NAIST-IS-DD1561025

Doctoral Dissertation

Feedback, computation, and robust learning algorithms

Matthew James Holland

December 15, 2017

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Matthew James Holland

Thesis Committee:

Kazushi Ikeda, Professor	(Supervisor)
Shoji Kasahara, Professor	(Co-supervisor)
Hiroaki Sasaki, Assistant Professor	(Co-supervisor)
Takafumi Kanamori, Professor	(Tokyo Institute of Technology)

Feedback, computation, and robust learning algorithms[*]

Matthew James Holland

Keywords:

Machine learning, robust inference, learning algorithms

[*] Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1561025, December 15, 2017.

Abstract

Modern machine learning methods are being used as tools to solve increasingly important and challenging problems in science, engineering, and the humanities. Be it decoding a large array of neural signals to control a prosthetic limb, identifying the linguistic or stylistic similarities of classical literature, early detection of malignant tumors from noisy images, or the autonomous control of complex systems such as automobiles and power grids, a tremendous amount of resources are now spent on using learning algorithms to tackle key problems of the 21st century.

Many sophisticated machine learning algorithms have been developed over the years, but in essence, they are all based on techniques for solving two fundamental problems: statistical inference and multi-parameter optimization. The former is critical because the observed phenomena only paint a small picture of the entire system of interest, and proper inductive decisions must be reliably made in order to succeed under such limited information. The latter is important because automatic setting of numerous parameters must be done in an efficient manner in order for these tools to be useful in practice. Traditionally, these two sub-routines have been studied almost entirely separately. On the statistical side, an evaluation framework centred on the risk, or the expected value of a pre-designed loss function, has become standard. Within this framework, minimization of the empirical risk estimate (ERM) has become a cornerstone of algorithm design. On the computational side, optimization algorithms for minimizing sums of loss functions in many parameters have been rigorously studied, and these procedures are a direct complement to the empirical risk minimization learning strategy.

This well-established present framework, however, leaves a major gap in our understanding of the performance of learning algorithms, and consequently our ability to engineer more efficient and reliable procedures. This gap is due to the fact that formal performance guarantees for empirical risk minimizers are almost without exception given for *arbitrary* solutions, meaning the actual method of implementation is completely abstracted away. This makes analysis simpler, but drastically limits the guarantees that can be made for procedures that are actually used in practice today. Furthermore, while the simplicity of ERM is appealing, there are many situations in which the estimates forged by this procedure are demonstrably sub-optimal, and pursuing the “learner” analogy further, leading to unreliable feedback to the learner.

In this thesis, we pursue a new methodology based on data-robust statistical inference, that explicitly takes into consideration the computational side of the learning task. In doing so, we seek to demonstrate, both theoretically and empirically, that by committing a small computational overhead to better statistical inference, it is possible to substantially improve learning efficiency and robustness of the overall procedure. That is, better performance can be achieved with less net computational resources, for more problems. In addition, by considering computation and inference together, it is possible to provide guarantees for real-world algorithms that can actually be implemented. Taking our proposed new algorithms and the accompanying performance analysis together, this thesis represents a first step beyond the ERM-centric framework, towards a more flexible and general-purpose algorithm design methodology that is capable of solving modern learning problems.

Acknowledgements

Over nine years have passed since I left Canada for Japan, with five of those years spent at NAIST as a graduate student. Looking back on this period, it was intense, enriching, productive, and very enjoyable. Nara is a wonderful place to live, especially for scrambling up mountains and road cycling, and I do hope to return before too long.

I will try to be concise with my personal acknowledgements. First and foremost, I must thank Professor Kazushi Ikeda for welcoming me into his lab, and for being a true role model for me over the past five years. From technical elements involved in formulating and solving interesting and important research problems, to the skills needed to communicate one's ideas to all manner of audiences, the impact Ikeda-sensei has had on my development as a researcher and educator cannot be overstated.

I must also thank Associate Professor Takatomi Kubo, whose breadth of knowledge, intense passion, unwavering discipline and seemingly inhuman stamina have been an inspiration to me. Discussions with Kubo-sensei have directly contributed to many of the ideas developed in this thesis, and will have a lasting effect on the research I pursue in the future.

Nearly one-third of my life has now been spent abroad, and I am forever indebted to my family for their understanding and support. Finally, I assuredly would not have had the fortitude nor the stamina to make it through nearly a decade of intense work if it were not for my wife Eriko.

Contents

Chapter 1

Introduction

How comes it that human beings, whose contacts with the world are brief and personal and limited, are nevertheless able to know as much as they do know?

- Bertrand Russell

The quotation above, also referred to as "Plato's problem" by Noam Chomsky, eloquently describes the problem of how humans acquire, cultivate, and retain knowledge, or put more bluntly, how humans learn. The chief topic of this thesis is the design and analysis of *learning algorithms*, which are procedures for automated inductive inference in machines, rather than humans. That said, the human learning mechanism provides a natural basis for many computational learning techniques [46], and an intuitive grasp of our own limitations can provide useful insights in the context of designing better machine learning systems. One key point of interest here is the ubiquity of *uncertainty* in learning processes. Learning is done with highly incomplete information and an inductive procedure, and thus is inherently uncertain [20, 21, Ch. 3]. To make this point lucid, let us consider some simple examples. We begin by looking at the paintings in Figure 1.1 below.

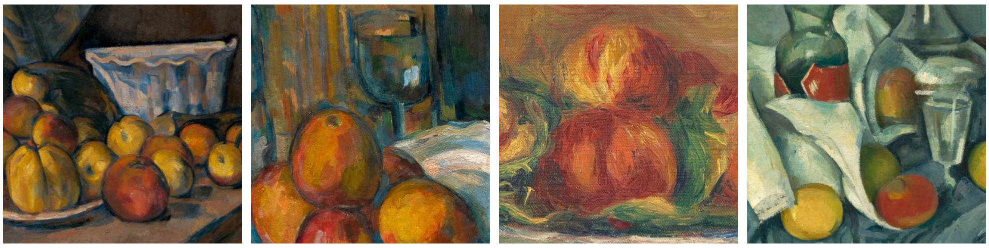

Figure 1.1: Four distinct paintings with similar subject matter. Image credits in A.6.

Which paintings, if any, are from the same artist? In trying to answer this simple, albeit non-trivial question, one might pay attention to the use of colour, the shape of recognizable objects, the nature of brush strokes, among countless other visual features. Based on these features, at the very least, it is intuitively clear that some pairs of paintings are more similar to each other than others, and such an observation might suggest that multiple paintings have a common creator. On the other hand, there can be tremendous variation in such visual features from painting to painting in any particular artist. This potential for variation, coupled with our limited ability to identify characterizing features, induces uncertainty into the task. This uncertainty can be mitigated, although not entirely eliminated, by experience, which in this case can potentially provide background knowledge of artists' traits, or a sharper eye for abstract visual qualities [31].

Let us consider another example in Figure 1.2 below.

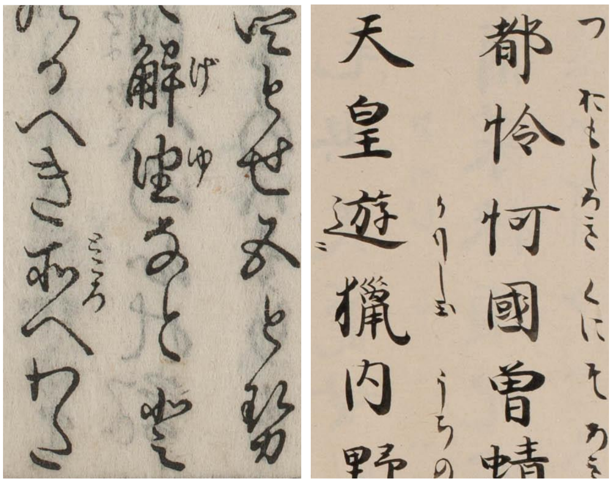

Figure 1.2: Excerpts from *Tosa Nikki* (left) and *Man'yōshū* (right). Image credits in A.6.

Both are Japanese texts dating back over a millenium, but in actually trying to understand each, there are unique challenges. First, in the left-hand figure, the characters are written in a fluid style, including both *kanji* (Chinese characters) and *hiragana* (characters of the Japanese syllabary). Even if one has background knowledge of both character sets, numerous characters are difficult to identify by visual inspection, and indeed may be written differently depending on the context of preceding and proceeding characters. This uncertainty can be mitigated by experience with the author's handwriting (for identifying common character patterns), as well as language knowledge, to forge hypotheses which are grammatical and linguistically plausible. The right-hand image is qualitatively different; in the image displayed, while we have no trouble identifying which characters are written, significant uncertainty of a linguistic nature exists. This is because while only Chinese characters are used, the language is Japanese. Thus, based on knowledge of the language, one must infer the sounds associated with each character, and based on a candidate sequence of sounds, re-construct a plausible text.

The uncertainty highlighted in the examples above is present in the case of machine learning as well. For inference based on visual features, we can provide a computer with a vector of RGB values. For example, the first row of the left-most painting in Figure 1.1 has hexademical RGB values of

```
["#5D4F2C", "#5B4D2A", "#5B4E2C", "#5A502D", "#594E30", "#585134", ...]
```

and the character sequence can be characterized using standardized encodings:

天	皇	遊	獵
E5A4A9	E79A87	E9818A	E78C9F
内	野	之	時
E58685	E9878E	E4B98B	E69982

Figure 1.3: Hexidecimal UTF-8 byte values for a subset of characters on the right-hand side of Figure 1.2.

Implementing a learning procedure on a machine has additional challenges, since there is often a substantial gap between the information that we possess, and the information that we can actually provide the machine with. In the examples above, while we can perceive a wide range of visual features, translating what we perceive into a form digestible by the machine is a non-trivial process.[1] Furthermore, and most importantly, *the actual learning procedure must be explicitly implemented in machines.* For humans, our ability to solve complex tasks based on experience is the product of sophisticated mechanisms for memory and adaptation [14], with a critical role played by feedback [53, 10]. Mere introspection certainly will not give us any hints into the workings of this process, and an intricate neuro-biological explanation has limited utility beyond providing us with a basis for loose analogies. We thus face some challenging questions:

1. What kind of feedback should we provide the learning machine in a given task?

2. How should the machine respond to feedback?

3. How do these feedback/response mechanisms impact performance?

The study of learning algorithms is concerned with providing insights, both of a theoretical and experimental nature, into these realms of inquiry. In this thesis, we propose new algorithms which can be readily implemented in a variety of machine learning problems. The main message that we would like the reader to take away from this thesis is as follows:

Paying a small computational price for better feedback can lead to substantial payoffs in terms of the stability and robustness of learning.

To elucidate and provide evidence for the validity of this statement, we analyze the behaviour and performance of the proposed routines using both numerical simulations, real-world data, and the formal theory of statistical inference. By synthesizing and distilling these formal and empirical insights, we endeavour to draw more general conclusions about the design of learning machines. Doing so, we contribute to a new methodology of learning algorithm design, while making a practical contribution in the form of new machine learning tools.

1.1 Overview of related literature

To better position this work within the machine learning literature, here we take a high-level look at the existing literature which is related to the content of this thesis, without delving into technical details.

1.1.1 On performance evaluation

Performance evaluation is one of the most fundamental issues in algorithm design. As a first step, virtually all domains of machine learning make use of probabilistic models for the data-generating process, and to define ideal "success" metrics on top of this foundation [42, 8, 39]. This lets us explicitly reflect the noise and uncertainty inherent in real-world systems of interest. Given a metric of success, a useful framework for evaluation is the PAC model[2] for learning, originally due to Valiant [50]. Within this model, one explicitly specifies requirements of accuracy (in the success metric), confidence (over the random draw of the data), and computational complexity (in executing the algorithm). Traditionally, one compares rival algorithms

[1] This remains an important issue in machine learning; the "Reliable Machine Learning in the Wild" workshops held at the NIPS 2016 and ICML 2017 conferences covered this important issue.

[2] PAC: Probably approximately correct. Ripley [42, 2.8] provides a good introduction with solid references, and Kearns and Vazirani [30] is a standard reference with an emphasis on computational complexity. A rich treatment in the context of neural networks is due to Anthony and Bartlett [4].

by requiring ε-accuracy at $(1-\delta)$-confidence, and only accepting the algorithm(s) that satisfy the (ε, δ) condition with the least samples, and in polynomial time.

This framework is useful because it gives us a consistent formal evaluation strategy, and suggests a natural approach for carrying out experimental tests via numerical simulation. For truly useful performance evaluation, however, additional effort is required. As has been pointed out by Agarwal [1, 1.2.1], many polynomial-time algorithms may satisfy the (ε, δ) requirement, and the PAC framework does not immediately let us identify the "great" algorithms from amongst the "good" algorithms. Another interesting facet of performance is the inevitable tradeoff between accuracy and confidence; for a fixed task and algorithm, guaranteeing higher accuracy demands a reduction in confidence. Recent work by Lugosi and Mendelson [36] looks at evaluating algorithms in terms of the quality of this tradeoff.

Yet another angle on performance within the PAC setting will be explored in this thesis: a comparison of the assumptions made on the data distribution. Given two algorithms with identical (ε, δ)-sample complexity, choose the one which is more "data-robust," that is, the algorithm whose guarantees hold over a larger class of distributions. A recent line of theoretical work can be readily interpreted within this context [33, 37, 24, 25, 9]. Evaluating the robustness of algorithms in this way can lead to useful new insights, most starkly in the sense of developing algorithms superior to empirical risk minimization (ERM), the *de facto* approach to designing learning algorithms [51]. When we only consider notions of learnability, it is often the case that learnability by ERM is in fact necessary for learnability by any algorithm, which makes the investigation of other algorithms all but pointless [2, 43]. Of course, this is unintuitive, both by practical experience and considering that there are wide data classes where the sample mean is highly sub-optimal as a location estimator [11, 12, 13]. Paying a small computational cost for a large increase in data-robustness is a theme that will be explored in Chapter 3.

Generalizing our discussion once more, it must be noted that there is a fundamental limitation to any evaluation framework (including PAC) that starts with some pre-defined success metric. This limitation is the potential gap between the success metric used, and the intentions of the system designer. Some problems, like binary classification, have an obvious success metric, but do not admit obvious feedback mechanisms and necessitate the introduction of "proxies" [41, 7]. On the other hand, the task of "clustering" data, while intuitively appealing, is vague and does not even admit an obvious success metric, let alone a feedback mechanism, for all but the most rudimentary scenarios. Indeed, depending on our subjective perception of clustering, a wide variety of natural success metrics have been demonstrated [47]. In the literature, we note that the notion of learning reward functions based on expert demonstrations has long received attention in the reinforcement learning community [40], and "learning to learn" has appeared in the machine learning community, notably in the form of a method for learning a first-order update rule [3]. These techniques are explicitly learning data-based update rules, but there is an implicit success metric being approximated. In terms of closing the gap between our intentions and our implementations, future results in this domain are of definite interest in the future, though we do not pursue this direction any further.

1.1.2 On feedback and response mechanisms

In many learning problems, in order to link computable quantities with ideal (unobservable) success metrics, we utilize *loss functions* which depend on observable data and the current candidate parameter.[3] Starting with a loss function, we can then induce ideal success metrics

[3] Virtually all learning problems can be reduced to the process of choosing a vector or function from a class, or *model*. Since this will typically be done in an iterative manner, the "current" parameter refers to the state of the learner at an arbitrary iteration.

from the distribution of the loss, over the random draw of new data.[4] Since learning is then equivalent to optimizing this success metric, it is natural to consider any (sample) statistic of the loss to be potential *feedback* to the learner, and any update rule based on this feedback to be a possible *response*.[5] Analysis of competing learning algorithms then consists of comparing different feedback-response pairs.

We argue in this thesis that the feedback-response paradigm is conducive to more useful performance analysis. More traditionally, the statistical estimation side of learning has been kept almost entirely separate from the computational side, which is to say, the problem of *how to implement* the algorithm. As a case in point, there is a massive body of literature which focuses on ERM deployed to the risk minimization problem [29, 2, 5, 6]. Note that ERM admits any minimizer of the sample mean of the losses, and does not by nature specify a computational method for actually carrying out the minimization. Can all ERM solutions be treated equal? Daniely and Shalev-Shwartz [17], Daniely et al. [16] prove that even for problems as simple as multi-class classification, they cannot. More specifically, there exist "good" and "bad" ERM solutions which have dramatically different generalization properties. Feldman [19] provides a lucid illustration of how a "bad" ERM choice can be disastrous in the general setting of convex risk minimization. These findings clearly show that any meaningful performance analysis must incorporate the implementation of ERM. Some seminal recent works have looked at the generalization ability of ERM when implemented by stochastic gradient descent (SGD), hereby denoted ERM-SGD. Hardt et al. [23] use algorithmic stability as tool for analysis, and giving conditions for the stability of the SGD update, provide learning guarantees for ERM-SGD. Important work from Lin and Rosasco [34] also looks at ERM-SGD, providing sharp conditions in the PAC framework, for the functional optimization setting. Such results are appealing because they account for computational error as well as statistical error, providing much more realistic guarantees.

A good feedback mechanism should, coupled with a response procedure, provide the learner with estimates that lead to *efficient* and *reliable* optimization of the underlying success metric. Efficiency here includes the computational cost of feedback, the cost of a feedback-driven response, and the accumulating effect of the iterations needed to achieve a given level of performance. Reliability, on the other hand, refers to the variance in performance over the random draw of the sample. Due to the ubiquity of the sample mean as a feedback mechanism, much work has been done on the optimization of finite sums of typically convex functions. Well-known procedures include stochastic gradient descent (SAG) [32], stochastic variance-reduced gradient descent (SVRG) [28], stochastic dual coordinate ascent (SDCA) [44], doubly stochastic gradient descent [15, 35], as well as numerous variants using acceleration tweaks and proximity operators in the case of non-smooth objectives [38]. Linking update procedures of this nature to more reliable feedback mechanisms is an important research goal of ours, and in Chapter 4 we take the first step in this direction by proposing a robust gradient-based update protocol. It should be mentioned that in recent years, the notion of "robust estimation" has appeared in the machine learning literature, in a sense strikingly similar to that of classical robust statistics [27, 22, 48]. Sophisticated proposals from Diakonikolas et al. [18] and recent interesting work from Steinhardt et al. [45] are notable for exploring classical robustness in high dimensions. While the details differ, these methods have a common underlying strategy of trying to identify and remove "outliers" from the data. While conceptually and technically appealing, the bias

[4] Without question, the most popular success metric is the *risk*, defined to be the expected value of the loss, taken with respect to the underlying data distribution, evaluated at a particular candidate. This quantity dates back to classical statistical decision theory [52].

[5] While intuitive, it should be noted that the feedback-response terminology is not widespread in the current machine learning literature.

induced by these approaches when the data is simply heavy-tailed (but uncorrupted) is a matter that merits further investigation. Another straightforward but apparently powerful new approach when the data is known to be corrupted is the iterative re-weighting/minimization technique of Vainsencher et al. [49].

1.2 Overview of contributions

This thesis presents two new classes of learning algorithms, along with rigorous experimental and theoretical performance analysis. Given the context of the preceding section, the key novelty here is that we explicitly design new feedback mechanisms, coupled with efficient feedback-based response techniques. This allows us to construct procedures which are data-robust, scale well in dimension and sample size (to both small and large data settings), and are theoretically tractable. While the tradeoffs discussed in section 1.1.1 are naturally unavoidable, we argue that an algorithm design which integrates the feedback and response mechanisms makes it far easier to build reliable and efficient learning machines. We are particularly interested not just in when learning algorithms succeed, but when they fail as well, and our comprehensive experimental analysis sheds new light on algorithm limitations. Alongside these performance analysis results, the specific technical proposals made in the remaining chapters can be taken as concrete evidence which provides a partial answer to the three questions raised at the start of this chapter.

Our core techniques all depend on a common tool for efficiently and flexibly robustifying the feedback provided to the learner. This tool is a class of M-estimators of location and scale, designed in the vein of the classic work of Huber [26], and reflecting key new insights from Catoni [12], plus our own technical modifications. This allows us to take advantage of tightly concentrated location estimators, at the cost of some estimation bias. Fortunately, with data-dependent re-scaling, we demonstrate that this bias can be satisfactorily controlled. In Chapter 2, we provide additional technical background by formulating some of the key concepts discussed above in 1.1.

In Chapter 3, we look at the regression problem, with non-parametric assumptions on the underlying data. Our chief contribution here is an algorithm which utilizes robust estimates of the location of the loss, and uses these location estimates as a new objective to be minimized. A naive attack on this optimization is computationally uncongenial, and to circumvent this we propose an iterative approximate-check routine. Since the new objective function can at least be evaluated efficiently, we formulate a re-weighted least squares problem to approximate the desired update, and then simply check whether the new objective value has improved. This two-step routine is repeated until convergence or until the approximation fails to improve the objective. We also provide theoretical analysis of the core procedure, in particular showing that the desired estimators are well-defined for virtually all problems of interest, and that the robustification technique used allows for intuitive control of bias. Thorough empirical analysis highlights the data-robustness of our algorithms compared with a large number of well-known rival methods, both classical and modern. In both simulations and real data, our algorithm is highly competitive, irrespective of the data distribution used, with no access to prior knowledge. In contrast, the rival techniques show significant disparity between situations in which they succeed and fail. As a whole, we have feedback-response mechanism which is rooted in theory, has a high degree of practical utility, and most importantly demonstrates that better feedback indeed can lead to more efficient and reliable learning.

In Chapter 4, we shift our focus to the general setting of risk minimization problems. In contrast with the previous chapter, instead of beginning with feedback design, this time we start with an optimization technique, and craft the feedback which can best be utilized by the optimizer. The optimizer is steepest descent in the Euclidean metric, namely gradient descent on $\mathbb{R}^d$. Ideal feedback in this case is immediate: the gradient of the (unknown) risk function. Since the loss and its gradient are observable, a by-coordinate robust estimate of the gradient yields a natural "robust gradient descent" procedure. This procedure, as well as in-depth theoretical and empirical performance analysis, form the chief technical contributions of this chapter. We provide upper bounds on the excess risk, demonstrating that strong performance guarantees are available for our routine under much weaker assumptions that are required to make similar assurances for ERM-GD. Detailed numerical experiments illustrate how ERM-GD can fail to be efficient under "inconvenient" data distributions, while the proposed routine remains dominant over a much wider class of data, even under very small samples. Perhaps most notably, we observe a high degree of efficiency in terms of the number of iterations required to converge—by making less wasteful jumps within parameter space, the proposed routine demonstrably gets to a better solution, faster.

Finally, in Chapter 5 we recapitulate the key findings of the preceding technical chapters, attempt to distill them into lucid conclusions, and subsequently lay out a clear roadmap for future work, given the context of the literature introduced in 1.1 and Chapter 2. The Appendix following the final core chapter provides auxiliary technical materials and some history which may be of interest to some readers. Proofs of theoretical results, where applicable, are given in the final section of each chapter.

Bibliography

[1] Agarwal, A. (2012). *Computational Trade-offs in Statistical Learning.* PhD thesis, UC Berkeley.

[2] Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. (1997). Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631.

[3] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29*, pages 3981–3989.

[4] Anthony, M. and Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations.* Cambridge University Press.

[5] Bartlett, P. L., Long, P. M., and Williamson, R. C. (1996). Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452.

[6] Bartlett, P. L. and Mendelson, S. (2006). Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334.

[7] Ben-David, S., Loker, D., Srebro, N., and Sridharan, K. (2012). Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1863–1870.

[8] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

[9] Brownlees, C., Joly, E., and Lugosi, G. (2015). Empirical risk minimization for heavy-tailed losses. *Annals of Statistics*, 43(6):2507–2536.

[10] Cannon, W. B. (1932). *The Wisdom of the Body.* WW Norton & Company.

[11] Catoni, O. (2009). High confidence estimates of the mean of heavy-tailed real random variables. *arXiv preprint arXiv:0909.5366.*

[12] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148–1185.

[13] Catoni, O. (2016). PAC-Bayesian bounds for the Gram matrix and least squares regression with a random design. *arXiv preprint arXiv:1603.05229.*

[14] Chaudhuri, R. and Fiete, I. (2016). Computational principles of memory. *Nature Neuroscience*, 19(3):394–403.

[15] Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems 27*, pages 3041–3049.

[16] Daniely, A., Sabato, S., Ben-David, S., and Shalev-Shwartz, S. (2015). Multiclass learnability and the ERM principle. *Journal of Machine Learning Research*, 16:2377–2404.

[17] Daniely, A. and Shalev-Shwartz, S. (2014). Optimal learners for multiclass problems. In *27th Annual Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 287–316.

[18] Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2017). Being robust (in high dimensions) can be practical. *arXiv preprint arXiv:1703.00893*.

[19] Feldman, V. (2016). Generalization of ERM in stochastic convex optimization: The dimension strikes back. In *Advances in Neural Information Processing Systems 29*, pages 3576–3584.

[20] Fisher, R. A. (1935). The logic of inductive inference. *Journal of the Royal Statistical Society*, 98(1):39–82.

[21] Godfrey-Smith, P. (2003). *Theory and Reality: An Introduction to the Philosophy of Science*. University of Chicago Press.

[22] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

[23] Hardt, M., Recht, B., and Singer, Y. (2015). Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*.

[24] Hsu, D. and Sabato, S. (2014). Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML2014)*, pages 37–45.

[25] Hsu, D. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, 17(18):1–40.

[26] Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

[27] Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, 1st edition.

[28] Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323.

[29] Kearns, M. J. and Schapire, R. E. (1994). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497.

[30] Kearns, M. J. and Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory*. MIT Press.

[31] Koide, N., Kubo, T., Nishida, S., Shibata, T., and Ikeda, K. (2015). Art expertise reduces influence of visual salience on fixation in viewing abstract-paintings. *PLOS ONE*, 10(2):e0117696.

[32] Le Roux, N., Schmidt, M., and Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2663–2671.

[33] Lerasle, M. and Oliveira, R. I. (2011). Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914*.

[34] Lin, J. and Rosasco, L. (2016). Optimal learning for multi-pass stochastic gradient methods. In *Advances in Neural Information Processing Systems 29*, pages 4556–4564.

[35] Lin, J. and Rosasco, L. (2017). Generalization properties of doubly online learning algorithms. *arXiv preprint arXiv:1707.00577*.

[36] Lugosi, G. and Mendelson, S. (2016). Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757*.

[37] Minsker, S. (2015). Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335.

[38] Murata, T. and Suzuki, T. (2017). Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization. *arXiv preprint arXiv:1703.00439*.

[39] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[40] Ng, A. Y. and Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670.

[41] Nock, R. and Nielsen, F. (2008). On the efficient minimization of classification calibrated surrogates. In *Advances in Neural Information Processing Systems 21*.

[42] Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

[43] Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2010). Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670.

[44] Shalev-Shwartz, S. and Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599.

[45] Steinhardt, J., Charikar, M., and Valiant, G. (2017). Resilience: A criterion for learning in the presence of arbitrary outliers. *arXiv preprint arXiv:1703.04940*.

[46] Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

[47] Thomann, P., Steinwart, I., and Schmid, N. (2015). Towards an axiomatic approach to hierarchical clustering of measures. *Journal of Machine Learning Research*, 16:1949–2002.

[48] Tukey, J. W. (1960). A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics: Essays in honor of Harold Hotelling*, pages 448–485. Stanford University Press.

[49] Vainsencher, D., Mannor, S., and Xu, H. (2017). Ignoring is a bliss: Learning with large noise through reweighting-minimization. In *30th Annual Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1849–1881.

[50] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

[51] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

[52] Wald, A. (1949). Statistical decision functions. *Annals of Mathematical Statistics*, pages 165–205.

[53] Wiener, N. (1961). *Cybernetics*. MIT Press, 2nd edition.

Chapter 2

Background

Here we look at key concepts discussed in section 1.1 of the introduction, and endeavour to make them more concrete.

2.1 Probabilistic learning models

The use of probabilistic models is ubiquitous in machine learning. Randomness allows us to account for the uncertainty that exists in nature, the difficulty of inductive inference based on incomplete information, and even "intentional" uncertainty where certain computational routines are randomized, often for reasons of efficiency [77, Ch. 11]. The chief source of randomness that we are interested in comes from the observed data, denoted $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. We call these random variables, taking values in some space $\mathcal{Z}$. That is to say, *once observed*, we have $\boldsymbol{z}_i \in \mathcal{Z}$ for each $i = 1, \ldots, n$. More important than the exact values, however, is the random process that generates them. Classical probability theory gives us an extremely flexible framework for expressing this randomness mathematically [16, 6, 79]. Just to give readers a flavour of this framework, we introduce key terms here.[1] We start with a set of objects Ω, called the sample space. Subsets $A \subseteq \Omega$ correspond to events, intuitively observable phenomena.[2] A collection of subsets of Ω is denoted by $\mathcal{A}$, so each $A \in \mathcal{A}$ satisfies $A \subset \Omega$. Why do we introduce this new class? It is useful when it comes to *measuring* the "probability" of phenomena, our chief interest. This is done by introducing functions which measure the size of sets. This function $\mu : \mathcal{A} \to \mathbb{R}$ assigns to each $A \in \mathcal{A}$ a real number $\mu(A)$. As most readers are assuredly aware, we only formally call such a number a probability if it takes a value on the unit interval $[0, 1]$. This non-negativity alongside additivity of disjoint events leads to an intuitive interpretation of unions of events being more likely than their individual constituents; a rigorous background is available in the first six chapters of Halmos [45], and the first two chapters of Ash and Doléans-Dade [6]. How does this link up with our random variables $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$? Typically $\mathcal{Z}$ is a space we know and understand, say a subset of $\mathbb{R}^d$, or a set of polynomials. The $\boldsymbol{z}_i$ are defined as functions on Ω, and the so-called measurable space $(\Omega, \mathcal{A})$ works quietly in the background. Calling $\boldsymbol{z}_i : \Omega \to \mathcal{Z}$ a random variable is equivalent to saying that for most reasonable choices of $Z \subseteq \mathcal{Z}$, we have that $\{\omega \in \Omega : \boldsymbol{z}_i \in Z\} \in \mathcal{A}$, which means we can use μ to measure the probability of this event. All we are interested in is a systematic description of the random process generating the actual observation $\boldsymbol{z}_i \in \mathcal{Z}$, and thus the

[1] In later chapters, more advanced material appears, although exclusively for matters of technical interest. For references on these topics, see the bibliography [78, 97, 96, 19].

