
4: Put $\log(w^{\mathrm{den}}_{\mathbf{v}}) \leftarrow -\eta L_{\mathbf{v}}$
5: **else**
6: Put $\log(w^{\mathrm{den}}_{\mathbf{v}}) \leftarrow \log\big(\frac{1}{2}e^{-\eta L_{\mathbf{v}}} + \frac{1}{2}e^{\log(w^{\mathrm{den}}_{\mathbf{v}0}) + \log(w^{\mathrm{den}}_{\mathbf{v}1})}\big)$
7: **end if**
8: **end for**
9: **return** The set of log-weights $\{\log(w^{\mathrm{den}}_{\mathbf{v}}) : \mathbf{v} \in \mathrm{nodes}(\mathcal{T})\}$

Algorithm 4 is applied once $\mathcal{T}$ is fully grown, so that WW is ready to produce predictions using Algorithm 5 below. Note that hyperoptimization of η or α, if required, does not need to grow $\mathcal{T}$ again, but only to update $w^{\mathrm{den}}_{\mathbf{v}}$ for all $\mathbf{v} \in \mathrm{nodes}(\mathcal{T})$ with Algorithm 4, making hyperoptimization of these parameters particularly efficient.

Algorithm 5 Computation of $\widehat{f}(x)$ for any $x \in C$.

1: **Inputs:** Tree $\mathcal{T}$, losses $L_{\mathbf{v}}$ and log-weights $\log(w^{\mathrm{den}}_{\mathbf{v}})$ computed by Algorithm 4
2: Find $\mathbf{v}_{\mathcal{T}}(x) \in \mathrm{leaves}(\mathcal{T})$ (the leaf containing x) and put $\mathbf{v} \leftarrow \mathbf{v}_{\mathcal{T}}(x)$
3: Put $\widehat{f}(x) \leftarrow \widehat{y}_{\mathbf{v}}$ (the node $\mathbf{v}$ forecaster, such as (7.2.2) for regression or (7.2.3) for classification)
4: **while** $\mathbf{v} \neq \texttt{root}$ **do**
5: Put $\mathbf{v} \leftarrow \mathrm{parent}(\mathbf{v})$
6: Put $\alpha \leftarrow \frac{1}{2}\exp(-\eta L_{\mathbf{v}} - \log(w^{\mathrm{den}}_{\mathbf{v}}))$
7: Put $\widehat{f}(x) \leftarrow \alpha\widehat{y}_{\mathbf{v}} + (1-\alpha)\widehat{f}(x)$
8: **end while**
9: **return** The prediction $\widehat{f}(x)$

The recursion used in Algorithm 5 has a complexity $O(|\texttt{path}(x)|)$ which is the complexity required to find the leaf $\mathbf{v}_{\mathcal{T}}(x)$ containing $x \in C$: Algorithm 5 *only increases by a factor* 2 the prediction complexity of a standard RF (in order to go down to $\mathbf{v}_{\mathcal{T}}(x)$ and up again to `root` along $\texttt{path}(x)$). More details about the construction of Algorithms 4 and 5 can be found in Section 7.6.1 below.

7.3 Theoretical guarantees

This section proposes some theoretical guarantees on the subtrees aggregation (7.2.5) used in WW. We say that a loss function ℓ is η-exp-concave for some $\eta > 0$ whenever $z \mapsto \exp(-\eta\ell(z, y))$ is concave for any $y \in \mathcal{Y}$. We consider a fully-grown tree $\mathcal{T}$ computed using `itb` samples and the set of `oob` samples $(x_i, y_i)_{i \in I_{\texttt{oob}}}$ on which L_T is computed using (7.2.4), and we denote $n_{\texttt{oob}} := |I_{\texttt{oob}}|$.

Theorem 7.2 (Oracle inequality)**.** *Assume that the loss function ℓ is η-exp-concave. Then, the prediction function $\widehat{f}$ given by* (7.2.5) *satisfies the oracle inequality*

$$\frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{f}(x_i), y_i) \leqslant \inf_{T \subset \mathcal{T}} \left\{ \frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{y}_T(x_i), y_i) + \frac{\log 2}{\eta} \frac{\|T\|}{n_{\texttt{oob}} + 1} \right\},$$

where the infimum is over any subtree $T \subset \mathcal{T}$ and where we recall that $\|T\|$ is the number of nodes in T minus its number of leaves that are also leaves of $\mathcal{T}$.

Theorem 7.2 proves that, for a general loss function, the prediction function of WW is able to perform nearly as well as the best *oracle* subtree $T \subset \mathcal{T}$ on `oob` samples, with a $O(\|T\|/n_{\texttt{oob}})$ rate which is optimal for model-selection oracle inequalities [419] ($\|T\| = O(\log N_{\mathcal{T}})$ with a number of "experts" $N_{\mathcal{T}} = |\{T : T \subset \mathcal{T}\}|$ for a well-balanced $\mathcal{T}$). Let us stress again that, while finding an oracle $\arg\min_{T \subset \mathcal{T}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{y}_T(x_i), y_i)$ is computationally infeasible, since it requires to try out all possible subtrees, WW's prediction function (7.2.5) comes at a cost comparable to that of a standard Random Forest, as explained in Section 7.2.3 above.

The proof of Theorem 7.2 is given in Section 7.6.2 below and relies on techniques from PAC-Baysesian theory [304, 303, 74]. The arguments hinge upon the mixability of the loss, a consequence of its exp-concavity, as well as the Donsker-Varadhan variational formula. Note that the statement concerns only the performance of a single tree and not the whole forest. Indeed, extrapolating the benefit brought by the exponential weights' aggregation within each tree to the forest itself remains a difficult open problem even though it is intuitive enough that improved individual trees should enhance the overall performance. Compared with [327] about online learning, our proof differs in that we do not use results specialized to online learning such as [435] nor online-to-batch conversion [78]. Note that Theorem 7.2 does not address the generalization error, since it would require to study the generalization error of the random forest itself (and of the fully grown tree $\mathcal{T}$), which is a topic way beyond the scope of this paper, and still a very difficult open problem: recent results [161, 9, 397, 395, 396, 328] only study stylized versions of RF (called purely random forests).

Consequences of Theorem 7.2 are Corollary 7.1 for the log-loss (classification) and Corollary 7.2 for the least-squares loss (regression).

Corollary 7.1 (Classification)**.** *Consider K-class classification ($\mathcal{Y} = \{1, \dots, K\}$) and consider the prediction function $\widehat{f}$ given by* (7.2.5)*, where node predictions are given by* (7.2.3) *with $\alpha =$*

1/2 (*WW's default*), *where* ℓ *is the log-loss and where* $\eta = 1$. *Then, we have*

$$\frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{f}(x_i), y_i) \leqslant \inf_{T \subset \mathcal{T}} \left\{ \frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(g_T(x_i), y_i) + \frac{K + 4\log 2 - 1}{4} \frac{\|T\| + 1}{n_{\texttt{oob}}} \right\},$$

where g_T *is any constant function on the leaves of* T.

Corollary 7.2 (Regression)**.** *Consider regression with* $\mathcal{Y} = [-B, B]$ *for some* $B > 0$ *and the prediction function* $\widehat{f}$ *given by* (7.2.5)*, where node predictions are given by* (7.2.2)*, where* ℓ *is the least-squares loss and where* $\eta = 1/(8B^2)$. *Then, we have*

$$\frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{f}(x_i), y_i) \leqslant \inf_{T \subset \mathcal{T}} \left\{ \frac{1}{n_{\texttt{oob}}} \sum_{i \in I_{\texttt{oob}}} \ell(g_T(x_i), y_i) + 8(\log 2) B^2 \frac{\|T\|}{n_{\texttt{oob}}} \right\},$$

where g_T *is any function constant on the leaves of* T.

The proofs of Corollaries 7.1 and 7.2 are given in Section 7.6.2. These corollaries motivate the default hyperparameter values of η, in particular $\eta = 1$ for classification.

7.4 Experiments

Our implementation of WildWood is available at the `GitHub` repository `https://github.com/pyensemble/wildwood.git` under the `BSD3-Clause` license on `GitHub` and available through `PyPi`. It is a `Python` package that follows `scikit-learn`'s API conventions, that is JIT-compiled to machine code using `numba` [254]. Trees in the forest are grown in parallel using `joblib` [220] and CPU threads, GPU training will be supported in future updates.

Baselines. We compare WildWood (denoted `WW`n for n trees) with several strong baselines including :

- `RF`n: `scikit-learn`'s implementation of Random Forest [355, 281] using n trees;
- an elementary random forest algorithm we refer to as `RD`n using n trees, each grown with a randomized depth limit (uniform from 3 to 50), this is implemented using WildWood by deactivating aggregation and categorical feature support;
- `HGB`: a histogram-based implementation of extreme gradient boosting (inspired by LightGBM) from `scikit-learn`;
- several state-of-the-art and widely adopted extreme gradient boosting libraries including `XGB`: XGBoost [87]; `LGBM`: LightGBM [231] and `CB`: CatBoost [367, 134].

Note that we focus on computationally competitive algorithms which excludes MCMC based methods such as BART [94].

Computational resources. We used a 32-cores server with two Intel Xeon Gold CPUs, two Tesla V100 GPUs and 384GB RAM for the experiments involving hyperoptimization (Table 7.1) and used a 12-cores Intel i7 MacBook Pro with 32GB RAM and no GPU to obtain training times achievable by a "standard user" (Table 7.2). All experiments can be reproduced using `Python` scripts on the repository.

Data. We use publicly available and open-source datasets from the UCI repository [136], including small datasets (hundreds of rows) and large datasets (millions of rows), their main characteristics are given in Table 7.6 together with URLs in Table 7.7, see Section 7.7.2 below. Each dataset is randomly split into a training set (70%) and a test set (30%). We specify which features are categorical to algorithms that natively support it (`HGB`, `LGBM`, `CB` and `WW`n) and simply integer-encode them, while we use one-hot encoding for other algorithms (`RF`n, `RD`n, `XGB`).

7.4.1 Performance on classification tasks

For each dataset and algorithm, we evaluate the performance after tuning the hyperparameters. Hyperoptimization is performed as follows: from the training set, we use 4/5 for training and 1/5 for validation and do 50 steps of sequential optimization using the Tree Parzen Estimator implemented in the `hyperopt` library [30]. More details about hyperoptimization are provided in Section 7.7.1 below. Then, we refit on the whole training set with the best hyperparameters and report scores on the test set. This is performed 5 times in order to report standard deviations. We use the area under the ROC curve (AUC), for K-class datasets with $K > 2$ we average the AUC of each class versus the rest. This leads to the test AUC scores displayed in Table 7.1 (the same scores with standard deviations are available in Table 7.3).

Table 7.1: Test AUC of all algorithms after hyperoptimization on the considered datasets. Standard-deviations are reported in Table 7.3 and results in terms of log-loss in Table 7.4. We observe that WW has better (or identical in some cases) performances than RF on all datasets and that it is close to that of EGB libraries (bold is for best EGB performance, underline for best `RF`n, `WW`n or `RD`n performance).

	XGB	LGBM	CB	HGB	RF10	RF100	WW10	WW100	RD10	RD100
adult	0.930	**0.931**	0.927	0.930	0.915	0.918	0.916	0.919	0.915	0.917
bank	0.933	**0.935**	0.925	0.930	0.919	0.929	0.926	0.931	0.919	0.922
breastcancer	0.991	0.993	0.987	**0.994**	0.987	0.989	0.992	0.992	0.987	0.985
car	0.999	**1.000**	**1.000**	**1.000**	0.997	0.998	0.998	0.998	0.993	0.993
covtype	**0.999**	**0.999**	0.998	**0.999**	0.997	0.998	0.996	0.998	0.974	0.996
default-cb	0.780	**0.783**	0.780	0.779	0.765	0.775	0.774	0.778	0.772	0.773
higgs	0.853	**0.857**	0.847	0.853	0.820	0.837	0.820	0.837	0.813	0.815
internet	0.934	0.910	**0.938**	0.911	0.917	0.935	0.926	0.928	0.925	0.928
kddcup	**1.000**	**1.000**	**1.000**	**1.000**	0.997	0.998	1.000	1.000	1.000	1.000
kick	**0.777**	0.770	**0.777**	0.771	0.749	0.764	0.756	0.763	0.752	0.754
letter	**1.000**	**1.000**	**1.000**	**1.000**	0.997	0.999	0.997	0.999	0.995	0.993
satimage	**0.991**	**0.991**	**0.991**	0.987	0.985	0.991	0.986	0.991	0.984	0.983
sensorless	**1.000**	**1.000**	**1.000**	**1.000**	1.000	1.000	1.000	1.000	0.999	1.000
spambase	**0.990**	**0.990**	0.987	0.986	0.980	0.987	0.983	0.987	0.972	0.974

We observe in Table 7.1 that EGB algorithms, when hyperoptimized, lead to the best performances over the considered datasets compared with RF algorithms, and we observe that WW always improves the performance of RF, at the exception of few datasets for which the performance is similar. As for the randomized depth random forests `RD`n, we observe that their performance is generally inferior to that of WW and RF. This shows that the benefit brought by the weighted subtree aggregation mechanism cannot be simulated by averaging over trees of diverse depth.

When using default hyperparameters for all algorithms, we observe in Table 7.2 that the test AUC scores can decrease significantly for EGB libraries while RF algorithms seem more stable, and that there is no clear best performing algorithm in this case.

Table 7.2: Training times (seconds) of all algorithms with their default hyperparameters (no hyperoptimization) on the 5 largest considered datasets and test AUC corresponding to these training times. Test AUC scores are worse than that of Table 7.1, since no hyperoptimization is used. WW, which uses only 10 trees here (default number of trees), is generally among the fastest algorithms, for performances comparable to that of all baselines (bold is for best EGB training time or performance, underline for best RFn, WWn or RDn training time or performance). Standard deviations are reported in Table 7.5.

	Training time (seconds)						Test AUC					
	XGB	LGBM	CB	HGB	RF	WW	XGB	LGBM	CB	HGB	RF	WW
covtype	10	**3**	120	14	21	3	0.986	0.978	**0.989**	0.960	0.998	0.979
higgs	36	**30**	653	85	1389	179	0.823	0.812	**0.840**	0.812	0.838	0.813
internet	9	**4**	188	8	0.4	0.3	**0.918**	0.828	0.910	0.500	0.862	0.889
kddcup	175	41	2193	**31**	208	12	**1.000**	0.638	0.988	0.740	0.998	1.000
kick	7	**0.4**	50	0.7	31	5	0.768	0.757	**0.781**	0.773	0.747	0.751

7.4.2 Training time

We provide a study of training times over an increasing fraction of the same datasets in Figure 7.5. We report also in Table 7.2 (see also Table 7.5 for standard deviations) the test AUC scores obtained with default hyperparameters of all algorithms on the 5 largest considered datasets together with their training times (timings can vary by several orders of magnitude with varying hyperparameters for EGB libraries, as observed by the timing differences between Figure 7.3 and Table 7.2). Figure 7.5 further highlights the swiftness of WildWood's training procedure as compared to other models when trained on increasing fractions of the same dataset. Paradoxically, certain boosting algorithms appear to train faster on bigger data fractions, we attribute this to their stopping criteria on default parameters.

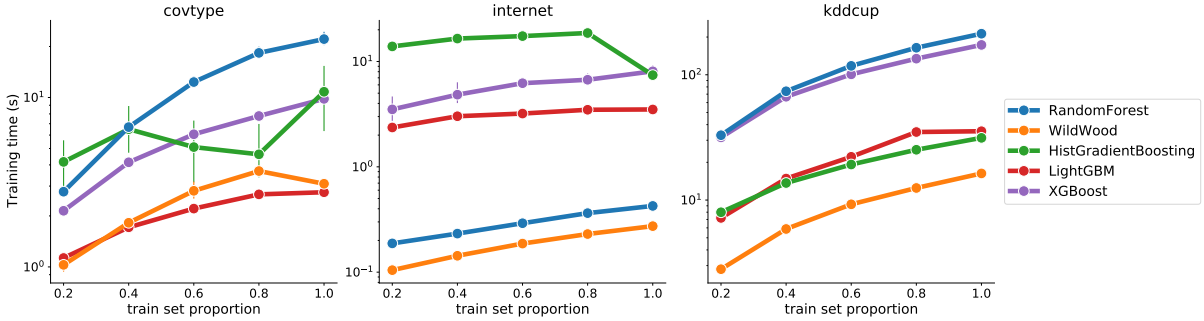

Figure 7.5: Training times on increasing fractions of a few large datasets for WildWood and the other baselines of Table 7.2 (except for CatBoost which is not competitive). The results are averaged over 5 runs using default parameters for each algorithm. WildWood clearly outpaces its competitors in almost all cases.

The results on both Tables 7.1 and 7.2 show that WW is competitive with respect to all baselines both in terms of performance and computational times: it manages to always reach at least comparable performance with the best algorithms despite only using 10 trees as a default. In this respect, WW maintains high scores at a lower computational cost.

7.4.3 Model size

We also train random forest models with number of trees from one to ten (and default parameters for the rest) on a few datasets using random forest algorithms (WW and `scikit-learn`'s implementations of RF [281] and Extra Trees [163]). We plot the mean test AUC over 10 repetitions against model size measured through the associated pickle file in megabytes. The result is displayed on Figure 7.6 and demonstrates WildWood's ability to offer lightweight random forest models with improved performance. This is a valuable advantage for applications using learning algorithms with limited memory and computational resources. Notable examples include embedded systems [214, 234, 213] or IoT devices [444, 126, 373].

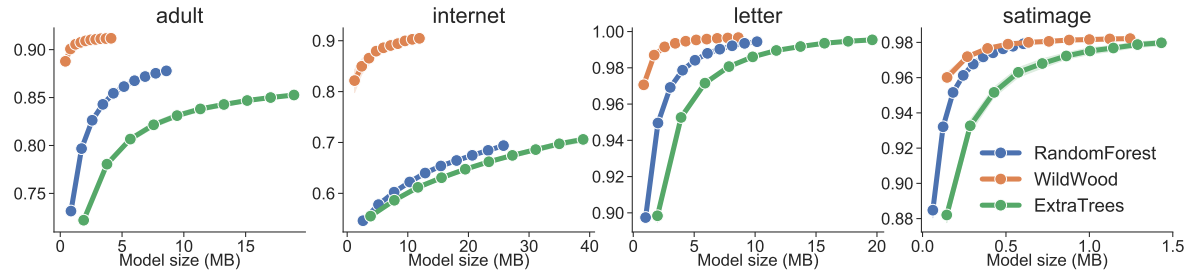

Figure 7.6: Mean test AUC and standard-deviations (y-axis) as a function of model size in megabytes (x-axis) using 10 train/test splits for WW and `scikit-learn`'s implementations of RF [281] and Extra Trees [163] on a few datasets. We use one to ten trees for each algorithm with default parameters. Wildwood is able to achieve better performance with smaller random forest models.

7.4.4 Regression experiment

We run an elementary regression experiment comparing WW to RF and Extra Trees again. The task is to recover four noisy test signals: *Doppler*, *Heavisine*, *Blocks* and *Bumps* originally analyzed by [128, 130]. We test different noise intensities measured through the signal-to-noise ratio (SNR) and show the results on Figure 7.7. We observe that WildWood clearly outperforms its competitors on the signal recovery tasks at low SNR levels. This is thanks to the regularizing effect of aggregation making the algorithm more robust to noise. We note, nonetheless, that this may slightly degrade performance on relatively irregular signals at high SNR (*Bumps*).

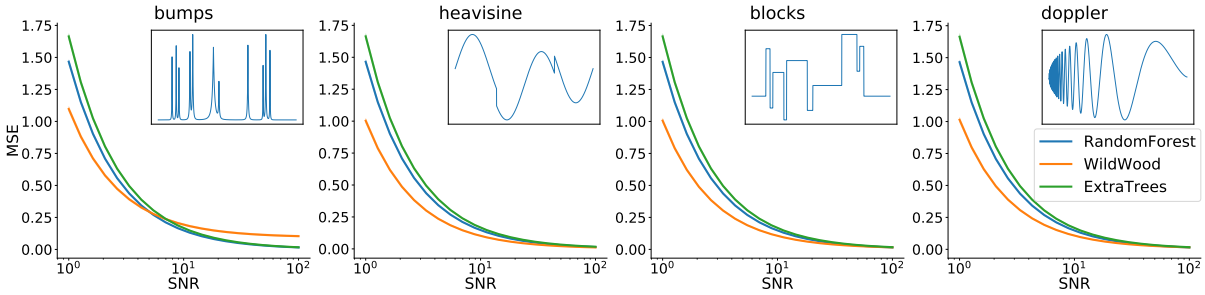

Figure 7.7: Averaged mean squared errors (y-axis) at increasing SNR levels (x-axis) over 10 repetitions of the noisy signal recovery task for WW and `scikit-learn`'s implementations of RF [281] and Extra Trees [163] on the four test signals *Doppler*, *Heavisine*, *Blocks* and *Bumps*. The noiseless signals are displayed in the inner frames. Each algorithm uses 100 trees and default values of the remaining parameters. Wildwood achieves smaller errors at low SNR levels thanks to its regularization effect.

7.4.5 Decision Boundaries

Finally, we provide an additional illustration of the interpretability advantage of WW over RF similar to Figure 7.1 but with real data. For this purpose, we plot the decision boundaries obtained on simple binary classification datasets for which good performance can be attained using only a pair of features. In each dataset, we choose such a pair among the most important features as measured by the Mean Decrease in Impurity (MDI) and fit WW and RF on the datasets restricted to these features. The results are shown on Figure 7.8. As before, we observe that WW yields more regular decision boundaries making its model predictions less prone to overfitting and more interpretable.

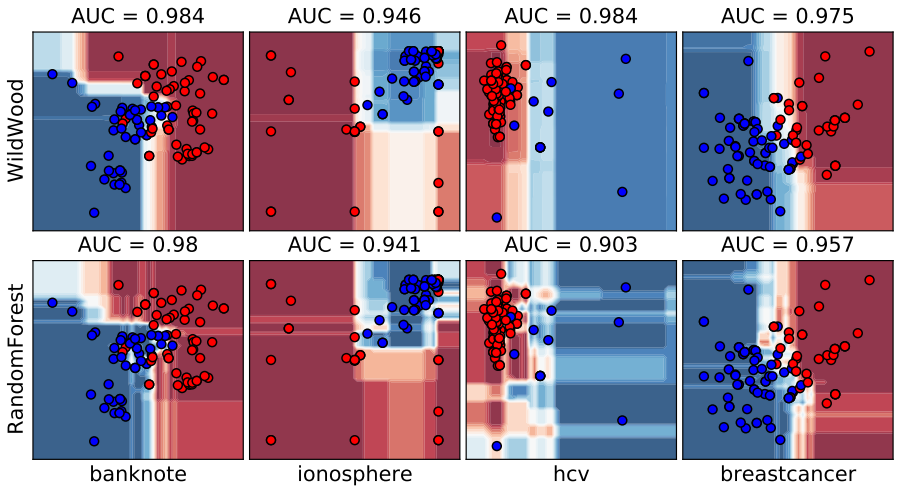

Figure 7.8: Illustration of the decision functions obtained with WW and RF using 10 trees each on simple binary classification datasets restricted to a pair of the most important features (according to MDI). The scattered samples represent the test set (colors indicate labels) while the decision boundaries use the train set. Thanks to aggregation, WW obtains more regular decision boundaries decreasing the risk of overfitting and improving interpretability..