[2] The classical example is $\Omega = \{1, \ldots, 6\}$ for the roll of a six-sided die. The event of an odd number turning up is modeled by $\{1, 3, 5\}$. The event of a number greater than 3 turning up is given by $\{4, 5, 6\}$, and so on. In more complicated problems, the precise form of Ω, however, is not of critical importance.

underlying measure μ encodes everything we need to know. When we say that two random variables $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ have the same "distribution" μ, this means that $\mu\{\boldsymbol{z}_i \in Z\} = \mu\{\boldsymbol{z}_j \in Z\}$ for all events Z within our analytical scope. The functions $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ could be totally distinct; if their random behaviour is the same, within the context of a probabilistic model they are indistinguishable.

Let us move towards a model of learning, in particular, a model which evaluates the performance of learning algorithms. We invoke a useful abstraction here: virtually all learning tasks can be formulated as *algorithms choosing elements from sets, based on data* [2]. We have a model $\mathcal{H}$, often called a hypothesis class, which is a set of elements (numbers, vectors, functions) that will be interpreted as candidate values for the parameters to be determined (or "learned"). A learning algorithm, say $\widehat{h}$, can be understood as a procedure mapping the observations into the $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\} \mapsto \widehat{h} \in \mathcal{H}$. Batch algorithms take the data set as input, online algorithms take a sequence $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$, and in both cases the procedure may include a probabilistic element completely independent from the random draw of the data. In order to evaluate how well an algorithm is doing, a loss function $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ is introduced, with the interpretation that smaller is better. Assuming a plausible task-dependent choice of l, the distribution of $l(h; \boldsymbol{z})$ over the random draw of $\boldsymbol{z} \sim \mu$ is the richest evaluation of the quality of h. Furthermore, the distribution of $l(\widehat{h}; \boldsymbol{z})$, which depends on both the random draw of the sample and a novel point $\boldsymbol{z}$, is used to evaluate the performance of algorithm $\widehat{h}$. Here, two problems arise. First, since μ is unknown, so is the distribution of $l(h; \boldsymbol{z})$ for any choice of h. Second, even if this distribution was known for each candidate $h \in \mathcal{H}$, how would we interpret it? These problems, in particular the latter, are very interesting and highly non-trivial. Most typically, to circumvent these difficulties we elect to look at the expected value of $l(h; \boldsymbol{z})$, namely the risk $R(h) := \mathbf{E}_\mu \, l(h; \boldsymbol{z})$. This provides a partial solution to both problems, since $R(h)$ provides an intuitive quantity to *minimize*, and because estimating $R(\cdot)$ based on data is far easier than estimating the law of $l(h; \boldsymbol{z})$. The only remaining randomness is due to the sample, and it is precisely the random quantity $R(\widehat{h}_n)$ to which we apply the PAC learning framework discussed in 1.1.1.

In learning there are always inevitable tradeoffs; less data means more uncertainty, and all else constant, we can only be more confident in a performance guarantee if we weaken the precision required. Understanding the performance of algorithms requires looking at estimation error, namely the error that is due to the procedure $\widehat{h}$, given model $\mathcal{H}$. The PAC-(ε, δ) condition is often stated as the inequality

$$\mathbf{P}\left\{R(\widehat{h}_n) > R^* + \varepsilon\right\} \leq \delta, \quad R^* := \inf\{R(h) : h \in \mathcal{H}\}$$

for $\varepsilon > 0$ and $\delta \in (0, 1)$, where $\mathbf{P}$ denotes the product measure induced by the sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ and any randomness internal to $\widehat{h}_n$. The smallest n such that this condition holds is called the sample complexity of $\widehat{h}$. Analogously, fixing n and δ, the smallest ε for which $\widehat{h}_n$ satisfies the PAC-(ε, δ) condition is called the accuracy of $\widehat{h}_n$. A smaller ε or δ will push the sample complexity higher, and a larger n should imply stronger guarantees. Similarly, a decrease in ε forces an increase in n and δ, and an investigation of the nature of this tradeoff falls into the realm of fascinating new work from Lugosi and Mendelson [66]. Here, we are particularly interested in a complementary angle on performance, which may be understood as a modern version of algorithm *robustness*, formalized as follows. Let $\widehat{g}$ and $\widehat{h}$ be competing algorithms. Write $\mathcal{P}(\mathcal{Z})$ for all probability distributions on $\mathcal{Z}$, and define

$$\mathcal{P}_{\varepsilon,\delta,n}(\widehat{h}) := \left\{\mu \in \mathcal{P}(\mathcal{Z}) : \widehat{h}_n \text{ is PAC-}(\varepsilon, \delta)\right\}.$$

The basic idea then, is to compare distinct procedures, saying

$$\mathcal{P}_{\varepsilon,\delta,n}(\widehat{g}) \subseteq \mathcal{P}_{\varepsilon,\delta,n}(\widehat{h}) \implies \widehat{h} \text{ is more data-robust than } \widehat{g}$$

under the task conditions specified by n, ε, and δ. As this robustness depends on the sample size, an interesting notion is that of superior robustness at both very large and very small sample sizes, and similarly robustness which holds (or fails to hold) over very high/low confidence and accuracy ranges. This notion of robustness appeared in a series of ground-breaking works in the theoretical statistics literature [23, 24, 7, 62, 30], and has these technical contributions have recently been extended and applied to the context of learning theory, with particularly important works by Minsker [68], Hsu and Sabato [50], and Brownlees et al. [18] breaking a new path in the field. In this thesis, we aim to follow this path, building on existing technical results and introducing new algorithmic ideas in the pursuit of both stronger theoretical performance guarantees and a high level of utility in practice.

It is worth noting that the learning model discussed above suggests a straightforward methodology for experimental testing, using numerical simulations. For a given distribution μ, we can generate two data sets, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ and $\{\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_m\}$, and feeding the first n-sized data set to the algorithm $\widehat{h}$ and getting the output $\widehat{h}_n$, we can use the second m-sized data set to approximate the risk as

$$\widehat{R}(\widehat{h}_n) := \frac{1}{m}\sum_{j=1}^{m} l(\widehat{h}_n; \boldsymbol{z}'_j) \approx R(\widehat{h}_n),$$

noting that conditioned on the "training" set, the law of large numbers gives us almost sure convergence of the left-hand side to the risk as $m \to \infty$. In the PAC framework, we are interested in the random draw of the sample, and thus the data generation procedure just described must be repeated multiple times. Running T trials, we can compute T different sample-dependent outputs, namely $\widehat{h}_n^1, \ldots, \widehat{h}_n^T$. Ideally, we would like to have $R(\widehat{h}_n^1), \ldots, R(\widehat{h}_n^T)$, but we must in practice settle for $\widehat{R}(\widehat{h}_n^1), \ldots, \widehat{R}(\widehat{h}_n^T)$. For any ε then, assuming R^* is known we can approximate

$$\frac{1}{T}\sum_{t=1}^{T} I\{\widehat{R}(\widehat{h}_n^t) > R^* + \varepsilon\} \approx \frac{1}{T}\sum_{t=1}^{T} I\{R(\widehat{h}_n^t) > R^* + \varepsilon\} \approx \mathbf{P}\left\{R(\widehat{h}_n) > R^* + \varepsilon\right\}.$$

In this way, we can compare different algorithms, as plot the increase in the probability of the event $\{R(\widehat{h}_n) > R^* + \varepsilon\}$ as we take ε progressively smaller. To evaluate robustness, this multi-trial procedure must now be repeated for multiple distributions $\mu_1, \mu_2, \ldots$, and if strong performance is confirmed uniformly over these distributions, that can be taken as preliminary evidence for data-robustness. Naturally, the "size" of the space holding these distributions depends fundamentally on the distributions used, but even using well-understood parametric distributions, we can make use of a wide array of distributions, both bounded and unbounded, symmetric and asymmetric, exponentially-tailed and polynomially-tailed, and so forth. See sections 2.2 and 2.6 for an exposition of the distributions used in the numerical experiments contained in this thesis. Both Catoni [24] and Devroye et al. [30] include illuminating discussions of how classes of data distributions can impact the performance guarantees that are possible.

2.2 Testing distributional robustness

In the subsequent chapters of this thesis, we shall make use of numerous parametric distribution families for numerical simulations used to empirically evaluate the performance of different learning algorithms. Here we introduce some qualitative features of these distributions, and give a compact summary that highlights the wide variety of potential data-generating processes covered here. Explicit definitions and additional background on verifying these properties is given in section 2.6.

To begin, among the numerous possible characteristics of probability distributions one could conceivably pay attention to, we shall focus on the following.

- Is the distribution symmetric?
- Do all moments exist? Are they all finite?
- How do the tails behave?

Of these three points, the first two are conceptually very simple, though there may be technical challenges in actually verifying such properties. We are concerned with observations $x \in \mathbb{R}$, with distribution $x \sim \mu$, and thus symmetry is intuitively the property that x is just as likely to take on a value displaced by α to the *right* of $\mathbf{E}_{\mu} x$ as it is to take on a value displaced by α to the *left* of $\mathbf{E}_{\mu} x$. More precisely, there exists some pivot $x_0 \in \mathbb{R}$ such that

$$\mathbf{P}\{x - x_0 \leq \alpha\} = \mathbf{P}\{-(x - x_0) \leq \alpha\}, \quad \forall \, \alpha \in \mathbb{R}.$$

Taking α over all of $\mathbb{R}$ may be superfluous; it need only run over the support of the centred random variable. It follows immediately that if such a x_0 exists, then $x_0 = \mathbf{E}_{\mu} x$. Put simply, then, x is symmetric if $(x - \mathbf{E}_{\mu} x)$ and $-(x - \mathbf{E}_{\mu} x)$ have the same distribution. As regards the existence and finiteness of moments, note that for $m \in \mathbb{N}$,

$$\mathbf{E}_{\mu} x^m = \int x^m \, d\mu = \int (x^m)_+ \, d\mu - \int (x^m)_- \, d\mu$$

where $(u)_+ := \max(u, 0)$ and $(u)_- := \max(-u, 0)$. Both summands on the right-hand side are necessarily on the extended real line $\mathbb{R} \cup \{\infty\}$, but when the right-hand side takes the form $\infty - \infty$, the integral $\mathbf{E}_{\mu} x^m$, and thus the mth moment about zero, is said to be undefined [6, Chapter 1]. In all other cases, it is defined, but may potentially be ±∞.

Regarding the behaviour of the "tails" of the distribution μ, this is more conceptually subtle than the previous two notions, and as such the analyst has some liberty in trying to characterize such behaviour. As a starting point, perhaps the most common dichotomy is that of tails which are "heavy" and those which are "light." Put in completely intuitive terms, the former refers to the quality of x taking on errant values with some non-negligible probability, while the latter refers to distributions where such events have negligible probability. The subtlety arises because the terms *errant* and *negligible* are ambiguous.

To formalize these notions, we shall leverage some well-known properties, which allow us to classify the data-generating processes in a manner useful for performance analysis. As is common in statistics, the Normal (Gaussian) distribution provides a useful initial benchmark.

Definition 1 (Sub-Gaussian distribution)**.** We say that x (or μ) is *sub-Gaussian* if there exists constant(s) such that any of the following hold.

1. $\mathbf{E}_{\mu} \exp(tx) \leq \exp(c_1 t^2/2), \quad t \in \mathbb{R}$.
2. $\mathbf{P}\{|x| \geq \alpha\} \leq c_2 \exp(-\alpha^2/k_2^2), \quad \alpha \geq 0$.
3. $\|x\|_q \leq c_3 \sqrt{q}, \quad q \geq 1$.
4. $\mathbf{E}_{\mu} \exp(x^2/c_4^2) < \infty$.

Call c_1 in 1 the *variance factor* associated with the moment generating function (MGF) condition. Condition 2 is super-exponential tail decay. Condition 3 requires control of all moments, noting $\|x\|_q := (\mathbf{E}_\mu |x|^q)^{1/q}$.

Such a definition is only useful if we have a portmanteau result connecting all these conditions.

Proposition 2 (Sub-Gaussian portmanteau). *For any distribution μ on $\mathbb{R}$, the conditions in Definition 1 satisfy*

$$1 \implies 2 \iff 3 \iff 4.$$

Furthermore, if $\mathbf{E}_\mu x = 0$, then 2 $\implies$ 1.

To prove these facts, see for example Chafaï et al. [25], Vershynin [100], Boucheron et al. [11]. Some technical background on Orlicz spaces may be useful, see van der Vaart and Wellner [97, Chapter 2], Kosorok [58, Chapter 8], Pollard [79, Ch. 2, problems 22–24]. Even though the first condition is not necessary for the other conditions in the non-centred case (where $\mathbf{E}_\mu x \neq 0$), the latter conditions capture the qualities that we are interested in, and thus we have elected to attach the sub-Gaussian name to the union of all distributions satisfying any such conditions. For distributions with unbounded support, the sub-Gaussian tail behaviour can be considered highly congenial. All bounded random variables are sub-Gaussian [11, Lemma 2.2], but we note that this does not preclude the possibility of heavy tails, especially in the case of distributions supported on a finite set [24, Proof of Proposition 6.2]. See also Vershynin [100, Section 5.7].

To extend beyond sub-Gaussianity, we consider two closely related properties as defined by Boucheron et al. [11], which explicitly weaken the moment condition of Definition 1(3).

Definition 3 (Sub-Exponential and sub-Gamma distributions). If for integer $q \geq 1$ there exists a constant $c > 0$ such that

$$\mathbf{E}_\mu x^q \leq c^q q!, \tag{2.1}$$

then call $x \sim \mu$ *sub-Exponential.* Further weakening this, if for some $c, k > 0$ we have

$$\mathbf{E}_\mu x^{2q} \leq k^q q! + c^{2q}(2q)!, \tag{2.2}$$

then call $x \sim \mu$ *sub-Gamma.*

Note immediately that all sub-Exponential distributions are sub-Gamma. For the reader's reference, we aggregate a few more basic facts.

Proposition 4 (Properties of sub-Exponential/Gamma). *If x is sub-Exponential, then*

$$\mathbf{E}_\mu \exp(tx) \leq \frac{1}{(1-ct)}, \quad 0 < t < 1/c. \tag{2.3}$$

If $u := x - \mathbf{E}_\mu x$ is sub-Gamma, then

$$\mathbf{E}_\mu \exp(tu) \leq \exp\left(\frac{2(k+c^2)t^2}{(1-2ct)}\right), \quad 0 < t < 1/(2c). \tag{2.4}$$

For reference, this is called "sub-Gamma on the right tail" by [11, p. 28], and when $-u$ satisfies this property, u is said to be "sub-Gamma on the left tail." Similarly, tail properties also hold.

As one would expect, sub-exponential random variables have exponential tails, in that for some $k_1, k_2 > 0$, we have

$$\mathbf{P}\{|x| \geq \alpha\} \leq k_1 \exp(-\alpha/k_2). \tag{2.5}$$

Sub-Gamma random variables satisfy a similar property. If x is sub-Gamma, defining

$$h(\alpha) := \mathbf{E}_{\mu}\, x + 2\sqrt{(k+c^2)\alpha} + 2c\alpha,$$

we have

$$\mathbf{P}\{x \geq h(\alpha)\} \vee \mathbf{P}\{x < -h(\alpha)\} \leq \exp(-\alpha). \tag{2.6}$$

For inequality (2.3), see exercises 2.22 and 2.23 of Boucheron et al. [11, Chapter 2]. Inequality (2.4) comes from their Theorem 2.3. For (2.5), see Vershynin [100, Lemma 5.5 and Section 5.2.4]. Finally, (2.6) follows from Boucheron et al. [11, p. 29], which takes the form of what is typically called a Bernstein inequality; see their Theorem 2.10 and Corollary 2.11, and Pollard [78, Appendix B] for more background. It should be noted that the use of the term "sub-Exponential" appears frequently in the statistical literature dealing with characterizing heavy-tailed distributions, and in particular in the context of extreme values. A well-known work on evaluating and estimating the heaviness of a distribution's tails is due to Smith [88]. See Goldie and Klüppelberg [44] and the references within for supplementary information.

In this thesis, we make use of a number of well-known families of probability distributions, whose properties span numerous combinations of those discussed in the preceding paragraphs. An introduction to all of these is rather tedious, and thus the details are relegated to section 2.6 at the end of this chapter, with table 2.1 providing a concise summary here.

2.3 Statistical inference and learning

We shall now begin to shift our focus away from methods of evaluating performance, and towards the procedure for determining $\widehat{h}_n$ given sample $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, namely the learning algorithm itself. If the goal is to make $R(\widehat{h})$ as small as possible with the highest possible probability over the random draw of the sample, it goes without saying that any successful algorithm will require information regarding the function $R(\cdot)$. If R were known, learning would amount to the optimization of a known function. Of course, it is precisely because R is unknown that machine learning problems closely parallel the learning tasks of humans, as discussed at the start of Chapter 1. Hence, we must use the data to approximate R directly, or properties of R that will be conducive to its efficient minimization. Since $R(h)$ is a parameter of the distribution of $l(h; \boldsymbol{z})$, an obvious entry point is any statistic defined on the set $\{l(h; \boldsymbol{z}_i)\}_{i=1}^{n}$, readily interpreted as feedback regarding the quality of candidate h. Most simply, the empirical mean of the measured losses comes to mind, and suggests a simple algorithm taking the form

$$\widehat{h}_{\text{ERM}} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} l(h; \boldsymbol{z}_i).$$

This is the principle of empirical risk minimization (ERM),[3] a cornerstone of modern learning theory and virtually all learning algorithms used in practice today [98]. If the sample mean

[3] For any real-valued function f on $\mathcal{H}$, we write $\operatorname{arg\,min}_{h \in \mathcal{H}} f(h) := \{h : f(h) = \inf\{f(g) : g \in \mathcal{H}\}\}$. This could be empty (no solutions), contain one, or even infinitely many solutions. When f is the empirical risk, note that the ERM principle admits any valid solution.

	Symmetric	*Bounded*	*Moment Control*
Arcsine	◯	◯	G
Beta	×	◯	G
Beta Prime	×	×	×
Chi-squared	×	×	E
Exponential	×	×	E
Exponential-Logarthmic	×	×	E
Fisher’s F	×	×	×
Folded Normal	×	×	E
Fréchet	×	×	×
Gamma	×	×	E
Gaussian mixture	×	×	G
Gompertz	×	×	G
Gumbel	×	×	Γ
Hyperbolic secant	◯	×	Γ
Irwin-Hall	◯	◯	G
Laplace	◯	×	E
Log-Logistic	×	×	×
Log-Normal	×	×	×
Logistic	◯	×	E
Maxwell	×	×	G
Pareto	×	×	×
Rayleigh	×	×	G
Semi-circle	◯	◯	G
Student’s t	◯	×	×
Triangle	×	◯	G
U-Power	◯	◯	G
Weibull	×	×	G/E

Table 2.1: Table of distributions and their qualities of interest. Having × under *Symmetric* means that there are parameter settings in which the distribution is asymmetric, while ◯ means symmetry holds for all parameter values. The characters G, E, and Γ respectively correspond to sub-Gaussian, sub-Exponential, and sub-Gamma properties. A × mark in the *Moment Control* column means that higher-order moments are infinite.

is an accurate representation of the risk, then we can expect the ERM solution to succeed. Conversely, when the sample mean poorly approximates the risk, it is naturally poor feedback, and we cannot expect any algorithm depending on such feedback to succeed with high probability. The first question to ask, then, is whether there are plausible scenarios in which the sample mean actually does fail. If an affirmative answer can be made to this question, then we must ask whether an alternative procedure performs better. In this way, a key concept here, and throughout the rest of this thesis, is that of dedicating computational resources to providing the learner with better feedback. It turns out that both questions can be answered in the affirmative, and in fact very precise answers can be given. We begin with some broad historical context, before giving a straightforward technical example to illustrate the key idea.

Estimation of moments is the canonical statistical estimation task, and the literature is unsuprisingly massive. For the case of parametric models, we refer the reader to standard texts for background [22, 61]. The classic paradigm assumes that the underlying distribution of interest, say μ, is known up to a finite number of parameters, and lives in a space $\mathcal{P}(\Theta) =$

$\{p_\theta : \theta \in \Theta\}$, whose elements (typically density functions) can be specified by a subset of finite-dimensional linear space, here Θ. It is in this context that likelihood maximization became the central algorithm design principle, an approach which is cohesive with the notion that all we need to do is "fit" the model to the data.

As computing technology advanced, new problems arose that hightlighted issues with the classical framework; it is hard to justify parametric models and the maximum likelihood methodology when our samples are "contaminated" by errant observations, an idea proliferated by Tukey [95]. The subtext here is that such observations are irrelevant to our original problem of interest, and should be discarded or ignored. In such a case, we would have $\mu = (1-\delta)\mu_0 + \delta\nu$, where μ_0 belongs to a parametric family known to the statistician, and ν is an arbitrary probability controlling the contaminating noise. To deal with problems of this nature and automatically reduce the influence of undesirable data, through the 1960s a new program, borne of the work of Huber [51] who put things on a solid theoretical footing, developed into what is now called robust statistics. Large statistical libraries of software, a rich theory of influence functions, breakdown point analysis, and M/L/R-estimation are among the most important legacies of this period [52, 46].

When we completely abandon the parametric assumption, and consider non-parametric models such as

$$\mathcal{P}_k(a) := \{\mu \in \mathcal{P}(\mathcal{Z}) : \mathbf{E}_\mu |l(h;\boldsymbol{z})|^k \leq a\}, \quad 0 < a \leq \infty$$

how does the situation change? Note that this is the norm in the current era of machine learning, where data sets are complicated and diverse, reducing the applicability of *a priori* domain knowledge akin to the near-omniscient statistician of the mid-20th century. In this setting, the empirical distribution μ_n induced on the sample $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, namely

$$\mu_n(Z) := \frac{1}{n}\sum_{i=1}^{n} I\{\boldsymbol{z}_i \in \mathcal{Z}\}, \quad \text{for each event } Z \subseteq \mathcal{Z}$$

was made the focal point. Using $\mu_n \approx \mu$, the strategy naturally shifted to minimizing $\mathbf{E}_{\mu_n} l(h;\boldsymbol{z})$, a computable quantity which is a natural objective function to be minimized by the computer. As is lucidly discussed by Vapnik [98, Ch. 1], the original theoretical impetus for this approach is undoubtedly the uniform consistency result of Glivenko and Cantelli for the empirical distribution function on $\mathbb{R}$, which for any fixed $h \in \mathcal{H}$ says that taking $n \to \infty$, we have

$$\sup_{\alpha \in \mathbb{R}} |\mathbf{E}_{\mu_n} l(h;\boldsymbol{z}) - \mathbf{E}_\mu l(h;\boldsymbol{z})| \to 0$$

with probability one. Keeping the risk minimization task in mind, using the fact that a wide class of distributions are tightly "concentrated" about their means [92, 11], whenever this holds for $\mathbf{E}_{\mu_n} l(h;\boldsymbol{z})$, a random variable depending on the sample, it is natural to use this canonical estimate.

Unfortunately, as we will show in the following paragraphs, the class of data distributions where the loss is well-behaved does not nearly account for rich models like $\mathcal{P}_k(a)$. This presents a clear technical problem of interest: developing computable moment estimators which have desirable properties under weak assumptions. In addition to this, as applied statistics and machine learning practictioners continue to diversify, it seems unreasonable to expect a large portion of users to check model assumptions, or even be aware of them, which has substantial well-understood risks [41], especially as socio-economic data becomes more widely available. Given this context, it appears fruitful to carry out some modeling decisions on the "back-end," that is to pre-program them into general-purpose learning algorithms. This is precisely the motivation for designing more data-robust algorithms, as discussed in the previous section. Let us now provide some more formal context for the technical problem.

For simplicity, consider the case of $x \sim \mu$ on the real line, with sample $x_1, \ldots, x_n$ and non-parametric model $\mathcal{P}_k(a) = \{\mu : \mathbf{E}_\mu |x|^k \leq a\}$. Writing $\bar{x}_n = n^{-1} \sum_{i=1}^n x_i$, one of the more prominent classical facts is that for any $0 < a < \infty$, one has

$$\sup_{\mu \in \mathcal{P}_2(a)} \mathbf{E}(\bar{x}_n - \mathbf{E}_\mu x)^2 = \inf_{\chi} \sup_{\mu \in \mathcal{P}_2(a)} \mathbf{E}(\chi - \mathbf{E}_\mu x)^2$$

where the infimum over χ is taken over effectively all measurable estimators [101, Section 3], and the unmarked expectation $\mathbf{E}$ is taken with respect to the sample [61, Ch. 4]. This is a so-called minimax optimality result for the sample mean: in the worst-case situation in terms of average deviations between $\bar{x}_n$ and the target $\mathbf{E}_\mu x$ to be estimated, it outperforms all rivals.

On the other hand, a clear issue can be raised: are average deviations a useful description of performance? Going back to the PAC learning framework, note that we seek high-probability guarantees over random draw of the sample, rather that statements about the average behaviour. One clear reason for doing this is because there is a distinct possibility that in "integrating out" the sample, we may be undervaluing the potential impact of errant observations in any given (finite) realization of $x_1, \ldots, x_n$ (correspondingly, the losses $\{l(h; \boldsymbol{z}_i)\}_{i=1}^n$). Following Catoni [24] and Hsu and Sabato [50], this can be made more concrete with an example on $\mathbb{R}$. Consider a class of "good algorithms", written $\mathbb{A}$, defined for data generated from distributions in a class $\mathcal{P}$. Here an algorithm is "good" if it enjoys a var_μ-modulated confidence interval of order $O(n^{-1/2})$, uniformly over $\mu \in \mathcal{P}$. More precisely, for $\widehat{x}$ to belong to $\mathbb{A}$, we require that given large enough n, for all $\forall\, \mu \in \mathcal{P}$, we have

$$\mathbf{P}\{|\widehat{x}_n - \mathbf{E}_\mu x| \leq b_\mu(\widehat{x})\} \geq 1 - 2\delta, \quad b_\mu(\widehat{x}) \coloneqq c_0 c_\delta(\widehat{x}) \sqrt{\frac{\text{var}_\mu x}{n}}.$$

Here c_0 is an arbitrary constant, and $c_\delta(\cdot)$ is a factor that depends on δ and the choice of algorithm $\widehat{x}$. The dependence of $b_\mu(\widehat{x})$ on n and δ is suppressed in the notion for readability. When the data is particularly "well-behaved," there is no issue with the classical approach. To see this clearly, observe the fact that for any $\widehat{x} \in \mathbb{A}$ and sample size n,

$$b_\mu(\widehat{x}) \geq b_\mu(\bar{x}), \quad \forall\, \mu \in \{\mathrm{N}(u, \sigma) : u \in \mathbb{R}, \sigma > 0\}. \tag{2.7}$$

This says that when $\mathcal{P}$ is the set of all Gaussian distributions on $\mathbb{R}$, among all the good algorithms populating $\mathbb{A}$, none are superior to the sample mean $\bar{x}$. When $\widehat{x} = \bar{x}$ and $\mu = N(u, \sigma)$ for any $u \in \mathbb{R}$, we can evaluate the bound b_μ exactly. It takes the form

$$b_\mu(\bar{x}) = \Phi^{-1}(1 - \delta) \sqrt{\frac{\sigma^2}{n}}$$

where $\mu = \mathrm{N}(u, \sigma)$, and Φ is the Normal distribution function. As noted by Devroye et al. [30], sending $\delta \to 0$ we get $b_\mu(\bar{x}) \sim \sqrt{(2 \log(\delta^{-1})\sigma)/n}$. Since virtually any non-parametric model will at least include the Gaussian model as a subset, this can be considered a natural lower bound, i.e., an optimal performance benchmark for the members of $\mathbb{A}$ given a more general $\mathcal{P}$. Based on what we have seen thus far, if we have a very restrictive parametric model, the empirical mean is *optimal*, just as it was in the minimax expected loss framework. Assuredly, this does not match the experience of most practitioners, which suggests that at the very least, the sample mean should be sub-optimal when extreme values are relatively likely, i.e., when the distribution is "heavy-tailed." An important question: under more general models within this new (PAC-type) evaluation paradigm, will issues with the empirical mean become more salient?

An affirmative answer can be made to this question. To see this, consider the rich non-parametric model $\mathcal{P}_2 = \bigcup_{a=1}^{\infty} \mathcal{P}_2(a)$, a massive model by any standard, and note that Chebyshev's inequality implies

$$\mathbf{P}\left\{ |\overline{x}_n - \mathbf{E}_{\mu} x| > \sqrt{\frac{\operatorname{var}_{\mu}(x)\delta^{-1}}{2n}} \right\} \leq 2\delta, \qquad \text{for any } \mu \in \mathcal{P}_2.$$

Since this guarantee only gives us an upper bound, one is naturally wary that the bound could be loose, and potentially not an accurate portrayal of the accuracy of the estimate given by $\overline{x}_n$. In particular, the linear dependence on $1/\delta$ is undesirable when the confidence level $1-\delta$ is high. For what kind of distribution would we expect the classical approach begin to fail? We know through experience that if μ assigns a relatively large amount of density far away from regions of central tendency, errant observations can wreak havoc with estimates based on $\overline{x}_n$. This empirical insight is essentially correct. Consider the following example based on a proof given by Catoni [24]. For arbitrary but fixed integer n, from large class $\mathcal{P}_2$, consider a class of symmetric distributions with support of three points $\{u_0 - p, u_0, u_0 + p\}$ denoted $\nu(p)$, and defined by

$$\nu(p)\{u_0\} = 1 - \frac{1}{n^2 p^2}$$

where each distribution is specified by parameter $p \geq 1/n$. Of interest to us is the fact that if $x \sim \nu(p)$, then as $x_1, \ldots, x_n$ is an independent sample, we have that for any $0 < \delta \leq 1/(2e)$,

$$\mathbf{P}\left\{ |\overline{x}_n - \mathbf{E}_{\nu(p)} x| \geq \sqrt{\frac{\delta^{-1}}{2n}\left(1 - \frac{2e\delta}{n}\right)^{n-1}} \right\} \geq 2\delta.$$

This lower bound says that when the distribution is $\mu = \nu(p)$ for any valid p, there exists an unavoidable "bad event" of non-zero probability, where the general upper bound on $|\overline{x}_n - \mathbf{E}_{\mu} x|$ is in fact *tight* in terms of dependence on δ and n. Since this lower bound essentially matches the upper bound, the confidence interval is as poor as it can possibly be on the large class $\mathcal{P}_2$. Intuitively, this is an example of a class where the sample mean looks to break down. What can we say about the qualities of this class $\{\nu(p) : p \geq 1/n\} \subset \mathcal{P}_2$? Note that if $\nu(p)\{u_0\} = 1 - \gamma$, then by definition $p = \sqrt{\gamma^{-1}}/n$, i.e., the breadth of the distribution grows quickly as γ is taken smaller, and values in the tails grow farther from the mean. This can be placed in stark contrast with other possible symmetric distributions with three-point support, whose breadth may be comparatively small. The message here is that when data comes from distributions with a heavy-tailed form, they are not conducive to good estimates via the sample mean.

There remain two key questions: (1) do there exist members of $\mathbb{A}$ that realize more desirable error bounds than the sample mean over non-parametric models like $\mathcal{P}_k$ for small k, and if so, (2) are they computationally tractable? Now coming full circle and returning to the path-breaking works cited at the end of the previous section, the existence of such dominant algorithms in the class $\mathbb{A}$, as well as other related classes, has been proved by Catoni [24], and more recently by Devroye et al. [30]. The former work considers an elegant new class of readily computable M-estimators [24, Section 2], though re-scaling of observations relies on variance oracles in order to be done efficiently. Some sophisticated adaptive routines for re-scaling without a variance oracle are given [24, Section 3–4], though these require either a kurtosis oracle or an adaptive procedure not amenable to computation. The latter work considers broad several model and algorithm classes, gives lower bounds on estimation error (implying an optimal performance criterion), and provides some artificial routines which have near-optimal formal properties. Unfortunately, due to their design, these routines are not computationally amenable, and efficient methods for realizing analogous performance has been left as an open problem. Our interest in this thesis is an investigation of how more robust statistics of based on observed loss values $\{l(h;\boldsymbol{z}_i)\}_{i=1}^{n}$, that is more robust feedback, can be used productively in computationally congenial optimization procedures.

2.4 Objective-based feedback

Bridging the conceptual gap between feedback and response is easy when the former is provided in the form of an *objective function* for the latter to minimize. This is precisely the case when we invoke the ERM principle: the empirical risk is an objective function, and typically the response is an as-yet unspecified procedure for actually carrying out the optimization. In close relation to the content of the previous section, the design of an objective function can indeed be the central step in engineering a learning algorithm. In this section, we momentarily restrict our focus to a specific learning task, with the goal of illustrating both the important role (in learning) that such functions can play, and the wide variety of technical approaches taken to this problem in the literature.

Here we look at the "regression" problem, under a linear model, with non-parametric assumptions on the data distribution. Our data takes the particular form $\boldsymbol{z} = (\boldsymbol{x}, y)$, and the goal is to predict output y given a novel instance $\boldsymbol{x}$. We are given n observations $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, all independent copies of $(\boldsymbol{x}, y) \sim \mu$, taking values in $\boldsymbol{x} \in \mathbb{R}^d$, $y \in \mathbb{R}$. Compactly denote $\boldsymbol{y} := (y_1, \ldots, y_n)$ and X for the $n \times d$ "design matrix" of inputs. The classical situation assumes $n > d$, but an important and interesting case is that where $n \ll d$, given a sparsity scenario. The underlying process is assumed, for simplicity, to be

$$y = \boldsymbol{x}^T \boldsymbol{w}^* + \epsilon, \quad \mathbf{E}_\mu \, \epsilon = 0, \, \mathbf{E}_\mu \, \epsilon^2 < \infty.$$

Here the "signal" is determined by some pre-fixed but unknown $\boldsymbol{w}^* \in \mathbb{R}^d$. In addition to the second moment being finite, we assume the noise/residual term ϵ is independent of all components of $\boldsymbol{x}$. The notion of data-robustness here will correspond directly to the strength of additional assumptions that are required of the noise distribution in order to guarantee satisfactory performance. Sparsity assumptions are made precise through explicit constraints on the effective dimension of the underlying model. For example, constraining the value of

$$\|\boldsymbol{w}^*\|_q = \left(\sum_{j=1}^{d} |w_j^*|^q \right)^{1/q}, \quad q \in (0, 1)$$

enforces a natural form of sparsity, with the strongest and most common case being in the limit of $q \to 0$, where one constrains $\|\boldsymbol{w}^*\|_0 := \sum_{j=1}^{d} I\{w_j^* \neq 0\}$.

The problem formulated above has a long history, and spans many fields, often with distinct research goals. Here we briefly review the basic ideas and key literature. This task can be cast as a special case of a more general problem, namely that of approximating some functional relationship, given inputs $\boldsymbol{x} \in \mathbb{R}^d$ and outputs $y \in \mathbb{R}$, as a linear combination of "basis" functions $\phi : \mathbb{R}^d \to \mathbb{R}$. In practice, these functions are selected from a finite collection, say

$\{\phi_1, \ldots, \phi_k\}$, assumed to contain a sufficient variety of functions. We will have

$$y = \sum_{j=1}^{k} w_j \phi_j(\boldsymbol{x}) + \epsilon(k)$$

where $\epsilon(k)$ represents some residual, due to only having an incomplete set of input/output pairs available from which to infer the functional relationship. The use of systems of simple and convenient functions to approximate complex relations is ubiquitous; a canonical example when $d = 1$ is the system $\{\phi(u) = e^{iju}/\sqrt{2\pi}\}_{j=1}^{k}$, which may be used to construct the basic Fourier series handled in elementary analysis [94]. Our general problem remains: for some $\varepsilon > 0$ benchmark and an appropriate norm $\|\cdot\|$, to achieve $\|\epsilon(k)\| \leq \varepsilon$, how should one set the free parameters $\boldsymbol{w} = (w_1, \ldots, w_k)$ based on data? It may be impossible, in which case there is no issue. In suprisingly many cases, however, such problems are soluble. Some illustrative examples appear in the signal processing literature, where the utility of systems of sine waves, Gabor functions, and wavelets, to name a few examples [29], are well-known. In such problems, using the full expressive power of k basis functions results in a large amount of redundancy, and far more free parameters than are necessary to achieve ε-accuracy. Furthermore, when examples are limited, with so many parameters one faces an ill-posed problem [98, Ch. 1], and in practice one faces a high risk of arriving at sub-standard solutions.

As put forward in the highly influential work of Chen et al. [26], a useful algorithm design principle follows from the points made above. One should strive for ε-accuracy using the smallest number of basis functions possible, i.e., a “sparse” approximation of the functional relation of interest. That is, algorithms should be explicitly encouraged to discard as many basis functions as possible, by setting many weights at or near zero. The linear regression problem formulated above coincides with the case of $k = d$, and $\phi_l(\boldsymbol{x}) = x_l$. Given n examples, one seeks to approximate $\boldsymbol{y}$ as a superposition of $\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(d)}$, the columns of X. To enforce sparsity, the approach of Chen et al. [26], called (atomic) basis pursuit, advocates the use of an ℓ_1 penalty to do this. Written explicitly, the routine is

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad \|\boldsymbol{y} - X\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1$$

for some pre-fixed $\lambda > 0$. It is worth noting that in the signal processing community, this strand of research continued and developed into a major body of work. In the “compressed sensing” literature originating with Donoho [32], one important problem is that of explicitly finding the sparsest representation of $\boldsymbol{y}$ using the columns of X, namely executing

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad \|\boldsymbol{w}\|_0, \quad \text{s.t. } \boldsymbol{y} = X\boldsymbol{w}.$$

Interestingly, Donoho [31] showed that when a solution exists, it is often unique, and in fact can be obtained in a computationally congenial way, namely by running

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad \|\boldsymbol{w}\|_1, \quad \text{s.t. } \boldsymbol{y} = X\boldsymbol{w}.$$

This has the substantial advantage of being a convex program. Candès and Tao [21] showed that in the noiseless case, by solving the ℓ_1-minimization problem, one can exactly recover the optimal solution, using only a finite sample. In the case of Gaussian noise, Candès and Tao [20] introduced the “Dantzig selector,” which proposes a new algorithm, namely the ℓ_1 norm minimization plus a novel constraint. In addition, they also demonstrate that near-optimal solutions can be obtained even under noisy observations, using routines which can be readily implemented.

It is certainly possible to interpret this approach to algorithm design from the perspective of the statistician. Taking the classical approach of minimizing $\|\boldsymbol{y} - X\boldsymbol{w}\|_2^2$ can be risky, since the squared error can be sensitive to noisy, errant observations. If our learning procedure "overfits" in the sense that algorithm output is unduly influenced by unimportant random idiosyncrasies in the n-sized sample, then we cannot reasonably expect strong predictive capabilities off-sample. In addition, when d is large, it is far easier to interpret major contributions from a few covariates, rather than minute contributions from a plethora of covariates. Breiman [17] proposed the "non-negative garrote," namely the constrained convex program

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \; \|\boldsymbol{y} - X\boldsymbol{w}\|_2^2 \,, \quad \text{s.t. } \|\boldsymbol{w}\|_1 \leq t, \;\; w_l \geq 0, \, l \in [d].$$

The non-negative weights make interpreting relative contribution easy, and the ℓ_1 constraint encourages sparsity. Motivated by this work, Tibshirani [93] removed the non-negativity, and provided lucid analysis and novel computational routines, resulting in the well-known LASSO procedure. This work has obvious links to the developments in signal processing noted above; the work of Bickel et al. [9] compared the estimation behaviour of the Dantzig selector with that of the LASSO, and showed that they have fundamental similarities. As a part of the formal analysis, they introduced the "restricted eigenvalue conditions," which have become a mainstay of high-dimensional statistical analysis ever since. Over the years, a tremendous number of extensions, improvements, and analyses of ℓ_1-penalized ERM algorithms has been carried out. Important theoretical foundations have been laid by work such as Knight and Fu [55], Zhao and Yu [106], Donoho et al. [33], Meinshausen and Yu [67], Koltchinskii [57]. Influential computational innovations have come from Osborne et al. [76], Efron et al. [36], Friedman et al. [42], Wu and Lange [105] to name a few.

Despite the impressive advances described above, these feedback mechanisms do not necessarily provide solutions to our problem of interest. In the compressed sensing context, when our observations are corrupted by non-Gaussian noise, the problem becomes more difficult and existing solutions may fail [103]. While regularization can be used to mitigate the impact of overfitting due to, for example, a quadratic error term, setting the additional weight parameter can be extremely sample-sensitive and time-intensive, and indeed may be insufficient without prior information regarding model constraints. It is within this context that many researchers have investigated more robust sparse regression routines, in the sense that they can be applied to tasks with data arising from a larger class of distributions than classical methods can. In keeping with our focus on objective functions, we introduce some well-known work which has pursued "robustification" in this vein.

Many studies have focused on designing more sophisticated loss functions for use in the objective. A general form for the objective function can be given as

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \; \frac{1}{n} \sum_{i=1}^{n} \rho \left(y_i - \boldsymbol{x}_i^T \boldsymbol{w} \right) + \lambda \Omega(\boldsymbol{w})$$

where ρ is a generalized loss, and Ω is a penalty designed to induce sparsity in the estimate. Starting with the simplest case, recall the classic dichotomy between ordinary least squares ($\min_{\boldsymbol{w}} \|\boldsymbol{y} - X\boldsymbol{w}\|_2^2$) and least absolute deviations ($\min_{\boldsymbol{w}} \|\boldsymbol{y} - X\boldsymbol{w}\|_1$) in the low-dimensional case. In the former, one effectively seeks the conditional mean $\mathbf{E}_\mu(y; \boldsymbol{x})$, while in the latter one seeks the conditional median $\text{med}_\mu(y; \boldsymbol{x})$. The median is insensitive to the distribution tails, and thus given a finite sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the influence of errant observations is limited. Wang et al. [102] consider the LAD-LASSO, namely the case of

$$\rho(u) = |u|, \quad \Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$$

and provide both asymptotic analysis of the resulting estimator, as well as introductory simulations using heavy-tailed noise. Under negligible noise, residuals may tend to be very small, and the ℓ_1 loss will over-penalize such deviations, introducing potentially significant bias; one is faced with a difficult choice between two extremes.

This can be continuously modulated by introducing parameter $l \in [1,2]$ and setting $\rho_l(u) = |u|^l/l$. The properties of this function have been investigated by Rey [80, Section 6] in the context of robust (non-penalized) regression. Rather than focusing on values between the mean and median, another approach is to look at the spectrum of quantiles for the distribution of interest (here, y conditioned on $\boldsymbol{x}$). Quantile regression [56] generalized the problem using a new loss

$$\rho_\tau(u) := \begin{cases} \tau u, & u > 0 \\ (\tau - 1)u, & u \leq 0 \end{cases} \quad \tau \in (0,1)$$

and minimizing $\mathbf{E}_\mu \, \rho_\tau(y - \boldsymbol{x}^T \boldsymbol{w})$, noting that any τ-level quantile of $y|\boldsymbol{x}$ is a solution. This has been considered for non-parametric tasks in the machine learning literature [91], and also formally treated in the penalized high-dimensional regression setting by Belloni and Chernozhukov [8]. In their work, the procedure is defined using the objective function

$$\rho(u) = \rho_\tau(u), \quad \Omega(\boldsymbol{w}) = \frac{\sqrt{\tau(1-\tau)}}{n} \sum_{j=1}^{d} m_j |w_j|$$

where the m_j are extra weights to be estimated from the data, a notion which has been in the literature for some time [39]. Denoting the algorithm output by $\widehat{\boldsymbol{w}}_n$, their results include near-optimal convergence rates of $\|\widehat{\boldsymbol{w}}_n - \boldsymbol{w}_0\|_2$ as $n \to \infty$, which hold *uniformly* over many choices of quantile level τ. These results are highly suggestive of the utility of replacing the ℓ_2 loss with another, more robust choice. A very similar problem was considered by Fan et al. [37], where the penalty was revised slightly to

$$\rho(u) = \rho_\tau(u), \quad \Omega(\boldsymbol{w}) = \|\boldsymbol{v} \circ \boldsymbol{w}\|_1$$

where $\circ$ denotes the Hadamard product, $\boldsymbol{v} \circ \boldsymbol{w} := (v_1 w_1, \ldots, v_d w_d)$, and they pay close attention to a strategy for adaptively determining the $\boldsymbol{v}$ weights based on initial estimates. Consistency guarantees are given, under much weaker assumptions on μ than related analysis done by Bradic et al. [15]. In addition, their Proposition 1 gives a salient example of how the traditional ℓ_2-based LASSO can break down under heavy tails.

Unfortunately, when we have minimal prior information on μ and small samples, selecting the quantile τ to use is a non-trivial problem, just as the OLS versus LAD decision is difficult. It is natural, then, to seek out alternative parameters of the target distribution $y|\boldsymbol{x}$, which closely approximate $\mathbf{E}_\mu(y; \boldsymbol{x})$ when outliers are negligible, but are less sensitive to distribution tails when those tails become heavy. The idea is that empirical estimates of such parameters will, in turn, be less sensitive to random idiosyncrasies in the sample. One very interesting work which adopts this approach is that of Lambert-Lacroix and Zwald [59], which looks at using the classic choice of the function of Huber [51, 52], defined by

$$\rho_M(u) := \begin{cases} u^2, & |u| \leq M \\ 2M|u| - M^2, & |u| > M \end{cases}$$

and re-scaling observations simultaneously, using a program specified by

$$\rho(u,s) = \begin{cases} ns + \rho_M\left(\frac{u}{s}\right) s, & s > 0 \\ 2M|u|, & s = 0 \\ +\infty, & s < 0 \end{cases}, \quad \Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1.$$

where in practice $u = y - \boldsymbol{x}^T \boldsymbol{w}$ and $(\boldsymbol{w}, s)$ are the parameters over which one minimizes. Formal results regarding the asymptotic distribution of the estimator, as well as detailed numerical experiments using simulated and real data, and a discussion of the utility of information criteria were given. The same algorithm using a pre-fixed scale (i.e., minimize over $\boldsymbol{w}$ only using ρ_M) was given a lucid formal treatment in a pre-print by Fan et al. [38], for which sharp bounds on ℓ_2-error under finite-sample estimates under composite gradient-type updates are given, under weak assumptions on the noise/residual. Initial results were given for an adaptation of the M-estimator of Catoni [24] as well, however no strategies for setting s and λ in this problem were offered.

Finally, we note that some rather unique approaches to the problem have also been proposed in recent years. Some interesting methods have come from the signal processing and computer vision communities. One nice example from Wright and Ma [103] analyzes the algorithm

$$\min_{(\boldsymbol{w},\boldsymbol{e}) \in \mathbb{R}^{d+n}} \|\boldsymbol{w}\|_1 + \|\boldsymbol{e}\|_1, \quad \text{s.t. } \boldsymbol{y} = X\boldsymbol{w} + \boldsymbol{e}$$

where the residual is treated as an additional parameter to be controlled. Rather strict assumptions are still required on the underlying distribution, but the practical utility in several tasks was clearly demonstrated [104]. A similar strategy appears in the work of Nasrabadi et al. [70] and Nguyen and Tran [74], where they use

$$\min_{(\boldsymbol{w},\boldsymbol{e}) \in \mathbb{R}^{d+n}} \|\boldsymbol{y} - X\boldsymbol{w} - \boldsymbol{e}\|_2^2 + \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{e}\|_1$$

and provide some novel insights into the role that the ratios of n, d, and $\|\boldsymbol{w}^*\|_0$ play on the performance of this "extended" LASSO.

In recent years, novel extensions of the median-of-means strategy to high-dimensional problems by Minsker [68] and Hsu and Sabato [49, 50] have received attention. The basic idea is to partition the data into k segments, say with indices $[n] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_k$, and to run the usual LASSO on each independent segment, setting

$$\widehat{\boldsymbol{w}}^{(l)} \in \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i \in \mathcal{I}_l} (y_i - \boldsymbol{x}_i^T \boldsymbol{w})^2 + \lambda \|\boldsymbol{w}\|_1, \quad l = 1, \ldots, k.$$

The estimate is then set as

$$\widehat{\boldsymbol{w}}_n := \operatorname{med}\{\widehat{\boldsymbol{w}}^{(1)}, \ldots, \widehat{\boldsymbol{w}}^{(k)}\}$$

where the "median" here is defined using the geometric median on $\mathbb{R}^d$ [99], namely

$$\operatorname{med}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\} := \arg\min_{\boldsymbol{u} \in \mathbb{R}^d} \sum_{l=1}^{k} \|\boldsymbol{u} - \boldsymbol{u}_l\|_2.$$

Finite-sample ℓ_2-error bounds for this routine are given by Minsker [68, Section 4.3], under very weak assumptions such as a few finite moments of the noise and some regularity conditions on the carrier matrix. Comparable results are given by Hsu and Sabato [50, Section 6.1], using a different routine for selecting from the candidates $\mathcal{W} := \{\widehat{\boldsymbol{w}}^{(1)}, \ldots, \widehat{\boldsymbol{w}}^{(k)}\}$. Their Algorithm 2 says to set

$$\begin{aligned} r(l) &:= \min\{r \geq 0 : |B_r(\widehat{\boldsymbol{w}}^{(l)}) \cap \mathcal{W}| > k/2\} \\ \star &:= \arg\min_{l \in [k]} r(l) \\ \widehat{\boldsymbol{w}}_n &:= \widehat{\boldsymbol{w}}^{(\star)} \end{aligned}$$

where $B_r(\boldsymbol{w}) = \{\boldsymbol{u} : \|\boldsymbol{u} - \boldsymbol{w}\|_2 \leq r\}$. This is also a very appealing idea, effectively choosing the candidate that is close to "most" of the others, a natural generalization of the median notation. One concern is that under small sample sizes there is potential for significant bias, limiting utility for smaller samples, and conversely, it may be that once the sample size is large enough for numerous partitions can be made, less sophisticated techniques may suffice. In any case, the formal properties, intuition, and ease of implementation are attractive, and seriously warrant further investigation.

2.5 Implementation of learning rules

In sections 2.3 and 2.4, we considered the statistical impact of different feedback mechanisms, and highlighted some specific examples of procedures centred around an explicit objective function, all closely tied to the ERM learning principle. To gain a more thorough understanding of the behaviour, generalization ability, and fundamental limitations of learning algorithms, however, we must make concrete the actual implementation, since the only statistical estimates the learner will ever have in practice are those that it can compute. In the examples of section 2.4 using objective functions, the chief question of implementation revolves around the method by which we minimize the objective. This naturally leads us to the need to account for *optimization error*, namely the error incurred as a result of not being able to reach an optimal value in a finite number of steps. In addition, when multiple optima exist, there may be significant repurcussions in terms of the generalization performance of "good" versus "bad" optima. We begin this section by highlighting the statistical gap that can exist between distinct optima, and then shifting our focus to optimization error, we review some important optimization procedures studied in the machine learning literature.

Keeping with the notation of section 2.1, we start by looking at a special case that follows from Feldman [40, 3.1]. For concreteness, let $\mathcal{H} = \{h \in \mathbb{R}^d : \|h\|_p \leq 1\}$, the unit ball in ℓ_p norm, for $1 \leq p \leq \infty$. Let $l(\cdot;\boldsymbol{z})$ be 1-Lipschitz on $\mathcal{H}$ in the ℓ_p norm. Then, writing $\widehat{R}(h) := n^{-1}\sum_{i=1}^{n} l(h;\boldsymbol{z}_i)$, where $\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \sim \mu$ are our independent observations, we have that

$$n \geq O\left(\frac{d\log(d/(\varepsilon\delta))}{\varepsilon^2}\right) \implies \mathbf{P}\left\{\sup_{h\in\mathcal{H}} |R(h) - \widehat{R}(h)| \geq \varepsilon\right\} \leq \delta$$

where $\mathbf{P}$ measures the probability over the random draw of the n-sized sample. As one would expect, for any valid ERM estimate $\widehat{h}$ minimizing $\widehat{R}(\cdot)$, on this high-probability event we have

$$\begin{aligned} R(\widehat{h}) - R^* &= R(\widehat{h}) - \widehat{R}(\widehat{h}) + \widehat{R}(\widehat{h}) - R^* \\ &\leq 2\sup_{h\in\mathcal{H}} |R(h) - \widehat{R}(h)| \\ &\leq 2\varepsilon \end{aligned}$$

where the inequality follows from the optimality of $\widehat{h}$. Thus ignoring log terms, the sample complexity of any ERM solution for this setting is bounded above by $O(d/\varepsilon^2)$. We point out the dependence on d here for comparison puposes; Theorem 3.3 of Feldman [40] provides a *lower* bound on the sample complexity of ERM in the problem setting just described, with $p = 2$ for simplicity. It is shown that there exists a distribution μ and loss function $l(\cdot;\boldsymbol{z})$ satisfying the assumptions of this scenario, where a "bad" ERM implementation $\widehat{h}$ is such that

$$n \leq d/6 \implies \mathbf{P}\left\{R(\widehat{h}) - R^* > 1/4\right\} > 1/2,$$

or more generally, that the sample complexity of a poor ERM implementation is bounded below by $O(d/\varepsilon)$, showing that in the worst case, the sample size must scale linearly with d for the generic ERM. The notion of a "poor" implementation sounds a bit subtle, but in fact all this means is that one can construct an algorithm which, given any sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, returns a valid ERM estimate $\widehat{h}$, which is poor in the sense of the lower bound just given. Since an unimplemented ERM procedure naturally includes this contingency, only very weak guarantees can be made. For comparison, Theorem 3.2 of the cited paper shows that there exist alternative algorithms which, in the same setting, have sample complexity which is free of d. To better understand when ERM (or any other learning principle) succeeds, and what its limitations are, it is now clear that the nature of its implementation cannot be left abstract.

Having established via example the notion that implementation plays an important role in statistical learning, how are optimization tasks typically solved in the machine learning community? Due to the simplicity and theoretical groundwork available for ERM, most researchers formulate the problem as a finite-sum minimization,

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} f_i(h)$$

where the $f_i : \mathcal{H} \to \mathbb{R}_+$ are random functions drawn independently from a common distribution. When $f_i(h) = l(h; \boldsymbol{z}_i)$, this problem reduces to the usual empirical risk scenario, though the setting is far more general, including all varieties of regularization, for example $f_i(h) = l(h; \boldsymbol{z}_i) + \lambda \|h\|^k$ for some norm $\|\cdot\| : \mathcal{H} \to \mathbb{R}_+$, $\lambda > 0$ and $k \in \mathbb{N}$. Different assumptions on the "sample" functions $\{f_i\}_{i=1}^{n}$ leads to different strategies, and here we take time to introduce some of the more powerful and influential ideas from the literature.

Under the assumption that the objective is differentiable, the vast majority of implementations in machine learning use "first order descent" methods. Writing the objective as $f(h) = \sum_{i=1}^{n} f_i(h)$, and its differential[4] at $h \in \mathcal{H}$ by $f'_h : \mathcal{H} \to \mathbb{R}_+$, this terminology comes from the first order approximation

$$f(h+u) = f(h) + f'_h(u) + o(\|u\|)$$

used to derive iterative procedures that result in sequences $\{h_{(t)}\}_{t=0}^{\infty}$ which ideally satisfy $f(h_{(t+1)}) < f(h_{(t)})$ for each $t \geq 0$, which is to say that they descend the objective in a monotone fashion, hence the naming. To derive a natural update algorithm, invoke the idea of "steepest descent", which says that we should set the update direction u such that the reduction in value of the first-order approximation of f at h, namely $f(h) + f'_h(u) \approx f(h+u)$, is as large as possible. Assuming $\mathcal{H}$ is an inner product space,[5] the differential at h with increment g takes the form $f'_h(u) = \langle \nabla f(h), g \rangle$, where $\nabla f(h)$ denotes the gradient of f at h, a vector of partial derivatives when $\mathcal{H}$ happens to be a subset of Euclidean space. Writing $\widehat{f}(u) \coloneqq f(h) + f'_h(u)$ for pre-fixed $h \in \mathcal{H}$, we seek to minimize

$$\widehat{f}(u) - \widehat{f}(0) = \langle \nabla f(h), u \rangle, \quad s.t.\ \|u\| = 1.$$

The norm contraint here lets us identify the optimal direction; as long as this is clear, to finalize an update will simply require re-scaling the unit update vector. Applying the Cauchy-Schwartz inequality for the two cases of u and $-u$ with unit norm, we have

$$-\|\nabla f(h)\| \leq \widehat{f}(u) - \widehat{f}(0) \leq \|\nabla f(h)\|.$$

[4] For background on differentiation, see Rudin [83, Ch. 9] for Euclidean space, and Luenberger [65, Ch. 7] for more general normed linear spaces.

[5] Analogous arguments can easily be given for general metric spaces, minimizing $f'_h(u)$ in u on the corresponding unit ball [14, Sec. 9.4].

Setting $u = -\nabla f(h)/\|\nabla f(h)\|$, we have $\widehat{f}(u) - \widehat{f}(0) = -\|\nabla f(h)\|$, achieving the lower bound. To see that this is indeed optimal, if there existed an alternate u' with $\|u'\| = 1$ and $\widehat{f}(u') - \widehat{f}(0) < -\|\nabla f(h)\|$, this would imply $\widehat{f}(-u') - \widehat{f}(0) > \|\nabla f(h)\|$, contradicting the upper bound just given. Updating the first-order approximation in the direction of steepest descent (on an inner product space) and re-scaling at iteration t with $\alpha_{(t)} > 0$, we have the popular gradient descent (GD) update

$$h_{(t+1)} = h_{(t)} - \alpha_{(t)} \sum_{i=1}^{n} \nabla f_i(h_{(t)}),$$

which follows from setting the update vector to $u = \alpha_{(t)}(-1)\nabla f(h)$, and the linearity of the gradient operator. Let us take a moment to recapitulate: in the special case of $f_i(h) = l(h; \boldsymbol{z}_i)$, this update procedure is none other than a GD implementation of the ERM learning principle. As most learning algorithms in widespread use today are variants of the ERM-GD procedure, this represents a useful starting point for comparing and contrasting existing techniques.