7.5 Conclusion

We introduced WildWood, a new Random Forest algorithm for batch supervised learning. Tree predictions in WildWood are aggregation with exponential weights of the predictions of all subtrees, with weights computed on bootstrap out-of-bag samples. This leads to improved predictions in each individual tree, at a small computational cost, since WildWood's prediction complexity is similar to that of a standard Random Forest. Moreover, thanks to the histogram strategy, WildWood's implementation is competitive with strong baselines including popular extreme boosting libraries, both in terms of performance and training times. Note also that WildWood has few hyperparameters to tune and that the performances obtained with default hyperparameters are usually good enough in our experiments.

WildWood's implementation is still evolving and many improvements coming with future updates are planned, including the computation of feature importance, GPU training, distributed training (we only support single-machine training for now), among other enhancements that will further improve performances and accelerate computations. Room for improvement in WildWood comes from the fact that the overall forest prediction is a simple arithmetic mean of each tree prediction, while we could perform also exponentially weighted aggregation between trees. Future works include a WildWood-based implementation of isolation-forest [272], using the same subtrees aggregation mechanism with the log loss for density estimation, to propose a new algorithm for outliers detection.

7.6 Proofs

7.6.1 Proof of Theorem 7.1 and construction of Algorithms 4 and 5

The expression in Equation (7.2.5) involves sums over all subtrees T of the fully grown tree $\mathcal{T}$ (involving an exponential in the number of leaves of $\mathcal{T}$). However, it can be computed efficiently because of the specific choice of the prior π. More precisely, we will use the following lemma [187, Lemma 1] several times to efficiently compute sums of products. Let us recall that nodes($\mathcal{T}$) stands for the set of nodes of $\mathcal{T}$.

Lemma 7.1. *Let $g : \text{nodes}(\mathcal{T}) \to \mathbb{R}$ be an arbitrary function and define $G : \text{nodes}(\mathcal{T}) \to \mathbb{R}$ as*

$$G(\mathbf{v}) = \sum_{T \subset \mathcal{T}_\mathbf{v}} 2^{-\|T\|} \prod_{\mathbf{v}' \in \text{leaves}(T)} g(\mathbf{v}'), \tag{7.6.1}$$

where the sum over $T \subset \mathcal{T}_\mathbf{v}$ means the sum over all subtrees T of $\mathcal{T}$ rooted at $\mathbf{v}$. Then, $G(\mathbf{v})$ can be computed recursively as follows:

$$G(\mathbf{v}) = \begin{cases} g(\mathbf{v}) & \text{if } \mathbf{v} \in \text{leaves}(\mathcal{T}) \\ \frac{1}{2} g(\mathbf{v}) + \frac{1}{2} G(\mathbf{v}0) G(\mathbf{v}1) & \text{otherwise,} \end{cases}$$

for each node $\mathbf{v} \in \text{nodes}(\mathcal{T})$.

For the sake of completeness, we include a proof of this statement.

Proof. First, let us notice that the case $\mathbf{v} \in \text{leaves}(\mathcal{T})$ is straightforward since there is only one pruning T of $\mathcal{T}_\mathbf{v}$ which satisfies $\|T\| = 0$ (recall that $\|T\|$ is the number of internal nodes and leaves in T *minus* the number of leaves in T that are also leaves of $\mathcal{T}_\mathbf{v}$). For the second case, we can expand $G(\mathbf{v})$ by taking into account the pruning which only leaves $\mathbf{v}$ as a leaf, the rest of the prunings can be expressed through pairs of prunings T_0 and T_1 of $\mathcal{T}_{\mathbf{v}0}$ and $\mathcal{T}_{\mathbf{v}1}$ respectively. Moreover, it can be shown that such a pruning T satisfies $\|T\| = 1 + \|T_0\| + \|T_1\|$, thus we get the following expansion :

$$\begin{aligned} G(\mathbf{v}) &= \frac{1}{2} g(\mathbf{v}) + \sum_{T_0 \subset \mathcal{T}_{\mathbf{v}0}} \sum_{T_1 \subset \mathcal{T}_{\mathbf{v}1}} 2^{-(1+\|T_0\|+\|T_1\|)} \prod_{\mathbf{v}' \in T_0} g(\mathbf{v}0\mathbf{v}') \prod_{\mathbf{v}'' \in T_1} g(\mathbf{v}1\mathbf{v}'') \\ &= \frac{1}{2} g(\mathbf{v}) + \frac{1}{2} \Big(\sum_{T_0 \subset \mathcal{T}_{\mathbf{v}0}} 2^{-\|T_0\|} \prod_{\mathbf{v}' \in T_0} g(\mathbf{v}0\mathbf{v}') \Big) \cdot \Big(\sum_{T_1 \subset \mathcal{T}_{\mathbf{v}1}} 2^{-\|T_1\|} \prod_{\mathbf{v}'' \in T_1} g(\mathbf{v}1\mathbf{v}'') \Big) \\ &= \frac{1}{2} g(\mathbf{v}) + \frac{1}{2} G(\mathbf{v}0) G(\mathbf{v}1). \end{aligned}$$

This concludes the proof of Lemma 7.1. □

Let us introduce $w_T = \pi(T) \exp(-\eta L_T)$ for any $T \subset \mathcal{T}$, so that Equation (7.2.5) writes

$$\widehat{f}(x) = \frac{\sum_{T \subset \mathcal{T}} w_T \widehat{y}_T(x)}{\sum_{T \subset \mathcal{T}} w_T}, \tag{7.6.2}$$

where the sums hold over all the subtrees T of $\mathcal{T}$ rooted at `root` (the root of the full tree $\mathcal{T}$). We will show how to efficiently compute and update the numerator and denominator in Equation (7.6.2). Note that w_T may be written as

$$w_T = \pi(T) \exp(-\eta L_T)$$

$$
\begin{aligned}
&= 2^{-\|T\|} \exp\left(-\eta \sum_{i \in I_{\texttt{oob}}} \ell(\widehat{y}_{\mathbf{v}_T(x_i)}, y_i) \right) \\
&= 2^{-\|T\|} \exp\left(-\eta \sum_{\mathbf{v} \in \text{leaves}(T)} \sum_{i \in I_{\texttt{oob}} \,:\, x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_{\mathbf{v}_T(x_i)}, y_i) \right) \qquad (7.6.3) \\
&= 2^{-\|T\|} \exp\left(-\eta \sum_{\mathbf{v} \in \text{leaves}(T)} \sum_{i \in I_{\texttt{oob}} \,:\, x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_{\mathbf{v}}, y_i) \right) \qquad (7.6.4) \\
&= 2^{-\|T\|} \exp\left(-\eta \sum_{\mathbf{v} \in \text{leaves}(T)} L_{\mathbf{v}} \right) \\
&= 2^{-\|T\|} \prod_{\mathbf{v} \in \text{leaves}(T)} w_{\mathbf{v}}, \qquad (7.6.5)
\end{aligned}
$$

where we recall that

$$
L_{\mathbf{v}} = \sum_{i \in I_{\texttt{oob}} \,:\, x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_{\mathbf{v}}, y_i) \quad \text{and} \quad w_{\mathbf{v}} = \exp(-\eta L_{\mathbf{v}}).
$$

Equality (7.6.3) comes from the fact that the set of cells $\{C_{\mathbf{v}} : \mathbf{v} \in \text{leaves}(T)\}$ is a partition of C by construction, and that the stopping criterion used to build $\mathcal{T}$ ensures that each leaf node in leaves(T) contains at least one sample from $I_{\texttt{oob}}$ (see Section 7.2.2). Equality (7.6.4) comes from the fact that the prediction of a node is constant and equal to $\widehat{y}_{\mathbf{v}}$ for any $x \in C_{\mathbf{v}}$.

Denominator of Equation (7.6.2)**.** For each node $\mathbf{v} \in \text{nodes}(\mathcal{T})$, denote

$$
w_{\mathbf{v}}^{\text{den}} = \sum_{T \subset \mathcal{T}_{\mathbf{v}}} 2^{-\|T\|} \prod_{\mathbf{v}' \in \text{leaves}(T)} w_{\mathbf{v}'}, \qquad (7.6.6)
$$

where once again the sum over $T \subset \mathcal{T}_{\mathbf{v}}$ means the sum over all subtrees T of $\mathcal{T}$ rooted at $\mathbf{v}$. We have that (7.6.5) entails

$$
w_{\texttt{root}}^{\text{den}} = \sum_{T \subset \mathcal{T}_{\texttt{root}}} 2^{-\|T\|} \prod_{\mathbf{v} \in \text{leaves}(T)} w_{\mathbf{v}} = \sum_{T \subset \mathcal{T}_{\texttt{root}}} w_T = \sum_{T \subset \mathcal{T}} w_T \,. \qquad (7.6.7)
$$

So, we can compute recursively $w_{\texttt{root}}^{\text{den}}$ very efficiently, using a recursion on the weights $w_{\mathbf{v}}^{\text{den}}$ using Lemma 7.1 with $g(\mathbf{v}) = w_{\mathbf{v}}$. This leads to the recursion stated in Theorem 7.1, see Equation (7.2.7).

Now, we can exploit the fact that decision trees are built in a depth-first fashion in WildWood: all the nodes $\mathbf{v} \in \mathcal{T}$ are stored in a "flat" array, and by construction both the child nodes $\mathbf{v}0$ and $\mathbf{v}1$ have indexes that are larger than the one of $\mathbf{v}$. So, we can simply loop over the array of nodes in reverse order, and compute $w_{\mathbf{v}}^{\text{den}} = w_{\mathbf{v}}$ if $\mathbf{v} \in \text{leaves}(\mathcal{T})$ and $w_{\mathbf{v}}^{\text{den}} = \frac{1}{2} w_{\mathbf{v}} + \frac{1}{2} w_{\mathbf{v}0}^{\text{den}} w_{\mathbf{v}1}^{\text{den}}$ otherwise: we are guaranteed to have computed $w_{\mathbf{v}0}^{\text{den}}$ and $w_{\mathbf{v}1}^{\text{den}}$ before computing $w_{\mathbf{v}}^{\text{den}}$. This algorithm is described in Algorithm 4. Since these computations involve a large number of products with exponentiated numbers, it typically leads to strong over- and under-flows: we describe in Algorithm 4 a version of this algorithm which works recursively over the logarithms of the weights. At the end of this loop, we end up at $\mathbf{v} = \texttt{root}$ and have computed $w_{\texttt{root}}^{\text{den}} = \sum_{T \subset \mathcal{T}} w_T$ with a very efficient $O(|\text{nodes}(\mathcal{T})|)$ complexity. Note also that it is sufficient to store both $w_{\mathbf{v}}$ and $w_{\mathbf{v}}^{\text{den}}$ for all $\mathbf{v} \in \mathcal{T}$, which makes for a $O(|\text{nodes}(\mathcal{T})|)$ memory consumption.

Numerator of Equation (7.6.2)**.** The numerator of Equation (7.6.2) almost follows the exact same argument as the denominator, but since it depends on the input vector $x \in C$ of features for which we want to produce a prediction, it is performed at inference time. Recall that $\texttt{path}(x)$ is the sequence of nodes that leads to the leaf $\mathbf{v}_{\mathcal{T}}(x)$ containing $x \in C$ and define, for any $\mathbf{v} \in \text{nodes}(\mathcal{T})$, $\widehat{w}_{\mathbf{v}}(x) = w_{\mathbf{v}}\widehat{y}_{\mathbf{v}}(x)$ if $\mathbf{v} \in \texttt{path}(x)$, and $\widehat{w}_{\mathbf{v}}(x) = w_{\mathbf{v}}$ otherwise. We have

$$\begin{aligned}\sum_{T\subset\mathcal{T}} w_T\widehat{y}_T(x) &= \sum_{T\subset\mathcal{T}_{\texttt{root}}} w_T\widehat{y}_{\mathbf{v}_T(x)} \\ &= \sum_{T\subset\mathcal{T}_{\texttt{root}}} 2^{-\|T\|} \prod_{\mathbf{v}\in\text{leaves}(T)} w_{\mathbf{v}}\widehat{y}_{\mathbf{v}_T(x)} \qquad (7.6.8)\\ &= \sum_{T\subset\mathcal{T}_{\texttt{root}}} 2^{-\|T\|} \prod_{\mathbf{v}\in\text{leaves}(T)} \widehat{w}_{\mathbf{v}}(x). \qquad (7.6.9)\end{aligned}$$

Note that (7.6.8) comes from (7.6.5) while (7.6.9) comes from the definition of $\widehat{w}_{\mathbf{v}}(x)$ (note that a single term from the product over $\mathbf{v} \in \text{leaves}(T)$ corresponds to $\mathbf{v} = \mathbf{v}_T(x)$ since $\{C_{\mathbf{v}} : \mathbf{v} \in \text{leaves}(T)\}$ is a partition of C). We are now in position to use again Lemma 7.1 with $g(\mathbf{v}) = \widehat{w}_{\mathbf{v}}(x)$. Defining

$$w_{\mathbf{v}}^{\text{num}}(x) = \sum_{T\subset\mathcal{T}_{\mathbf{v}}} 2^{-\|T\|} \prod_{\mathbf{v}'\in\text{leaves}(T)} \widehat{w}_{\mathbf{v}'}(x)\,,$$

we can conclude that

$$w_{\texttt{root}}^{\text{num}}(x) = \sum_{T\subset\mathcal{T}} w_T\widehat{y}_T(x) \qquad (7.6.10)$$

and that the following recurrence holds:

$$w_{\mathbf{v}}^{\text{num}}(x) = \begin{cases}\widehat{w}_{\mathbf{v}}(x) & \text{if } \mathbf{v} \in \text{leaves}(\mathcal{T}) \\ \frac{1}{2}\widehat{w}_{\mathbf{v}}(x) + \frac{1}{2}w_{\mathbf{v}0}^{\text{num}}(x)w_{\mathbf{v}1}^{\text{num}}(x) & \text{otherwise.}\end{cases} \qquad (7.6.11)$$

This recurrence allows to compute $w_{\mathbf{v}}^{\text{num}}(x)$ from $\widehat{w}_{\mathbf{v}}(x)$, but note that a direct use of this formula would lead to a complexity $O(|\text{nodes}(\mathcal{T})|)$ to produce a prediction for a single input $x \in C$. It turns out can we can do much better than that.

Indeed, whenever $\mathbf{v} \notin \texttt{path}(x)$, we have by definition that $\widehat{w}_{\mathbf{v}}(x) = w_{\mathbf{v}}$ and that $\widehat{w}_{\mathbf{v}'}(x) = w_{\mathbf{v}'}$ for any descendant $\mathbf{v}'$ of $\mathbf{v}$, which entails by induction that $w_{\mathbf{v}}^{\text{num}}(x) = w_{\mathbf{v}}^{\text{den}}$ for any $\mathbf{v} \notin \texttt{path}(x)$. Therefore, we only need to explain how to compute $w_{\mathbf{v}}^{\text{num}}(x)$ for $\mathbf{v} \in \texttt{path}(x)$. This is achieved recursively, thanks to (7.6.11), starting at the leaf $\mathbf{v}_{\mathcal{T}}(x)$ and going up in the tree to $\texttt{root}$:

$$w_{\mathbf{v}}^{\text{num}}(x) = \begin{cases}w_{\mathbf{v}}\widehat{y}_{\mathbf{v}} & \text{if } \mathbf{v} = \mathbf{v}_{\mathcal{T}}(x) \\ \frac{1}{2}w_{\mathbf{v}}\widehat{y}_{\mathbf{v}} + \frac{1}{2}w_{\mathbf{v}(1-a)}^{\text{den}}w_{\mathbf{v}a}^{\text{num}}(x) & \text{otherwise, where } a \in \{0,1\} \text{ is s.t. } \mathbf{v}a \in \texttt{path}(x).\end{cases} \qquad (7.6.12)$$

Let us explain where this comes from: firstly, one has obviously that $\text{leaves}(\mathcal{T}) \cap \texttt{path}(x) = \{\mathbf{v}_{\mathcal{T}}(x)\}$, so that $w_{\mathbf{v}}^{\text{num}}(x) = g(\mathbf{v}) = \widehat{w}_{\mathbf{v}}(x) = w_{\mathbf{v}}\widehat{y}_{\mathbf{v}}(x)$ for $\mathbf{v} = \mathbf{v}_{\mathcal{T}}(x)$. Secondly, we go up in the tree along $\texttt{path}(x)$ and use again (7.6.11): whenever $\mathbf{v} \in \text{intnodes}(\mathcal{T})$ and $\mathbf{v}a \in \texttt{path}(x)$ for $a \in \{0,1\}$, we have $w_{\mathbf{v}(1-a)}^{\text{num}}(x) = w_{\mathbf{v}(1-a)}^{\text{den}}$ since $\mathbf{v}(1-a) \notin \texttt{path}(x)$. This recursion has a complexity $O(|\texttt{path}(x)|)$ where $|\texttt{path}(x)|$ is the number of nodes in $\texttt{path}(x)$, and is typically orders of magnitude smaller than $|\text{nodes}(\mathcal{T})|$ (in a well-balanced binary tree, one has the relation $|\texttt{path}(x)| = O(\log_2(|\text{nodes}(\mathcal{T})|))$). Moreover, we observe that the recursions used in (7.2.7) and (7.6.12) only need to save both $w_{\mathbf{v}}$ and $w_{\mathbf{v}}^{\text{den}}$ for any $\mathbf{v} \in \text{nodes}(\mathcal{T})$.

Finally, we have using (7.6.7) and (7.6.10) that

$$\widehat{f}(x) = \frac{\sum_{T\subset\mathcal{T}} w_T \widehat{y}_T(x)}{\sum_{T\subset\mathcal{T}} w_T} = \frac{w^{\mathrm{num}}_{\mathtt{root}}(x)}{w^{\mathrm{den}}_{\mathtt{root}}} =: \widehat{f}_{\mathtt{root}}(x),$$

and we want to compute $\widehat{f}_{\mathtt{root}}(x)$ recursively from $\widehat{f}_{\mathbf{v}}(x)$ where $\mathbf{v} \in \mathtt{path}(x)$. First, whenever $\mathbf{v} = \mathbf{v}_{\mathcal{T}}(x)$ we have

$$\widehat{f}_{\mathbf{v}}(x) = \frac{w^{\mathrm{num}}_{\mathbf{v}}(x)}{w^{\mathrm{den}}_{\mathbf{v}}} = \frac{w_{\mathbf{v}}\widehat{y}_{\mathbf{v}}}{w_{\mathbf{v}}} = \widehat{y}_{\mathbf{v}},$$

while for $\mathbf{v} \neq \mathbf{v}_{\mathcal{T}}(x)$ and $\mathbf{v} \in \mathtt{path}(x)$, we write

$$\widehat{f}_{\mathbf{v}}(x) = \frac{w^{\mathrm{num}}_{\mathbf{v}}(x)}{w^{\mathrm{den}}_{\mathbf{v}}} = \frac{\frac{1}{2}w_{\mathbf{v}}\widehat{y}_{\mathbf{v}} + \frac{1}{2}w^{\mathrm{den}}_{\mathbf{v}(1-a)}w^{\mathrm{num}}_{\mathbf{v}a}(x)}{w^{\mathrm{den}}_{\mathbf{v}}} \tag{7.6.13}$$

$$= \frac{1}{2}\frac{w_{\mathbf{v}}}{w^{\mathrm{den}}_{\mathbf{v}}}\widehat{y}_{\mathbf{v}} + \frac{1}{2}\frac{w^{\mathrm{den}}_{\mathbf{v}(1-a)}w^{\mathrm{den}}_{\mathbf{v}a}}{w^{\mathrm{den}}_{\mathbf{v}}}\frac{w^{\mathrm{num}}_{\mathbf{v}a}(x)}{w^{\mathrm{den}}_{\mathbf{v}a}} \tag{7.6.14}$$

$$= \frac{1}{2}\frac{w_{\mathbf{v}}}{w^{\mathrm{den}}_{\mathbf{v}}}\widehat{y}_{\mathbf{v}} + \Big(1 - \frac{1}{2}\frac{w_{\mathbf{v}}}{w^{\mathrm{den}}_{\mathbf{v}}}\Big)\widehat{f}_{\mathbf{v}a}(x), \tag{7.6.15}$$

where (7.6.13) comes from (7.6.12) while (7.6.15) comes from (7.2.7). This proves the recursion stated in Equation (7.2.6) from Theorem 7.1, and to Algorithm 5. This concludes the proof of Theorem 7.1. □

7.6.2 Proofs of the results from Section 7.3

The proof of Theorem 7.2 is partly inspired from the proof of Theorem 2 in [102], that we generalize to exp-concave losses, while only least-squares regression is considered therein. Let E be a measurable space and P, Q be probability measures on it. The Kullback-Leibler divergence between P and Q is defined by

$$\mathrm{KL}(P,Q) = \int_E \log\Big(\frac{\mathrm{d}P}{\mathrm{d}Q}\Big)\mathrm{d}P$$

whenever P is absolutely continuous with respect to Q and equal to $+\infty$ otherwise. Also, if $h : E \to \mathbb{R}$ is a measurable function such that $\int_E h\mathrm{d}P$ is well-defined on $\mathbb{R}\cup\{-\infty,+\infty\}$, we introduce

$$P_h := \frac{h}{\int_E h\mathrm{d}P}\cdot P,$$

the probability measure on E with density $h/\int h\mathrm{d}P$ with respect to P. A classical result is the Donsker-Varadhan variational formula [133], which is at the core of the proofs of many PAC-Bayesian theorems [303, 74] and that we use here as well in the proof of Theorem 7.2. It states that

$$\log\Big(\int_E \exp(h)\mathrm{d}Q\Big) + \mathrm{KL}(P,Q) - \int h\mathrm{d}P = \mathrm{KL}(P, Q_{\exp(h)}) \tag{7.6.16}$$

holds for any probability measures P and Q on E and any measurable function $h : E \to \mathbb{R}$. This entails in particular that

$$\log\Big(\int_E \exp(h)\mathrm{d}Q\Big) = \sup_P\Big\{\int h\mathrm{d}P - \mathrm{KL}(P,Q)\Big\},$$

where the supremum is over all probability measures on E, and where the supremum is achieved for $P = Q_{\exp(h)}$ whenever the term on the left-hand side is finite.