We now introduce some key variants of the first-order steepest descent procedure studied and used in the machine learning community. While gradient descent has been studied in the context of numerical optimization for over sixty years [28], a seminal application of the technique is found in the "back-propagation" algorithm for training feed-forward neural networks with logistic units, due to Rumelhart et al. [84, 85]. One important idea found in their work is an "online" variant in which the learner is presented with examples sequentially, resulting in the update

$$h_{(t+1)} = h_{(t)} - \alpha_{(t)} \nabla f_{I(t)}(h_{(t)})$$

where $I(t)$ denotes the sample index. This can be readily implemented in the batch setting as well, where n observations $f_1, \ldots, f_n$ are given, and the indices $I(t)$ are randomized. This randomization can be done with replacement, where $I(t) \sim \text{Unif}\{1, \ldots, n\}$ independent of the step number t, or without replacement, where $I(t)$ cycles over some permutation of $\{1, \ldots, n\}$. One natural motivation for this technique comes from the fact that when $I(t)$ follows the uniform distribution, the expectation (conditioned on $h_{(t)}$) is

$$\mathbf{E}\, \nabla f_{I(t)}(h_{(t)}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(h_{(t)}),$$

the "full gradient" computed on the entire sample. Noting that initialized at some $h_{(0)} \in \mathcal{H}$, and fixing $\alpha_{(t)} = \alpha$ for simplicity, we have

$$\begin{aligned} h_{(t+1)} &= h_{(0)} - \alpha \sum_{k=0}^{t} \nabla f_{i(k)}(h_{(k)}) \\ &\approx h_{(0)} - \alpha(t+1)\, \mathbf{E}\, \nabla f_{I(t)}(h_{(t)}) \end{aligned}$$

which suggests that at the very least, after $t \approx n$ iterations with $\alpha \propto 1/(t+1)$, the variations should iron out such that it approximates one update step using the full gradient. There is clear appeal here when, for example, n is large and many samples are similar to one another. In this case, as long as we do not run into wildly errant observations, presumably a good approximation to the full gradient update can be achieved in $t \ll n$ steps, meaning substantial computational savings in high-dimensional problems where computing each ∇f_i is expensive. Obviously, this randomization can introduce a substantial amount of variance into the update trajectory when compared to the full gradient case; this can be viewed as both a demerit and a merit, depending on the perspective and learning task. On the negative side, since we can understand this procedure as being a cheap approximation to the ideal full gradient update, excessive deviations from the ideal path will necessitate numerous iterations to correct for the idiosyncratic updates resulting from single-point estimates. Even though per-step costs are dramatically reduced, the possibility exists that the number of iterations required may be so large that the procedure ends up costing more to reach a solution comparable to the full gradient update after a few iterations. On the positive side, it is precisely this variance that can be useful for jump-starting the optimizer when it gets stuck in a flat region (where gradient updates become negligible) or a sub-optimal local minima in the case that f is not convex [82, p. 156]. This intuitive algorithm has been generalized to utilize "mini-batches," where $\mathcal{I} \subset \{1, \ldots, n\}$ is randomly sub-sampled, and the update takes the form

$$h_{(t+1)} = h_{(t)} - \alpha_{(t)} \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \nabla f_i(h_{(t)})$$

where typically $|\mathcal{I}| \ll n$. This intuitive update, coupled with both heuristics and theory for setting step size $\alpha_{(t)}$, is the modern *stochastic gradient descent* (SGD) algorithm, a cornerstone of modern neural network applications [13].

Another important notion found in the original paper of Rumelhart et al. [84] is that of introducing a *momentum* term to the gradient-based update term. Writing the update as $h_{(t+1)} = h_{(t)} + \Delta_{(t)}$, their proposal is to update using

$$\Delta_{(t)} = \alpha_{(t)}(-1)\nabla f(h_{(t)}) + \eta \Delta_{(t-1)}$$

where the term $\Delta_{(t)} = h_{(t+1)} - h_{(t)}$ is the "velocity" of the parameter sequence, namely the change in position per $(t+1)-t = 1$ unit of time (i.e., per iteration), and $\eta \in (0, 1)$ is the "mass" parameter controlling the impact of this modification. Fixing $\alpha_{(t)} = \alpha > 0$ for simplicity, we see that η is an exponential decay factor in the sense that

$$h_{(t+1)} = h_{(t)} - \alpha \sum_{k=0}^{t} \eta^k \nabla f(h_{(t-k)}).$$

If we set $\alpha = (1-\eta)/(1-\eta^{t+1})$, the weights $\{\eta^k/\alpha\}_{k=0}^{t}$ induce a convex combination of the gradients observed at previous steps, with weights that decrease at a geometric rate over time. Assuming an infinitely long past, $\alpha \to 1 - \eta$ and we have the straightforward update

$$\Delta_{(t)} = (1-\eta)(-1)\nabla f(h_{(t)}) + \eta \Delta_{(t)}. \tag{2.8}$$

This has the natural interpretation of being a simple exponential smoothing of the sequence $(\nabla f(h_{(t)}))$, a well-established technique in the time-series analysis domain [1]. Further variations on this procedure exist in the literature [82, p. 153–158], including the case where the smoothed update term is multiplied by a separate learning rate, i.e.,

$$h_{(t+1)} = h_{(t)} + \alpha_{(t)} \Delta_{(t)}$$

where $\Delta_{(t)}$ is as in (2.8). The practical purpose of this modification is to *accelerate* the convergence of this routine to a desirable solution, with the aim that since $\Delta_{(t)}$ should mitigate the impact of an exceedingly steep slope, the learning rate $\alpha_{(t)}$ can be left relatively large. For reference, modern machine learning researchers have paid much attention to the accelerated gradient method of Nesterov [73], drawing connections to the "classical" momentum method given above [90].

Moving forward chronologically, we set our sights on more modern proposals which have their roots in the first-order GD variants discussed above. In contrast to GD with momentum, where the update direction is a non-uniform average using geometrically decreasing weights, Blatt et al. [10] propose the (incremental) aggregated gradient (IAG) update taking the form

$$h_{(t+1)} = h_{(t)} - \frac{\alpha}{n} \sum_{k=0}^{t} \nabla f_{I(k)}(h_{(k)})$$

for all $t \geq n-1$ (i.e., first n candidates are initialized), where the index $I(k)$ is k modulo n, cycling over $\{1, 2, \ldots, n\}$. Here we take an average of past gradients, with weights that do not decay with time. A well-known randomized version of this algorithm, called stochastic average gradient (SAG), was proposed by Le Roux et al. [60]. In the SAG routine, IAG is generalized and randomized, with updates of the form

$$h_{(t+1)} = h_{(t)} - \frac{\alpha_{(t)}}{T} \sum_{k=0}^{t} y_{(t)}^{k}, \quad y_{(t)}^{i} := \begin{cases} \nabla f_i(h_{(t)}) & I(t) = i \\ y_{(t-1)}^{i} & I(t) \neq i \end{cases}$$

where $I(t) \in \{1, 2, \ldots, T\}$ is randomly generated at each step. Thus, gradients for each of the sample functions $f_1, \ldots, f_n$ are computed one at a time, updated in random order determined by the value of $I(t)$. A closely related technique stochastic gradient averaging

$$h_{(t+1)} = h_{(t)} - \frac{\alpha_{(t)}}{t+1} \sum_{k=0}^{t} \nabla f_{I(k)}(h_{(k)}),$$

where at each step one new gradient is computed, and all gradients over time (not just n) are stored and averaged for this update term. This appears in the the dual averaging method of Nesterov [72]. The randomization in these methods saves on per-iteration costs, and compared to the case of using just one ∇f_i to update at each step, the update variance is reduced via the aggregation, at the cost of additional memory requirements.

Much attention has been paid to work by Johnson and Zhang [54], who devise an appealing new strategy for reducing the update variance, with comparatively small memory requirements. Their algorithm has a two-loop structure. On the outer loop, given a stored candidate $\widetilde{h}$, they compute the full gradient

$$\widetilde{g} := \nabla f(\widetilde{h}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\widetilde{h}).$$

This $\widetilde{g}$ is stored for use in the inner loop, which runs as

$$\begin{aligned} \widehat{g}_{(t)} &:= \nabla f_{I(t)}(h_{(t)}) - \left(f_{I(t)}(\widetilde{h}) - \widetilde{g} \right) \\ h_{(t+1)} &= h_{(t)} - \alpha_{(t)} \widehat{g}_{(t)}. \end{aligned}$$

After T iterations of the inner loop, the "snapshot" $\widetilde{h}$ is updated to $h_{(T)}$ or an average of the $h_{(1)}, \ldots, h_{(T)}$ candidates computed in the inner loop, and the whole process repeats until convergence. Here we see that the reference candidate $\widetilde{h}$ is used to mitigate the deviation of each ∇f_i from ∇f, with the tacit assumption that the degree and direction of this deviation is mostly invariant to the parameter being evaluated. As with all the other randomized objectives, the aim is to decrease the variance as needed such that the step size $\alpha_{(t)}$ can be left as large as possible to maximize the efficiency of the updates.

In the preceding paragraphs, ostensibly the learning rate $\alpha_{(t)}$ played an auxiliary role, but its practical importance should not be understated, and indeed even from the few examples given above, it is clear that much previous research has been dedicated to optimizing the tradeoff between stability and speed, controlled directly by the step size. Fixing $\alpha_{(t)} = \alpha$ over all iterations is one option, but for non-convex objectives, convergence becomes a problem. To ensure convergence, a fixed "schedule" with $\alpha_{(t)}$ proportional to $1/t$ is a straightforward strategy [13]. Since this can require a tremendous number of iterations due to the slow progress, more sophisticated methods consider adaptive learning rates to make learning more efficient, often with step size determined based on both recent performance and local topography of the objective [5, 34, 86].

We have thus far implicitly assumed that all the iterative updates are closed with respect to $\mathcal{H}$, namely that if $h_{(t)} \in \mathcal{H}$, then the updated $h_{(t+1)} \in \mathcal{H}$ will also be. Depending on the definition of $\mathcal{H}$, this certainly need not hold in general, as model constraints reflecting prior information (or a lack of it) play a critical role in the learning process. Arguably the simplest case is where we are constrained to a ball of radius $r > 0$ in norm $\|\cdot\|$, namely

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} f_i(h), \quad \text{s.t. } \|h\| \leq r.$$

Given the context of first-order steepest descent discussed above, perhaps the most direct method of enforcing these constraints is via projection, namely updating as

$$h_{(t+1)} = \Pi_H \left(h_{(t)} - \alpha_{(t)} \nabla f(h_{(t)}) \right)$$

where the projection operator Π_H is defined by $\Pi_H(h) \coloneqq \arg\min_{g \in H} \|g - h\|$ for any subset $H \subseteq \mathcal{H}$, and in the r-ball constraint case, we set $H = \{h \in \mathcal{H} : \|h\| \leq r\}$. The complexity of carrying out this computation depends on the norm and the underlying space. If $\mathcal{H} = \mathbb{R}^d$, using the ℓ_2 norm simply requires multiplying the update $h_{(t)} + \Delta_{(t)}$ by the scalar $r/\|h_{(t)} + \Delta_{(t)}\|_2$, and this task appears naturally when applying support vector machines [87]. For high-dimensional sparsity scenarios using the ℓ_1 norm, the procedure is slightly more involved, but efficient algorithms have been proposed [35]. More generally, however, the projection operation can be expensive, and an alternative first-order procedure for inner product spaces is the *conditional gradient* technique of Frank and Wolfe, recently re-introduced into the machine learning community in recent years [53]. For the more general constrained minimization task of

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} f_i(h), \quad \text{s.t. } h \in H$$

for some convex, compact $H \subset \mathcal{H}$, the procedure runs as

$$\widetilde{h}_{(t)} = \arg\min_{u \in H} \langle u, \nabla f(h_{(t)}) \rangle$$

$$h_{(t+1)} = (1 - \eta_{(t)}) h_{(t)} + \eta_{(t)} \widetilde{h}_{(t)}$$

with $\eta_{(t)} \in (0, 1)$ a parameter that typically decreases as $\eta_{(t)} \propto 1/t$. The interpretation is straightforward. First a linear approximation to f at $h_{(t)}$ is made, and this proxy is then minimized within the feasible region. This new minimum $\widetilde{h}_{(t)}$ is then used to specify a desirable descent direction, and the updated parameter is taken on the line between $\widetilde{h}_{(t)}$ and $h_{(t)}$. When the constrained minimization of this linear function can be done more efficiently than projecting on to H, the approach becomes particularly appealing, and there are well-known learning tasks fall into this scenario [48].

Closely related to the constrained setting is the situation where our objective f is a convex composite function including an explicit regularization term, namely

$$f(h) = \frac{1}{n} f_i(h) + \Omega(h), \quad h \in \mathcal{H}$$

where the f_i are convex, and smooth in the sense that $\|\nabla f_i(g) - \nabla f_i(h)\| \leq L_i \|g - h\|$ for all $g, h \in \mathcal{H}$ for some constant $L_i > 0$, and with Ω being convex on $\mathcal{H}$ but potentially non-differentiable. An important sub-class of such problems is ERM with a regularizer induced by a norm. In the convex composite setting, *proximal* techniques from the convex optimization literature have been thoroughly explored [75]. The core idea is quite straightforward. For arbitrary convex function $F : \mathcal{H} \to \mathbb{R}$, define the proximal operator on $\mathcal{H}$ by

$$\operatorname{prox}_F(h) := \operatorname*{arg\,min}_{g \in \mathcal{H}} \left(\frac{1}{2} \|g - h\|^2 + F(g) \right).$$

Given the context of first-order steepest descent as above, *proximal gradient descent* (PGD) takes the form of

$$h_{(t+1)} = \operatorname{prox}_{\Omega} \left(h_{(t)} - \frac{\alpha_{(t)}}{n} \sum_{i=1}^{n} \nabla f_i(h_{(t)}) \right)$$

which amounts to computing the usual finite-sum steepest descent update, first ignoring the impact on Ω, and then minimizing a strongly convex proxy function (in the definition of prox) to get a final updated parameter which is close to the "unregularized" version but is constrained in the sense that Ω is not allowed to be too large. Another well-known alternative is the composite gradient method of Nesterov [71], is closely related to both PGD and the projected gradient descent given above [3]. When computing $\operatorname{prox}_{\Omega}(h)$ is not prohibitively expensive, using PGD has obvious practical appeal, and has led to the proposal of numerous proximal variations of all the procedures discussed above; all manner of combinations of acceleration, proximal approximation, variance reduction, randomized sub-sampling, and dual representations has lead to a massive body of work Murata and Suzuki [69, Sec. 1]. Finally, a particularly active area during the first decade of the 21st century was development of efficient algorithms for the LASSO model and its multitude of extensions [19, Ch. 2], chiefly in the statistics community. Here $\Omega(h) = \lambda \|h\|_1$ on $\mathcal{H} = \mathbb{R}^d$, where typically $n \ll d$ and a sparsity assumption is in place. As the role played by λ is significant, "path-following" procedures which optimize f for a whole array of λ choices [36]. For very high-dimensional (sparse) settings, a popular class of path-following algorithms are those that operate in a coordinate-wise fashion. Writing $h = (h_1, \ldots, h_d) \in \mathbb{R}^d$, the basic operation is simply

$$h_j = \operatorname*{arg\,min}_{u \in \mathbb{R}} f(h_1, \ldots, h_{j-1}, u, h_{j+1}, \ldots, h_d)$$

where in the simplest case we simply cycle over $j \in \{1, 2, \ldots, d\}$, although more sophisticated strategies which skip variables deemed inactive early on have been proposed [63]. In the context of first-order approximations using sub-gradients, *coordinate descent* routines effectively implementing a by-coordinate steepest descent procedure have been thoroughly studied and rich software libraries have been developed [42, 43].

To close this section, let us recapitulate the key points made here. We began by noting that when we only consider feedback to the learner and do not specify a response mechanism, as in the ERM learning principle, we can rarely say anything definitive about learning performance, and that analysis that accounts for the response mechanism, typified by the optimization procedures introduced in the preceding paragraphs, is the only way to provide guarantees that apply to the learning machines we actually implement. Tackling this problem theoretically is challenging, though important breakthroughs have been made by Hardt et al. [47], Lin and Rosasco [64], Chen et al. [27] recently, and the content of our chapter 4 represents a novel contribution to this line of work.

2.6 Data distributions used in simulation

In this section, we introduce the families of distributions that we leveraged for use in our numerical experiments, and explicitly highlight their properties for later classification, given the context of the previous section. Denote random variable of interest in each case by X. For ease of readability, we do not make reference to all sources, since most of the facts given below are part of the existing literature, or follow directly from it. Some encylopedic sources we have made substantial use of are Stuart and Ord [89], Rinne [81], and a useful online resource due to K. Siegrist.[6]

Arcsine

Parameters are $a \in \mathbb{R}$ (shift), $b > 0$ (scale). Support is $[a, a+b]$. Distribution function is

$$\mathbf{P}\{X \leq x\} = \frac{2}{\pi} \sin^{-1}\left(\sqrt{\frac{x-a}{b}}\right). \tag{2.9}$$

Distribution is symmetric, bounded, and thus sub-Gaussian.

Beta

Parameters are a and b (both shape). Support is $[0, 1]$. Distribution is defined as follows. Recall the Beta function of Euler, defined by

$$B(a,b) := \int_0^1 u^{a-1}(1-u)^{b-1}\, du = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)},$$

where Γ denotes the usual gamma function

$$\Gamma(x) := \int_0^\infty u^{x-1} e^{-u}\, dx.$$

With this notation in hand, a density function is naturally constructed as

$$p(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}. \tag{2.10}$$

The distribution is then simply

$$\mathbf{P}\{X \leq x\} = \int_0^x p(u)\, du. \tag{2.11}$$

If $a = b$ then the distribution is symmetric about $1/2$, otherwise it is asymmetric. Beta is sub-Gaussian.

[6] The project is called "Random: Probability, Mathematical Statistics, Stochastic Processes", and at the time of publishing this thesis, is available online at http://www.math.uah.edu/stat/.

Beta Prime

Parameters are a and b (both shape). Support is $(0, \infty)$. Distribution is defined by

$$\mathbf{P}\{X \le x\} = \mathbf{P}\left\{\text{Beta}(a,b) \le \frac{x}{1+x}\right\}, \tag{2.12}$$

where Beta(a, b) denotes a Beta random variable with shape parameters (a, b). Beta Prime is asymmetric, and note that

$$\mathbf{E}\, X^m = \begin{cases} \infty & m \ge b \\ \prod_{k=1}^{m} \frac{a+k-1}{b-k} & m < b. \end{cases}$$

This implies that the distribution is not sub-Gamma. To see this, look at Boucheron et al. [11, Theorem 2.3], where their equations (2.7) and thus (2.6) fail to hold.

Chi-squared

Parameter is $d > 0$ (degress of freedom). Support is $[0, \infty)$. Density function is

$$p(x) = \frac{1}{2^{d/2}\Gamma(d/2)} x^{d/2-1} e^{-x/2} \tag{2.13}$$

and thus the distribution is specified by

$$\mathbf{P}\{X \le x\} = \int_0^x p(u)\, du. \tag{2.14}$$

The distribution is asymmetric, and has integer moments of

$$\mathbf{E}\, X^m = 2^m \left(\frac{d}{2} + m - 1\right) \cdots \left(\frac{d}{2} + 1\right) \left(\frac{d}{2}\right).$$

Defining $f(k) = (d/2 + k - 1)$, note that for fixed $d > 0$ we can always take a constant $c > 0$ such that $ck > f(k)$ for any integer k. It thus follows

$$\mathbf{E}\, X^m \le (2c)^m m!$$

for all integer $m > 0$. We thus have that the Chi-squared distribution is sub-Exponential.

Exponential

Parameter is $r > 0$ (rate). Support is $[0, \infty)$. Distribution function is

$$\mathbf{P}\{X \le x\} = 1 - e^{-rx}. \tag{2.15}$$

The distribution is asymmetric, and its moment distribution function is

$$\mathbf{E} \exp(tX) = \frac{1}{1 - t/r}, \quad t \in (0, r).$$

While not sub-Gaussian, this is sub-Exponential, noting Proposition 4 and Boucheron et al. [11, p. 50].

Exponential-Logarithmic

Parameters $a \in (0,1)$ (shape) and $b > 0$ (scale). Support is $[0,\infty)$. Distribution function is

$$\mathbf{P}\{X \leq x\} = 1 - \frac{\log(1-(1-a)\exp(-x/b))}{a}. \tag{2.16}$$

This distribution is asymmetric, and sub-Exponential. To see this, note that the moments for integer m are

$$\mathbf{E}\,X^m = -b^m m! \frac{\mathrm{Li}_{m+1}(1-a)}{\log(a)},$$

where Li denotes the "polylogarithm" (often attributed to A. Jonquière), defined

$$\mathrm{Li}_m(u) := \sum_{k=1}^{\infty} \frac{u^k}{k^m}, \quad u \in (-1,1).$$

For any m, when $a \to 0$, $\mathrm{Li}_m(1-a)$ converges to a constant (Riemann's zeta function), and the denominator $-\log(a)$ heads to infinity. On the other hand, when $a \to 1$, it can be readily shown that $\mathrm{Li}_m(1-a)/\log(a) \to 1$. From this, for any shape $a \in (0,1)$ we can find a constant $c > 0$ such that

$$\mathbf{E}\,X^m \leq c^m m!,$$

implying the sub-Exponential property.

Fisher's F

Parameters $d_U, d_L > 0$ (degrees of freedom, **U**pper and **L**ower). Support is $[0,\infty)$. This distribution can be defined using the rather complicated density

$$p(x) = \left(\frac{d_U}{d_L} \frac{1}{B(d_U/2, d_L/2)} \frac{\left(\frac{d_U}{d_L}x\right)^{d_U/2-1}}{\left(1+\frac{d_U}{d_L}x\right)^{d_U/2+d_L/2}} \right), \tag{2.17}$$

and distribution function

$$\mathbf{P}\{X \leq x\} = \int_0^x p(u)\,du. \tag{2.18}$$

The distribution is asymmetric, and has moments

$$\mathbf{E}\,X^m = \begin{cases} \infty & d_L \leq 2m \\ \left(\frac{d_L}{d_U}\right)^m \frac{\Gamma(d_L/2-m)\Gamma(d_U/2+m)}{\Gamma(d_L/2)\Gamma(d_U/2)} & d_L > 2m. \end{cases}$$

It follows that the distribution cannot be sub-Gamma.

Folded Normal

Parameters $a \in \mathbb{R}$ (Normal shift), $b > 0$ (Normal scale). Support is $[0,\infty)$. Distribution function is

$$\mathbf{P}\{X \leq x\} = \Theta\left(\frac{x-a}{b}\right) - \Theta\left(-\frac{x-a}{b}\right). \tag{2.19}$$

This distribution is asymmetric, and in the case of $a = 0$ (the "Half-Normal"), moments can be easily computed as

$$\mathbf{E}\,X^{2m} = \frac{b^{2m}(2m)!}{2^m m!}$$
$$\mathbf{E}\,X^{2m+1} = \sqrt{\frac{2}{\pi}} b^{2m+1} 2^m m!$$

which implies sub-Gaussianity as follows. Noting that for even-valued m we have

$$\frac{m!}{(m/2)!} = m(m-1)\cdots(m/2+1) < m^{m/2},$$

and it follows immediately that for an appropriate constant $c > 0$,

$$\mathbf{E}\,X^m < c^m m^{m/2}$$

holds for each even m. For the odd m case, we have

$$\mathbf{E}\,X^m < c^m \left(\frac{m-1}{2}\right)! < c^m m^{m/2}$$

again for some appropriate constant $c > 0$. This gives us sub-Gaussianity by Definition 1(3).

Fréchet

Parameters $a \in \mathbb{R}$ (shift), $b > 0$ (scale), and $k > 0$ (shape). Support is (a, ∞). Distribution is defined by

$$\mathbf{P}\{X \leq x\} = \exp\left(-\left(\frac{x-a}{b}\right)^{-k}\right). \tag{2.20}$$

It is asymmetric, and as for any fixed $k > 0$, high-order moments are infinite, this distribution is not sub-Gamma.

Gamma

Parameters $b > 0$ (scale) and $k > 0$ (shape). Support is $(0, \infty)$. Density is defined by

$$p(x) = \frac{1}{\Gamma(k) b^k} x^{k-1} \exp(-x/b) \tag{2.21}$$

with the distribution function

$$\mathbf{P}\{X \leq x\} = \int_0^x p(u)\,du. \tag{2.22}$$

The Gamma distribution is asymmetric, and has moments of the form

$$\mathbf{E}\,X^m = b^m \frac{\Gamma(m+k)}{\Gamma(k)} \leq c^m m!$$

for an appropriate constant $c > 0$, and is thus sub-Exponential. This is just as seen in the section on Chi-squared, which is a special case of the Gamma.

Gaussian mixture

A k-component Gaussian mixture takes the form

$$X = I\{i=1\}N(a_1, b_1) + \cdots + I\{i=k\}N(a_k, b_k)$$

where i is a Categorical random variable with probabilities $p_j := \mathbf{P}\{i=j\} \in (0,1)$, $j \in [k]$, and $N(a,b)$ denotes a Normal random variable with mean a and standard deviation $b > 0$. Thus the k-component mixture has $3k$ parameters in total. The distribution function takes the form

$$\mathbf{P}\{X \le x\} = p_1\Phi_1(x) + \cdots + p_k\Phi_k(x)$$

where $\Phi_j(x) := \mathbf{P}\{N(a_j, b_j) \le x\}$. The support is $\mathbb{R}$. One would intuitively expect that this distribution is sub-Gaussian. Indeed this is the case. We have the following series of inequalities,

$$\begin{aligned}
\mathbf{E}\,|X|^m &= \int |X|^m \, d\mu \\
&\le 2^{m-1} \int (I\{i=1\}|N(a_1,b_1)|^m + \cdots + I\{i=k\}|N(a_k,b_k)|^m) \, d\mu_i \, d\mu_N \\
&= 2^{m-1} \int (p_1|N(a_1,b_1)|^m + \cdots + I\{i=k\}|N(a_k,b_k)|^m) \, d\mu_N \\
&\le 2^{m-1} \left(p_1 c_1^m m^{m/2} + \cdots + p_k c_k^m m^{m/2}\right) \\
&\le m^{m/2} c^m
\end{aligned}$$

where $c > 0$ is an appropriate constant, and μ_i denotes the distribution of i conditioned on the Normals, and μ_N denotes the distribution of the Normals. Using Definition 1(3), the Gaussian mixture is sub-Gaussian.

Gompertz

Parameters $a > 0$ (shape) and $b > 0$ (scale), and support $[0, \infty)$. The distribution function takes the form

$$\mathbf{P}\{X \le x\} = 1 - \exp\left(-a\left(\exp(x/b) - 1\right)\right).$$

With the observation that $\exp(u) - 1 \ge u^2$ for all $u \ge 0$, the tails can be bounded as

$$\mathbf{P}\{X > x\} = \exp\left(-a\left(\exp(x/b) - 1\right)\right) \le \exp\left(-a\frac{x^2}{b^2}\right)$$

which implies sub-Gaussianity by Definition 1(2), super-exponential tail decay.

Gumbel

With parameters $a \in \mathbb{R}$ (shift) and $b > 0$ (scale), support of $(-\infty, \infty)$, the Gumbel distribution, also known as the maximal type-1 extreme value distribution, takes the form

$$\mathbf{P}\{X \le x\} = \exp\left(-\exp\left(-\frac{x-a}{b}\right)\right).$$

First, observe that the left tail is sub-Gaussian, in the following sense. Letting $a = 0$ for simplicity, note that for $x \geq 0$, we have

$$\mathbf{P}\{-X > x\} = \exp\left(-\exp\left(\frac{x}{b}\right)\right) \leq \exp\left(-\frac{x^2}{b^2}\right)$$

which is the super-exponential tail decay of Definition 1(2). On the other hand, from Boucheron and Thomas [12] and [11, p. 51], we have that the right tail is effectively sub-Gamma. To see this, for the special case of $a = 0$, $b = 1$, one can show that

$$\log \mathbf{E} \exp(t(X - \mathbf{E}\, X)) \leq \operatorname{var}(X) \frac{t^2}{2(1-t)}, \quad t \geq 0$$

which is used as a standard definition of the one-tailed sub-Gamma property (see Proposition 4). Since sub-Gaussianity on the left tail implies the sub-Gamma property, the distribution as a whole can be considered sub-Gamma.

Hyperbolic secant

With parameters $a \in \mathbb{R}$ (shift) and $b > 0$ (scale), on support $(-\infty, \infty)$, the Hyperbolic secant distribution is determined by

$$\mathbf{P}\{X \leq x\} = \frac{2}{\pi} \operatorname{atan}\left(\exp\left(\pi \frac{x-a}{2b}\right)\right).$$

The density function is symmetric, and all moments clearly exist as the moment generating function takes the form

$$\mathbf{E} \exp(tX) = \sec(t), \quad t \in (-\pi/2, \pi/2).$$

The function is not quite sub-Gaussian by direct inspection, but we can readily prove that it is sub-Gamma, in that

$$\log \mathbf{E} \exp(tX) \leq \frac{t^2}{(1-ct)}, \quad 0 < t < \frac{1}{c}$$

for an appropriate $c > 0$. To show this, first note that

$$\sec''(t) = \left(\frac{\tan(t)}{\cos(t)}\right)' = \sec^3(t)(1 + \sin^2(t))$$

and is thus $\sec''(0) = 1$. On the other hand,

$$(\exp(t^2))'' = 4t^2 \exp(t^2) + 2\exp(t^2) \geq 2, \quad t \geq 0.$$

Since at $t = 0$, we have both $(\exp(t^2))' = 0 = \sec'(t)$ and $\exp(t^2) = 1 = \sec(t)$, we thus conclude by continuity that for an appropriate $k > 0$,

$$\sec(t) \leq \exp(t^2), \quad 0 \leq t < k.$$

It thus follows immediately that

$$\log \mathbf{E} \exp(tX) = \log \sec(t) \leq t^2 \leq \frac{t^2}{(1-ct)}$$

for all $t \in (0, 1/c)$, where $c = 1/k$. As a note on the nomenclature, the name of the distribution comes from the fact that the characteristic function is the hyperbolic secant function.

Irwin-Hall

The k-component Irwin-Hall distribution is defined by the random variable

$$X = U_1[0,1] + \cdots + U_k[0,1]$$

where the $U_i[0,1]$, $i \in [k]$, denote independent Uniform random variables on the unit interval. The support is thus $[0,k]$, and k is the only parameter. As it is bounded, this distribution is sub-Gaussian. The density function takes the form

$$p(x) = \frac{1}{2(k-1)!} \sum_{j=1}^{k} (-1)^j \binom{k}{j} \operatorname{sign}(x-j)(x-j)^{k-1}$$

is symmetric for all $k \geq 1$, and for $k > 1$ the mean/mode are unique, and take the value $\mathbf{E}\, X = k/2$.

Laplace

With parameters $a \in \mathbb{R}$ (shift) and $b > 0$ (scale), the Laplace distribution has the probability density

$$p(x) = \frac{1}{2b} \exp\left(\frac{|x-a|}{b}\right)$$

with distribution function

$$\mathbf{P}\{X \leq x\} = \begin{cases} \frac{1}{2}\exp\left(\frac{|x-a|}{b}\right), & x \leq a \\ 1 - \frac{1}{2}\exp\left(-\frac{x-a}{b}\right), & x > a \end{cases}$$

over support $(-\infty, \infty)$. The Laplace distribution is symmetric, and a canonical example of a distribution with tails that are exponential, but not super-exponential as in the Gaussian case. Note first that all the moments are finite and bounded as

$$\mathbf{E}(X-a)^k = \begin{cases} 0, & k \text{ odd} \\ b^k k!, & k \text{ even} \end{cases}.$$

This means for $a = 0$, we have $\mathbf{E}\, X^k \leq k!/(1/b)^k$ which by Boucheron et al. [11, p. 50] implies that the distribution is sub-Exponential.

Log-Logistic

With parameters $a > 0$ (shape) and $b > 0$ (scale), the log-Logistic distribution is specified by

$$\mathbf{P}\{X \leq x\} = \frac{1}{1 + (b/x)^a}$$

with support $(0, \infty)$. This distribution is bounded below on the left side, and for all $m \geq a$, the moments are $\mathbf{E}\, X^m = \infty$. Thus the distribution cannot even be sub-Gamma.

Log-Normal

The log-Normal distribution is defined by $x = \exp(N(a,b))$, where $N(a,b)$ is a Normal random variable with mean $a \in \mathbb{R}$ and standard deviation $b > 0$. As the name implies, "the log is Normal," which means the the distribution function is

$$\mathbf{P}\{X \leq x\} = \Phi\left(\frac{\log(x) - a}{b}\right)$$

with support of $(0, \infty)$, where Φ is the distribution function of the standard Normal $N(0,1)$. The log-Normal distribution is typically perceived as a "heavy-tailed" distribution, which is natural considering the fact that it is the exponential of an unbounded sub-Gaussian distribution. That said, as

$$\mathbf{E}\,X^m = \exp\left(am + \frac{1}{2}(bm)^2\right)$$

for all $m \in \mathbb{N}$, we have that all moments are defined and indeed finite. The distribution is asymmetric, and is not sub-Exponential. To see this simply requires some algebra. For simplicity, consider the special case of $a = 0$, $b = \sqrt{2}$. We want to show

$$\mathbf{E}\,X^m = e^{m^2} > \frac{2^{m+1}}{a^m}m!$$

for any pre-fixed $a > 0$. To do this, note that for any constant $c > 0$, using the obvious inequality $m! \leq m^m$, it follows that

$$c(m!)^{1/m} \leq cm < e^m$$

for large enough $m > 0$. Taking powers of m, we have

$$c^m m! \leq e^{m^2} = \mathbf{E}\,X^m.$$

For any arbitrary value of $a > 0$ then, if we set $c = 4/a$, this implies $c^m \geq 2^{m+1}/a^m$ for all $m > 0$, and thus connecting the inequalities, implies

$$\frac{2^{m+1}}{a^m}m! \leq c^m m! \leq \mathbf{E}\,X^m,$$

which is the desired result. In this sense, the log-Normal is a rather interesting sort of hybrid distribution, since all moments are finite, but it is not sub-Exponential.

Logistic

Defined in terms of the well-known logistic function, with parameters $a \in \mathbb{R}$ (shift) and $b > 0$ (scale), the distribution function takes the form

$$\mathbf{P}\{X \leq x\} = \frac{1}{1 + \exp(-(x-a)/b)}$$

over support $(-\infty, \infty)$. The distribution is symmetric and unbounded, and when $a = 0$, its moments take the form

$$\mathbf{E}\,X^m = \begin{cases} 0, & m \text{ odd} \\ (2^m - 2)(\pi b)^m |B_m|, & m \text{ even} \end{cases}$$

where the B_m denote Bernoulli numbers for even-valued integers m. From Alzer [4], there exists $l > 0$ such that

$$|B_m| \leq \frac{2m!}{(2\pi)^m(1-2^{l-m})}$$

holds for all even m. This implies the sub-Exponential property, and suggests that the distribution is, as visual inspection suggests, similar to the Gaussian distribution but has slightly heavier tails.

Maxwell

The standardized Maxwell distribution can be considered the Euclidean norm of a vector in three-dimensional space, where the elements are independent standard Normal random variables. That is, writing $n_i = N_i(0,1)$, $i = 1,2,3$, representing three independent standard Normal random variables, the standardized Maxwell random variable takes the form

$$U := \sqrt{n_1^2 + n_2^2 + n_3^2}.$$

The generalized version introduces a scaling parameter $b > 0$, and is defined by $X := bU$. The distribution function takes the form

$$\mathbf{P}\{X \leq x\} = \mathbf{P}\{U \leq x/b\}$$

where the standardized distribution function is

$$\mathbf{P}\{U \leq u\} = 2\Phi(u) - \sqrt{\frac{2}{\pi}} u \exp(-u^2/2) - 1$$

over support $[0, \infty)$. Intuitively, this is a quantity we would expect to be sub-Gaussian, and this is indeed the case. To see this, note that for arbitrary integer $m > 0$, the moments take the form

$$\mathbf{E}\, X^m = b^m \sqrt{\frac{2^{m+2}}{\pi}} \Gamma\left(\frac{m+3}{2}\right).$$

Dealing with the Gamma function factor first, recalling for integer a we have $\Gamma(a+1) = a!$, we can easily bound this for all integer $m > 0$ by

$$\begin{aligned}\Gamma\left(\frac{m+3}{2}\right) &\leq (\lceil (m+3)/2 \rceil - 1)! \\ &\leq m \times \cdots \times m, \quad \text{with } (m+3)/2 \text{ multiplicands} \\ &= m^{3/2} m^{m/2} \\ &\leq c_1^m m^{m/2}\end{aligned}$$

for large enough constant $c_1 > 0$. Similarly, one can take a $c_2 > 0$ large enough that

$$b^m \sqrt{\frac{2^{m+2}}{\pi}} \leq c_2^m$$

for all m. Setting $c = c_1 c_2$, we have

$$\mathbf{E}\, X^m \leq c^m m^{m/2}$$

which implies sub-Gaussianity by Definition 1(3).

Pareto

The Pareto distribution is the canonical "polynomial-tailed" distribution, and with parameters $a > 0$ (shape) and $b > 0$ (scale), takes the form

$$\mathbf{P}\{X \leq x\} = 1 - \left(\frac{b}{x}\right)^{a}$$

with support of $[b, \infty)$. The distribution is bounded below, asymmetric, and just looking at the tails, one can observe that they decay exponentially slower than sub-Gaussian tails. The higher-order moments are infinite, with

$$\mathbf{E}\, X^{m} = \begin{cases} \infty, & m \geq a \\ b^{m}\frac{a}{a-m}, & m < a \end{cases}$$

implying that the Pareto distribution cannot be sub-Gaussian.

Rayleigh

Analogous to the Maxwell distribution, the Rayleigh distribution corresponds to the norm of a vector of independent standard Gaussian components, written $n_i = N_i(0,1)$, $i = 1, 2$, taking the form

$$U = \sqrt{n_1^2 + n_2^2}.$$

The generalized Rayleigh has a scale parameter $b > 0$, and is defined $X := bU$. The distribution function is

$$\mathbf{P}\{X \leq x\} = 1 - \exp\left(-\frac{x^2}{2b^2}\right)$$

over support $[0, \infty)$. The distribution is bounded below, and asymmetric. As the moments about zero take the form $\mathbf{E}\, X^{m} = b^{m} 2^{m/2} \Gamma(1 + m/2)$, it follows that

$$\mathbf{E}\, X^{2m} = (2b^2)^{m} m!$$

for all integer $m > 0$, thereby implying that Rayleigh is sub-Gaussian by Definition 1(3).

Semi-circle

Named for the shape of the graph of its density function, the semi-circle distribution is symmetric and has two parameters, $c \in \mathbb{R}$ and $r > 0$, which respectively correspond to the centre and radius of the circle concerned. The distribution function takes the form

$$\mathbf{P}\{X \leq x\} = \frac{1}{2} + \frac{x - c}{\pi r^2} \sqrt{r^2 - (x - c)^2}$$

with support of $[c - r, c + r]$. Since this distribution has bounded support, it is sub-Gaussian.

Student's t

Another canonical example of a distribution whose tails decay much slower than a Gaussian, the so-called "t distribution" of Student is specified by integer parameter $k > 0$ (degrees of freedom), and density function

$$p(x) = \frac{\Gamma((k+1)/2)}{\sqrt{\pi k}\Gamma(k/2)} \left(1 + \frac{x^2}{k}\right)^{-(k+1)/2}$$

over support $(-\infty, \infty)$. The distribution is symmetric and unbounded. For the mth moment about zero, if $m \geq k$, then $\mathbf{E}\, X^m$ is either infinite (when m is even), or it is undefined (when m is odd), and thus Student's t distribution cannot be sub-Gaussian.

Triangle

Analogous to the semi-circle distribution, the Triangle distribution is named for its triangle-shaped density function. It is bounded and thus sub-Gaussian, and depending on parameter settings can be either symmetric or asymmetric. The distribution function is elementary but somewhat convoluted, though we reproduce it here for reference. We have three parameters: vertex $v \in [0,1]$, shift $a \in \mathbb{R}$, and scale $b > 0$. The distribution function is defined as follows. For the limit cases, we have

$$\mathbf{P}\{X \leq x\} = \begin{cases} 1 - \frac{1}{b^2}(a+b-x)^2 & v = 0 \\ \frac{1}{b^2}(x-a)^2 & v = 1 \end{cases}$$

and for the case of $0 < v < 1$, we have

$$\mathbf{P}\{X \leq x\} = \begin{cases} 1 - \frac{1}{b^2}(a+b-x)^2 & x \in [a, a+vb] \\ \frac{1}{b^2}(x-a)^2 & x \in [a+vb, a+b] \end{cases}$$

and clearly the support is $[a, a+vb]$. Shifting the vertex v along the unit interval moves the location of the "vertex" (i.e., the mode) of the density function between a and $a+b$, with the symmetric case being where $v = 1/2$.

U-Power

This bounded distribution has a bowl-shaped density function, and is specified by parameters $k \in \mathbb{N}$ (shape), $a \in \mathbb{R}$ (shift), and $b > 0$ (scale), taking the form

$$\mathbf{P}\{X \leq x\} = \frac{1}{2}\left(1 + \left(\frac{x-a}{b}\right)^{2k+1}\right)$$

over support $[a-b, a+b]$. The distribution is symmetric and thus sub-Gaussian.

Weibull

With parameters $a > 0$ (shape) and $b > 0$ (scale), the Weibull distribution is defined by

$$\mathbf{P}\{X \leq x\} = 1 - \exp\left(-\left(\frac{x}{b}\right)^a\right)$$

with support $[0, \infty)$. Weibull moments take the form

$$\mathbf{E}\, X^m = b^m \Gamma(1 + m/a)$$

and thus are finite for all integer $m > 0$. The shape parameter a dramatically impacts the tail behaviour of this distribution. Note that for $a \geq 2$, we have

$$\mathbf{E}\, X^{2m} = b^{2m}(2m/a)! \leq (b^2)^m m!$$

and thus is clearly sub-Gaussian by Definition 1(3). For the case of $a \geq 1$, from Definition 3 we clearly have that the distribution is sub-Exponential. The tails grow heavier and heavier as $a < 1$ approaches zero. It is easy to check that $a < 2$ implies that the Weibull is *not* sub-Gaussian. To see this, assuming $0 < a < 2$, note that for any positive constants c_1 and c_2, there is always a value x_0 such that

$$x > x_0 \implies c_1 \exp(-(x/c_2)^2) > \exp(-(x/b)^a) = \mathbf{P}\{X > x\}$$

and thus the tails are not sub-Gaussian by Definition 1(2).

Bibliography

[1] Abraham, B. and Ledolter, J. (1983). *Statistical Methods for Forecasting.* John Wiley & Sons.

[2] Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning From Data.*

[3] Agarwal, A., Negahban, S., and Wainwright, M. J. (2012). Fast global convergence rates of gradient methods for high-dimensional statistical recovery. *Annals of Statistics*, 40(5):2452–2482.

[4] Alzer, H. (2000). Sharp bounds for the Bernoulli numbers. *Archiv der Mathematik*, 74(3):207–211.

[5] Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

[6] Ash, R. B. and Doléans-Dade, C. A. (2000). *Probability and Measure Theory.* Academic Press, 2nd edition.

[7] Audibert, J.-Y. and Catoni, O. (2011). Robust linear least squares regression. *Annals of Statistics*, 39(5):2766–2794.

[8] Belloni, A. and Chernozhukov, V. (2011). ℓ_1-penalized quantile regression in high-dimensional sparse models. *Annals of Statistics*, 39(1):82–130.

[9] Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, pages 1705–1732.

[10] Blatt, D., Hero, A. O., and Gauchman, H. (2007). A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51.

[11] Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: a nonasymptotic theory of independence.* Oxford University Press.

[12] Boucheron, S. and Thomas, M. (2012). Concentration inequalities for order statistics. *Electronic Communications in Probability*, 17(51).

[13] Bousquet, O. and Bottou, L. (2008). The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 21*, pages 161–168.

[14] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

[15] Bradic, J., Fan, J., and Wang, W. (2011). Penalized composite quasi-likelihood for ultrahigh dimensional variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):325–349.

[16] Breiman, L. (1968). *Probability.* Addison-Wesley.

[17] Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384.

[18] Brownlees, C., Joly, E., and Lugosi, G. (2015). Empirical risk minimization for heavy-tailed losses. *Annals of Statistics*, 43(6):2507–2536.

[19] Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer.

[20] Candès, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, pages 2313–2351.

[21] Candès, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215.

[22] Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, 2nd edition.

[23] Catoni, O. (2009). High confidence estimates of the mean of heavy-tailed real random variables. *arXiv preprint arXiv:0909.5366*.

[24] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148–1185.

[25] Chafaï, D., Guédon, O., Lecué, G., and Pajor, A. (2012). Interactions between compressed sensing, random matrices, and high dimensional geometry. *Panoramas et synthèses*, 37.

[26] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20(1):33–61.

[27] Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *arXiv preprint arXiv:1705.05491*.

[28] Crockett, J. B. and Chernoff, H. (1955). Gradient methods of maximization. *Pacific Journal of Mathematics*, 5(1):33–50.

[29] Daubechies, I. (1992). *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM.

[30] Devroye, L., Lerasle, M., Lugosi, G., and Oliveira, R. I. (2015). Sub-Gaussian mean estimators. *arXiv preprint arXiv:1509.05845*.

[31] Donoho, D. L. (2004). For most large undetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. *Manuscript via author's homepage*.

[32] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

[33] Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18.

[34] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

[35] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*.

[36] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499.

[37] Fan, J., Fan, Y., and Barut, E. (2014a). Adaptive robust variable selection. *Annals of Statistics*, 42(1):324.

[38] Fan, J., Li, Q., and Wang, Y. (2014b). Robust estimation of high-dimensional mean regression. *arXiv preprint arXiv:1410.2150v1*.

[39] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

[40] Feldman, V. (2016). Generalization of ERM in stochastic convex optimization: The dimension strikes back. In *Advances in Neural Information Processing Systems 29*, pages 3576–3584.

[41] Freedman, D. A. (2009). *Statistical Models: Theory and Practice*. Cambridge University Press, revised edition.

[42] Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332.

[43] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

[44] Goldie, C. M. and Klüppelberg, C. (1998). Subexponential distributions. In Adler, R. L., Feldman, R., and Taqqu, M. S., editors, *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Birkhäuser.

[45] Halmos, P. R. (1950). *Measure Theory*, volume 18 of *Graduate Texts in Mathematics*. Springer.

[46] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

[47] Hardt, M., Recht, B., and Singer, Y. (2015). Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*.

[48] Hazan, E. and Kale, S. (2012). Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 521–528.

[49] Hsu, D. and Sabato, S. (2014). Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML2014)*, pages 37–45.

[50] Hsu, D. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, 17(18):1–40.

[51] Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

[52] Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, 1st edition.

[53] Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435.

[54] Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323.

[55] Knight, K. and Fu, W. (2000). Asymptotics for lasso-type estimators. *Annals of Statistics*, pages 1356–1378.

[56] Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1):33–50.

[57] Koltchinskii, V. (2011). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: 2008 Saint-Flour Lectures*, volume 38. Springer.

[58] Kosorok, M. R. (2008). *Introduction to Empirical Processes and Semiparametric Inference.* Springer.

[59] Lambert-Lacroix, S. and Zwald, L. (2011). Robust regression through the Huber's criterion and adaptive lasso penalty. *Electronic Journal of Statistics*, 5:1015–1053.

[60] Le Roux, N., Schmidt, M., and Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2663–2671.

[61] Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation.* Springer, 2nd edition.

[62] Lerasle, M. and Oliveira, R. I. (2011). Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914.*

[63] Li, Y. and Osher, S. (2009). Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503.

[64] Lin, J. and Rosasco, L. (2016). Optimal learning for multi-pass stochastic gradient methods. In *Advances in Neural Information Processing Systems 29*, pages 4556–4564.

[65] Luenberger, D. G. (1969). *Optimization by Vector Space Methods.* John Wiley & Sons.

[66] Lugosi, G. and Mendelson, S. (2016). Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757.*

[67] Meinshausen, N. and Yu, B. (2009). Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, pages 246–270.

[68] Minsker, S. (2015). Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335.

[69] Murata, T. and Suzuki, T. (2017). Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization. *arXiv preprint arXiv:1703.00439.*

[70] Nasrabadi, N. M., Tran, T. D., and Nguyen, N. (2011). Robust lasso with missing and grossly corrupted observations. In *Advances in Neural Information Processing Systems 24*, pages 1881–1889.

[71] Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. Technical report, Université Catholique de Louvain.

[72] Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming, Series B*, 120(1):221–259.

[73] Nesterov, Y. E. (1983). A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376.

[74] Nguyen, N. H. and Tran, T. D. (2013). Robust lasso with missing and grossly corrupted observations. *IEEE Transactions on Information Theory*, 59(4):2036–2058.

[75] Nitanda, A. (2014). Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems 27*, pages 1574–1582.

[76] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337.

[77] Papadimitriou, C. H. (1994). *Computational Complexity.* Addison-Wesley.

[78] Pollard, D. (1984). *Convergence of Stochastic Processes.* Springer-Verlag.

[79] Pollard, D. (2002). *A User's Guide to Measure Theoretic Probability.* Cambridge University Press.

[80] Rey, W. J. J. (1983). *Introduction to Robust and Quasi-Robust Statistical Methods.* Springer.

[81] Rinne, H. (2009). *The Weibull Distribution: A Handbook.* CRC Press.

[82] Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

[83] Rudin, W. (1976). *Principles of Mathematical Analysis.* McGraw-Hill, 3rd edition.

[84] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[85] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1987). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8. MIT Press.

[86] Schoenauer-Sebag, A., Schoenauer, M., and Sebag, M. (2017). Stochastic gradient descent: Going as fast as possible but not faster. *arXiv preprint arXiv:1709.01427.*

[87] Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated subgradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning.*

[88] Smith, R. L. (1987). Estimating tails of probability distributions. *Annals of Statistics*, 15(3):1174–1207.

[89] Stuart, A. and Ord, J. K. (1994). *Kendall's Advanced Theory of Statistics Volume 1: Distribution Theory.* Hodder Arnold, 6th edition.

[90] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147.

[91] Takeuchi, I., Le, Q. V., Sears, T. D., and Smola, A. J. (2006). Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264.

[92] Talagrand, M. (1996). A new look at independence. *Annals of Probability*, 24(1):1–34.

[93] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1):267–288.

[94] Tolstov, G. P. (1962). *Fourier Series.* Prentice-Hall.

[95] Tukey, J. W. (1960). A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics: Essays in honor of Harold Hotelling*, pages 448–485. Stanford University Press.

[96] van der Vaart, A. W. (1998). *Asymptotic Statistics.* Cambridge University Press.

[97] van der Vaart, A. W. and Wellner, J. A. (1996). *Weak Convergence and Empirical Processes.* Springer.

[98] Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley.

[99] Vardi, Y. and Zhang, C.-H. (2000). The multivariate L_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426.

[100] Vershynin, R. (2012). Introduction to the non-asymptotic analysis of random matrices. In Eldar, Y. C. and Kutyniok, G., editors, *Compressed Sensing: Theory and Applications*, chapter 5. Cambridge University Press.

[101] Wald, A. (1949). Statistical decision functions. *Annals of Mathematical Statistics*, pages 165–205.

[102] Wang, H., Li, G., and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the LAD-Lasso. *Journal of Business & Economic Statistics*, 25(3):347–355.

[103] Wright, J. and Ma, Y. (2010). Dense error correction via ℓ^1-minimization. *IEEE Transactions on Information Theory*, 56(7):3540–3560.

[104] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227.

[105] Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, pages 224–244.

[106] Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563.

Chapter 3

Learning using robust objectives

3.1 Introduction

Accurate prediction of response $y \in \mathbb{R}$ from novel pattern $\boldsymbol{x} \in \mathbb{R}^d$, based on an observed sample sequence of pattern-response pairs $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$, $\boldsymbol{z} := (\boldsymbol{x}, y)$, is one of the most fundamental of statistical estimation tasks. Under particular assumptions such as bounded losses or sub-Gaussian residuals, a rich theory has developed in recent decades [32, 6, 2, 7, 44, 8], with variants of empirical risk minimization (ERM) routines playing a central role. The principle underlying such procedures is the use of the sample mean to approximate the risk (expected loss), which in turn functions as a location parameter of the unknown loss distribution. When the loss is concentrated around this value, this approximation is accurate, and ERM procedures perform well with appealing optimality properties [43].

Unfortunately, these assumptions are stringent, and in general, without *a priori* evidence of the contrary, our data cannot reasonably be expected to satisfy them. The fundamental problem manifests itself clearly in the simple setting of heavy-tailed real observations, in which the sub-optimality of the empirical mean is well-known [14]. A simple solution when using ERM is to leverage slower-growing loss functions (e.g., ℓ_1 instead of ℓ_2), but making this decision is inherently *ad hoc* and requires substantial prior information. Another option is model regularization [47, 8, 26], potentially combined with quantile regression [33, 46], though both methods introduce new parameters and we are faced with a difficult model selection problem [15], whose optimal solution is in practice often very sensitive to the empirical distribution. Put simply, in a non-parametric setting, one incurs a major risk of bias in the form of minimizing an impractical location parameter (e.g., the median under asymmetric losses), in order to ensure estimates are stable.

Considering these issues, it would be desirable to design an objective function which achieves the desired stability, but pays a smaller price in terms of bias, and therefore has minimal *a priori* requirements (Fig. 3.1). It is the objective of this chapter to derive a regression algorithm which utilizes such a mechanism at tolerable computational cost. In section 3.2 we review the technical literature, giving our contributions against this backdrop. Section 3.3 introduces the core routine and important ideas underlying its construction in an intuitive manner, with formal justification and convergence analysis following in 3.4. Numerical performance tests are given in section 3.5, with key take-aways summarized in section 4.5.

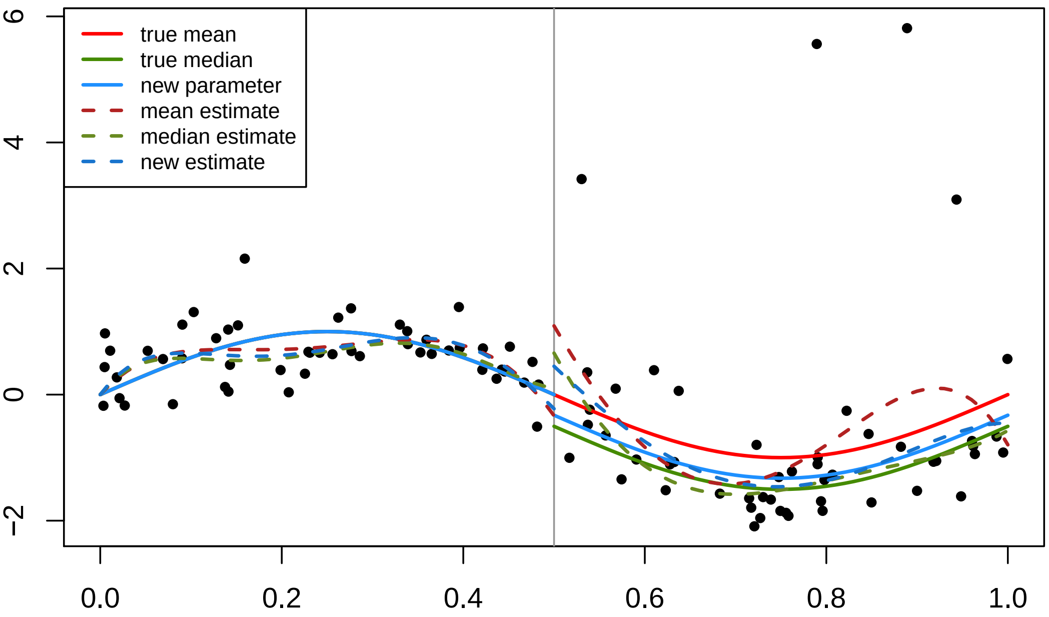

Figure 3.1: A one-dimension regression example (see Appendix 3.6). When additive noise is heavy-tailed (the right half), estimating $\mathbf{E}(y; \boldsymbol{x})$ via least squares is difficult under small samples. On the other hand, estimating $\mathrm{med}(y; \boldsymbol{x})$ often introduces an unacceptable bias. In this chapter we investigate "robust objectives" which act as all-purpose parameters to be estimated under diverse settings.

3.2 Background and contributions

In this section we review the technical literature which is closely related to our work, and then within this context establish the main contributions made in this chapter.

Related work Many tasks involve minimizing a function, say $L(\cdot)$, as a function of candidate $h \in \mathcal{H}$, which depends on the underlying distribution and is thus unknown. One line of work explicitly looks at refining the approximate objective function used. A key theme is to down-weight errant observations automatically, and to construct a new estimate $\widehat{L}(h) \approx L(h)$ of the risk, re-coding the algorithm as $\widehat{h} := \arg\min_{h \in \mathcal{H}} \widehat{L}(h)$. The now-classic work of Rousseeuw and Yohai [40] on S-estimators highlights important concepts in our work. They use the M-estimator of scale of the residual $h(\boldsymbol{x}) - y$, written $\widehat{s}(h)$, directly as objective function, setting $\widehat{L}(h) = \widehat{s}(h)$. The idea is appealing and has (classical) robustness properties, though serious issues of stability and computational cost have been raised [31], and indeed even the fast modern routines are designed only for the rather special parametric setting where errant data can be discarded [42], which severely limits utility in our setting.

Re-weighting of extreme observations using M-estimators of the mean has been recently revisited by Catoni [13], later revised and published as Catoni [14]. A multi-dimensional extension of this theory appears in Audibert and Catoni [4], where they propose a function of the form

$$d(h, h') := \lambda(\|h\|^2 - \|h'\|^2) + \mathbf{E}\,\psi_C\left(l(h; \boldsymbol{z}) - l(h'; \boldsymbol{z})\right),$$

where $\lambda > 0$ is a user-set parameter, $l(h; \boldsymbol{z})$ is a penalty assigned to h on the event of observing $\boldsymbol{z}$, and ψ_C is a sigmoidal truncation function

$$\psi_C(u) := \begin{cases} -\log(1 - u + u^2/2), & 0 \leq u \leq 1 \\ \log(2), & u \geq 1 \\ -\psi(-u), & u \leq 0. \end{cases}$$

The refined loss is then $\widehat{L}(h) = \sup\{d(h, h') : h' \in \mathcal{H}\}$, and is effectively a robust proxy of the "ridge risk" $\mathbf{E}\, l(h; \boldsymbol{z}) + \lambda \|h\|^2$. Many novel results are given, but it is not established whether an algorithm realizing the desired performance actually exists or not. More precisely, they show that one requires $\widehat{L}(\widehat{h}) = \inf_{h \in \mathcal{H}} \widehat{L}(h) + O(d/n)$ where d is model dimension. Unfortunately, construction of such a $\widehat{h}$ is left as future work, though a sophisticated iterative attempt is proposed by the authors. Another natural extension is given by Brownlees et al. [12], who directly apply these foundational results by using the Catoni class of M-estimators of risk, generalizing ψ_C above, to build $\widehat{L}$, which amounts to minimizing the root of the sample mean of $\{\psi_C(l(h; \boldsymbol{z}_i) - \theta)\}_{i=1}^n$ in θ. Novel bounds on excess risk are given, but this depends on an "optimal" scaling procedure which requires knowledge of the true variance. In addition, as this "robust loss" is defined implicitly, actually minimizing it is a non-trivial and expensive computational task.