Proof of Theorem 7.2. Recall that the `oob` loss of a subtree $T \subset \mathcal{T}$ is given by

$$L_T = \sum_{i \in I_{\mathtt{oob}}} \ell(\widehat{y}_T(x_i), y_i)$$

and let us introduce

$$p_T = \frac{\pi(T)\exp(-\eta L_T)}{\sum_{T'} \pi(T')\exp(-\eta L_{T'})} \tag{7.6.17}$$

for any subtree $T \subset \mathcal{T}$. First, we use the fact that ℓ is a η-exp-concave loss function, hence η-mixable, see Section 3.3 from [79], which entails, since p_T is a probability measure over the set of all subtrees $T \subset \mathcal{T}$, that

$$\ell\Big(\sum_T p_T \widehat{y}_T(x_i), y_i\Big) \leqslant -\frac{1}{\eta}\log\Big(\sum_T p_T \exp(-\eta\ell(\widehat{y}_T(x_i), y_i))\Big),$$

where the sums over T are over all subtrees $T \subset \mathcal{T}$. Now, summing this inequality over $i \in I_{\mathtt{oob}}$ and using the convexity of the log-sum-exp function leads to

$$\begin{aligned}
\sum_{i \in I_{\mathtt{oob}}} \ell\Big(\sum_T p_T \widehat{y}_T(x_i), y_i\Big) &\leqslant -\frac{1}{\eta}\sum_{i \in I_{\mathtt{oob}}} \log\Big(\sum_T p_T \exp(-\eta\ell(\widehat{y}_T(x_i), y_i))\Big) \\
&\leqslant -\frac{n_{\mathtt{oob}}}{\eta}\log\Big(\sum_T p_T \exp\Big(-\frac{\eta}{n_{\mathtt{oob}}}\sum_{i \in I_{\mathtt{oob}}} \ell(\widehat{y}_T(x_i), y_i)\Big)\Big) \\
&= -\frac{n_{\mathtt{oob}}}{\eta}\log\Big(\sum_T p_T \exp\Big(-\frac{\eta}{n_{\mathtt{oob}}} L_T\Big)\Big).
\end{aligned}$$

By plugging the definition of p_T into the previous expression, and by introducing $\rho(T) := \eta L_T / n_{\mathtt{oob}}$, we obtain

$$\begin{aligned}
\frac{S}{n_{\mathtt{oob}}} &:= \frac{1}{n_{\mathtt{oob}}}\sum_{i \in I_{\mathtt{oob}}} \ell(\widehat{f}(x_i), y_i) \\
&\leqslant -\frac{1}{\eta}\log\Big(\sum_T \pi(T)\exp\big(-(n_{\mathtt{oob}}+1)\rho(T)\big)\Big) + \frac{1}{\eta}\log\Big(\sum_T \pi(T)\exp\big(-n_{\mathtt{oob}}\rho(T)\big)\Big).
\end{aligned}$$

The Hölder inequality implies that

$$\sum_T \pi(T)\exp\big(-n_{\mathtt{oob}}\rho(T)\big) \leqslant \Big(\sum_T \pi(T)\exp\big(-(n_{\mathtt{oob}}+1)\rho(T)\big)\Big)^{n_{\mathtt{oob}}/(n_{\mathtt{oob}}+1)},$$

thus

$$\frac{S}{n_{\mathtt{oob}}} \leqslant -\frac{1}{\eta(n_{\mathtt{oob}}+1)}\log\Big(\sum_T \pi(T)\exp\big(-(n_{\mathtt{oob}}+1)\rho(T)\big)\Big).$$

Using (7.6.16) with $h(T) = -(n_{\mathtt{oob}}+1)\rho(T)$ and $Q = \pi$, we have

$$\begin{aligned}
&\log\Big(\sum_T \pi(T)\exp\big(-(n_{\mathtt{oob}}+1)\rho(T)\big)\Big) \\
&\qquad = -\sum_T P(T)(n_{\mathtt{oob}}+1)\rho(T) - \mathrm{KL}(P,\pi) + \mathrm{KL}(P,\pi_{\exp(h)})
\end{aligned}$$

for any probability measure P over the set of subtrees of $\mathcal{T}$. This leads to

$$\frac{1}{n_{\mathsf{oob}}}\sum_{i\in I_{\mathsf{oob}}}\ell(\widehat{f}(x_i),y_i)\leqslant\frac{1}{n_{\mathsf{oob}}}\sum_{T}P(T)L_T+\frac{1}{\eta(n_{\mathsf{oob}}+1)}\mathrm{KL}(P,\pi)$$

for any P, since $\mathrm{KL}(P,\pi_{\exp(h)})\geqslant 0$. So for the particular choice $P=\delta_T$ (the Dirac mass at T) for any subtree $T\subset\mathcal{T}$, we have

$$\begin{aligned}\frac{1}{n_{\mathsf{oob}}}\sum_{i\in I_{\mathsf{oob}}}\ell(\widehat{f}(x_i),y_i)&\leqslant\frac{1}{n_{\mathsf{oob}}}L_T+\frac{1}{\eta(n_{\mathsf{oob}}+1)}\log\left(\pi(T)^{-1}\right)\\&\leqslant\frac{1}{n_{\mathsf{oob}}}L_T+\frac{\log 2}{\eta}\frac{\|T\|}{n_{\mathsf{oob}}+1},\end{aligned}$$

which concludes the proof of Theorem 7.2. □

The proof of Corollary 7.1 requires the next Lemma.

Lemma 7.2. *Consider classification with $\mathcal{Y}=\{1,\dots,K\}$ and a node $\mathbf{v}\in\mathrm{nodes}(\mathcal{T})$. Denote $n_{\mathbf{v}}(k)$ the number of samples of class k in node $\mathbf{v}$. We consider the Krichevsky-Trofimov estimator*

$$\widehat{y}(k)=\frac{n_{\mathbf{v}}(k)+1/2}{n_{\mathbf{v}}+K/2}$$

where $n_{\mathbf{v}}=\sum_{k=1}^{K}n_{\mathbf{v}}(k)$ and the log loss $\ell(y',y)=-\log y'(y)$. Then, we have the inequality

$$\sum_i\ell(\widehat{y},y_i)-\inf_p\sum_i\ell(p,y_i)\leqslant\frac{K-1}{2}.$$

Proof. We know that the optimal p is given by $p_k=n_{\mathbf{v}}(k)/n_{\mathbf{v}}$. Indeed, it is the solution to the following constrained convex optimization problem

$$\min_p\ -\sum_{k=1}^{K}n_{\mathbf{v}}(k)\log p_k\quad\text{ subject to }\quad\sum_{k=1}^{K}p_k=1,$$

where we consider non-negativity to be already enforced by the objective and imposing $p_k\leqslant 1$ is redundant with the constraint $\sum_{k=1}^{K}p_k=1$. We can write the Lagrangian function as

$$L(p,\lambda)=-\sum_{k=1}^{K}n_{\mathbf{v}}(k)\log p_k+\lambda\Big(\sum_{k=1}^{K}p_k-1\Big)$$

and one can check the KKT conditions when taking $p_k=n_{\mathbf{v}}(k)/n_{\mathbf{v}}$ and $\lambda=n_{\mathbf{v}}$. Since we are dealing with a convex problem with linear constraints, this is a sufficient optimality condition. Straightforward computations give

$$\begin{aligned}\sum_i\ell(\widehat{y},y_i)-\inf_p\sum_i\ell(p,y_i)&=\sum_{k=1}^{K}-n_{\mathbf{v}}(k)\log(\widehat{y}(k))-\sum_{k=1}^{K}-n_{\mathbf{v}}(k)\log p_k\\&=\sum_{k=1}^{K}-n_{\mathbf{v}}(k)\log\frac{n_{\mathbf{v}}(k)+1/2}{n_{\mathbf{v}}+K/2}-\sum_{k=1}^{K}-n_{\mathbf{v}}(k)\log\frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}}\end{aligned}$$

$$= \sum_{k=1}^{K} -n_{\mathbf{v}}(k)\Big(\log \frac{n_{\mathbf{v}}}{n_{\mathbf{v}} + K/2} + \log \frac{n_{\mathbf{v}}(k) + 1/2}{n_{\mathbf{v}}} - \log \frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}} \Big)$$
$$= -n_{\mathbf{v}} \log \frac{n_{\mathbf{v}}}{n_{\mathbf{v}} + K/2} + \sum_{k=1}^{K} -n_{\mathbf{v}}(k)\Big(\log \frac{n_{\mathbf{v}}(k) + 1/2}{n_{\mathbf{v}}} - \log \frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}} \Big)$$
$$= n_{\mathbf{v}} \log \frac{n_{\mathbf{v}} + K/2}{n_{\mathbf{v}}} + \sum_{k=1}^{K} n_{\mathbf{v}}(k) \log \frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}(k) + 1/2}.$$

Now, using the concavity of the logarithm gives

$$\sum_i \ell(\widehat{y}, y_i) - \inf_p \sum_i \ell(p, y_i) \leqslant n_{\mathbf{v}} \log \frac{n_{\mathbf{v}} + K/2}{n_{\mathbf{v}}} + n_{\mathbf{v}} \log \Big(\sum_{k=1}^{K} \frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}} \frac{n_{\mathbf{v}}(k)}{n_{\mathbf{v}}(k) + 1/2} \Big),$$

and the fact that $x \mapsto x/(x + 1/2)$ is non-decreasing and $n_{\mathbf{v}}(k) \leqslant n_{\mathbf{v}}$ leads to

$$\begin{aligned} \sum_i \ell(\widehat{y}, y_i) - \inf_p \sum_i \ell(p, y_i) &\leqslant n_{\mathbf{v}} \log \frac{n_{\mathbf{v}} + K/2}{n_{\mathbf{v}}} + n_{\mathbf{v}} \log \frac{n_{\mathbf{v}}}{n_{\mathbf{v}} + 1/2} \\ &= n_{\mathbf{v}} \log \frac{n_{\mathbf{v}} + 1/2 + (K-1)/2}{n_{\mathbf{v}} + 1/2} \\ &= n_{\mathbf{v}} \log \Big(1 + \frac{K-1}{2n_{\mathbf{v}} + 1} \Big) \leqslant \frac{K-1}{2}. \end{aligned}$$

This concludes the proof of Lemma 7.2. □

Proof of Corollary 7.1. The log-loss is trivially 1-exp-concave, so that we can choose $\eta = 1$. Following Theorem 7.2, it remains to bound the regret of the tree forecaster T with respect to the optimal labeling of its leaves. For classification and the log loss, we use Lemma 7.2 to obtain

$$\sum_{i \in I_{\mathsf{oob}} : x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_T(x_i), y_i) - \inf_p \sum_{i \in I_{\mathsf{oob}} : x_i \in C_{\mathbf{v}}} \ell(p, y_i) \leqslant \frac{K-1}{2}$$

for any subtree T and any $\mathbf{v} \in \text{leaves}(T)$. Now, summing over $\mathbf{v} \in \text{leaves}(T)$, of cardinality $(\|T\| + 1)/2$, leads to

$$\sum_{i \in I_{\mathsf{oob}}} \ell(\widehat{f}(x_i), y_i) - \sum_{i \in I_{\mathsf{oob}}} \ell(g_T(x_i), y_i) \leqslant \|T\| \log 2 + \frac{(K-1)(\|T\| + 1)}{4},$$

which concludes the proof of Corollary 7.1. □

Proof of Corollary 7.2. The square loss is $1/(8B^2)$-exp-concave on $[-B, B]$, see [79], so we can choose $\eta = 1/(8B^2)$. Following Theorem 7.2, it remains to bound the regret of the tree forecaster T with respect to the optimal labeling of its leaves. For regression with the least-squares loss, since we use the empirical mean forecaster (7.2.2), we have

$$\sum_{i \in I_{\mathsf{oob}} : x_i \in C_{\mathbf{v}}} \ell(\widehat{y}_T(x_i), y_i) - \inf_b \sum_{i \in I_{\mathsf{oob}} : x_i \in C_{\mathbf{v}}} \ell(b, y_i) = 0$$

for any subtree T and any leaf $\mathbf{v} \in \text{leaves}(T)$. The rest of the proof follows that of Corollary 7.1. □

7.7 Experimental details

We report in Table 7.3 the same test AUC scores as in Table 7.1 of all algorithms after hyper-optimization on the considered datasets. Standard-deviations displayed between parentheses are computed from 5 trainings with different random seeds. We observe that WW has better (or identical in some cases) performances than RF on all datasets and that it is close to that of EGB libraries (bold is for best EGB performance, underline for best `RF`n, `WW`n or `RD`n performance). Table 7.4 displays the results of the same experiment measured using the log loss.

We report also in Table 7.5 the same training time and test AUC as in Table 7.2, with standard-deviations displayed between parentheses computed from 5 trainings with different random seeds, all with default hyperparameters of each algorithm. We observe that WW is generally among the fastest algorithms, for performances comparable to ones of all baselines (bold is for best EGB training time or performance, underline for best `RF` or `WW` training time or performance).

7.7.1 Supplementary details about hyperparameter tuning

In this Section, we provide extra information about hyperparameters optimization. For XGBoost, LightGBM and CatBoost, with all other hyperparameters fixed, we use early stopping by monitoring the log loss on the validation set, the maximum number of boosting iterations being set at $5,000$. The best number of iterations is used together with other best hyperparameters to refit over the whole training set before evaluation on the test set. For `scikit-learn`'s Random Forest and WildWood, we report results both for 10 and 100 trees, note that the default choice is 10 for WildWood (since subtrees aggregation allows to use fewer trees than RF) while default is 100 in `scikit-learn`. We list the hyperparameters search space of each algorithm below.

XGBoost

- `eta`: log-uniform distribution $[e^{-7}, 1]$;
- `max_depth`: discrete uniform distribution $[2, 10]$;
- `subsample`: uniform $[0.5, 1]$;
- `colsample_bytree`: uniform $[0.5, 1]$;
- `colsample_bylevel`: uniform $[0.5, 1]$;
- `min_child_weight`: log-uniform distribution $[e^{-16}, e^{5}]$;
- `alpha`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;
- `lambda`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;
- `gamma`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;

LightGBM

- `learning_rate`: log-uniform distribution $[e^{-7}, 1]$;
- `num_leaves`: discrete log-uniform distribution $[1, e^{7}]$;
- `feature_fraction`: uniform $[0.5, 1]$;
- `bagging_fraction`: uniform $[0.5, 1]$;
- `min_data_in_leaf`: discrete log-uniform distribution $[1, e^{6}]$;
- `min_sum_hessian_in_leaf`: log-uniform distribution $[e^{-16}, e^{5}]$;
- `lambda_l1`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;
- `lambda_l2`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;

Table 7.3: The same test AUC scores as in Table 7.1 of all algorithms after hyperoptimization on the considered datasets. Standard-deviations displayed between parentheses are computed from 5 trainings with different random seeds. We observe that WW has better (or identical in some cases) performances than RF on all datasets and that it is close to that of EGB libraries (bold is for best EGB performance, underline for best RFn, WWn or RDn performance).

	XGB	LGBM	CB	HGB	RF10	RF100	WW10	WW100	RD10	RD100
adult	0.930 (2.7e-04)	**0.931** (1.2e-04)	0.927 (2.9e-04)	0.930 (2.9e-04)	0.915(5.0e-04)	0.918(1.7e-04)	0.916(4.1e-04)	0.919(1.4e-04)	0.915(3.4e-04)	0.917(1.5e-04)
bank	0.933 (1.5e-04)	**0.935** (4.1e-05)	0.925 (6.5e-04)	0.930 (7.4e-04)	0.919(6.0e-04)	0.929(2.1e-04)	0.926(3.8e-04)	0.931(1.6e-04)	0.919(9.0e-04)	0.922(2.2e-04)
breastcancer	0.991 (4.4e-04)	0.993 (1.1e-04)	0.987 (6.7e-03)	**0.994** (0.0e+00)	0.987(2.1e-03)	0.992(4.1e-04)	0.989(2.6e-03)	0.992(3.1e-04)	0.987(2.9e-03)	0.985(8.3e-04)
car	0.999 (2.3e-04)	**1.000** (3.8e-05)	**1.000** (6.0e-05)	**1.000** (0.0e+00)	0.997(1.1e-03)	0.998(2.8e-04)	0.998(6.3e-04)	0.998(1.5e-04)	0.993(8.7e-04)	0.993(1.4e-04)
covtype	**0.999** (7.5e-06)	**0.999** (3.6e-06)	0.998 (2.5e-05)	**0.999** (5.2e-06)	0.997(1.4e-04)	0.998(1.4e-04)	0.996(1.9e-04)	0.998(3.7e-05)	0.974(7.6e-04)	0.996(4.2e-05)
default-cb	0.780 (3.5e-04)	**0.783** (1.2e-04)	0.780 (3.9e-04)	0.779 (6.7e-04)	0.765(3.9e-03)	0.775(7.4e-04)	0.774(4.4e-04)	0.778(1.6e-03)	0.772(9.0e-04)	0.773(3.5e-04)
higgs	0.853 (6.7e-05)	**0.857** (1.8e-05)	0.847 (2.2e-05)	0.853 (7.0e-05)	0.820(9.9e-05)	0.837(4.5e-05)	0.820(1.5e-04)	0.837(6.4e-05)	0.813(1.1e-04)	0.815(5.3e-05)
internet	0.934 (2.9e-04)	0.910 (1.8e-04)	**0.938** (1.3e-03)	0.911 (1.1e-16)	0.917(2.3e-03)	0.935(6.1e-04)	0.926(1.5e-03)	0.928(7.9e-04)	0.925(4.5e-04)	0.928(8.9e-04)
kddcup	**1.000** (6.1e-08)	**1.000** (4.1e-07)	**1.000** (5.8e-07)	**1.000** (5.7e-07)	0.997(5.2e-05)	0.998(2.1e-03)	1.000(1.0e-05)	1.000(6.0e-06)	1.000(1.6e-05)	1.000(6.8e-06)
kick	**0.777** (7.7e-04)	0.770 (2.8e-04)	**0.777** (6.8e-04)	0.771 (1.6e-03)	0.749(1.1e-03)	0.764(7.9e-04)	0.756(8.3e-04)	0.763(5.4e-04)	0.752(9.7e-04)	0.754(5.2e-04)
letter	**1.000** (1.3e-05)	**1.000** (2.7e-06)	**1.000** (4.1e-06)	**1.000** (2.1e-05)	0.997(2.6e-04)	0.999(1.2e-04)	0.997(2.3e-04)	0.999(5.5e-05)	0.995(4.8e-04)	0.993(2.4e-04)
satimage	**0.991** (2.1e-04)	**0.991** (3.6e-05)	**0.991** (2.3e-04)	0.987 (0.0e+00)	0.985(1.3e-03)	0.991(3.8e-04)	0.986(1.2e-03)	0.991(1.7e-04)	0.984(6.4e-04)	0.983(1.9e-04)
sensorless	**1.000** (4.9e-07)	**1.000** (1.4e-07)	**1.000** (4.4e-06)	**1.000** (2.9e-06)	1.000(2.3e-05)	1.000(5.0e-07)	1.000(1.1e-04)	1.000(1.4e-05)	0.999(5.4e-05)	1.000(7.3e-06)
spambase	**0.990** (1.5e-04)	**0.990** (5.2e-05)	0.987 (1.2e-03)	0.986 (0.0e+00)	0.980(1.8e-03)	0.987(2.3e-04)	0.983(1.0e-03)	0.987(2.2e-04)	0.972(1.1e-03)	0.974(2.9e-04)

Table 7.4: The results of Table 7.1 measured through log-loss corresponding to all algorithms after hyperoptimization on the considered datasets. We observe that WW often improves the performance of RF or achieves a close result (smaller is better, bold is for best EGB performance, underline for best RFn, WWn or RDn performance).

	XGB	LGBM	CB	HGB	RF10	RF100	WW10	WW100	RD10	RD100
adult	0.273	**0.271**	0.281	0.275	0.302	0.294	0.296	0.293	0.299	0.296
bank	0.200	**0.196**	0.208	0.202	0.214	0.207	0.206	0.201	0.213	0.210
breastcancer	0.126	**0.101**	0.138	0.119	0.155	0.122	0.135	0.126	0.164	0.176
car	0.048	0.040	0.051	**0.015**	0.163	0.085	0.078	0.078	0.194	0.190
covtype	0.087	**0.075**	0.126	0.079	0.164	0.118	0.141	0.123	0.495	0.233
default-cb	0.430	**0.429**	0.430	0.431	0.439	0.432	0.434	0.432	0.435	0.435
higgs	0.472	**0.467**	0.481	0.484	0.517	0.499	0.518	0.499	0.525	0.523
internet	1.446	1.552	**1.413**	1.573	1.564	1.450	1.505	1.520	1.561	1.560
kddcup	**0.000**	**0.000**	**0.000**	**0.000**	0.000	0.000	0.000	0.000	0.001	0.001
kick	**0.293**	0.298	**0.293**	0.295	0.306	0.297	0.308	0.306	0.303	0.302
letter	0.116	**0.111**	0.113	0.138	0.473	0.276	0.358	0.274	0.691	0.807
satimage	**0.227**	0.234	0.228	0.265	0.333	0.261	0.313	0.265	0.372	0.365
sensorless	**0.004**	**0.004**	**0.004**	0.005	0.033	0.025	0.035	0.027	0.074	0.075
spambase	**0.115**	0.121	0.136	0.137	0.203	0.216	0.178	0.160	0.215	0.210

Table 7.5: The same training time table as in Table 7.2, as average over 5 runs, with standard deviation computed from 5 runs reported between parenthesis, for default parameters for each model. Top: training time in seconds; bottom: test AUC. We observe that WW is generally among the fastest algorithms, for performances comparable to ones of all baselines (bold is for best EGB training time or performance, underline for best RF or WW training time or performance).

	Training time (seconds)					
	XGB	LGBM	CB	HGB	RF	WW
covtype	10 (0.6)	**3** (0.1)	120 (9.3)	14 (7.7)	21 (0.9)	3 (0.1)
higgs	36 (0.6)	**30** (1.4)	653 (8.7)	85 (0.2)	1389 (11.1)	179 (4.5)
internet	9 (0.7)	**4** (0.1)	188 (2.4)	8 (0.3)	0.4 (0.0)	0.3 (0.0)
kddcup	175 (5.1)	41 (2.6)	2193 (13.2)	**31** (0.2)	208 (3.8)	12 (0.8)
kick	7 (0.2)	**0.4** (0.0)	50 (0.7)	0.7 (0.1)	31 (0.1)	5 (0.0)

	Test AUC					
	XGB	LGBM	CB	HGB	RF	WW
covtype	0.986 (2e-04)	0.978 (2e-03)	**0.989** (9e-05)	0.960 (1e-02)	0.998 (6e-05)	0.979 (5e-04)
higgs	0.823 (3e-04)	0.812 (2e-04)	**0.840** (8e-05)	0.812 (2e-04)	0.838 (9e-05)	0.813 (1e-04)
internet	**0.918** (2e-05)	0.828 (0e+00)	0.910 (8e-03)	0.500 (0e+00)	0.862 (3e-03)	0.889 (7e-03)
kddcup	**1.000** (3e-07)	0.638 (3e-02)	0.988 (7e-03)	0.740 (6e-02)	0.998 (2e-03)	1.000 (3e-06)
kick	0.768 (4e-04)	0.757 (0e+00)	**0.781** (3e-04)	0.773 (2e-03)	0.747 (2e-03)	0.751 (2e-03)

CatBoost

- `learning_rate`: log-uniform distribution $[e^{-7}, 1]$;
- `random_strength`: discrete uniform distribution over $\{1, 20\}$;
- `one_hot_max_size`: discrete uniform distribution over $\{0, 25\}$;
- `l2_leaf_reg`: log-uniform distribution $[1, 10]$;
- `bagging_temperature`: uniform $[0, 1]$.