Another interesting line of recent work revisits the merits of aggregation, a well-known notion from, for example, the bagging and boosting literature [10, 22]. The idea is to construct k candidates $\widehat{h}_{(1)}, \ldots, \widehat{h}_{(k)}$, typically by partitioning the data $D = \cup_{j=1}^k D_j$, and to aggregate them such that estimates derived from errant or uncharacteristic sub-samples are downweighted. One lucid example is the work of Minsker [36], who uses

$$\widehat{h} = \operatorname*{arg\,min}_{h \in \mathcal{H}} \sum_{i=1}^k \|h - \widehat{h}_{(i)}\|, \quad \widehat{h}_{(i)} := \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{|D_i|} \sum_{j \in D_i} l(h; \boldsymbol{z}_j)$$

namely the geometric median of the candidates (in norm $\|\cdot\|$), where each $\widehat{h}_{(i)}$ is the ERM estimate on the ith partition. The key notion here is that as long as most of the candidates are not overly poor, the aggregrate will be strong. This same notion was explored by Lerasle and Oliveira [34], where the "not overly poor" notion was made concrete with margin type conditions (section 5.1, page 14). As well, the work of Hsu and Sabato [27, 28] generalizes the formulation of these two works, casting the aggregation task as a "robust distance approximation," which is highly intuitive, is suggestive of algorithm design techniques, and yields tools applicable to many other problems [17, 35]. One major issue is that when sample sizes are small, very few partitions can be made. The key concern then is that when samples are large enough that k can be taken large, a less sophisticated method might already perform equally well on the full sample.

Our contributions In this work, the key idea is to use an approximate minimization technique to efficiently make use of powerful but computationally unwieldy robust losses. We propose a novel routine which is rooted in theoretical principles, but makes enough concessions to be useful in practice. Our main contributions can be summarized as follows:

- A fast minimizer of robust losses for general regression tasks, which is easily implemented, inexpensive, and requires no knowledge of higher-order moments of the data.
- Analysis of conditions for existence and convergence of the core routine.

- Comprehensive empirical performance testing, illustrating dominant robustness in both simulated settings and on real-world benchmark data sets.

Taken together, the theoretical and empirical insights suggest that we have a routine which behaves as we would expect statistically, converges quickly in practice, and which achieves a superior balance between cost and performance in the non-parametric setting standard to machine learning problems.

3.3 Fast minimization of robust objectives

In this section, we introduce the learning task of interest and give an intuitive derivation of our proposed algorithm. More formal analysis of the convergence properties of this procedure, from both statistical and computational viewpoints, is carried out in section 3.4.

3.3.1 A general learning task

Given "candidate" $h \in \mathcal{H}$, member of a class of vectors or functions, and particular input/output instance $\boldsymbol{z} = (\boldsymbol{x}, y)$, we assign a penalty, $l(h; \boldsymbol{z}) \geq 0$ via loss function l—smaller is better—and evaluate the quality of h. Assuredly, doing this for a single observation $\boldsymbol{z}$ is insufficient; as this is a *learning* task, given incomplete prior information, we must choose h such that when we draw $\boldsymbol{z}$ randomly from an unknown probability distribution μ, representing unknown physical or social processes in our system of interest, the (random) quantity $l(h; \boldsymbol{z})$ is small. If the expected value $L_\mu(h) := \mathbf{E}_\mu \, l(h; \boldsymbol{z})$, also called the *risk*, is small, then we expect the penalty $l(h; \boldsymbol{z})$ to be small on average. As such, a natural strategy is to choose a "best" candidate by the following program:

$$\min L_\mu(h), \quad \text{s.t. } h \in \mathcal{H}.$$

At this point, we run into a problem: μ is unknown, and thus L_μ is unknown. All we have access to is n independent draws of $\boldsymbol{z}$, namely the sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, and from this we must *approximate* the true objective, and then *minimize* this approximation as a proxy of L_μ.

Example 5 (Typical formulations). The pattern recognition problem has generic input space $\mathcal{X}$ and discrete labels, namely $\boldsymbol{x} \in \mathcal{X}$ and $y \in \{1, \ldots, C\}$. Here the "zero-one" loss $l(h; \boldsymbol{z}) = I\{h(\boldsymbol{x}) \neq y\}$ makes for a natural penalty to classifier h. More generally, the regression problem task has response $y \in \mathbb{R}$, and the classic metric for evaluating the quality of predictor $h : \mathcal{X} \to \mathbb{R}$ is the quadratic loss $l(h; \boldsymbol{z}) = (y - h(\boldsymbol{x}))^2$.

3.3.2 Issues to overcome

Intuitively, if our approximation, say $\widehat{L}$, of L_μ, is not very accurate, then any minima of $\widehat{L}$ will likely be useless. Thus the first item to deal with is making sure the approximation $\widehat{L} \approx L_\mu$ is sharp. Perhaps the most typical approach is to set $\widehat{L}(h)$ to the sample mean, $\sum_{i=1}^{n} l(h; \boldsymbol{z}_i)/n$. In this case, the estimate is "unbiased" as $\mathbf{E}\,\widehat{L}(h) = L_\mu(h)$, but unfortunately the variance can be highly undesirable [13, 14]. There is no need to constrain ourselves to unbiased estimators, as Figure 3.2(a) illustrates; paying a small cost in term of bias (allowing $\mathbf{E}\,\widehat{L}(h) \neq L_\mu(h)$) for much stabler output (large reduction in variance of $\widehat{L}$) is an appealing route.

One strategy to do this is as follows. Consider a "re-weighted" average approximation, namely $\widehat{L}(h; \boldsymbol{\alpha})$ given as

$$\widehat{L}(h; \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i l(h; \boldsymbol{z}_i)$$

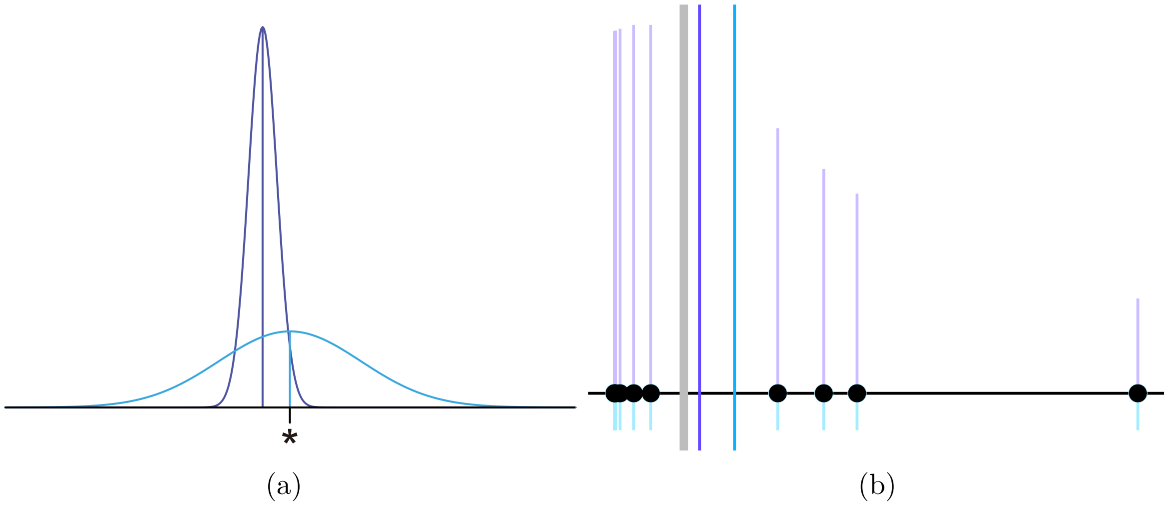

Figure 3.2: (a) Schematic of two estimators of L_{μ} (their density in n-sample space), one unbiased but with high variance (turquoise), another biased but concentrated (purple). (b) Points along the black line are observations $x_1, \ldots, x_n \in \mathbb{R}$ sampled from a heavy-tailed distribution ($n = 7$). The three vertical rules are: true mean (thick grey), sample mean (turquoise), and the M-estimate of location (purple). Vertical ranges associated with each point denote weight sizes, computed by $1/n$ (pale turquoise) and $\rho'(x_i - \gamma)/(x_i - \gamma)$ (pale purple). Down-weighting errant observations has a clear positive impact on estimates.

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ with $0 \leq \alpha_i \leq 1$ are our weights. In the sample mean case, $\alpha_i = 1/n$ for all observation points. However, since n is finite, one often runs into "errant" points which, when given the same amount of weight as all other points, do not accurately reflect the true underlying distribution. Thus, down-weighting these errant points by assigning them small weights (α_i near 0), and subsequently treating all the "typical" points as equals, should in principle allow us to overcome this issue. A mechanism which effectively does this for us is to use the M-estimate of location [29]; that is, to set

$$\widehat{L}(h; \rho, s) = \arg\min_{\theta} \sum_{i=1}^{n} \rho\left(\frac{l(h; \boldsymbol{z}_i) - \theta}{s}\right) \tag{3.1}$$

for each $h \in \mathcal{H}$. Here ρ is a convex function which is effectively quadratic around the origin, but grows much more slowly (Figure 3.3), and $s > 0$ is a scaling parameter. The re-weighting is implicit here, enacted via a "soft" truncation of errant points. Data points which are fairly close to the bulk of the sample are taken as-is (in the region where ρ is quadratic), while the impact of outlying points is attenuated (in the region where ρ is linear). We remark that such an estimator is assuredly biased in the sense that $\mathbf{E}\,\widehat{L}(h; \rho, s) \neq L_{\mu}(h)$ in most cases, but the desired impact is readily confirmed via simple tests, as in Figure 3.2(b).

Following such a strategy, the algorithm to run is

$$\widehat{h} = \arg\min_{h \in \mathcal{H}} \widehat{L}(h; \rho, s).$$

Given knowledge of the true variance, the utility of this approach from a statistical perspective has been elegantly analyzed by Brownlees et al. [12]. That we do not know the true variance is one issue; another critical issue is that this new "robust loss" $\widehat{L}(h; \rho, s)$ is defined *implicitly*, and is thus computationally quite uncongenial. Derivatives are not available in closed form, and every call to $\widehat{L}(h; \rho, s)$ requires an iterative sub-routine, a major potential roadblock. In what follows, we propose a principled, practical solution to these problems.

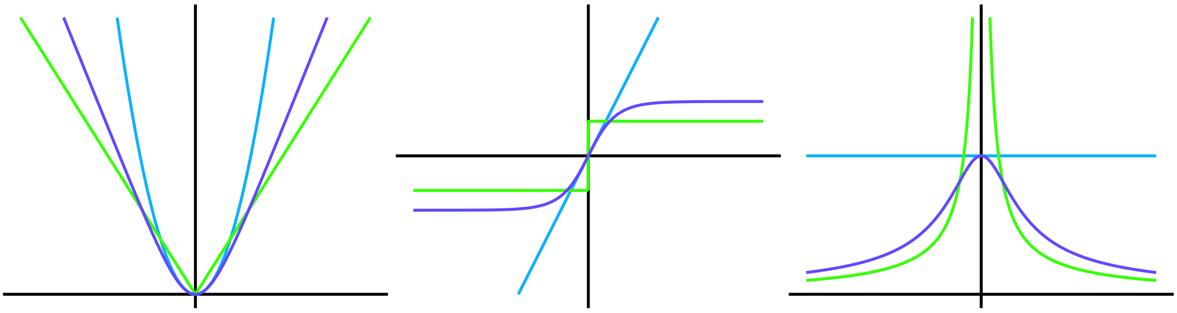

Figure 3.3: From left to right, each figure houses the graphs of $\rho(u)$, $\rho'(u)$, and $\rho(u)/u$ respectively. Colours denote different choices for ρ, namely the ℓ_2 loss (turquoise), the ℓ_1 loss (green), and the Gudermannian function (purple) from Example 7.

3.3.3 Deriving a fast minimizer

Here we pursue an efficient routine for approximately minimizing the robust loss $\widehat{L}(h;\rho,s)$, in the context of the general regression task ($\boldsymbol{z}=(\boldsymbol{x},y)$, with $y\in\mathbb{R}$). A useful heuristic strategy follows from noting that given any candidate $h\in\mathcal{H}$, and computing a central tendency metric γ (e.g., the median or average of $\{l(h;\boldsymbol{z}_i)\}_{i=1}^{n}$), since $l\geq 0$, in order for $\widehat{L}(h;\rho,s)$ to be small, it is *necessary* that the deviations $|l(h;\boldsymbol{z}_i)-\gamma|$ be small for most i. To see this, note that if most deviations are say larger than A, then there must be some points where $\widehat{L}$ is far to the right, that is i where

$$\widehat{L}(h;\rho,s)-l(h;\boldsymbol{z}_i)>A, \text{ which implies } \widehat{L}(h;\rho,s)>A.$$

With this condition in hand, note that the quantity

$$q(h):=\sum_{i=1}^{n}\rho\left(\frac{l(h;\boldsymbol{z}_i)-\gamma}{s}\right)$$

in fact directly measures these deviations. If most points are far away from γ, then $q(h)$ will be large; if most points are close to γ, then $q(h)$ will be small.

Our new task then, is to minimize $q(\cdot)$ in h. Fortunately, this can be done efficiently, using the re-weighting idea (see $\widehat{L}(h;\boldsymbol{u})$) discussed earlier. More precisely, let us set the weights to

$$\boxed{\alpha_i(h)=\rho'\left(\frac{(l(h;\boldsymbol{z}_i)-\gamma)}{s}\right)/\left(\frac{l(h;\boldsymbol{z}_i)-\gamma}{s}\right)}$$

For proper ρ (see section 3.4), we can ensure $0\leq\alpha_i\leq 1$, and intuitively α_i will be very small when $l(h;\boldsymbol{z}_i)$ is inordinately far away from γ. Solving a re-weighted least squares problem, namely

$$\min\sum_{i=1}^{n}\alpha_i(y_i-g(\boldsymbol{x}_i))^2, \quad \text{s.t. } g\in\mathcal{H}$$

can typically be done very quickly, as Example 6 illustrates. What does this re-weighted least squares solution have to do with minimizing $q(\cdot)$? Fortunately, fixing any h, if we set update F as

$$\boxed{F(h):=\operatorname*{arg\,min}_{g\in\mathcal{H}}\sum_{i=1}^{n}\alpha_i(h)(y_i-g(\boldsymbol{x}_i))^2}$$

then using classic results from the robust statistics literature [30, Ch. 7], we have that

$$q(F(h)) \leq q(h)$$

meaning the update from h to $F(h)$ is guaranteed to move us "in the right direction." That said, as our motivating condition was necessary, but not sufficient, the simplest approach is to *check* if this update actually monotonically improves the objective $\widehat{L}(\cdot;\rho,s)$, namely:

$$\boxed{\text{Update to } F(h) \text{ if and only if } \widehat{L}(F(h)) < \widehat{L}(h).}$$

The merits that this technique offers are clear: if we limit the number of iterations to T, then over $t = 1, 2, \ldots, T$ we need only compute $\widehat{L}$ *once* per iteration, meaning that the sub-routine for acquiring $\widehat{L}$ will only be called upon at most T times total. Initializing some $h_{(0)}$ and following the procedure just given, with re-centred (via the term γ) and re-scaled (via the factor s) observations at each step, we get Algorithm 1 below.

Algorithm 1 Fast robust loss minimizer (`fRLM`)

```
for $t \in [T]$ do
    $u_i \leftarrow \left(l(h_{(t-1)}; \boldsymbol{z}_i) - \gamma(D_{(t-1)})\right) / s(D_{(t-1)})$
    $\alpha_i \leftarrow \rho'(u_i)/u_i$    ▷ Downweight errant points; $i \in [n]$.
    $\widetilde{h} \leftarrow \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \alpha_i (y_i - h(\boldsymbol{x}_i))^2$    ▷ Fast approximate update.
    $D_{(t)} \leftarrow \{l(\widetilde{h}; \boldsymbol{z}_i)\}_{i=1}^{n}$    ▷ Compute loss for new candidate.
    $\widehat{L}_{(t)} \leftarrow \arg\min_{\theta \in \mathbb{R}} \sum_{i=1}^{n} \rho\left(\frac{l(\widetilde{h}; \boldsymbol{z}_i) - \theta}{s(D_{(t)})}\right)$    ▷ Evaluate using robust loss.
    if $\widehat{L}_{(t)} < \widehat{L}_{(t-1)}$ then    ▷ Check for monotonic improvement.
        $h_{(t)} \leftarrow \widetilde{h}$
    else
        return $h_{(t-1)}$
    end if
end for
```

Example 6 (Update under linear model). In the special case of a linear model where $h(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$ for some vector $\boldsymbol{w} \in \mathbb{R}^d$, then inverting a $d \times d$ matrix and then some matrix multiplication is all that is required. Writing $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T$ for the $n \times d$ design matrix, $\boldsymbol{y} = (y_1, \ldots, y_n)$, $h(X) = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_n))$, and $U = \operatorname{diag}(u_1, \ldots, u_n)$, then the solution is $(X^TUX)^{\dagger}X^TU(\boldsymbol{y} - h(X))$, where $(\cdot)^{\dagger}$ denotes the Moore-Penrose inverse. ■

Example 7 (Choice of ρ function). Extreme examples of ρ, the convex function used in (4.4), are the ℓ_2 and ℓ_1 losses, namely $\rho(u) = u^2$ and $\rho(u) = |u|$. These result in estimates of the sample mean and median respectively. A more balanced choice might be $\rho(u) = \log\cosh(u)$. We can also define ρ in terms of its derivative; for example, one useful choice is

$$\rho(u) = \int_0^u \psi(x)\,dx, \quad \psi(u) = 2\operatorname{atan}(\exp(u)) - \pi/2,$$

where ψ here is the function of Gudermann [1, Ch. 4], though there are numerous alternatives (see Appendix A.1). ■

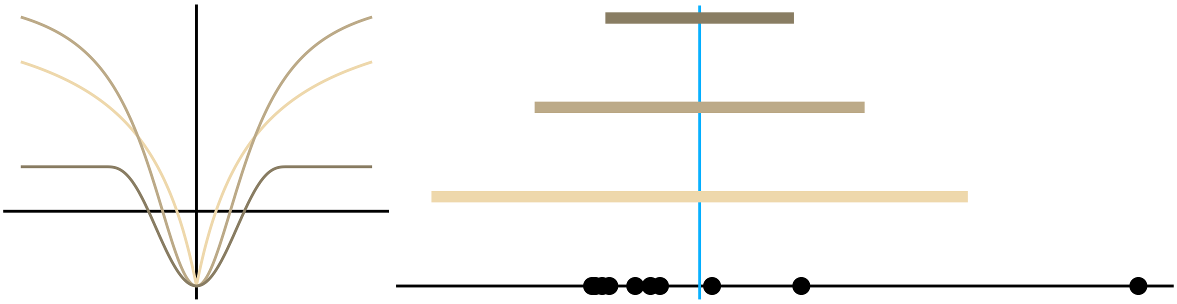

Figure 3.4: In the left plot, we have the graphs of three χ choices with common value $\chi(0)$. From light to dark brown, χ is respectively the absolute Geman-type, quadratic Geman-type, and Tukey function (see Example 8). In the right plot, we have randomly generated data D, and solved (4.6) using the three χ functions in the left plot (colours correspond), with $\gamma(D)$ as the sample mean (turquoise rule). Coloured horizontal rules in ± direction from $\gamma(D)$ represent $s(D)$ for each choice of χ.

3.3.4 Actual computation of key quantities

Here we discuss precisely how we carry out the various sub-routines required in Algorithm 1, namely the tasks of initialization, re-centring, re-scaling, and finding robust loss estimates. Initialization is the first and the easiest: $h_{(0)}$ is initialized to the ℓ_2 empirical risk minimizer. When this value is optimal, it should be difficult to improve $\widehat{L}$, and thus the algorithm should finish quickly; when it is highly sub-optimal, this should result in a large value for $\widehat{L}(h_{(0)}; \rho, s)$, upon which subsequent steps of the algorithm seek to improve.

The “pivot” term γ is computed given a set of losses $D = \{l(h; \boldsymbol{z}_i)\}_{i=1}^{n}$ evaluated at some h; in particular, the losses are computed for $h_{(t-1)}$ at iteration t of Algorithm 1. This $\gamma(D)$ is used to centre the data; terms $l(h; \boldsymbol{z}_i)$ which are inordinately far away from $\gamma(D)$, either above or below, are treated as errant. One natural choice that requires sorting the data is the median D. A rough but fast choice is the arithmetic mean of D, which we have used throughout our tests.

As with γ, we carry out the re-scaling of our observations using D, denoting a set of losses. While there exist theoretically optimal scaling strategies [14], these require knowledge of $\operatorname{var}_{\mu} l(h; \boldsymbol{z})$ and setting of an additional confidence parameter. Since estimating second-order moments in order to estimate first-order moments is highly inefficient, we take the natural approach of using γ to centre the data, seeking a measure of how dispersed these losses are about this pivot, which will be our scale estimate. More concretely, for D induced by $h \in \mathcal{H}$, we seek any s satisfying

$$\sum_{i=1}^{n} \chi\left(\frac{l(h; \boldsymbol{z}_i) - \gamma(D)}{s}\right) = 0, \quad s > 0. \tag{3.2}$$

as our choice for $s(D)$. Here χ is an even function, assumed to satisfy $\chi(0) < 0$ and $\chi(u) > 0$ as $u \to \pm\infty$, ensuring that the scale is neither too big nor too small when compared with the deviations; see Figure 3.4 and Hampel et al. [25] for both theory and applications of this technique.

Our definition of $s(D)$ in (4.6) is implicit, as indeed is the robust loss computation $\widehat{L}$ in (4.4). We thus require iterative procedures to acquire sufficiently good approximations to these desired quantities. Updates taking a fixed-point form are typical for this sort of exercise, and we use the following two routines. Starting with the location estimate for h and given $s > 0$, we run

$$\widehat{\theta}_{(k+1)} \leftarrow \widehat{\theta}_{(k)} + \frac{s}{n} \sum_{i=1}^{n} \rho' \left(\frac{l(h; \boldsymbol{z}_i) - \widehat{\theta}_{(k)}}{s} \right) \tag{3.3}$$

noting that this has the desired fixed point, namely a stationary point of the function in (4.4) to be minimized in θ. For the scale updates, centred by $\gamma \in \mathbb{R}$, we run

$$s_{(k+1)} \leftarrow s_{(k)} \left(1 - \frac{1}{\chi(0)n} \sum_{i=1}^{n} \chi \left(\frac{l(h; \boldsymbol{z}_i) - \gamma}{s_{(k)}} \right) \right)^{1/2} \tag{3.4}$$

which has a fixed point at the desired root sought in (4.6).

Intuitively, for h and D, we expect that as $k \to \infty$

$$\widehat{\theta}_{(k)} \to \widehat{L}(h; \rho, s) \text{ and } s_{(k)} \to s(D),$$

and indeed such properties can be both formally and empirically established (see section 3.4.4).

Example 8 (Role of scale, choice of χ). Take the simple choice of $\chi(u) := u^2 - \beta$ for any fixed $\beta > 0$. If we have data set D with $|D| = n$, and let $\gamma(D)$ be the sample mean, then it immediately follows from (4.6) that $s(D)^2 = (n-1)\,\mathrm{sd}(D)/(n\beta)$, namely a re-scaled sample standard deviation. Countless alternatives exist; one simple and useful choice is the Geman type function

$$\chi(u) = \frac{|u|^p}{1 + |u|^p} - \beta, \quad p \in \{1, 2\}$$

which originate in widely-cited image processing literature [24, 23] and also appear in machine learning work [50]. More classical choices include the bi-weight antiderivative of Tukey (see Appendix A.1), which has seen much use in robust statistics over the past half-century [25, Section 2.6]. ■

3.3.5 Summary of `fRLM` algorithm

To recapitulate, we have put forward a procedure for minimizing the robust loss $\widehat{L}(h; \rho, s)$ in h, by using a fast re-weighted least squares technique that is guaranteed to improve a quantity (q above) very closely related to the actual unwieldy objective $\widehat{L}$. Using the iterative nature of this routine, we can perform the re-scaling and location estimates sequentially (rather than simultaneously), making for simple and fast updates. All together, this allows us to leverage the ability of ρ to truncate errant observations, while utilizing the fast approximate minimization program to alleviate issues with $\widehat{L}$ being implicit, all without using moment oracles for scaling as in the analysis of Catoni [14] and Brownlees et al. [12], which are notable merits of our proposed approach.

This algorithm makes use of statistical quantities that are defined as the minimizer of a class of estimators. As discussed in our literature review of section 3.2, the properties of learning algorithms that leverage these statistics have been analyzed by Brownlees et al. [12]. This does not, however, capture the properties of the resulting estimator itself: how does it behave as a function of sample size? Does it converge to a readily-interpreted parameter? We address these questions in the following section.

3.4 Theoretical analysis

In this section, we formulate the problem of interest with a bit more rigour in 3.4.1, give some fundamental existence results in 3.4.2, and then show that robust loss minimizers converges in a manner analogous to classical M-estimators in 3.4.3, using computationally convergent sub-routines examined in 3.4.4. All proofs are relegated to section 3.8.

3.4.1 Preliminaries

Data model The learning problem, as discussed in the previous sections, is that of predicting response $y \in \mathbb{R}$, given an instance $\boldsymbol{x} \in \mathbb{R}^d$, based on a sequence of pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ generated independently from an unknown distribution, $\boldsymbol{z} := (\boldsymbol{x}, y) \sim \mu$. Denote by $\mathcal{H}$ a collection of candidates $h : \mathcal{X} \to \mathbb{R}$ from which the learning algorithm will select an appropriate member. The task is of an “agnostic” nature, in that we do not know or assume knowledge of the relation between y and $\boldsymbol{x}$, all we want is to find an $h \in \mathcal{H}$ which reliably approximates $h(\boldsymbol{x}) \approx y$, without concern of identifying any true underlying model.

Evaluation mechanism To facilitate both formal analysis and the learning decision process, a loss function $l(h; \boldsymbol{z}) \geq 0$ will be utilized, which evaluates candidate h upon the random draw of $\boldsymbol{z}$, with smaller values being interpreted as more desirable, or a “better fit.” We shall frequently use $\widehat{h}$ to denote the output of an algorithm, typically as $\widehat{h}_n(\boldsymbol{x}) := \widehat{h}(\boldsymbol{x}; \boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$, a process which takes the n-sized data sample and returns a function $\widehat{h}_n \in \mathcal{H}$ to be used for prediction. A standard metric of generalization ability is the risk

$$L_\mu(h) := \mathbf{E}_\mu \, l(h; \boldsymbol{z}) = \int l(h; \cdot) \, d\mu.$$

One considers the performance of an algorithm $\widehat{h}_n$ to be good if the risk is sufficiently small, up to computational cost. Since μ is unknown, this can either be estimated formally, using inequalities that provide high-probability confidence intervals for this error over the random draw of the sample, or via controlled simulations where the performance metrics are computed over many independent trials.

Example 9. As a concrete case, the classical linear regression model with quadratic risk has $\boldsymbol{z} = (\boldsymbol{x}, y)$ with $h(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ for some $\boldsymbol{w} \in \mathbb{R}^d$, and $l(h; \boldsymbol{z}) = (y - \boldsymbol{w}^T \boldsymbol{x})^2$. When the model is correctly specified, i.e., when we have $y = \boldsymbol{w}_0^T + \epsilon$ for an unknown $\boldsymbol{w}_0 \in \mathbb{R}^d$, and noise $\mathbf{E}_\mu \, \epsilon = 0$, the loss takes on a convenient form, making additional results easy to obtain, though our general approach does not require such assumptions.

Additional notation We shall denote by μ a probability on $\mathbb{R}^{d+1}$, equipped with some appropriate σ-field, say the Borel sets $\mathcal{B}_{d+1}$. Let μ_n denote the empirical measure supported on the sample, namely $\mu_n(B) := n^{-1} \sum_{i=1}^{n} I\{\boldsymbol{z}_i \in B\}$, $B \in \mathcal{B}_{d+1}$. Expectation of vectors is naturally element-wise, namely $\mathbf{E}_\mu(\boldsymbol{x}, y) = (\mathbf{E}_\mu \, x_1, \ldots, \mathbf{E}_\mu \, x_d, \mathbf{E}_\mu \, y)$, and we shall use $\operatorname{var}_\mu \boldsymbol{z}$ to denote the $(d+1) \times (d+1)$ covariance matrix of $\boldsymbol{z}$, and so forth. $\mathbf{P}$ will be used to denote a generic probability measure, though in almost all cases it will be over the n-sized data sample, and thus correspond to the product measure μ^n. Let $[k] := \{1, \ldots, k\}$ for integer k.

3.4.2 Properties of the robust objective

Generalization performance is completely captured by the *distribution* of $l(h; \boldsymbol{z})$. Unfortunately, inferring this distribution from a finite sample is exceedingly difficult, and so we estimate parameters of this distribution to gain insight into performance; the expected value

$L_\mu(h)$ is a case in point. In pursuit of a routine for estimating the risk, with low variance and controllable risk, the basic strategy ideas in section 3.3 seem intuitively promising. Here we show that following the strategy outlined, one can create a procedure which is valid in a statistical sense, under very weak assumptions.

Our starting point is to introduce new parameters, distinct from the risk, which have controllable bias, and can be approximated more reliably than the expected value, using a finite sample. The following definition specifies such a parameter class.

Definition 10 (General target parameters)**.** For $\rho : \mathbb{R} \to [0, \infty)$ and scale $s > 0$, define

$$\theta^*(h) \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \mathbf{E}_\mu \, \rho\left(\frac{l(h;\boldsymbol{z}) - \theta}{s}\right) \tag{3.5}$$

where s may depend on h. We require that ρ be symmetric about 0, with $\rho(0) = 0$, and further that

$$\rho(u) = O(u), \text{ as } u \to \pm\infty$$
$$\frac{\rho(u)}{u^2} \to K < \infty, \text{ as } u \to 0.$$

For clean notation, normalize such that $K = 1/2$. If ρ is differentiable, denote $\psi := \rho'$. If twice-differentiable and $\psi' > 0$, say that ρ *specifies a robust objective*, namely $\theta^*(\cdot)$.

Remark 11. The logic here is as follows: the mean $L_\mu(h)$ can be considered a good target if the data are approximately symmetric, or if (regardless of symmetry) they are tightly concentrated about the mean. In both of these cases, we have $\theta^*(h) \approx L_\mu(h)$. To see this, If $l(h;\boldsymbol{z})$ is symmetric about some l_0, that is to say for all $\varepsilon > 0$,

$$\mathbf{P}\{l(h;\boldsymbol{z}) - l_0 \geq \varepsilon\} = \mathbf{P}\{-(l(h;\boldsymbol{z}) - l_0) \geq \varepsilon\},$$

it is sufficient to minimize

$$\int_{\{l(h;\boldsymbol{z}) \geq l_0\}} \rho\left(\frac{l(h;\boldsymbol{z}) - \theta}{s}\right) d\mu$$

on $[l_0, \infty)$, where $\theta = l_0 = L_\mu(h)$ is a solution. Thus in the symmetric case, we end up with $\theta^*(h) = L_\mu(h)$, irrespective of scaling and truncating mechanisms. Here "tightly concentrated" is relative, in the sense that

$$|l(h;\boldsymbol{z}) - L_\mu(h)| < s$$

with high probability. Since we have required $\rho(u) \sim u^2$, tight concentration would imply $\theta^*(h) \approx L_\mu(h)$. As for the linear growth requirement, $\rho(u) = o(u^2)$ as $u \to \pm\infty$ is necessary if we are to reduce dependence on the tails, but making the much stronger requirement of $\rho(u) = O(u)$ is very useful as it implies that ψ is bounded. Note that of the functions ρ given in Example 7, the ℓ_p choices do not meet our criteria, but the Gudermannian and log cosh choices both satisfy all conditions. ■

This $\theta^*(\cdot)$, a new parameter of the loss $l(\cdot;\boldsymbol{z})$, can be readily interpreted as an alternative performance metric to the risk $L_\mu(\cdot)$. Denote optimal performance in this metric on $\mathcal{H}$ by

$$\theta^*(\mathcal{H}) := \inf_{h \in \mathcal{H}} \theta^*(h) \geq 0 \tag{3.6}$$

and the empirical estimate of these parameters by

$$\widehat{\theta}(h) \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \rho\left(\frac{l(h;\boldsymbol{z}_i) - \theta}{s}\right). \tag{3.7}$$

Note that we call this the empirical estimate as we have simply replaced μ by μ_n in the definition of θ^* to derive $\widehat{\theta}$. The procedure of Algorithm 1 outputs an approximation of

$$\widehat{h}_n \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \widehat{\theta}(h) \tag{3.8}$$

which is none other than a minimizer of the robust loss $\widehat{\theta}$, an empirical estimate of the alternative performance metric θ^*.

First, we show that these new "objectives" are indeed well-defined objective functions, which is important since our algorithm seeks to minimize them.

Lemma 12 (Existence of parameter and its estimate)**.** *Let ρ specify a robust objective $\theta^*(h)$. This function is well-defined in h, in that for each $h \in \mathcal{H}$, the value of $\theta^*(h)$ is uniquely determined, characterized by*

$$\mathbf{E}_{\mu}\, \psi\left(\frac{l(h;\boldsymbol{z}) - \theta^*(h)}{s}\right) = 0. \tag{3.9}$$

Analogously, the empirical estimate is uniquely defined, and almost surely given by

$$\sum_{i=1}^{n} \psi\left(\frac{l(h;\boldsymbol{z}_i) - \widehat{\theta}(h)}{s}\right) = 0. \tag{3.10}$$

With a well-defined objective function, next we consider the existence of the minimizer of this new objective. While measurability is by no means our chief concern here, for completeness we include a technical result useful for proving the existence of a valid minimizer of the proxy objective.

Lemma 13. *Let ρ be even and continuously differentiable with ρ' non-decreasing on $\mathbb{R}$. Let $s_h : \mathbb{R}^{d+1} \to \mathbb{R}_+$ be measurable for all $h \in \mathcal{H}$. For any $n \in \mathbb{N}$, denote sequence space $\mathcal{Z} := (\mathbb{R}^{d+1})^n$. Then defining*

$$\widehat{\theta}(h) := \inf\left(\operatorname*{arg\,min}_{u \in \mathbb{R}} \sum_{i=1}^{n} \rho\left(\frac{l(h;\boldsymbol{z}_i) - u}{s_h(\boldsymbol{z}_i)}\right)\right), \tag{3.11}$$

we have that $\widehat{\theta}$ is measurable as a function on $\mathcal{H} \times \mathcal{Z}$.

This gives us a formal definition of $\widehat{\theta}(h)$ which has the desired property specified by (3.7). It simply remains to show that we can always minimize this objective in h.

Theorem 14 (Existence of minimizer)**.** *Let $h \mapsto s_h$ be continuous and $s_h > 0$, $h \in \mathcal{H}$. Using $\widehat{\theta}$ from Lemma 13, define*

$$\widehat{\theta}(\mathcal{H}) := \inf_{h \in \mathcal{H}} \widehat{\theta}(h). \tag{3.12}$$

For any ρ specifying a robust objective (Defn. 10), and any sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$,

$$\exists\, \widehat{h} \in \mathcal{H}, \quad \widehat{\theta}(\widehat{h}) = \widehat{\theta}(\mathcal{H}),$$

and there exists a random variable $\widehat{h}_n$ such that $\mathbf{P}\{\widehat{\theta}(\widehat{h}_n) = \widehat{\theta}(\mathcal{H})\} = 1$.

There are many potential methods for carrying out the scaling in practice. Here we verify that the simple method proposed in section 3.3 does not disrupt the assurances above. Let us start with a definition.

Definition 15 (General-purpose scale)**.** For random variable $x \sim \nu$, introduce even function $\chi : \mathbb{R} \to \mathbb{R}$, non-decreasing on $\mathbb{R}_+$, which satisfies

$$0 < \lim_{|u|\to\infty} \chi(u), \quad \chi(0) < 0.$$

Let $\beta \geq 0$ be the value such that $\chi(0) = -\beta$. With the help of χ and pivot term γ_ν which may depend on ν, define

$$\sigma_\nu := \inf \left\{ \sigma > 0 : \mathbf{E}\,\chi\left(\frac{x - \gamma_\nu}{\sigma}\right) = 0 \right\}. \tag{3.13}$$

With this definition in place, substituting $\nu = \mu_n$ yields an empirical scale estimate

$$s_h = \inf \left\{ \sigma > 0 : \sum_{i=1}^{n} \chi\left(\frac{l(h; \boldsymbol{z}_i) - \gamma_{\mu_n}(h)}{\sigma}\right) = 0 \right\} \tag{3.14}$$

with $\sum_{i=1}^{n} l(h; \boldsymbol{z}_i)/n$ a natural pivot value, though we certainly have more freedom in constructing $\gamma_{\mu_n}(h)$, as the following result shows.

Proposition 16 (Validity of scaling mechanism)**.** *If $\gamma_{\mu_n}(h) < \infty$ almost surely for all $h \in \mathcal{H}$, and χ (Defn. 15) is increasing on $\mathbb{R}_+$, then the minimizer $\widehat{h}_n$ (3.8) as constructed in Theorem 14 satisfies*

$$\widehat{\theta}(\widehat{h}_n) = \widehat{\theta}(\mathcal{H})$$

almost surely when scaling with $s = s_h$ as in (3.14).

Note that $\gamma_{\mu_n}(h)$ here corresponds directly to $\gamma(D)$ in Algorithm 1, where $D = \{l(h; \boldsymbol{z}_i)\}_{i=1}^{n}$. One intuitively expects that taking a large $s > 0$ would imply $\theta^* \approx \mathbf{E}_\mu\, x$. The following basic fact makes this precise.

Theorem 17 (Scaling parameter and bias)**.** *Let ρ specify a robust target, normalized such that $\rho'(0) = 1$. Let x be an arbitrary random variable with distribution ν. Assuming $\mathbf{E}_\nu\, |x|^3 < \infty$ and $\|\rho^{(4)}\|_\infty < \infty$, it follows that*

$$|\theta^* - \mathbf{E}_\nu\, x| \leq c s^{-2}, \quad s > 0$$

for constant $c > 0$.

Example 18 (Higher-order moments of ρ). It is admittedly tedious to check the condition of $\|\rho^{(4)}\|_\infty$, but for typical examples we can easily see that the property is satisfied. For example for the Gudermannian function, we have

$$\rho^{(4)}(u) = \eta'(u) - \frac{12\exp(3u)}{(1+\exp(2u))^2} + \frac{16\exp(5u)}{(1+\exp(2u))^3}.$$

Since η is Lipschitz and the latter two terms vanish in the limit, $|\rho^{(4)}|_\infty < \infty$ is immediate. An upper bound of 1 is sufficient.

Similarly for the logistic function,

$$\rho^{(4)}(u) = \frac{6c_1c_2^3\exp(-3c_2u)}{(1+\exp(-c_2u))^4} - \frac{4c_1c_2^3\exp(-2c_2u)}{(1+\exp(-c_2u))^3} - c_2\eta'(u),$$

and for log cosh,

$$\rho^{(4)}(u) = (-2)\left((\eta(u))^2 + \tanh(u)\eta'(u)\right),$$

for which bounds follow again via bounds on η and its derivative. ■

With basic facts in hand pertaining to the estimator used to construct a robust objective, we proceed to look at some convergence properties of the estimators and computational procedures concerned in the sections 3.4.3–3.4.4.

3.4.3 Statistical convergence

For some context, we start with a well-known consistency property of M-estimators, adapted to our setting.

Theorem 19 (Pointwise consistency under known scale)**.** *For any ρ specifying a robust objective, fixing any $h \in \mathcal{H}$ and $s > 0$, then*

$$\mathbf{P}\left\{\lim_{n\to\infty} \widehat{\theta}(h) = \theta^*(h)\right\} = 1.$$

Note that this strong consistency result is "pointwise" in the sense that the event of probability 1 is dependent on the choice of $h \in \mathcal{H}$. Were we to take a different $h' \in \mathcal{H}$, while the probability would still be one, the events certainly need not coincide. This becomes troublesome since $\widehat{h}_n$ will in all likelihood take a different h value for distinct samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. Intuitively, we do expect that as n grows, the estimate $\widehat{h}_n$ should get progressively better and in the limit we should have

$$\widehat{\theta}(\widehat{h}_n) \to \theta^*(\mathcal{H}), \quad n \to \infty.$$

Here we show that such a property does indeed hold, focusing on the case where $\mathcal{H}$ is a linear model, though the assumptions on $\boldsymbol{x}$ and y are still completely general (agnostic). More precisely, we assume that $\mathcal{H}$ is defined by a collection of real-valued functions $\varphi_1, \ldots, \varphi_k$ on $\mathbb{R}^d$, and a bounded parameter space $\mathcal{W} \subset \mathbb{R}^k$. The model is thus of the form

$$\mathcal{H} = \left\{ h = \sum_{j=1}^{k} w_j\varphi_j : (w_1, \ldots, w_k) \in \mathcal{W} \right\}. \tag{3.15}$$

Under this model, the class of parameters given in Defn. 10 and the corresponding estimators (3.7) are such that convenient uniform convergence results are available using standard combinatorial arguments. First a general lemma of a technical nature.

Lemma 20 (Uniform strong convergence)**.** *Let $\mathcal{H}$ satisfy (3.15), and ρ specify a robust objective (Defn. 10). Denoting $\Lambda := \mathcal{H} \times \mathbb{R} \times \mathbb{R}_+$, $\lambda := (h, u, s) \in \Lambda$, and*

$$\psi(\boldsymbol{z}; \lambda) := \psi\left(\frac{l(h; \boldsymbol{z}) - u}{s}\right),$$

we have that

$$\lim_{n\to\infty} \sup_{\lambda\in\Lambda} \left| \frac{1}{n}\sum_{i=1}^{n} \psi(\boldsymbol{z}_i; \lambda) - \mathbf{E}_\mu\, \psi(\boldsymbol{z}; \lambda) \right| = 0$$

almost surely.

A corollary of this general result will be particularly useful.

Corollary 21. *The robust objective minimizer $\widehat{h}_n$ defined in (3.8), equipped with any scaling mechanism s depending on $\widehat{h}_n$ (and thus potentially random), satisfies*

$$\lim_{n\to\infty} \mathbf{E}_{\mu}\, \psi\left(\frac{l(\widehat{h}_n;\boldsymbol{z}) - \widehat{\theta}(\widehat{h}_n)}{s}\right) = 0$$

almost surely.

These facts are sufficient for showing that a very natural analogue of the strong pointwise consistency of M-estimators (Theorem 19) holds in a uniform fashion for our robust objective minimizer $\widehat{h}_n$.

Theorem 22 (Consistency analogue)**.** *Let $\widehat{h}_n$ be determined by (3.8) equipped with any fixed scaling mechanism $s_h : \mathbb{R}^{d+1} \to \mathbb{R}_+$. Let ρ specify a robust objective, with ρ' concave on $\mathbb{R}_+$. If there exists constants s_1, s_2, ϵ such that*

$$\begin{aligned} &0 < s_1 \leq s_h(\boldsymbol{z}) \leq s_2 < \infty \\ &0 < \epsilon \leq \inf_{h \in \mathcal{H}} \mathbf{E}_{\mu}\, \psi'(l(h;\boldsymbol{z})/s_1) \end{aligned}$$

then it follows that

$$\mathbf{P}\left\{\lim_{n\to\infty} \widehat{\theta}(\widehat{h}_n) = \theta^*(\mathcal{H})\right\} = 1.$$

That is, $\widehat{\theta}(\widehat{h}_n)$ is a strongly consistent estimator of the optimal value $\theta^*(\mathcal{H})$.

With these rather natural statistical properties understood, we shift our focus to the behaviour of the computational routines used.

3.4.4 Computational convergence

As regards computational convergence, since Algorithm 1 is meant to be a fast approximation to a minimizer of $\widehat{L}(\cdot)$ on $\mathcal{H}$, we should not expect the $\widehat{h}$ produced after $t \to \infty$ iterations to actually converge to the true $\widehat{h}_n$ in (3.8). What we should expect, however, is that the sub-routines (3.3) and (3.4), used to compute $\widehat{L}_{(t)}$ and $s(D_{(t)})$ for *each* t, should converge to the true values specified by (4.4) and (4.6) respectively. We show that this convergence holds.

Proposition 23 (Convergence of updates)**.** *Let ρ specify a robust objective (Defn. 10). Fixing $s > 0$, and any initial value $\widehat{\theta}_{(0)}$, the iterative update $(\widehat{\theta}_{(k)})$ specified in (3.3) satisfies*

$$\lim_{k\to\infty} \widehat{\theta}_{(k)} = \widehat{\theta}(h),$$

recalling that $\widehat{\theta}(h) = \widehat{L}(h;\rho,s)$ from section 3.3. Similarly, for χ as specified by Defn. 15, under some additional regularity conditions on χ, (see proof) we have that for any initialization $s_{(0)} > 0$, the update $(s_{(k)})$ in (3.4) satisfies

$$\sum_{i=1}^{n} \chi\left(\frac{l(h;\boldsymbol{z}_i) - \gamma}{\lim_{k\to\infty} s_{(k)}}\right) = 0.$$

Using ρ as in Defn. 10 and χ as in Prop. 16, note that the above convergence guarantees will not be ambiguous, since the location and scale estimates are uniquely determined.

3.5 Empirical analysis

We derived a new algorithm in 3.3, formally investigated statistical properties in 3.4.2–3.4.3, and computational properties in 3.4.4. Here we evaluate the actual performance of this algorithm against standard competitive algorithms in a variety of situations, including both tightly controlled numerical simulations and real-world benchmark data sets. We seek to answer the following questions.

1. How does the robustification sub-routine depend on the task?
2. How well does `fRLM` (Algorithm 1) generalize off-sample?
3. Fixing ρ, can we still succeed under both light- and heavy-tailed noise?
4. How does performance depend on n and d?

We look directly at the first question in sub-section 3.5.1. For the remaining three questions, our experimental setup and competing algorithms used are described in 3.5.2–3.5.3, and results follow in 3.5.4–3.5.5 where we give concrete responses to these inquiries. All experimental parameters, as well as source code for all methods used, are included in the supplementary source code.[1]

3.5.1 Efficiency of iterative sub-routines

As a complement to the formal convergence properties just examined, we conduct numerical tests in which we run (3.3) and (3.4) until they respectively compute the true $\widehat{\theta}$ and s values up to a specified degree of precision. It is of practical importance to answer the following questions: Do the iterative routines reliably converge to the correct optimal value? How many iterations does this take on average? How does this depend on factors such as the data distribution, sample size, and our choice of ρ and χ?

To investigate these points, we carry out the following procedure. Generating $x_1, \ldots, x_n \in \mathbb{R}$ from some distribution, denote

$$f_1(u) := \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{x_i - u}{s}\right), \quad f_2(u) := \frac{1}{n}\sum_{i=1}^{n} \chi\left(\frac{x_i - \gamma}{u}\right).$$

The location task is to minimize f_1 on $\mathbb{R}$, and the scale task is to seek a root of f_2 on $\mathbb{R}_+$. Two choices of distribution were used. First is $x \sim N(0,3)$, i.e., centred Normal random variables with variance of nine. The second is asymmetric and heavy-tailed, generated as $\exp(x)$ where x is again $N(0,3)$; this is the log-Normal distribution. For f_1, the s value is a parameter; this is set to the standard deviation of the x_i. For f_2, the γ value is a parameter; this is set to the sample mean of the x_i. As for ρ and χ, we examine five choices of each,[2] all defined in Appendix A.1. Initial values are $\widehat{\theta}_{(0)} = n^{-1}\sum_{i=1}^{n} x_i$ and $s_{(0)} = \mathrm{sd}\{x_i\}_{i=1}^{n}$.

In Figure 3.5, we show the average *iterations to converge*, as a function of sample size n, computed as follows. The terminating iteration for these tasks, at accuracy level ε, is defined

$$K_\varepsilon(\widehat{\theta}) := \min\{k : |\widehat{\theta}_{(k)} - \widehat{\theta}_{OR}| \leq \varepsilon\}, \quad K_\varepsilon(s) = \min\{k : |s_{(k)} - s_{OR}| \leq \varepsilon\}$$

[1] All materials available at https://github.com/feedbackward/rtm_code.

[2] For location, we have used the simple algebraic function (denoted here by `al`), the Gudermannian function (`gud`), the modified Huber function (`hmd`), the original Huber function (`hub`), and the log hyperbolic cosine function (`lch`). For scale, we have used the absolute and quadratic settings of the Geman function (respectively `ga` and `gq`), the two transformed robust loss derivatives (quadratic: `pq`; log-quadratic: `lq`), and the bi-weight antiderivative of Tukey (`tuk`). We use ρ set to `lch` to specify χ in the case of `pq` and `lq`.

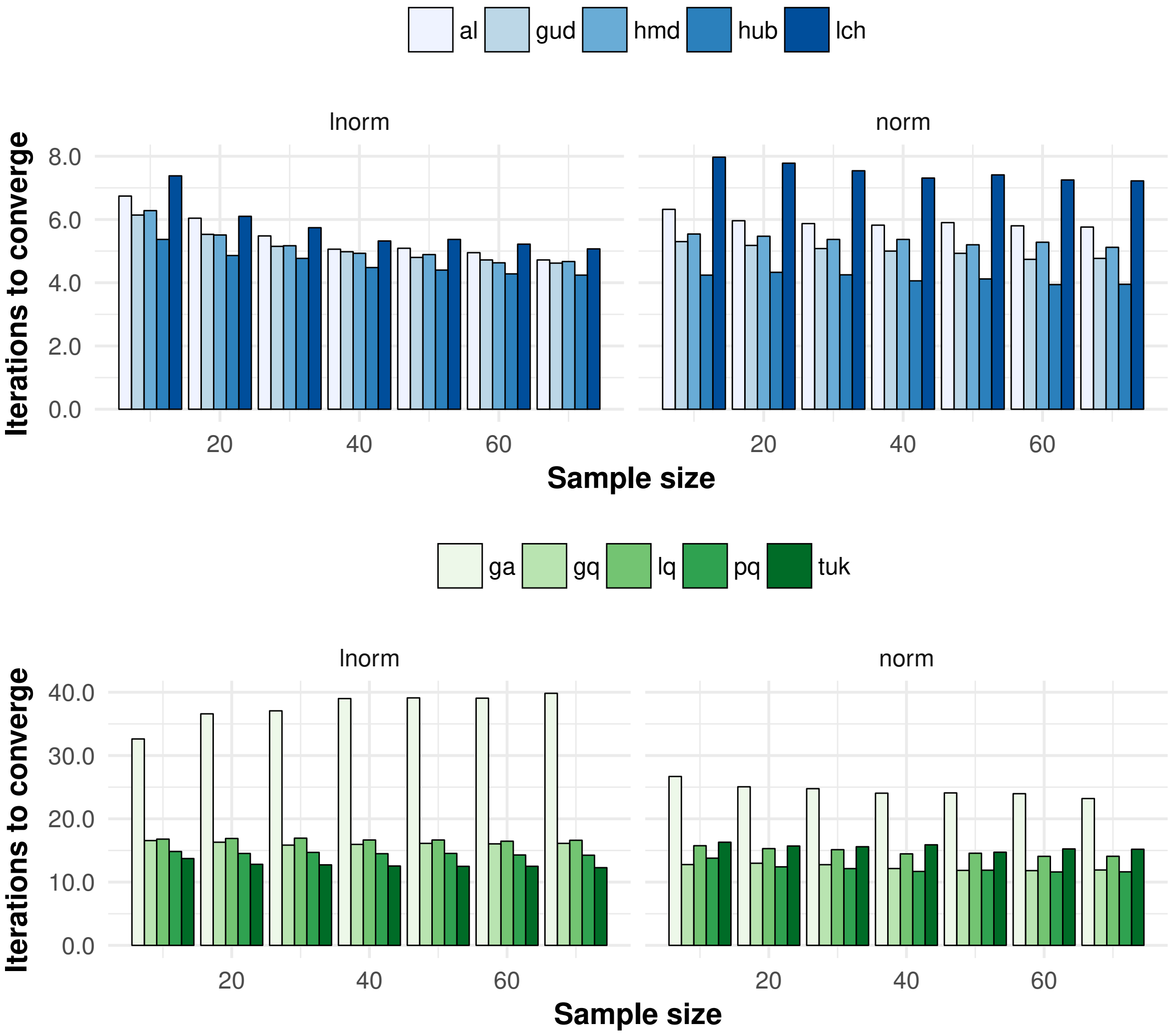

Figure 3.5: Iterations required to reach ε-accurate estimates given n sample, under Normal/log-Normal observations. Top row: average $K_{\varepsilon}(\widehat{\theta})$. Bottom row: average $K_{\varepsilon}(s)$. See Appendix A.1 for ρ and χ definitions.

where $\widehat{\theta}_{OR}$ and s_{OR} are "oracle" values of the minimum/root of f_1/f_2. These are obtained via `uniroot` in R [39], an implementation of Brent's univariate root finder [11], recalling the ρ minimization can be cast as a root-finding problem (Lemma 12). These K_{ε} values are thus the number of iterations required; 100 independent trials are carried out, and the arithmetic mean of these values is taken. Updates $\widehat{\theta}_{(k)}$ and $s_{(k)}$ are precisely as in (3.3) and (3.4). Accuracy level is $\varepsilon = 10^{-4}$ for all trials.

We have convergence at a high level of precision, requiring very few iterations, and this holds uniformly across the conditions observed. As such, the convergence of the routines is just as expected (Proposition 23), and the speed is encouraging. In general, convergence tends to speed up for larger n, and the relative difference in speed is very minor across distinct ρ choices, though slightly more pronounced in the case of χ, but even the slowest choice seems tolerable. Finally, location estimation is slightly slower in the Normal case than in the log-Normal case, while the opposite holds for scale estimation.

3.5.2 Experimental setup

Every experimental condition and trial has us generating n training observations, of the form $y_i = \boldsymbol{w}_0^T \boldsymbol{x} + \epsilon_i, i \in [n]$. Distinct experimental conditions are specified by the setting of (n, d) and μ. Inputs $\boldsymbol{x}$ are assumed to follow a d-dimensional isotropic Gaussian distribution, and thus to determine μ is to specify the distribution of noise ϵ. In particular, we look at several families of distributions, and within each family look at 15 distinct *noise levels*. Each noise level is simply a particular parameter setting, designed such that $\operatorname{sd}_\mu(\epsilon)$ monotonically increases over the range 0.3–20.0, approximately linearly over the levels (cf. Appendix A.5).

To ensure a wide range of signal/noise ratios is spanned, for each trial, $\boldsymbol{w}_0 \in \mathbb{R}^d$ is randomly generated as follows. Defining the sequence $w_k := \pi/4 + (-1)^{k-1}(k-1)\pi/8, k = 1, 2, \ldots$ and uniformly sampling $i_1, \ldots, i_d \in [d_0]$ with $d_0 = 500$, we set $\boldsymbol{w}_0 = (w_{i_1}, \ldots, w_{i_d})$. As such, given our control of noise standard deviation, and noting that the signal to noise ratio in this setting is computed as $\mathrm{SN}_\mu = \|\boldsymbol{w}_0\|_2^2 / \operatorname{var}_\mu(\epsilon)$, the ratio ranges between $0.2 \leq \mathrm{SN}_\mu \leq 1460.6$.

Regarding the noise distribution families, the tests described above were run for 27 different families, but as space is limited, here we provide results for four representative families: log-Normal (denoted `lnorm` in figures), Normal (`norm`), Pareto (`pareto`), and Weibull (`weibull`). Even with just these four, we capture both symmetric and asymmetric families, sub-Gaussian families, as well as heavy-tailed families both with and without finite higher-order moments.

Our chief performance indicator is *prediction error*, computed as follows. For each condition and each trial, an independent test set of m observations is generated identically to the corresponding n-sized training set. All competing methods use common sample sets for training and are evaluated on the same test data, for all conditions/trials. For each method, in the kth trial, some estimate $\widehat{\boldsymbol{w}}$ is determined. To approximate the ℓ_2-risk, compute root mean squared error $e_k(\widehat{\boldsymbol{w}}) := (m^{-1} \sum_{i=1}^{m} (\widehat{\boldsymbol{w}}^T \boldsymbol{x}_{k,i} - y_{k,i})^2)^{1/2}$, and output prediction error as the average of normalized errors $e_k(\widehat{\boldsymbol{w}}(k)) - e_k(\boldsymbol{w}_0(k))$ taken over all trials. While n and d values vary, in all experiments the number of trials is fixed at 250, and test size $m = 1000$.

3.5.3 Competing methods

Benchmark routines used in these experiments are as follows. Ordinary least squares, denoted `ols` and least absolute deviations, denoted `lad`, represent classic methods. In addition, we look at three very modern alternatives, namely three routines directly from the references papers of Minsker [36] (`geomed`), Brownlees et al. [12] (`bjl`), and Hsu and Sabato [28] (`hs`). The `hs` routine used here is a faithful R translation of the MATLAB code published by the authors. Our implementation of `geomed` uses the geometric median algorithm of [49, Eqn. 2.6], and all partitioning conditions as given in the original paper are satisfied. Regarding `bjl`, scaling is done using a sample-based estimate of the true variance bound used in their analysis, with optimization carried out using the Nelder-Mead gradient-free method implemented in the R function `optim`.

For our `fRLM` (Algorithm 1, section 3.3), we tried several different choices of ρ and χ, including those in Appendix A.1, and overall trends were almost identical. Thus as a representative, we use the Gudermannian for ρ and $\chi(u) = \operatorname{sign}(|u|-1)$ as a particularly simple and illustrative example implementation. Estimates of location and scale were carried out by (3.3) and (3.4).

3.5.4 Test results: simulation

Here we assemble the results of distinct experiments which highlight different facets of the statistical procedures being evaluated.