HistGradientBoosting

- `learning_rate`: log-uniform distribution $[e^{-4}, 1]$;
- `max_leaf_nodes`: discrete log-uniform distribution $[1, e^{7}]$;
- `min_samples_leaf`: discrete log-uniform distribution $[1, e^{6}]$;
- `l2_regularization`: 0 with probability 0.5, and log-uniform distribution $[e^{-16}, e^{2}]$ with probability 0.5;

RandomForest

- `max_features`: uniform among `None`, `sqrt`, `log2`, `0.25`, `0.5` and `0.75`;
- `max_depth`: uniform among `None`, `sqrt` and `log2`, the latter two are meant in terms of the train sample size;
- `min_samples_leaf`: uniform over $\{1, 5, 10\}$ and we set `min_samples_split` = $2 \times$ `min_samples_leaf`;

WildWood

- `multiclass`: `multinomial` with probability 0.5, and `ovr` with probability 0.5;
- `min_samples_leaf`: uniform over $\{1, 5, 10\}$ and we set `min_samples_split` = $2 \times$ `min_samples_leaf`;
- `step`: log-uniform distribution $[e^{-3}, e^{6}]$;
- `dirichlet`: log-uniform distribution $[e^{-7}, e^{2}]$;
- `cat_split_strategy`: `binary` with probability 0.5, and `all` with probability 0.5;
- `max_features`: uniform among `None`, `sqrt`, `log2`, `0.25`, `0.5` and `0.75`;
- `max_depth`: uniform among `None`, `sqrt` and `log2`, the latter two are meant in terms of the train sample size;

7.7.2 Datasets

The main characteristics of the datasets used in the paper are summarized in Table 7.6. We provide in Table 7.7 the URL of all the datasets used, most of them are from the UCI machine learning repository [136].

Note that the HCV data set labels were binarized by considering the class "Blood Donor" against all the others.

7.7.3 Sensitivity of hyperparameters of Wildwood

In Table 7.8 we illustrate the effects of hyperparameters on WW's performance on a few datasets, measured by the test AUC. We can observe in this table that it is only weakly affected by different combinations of hyperparameters.

7.7.4 Supplementary details about assets used (versions and licenses)

The versions and licenses of the libraries used in our experiments are:

- `catboost` (0.25.1), Apache License 2.0
- `hyperopt` (0.2.5), license: `https://github.com/hyperopt/hyperopt/blob/master/LICENSE.txt`
- `joblib` (0.17), BSD-3-Clause License
- `lightgbm` (3.2.1), MIT License

Dataset	# Samples	# Features	# Categorical features	# Classes	Gini
adult	48,842	14	8	2	0.36
bank	45,211	16	10	2	0.21
banknote	1,372	4	0	2	0.49
breastcancer	569	30	0	2	0.47
car	1,728	6	6	4	0.46
covtype	581,012	54	0	7	0.62
default_cb	30,000	23	3	2	0.34
higgs	11,000,000	28	0	2	0.50
HCV	615	13	2	2	0.19
internet	10,108	70	70	46	0.88
ionosphere	351	34	0	2	0.46
kddcup99	4,898,431	41	7	23	0.58
kick	72,983	32	18	2	0.22
letter	20,000	16	0	26	0.96
satimage	5,104	36	0	6	0.81
sensorless	58,509	48	0	11	0.91
spambase	4,601	57	0	2	0.48

Table 7.6: Main characteristics of the datasets used in experiments, including number of samples, number of features, number of categorical features, number of classes and the Gini index of the class distribution on the whole datasets (rescaled between 0 and 1), in order to quantify class unbalancing.

Dataset	URL
adult	`https://archive.ics.uci.edu/ml/datasets/Adult`
bank	`https://archive.ics.uci.edu/ml/datasets/bank+marketing`
banknote	`https://archive.ics.uci.edu/ml/datasets/banknote+authentication`
breastcancer	`https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)`
car	`https://archive.ics.uci.edu/ml/datasets/car+evaluation`
covtype	`https://archive.ics.uci.edu/ml/datasets/covertype`
default_cb	`https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`
HCV	`https://archive.ics.uci.edu/ml/datasets/HCV+data`
higgs	`https://archive.ics.uci.edu/ml/datasets/HIGGS`
internet	`https://kdd.ics.uci.edu/databases/internet_usage/internet_usage.html`
ionosphere	`https://archive.ics.uci.edu/ml/datasets/ionosphere`
kddcup99	`https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
kick	`https://www.openml.org/d/41162`
letter	`https://archive.ics.uci.edu/ml/datasets/letter+recognition`
satimage	`https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)`
sensorless	`https://archive.ics.uci.edu/ml/datasets/dataset+for+sensorless+drive+diagnosis`
spambase	`https://archive.ics.uci.edu/ml/datasets/spambase`

Table 7.7: The URLs of all the datasets used in the paper, giving direct download links and supplementary details.

adult			
$n_{\text{min-leaf}}$	α	η	AUC
1	0.1	0.1	0.913
1	0.1	1.0	0.916
1	0.1	10.0	0.918
1	0.5	0.1	0.913
1	0.5	1.0	0.917
1	0.5	10.0	0.919
1	2.5	0.1	0.913
1	2.5	1.0	0.917
1	2.5	10.0	0.919
5	0.1	0.1	0.913
5	0.1	1.0	0.916
5	0.1	10.0	0.918
5	0.5	0.1	0.913
5	0.5	1.0	0.916
5	0.5	10.0	0.918
5	2.5	0.1	0.913
5	2.5	1.0	0.917
5	2.5	10.0	0.918
10	0.1	0.1	0.913
10	0.1	1.0	0.916
10	0.1	10.0	0.917
10	0.5	0.1	0.913
10	0.5	1.0	0.916
10	0.5	10.0	0.917
10	2.5	0.1	0.913
10	2.5	1.0	0.916
10	2.5	10.0	0.918

bank			
$n_{\text{min-leaf}}$	α	η	AUC
1	0.1	0.1	0.919
1	0.1	1.0	0.926
1	0.1	10.0	0.929
1	0.5	0.1	0.920
1	0.5	1.0	0.927
1	0.5	10.0	0.929
1	2.5	0.1	0.921
1	2.5	1.0	0.927
1	2.5	10.0	0.929
5	0.1	0.1	0.919
5	0.1	1.0	0.926
5	0.1	10.0	0.928
5	0.5	0.1	0.920
5	0.5	1.0	0.926
5	0.5	10.0	0.928
5	2.5	0.1	0.921
5	2.5	1.0	0.927
5	2.5	10.0	0.928
10	0.1	0.1	0.920
10	0.1	1.0	0.926
10	0.1	10.0	0.927
10	0.5	0.1	0.920
10	0.5	1.0	0.926
10	0.5	10.0	0.927
10	2.5	0.1	0.920
10	2.5	1.0	0.926
10	2.5	10.0	0.928

car			
$n_{\text{min-leaf}}$	α	η	AUC
1	0.1	0.1	0.992
1	0.1	1.0	0.995
1	0.1	10.0	0.995
1	0.5	0.1	0.992
1	0.5	1.0	0.994
1	0.5	10.0	0.995
1	2.5	0.1	0.990
1	2.5	1.0	0.992
1	2.5	10.0	0.993
5	0.1	0.1	0.990
5	0.1	1.0	0.992
5	0.1	10.0	0.992
5	0.5	0.1	0.990
5	0.5	1.0	0.992
5	0.5	10.0	0.992
5	2.5	0.1	0.987
5	2.5	1.0	0.991
5	2.5	10.0	0.991
10	0.1	0.1	0.983
10	0.1	1.0	0.987
10	0.1	10.0	0.987
10	0.5	0.1	0.983
10	0.5	1.0	0.987
10	0.5	10.0	0.987
10	2.5	0.1	0.981
10	2.5	1.0	0.985
10	2.5	10.0	0.986

Table 7.8: Areas under the ROC curves (AUC) obtained on test samples with WildWood (using 100 trees in the forest) on the adult, bank and car datasets with combinations of several hyper parameters. We observe that WildWood's performance does not vary significantly with respect to these hyperparameters.

- `matplotlib` (3.3.1), license: `https://github.com/matplotlib/matplotlib/blob/master/LICENSE/LICENSE`
- `numba` (0.52), BSD 2-Clause "Simplified" License
- `numpy` (1.19.2), BSD-3-Clause License
- `pandas` (1.2.4), BSD-3-Clause License
- `python` (3.7.9), Python Software Fundation Licence version 2
- `scikit-learn` (0.24.2), BSD 3-Clause License
- `scipy` (1.5.4), BSD 3-Clause License
- `seaborn` (0.11), BSD-3-Clause License
- `xgboost` (1.4.1), Apache License 2.0

All the datasets used are publicly accessible and have no copyright restrictions.

Appendix A

Open Problem: Efficient Optimal Regret Logistic Regression

This chapter is based on joint work with Yiyang Yu, Jaouad Mourtada and Stéphane Gaïffas.

Contents

A.1 Introduction

We consider the logistic regression problem which is a classical and widely used model for binary classification.

Given inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} = \mathbb{R}^d$ a Euclidean space and $\mathcal{Y} = \{\pm 1\}$ representing binary labels, Logistic regression predicts the probability $\mathbb{P}(y = +1|x)$ using a linear parameter $\theta \in \mathbb{R}^d$:

$$p(y = +1|x, \theta) = 1 - p(y = -1|x, \theta) = \sigma(\widehat{y}) \quad \text{with} \quad \widehat{y} = \theta^\top x,$$

where $\sigma(u) = e^u/(1 + e^u)$ is the sigmoid function. This corresponds to the generalized linear model using the sigmoid as link function. The performance of the above prediction is evaluated by the log-loss

$$\ell(\widehat{y}, y) = \begin{cases} -\log(p(y = +1|x, \theta)) & \text{if } y = -1 \\ -\log(p(y = -1|x, \theta)) & \text{otherwise.} \end{cases}$$

Given the definition of the sigmoid function, this is equivalent to using the logistic loss function on the linear prediction $\widehat{y}$ as follows

$$\ell(\widehat{y}, y) = \log(1 + \exp(-\widehat{y}y)).$$

Extension to multiple classes. The logistic regression model can be extended to multiple label problems $y \in \{1, \dots, K\}$ with $K > 2$ by replacing the logistic function with the softmax function $\boldsymbol{\sigma} : \mathbb{R}^K \to \Delta_K$ defined as $\boldsymbol{\sigma}(v) = \left(\frac{e^{v_k}}{\sum_{j=1}^K e^{v_j}}\right)_{k=1}^K$ where Δ_K is the probability simplex of

ditributions over $\{1, \dots, K\}$. The linear predictions are then computed as $\widehat{y} = Wx \in \mathbb{R}^K$ for a matrix $W \in \mathbb{R}^{K \times d}$.

We will consider the online setting of logistic regression where instances $(x_t, y_t)_{t \geqslant 1}$ arrive sequentially. More precisely, the two following steps occur at each turn $t \geqslant 1$:

- The agent receives a sample x_t from the environment and uses it along with the history up to this point $H_t = \{(x_1, y_1), \dots, (x_{t-1}, y_{t-1})\}$ in order to make a prediction $\widehat{y}_t$ of y_t.
- Then, the environment reveals the value of the label y_t and the agent incurs the logistic loss $\ell(\widehat{y}_t, y_t)$.

The performance of the agent's predictions is evaluated through the notion of regret. Given a reference linear predictor $\theta \in \mathbb{R}^d$, the regret with respect to θ after n rounds is defined as

$$R_n(\theta) = \sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \sum_{t=1}^{n} \ell(\theta^\top x_t, y_t).$$

Conventionally, the reference point θ is taken as the best competitor in hindsight within a comparison class $\Theta \subset \mathbb{R}^d$, yielding the worst case regret as

$$R_n := \max_{\theta \in \Theta} R_n(\theta) = \sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \min_{\theta \in \Theta} \sum_{t=1}^{n} \ell(\theta^\top x_t, y_t).$$

Depending on the strategy employed by the agent, the regret R_n will have different asymptotic rates in the number of rounds n. In the worst case, the regret R_n has a linear rate in n meaning that the agent fails to learn the task at hand since at least a constant loss is incurred at each round on average. Good introductory references to algorithm design for online learning are provided by the manuals of [183, 77, 347]. The following pair of basic assumptions define the conventional setting for online logistic regression [185, 218, 325].

Assumption A.1. *The features are bounded i.e. for all t we have $\|x_t\| \leqslant R$.*

Assumption A.2. *The comparison class determining the regret is fixed as the Euclidean ball of radius B i.e. $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\| \leqslant B\}$.*

The online logistic regression problem is an instance of online convex optimization [183] and, as such, it can be solved using algorithms such as Online Gradient Descent (OGD) [460] or Online Newton Step (ONS) [184]. The two previous algorithms achieve regret bounds of order $O(BR\sqrt{n})$ and $O(de^{BR}\log(n))$ with complexities $O(nd)$ and $O(nd^2)$ respectively. While the logarithmic dependence in n of ONS is far better than $\sqrt{n}$, the exponential factor in B which comes along makes it only interesting as long as $B \leqslant \frac{1}{2}\log(n)$. The question of whether a logarithmic regret bound can be obtained without this exponential dependence was raised in [306] and later answered negatively by [185] who showed a $\Omega(\sqrt{n})$ lower bound in the regime $B \approx \log(n)$. However, this limitation was shown to be superable in the work of [149] using an *improper* algorithm.

A learning algorithm is said to be proper if, at each round t, it makes predictions $\widehat{y}_t = \theta_t^\top x_t$ with θ_t depending only on samples up to $t-1$ and not on x_t. An improper algorithm is one which also uses the knowledge of x_t for the determination of the linear parameter θ_t used to compute the prediction $\widehat{y}_t$ at round t.

Thanks to an improper Bayesian algorithm, [149] managed to obtain regret $O(d\log(BRn))$ which is optimal. However, the algorithm they propose requires the computation of a Bayesian mixture posterior using MCMC integration methods with a prohibitive complexity of $O(B^6 n^{12}(Bn+$

d)[12]). Later works [218, 4, 219] proposed other more efficient improper algorithms but with sub-optimal regret bounds of order $O(dBR\log(BRn))$.

In this chapter, we present a new online learning algorithm which is likely to achieve the optimal regret bound at an improved computational cost in $O(n^2d^2)$.

A.2 Literature review

We first provide a quick nonexhaustive review of the important online logistic regression algorithms in the literature. Recall that, for an individual sample, the loss function was defined as $\ell(\theta^\top x_t, y_t) = -\log\sigma(y_t\theta^\top x_t)$. With a slight abuse of notation, we define, for each round t, the function $\ell_t(\theta) = \log(1+\exp(-y_t\theta^\top x_t))$.

Regret analyses crucially rely on the properties of the logistic function which can be translated on the episodic losses ℓ_t. The latter is differentiable w.r.t. θ and we have $\nabla\ell_t(\theta) = -y_tx_t\sigma(y_t\theta^\top x_t)$ and $\nabla^2\ell(\theta) = \sigma'(y_t\theta^\top x_t)x_tx_t^\top$ where $\sigma'(u) = \sigma(u)(1-\sigma(u))$.

Online Gradient Descent [460] makes predictions based on iterates θ_t updated using the iteration

$$\theta_t = \mathrm{proj}_\Theta(\theta_{t-1} - \beta_t\nabla\ell_{t-1}(\theta_{t-1})),$$

where the β_ts are step sizes. The regret analysis relies on the convexity of the episodic losses ℓ_t which yields the lower bound

$$\ell_t(\theta) \geqslant \ell_t(\theta_t) + \langle\nabla\ell_t(\theta_t), \theta - \theta_t\rangle.$$

The regret upperbound $O(BR\sqrt{n})$ is then obtained by setting the step-sizes as $\beta_t = B/(R\sqrt{t})$. In order to obtain a logarithmic regret, one must exploit the strong convexity of the logistic loss function. This was first done by the Online Newton Step [184] which makes predictions based on the parameter θ_t updated with the iteration

$$\theta_t = \mathrm{proj}_\Theta(A_{t-1}^{-1}b_{t-1}),$$

where proj_Θ is the projection operator onto the feasible set Θ and with

$$A_t = \sum_{s=1}^t \nabla\ell_s(\theta_s)\nabla\ell_s(\theta_s)^\top \quad \text{and} \quad b_t = \sum_{s=1}^t \nabla\ell_s(\theta_s)\nabla\ell_s(\theta_s)^\top\theta_s - \frac{1}{\beta}\nabla\ell_s(\theta_s).$$

The analysis of ONS relies on the exp-concavity property of the logistic loss which provides the quadratic lower bound

$$\ell_t(\theta) \geqslant \ell_t(\theta_t) + \langle\nabla\ell_t(\theta_t), \theta-\theta_t\rangle + \frac{\beta}{2}(\theta-\theta_t)^\top\nabla\ell_t(\theta_t)\nabla\ell_t(\theta_t)^\top(\theta-\theta_t),$$

with $\beta \leqslant \exp(-BR)$ the exp-concavity parameter. The latter is obtained using a uniform lower bound on $\sigma'(\theta^\top x_t)$ and is the cause for the exponential factor in the regret of ONS which is $O(de^{BR}\log(n))$. As mentioned earlier, this poor dependence can be sidestepped using an improper algorithm, a first example of which was proposed by [149] which makes predictions based on a Bayesian posterior (see Algorithm 6).

The analysis of [149] relies on the following *mixability* property.

Definition A.1. *The loss function $\ell : \mathcal{Z}\times\mathcal{Y}\to\mathbb{R}$ is said to be η-mixable if for any probability distribution π over $\mathcal{Z}$, there exists a mixed prediction z_π such that $\mathbb{E}_{z\sim\pi}[\exp(-\eta\ell(z,y))] \leqslant \exp(-\eta\ell(z_\pi, y))$.*

Algorithm 6 [149, Algorithm 1]

Inputs: feasible set Θ, mixability constant L and smoothing parameter $\mu \in [0, 1/2]$.
Initialize P_1 to be the uniform distribution over Θ.
for $t = 1, \dots, n$ **do**
    Receive x_t and predict $\widehat{y}_t = \sigma^+(\mathbb{E}_{\theta \sim P_t}[\sigma(\theta^\top x_t)])$ ▷ $\sigma^+(u) = \log(\frac{u}{1-u})$
    Receive y_t and define P_{t+1} as the distribution over Θ with density
        $P_{t+1}(\theta) \propto \exp\big(-\frac{1}{L}\sum_{s=1}^t \ell(\theta^\top x_s, y_s)\big)$.
end for

Using the fact that the logistic loss is 1-mixable, [149] show that the Bayesian procedure defined by Algorithm 6 achieves the optimal regret in $O(d\log(BRn))$. However, this comes at the cost of very high complexity since MCMC integration is necessary for the computation of the posteriors.

Another improper and computationally more efficient algorithm called AIOLI was proposed in [218] and achieves regret $O(dBR\log(BRn))$. The regret analysis of AIOLI uses a new quadratic lower bound on the episodic losses ℓ_t which writes

$$\ell_t(\theta) \geqslant \ell_t(\theta_t) + \nabla\ell_t(\theta)^\top(\theta - \theta_t) + \frac{\exp(y_t\theta_t^\top x_t)}{2(1+BR)}(\theta-\theta_t)^\top\nabla\ell_t(\theta_t)\nabla\ell_t(\theta_t)^\top(\theta-\theta_t). \qquad \text{(A.2.1)}$$

This bound improves upon the one obtained using the exp-concavity property by replacing the uniform factor $\beta \leqslant \exp(-BR)$ in front of the quadratic term by an adaptive one in terms of $y_t\theta_t^\top x_t$. AIOLI uses this lower bound in order to make predictions based on θ_t defined as the optimum of an approximate objective

$$\theta_t = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^d}\Big\{\widetilde{L}_{t-1}(\theta) + \ell(\theta^\top x_t, +1) + \ell(\theta^\top x_t, -1) + \lambda\|\theta\|^2\Big\},$$

where $\widetilde{L}_{t-1}(\theta) = \sum_{s=1}^{t-1}\widetilde{\ell}_s(\theta)$ and $\widetilde{\ell}_s(\theta)$ are the lower bounds of (A.2.1) with θ_s instead of θ_t. Thus AIOLI uses quadratic approximations of the episodic losses of history instances in addition to losses associated to both possible labels for x_t and Ridge regularization. Thanks to this definition, AIOLI can be implemented at a computational cost of only $O(nd^2 + n\log(n))$. However, the final regret has order $O(dBR\log(BRn))$ which remains suboptimal by a factor of BR.

By extending the quadratic lower bound (A.2.1) for multi-class logistic regression, the subsequent work of [4] proposed FOLKLORE, an improper and efficient algorithm for the multi-class case which achieves regret $O(dBRK\log(n))$ for K classes with computational cost in $O(n(d^2K^3 + BRK\log(n)))$.

Another algorithm for multi-class logistic regression with the same regret was obtained in [219]. The latter adopts a Bayesian approach similar to Algorithm 6 from [149] but replacing the objectives L_t with quadratic approximations based on an extension of (A.2.1) for the multi-class case. This results in a second algorithm achieving regret $O(dBRK\log(BRn))$ for K multi-class logistic regression and bringing the complexity of the original Bayesian procedure of [149] down to $O(nd^2K^3 + K^2n^4)$. We summarize the regrets and complexities of the algorithms we mentioned in Table A.1.

A.3 A more efficient candidate algorithm for optimal regret

The algorithm we present here is inspired from the Sample Minmax Predictor (SMP) presented in [326]. The latter provides a conditional density estimation procedure which may be used for

Algorithm	Regret	Complexity
OGD [460]	$BR\sqrt{n}$	nd
ONS [184]	$e^{BR}\log(n)$	nd^2
Foster et al. [149]	$d\log(BRn)$	$(BR)^6 n^{12}(BRn+d)^{12}$
AIOLI [218]	$dBR\log(BRn)$	$nd^2 + n\log(n)$
FOLKLORE [4]	$dBRK\log(n)$	$nd^2K^3 + nBRK\log(n)$
GAF [219]	$dBRK\log(n)$	$nd^2K^3 + K^2n^4$

Table A.1: Regret upper bounds and complexities of the main algorithms we discuss for online logistic regression (we remove the big-O notations to improve readability). Algorithms with bounds involving K can handle the K multi-class case. Note that some works originally state bounds in terms of B rather than BR by setting $R = 1$. We write all bounds in terms of BR for uniformity.

various models including logistic regression. As presented in [326], SMP is destined to the batch setting where a dataset of i.i.d samples is available based on which a predictor $f : \mathcal{X} \to \widehat{\mathcal{Y}}$ must be chosen in order to minimize the risk

$$\mathcal{R}(f) := \mathbb{E}[\ell_\phi(f(X), Y)] = \mathbb{E}[\ell(f(X), Y) + \phi(f)],$$

for some loss function $\ell : \widehat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}$ and penalty function ϕ. Equivalently, a predictor's performance can be measured through its excess risk

$$\mathcal{R}(f) - \inf_g \mathcal{R}(g).$$

Given a sample $(X_i, Y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ of i.i.d instances and a fixed *virtual sample* $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$, define

$$\widehat{f}^z_{\phi,n} := \arg\min_f \Big\{ \sum_{i=1}^n \ell_\phi(f(X_i), Y_i) + \ell_\phi(f(x), y) \Big\}. \tag{A.3.1}$$

SMP (when it exists) is then defined as follows

$$\widetilde{f}_{\phi,n}(x) := \arg\min_{\widehat{y} \in \widehat{\mathcal{Y}}} \sup_{y \in \mathcal{Y}} \Big\{ \ell(\widehat{y}, y) - \ell_\phi\big(\widehat{f}^{(x,y)}_{\phi,n}(x), y\big) \Big\}.$$

For the particular case of logistic regression with Ridge penalty, predictors are defined as $f_\theta(x) = \theta^\top x$ for $\theta \in \mathbb{R}^d$ and we have $\phi(f_\theta) = \lambda\|\theta\|^2$ for some $\lambda > 0$. Following [326, Section 5], this leads to the following particular case of (A.3.1)

$$\widehat{\theta}^z_{\lambda,n} = \arg\min_\theta \Big\{ \frac{1}{n+1} \Big(\sum_{i=1}^n \ell(\theta^\top X_i, Y_i) + \ell(\theta^\top x, y) \Big) + \lambda\|\theta\|^2 \Big\},$$

where $z = (x, y) \in \mathbb{R}^d \times \{\pm 1\}$ is a virtual sample and ℓ is the logistic loss as defined before. Given this definition, SMP computes $\widehat{y}_{\mathrm{SMP}}$ which predicts the conditional probability $p(y|x)$ as follows

$$\sigma(\widehat{y}_{\mathrm{SMP}}) = \frac{\sigma\big(y\langle \widehat{\theta}^{(x,y)}_{\lambda,n}, x\rangle\big) e^{-\lambda\|\widehat{\theta}^{(x,y)}_{\lambda,n}\|^2}}{\sigma\big(\langle \widehat{\theta}^{(x,+1)}_{\lambda,n}, x\rangle\big) e^{-\lambda\|\widehat{\theta}^{(x,+1)}_{\lambda,n}\|^2} + \sigma\big(-\langle \widehat{\theta}^{(x,-1)}_{\lambda,n}, x\rangle\big) e^{-\lambda\|\widehat{\theta}^{(x,-1)}_{\lambda,n}\|^2}}.$$

Under Assumptions A.1 and A.2 and setting $\lambda = 2R^2/(n+1)$, the above estimator has excess risk upperbounded by [326, Corollary 2]

$$\frac{ed + B^2R^2}{n}.$$

This suggests that an online version of SMP is likely to achieve a regret upperbound of order $O((d + B^2R^2)\log(n))$ which would match the performance of [149] since one can argue that $BR \lesssim \sqrt{d}$ (see [326, Remark 2]). Note that SMP is an improper and non Bayesian predictor which replaces the computation of a Bayesian posterior with the solution of two logistic regression problems so that it opens the door for a considerable complexity gain over [149] for the same asymptotic regret.

In order to define an online version of SMP, we consider the λ-Ridge penalized regret

$$\sum_{s=1}^{t} \ell(\widehat{y}_s, y_s) - \Big(\sum_{s=1}^{t} \ell(\theta^\top x_s, y_s) + \lambda\|\theta\|^2 \Big).$$

We define the prediction $\widehat{y}_t$ at round t as follows

$$\widehat{y}_t = \operatorname*{arg\,min}_{y\in\mathbb{R}} \sup_{y_t\in\{\pm 1\}} \sup_{\theta\in\Theta} \Big\{ \widehat{L}_{t-1} + \ell(y, y_t) - \Big(L_{t-1}(\theta) + \ell(\theta^\top x_t, y_t) + \lambda\|\theta\|^2 \Big) \Big\}, \tag{A.3.2}$$

where

$$\widehat{L}_t = \sum_{s=1}^{t} \ell(\widehat{y}_s, y_s) \quad \text{and} \quad L_t(\theta) = \sum_{s=1}^{t} \ell(\theta^\top x_s, y_s).$$

We refer to the resulting algorithm as OSMP (Online SMP). The computed predictions $\widehat{y}_t$ minimize the maximal possible instant regret. Equivalently, reparametrizing with $\widehat{p}_t = \sigma(\widehat{y}_t)$, we predict

$$\widehat{p}_t = \operatorname*{arg\,min}_{p\in[0,1]} \sup_{y_t\in\{\pm 1\}} \sup_{\theta\in\Theta} \Big\{ \widehat{L}_{t-1} - \log(p) - \Big(L_{t-1}(\theta) + \ell(\theta^\top x_t, y_t) + \lambda\|\theta\|^2 \Big) \Big\}.$$

In order to study this new algorithm, we introduce the following notations

$$L_{\lambda,t}(\theta) := L_t(\theta) + \lambda\|\theta\|^2 \quad \text{and} \quad L^\star_{\lambda,t} = \inf_{\theta\in\Theta} L_{\lambda,t}(\theta).$$

In addition, for $y = \pm 1$ let

$$L^y_{\lambda,t}(\theta) := L_{t-1}(\theta) + \ell(\theta^\top x_t, y) \quad \text{and} \quad L^{y\star}_{\lambda,t} = \inf_{\theta\in\Theta} L^y_{\lambda,t}(\theta).$$

We also define $\theta_t = \arg\min_{\theta\in\Theta} L_{\lambda,t}(\theta)$ and $\theta_t^{(x,y)} = \arg\min_{\theta\in\Theta} L_{t-1}(\theta) + \ell(\theta^\top x, y)$. The following statement characterizes the prediction (A.3.2) of OSMP.

Lemma A.1. *The definition* (A.3.2) *is equivalent to* $\widehat{y}_t = -L^{+1\star}_{\lambda,t} + L^{-1\star}_{\lambda,t}$.

Proof. We can remove $\widehat{L}_{t-1}$ from (A.3.2) since it does not depend on y_t and θ. In addition, we can maximize w.r.t y_t before θ in order to obtain

$$\widehat{y}_t = \operatorname*{arg\,min}_{y\in\mathbb{R}} \sup_{\theta\in\Theta} \Big\{ \sup_{y_t\in\{\pm 1\}} \{\ell(y, y_t) - \ell(\theta^\top x, y_t)\} - L_{\lambda,t-1}(\theta) \Big\}$$

$$
\begin{aligned}
&= \arg\min_{y\in\mathbb{R}} \sup_{\theta\in\Theta} \Big\{ \sup_{y_t\in\{\pm 1\}} \log \frac{\sigma(y_t\theta^\top x_t)}{\sigma(yy_t)} - L_{\lambda,t-1}(\theta) \Big\} \\
&= \arg\min_{y\in\mathbb{R}} \sup_{\theta\in\Theta} \Big\{ \max\Big\{ \log \frac{\sigma(\theta^\top x_t)}{\sigma(y)}, \log \frac{\sigma(-\theta^\top x_t)}{1-\sigma(y)} \Big\} - L_{\lambda,t-1}(\theta) \Big\} \\
&= \arg\min_{y\in\mathbb{R}} \sup_{\theta\in\Theta} \max\Big\{ \log \frac{\sigma(\theta^\top x_t)\exp(-L_{\lambda,t-1}(\theta))}{\sigma(y)}, \log \frac{\sigma(-\theta^\top x_t)\exp(-L_{\lambda,t-1}(\theta))}{1-\sigma(y)} \Big\} \\
&= \arg\min_{y\in\mathbb{R}} \max\Big\{ \frac{\sup_{\theta\in\Theta}\exp(-L^{-1}_{\lambda,t-1}(\theta))}{\sigma(y)}, \frac{\sup_{\theta\in\Theta}\exp(-L^{+1}_{\lambda,t}(\theta))}{1-\sigma(y)} \Big\} \\
&= \arg\min_{y\in\mathbb{R}} \max\Big\{ \frac{\exp(-L^{-1\star}_{\lambda,t-1})}{\sigma(y)}, \frac{\exp(-L^{+1\star}_{\lambda,t})}{1-\sigma(y)} \Big\}
\end{aligned}
$$

In order to solve the optimization in y, we change the variable into $p = \sigma(y)$ and use the following equality valid for all $a, b > 0$:

$$
\arg\min_{p\in[0,1]} \max\Big\{ \frac{a}{p}, \frac{b}{1-p} \Big\} = \frac{a}{a+b}.
$$

This yields

$$
\sigma(\widehat{y}) = \frac{\exp(-L^{+1\star}_{\lambda,t-1})}{\exp(-L^{-1\star}_{\lambda,t-1}) + \exp(-L^{+1\star}_{\lambda,t-1})}.
$$

It only remains to apply the inverse function of the sigmoid $\sigma^+(p) = \log(\frac{p}{1-p})$ to $\sigma(\widehat{y})$ in order to complete the proof. □

We now turn to the study of OSMP's performances.

A.4 Regret analysis

Recall that the regret is expressed as

$$
R_n(\theta) = \sum_{t=1}^n \ell(\widehat{y}_t, y_t) - \sum_{t=1}^n \ell(\theta^\top x_t, y_t).
$$

Since we have $\sum_{t=1}^n \ell(\widehat{y}_t, y_t) = L_{\lambda,n}(\theta) - \lambda\|\theta\|^2$, we can rewrite it as

$$
R_n(\theta) = \sum_{t=1}^n \Big(\ell(\widehat{y}_t, y_t) - L^\star_{\lambda,t} + L^\star_{\lambda,t-1} \Big) + \lambda\|\theta\|^2,
$$

where the last term can be directly bounded by λB^2, therefore, we can focus on bounding the instant regret which we define as

$$
\widehat{r}_t = \ell(\widehat{y}_t, y_t) - L^\star_{\lambda,t} + L^\star_{\lambda,t-1}.
$$

The following lemma, drawn from [326] will be necessary in order the derive a bound on $\widehat{r}_t$.

Lemma A.2 ([326, Lemma 4]). *Let Ω be a nonempty open convex subset of $\mathbb{R}^d$ and $f : \Omega \to \mathbb{R}$ a differentiable function. Assume that f is Σ-strongly convex on Ω (where $\Sigma \in \mathbb{R}^{d\times d}$ is a symmetric*

positive definite matrix) in the sense that, for all $x, x' \in \Omega$:

$$f(x') \geqslant f(x) + \langle \nabla f(x), x' - x \rangle + \frac{1}{2}\|x' - x\|_\Sigma.$$

Assume that f *reaches its minimum at* $x^\star \in \Omega$. *Let* $g \in \mathbb{R}^d$ *and assume that* $x \mapsto f(x) - \langle g, x \rangle$ *reaches its minimum at* $\widetilde{x} \in \Omega$, *then we have*

$$\|\widetilde{x} - x^\star\|_\Sigma \leqslant \|g\|_{\Sigma^{-1}} \quad \textit{and} \quad \langle g, \widetilde{x} - x^\star \rangle \leqslant \|g\|^2_{\Sigma^{-1}}.$$

In the following statement, we bound the instant regret $\widehat{r}_t$ by using some arguments from [326].

Proposition A.1. *For all* $t \geqslant 0$, *the instant regret* $\widehat{r}_t$ *is upperbounded as*

$$\widehat{r}_t \leqslant e \cdot \sigma'(\langle \theta_t, x_t \rangle) \cdot \|x_t\|^2_{\nabla^2 L_{\lambda,t}(\theta_t)^{-1}},$$

where $\nabla^2 L_{\lambda,t}(\theta_t) = \sum_{s=1}^t \sigma'(\langle \theta_t, x_s \rangle) x_s x_s^\top + 2\lambda I_d$ *is the Hessian matrix of* $L_{\lambda,t}$.

Proof. By plugging the expression of $\widehat{y}_t$ given by Lemma A.1 into the definition of $\widehat{r}_t$, we find that

$$\begin{aligned}
\widehat{r}_t &= \log(1 + \exp(-y_t(-L_{\lambda,t}^{+1\star} + L_{\lambda,t}^{-1\star}))) - L_{\lambda,t}^\star + L_{\lambda,t-1}^\star \\
&= \log(1 + \exp(L_{\lambda,t}^{y_t\star} - L_{\lambda,t}^{-y_t\star}))) - L_{\lambda,t}^\star + L_{\lambda,t-1}^\star \\
&= \log\big(\exp(-L_{\lambda,t}^{y_t\star} + L_{\lambda,t-1}^\star) + \exp(-L_{\lambda,t}^{-y_t\star} + L_{\lambda,t-1}^\star)\big) \\
&= \log\big(\exp(-L_{\lambda,t}^{+1\star} + L_{\lambda,t-1}^\star) + \exp(-L_{\lambda,t}^{-1\star} + L_{\lambda,t-1}^\star)\big) \qquad \text{(A.4.1)}
\end{aligned}$$

where the third step follows by noticing that $L_{\lambda,t}^{y_t} = L_{\lambda,t}$ and therefore $L_{\lambda,t}^{y_t\star} = L_{\lambda,t}^\star$. Now, recall that we defined

$$\theta_t = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} L_{\lambda,t}(\theta) = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} \ell(\theta^\top x_t, y_t) + L_{t-1}(\theta) + \lambda\|\theta\|^2.$$

Let $z_t = -y_t x_t$ and define

$$\begin{aligned}
\theta_t^{-y_t} &= \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} \ell(\theta^\top x_t, -y_t) + L_{t-1}(\theta) + \lambda\|\theta\|^2 \\
&= \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} L_t(\theta) - \langle \theta, z_t \rangle + \lambda\|\theta\|^2,
\end{aligned}$$

where we used the identity $\log(1 + e^u) = \log(e^u(1 + e^{-u})) = u + \log(1 + e^{-u})$. Continuing from (A.4.1) and using the inequalities $L_{\lambda,t-1}(\theta_t) \geqslant L_{\lambda,t-1}^\star$, $L_{\lambda,t-1}(\theta_t^{-y_t}) \geqslant L_{\lambda,t-1}^\star$ and $\log(1 + u) \leqslant u$ and the identity $\sigma(-u) = 1 - \sigma(u)$ we get

$$\begin{aligned}
\widehat{r}_t &= \log\big(\exp(-(L_{\lambda,t}^{+1\star} - L_{\lambda,t-1}^\star)) + \exp(-(L_{\lambda,t}^{-1\star} - L_{\lambda,t-1}^\star))\big) \\
&= \log\big(\exp(-\ell(\langle \theta_t, x_t \rangle, y_t) - (L_{\lambda,t-1}(\theta_t) - L_{\lambda,t-1}^\star)) \\
&\quad + \exp(-\ell(\langle \theta_t^{-y_t}, x_t \rangle, -y_t) - (L_{\lambda,t-1}(\theta_t^{-y_t}) - L_{\lambda,t-1}^\star))\big) \\
&= \log\big(\sigma(-\langle \theta_t, z_t \rangle) \exp(-(L_{\lambda,t-1}(\theta_t) - L_{\lambda,t-1}^\star)) \\
&\quad + \sigma(-\langle \theta_t^{-y_t}, z_t \rangle) \exp(-(L_{\lambda,t-1}(\theta_t^{-y_t}) - L_{\lambda,t-1}^\star))\big) \\
&\leqslant \log(1 + \sigma(\langle \theta_t^{-y_t}, z_t \rangle) - \sigma(\langle \theta_t, z_t \rangle)) \\
&\leqslant \sigma(\langle \theta_t^{-y_t}, z_t \rangle) - \sigma(\langle \theta_t, z_t \rangle). \qquad \text{(A.4.2)}
\end{aligned}$$

The rest of the proof roughly follows the steps from [326, Theorem 5]. Given that $L_{\lambda,t}^{-y_t}(\theta) = L_{\lambda,t}(\theta) - \langle \theta, z_t \rangle$ and knowing that the function $L_{\lambda,t}$ is 2λ-strongly convex and $\|x_t\| \leqslant R$ by Assumption A.1, it follows from Lemma A.2 that

$$\|\theta_t - \theta_t^{-y_t}\| \leqslant \frac{R}{2\lambda} \quad \text{and} \quad 0 \leqslant \langle \theta_t - \theta_t^{-y_t}, z_t \rangle \leqslant \frac{R^2}{2\lambda} \leqslant 1/2$$

because $\lambda \geqslant R^2$. Next since $\log(\sigma')' = \sigma''/\sigma' = 1 - 2\sigma < 1$, we have for all $u \in \mathbb{R}$ and $v \in [0, 1/2]$ that $\log(\sigma'(u+v)) - \log(\sigma'(u)) \leqslant v$ hence $\sigma'(u+v) \leqslant e^v \sigma'(u) \leqslant e^{1/2}\sigma'(u)$. Hence we deduce the inequality $\sigma(u+v) - \sigma(u) \leqslant e^{1/2}\sigma'(u)v$ for all $u \in \mathbb{R}$ and $v \in [0, 1/2]$. In particular, for the choice $u = \langle \theta_t, z_t \rangle$ and $v = \langle \theta_t^{-y_t} - \theta_t, z_t \rangle$ we get

$$\sigma(\langle \theta_t^{-y_t}, z_t \rangle) - \sigma(\langle \theta_t, z_t \rangle) \leqslant e^{1/2} \cdot \sigma'(\langle \theta_t, z_t \rangle) \cdot \langle \theta_t^{-y_t} - \theta_t, z_t \rangle. \tag{A.4.3}$$

Next, we use the generalized self-concordance property of the function $L_{\lambda,t}$ with constant R to find that for all $\theta, \beta \in \mathbb{R}^d$ we have

$$\nabla^2 L_{\lambda,t}(\theta + \beta) \succeq e^{-R\|\beta\|} \cdot \nabla^2 L_{\lambda,t}(\theta).$$

Let $\varepsilon > 0$ and $\Omega_\varepsilon = \{\theta' \in \mathbb{R}^d : \|\theta' - \theta_t\| \leqslant 1/2 + \varepsilon\}$ be an open ball around θ_t and take $\theta = \theta_t$ and $\beta = \theta' - \theta_t$ in the above inequality, we have that $L_{\lambda,t}^{-y_t}(\theta) = L_{\lambda,t}(\theta) - \langle \theta, z_t \rangle$ reaches its minimum at $\theta^{-y_t} \in \Omega_\varepsilon$. Using Lemma A.2 again, we get

$$\langle \theta_t^{-y_t} - \theta_t, z_t \rangle \leqslant e^{1/2+\varepsilon} \cdot \|x_t\|^2_{\nabla^2 L_{\lambda,t}(\theta_t)^{-1}}.$$

and taking the limit $\varepsilon \to 0$ we find

$$\langle \theta_t^{-y_t} - \theta_t, z_t \rangle \leqslant e^{1/2} \cdot \|x_t\|^2_{\nabla^2 L_{\lambda,t}(\theta_t)^{-1}}.$$

Plugging back into (A.4.3), we finally find

$$\begin{aligned}
\sigma(\langle \theta_t^{-y_t}, z_t \rangle) - \sigma(\langle \theta_t, z_t \rangle) &\leqslant e \cdot \sigma'(\langle \theta_t, z_t \rangle) \cdot \|x_t\|^2_{\nabla^2 L_{\lambda,t}(\theta_t)^{-1}} \\
&= e \cdot \sigma'(\langle \theta_t, z_t \rangle) \cdot \left\langle \nabla^2 L_{\lambda,t}(\theta_t)^{-1} x_t, x_t \right\rangle \\
&= e \cdot \sigma'(\langle \theta_t, z_t \rangle) \cdot \operatorname{Tr}\left(\nabla^2 L_{\lambda,t}(\theta_t)^{-1} x_t x_t^\top\right).
\end{aligned}$$

It only remain to plug the last inequality into (A.4.2) and notice that σ' is an even function to finish the proof. □

Unfortunately, it is unclear how to continue the regret analysis beyond Proposition A.1. The classical approach uses the fact that the Hessian matrix of the loss is constituted as a sum of rank 1 terms in order to bound the instant regret with a telescopic term. This method was applied for the Vovk-Azoury-Warmuth foreacaster [436, 15] and later in [218, 456] and uses the following identity from [77, Lemma 11.11].

Lemma A.3. *Let $B \in \mathbb{R}^{d\times d}$ be a full-rank matrix and $x \in \mathbb{R}^d$. Let $A = B + xx^\top$, we have*

$$x^\top A^{-1} x = 1 - \frac{\det B}{\det A}.$$

Thus, given an instant regret upperbound of the form

$$\widehat{r}_t \leqslant \|x_t\|^2_{A_t^{-1}} = \langle x_t, A_t^{-1} x_t \rangle \quad \text{with} \quad A_t = \sum_{s=1}^{t} x_s x_s^\top + \lambda I_d,$$

Lemma A.3 and the inequality $1 - x \leqslant -\log x$ allow to compute

$$\sum_{t=1}^{n} \widehat{r}_t \leqslant \sum_{t=1}^{n} \Big(1 - \frac{\det A_{t-1}}{\det A_t}\Big) \leqslant \sum_{t=1}^{n} \log \frac{\det A_t}{\det A_{t-1}} = \log\det A_n - \log\det A_0,$$

which yields logarithmic regret due to the definition of A_t. However, in the case of OSMP, the Hessian matrices $\nabla^2 L_{\lambda,t}(\theta_t) = 2\lambda I_d + \sum_{s=1}^{t} \sigma'(\langle \theta_t, z_t \rangle) x_s x_s^\top$ depend on the parameter θ_t and the previous computation, started from the bound of Proposition A.1, leads to the following non telescopable terms

$$\sigma'(\langle \theta_t, x_t \rangle) \cdot \|x_t\|^2_{\nabla^2 L_{\lambda,t}(\theta_t)^{-1}} \leqslant 1 - \frac{\det \nabla^2 L_{\lambda,t-1}(\theta_t)}{\det \nabla^2 L_{\lambda,t}(\theta_t)} \leqslant \log \frac{\det \nabla^2 L_{\lambda,t}(\theta_t)}{\det \nabla^2 L_{\lambda,t-1}(\theta_t)}.$$

This problem does not emerge in the analysis of [326] because they bound the risk which is an expectation over a batch of interchangeable i.i.d samples. Conversely, the online setting requires to deal with an arbitrary sequence of instances satisfying only Assumption A.1. Note that it is possible to handle the dependence of the matrices $\nabla^2 L_{\lambda,t}(\theta_t)$ in θ_t by using uniform bounds on the $\sigma'(\cdot)$ factors. However, this causes an unwanted exponential factor $\exp(BR)$ to appear. Another solution is proposed by [218] which replaces the true Hessian matrices by fixed quadratic approximations thanks to the lower bound (A.2.1) but this leads to a suboptimal final bound with an excess factor of BR.

A.5 Discussion

Based on the known properties of SMP, the online predictions defined by OSMP are likely to achieve regret similar to [149] improving over the results presented in [218, 4, 219] while avoiding the prohibitive computational cost of a Bayesian method. It is clear that, in order to achieve a logarithmic rate, the regret analysis must leverage the curvature of the objective $L_{\lambda,t}$ induced by the strong convexity of the loss function ℓ. Our attempt for the analysis of OSMP focuses on this aspect by deriving a bound in terms of the Hessian matrix $\nabla^2 L_{\lambda,t}$. A similar scheme is followed for a least squares linear regression problem in the classical works of [436, 15]. The difficulty our analysis is confronted with seems to come from the lack of closed form expressions for the optima $\theta_t = \arg\min_\theta L_{\lambda,t}(\theta)$ when ℓ is the logistic loss whereas this is the case for the least squares loss. This very difficulty is bypassed in [218, 4, 219] using highly precise quadratic lower bounds of the logistic loss but sacrificing a factor BR in the final regret. A further investigation of the properties of the logistic function may be key to the derivation of a tight regret bound for OSMP.

Bibliography

[1] A. Acharya, A. Hashemi, P. Jain, S. Sanghavi, I. S. Dhillon, and U. Topcu. "Robust training in high dimensions via block coordinate geometric median descent". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 11145–11168.

[2] A. Agarwal, Y. S. Tan, O. Ronen, C. Singh, and B. Yu. "Hierarchical Shrinkage: Improving the accuracy and interpretability of tree-based models." In: *International Conference on Machine Learning*. PMLR. 2022, pp. 111–135.

[3] A. Agarwal, S. Negahban, and M. J. Wainwright. "Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions". In: *Advances in Neural Information Processing Systems* 25 (2012).

[4] N. Agarwal, S. Kale, and J. Zimmert. "Efficient Methods for Online Multiclass Logistic Regression". In: *International Conference on Algorithmic Learning Theory*. PMLR. 2022, pp. 3–33.

[5] K. Alghatani, N. Ammar, A. Rezgui, A. Shaban-Nejad, et al. "Predicting Intensive Care Unit Length of Stay and Mortality Using Patient Vital Signs: Machine Learning Model Development and Validation". In: *JMIR Medical Informatics* 9.5 (2021), e21347.

[6] N. Alon, Y. Matias, and M. Szegedy. "The space complexity of approximating the frequency moments". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. 1996, pp. 20–29.

[7] N. Alon, Y. Matias, and M. Szegedy. "The space complexity of approximating the frequency moments". In: *Journal of Computer and system sciences* 58.1 (1999), pp. 137–147.

[8] F. J. Anscombe. "Rejection of outliers". In: *Technometrics* 2.2 (1960), pp. 123–146.

[9] S. Arlot and R. Genuer. "Analysis of purely random forests bias". In: *arXiv preprint arXiv:1407.3939* (2014).

[10] L. Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives." In: *Pacific Journal of Mathematics* 16.1 (1966), pp. 1–3.

[11] S. Athey and G. Imbens. "Recursive partitioning for heterogeneous causal effects". In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7353–7360.

[12] S. Athey, J. Tibshirani, and S. Wager. *Generalized Random Forests*. 2018. arXiv: `1610.01271 [stat.ME]`.

[13] J.-Y. Audibert and O. Catoni. "Robust linear least squares regression". In: *The Annals of Statistics* 39.5 (2011), pp. 2766–2794.

[14] J.-Y. Audibert, R. Munos, and C. Szepesvári. "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits". In: *Theoretical Computer Science* 410.19 (2009). Algorithmic Learning Theory, pp. 1876–1902.

[15] K. S. Azoury and M. K. Warmuth. "Relative loss bounds for on-line density estimation with the exponential family of distributions". In: *Machine learning* 43 (2001), pp. 211–246.

[16] F. Bach. "Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 595–627.

[17] F. Bach and E. Moulines. "Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$". In: *Advances in Neural Information Processing Systems* 26 (2013).

[18] A. Bakshi and A. Prasad. "Robust linear regression: Optimal rates in polynomial time". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 102–115.

[19] S. Balakrishnan, S. S. Du, J. Li, and A. Singh. "Computationally efficient robust sparse estimation in high dimensions". In: *Conference on Learning Theory*. PMLR. 2017, pp. 169–212.

[20] R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola. "Sobol tensor trains for global sensitivity analysis". In: *Reliability Engineering & System Safety* 183 (2019), pp. 311–322.

[21] F. Barthe and C. Roberto. "Modified logarithmic Sobolev inequalities on". In: *Potential Analysis* 29.2 (2008), p. 167.

[22] P. L. Bartlett, O. Bousquet, and S. Mendelson. "Local rademacher complexities". In: *The Annals of Statistics* 33.4 (2005), pp. 1497–1537.

[23] P. H. Baxendale. "Renewal theory and computable convergence rates for geometrically ergodic Markov chains". In: *The Annals of Applied Probability* 15.1B (2005), pp. 700–738.

[24] A. Beck and M. Teboulle. "Mirror descent and nonlinear projected subgradient methods for convex optimization". In: *Operations Research Letters* 31.3 (2003), pp. 167–175.

[25] A. Beck and L. Tetruashvili. "On the Convergence of Block Coordinate Descent Type Methods". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2037–2060. eprint: `https://doi.org/10.1137/120887679`.

[26] W. Bednorz. "The Kendall's Theorem and its Application to the Geometric Ergodicity of Markov Chains". In: *arXiv preprint arXiv:1301.1481* (2013).

[27] P. C. Bellec, G. Lecué, and A. B. Tsybakov. "Slope meets lasso: improved oracle bounds and optimality". In: *The Annals of Statistics* 46.6B (2018), pp. 3603–3642.

[28] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Vol. 22. Springer Science & Business Media, 2012.

[29] K. S. Berenhaut and R. Lund. "Geometric renewal convergence rates from hazard rates". In: *Journal of Applied Probability* 38.1 (2001), pp. 180–194.

[30] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. "Hyperopt: a python library for model selection and hyperparameter optimization". In: *Computational Science & Discovery* 8.1 (2015), p. 014008.

[31] K. Bhatia, P. Jain, P. Kamalaruban, and P. Kar. "Consistent robust regression". In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 2110–2119.

[32] G. Biau. "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 1063–1095.

[33] G. Biau, L. Devroye, and G. Lugosi. "Consistency of random forests and other averaging classifiers". In: *Journal of Machine Learning Research* 9 (2008), pp. 2015–2033.

[34] G. Biau and E. Scornet. "A random forest guided tour". In: *TEST* 25.2 (2016), pp. 197–227.

[35] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. "Simultaneous analysis of Lasso and Dantzig selector". In: *The Annals of Statistics* 37.4 (2009), pp. 1705–1732.

[36] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[37] J. K. Blitzstein and J. Hwang. *Introduction to probability*. Crc Press, 2019.

[38] D. A. Bloch. "A Note on the Estimation of the Location Parameter of the Cauchy Distribution". In: *Journal of the American Statistical Association* 61 (1966), pp. 852–855.

[39] M. Blondel, K. Seki, and K. Uehara. "Block coordinate descent algorithms for large-scale sparse multiclass classification". In: *Machine learning* 93.1 (2013), pp. 31–52.

[40] T. Blumensath and M. E. Davies. "Iterative hard thresholding for compressed sensing". In: *Applied and Computational Harmonic Analysis* 27.3 (2009), pp. 265–274.

[41] T. Blumensath and M. E. Davies. "Normalized iterative hard thresholding: Guaranteed stability and performance". In: *IEEE Journal of Selected Topics in Signal Processing* 4.2 (2010), pp. 298–309.

[42] S. G. Bobkov and F. Götze. "Exponential integrability and transportation cost related to logarithmic Sobolev inequalities". In: *Journal of Functional Analysis* 163.1 (1999), pp. 1–28.

[43] S. Bobkov and M. Ledoux. "Poincaré's inequalities and Talagrand's concentration phenomenon for the exponential distribution". In: *Probability Theory and Related Fields* 107 (1997), pp. 383–400.

[44] M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès. "SLOPE—adaptive variable selection via convex optimization". In: *The Annals of Applied Statistics* 9.3 (2015), p. 1103.

[45] H. D. Bondell and B. J. Reich. "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR". In: *Biometrics* 64.1 (2008), pp. 115–123.

[46] L. Bottou and O. Bousquet. "The tradeoffs of large scale learning". In: *Advances in Neural Information Processing Systems* 20 (2007).

[47] L. Bottou and Y. Cun. "Large Scale Online Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press, 2003.

[48] L. Bottou, F. E. Curtis, and J. Nocedal. "Optimization methods for large-scale machine learning". In: *SIAM review* 60.2 (2018), pp. 223–311.

[49] L. Bottou and Y. Lecun. "On-line learning for very large data sets". In: *Applied Stochastic Models in Business and Industry* 21 (Mar. 2005), pp. 137–151.

[50] S. Boucheron, G. Lugosi, P. Massart, and M. Ledoux. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford: Oxford University Press, 2013.

[51] L. Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[52] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[53] L. Breiman. *Some infinity theory for predictor ensembles*. Tech. rep. 577. Statistics departement, University of California Berkeley, 2000.

[54] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. The Wadsworth statistics/probability series. Monterey, CA: CRC, 1984.

[55] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[56] L. A. Breslow and D. W. Aha. "Simplifying decision trees: A survey". In: *The Knowledge Engineering Review* 12.01 (1997), pp. 1–40.

[57] C. Brownlees, E. Joly, G. Lugosi, et al. "Empirical risk minimization for heavy-tailed losses". In: *The Annals of Statistics* 43.6 (2015), pp. 2507–2536.

[58] S. Bubeck. "Convex optimization: Algorithms and complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[59] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. "Bandits with heavy tail". In: *IEEE Transactions on Information Theory* 59.11 (2013), pp. 7711–7717.

[60] P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.

[61] F. Bunea, A. Tsybakov, and M. Wegkamp. "Sparsity oracle inequalities for the Lasso". In: *Electronic Journal of Statistics* 1 (2007), pp. 169–194.

[62] W. Buntine. "Learning classification trees". In: *Statistics and Computing* 2.2 (1992), pp. 63–73.

[63] A. Calviño. "On Random-Forest-Based Prediction Intervals". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2020*. Ed. by C. Analide, P. Novais, D. Camacho, and H. Yin. Cham: Springer International Publishing, 2020, pp. 172–184.

[64] L. M. Candanedo and V. Feldheim. "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models". In: *Energy and Buildings* 112 (2016), pp. 28–39.

[65] L. M. Candanedo, V. Feldheim, and D. Deramaix. "Data driven prediction models of energy use of appliances in a low-energy house". In: *Energy and Buildings* 140 (2017), pp. 81–97.

[66] E. Candès and T. Tao. "The Dantzig selector: Statistical estimation when p is much larger than n". In: *The Annals of Statistics* 35.6 (2007), pp. 2313–2351.

[67] E. J. Candès and Y. Plan. "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements". In: *IEEE Transactions on Information Theory* 57.4 (2011), pp. 2342–2359.

[68] E. J. Candès and B. Recht. "Exact matrix completion via convex optimization". In: *Foundations of Computational mathematics* 9.6 (2009), pp. 717–772.

[69] E. J. Candès, X. Li, Y. Ma, and J. Wright. "Robust principal component analysis?" In: *Journal of the ACM (JACM)* 58.3 (2011), pp. 1–37.

[70] H. Cardot, P. Cénac, and A. Godichon-Baggioni. "Online estimation of the geometric median in Hilbert spaces: Nonasymptotic confidence balls". In: *The Annals of Statistics* 45.2 (2017), pp. 591–614.

[71] H. Cardot, P. Cénac, and P.-A. Zitt. "Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm". In: *Bernoulli* 19.1 (2013), pp. 18–43.

[72] I. Castillo, J. Schmidt-Hieber, and A. van der Vaart. "Bayesian linear regression with sparse priors". In: *The Annals of Statistics* 43.5 (2015), pp. 1986–2018.

[73] O. Catoni. "Challenging the empirical mean and empirical variance: a deviation study". In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 48. Institut Henri Poincaré. 2012, pp. 1148–1185.

[74] O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*. Vol. 56. IMS Lecture Notes Monograph Series. Institute of Mathematical Statistics, 2007.

[75] O. Catoni. *Statistical Learning Theory and Stochastic Optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI - 2001*. Vol. 1851. Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg, 2004.

[76] O. Catoni and I. Giulini. "Dimension-free PAC-Bayesian bounds for the estimation of the mean of a random vector". In: *arXiv preprint arXiv:1802.04308* (2018).

[77] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[78] N. Cesa-Bianchi, A. Conconi, and C. Gentile. "On the generalization ability of on-line learning algorithms". In: *IEEE Transactions on Information Theory* 50.9 (Sept. 2004), pp. 2050–2057.

[79] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge, New York, USA: Cambridge University Press, 2006.

[80] K. Chandra, A. Xie, J. Ragan-Kelley, and E. Meijer. "Gradient Descent: The Ultimate Optimizer". In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022.

[81] M. Charikar, J. Steinhardt, and G. Valiant. "Learning from untrusted data". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017, pp. 47–60.

[82] H.-f. Chen and A.-J. Gao. "Robustness analysis for stochastic approximation algorithms". In: *Stochastics and Stochastic Reports* 26.1 (1989), pp. 3–20.

[83] H.-F. Chen, L. Guo, and A.-J. Gao. "Convergence and robustness of the Robbins-Monro algorithm truncated at randomly varying bounds". In: *Stochastic Processes and their Applications* 27 (1987), pp. 217–231.

[84] M. Chen, C. Gao, and Z. Ren. "Robust covariance and scatter matrix estimation under Huber's contamination model". In: *The Annals of Statistics* 46.5 (2018), pp. 1932–1960.

[85] M. Chen, C. Gao, and Z. Ren. "Robust covariance and scatter matrix estimation under Huber's contamination model". In: *The Annals of Statistics* 46.5 (2018), pp. 1932–1960.

[86] P. Chen, X. Jin, X. Li, and L. Xu. "A generalized Catoni's M-estimator under finite α-th moment assumption with $\alpha \in (1, 2)$". In: *Electronic Journal of Statistics* 15.2 (2021), pp. 5523–5544.

[87] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794.

[88] Y. Chen, C. Caramanis, and S. Mannor. "Robust sparse regression under adversarial corruption". In: *Proceedings of the 30th International Conference on Machine Learning*. PMLR. 2013, pp. 774–782.

[89] Y. Chen, L. Su, and J. Xu. "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1.2 (2017), pp. 1–25.

[90] Z. Chen, L. Zhou, and W. Yu. *ADASYN-Random Forest Based Intrusion Detection Model*. 2021. arXiv: `2105.04301 [cs.CR]`.

[91] Y. Cherapanamjeri, E. Aras, N. Tripuraneni, M. I. Jordan, N. Flammarion, and P. L. Bartlett. "Optimal robust linear regression in nearly linear time". In: *arXiv preprint arXiv:2007.08137* (2020).

[92] Y. Cherapanamjeri, N. Flammarion, and P. L. Bartlett. "Fast mean estimation with sub-Gaussian rates". In: *Conference on Learning Theory*. PMLR. 2019, pp. 786–806.

[93] H. A. Chipman, E. I. George, and R. E. McCulloch. "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1 (2010), pp. 266–298.

[94] H. A. Chipman, E. I. George, and R. E. McCulloch. "Bayesian CART Model Search". In: *Journal of the American Statistical Association* 93.443 (1998), pp. 935–948.

[95] F. Chung and L. Lu. "Concentration inequalities and martingale inequalities: a survey". In: *Internet Mathematics* 3.1 (2006), pp. 79–127.

[96] K. L. Chung. "On a stochastic approximation method". In: *The Annals of Mathematical Statistics* (1954), pp. 463–483.

[97] D. Coppersmith, S. Hong, and J. Hosking. "Partitioning Nominal Attributes in Decision Trees". In: *Data Mining and Knowledge Discovery* 3 (Jan. 1999), pp. 197–217.

[98] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, 2009.

[99] A. Criminisi, J. Shotton, and E. Konukoglu. "Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning". In: *Found. Trends Comput. Graph. Vis.* 7 (2012), pp. 81–227.

[100] A. d'Aspremont, L. Ghaoui, M. Jordan, and G. Lanckriet. "A direct formulation for sparse PCA using semidefinite programming". In: *Advances in Neural Information Processing Systems* 17 (2004).

[101] A. Dalalyan and P. Thompson. "Outlier-robust estimation of a sparse linear model using ℓ_1-penalized Huber's M-estimator". In: *Advances in Neural Information Processing Systems* 32 (2019).

[102] A. Dalalyan and A. B. Tsybakov. "Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity". In: *Machine Learning* 72.1 (Aug. 2008), pp. 39–61.

[103] A. Defazio, F. Bach, and S. Lacoste-Julien. "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives". In: *Advances in Neural Information Processing Systems* 27 (2014).

[104] A. Défossez and F. Bach. "Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions". In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 205–213.

[105] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. "A Bayesian CART algorithm". In: *Biometrika* 85.2 (1998), pp. 363–377.

[106] J. Depersin and G. Lecué. "Optimal robust mean and location estimation via convex programs with respect to any pseudo-norms". In: *Probability Theory and Related Fields* 183.3-4 (2022), pp. 997–1025.

[107] J. Depersin and G. Lecué. "Robust sub-Gaussian estimation of a mean vector in nearly linear time". In: *The Annals of Statistics* 50.1 (2022), pp. 511–536.

[108] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (2019), pp. 4171–4186.

[109] L. Devroye and L. Györfi. *Nonparametric Density Estimation: The L1 View*. Wiley Interscience Series in Discrete Mathematics. Wiley, 1985.

[110] L. Devroye, M. Lerasle, G. Lugosi, and R. I. Oliveira. "Sub-Gaussian mean estimators". In: *The Annals of Statistics* 44.6 (2016), pp. 2695–2725.

[111] L. Devroye, M. Lerasle, G. Lugosi, and R. I. Oliveira. "Sub-Gaussian mean estimators". In: *The Annals of Statistics* 44.6 (2016), pp. 2695–2725.

[112] P. Diaconis and L. Saloff-Coste. "Comparison theorems for reversible Markov chains". In: *The Annals of Applied Probability* 3.3 (1993), pp. 696–730.

[113] P. Diaconis and D. Stroock. "Geometric Bounds for Eigenvalues of Markov Chains". In: *The Annals of Applied Probability* 1.1 (1991), pp. 36–61.

[114] I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. "Robust estimators in high-dimensions without the computational intractability". In: *SIAM Journal on Computing* 48.2 (2019), pp. 742–864.

[115] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and A. Stewart. "Sever: A robust meta-algorithm for stochastic optimization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1596–1606.

[116] I. Diakonikolas and D. M. Kane. "Recent advances in algorithmic high-dimensional robust statistics". In: *arXiv preprint arXiv:1911.05911* (2019).

[117] I. Diakonikolas, D. M. Kane, and A. Pensia. "Outlier robust mean estimation with subgaussian rates via stability". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1830–1840.

[118] I. Diakonikolas, D. M. Kane, A. Pensia, and T. Pittas. "Streaming algorithms for high-dimensional robust statistics". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 5061–5117.

[119] I. Diakonikolas, D. M. Kane, and A. Stewart. "Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 73–84.

[120] I. Diakonikolas, W. Kong, and A. Stewart. "Efficient algorithms and lower bounds for robust linear regression". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2745–2754.

[121] A. Dieuleveut and F. Bach. "Nonparametric stochastic approximation with large step-sizes". In: (2016).

[122] A. Dieuleveut, A. Durmus, and F. Bach. "Bridging the gap between constant step size stochastic gradient descent and Markov chains". In: *The Annals of Statistics* 48.3 (2020), pp. 1348–1382.

[123] A. Dieuleveut, A. Durmus, and F. Bach. "Bridging the gap between constant step size stochastic gradient descent and Markov chains". In: *The Annals of Statistics* 48.3 (2020), pp. 1348–1382.

[124] A. Dieuleveut, N. Flammarion, and F. Bach. "Harder, better, faster, stronger convergence rates for least-squares regression". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3520–3570.

[125] W. J. Dixon. "Analysis of extreme values". In: *The Annals of Mathematical Statistics* 21.4 (1950), pp. 488–506.

[126] M. Domb, E. Bonchek-Dokow, and G. Leshem. "Lightweight adaptive Random-Forest for IoT rule generation and execution". In: *Journal of Information Security and Applications* 34 (2017), pp. 218–224.

[127] D. L. Donoho et al. "High-dimensional data analysis: The curses and blessings of dimensionality". In: *AMS Math Challenges Lecture* 1.2000 (2000), p. 32.

[128] D. L. Donoho. "Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data". In: *In Proceedings of Symposia in Applied Mathematics.* Citeseer. 1993.

[129] D. L. Donoho and P. J. Huber. "The notion of breakdown point". In: *A festschrift for Erich L. Lehmann* (1983), pp. 157–184.

[130] D. L. Donoho and J. M. Johnstone. "Ideal spatial adaptation by wavelet shrinkage". In: *biometrika* 81.3 (1994), pp. 425–455.

[131] D. L. Donoho and M. Gasko. "Breakdown Properties of Location Estimates Based on Halfspace Depth and Projected Outlyingness". In: *The Annals of Statistics* 20.4 (1992), pp. 1803–1827.

[132] D. L. Donoho and R. C. Liu. "The "Automatic" Robustness of Minimum Distance Functionals". In: *The Annals of Statistics* 16.2 (1988), pp. 552–586.

[133] M. D. Donsker and S. S. Varadhan. "Asymptotic evaluation of certain Markov process expectations for large time—III". In: *Communications on pure and applied Mathematics* 29.4 (1976), pp. 389–461.

[134] A. V. Dorogush, V. Ershov, and A. Gulin. "CatBoost: gradient boosting with categorical features support". In: *arXiv preprint arXiv:1810.11363* (2018).

[135] R. Douc, E. Moulines, and J. S. Rosenthal. "Quantitative bounds on convergence of time-inhomogeneous Markov chains". In: *The Annals of Applied Probability* 14.4 (2004), pp. 1643–1665.

[136] D. Dua and C. Graff. *UCI Machine Learning Repository.* 2017.

[137] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press, 2009.

[138] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. "Composite objective mirror descent." In: *Conference on Learning Theory.* Vol. 10. Citeseer. 2010, pp. 14–26.

[139] F. Y. Edgeworth. "On observations relating to several quantities". In: *Hermathena* 6.13 (1887), pp. 279–285.

[140] F. Y. Edgeworth. "XXII. On a new method of reducing observations relating to several quantities". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 25.154 (1888), pp. 184–191.

[141] B. Efron and R. Tibshirani. "Improvements on cross-validation: the 632+ bootstrap method". In: *Journal of the American Statistical Association* 92.438 (1997), pp. 548–560.

[142] C. Eisenhart. "Boscovich and the combination of observations". In: *Roger Joseph Boscovich* (1961), pp. 200–212.

[143] T. van Erven, S. Sachs, W. M. Koolen, and W. Kotlowski. "Robust Online Convex Optimization in the Presence of Outliers". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Vol. 134. PMLR, 2021, pp. 4174–4194.

[144] V. Fabian. "On asymptotic normality in stochastic approximation". In: *The Annals of Mathematical Statistics* (1968), pp. 1327–1332.

[145] J. Fan, W. Wang, and Z. Zhu. "A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery". In: *The Annals of statistics* 49.3 (2021), p. 1239.

[146] H. Fanaee-T and J. Gama. "Event labeling combining ensemble detectors and background knowledge". In: *Progress in Artificial Intelligence* 2.2 (2014), pp. 113–127.

[147] P. Filzmoser and K. Nordhausen. "Robust linear regression for high-dimensional data: An overview". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 13.4 (2021), e1524.

[148] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), pp. 381–395.

[149] D. J. Foster, S. Kale, H. Luo, M. Mohri, and K. Sridharan. "Logistic regression: The importance of being improper". In: *Conference On Learning Theory*. PMLR. 2018, pp. 167–208.

[150] D. A. Freedman. "On tail probabilities for martingales". In: *the Annals of Probability* (1975), pp. 100–118.

[151] J. Friedman, T. Hastie, and R. Tibshirani. "A note on the group lasso and a sparse group lasso". In: *arXiv preprint arXiv:1001.0736* (2010).

[152] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics, New York, 2001.

[153] S. Gaïffas and I. Merad. "Robust supervised learning with coordinate gradient descent". In: *arXiv preprint arXiv:2201.13372* (2022).

[154] S. Gaïffas, I. Merad, and Y. Yu. "WildWood: a new Random Forest algorithm". In: *IEEE Transactions on Information Theory* (2023), pp. 1–1.

[155] C. Gao et al. "Robust regression via mutivariate regression depth". In: *Bernoulli* 26.2 (2020), pp. 1139–1170.

[156] J. Gaudio, S. Amin, and P. Jaillet. "Exponential convergence rates for stochastically ordered Markov processes under perturbation". In: *Systems & Control Letters* 133 (2019), p. 104515.

[157] S. A. Geer and S. van de Geer. *Empirical Processes in M-estimation*. Vol. 6. Cambridge university press, 2000.

[158] S. A. van de Geer and P. Bühlmann. "On the conditions used to prove oracle results for the Lasso". In: *Electronic Journal of Statistics* 3.none (2009), pp. 1360–1392.

[159] A. Genkin, D. D. Lewis, and D. Madigan. "Large-scale Bayesian logistic regression for text categorization". In: *Technometrics* 49.3 (2007), pp. 291–304.

[160] I. Gentil, A. Guillin, and L. Miclo. "Modified logarithmic Sobolev inequalities and transportation inequalities". In: *Probability Theory and Related Fields* 133 (2005), pp. 409–436.

[161] R. Genuer. "Variance reduction in purely random forests". In: *Journal of Nonparametric Statistics* 24.3 (2012), pp. 543–562.

[162] C. Geoffrey, L. Guillaume, and L. Matthieu. "Robust high dimensional learning for Lipschitz and convex losses". In: *Journal of Machine Learning Research* 21 (2020).

[163] P. Geurts, D. Ernst, and L. Wehenkel. "Extremely randomized trees". In: *Machine learning* 63.1 (2006), pp. 3–42.

[164] S. Ghadimi and G. Lan. "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2061–2089.

[165] S. Ghadimi and G. Lan. "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2341–2368.

[166] E. Gorbunov, M. Danilova, and A. Gasnikov. "Stochastic optimization with heavy-tailed noise via accelerated gradient clipping". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15042–15053.

[167] N. Gozlan. "A characterization of dimension free concentration in terms of transportation inequalities". In: (2009).

[168] N. Gozlan and C. Léonard. "Transport inequalities. A survey". In: *arXiv preprint arXiv:1003.3852* (2010).

[169] F. E. Grubbs. "Procedures for detecting outlying observations in samples". In: *Technometrics* 11.1 (1969), pp. 1–21.

[170] A. Gupta and S. Kohli. "An MCDM approach towards handling outliers in web data: a case study using OWA operators". In: *Artificial Intelligence Review* 46 (2016), pp. 59–82.

[171] A. Gupta and S. Kohli. "An MCDM approach towards handling outliers in web data: a case study using OWA operators". In: *Artificial Intelligence Review* 46 (2016), pp. 59–82.

[172] L. Györfi and H. Walk. "On the averaged stochastic approximation for linear regression". In: *SIAM Journal on Control and Optimization* 34.1 (1996), pp. 31–61.

[173] N. Haghtalab, M. Jordan, and E. Zhao. "On-Demand Sampling: Learning Optimally from Multiple Distributions". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 406–419.

[174] F. R. Hampel. "A general qualitative definition of robustness". In: *The Annals of Mathematical Statistics* 42.6 (1971), pp. 1887–1896.

[175] F. R. Hampel, E. M. Ronchetti, P. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley-Interscience; New York, 1986.

[176] F. R. Hampel. "A General Qualitative Definition of Robustness". In: *The Annals of Mathematical Statistics* 42.6 (1971), pp. 1887–1896.

[177] F. R. Hampel. *Contributions to the Theory of robust Estimation*. University of California, Berkeley, 1968.

[178] N. J. Harvey, C. Liaw, Y. Plan, and S. Randhawa. "Tight analyses for non-smooth stochastic gradient descent". In: *Conference on Learning Theory*. PMLR. 2019, pp. 1579–1613.

[179] N. J. Harvey, C. Liaw, and S. Randhawa. "Simple and optimal high-probability bounds for strongly-convex stochastic gradient descent". In: *arXiv preprint arXiv:1909.00843* (2019).

[180] T. Hastie and D. Pregibon. *Shrinking trees*. AT & T Bell Laboratories, 1990.

[181] T. Hastie, R. Tibshirani, and M. Wainwright. "Statistical learning with sparsity". In: *Monographs on Statistics and Applied Probability* 143 (2015), p. 143.

[182] D. M. Hawkins. *Identification of outliers*. Vol. 11. Springer, 1980.

[183] E. Hazan et al. "Introduction to online convex optimization". In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325.

[184] E. Hazan, A. Agarwal, and S. Kale. "Logarithmic regret algorithms for online convex optimization". In: *Machine Learning* 69.2-3 (2007), pp. 169–192.

[185] E. Hazan, T. Koren, and K. Y. Levy. "Logistic regression: Tight bounds for stochastic and online optimization". In: *Conference on Learning Theory*. PMLR. 2014, pp. 197–209.

[186] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[187] D. P. Helmbold and R. E. Schapire. "Predicting Nearly As Well As the Best Pruning of a Decision Tree". In: *Machine Learning* 27.1 (1997), pp. 51–68.

[188] T. K. Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[189] C. A. R. Hoare. "Algorithm 65: Find". In: *Commun. ACM* 4.7 (July 1961), pp. 321–322.

[190] M. Holland. "Robustness and scalability under heavy tails, without strong convexity". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 865–873.

[191] M. Holland and K. Ikeda. "Better generalization with less data using robust gradient descent". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2761–2770.

[192] M. J. Holland. "Robust descent using smoothed multiplicative noise". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 703–711.

[193] M. J. Holland and K. Ikeda. "Efficient learning with robust gradient descent". In: *Machine Learning* 108.8 (2019), pp. 1523–1560.

[194] S. B. Hopkins and J. Li. "Mixture models, robustness, and sum of squares proofs". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 1021–1034.

[195] S. B. Hopkins. "Mean estimation with sub-Gaussian rates in polynomial time". In: *The Annals of Statistics* 48.2 (2020), pp. 1193–1213.

[196] D. Hsu and S. Sabato. "Heavy-tailed regression with a generalized median-of-means". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 37–45.

[197] D. Hsu and S. Sabato. "Loss minimization and parameter estimation with heavy tails". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 543–582.

[198] J. Huang, J. L. Horowitz, and F. Wei. "Variable selection in nonparametric additive models". In: *The Annals of Statistics* 38.4 (2010), p. 2282.

[199] P. J. Huber. "A robust version of the probability ratio test". In: *The Annals of Mathematical Statistics* (1965), pp. 1753–1758.

[200] P. J. Huber. "Robust estimation of a location parameter". In: *Breakthroughs in Statistics*. Springer, 1992, pp. 492–518.

[201] P. J. Huber. *Robust statistics*. Vol. 523. John Wiley & Sons, 2004.

[202] P. J. Huber. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[203] P. J. Huber. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[204] P. J. Huber. "Robust Regression: Asymptotics, Conjectures and Monte Carlo". In: *The Annals of Statistics* 1.5 (1973), pp. 799–821.

[205] P. J. Huber. *Robust Statistics*. Wiley, New York, 1981.

[206] P. J. Huber. "The 1972 wald lecture robust statistics: A review". In: *The Annals of Mathematical Statistics* 43.4 (1972), pp. 1041–1067.

[207] L. Iosipoi and A. Vakhrushev. "SketchBoost: Fast Gradient Boosted Decision Tree for Multioutput Problems". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 25422–25435.

[208] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, V. K. Pillutla, and A. Sidford. "A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares)". In: *arXiv preprint arXiv:1710.09430* (2017).

[209] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. "Parallelizing stochastic approximation through mini-batching and tail-averaging". In: *arXiv preprint arXiv:1610.03774* (2016).

[210] P. Jain, D. Nagaraj, and P. Netrapalli. "Making the last iterate of SGD information theoretically optimal". In: *Conference on Learning Theory*. PMLR. 2019, pp. 1752–1755.

[211] P. Jain, N. Rao, and I. S. Dhillon. "Structured sparse regression via greedy hard thresholding". In: *Advances in Neural Information Processing Systems* 29 (2016).

[212] P. Jain, A. Tewari, and P. Kar. "On iterative hard thresholding methods for high-dimensional M-estimation". In: *Advances in Neural Information Processing Systems* 27 (2014).

[213] M. Jeong, J. Nam, and B. C. Ko. "Lightweight Multilayer Random Forests for Monitoring Driver Emotional Status". In: *IEEE Access* 8 (2020), pp. 60344–60354.

[214] M. Jeong, M. Park, and B. C. Ko. "Intelligent Driver Emotion Monitoring Based on Lightweight Multilayer Random Forests". In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. Vol. 1. 2019, pp. 280–283.

[215] D. C. Jerison. "Quantitative convergence rates for reversible Markov chains via strong random times". In: *arXiv preprint arXiv:1908.06459* (2019).

[216] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. "Random generation of combinatorial structures from a uniform distribution". In: *Theoretical Computer Science* 43 (1986), pp. 169–188.

[217] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. "Random generation of combinatorial structures from a uniform distribution". In: *Theoretical Computer Science* 43 (1986), pp. 169–188.

[218] R. Jézéquel, P. Gaillard, and A. Rudi. "Efficient improper learning for online logistic regression". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2085–2108.

[219] R. Jézéquel, P. Gaillard, and A. Rudi. "Mixability made efficient: Fast online multiclass logistic regression". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23692–23702.

[220] Joblib Development Team. *Joblib: running Python functions as pipeline jobs*. 2020.

[221] U. Johansson, C. Sönströd, and T. Löfström. "One tree to explain them all". In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE. 2011, pp. 1444–1451.

[222] R. Johnson and T. Zhang. "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in Neural Information Processing Systems* 26 (2013).

[223] A. Juditsky, A. Kulunchakov, and H. Tsyntseus. "Sparse recovery by reduced variance stochastic approximation". In: *Information and Inference: A Journal of the IMA* 12.2 (Nov. 2022), pp. 851–896. eprint: `https://academic.oup.com/imaiai/article-pdf/12/2/851/49287824/iaac028.pdf`.

[224] A. Juditsky, A. Kulunchakov, and H. Tsyntseus. "Sparse recovery by reduced variance stochastic approximation". In: *Information and Inference: A Journal of the IMA* 12.2 (2023), pp. 851–896.

[225] A. Juditsky, J. Kwon, and É. Moulines. "Unifying mirror descent and dual averaging". In: *arXiv preprint arXiv:1910.13742* (2019).

[226] A. Juditsky, A. Nemirovski, et al. "First order methods for nonsmooth convex large-scale optimization, i: general purpose methods". In: *Optimization for Machine Learning* 30.9 (2011), pp. 121–148.

[227] A. Juditsky and Y. Nesterov. "Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization". In: *Stochastic Systems* 4.1 (2014), pp. 44–80.

[228] S. M. Kakade and A. Tewari. "On the generalization ability of online strongly convex programming algorithms". In: *Advances in Neural Information Processing Systems* 21 (2008).

[229] H. Karimi, J. Nutini, and M. Schmidt. "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*. Springer. 2016, pp. 795–811.

[230] S. J. Kazemitabar, A. A. Amini, A. Bloniarz, and A. Talwalkar. "Variable importance using decision trees". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 425–434.

[231] G. Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[232] D. G. Kendall. "Unitary dilations of Markov transition operators, and the corresponding integral representations for transition-probability matrices". In: *Probability and Statistics* (1959), pp. 139–161.

[233] M. Kijima. *Markov Processes for Stochastic Modeling*. Vol. 6. CRC Press, 1997.

[234] S. Kim, S. Kwak, and B. C. Ko. "Fast Pedestrian Detection in Surveillance Video Based on Soft Target Training of Shallow Random Forest". In: *IEEE Access* 7 (2019), pp. 12415–12426.

[235] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[236] A. Klivans, P. K. Kothari, and R. Meka. "Efficient algorithms for outlier-robust regression". In: *Conference On Learning Theory*. PMLR. 2018, pp. 1420–1430.

[237] A. R. Klivans, P. M. Long, and R. A. Servedio. "Learning Halfspaces with Malicious Noise." In: *Journal of Machine Learning Research* 10.12 (2009).

[238] D. E. Knuth. "Seminumerical Algorithms (The Art of Computer Programming 2)". In: *Reading, MA,: AddisonWesley* (1969), pp. 124–125.

[239] M. Koklu and I. A. Ozkan. "Multiclass classification of dry beans using computer vision and machine learning techniques". In: *Computers and Electronics in Agriculture* 174 (2020), p. 105507.

[240] V. Koltchinskii. "Local Rademacher complexities and oracle inequalities in risk minimization". In: *The Annals of Statistics* 34.6 (2006), pp. 2593–2656.

[241] V. Koltchinskii, K. Lounici, and A. B. Tsybakov. "Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion". In: *The Annals of Statistics* 39.5 (2011), pp. 2302–2329.

[242] A. Kontorovich and M. Raginsky. "Concentration of measure without independence: a unified approach via the martingale method". In: *Convexity and Concentration*. Springer, 2017, pp. 183–210.

[243] I. Kontoyiannis and S. P. Meyn. "Geometric ergodicity and the spectral gap of non-reversible Markov chains". In: *Probability Theory and Related Fields* 154.1-2 (2012), pp. 327–339.

[244] P. K. Kothari, J. Steinhardt, and D. Steurer. "Robust moment estimation and improved clustering via sum of squares". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 1035–1046.

[245] K. Koutroumbas and S. Theodoridis. *Pattern Recognition*. Academic Press, 2008.

[246] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[247] M. Kuhn and K. Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[248] S. Kullback and R. A. Leibler. "On information and sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[249] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Vol. 35. Springer New York, NY, 2003.

[250] S. Lacoste-Julien, M. Schmidt, and F. Bach. "A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method". In: *arXiv preprint arXiv:1212.2002* (2012).

[251] K. A. Lai, A. B. Rao, and S. Vempala. "Agnostic estimation of mean and covariance". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 665–674.

[252] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. "Mondrian Forests: Efficient Online Random Forests". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 3140–3148.

[253] C. Lakshminarayanan and C. Szepesvari. "Linear stochastic approximation: How far does constant step-size and iterate averaging go?" In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 1347–1355.

[254] S. K. Lam, A. Pitrou, and S. Seibert. "Numba: A LLVM-Based Python JIT Compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM '15. Austin, Texas: Association for Computing Machinery, 2015.

[255] G. Lan. *First-Order and Stochastic Optimization Methods for Machine Learning*. Vol. 1. Springer, 2020.

[256] H. Laurent and R. L. Rivest. "Constructing optimal binary decision trees is NP-complete". In: *Information Processing Letters* 5.1 (1976), pp. 15–17.

[257] G. Lecué and M. Lerasle. "Learning from MOM's principles: Le Cam's approach". In: *Stochastic Processes and their Applications* 129.11 (2019), pp. 4385–4410.

[258] G. Lecué and M. Lerasle. "Robust machine learning by median-of-means: Theory and practice". In: *The Annals of Statistics* 48 (Nov. 2017).

[259] G. Lecué and M. Lerasle. "Robust machine learning by median-of-means: Theory and practice". In: *The Annals of Statistics* 48.2 (2020), pp. 906–931.

[260] G. Lecué, M. Lerasle, et al. "Robust machine learning by median-of-means: theory and practice". In: *The Annals of Statistics* 48.2 (2020), pp. 906–931.

[261] G. Lecué, M. Lerasle, and T. Mathieu. "Robust classification via MOM minimization". In: *Machine learning* 109 (2020), pp. 1635–1665.

[262] G. Lecué, M. Lerasle, and T. Mathieu. "Robust classification via MOM minimization". In: *Machine Learning* 109.8 (2020), pp. 1635–1665.

[263] G. Lecué and S. Mendelson. "Learning subgaussian classes: upper and minimax bounds (2013)". In: *Topics in Learning Theory-Société Mathématique de France,(S. Boucheron and N. Vayatis Eds.)* (2013).

[264] G. Lecué and S. Mendelson. "Regularization and the small-ball method i: sparse recovery". In: *The Annals of Statistics* 46.2 (2018), pp. 611–641.

[265] Y. LeCun et al. "Learning algorithms for classification: A comparison on handwritten digit recognition". In: *Neural Networks: the Statistical Mechanics Perspective* 261.276 (1995), pp. 2–18.

[266] M. Ledoux. "On Talagrand's deviation inequalities for product measures". In: *ESAIM: Probability and Statistics* 1 (1997), pp. 63–87.

[267] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Berlin, Heidelberg, 1991.

[268] Z. Lei, K. Luh, P. Venkat, and F. Zhang. "A fast spectral algorithm for mean estimation with sub-Gaussian rates". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2598–2612.

[269] J. Li. "Robust sparse estimation tasks in high dimensions". In: *arXiv preprint arXiv:1702.05860* (2017).

[270] X. Li and F. Orabona. "A high probability analysis of adaptive SGD with momentum". In: *arXiv preprint arXiv:2007.14294* (2020).

[271] X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong. "On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6741–6764.

[272] F. T. Liu, K. M. Ting, and Z.-H. Zhou. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422.

[273] L. Liu, T. Li, and C. Caramanis. "High dimensional robust estimation of sparse models via trimmed hard thresholding". In: *arXiv preprint arXiv:1901.08237* (2019).

[274] L. Liu, Y. Shen, T. Li, and C. Caramanis. "High dimensional robust sparse regression". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 411–421.

[275] T. Liu and D. Tao. "Classification with noisy labels by importance reweighting". In: *IEEE Transactions on pattern analysis and machine intelligence* 38.3 (2015), pp. 447–461.

[276] Y. Liu et al. "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models". In: *arXiv preprint arXiv:2304.01852* (2023).

[277] Z. Liu and Z. Zhou. "Stochastic nonsmooth convex optimization with heavy-tailed noises". In: *arXiv preprint arXiv:2303.12277* (2023).

[278] Z. Lou, W. Zhu, and W. B. Wu. "Beyond sub-Gaussian noises: Sharp concentration analysis for stochastic gradient descent". In: *Journal of Machine Learning Research* 23 (2022), pp. 1–22.

[279] K. Lounici. "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators". In: *Electronic Journal of statistics* 2 (2008), pp. 90–102.

[280] K. Lounici, M. Pontil, A. B. Tsybakov, and S. Van De Geer. "Taking advantage of sparsity in multi-task learning". In: *arXiv preprint arXiv:0903.1468* (2009).

[281] G. Louppe. "Understanding random forests: From theory to practice". PhD thesis. University of Liege, 2014.

[282] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. "Understanding variable importances in forests of randomized trees". In: *Advances in Neural Information Processing Systems*. Ed. by C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger. Vol. 26. Curran Associates, Inc., 2013, pp. 431–439.

[283] G. Lugosi and S. Mendelson. "Risk minimization by median-of-means tournaments". In: *Journal of the European Mathematical Society* 22.3 (2019), pp. 925–965.

[284] G. Lugosi and S. Mendelson. "Mean estimation and regression under heavy-tailed distributions: A survey". In: *Foundations of Computational Mathematics* 19.5 (2019), pp. 1145–1190.

[285] G. Lugosi and S. Mendelson. "Near-optimal mean estimators with respect to general norms". In: *Probability Theory and Related Fields* 175.3-4 (2019), pp. 957–973.

[286] G. Lugosi and S. Mendelson. "Regularization, sparse recovery, and median-of-means tournaments". In: *Bernoulli* 25.3 (2019), pp. 2075–2106.

[287] G. Lugosi and S. Mendelson. "Robust multivariate mean estimation: The optimality of trimmed mean". In: *The Annals of Statistics* 49.1 (2021), pp. 393–410.

[288] G. Lugosi and S. Mendelson. "Robust multivariate mean estimation: the optimality of trimmed mean". In: *The Annals of Statistics* 49.1 (2021), pp. 393–410.

[289] G. Lugosi and S. Mendelson. "Sub-Gaussian estimators of the mean of a random vector". In: *The Annals of Statistics* 47.2 (2019), pp. 783–794.

[290] R. Lund, Y. Zhao, and P. C. Kiessler. "A monotonicity in reversible Markov chains". In: *Journal of Applied Probability* 43.2 (2006), pp. 486–499.

[291] R. B. Lund, S. P. Meyn, and R. L. Tweedie. "Computable exponential convergence rates for stochastically ordered Markov processes". In: *The Annals of Applied Probability* 6.1 (1996), pp. 218–237.

[292] R. B. Lund and R. L. Tweedie. "Geometric convergence rates for stochastically ordered Markov chains". In: *Mathematics of Operations Research* 21.1 (1996), pp. 182–194.

[293] S. Ma, R. Bassily, and M. Belkin. "The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning". In: *International Conference on Machine Learning*. 2017.

[294] S. Ma, R. Bassily, and M. Belkin. "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3325–3334.

[295] N. Madras and D. Sezer. "Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances". In: *Bernoulli* 16.3 (2010), pp. 882–908.

[296] B. Mandelbrot. "The Variation of Certain Speculative Prices". In: *The Journal of Business* 36 (1963).

[297] R. Martin and C. Masreliez. "Robust estimation via stochastic approximation". In: *IEEE Transactions on Information Theory* 21.3 (1975), pp. 263–271.

[298] P. Massart and É. Nédélec. "Risk bounds for statistical learning". In: *The Annals of Statistics* 34.5 (2006), pp. 2326–2366.

[299] T. Mathieu. "M-estimation and Median of Means applied to statistical learning". PhD thesis. Université Paris-Saclay, 2021.

[300] T. Mathieu. "Concentration study of M-estimators using the influence function". In: *Electronic Journal of Statistics* 16.1 (2022), pp. 3695–3750.

[301] A. Maurer and M. Pontil. "Concentration inequalities under sub-Gaussian and sub-exponential conditions". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 7588–7597.

[302] A. Maurer and M. Pontil. "Empirical Bernstein Bounds and Sample-Variance Penalization". In: *COLT*. 2009.

[303] D. A. McAllester. "PAC-Bayesian model averaging". In: *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*. ACM. 1999, pp. 164–170.

[304] D. A. McAllester. "Some PAC-Bayesian theorems". In: *Proceedings of the 11th Annual conference on Computational Learning Theory (COLT)*. ACM. 1998, pp. 230–234.

[305] C. McDiarmid. "Concentration". In: *Probabilistic Methods for Algorithmic Discrete Mathematics* (1998), pp. 195–248.

[306] H. B. McMahan and M. Streeter. "Open problem: Better bounds for online logistic regression". In: *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings. 2012, pp. 1–44.

[307] H. B. McMahan et al. "Ad click prediction: a view from the trenches". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 1222–1230.

[308] N. Meinshausen. "Quantile Regression Forests". In: *Journal of Machine Learning Research* 7.35 (2006), pp. 983–999.

[309] L. Mentch and S. Zhou. *Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success*. 2020. arXiv: `1911.00190 [stat.ML]`.

[310] I. Merad and S. Gaïffas. "Robust methods for high-dimensional linear learning". In: *Journal of Machine Learning Research* 24.165 (2023), pp. 1–44.

[311] I. Merad and S. Gaïffas. "Convergence and concentration properties of constant step-size SGD through Markov chains". In: *arXiv preprint arXiv:2306.11497* (2023).

[312] R. Messenger and L. Mandell. "A modal search technique for predictive nominal scale multivariate analysis". In: *Journal of the American statistical association* 67.340 (1972), pp. 768–772.

[313] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer London, 1993.

[314] S. P. Meyn and R. L. Tweedie. "Computable bounds for geometric convergence rates of Markov chains". In: *The Annals of Applied Probability* (1994), pp. 981–1011.

[315] S. Minsker. "Geometric median and robust estimation in Banach spaces". In: *Bernoulli* 21.4 (2015), pp. 2308–2335.

[316] S. Minsker et al. "Geometric median and robust estimation in Banach spaces". In: *Bernoulli* 21.4 (2015), pp. 2308–2335.

[317] S. Minsker et al. "Sub-Gaussian estimators of the mean of a random matrix with heavy-tailed entries". In: *The Annals of Statistics* 46.6A (2018), pp. 2871–2903.

[318] S. Minsker, M. Ndaoud, and L. Wang. "Robust and Tuning-Free Sparse Linear Regression via Square-Root Slope". In: *arXiv preprint arXiv:2210.16808* (2022).

[319] I. Mizera et al. "On depth and deep points: a calculus". In: *The Annals of Statistics* 30.6 (2002), pp. 1681–1736.

[320] V. Mnih, C. Szepesvári, and J.-Y. Audibert. "Empirical Bernstein stopping". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 672–679.

[321] V. Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[322] J. N. Morgan and J. A. Sonquist. "Problems in the analysis of survey data, and a proposal". In: *Journal of the American statistical association* 58.302 (1963), pp. 415–434.

[323] W. Mou, C. J. Li, M. J. Wainwright, P. L. Bartlett, and M. I. Jordan. "On linear stochastic approximation: Fine-grained Polyak-Ruppert and non-asymptotic concentration". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2947–2997.

[324] E. Moulines and F. Bach. "Non-asymptotic analysis of stochastic approximation algorithms for machine learning". In: *Advances in Neural Information Processing Systems* 24 (2011).

[325] J. Mourtada and S. Gaïffas. "An improper estimator with optimal excess risk in misspecified density estimation and logistic regression". In: *Journal of Machine Learning Research* 23.31 (2022), pp. 1–49.

[326] J. Mourtada and S. Gaïffas. "An improper estimator with optimal excess risk in misspecified density estimation and logistic regression". In: *Journal of Machine Learning Research* 23.31 (2022), pp. 1–49.

[327] J. Mourtada, S. Gaïffas, and E. Scornet. "AMF: Aggregated Mondrian forests for online learning". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 83.3 (2021), pp. 505–533.

[328] J. Mourtada, S. Gaïffas, E. Scornet, et al. "Minimax optimal rates for Mondrian trees and forests". In: *Annals of Statistics* 48.4 (2020), pp. 2253–2276.

[329] K. P. Murphy. *Probabilistic Machine Learning: an Introduction*. MIT press, 2022.

[330] A. V. Nazin, A. S. Nemirovsky, A. B. Tsybakov, and A. B. Juditsky. "Algorithms of robust stochastic optimization based on mirror descent method". In: *Automation and Remote Control* 80 (2019), pp. 1607–1627.

[331] A. V. Nazin, B. T. Polyak, and A. B. Tsybakov. "Optimal and robust kernel algorithms for passive stochastic approximation". In: *IEEE Transactions on Information Theory* 38.5 (1992), pp. 1577–1583.

[332] I. Necoara, Y. Nesterov, and F. Glineur. "Linear convergence of first order methods for non-strongly convex optimization". In: *Mathematical Programming* 175 (2019), pp. 69–107.

[333] A. Nedić and D. Bertsekas. "Convergence rate of incremental subgradient algorithms". In: *Stochastic Optimization: Algorithms and Applications* (2001), pp. 223–264.

[334] D. Needell, R. Ward, and N. Srebro. "Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm". In: *Advances in Neural Information Processing Systems* 27 (2014).

[335] S. Negahban and M. J. Wainwright. "Estimation of (near) low-rank matrices with noise and high-dimensional scaling". In: *The Annals of Statistics* 39.2 (2011), pp. 1069–1097.

[336] S. Negahban and M. J. Wainwright. "Restricted strong convexity and weighted matrix completion: Optimal bounds with noise". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 1665–1697.

[337] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers". In: *Statistical Science* 27.4 (2012), pp. 538–557.

[338] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. "Robust stochastic approximation approach to stochastic programming". In: *SIAM Journal on optimization* 19.4 (2009), pp. 1574–1609.

[339] A. S. Nemirovskij and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[340] Y. Nesterov and V. Shikhman. "Quasi-monotone subgradient methods for nonsmooth convex minimization". In: *Journal of Optimization Theory and Applications* 165.3 (2015), pp. 917–940.

[341] Y. Nesterov. "Efficiency of coordinate descent methods on huge-scale optimization problems". In: *SIAM Journal on Optimization* 22.2 (2012), pp. 341–362.

[342] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer New York, NY, 2004.

[343] Y. Nesterov. "Primal-dual subgradient methods for convex problems". In: *Mathematical Programming* 120.1 (2009), pp. 221–259.

[344] Y. Nesterov and A. Nemirovski. "On first-order algorithms for l1/nuclear norm minimization". In: *Acta Numerica* 22 (2013), pp. 509–575.

[345] G. Neu and L. Rosasco. "Iterate averaging as regularization for stochastic gradient descent". In: *Conference On Learning Theory*. PMLR. 2018, pp. 3222–3242.

[346] T. D. Nguyen, T. H. Nguyen, A. Ene, and H. L. Nguyen. "High probability convergence of clipped-SGD under heavy-tailed noise". In: *arXiv preprint arXiv:2302.05437* (2023).

[347] F. Orabona. "A Modern Introduction to Online Learning". In: *ArXiv* abs/1912.13213 (2019).

[348] L. Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[349] A. Owen. "A robust hybrid of lasso and ridge regression". In: *Contemporary Mathematics* 443.7 (2007), pp. 59–72.

[350] N. C. Oza and S. Russell. "Online Bagging and Boosting". In: *Proceedings of the 8th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2001.

[351] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl. *Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features*. 2021. arXiv: 2104.00629 [stat.ML].

[352] A. Patil and S. Singh. "Differential private random forest". In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2014, pp. 2623–2630.

[353] D. Paul, S. Chakraborty, and S. Das. "Robust Principal Component Analysis: A Median of Means Approach". In: *arXiv preprint arXiv:2102.03403* (2021).

[354] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[355] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[356] A. Pensia, V. Jog, and P.-L. Loh. "Robust regression with covariate filtering: Heavy tails and adversarial contamination". In: *arXiv preprint arXiv:2009.12976* (2020).

[357] G. C. Pflug. "Stochastic minimization with constant step-size: asymptotic laws". In: *SIAM Journal on Control and Optimization* 24.4 (1986), pp. 655–666.

[358] B. T. Polyak and A. B. Juditsky. "Acceleration of stochastic approximation by averaging". In: *SIAM Journal on Control and Optimization* 30.4 (1992), pp. 838–855.

[359] B. T. Polyak and Y. Z. Tsypkin. "Adaptive estimation algorithms: convergence, optimality, stability". In: *Automation and Remote Control* 40.3 (1979), pp. 378–389.

[360] B. T. Polyak and Y. Z. Tsypkin. "Robust pseudogradient adaptation algorithms". In: *Automation and Remote Control* 41.10 (1981), pp. 1404–1409.

[361] A. Prasad, S. Balakrishnan, and P. Ravikumar. "A Robust Univariate Mean Estimator is All You Need". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. PMLR, 2020, pp. 4034–4044.

[362] A. Prasad, S. Balakrishnan, and P. Ravikumar. "A robust univariate mean estimator is all you need". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4034–4044.

[363] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. "Robust estimation via robust gradient estimation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82 (2018).

[364] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. "Robust estimation via robust gradient estimation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82.3 (2020), pp. 601–627.

[365] E. Price and V. VandeLinde. "Robust estimation using the robbins-monro stochastic approximation algorithm". In: *IEEE Transactions on Information Theory* 25.6 (1979), pp. 698–704.

[366] P. Probst, M. N. Wright, and A.-L. Boulesteix. "Hyperparameters and tuning strategies for random forest". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.3 (2019), e1301.

[367] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. "CatBoost: unbiased boosting with categorical features". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018, pp. 6638–6648.

[368] Q. Qin and J. P. Hobert. "On the limitations of single-step drift and minorization in Markov chain convergence analysis". In: *The Annals of Applied Probability* 31.4 (2021), pp. 1633–1659.

[369] Q. Qin and J. P. Hobert. "Wasserstein-based methods for convergence complexity analysis of MCMC with applications". In: *The Annals of Applied Probability* 32.1 (2022), pp. 124–166.

[370] J. R. Quinlan. "Induction of decision trees". In: *Machine learning* 1.1 (1986), pp. 81–106.

[371] A. Rakhlin, O. Shamir, and K. Sridharan. "Making gradient descent optimal for strongly convex stochastic optimization". In: *arXiv preprint arXiv:1109.5647* (2011).

[372] G. Raskutti, M. J. Wainwright, and B. Yu. "Restricted eigenvalue properties for correlated Gaussian designs". In: *The Journal of Machine Learning Research* 11 (2010), pp. 2241–2259.

[373] D. Ren, N. Aubert-Kato, E. Anzai, Y. Ohta, and J. Tripette. "Random forest algorithms for recognizing daily life activities using plantar pressure information: a smart-shoe study". In: *PeerJ* 8 (2020), e10170.

[374] H. Robbins and S. Monro. "A stochastic approximation method". In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[375] G. O. Roberts and R. L. Tweedie. "Bounds on regeneration times and convergence rates for Markov chains". In: *Stochastic Processes and their Applications* 80.2 (1999), pp. 211–229.

[376] G. O. Roberts and R. L. Tweedie. "Geometric L_2 and L_1 convergence are equivalent for reversible Markov chains". In: *Journal of Applied Probability* 38.A (2001), pp. 37–41.

[377] G. O. Roberts and R. L. Tweedie. "Rates of convergence of stochastically monotone and continuous time Markov models". In: *Journal of Applied Probability* 37.2 (2000), pp. 359–373.

[378] A. Rohde and A. B. Tsybakov. "Estimation of high-dimensional low-rank matrices". In: *The Annals of Statistics* 39.2 (2011), pp. 887–930.

[379] E. M. Ronchetti and P. J. Huber. *Robust Statistics*. John Wiley & Sons Hoboken, NJ, USA, 2009.

[380] J. Rosenthal. "Quantitative convergence rates of Markov chains: A simple account". In: *Electronic Communications in Probability* 7 (2002), pp. 123–128.

[381] J. S. Rosenthal. "Convergence rates for Markov chains". In: *Siam Review* 37.3 (1995), pp. 387–405.

[382] T. J. Rothenberg, F. M. Fisher, and C. B. Tilanus. "A Note on Estimation from a Cauchy Sample". In: *Journal of the American Statistical Association* 59.306 (1964), pp. 460–463. eprint: `https://www.tandfonline.com/doi/pdf/10.1080/01621459.1964.10482170`.

[383] P. J. Rousseeuw and M. Hubert. "Robust statistics for outlier detection". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011), pp. 73–79.

[384] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John wiley & sons, 2005.

[385] M.-H. Roy and D. Larocque. "Prediction intervals with random forests". In: *Statistical Methods in Medical Research* 29.1 (2020). PMID: 30786820, pp. 205–229. eprint: `https://doi.org/10.1177/0962280219829885`.

[386] D. Ruppert. *Efficient estimations from a slowly convergent Robbins-Monro process*. Tech. rep. Cornell University Operations Research and Industrial Engineering, 1988.

[387] L. P. S. *Exposition du Système du Monde*. Mémoires de l'Académie Royale des Sciences de Paris, 1796.

[388] J. Sacks. "Asymptotic distribution of stochastic approximation procedures". In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 373–405.

[389] A. Sadiev et al. "High-Probability Bounds for Stochastic Optimization and Variational Inequalities: the Case of Unbounded Variance". In: *arXiv preprint arXiv:2302.00999* (2023).

[390] T. Sasai. "Robust and Sparse Estimation of Linear Regression Coefficients with Heavy-tailed Noises and Covariates". In: *arXiv preprint arXiv:2206.07594* (2022).

[391] T. Sasai and H. Fujisawa. "Outlier Robust and Sparse Estimation of Linear Regression Coefficients". In: *arXiv preprint arXiv:2208.11592* (2022).

[392] R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[393] M. Schmidt and N. L. Roux. "Fast convergence of stochastic gradient descent under a strong growth condition". In: *arXiv preprint arXiv:1308.6370* (2013).

[394] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. "Trust region policy optimization". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1889–1897.

[395] E. Scornet. "On the asymptotics of random forests". In: *Journal of Multivariate Analysis* 146 (2016), pp. 72–83.

[396] E. Scornet. "Random forests and kernel methods". In: *IEEE Transactions on Information Theory* 62 (2016), pp. 1485–1500.

[397] E. Scornet, G. Biau, and J.-P. Vert. "Consistency of random forests". In: *The Annals of Statistics* 43.4 (Aug. 2015), pp. 1716–1741.

[398] D. Scott and R. Tweedie. "Explicit rates of convergence of stochastically ordered Markov chains". In: *Athens Conference on Applied Probability and Time Series Analysis*. Springer. 1996, pp. 176–191.

[399] H. Sedghi, A. Anandkumar, and E. Jonckheere. "Multi-step stochastic ADMM in high dimensions: Applications to sparse optimization and matrix decomposition". In: *Advances in Neural Information Processing Systems* 27 (2014).

[400] S. Shalev-Shwartz, Y. Singer, and N. Srebro. "Pegasos: Primal estimated sub-gradient solver for svm". In: *Proceedings of the 24th International Conference on Machine Learning*. 2007, pp. 807–814.

[401] S. Shalev-Shwartz and A. Tewari. "Stochastic Methods for ℓ_1-Regularized Loss Minimization". In: *The Journal of Machine Learning Research* 12 (2011), pp. 1865–1892.

[402] O. Shamir and T. Zhang. "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 71–79.

[403] J. Shawe-Taylor, N. Cristianini, et al. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[404] Z. She, Z. Wang, T. Ayer, A. Toumi, and J. Chhatwal. *Estimating County-Level COVID-19 Exponential Growth Rates Using Generalized Random Forests*. Papers 2011.01219. arXiv.org, Oct. 2020.

[405] J. Shen and P. Li. "A tight bound of hard thresholding". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 7650–7691.

[406] S. K. Shevade and S. S. Keerthi. "A simple and efficient algorithm for gene selection using sparse logistic regression". In: *Bioinformatics* 19.17 (2003), pp. 2246–2253.

[407] S. Smith et al. "Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model". In: *arXiv preprint arXiv:2201.11990* (2022).

[408] N. Srebro, K. Sridharan, and A. Tewari. "Optimistic rates for learning with a smooth loss". In: *arXiv preprint arXiv:1009.3896* (2010).

[409] S. S. Stanković and B. D. Kovačević. "Analysis of robust stochastic approximation algorithms for process identification". In: *Automatica* 22.4 (1986), pp. 483–488.

[410] W. Su and E. Candès. "SLOPE is adaptive to unknown sparsity and asymptotically minimax". In: *The Annals of Statistics* 44.3 (2016), pp. 1038–1068.

[411] A. Subasi, E. Alickovic, and J. Kevric. "Diagnosis of chronic kidney disease by using random forest". In: *CMBEBIH 2017*. Springer, 2017, pp. 589–594.

[412] M. A. Taddy, R. B. Gramacy, and N. G. Polson. "Dynamic Trees for Learning and Design". In: *Journal of the American Statistical Association* 106.493 (2011), pp. 109–123. eprint: `http://dx.doi.org/10.1198/jasa.2011.ap09769`.

[413] A. Tavakoli, S. Kumar, M. Boukhechba, and A. Heydarian. *Driver State and Behavior Detection Through Smart Wearables*. 2021. arXiv: `2104.13889 [cs.HC]`.

[414] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[415] T. J. Tjalkens, Y. M. Shtarkov, and F. M. J. Willems. "Sequential weighting algorithms for multi-alphabet sources". In: *6th Joint Swedish-Russian International Workshop on Information Theory*. 1993, pp. 230–234.

[416] J. A. Tropp. "An Introduction to Matrix Concentration Inequalities". In: *Foundations and Trends® in Machine Learning* 8.1-2 (2015), pp. 1–230.

[417] C.-P. Tsai, A. Prasad, S. Balakrishnan, and P. Ravikumar. "Heavy-tailed streaming statistical estimation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 1251–1282.

[418] B. Tsirelson, L. Ibragimov, and V. Sudakov. "Norm of Gaussian sample function". In: *Proceedings of the 3rd Japan-USSR Symposium on Probability Theory. Lecture Notes in Math.* Vol. 550. 1976, pp. 20–41.

[419] A. B. Tsybakov. "Optimal rates of aggregation". In: *Learning theory and kernel machines*. Springer, 2003, pp. 303–313.

[420] J. Tu, W. Liu, X. Mao, and X. Chen. "Variance Reduced Median-of-Means Estimator for Byzantine-Robust Distributed Inference". In: *Journal of Machine Learning Research* 22.84 (2021), pp. 1–67.

[421] J. W. Tukey. "A survey of sampling from contaminated distributions". In: *Contributions to Probability and Statistics* (1960), pp. 448–485.

[422] J. W. Tukey. "A survey of sampling from contaminated distributions". In: *Contributions to probability and statistics* (1960), pp. 448–485.

[423] P. E. Utgoff. "Incremental induction of decision trees". In: *Machine learning* 4.2 (1989), pp. 161–186.

[424] A. W. v. d. Vaart. *Asymptotic Statistics (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, 1998.

[425] S. A. Van de Geer. "High-dimensional generalized linear models and the lasso". In: *The Annals of Statistics* 36.2 (2008), pp. 614–645.

[426] R. Van Handel. *Probability in high dimension*. Tech. rep. PRINCETON UNIV NJ, 2014.

[427] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, NY, 1999.

[428] Y. Vardi and C.-H. Zhang. "The multivariate L_1-median and associated data depth". In: *Proceedings of the National Academy of Sciences* 97.4 (2000), pp. 1423–1426. eprint: `https://www.pnas.org/content/97/4/1423.full.pdf`.

[429] A. Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[430] S. Vaswani, F. Bach, and M. Schmidt. "Fast and Faster Convergence of SGD for Over-Parameterized Models and an Accelerated Perceptron". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. PMLR, 2019, pp. 1195–1204.

[431] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta. "Chemical gas sensor drift compensation using classifier ensembles". In: *Sensors and Actuators B: Chemical* 166 (2012), pp. 320–329.

[432] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Vol. 47. Cambridge university press, 2018.

[433] R. Vershynin. "Introduction to the non-asymptotic analysis of random matrices". In: *arXiv preprint arXiv:1011.3027* (2010).

[434] C. Villani et al. *Optimal Transport: Old and New*. Vol. 338. Springer, 2009.

[435] V. Vovk. "A Game of Prediction with Expert Advice". In: *Journal of Computer and System Sciences* 56.2 (1998), pp. 153–173.

[436] V. Vovk. "Competitive on-line statistics". In: *International Statistical Review* 69.2 (2001), pp. 213–248.

[437] N. M. Vural, L. Yu, K. Balasubramanian, S. Volgushev, and M. A. Erdogdu. "Mirror descent strikes again: Optimal stochastic convex optimization under infinite noise variance". In: *Conference on Learning Theory*. PMLR. 2022, pp. 65–102.

[438] S. Wager and S. Athey. "Estimation and inference of heterogeneous treatment effects using random forests". In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242.

[439] G. A. Watson. "Characterization of the subdifferential of some matrix norms". In: *Linear Algebra and its Applications* 170 (1992), pp. 33–45.

[440] F. M. J. Willems. "The Context-Tree Weighting Method: Extensions". In: *IEEE Transactions on Information Theory* 44.2 (Mar. 1998), pp. 792–798.

[441] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. "The Context-Tree Weighting Method: Basic Properties". In: *IEEE Transactions on Information Theory* 41.3 (May 1995), pp. 653–664.

[442] S. J. Wright. "Coordinate descent algorithms". In: *Mathematical programming* 151.1 (2015), pp. 3–34.

[443] T. T. Wu and K. Lange. "Coordinate descent algorithms for lasso penalized regression". In: *The Annals of Applied Statistics* 2.1 (2008), pp. 224–244.

[444] M. T. Yazici, S. Basurra, and M. M. Gaber. "Edge machine learning: Enabling smart internet of things applications". In: *Big data and cognitive computing* 2.3 (2018), p. 26.

[445] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. "Byzantine-robust distributed learning: Towards optimal statistical rates". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5650–5659.

[446] L. Yu, K. Balasubramanian, S. Volgushev, and M. A. Erdogdu. "An analysis of constant step size SGD in the non-convex regime: Asymptotic normality and bias". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4234–4248.

[447] M. Yuan and Y. Lin. "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.

[448] C.-H. Zhang and J. Huang. "The sparsity and bias of the lasso selection in high-dimensional linear regression". In: *The Annals of Statistics* 36.4 (2008), pp. 1567–1594.

[449] H. Zhang, J. Zimmerman, D. Nettleton, and D. J. Nordman. "Random Forest Prediction Intervals". In: *The American Statistician* 74.4 (2020), pp. 392–406. eprint: `https://doi.org/10.1080/00031305.2019.1585288`.

[450] J. Zhang et al. "Why are adaptive methods good for attention models?" In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15383–15393.

[451] L. Zhang and Z.-H. Zhou. "ℓ_1-regression with Heavy-tailed Distributions". In: *Advances in Neural Information Processing Systems*. 2018, pp. 1084–1094.

[452] T. Zhang. "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2004, p. 116.

[453] T. Zhang. "Solving large scale linear prediction problems using stochastic gradient descent algorithms". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. 2004, p. 116.

[454] T. Zhang. "Some sharp performance bounds for least squares regression with l1 regularization". In: *The Annals of Statistics* 37.5A (2009), pp. 2109–2144.

[455] P. Zhao and B. Yu. "On model selection consistency of Lasso". In: *The Journal of Machine Learning Research* 7 (2006), pp. 2541–2563.

[456] F. Zhdanov and V. Vovk. "Competing with Gaussian linear experts". In: *arXiv preprint arXiv:0910.4683* (2009).

[457] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 2018.

[458] S. Zhou and L. K. Mentch. "Trees, forests, chickens, and eggs: when and why to prune trees in a random forest". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 16 (2021), pp. 45–64.

[459] Y. Zhou, Z. Zhou, and G. Hooker. "Approximation trees: Statistical stability in model distillation". In: *arXiv preprint arXiv:1808.07573* (2018).

[460] M. Zinkevich. "Online convex programming and generalized infinitesimal gradient ascent". In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 928–936.

[461] H. Zou. “The adaptive lasso and its oracle properties”. In: *Journal of the American statistical association* 101.476 (2006), pp. 1418–1429.