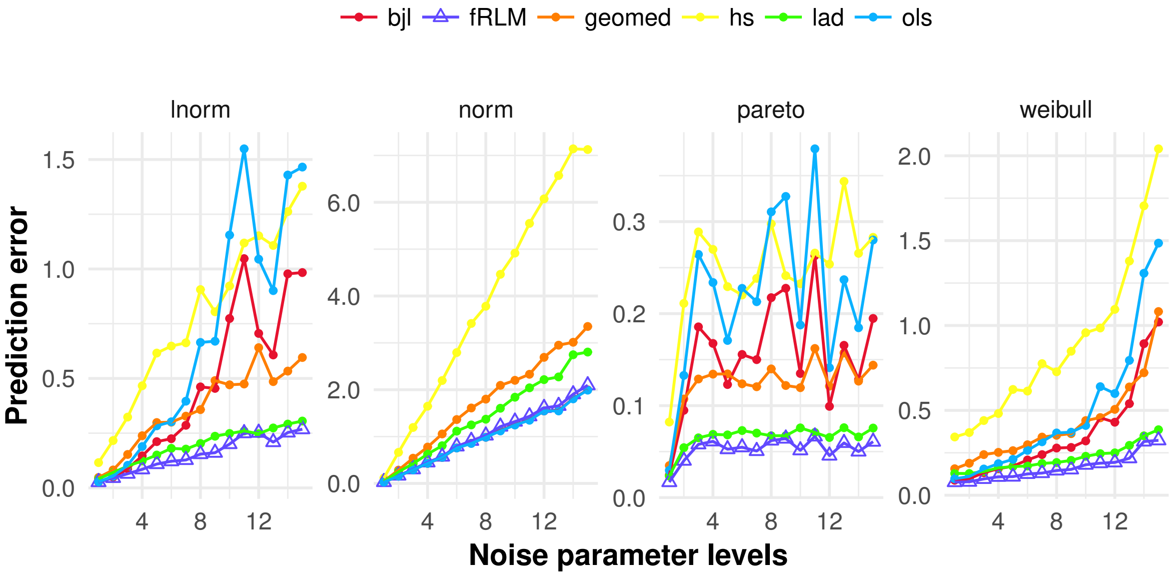

Figure 3.6: Prediction error as a function of noise level, with $n = 15$ and $d = 5$. Moving from left to right on the horizontal axis corresponds to larger noise magnitude.

Performance over noise levels Fig. 3.6 shows how predictive performance deteriorates as the noise magnitude (described in 3.5.2) grows larger, under fixed (n, d) setting. We see that our method closely follow the performance of `ols` only when it is strong (the Normal case), but critically remain stable under settings in which `ols` deteriorates rapidly (all other cases). Our method, much like the other robust methods, incurs a bias by designing objective functions using estimators for target parameters other than the true risk. It is clear, however, that the bias in the case of our method is orders of magnitude smaller than that of competing routines, suggesting that the proposed procedure for minimizing a robust loss is effective. Note that `bjl` needs an off-the-shelf non-linear optimizer and directly requires variance estimates; our routine circumvents these steps, and is seen to be better for it.

Impact of sample size (n grows, d fixed) In Fig. 3.7 we look at prediction error, at the middle noise level, for different settings of n under a fixed d. We have fixed $d = 5$ and the sample size ranges between 12–122. Once again we see that in the Normal case where `ols` is optimal, our routine closely mimics its behaviour and converge in the same way. On the other hand for the heavier-tailed settings, we find that the performance is once against extremely strong, with far better performance under small sample sizes, and a uniformly dominant rate of convergence as n gets large.

Impact of dimension The role played by model dimension is also of interest, and can highlight weaknesses in optimization routines that do not appear when only a few parameters are being determined. Such issues are captured most effectively by keeping the d/n ratio fixed and increasing the model dimension.

Prediction error results are given in Fig. 3.8, at the middle noise level, for different model dimensions ranging over $5 \leq d \leq 140$. The sample size is determined such that $d/n = 1/6$ holds; this is a rather generous size, and thus where we observe deterioration in performance, we infer a lack of utility in more complex models, even when a sample of sufficient size is available. We see clearly that most procedures considered see a performance drop as model

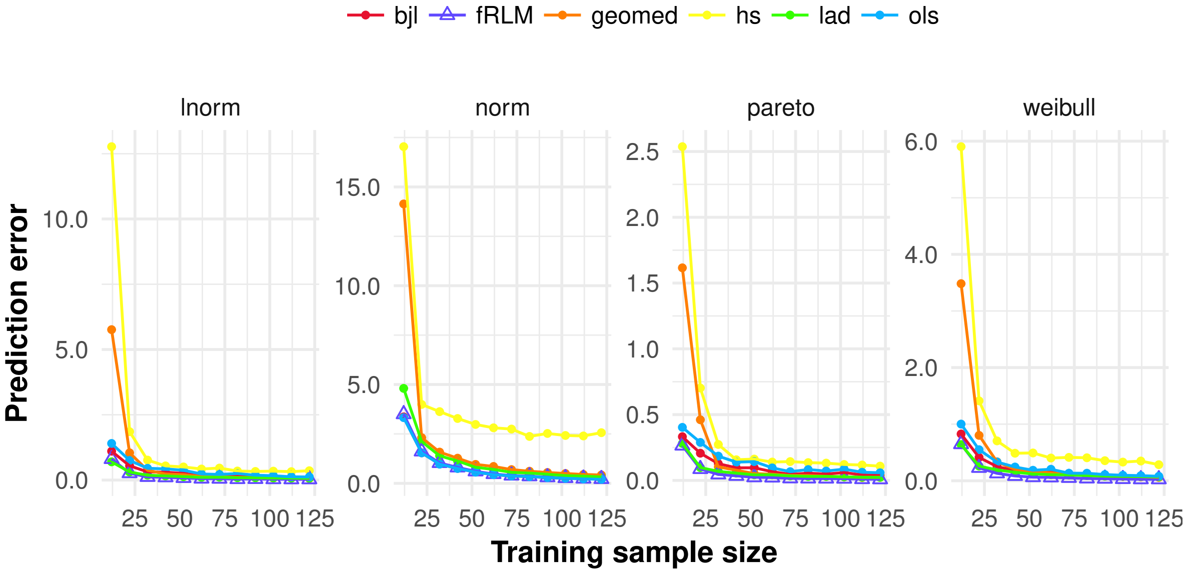

Figure 3.7: Prediction error as a function of sample size n, with $d = 5$, at noise level $= 8$.

dimension grows, whereas our routine performs exactly the same, regardless of dimension size. This is a particularly appealing result illustrating the scalability of our `fRLM` in "bigger" tasks.

3.5.5 Test results: real-world data

We have seen extremely strong performance in the simulated situation; let us see how this extends to a number of real-world domains. The algorithms run are precisely the same as in the simulated cases, just the data is new. We have selected four data sets from a database of benchmark data sets for testing regression algorithms.[3] Our choices were such that the data come from a wide class of domains. For reference, the response variable in `bpres` is blood pressure, in `psych` is psychiatric assessment scores, in `rent` is cost to rent land, and in `oct` is petrol octane rating. All the data sets used here are included with a description in the online code repository referred at the start of this section. Depending on the data set, the dimensionality and sample size of the data sets naturally differ. Our protocol for evaluation is as follows. If the full data set is $\{\boldsymbol{z}_i\}_{i=1}^{N}$, then we take $n = \lceil 0.3N \rceil$ for training, and $m = N - n$ observations for testing. We carry out 100 trials, each time randomly choosing the train/test indices, and averaging over these trials to get prediction error.

Results are given in Fig. 3.9. While the data sets come from wildly varying domains (economics, manufacturing of petroleum products, human physiology and psychology), it is apparent that the results here very closely parallel those of our simulations, which again are the kind of performance that the theoretical exposition of sections 3.3–3.4 would have us expect. Strong performance is achieved with no *a priori* information, and with no fine-tuning whatsoever. Exactly the same routine is deployed in all problems. Of particular importance here is that we are able to beat or match the `bjl` routine under all settings here as well; both of these routines attempt to minimize similar robust losses (defined implicitly), however our routine does it at a fraction of the cost, since we have no need to appeal to general-purpose non-linear optimizers, a very promising result.

[3] Compiled online by J. Burkardt at `http://people.sc.fsu.edu/~jburkardt/`.

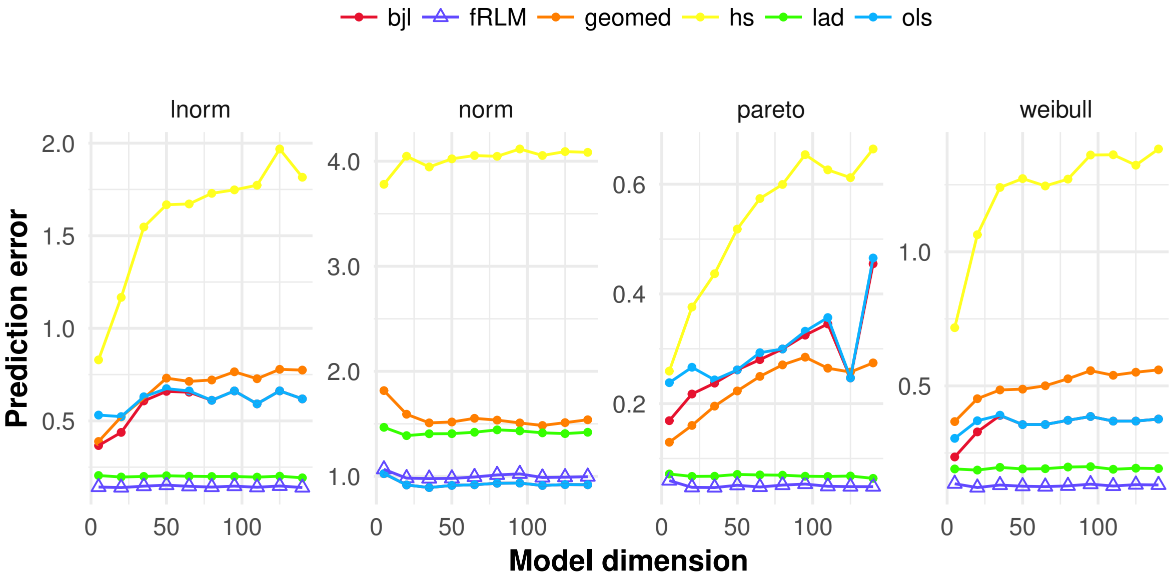

Figure 3.8: Prediction error as a function of model dimension d with fixed ratio $d/n = 1/6$, at noise level = 8.

3.6 Implementation of one-dimensional example

To create Figure 3.1, a simple experiment was carried out, as described in the following paragraphs.

Data generation On the left half, we have the well-behaved symmetric noise setting, while the noise on the right-hand side is asymmetric and heavy-tailed. More precisely, for the left side, we generated $(x_1, y_1), \ldots, (x_n, y_n)$ by $x \sim \text{Unif}[0, 0.5]$ and $y = f(x) + \epsilon$, where the noise ϵ is independent of x, and zero-mean Normal with variance $\mathbf{E}\,\epsilon^2 = 0.25$. The functional relation is $f(x) = \sin(2\pi x)$. To generate the right half, the same procedure was done, this time with $x \sim U[0.5, 1]$, and noise ϵ being a Fréchet random variable with shift 0, scale 1, and shape 2.1, shifted such that $\mathbf{E}\,\epsilon = 0$. The noise magnitude on the right side is larger as well, with $\sqrt{\mathbf{E}\,\epsilon^2} \approx 4$.

Solid curves We illustrate the deterministic parameters of the conditional distribution of response y as a function of input x using solid curves. The solid red curve is the graph of $\mathbf{E}(y; x) = \mathbf{E}\,\epsilon + f(x) = f(x)$, taking $x \in [0, 1]$. Solid green denotes the graph of $\text{med}(y; x) = \text{med}\,\epsilon + f(x)$. On the left side, $\text{med}\,\epsilon = 0$, but on the right half this is not the case and the two graphs diverge. Finally, solid blue denotes the ρ-induced M-estimate of the location of y, conditioned on x. More precisely, the graph of $\widetilde{y}(x) := \arg\min_{\theta} \mathbf{E}\,\rho((f(x) + \varepsilon - \theta)/s)$. Here we used the Gudermannian for ρ, and set $s = \text{med}\,|\epsilon|$ independent of x.

Dashed curves Next we look at statistical estimates of the deterministic parameters just mentioned, based on the sample. A simple fifth-degree polynomial model is assumed, taking the form $\widehat{h}(x) = \sum_{k=1}^{5} w_k x^k$. Running OLS to specify the weights results in the red dashed curve (the graph of $\widehat{h}$). Similarly, the green dashed line is the estimate due to running LAD. Finally, the blue dashed line is the product of running `fRLM` (Algorithm 1) just as specified in section 3.5.3. Each algorithm was run twice: once for the well-behaved data on the left domain, and once for the uncongenial data on the right domain. Finally, we should remark that while

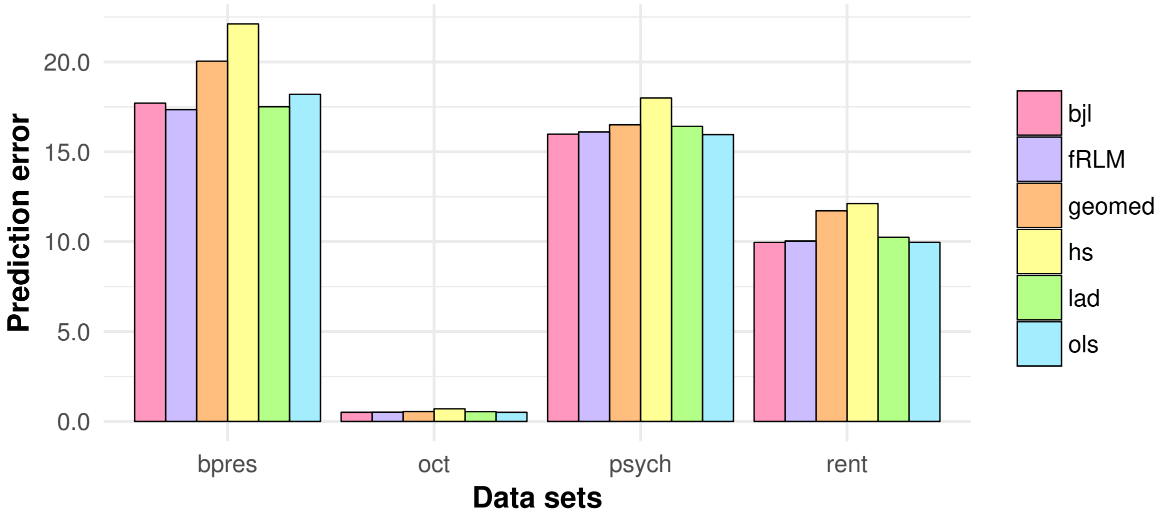

Figure 3.9: Prediction error on four distinct real-world data sets.

OLS and LAD do indeed correspond to trying to learn $\mathbf{E}(y;x)$ and $\text{med}(y;x)$ respectively, the correspondence is not quite so clear for `fRLM`. Here the routine explicitly tries to minimize $\theta^{*}(h)$ from Defn. 10, though the relation between h^{*} satisfying $\theta^{*}(h^{*}) = \theta^{*}(\mathcal{H})$ and the closeness of $h^{*} \approx \widetilde{y}$ remains a matter of both technical and conceptual interest.

3.7 Concluding remarks

In this work, we have introduced and explored a novel approach to the regression problem, using robust loss estimates and an efficient routine for minimizing these estimates without requiring prior knowledge of the underlying distribution. In addition to theoretical analysis of the fundamental properties of the algorithm being used, we showed through comprehensive empirical testing that the proposed technique indeed has extremely desirable robustness properties. In a wide variety of problem settings, our routine was shown to uniformly outperform well-known competitors both classical and modern, with cost requirements that are tolerable, suggesting a strong general approach for regression in the non-parametric setting.

Looking ahead, there are a number of interesting lines of work to be taken up. Extending this work to unsupervised learning problems is an immediate goal. Beyond this, a more careful look at the optimality of different algorithms from a cost/performance standpoint would assuredly be of interest. When is it more profitable (under some metric) to use "balanced" methods such as that of Minsker [36], Brownlees et al. [12], Hsu and Sabato [28] or ours, rather than committing to one of two extremes, say OLS or LAD? The former perform very well, but require extra computation. Characterizing such situations in terms of the underlying data distribution is both technically and conceptually interesting. Clear tradeoffs between formal assurances and extra computational cost could shed new light on precisely where traditional ERM algorithms and close variants fail to be economical.

3.8 Proofs

Proof of Lemma 12. For notational simplicity, given any $h \in \mathcal{H}$, write $x_i = l(h; \boldsymbol{z}_i)$, $i \in [n]$. Taking $u \in [\min_i\{x_i\}, \max_i\{x_i\}]$, clearly the right-hand side of (3.7) is non-empty, i.e., an

M-estimate exists. Since ρ is differentiable and strongly convex on $\mathbb{R}$, the minimum is uniquely determined, characterized by the $\mathbf{E}_{\mu_n} \psi$ condition in the Lemma statement, noting ψ is monotone increasing on its domain, we have that $\widehat{\theta}(h)$ is well-defined.

Regarding $\theta^{*}(h)$, writing $x = l(h;\boldsymbol{z})$, since $|\rho(u)| \leq c|u|$ for some $c > 0$, integrability follows by monotonicity of the Lebesgue integral, that is for any $u \in \mathbb{R}$, we have by $x \in \mathcal{L}_2(\mu)$ that

$$\int \rho\left(\frac{x-u}{s}\right) d\mu \leq \int \frac{c|x-u|}{s} d\mu < \infty.$$

Since $\rho' = \psi$ is bounded, again for any u we have that

$$\frac{d}{du} \mathbf{E}_{\mu} \rho\left(\frac{x-u}{s}\right) = \frac{-1}{s} \mathbf{E}_{\mu} \psi\left(\frac{x-u}{s}\right)$$

holds [3, Ch. 1.6]. Existence of the minimum, given as a root of the right-hand side of this equation, is now immediate. Uniqueness follows from the strong convexity of ρ, noting for any functions u and v of $\boldsymbol{z}$,

$$\mathbf{E}_{\mu} \rho(\alpha u(\boldsymbol{z}) + (1-\alpha)v(\boldsymbol{z})) < \alpha \mathbf{E}_{\mu} \rho(u(\boldsymbol{z})) + (1-\alpha) \mathbf{E}_{\mu} \rho(v(\boldsymbol{z}))$$

for any $\alpha \in (0,1)$. □

Proof of Lemma 13. Fix arbitrary values $l_1,\ldots,l_n \in \mathbb{R}_+$ and $s_1,\ldots,s_n > 0$. To compactly denote these variables, write $\boldsymbol{a} = (l_1,\ldots,l_n,s_1,\ldots,s_n)$. Denote $\mathcal{B}_0 \coloneqq \mathcal{B}(\mathbb{R}^{2n})$ here, and define

$$F(u,\boldsymbol{a}) \coloneqq \sum_{i=1}^{n} \rho\left(\frac{l_i - u}{s_i}\right), \quad f(u,\boldsymbol{a}) \coloneqq \frac{d}{dt} F(t,\boldsymbol{a})|_{t=u}, \quad u \in \mathbb{R}.$$

Let $\widehat{u} \coloneqq \inf \arg\min_u F(u,\boldsymbol{a})$, a map from $\mathbb{R}^{2n}$ to $\mathbb{R}$. If ρ specifies a robust penalty, then from Lemma 12 the minimizer is unique and thus the infimum is superfluous. More generally, even when the minimizer is not unique, the infimum $\widehat{u}$ will be a valid minimizer. To see this, denoting $\rho_0 \coloneqq \min_u F(u,\boldsymbol{a})$, say we have $F(\widehat{u},\boldsymbol{a}) > \rho_0$. By continuity and monotonicity, there exists $u_1 > \widehat{u}$ such that $\rho_0 < F(u_1,\boldsymbol{a}) < F(\widehat{u},\boldsymbol{a})$, and thus u_1 lower bounds the set $\arg\min_u F(u,\boldsymbol{a})$, a contradiction of $\widehat{\theta}(h)$ being the greatest lower bound. Thus $F(\widehat{u},\boldsymbol{a}) = \rho_0$. It follows that $\widehat{u}$ is also root of $f(\cdot,\boldsymbol{a})$.

For arbitrary $\alpha \in \mathbb{R}$, define events

$$\begin{aligned} \boldsymbol{A}_{\alpha} &\coloneqq \{\boldsymbol{a} \in \mathbb{R}^{2n} : \widehat{u} \leq \alpha\} \\ \boldsymbol{A}' &\coloneqq \bigcap_{k=1}^{\infty} \bigcup_{u \in U_{\alpha}} \left\{\boldsymbol{a} \in \mathbb{R}^{2n} : |f(u,\boldsymbol{a})| < \frac{1}{k}\right\}, \quad U_{\alpha} \coloneqq \{q \in \mathbb{Q} : q \leq \alpha\}. \end{aligned}$$

Indexing over the rationals is to make the union countable. First note that as $f(u,\cdot)$ is continuous, it is measurable for every u, and equivalently

$$\{|f(u,\boldsymbol{a})| < 1/k\} \in \mathcal{B}_0, \quad \forall\, u \in \mathbb{R}, k \in \mathbb{N}.$$

As such every set indexed in $\boldsymbol{A}'$ is measurable. As $\boldsymbol{A}'$ is a countable intersection of a countable union of measurable sets, $\boldsymbol{A}'$ itself is measurable. First, say $\boldsymbol{a} \in \boldsymbol{A}'$. On this occasion, for each integer $k > 0$, there exists a rational $u \leq \alpha$ such that the objective $f(\cdot,\boldsymbol{a})$ falls within $\pm\, k^{-1}$ of zero. Now assume $\widehat{u}(\boldsymbol{a}) > \alpha$ for this $\boldsymbol{a}$. By definition $f(\widehat{u}(\boldsymbol{a}),\boldsymbol{a}) = 0$. As f depends monotonically on u, and $\widehat{u}$ is infimal, we have for some $\epsilon > 0$ that

$$\exists\, u_1 \in (\alpha, \widehat{u}(\boldsymbol{a})) \cap \mathbb{Q}, \quad f(u_1,\boldsymbol{a}) \geq \epsilon.$$

Taking $k \in \mathbb{N}$ large enough (so that $1/k < \epsilon$), we can necessarily secure a rational $q \le \alpha$ such that $|f(q, \boldsymbol{a})| < \epsilon$. However as $q < u_1$, this means that

$$f(q, \boldsymbol{a}) \ge f(u_1, \boldsymbol{a}) \ge \epsilon > 0,$$

which is a contradiction. Thus $\widehat{u}(\boldsymbol{a}) \le \alpha$. The $\boldsymbol{a}$ choice was arbitrary, so $\boldsymbol{A}' \subseteq \boldsymbol{A}_\alpha$.

The converse is even simpler. Let $\boldsymbol{a} \in \boldsymbol{A}_\alpha$. We can always take a sequence (q_m) of $q_m \in \mathbb{Q}$ where $q_m \uparrow \widehat{u}(\boldsymbol{a})$. For any $k \in \mathbb{N}$, there exists $m_0 < \infty$ where

$$m \ge m_0 \implies f(q_m, w, \boldsymbol{z}) - f(\widehat{u}(\boldsymbol{a}), \boldsymbol{a}) < 1/k$$

which in turn implies $|f(q_m, \boldsymbol{a})| < 1/k$, that is $\boldsymbol{a} \in \boldsymbol{A}'$. We have $\boldsymbol{A}_\alpha \subseteq \boldsymbol{A}'$ and thus $\boldsymbol{A}_\alpha = \boldsymbol{A}'$, concluding that $\boldsymbol{A}_\alpha \in \mathcal{B}_0$ for any choice of α, and any $w \in \mathcal{W}$. Note $\boldsymbol{A}_\alpha$ is just $\widehat{u}^{-1}(-\infty, \alpha]$, the inverse image of this segment induced by $\widehat{u}$. Denoting these intervals as $\mathcal{D} = \{(-\infty, \alpha] : \alpha \in \mathbb{R}\}$, the σ-field generated by this class is $\sigma(\mathcal{D}) = \mathcal{B}(\mathbb{R})$, and the class $\mathcal{D}' = \{B \in \mathcal{B} : \widehat{u}^{-1}(B) \in \mathcal{B}_0\}$ is a σ-field [9, Ch. 2.7]. We proved above that $\mathcal{D} \subseteq \mathcal{D}'$, and by minimality of the generated field, $\mathcal{D}' = \mathcal{B}^1$. We conclude $\widehat{u}^{-1}(B) \in \mathcal{B}_0$ for all $B \in \mathcal{B}(\mathbb{R})$. With this, and the measurability of $l(\cdot;\cdot)$ and s_h, the Lemma follows; the specific requirement is $\mathcal{B}(\mathcal{H}) \times \mathcal{B}_{d+1}$ measurability of l and either $\mathcal{B}(\mathcal{H}) \times \mathcal{B}_{d+1}^n$ or $\mathcal{B}(\mathcal{H}) \times \mathcal{B}_{d+1}$ measurability of s_h, depending on whether it is determined by μ_n or individual instances. □

Proof of Theorem 14. Use $\widehat{\theta}(h)$ as in the statement of Lemma 13. Fix an arbitrary set of instances $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n) \in \mathcal{Z}$, and

$$\widehat{\theta}(\mathcal{H}) := \inf\left\{\widehat{\theta}(h) : h \in \mathcal{H}\right\}$$

$$f(u, h; \boldsymbol{Z}) := \sum_{i=1}^{n} \psi\left(\frac{l(h; \boldsymbol{z}_i) - u}{s_h(\boldsymbol{z}_i)}\right), \quad h \in \mathcal{H}, u \in \mathbb{R}.$$

Construct a sequence (θ_m) of $\theta_m \in \{\widehat{\theta}(h) : h \in \mathcal{H}\}$ such that $\theta_m \downarrow \widehat{\theta}(\mathcal{H})$. To each θ_m, there is an accompanying $h_m \in \mathcal{H}$ such that $f(\theta_m, h_m, \boldsymbol{Z}) = 0$. As $\sup_m \|h_m\| < \infty$, there exists a convergent subsequence (h_k). Denote $\widehat{h} := \lim_{k\to\infty} h_k$. Subsequence θ_k converges to $\widehat{\theta}(\mathcal{H})$. Continuity of L and s implies $f(\cdot, \cdot, \boldsymbol{Z})$ is continuous, and thus

$$f(\widehat{\theta}(\mathcal{H}), \widehat{h}, \boldsymbol{Z}) = \lim_{k\to\infty} f(\theta_k, h_k, \boldsymbol{Z}) = 0,$$

which by uniqueness of the root of $f(\cdot, \widehat{h}, \boldsymbol{Z})$ (Lemma 12) implies that

$$\forall\, \boldsymbol{Z}, \exists\, \widehat{h} \in \mathcal{H}, \widehat{\theta}(\widehat{h}) = \widehat{\theta}(\mathcal{H}). \tag{3.16}$$

That is, for any set of observations $\boldsymbol{Z}$, we can find such an $\widehat{h}$ minimizing the new objective function.

From this point, measurability is a purely technical endeavour. Useful references are Dudley [19, Ch. 5], Pollard [38, Appendix C], and Dellacherie and Meyer [16, Ch. 1–3]. We assume $\mathcal{H}$ ia separable; the special case of $\mathcal{H} \subset \mathbb{R}^d$ is an archetypal example. Index and assemble all possible (random) values of our objective in $\boldsymbol{\Theta} := \{\widehat{\theta}(h) : h \in \mathcal{H}\}$, with $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ left free to vary randomly. As $\widehat{\theta}(h)$ has been shown to be $\mathcal{H} \times \mathcal{Z}$-measurable (Lemma 13), under an innocuous regularity condition, [38, Appendix C, 1(ii)], the class $\boldsymbol{\Theta}$ is sufficiently regular, called "permissible." It is readily verified that $\widehat{\theta}(\mathcal{H})$ is $\mathcal{B}(\mathcal{Z})$-measurable. Next, define the set

$$\begin{aligned}\boldsymbol{A}_3 &:= \{(\boldsymbol{Z}, h) : \widehat{\theta}(h) = \widehat{\theta}(\mathcal{H})\} \\ &= \widetilde{\theta}^{-1}(-\infty, 0] \cap \widetilde{\theta}^{-1}(-\infty, 0)^c\end{aligned}$$

where we have written $\widetilde{\theta}(\boldsymbol{Z}, h) := (\widehat{\theta}(h) - \widehat{\theta}(\mathcal{H}))$. We have already verified the measurability of the two terms being subtracted, thus $\widetilde{\theta}$ is $\mathcal{B}(\mathcal{H}) \times \mathcal{B}(\mathcal{Z})$ measurable. Looking at the second equality, we have that $\boldsymbol{A}_3$ is an analytic subset of $\mathcal{Z} \times \mathcal{H}$. Taking the projection π of $\boldsymbol{A}_3$ onto the observation space, namely

$$\pi(\boldsymbol{A}_3) := \{\boldsymbol{Z} : (\boldsymbol{Z}, h) \in \boldsymbol{A}_3, h \in \mathcal{H}\},$$

and note that by our existence result (3.16), $\mathbf{P}\,\pi(\boldsymbol{A}_3) = 1$. From Pollard [38, Appendix C(d)], it follows that there exists a random variable $\widehat{h}(\boldsymbol{Z})$ such that $(\boldsymbol{Z}, \widehat{h}(\boldsymbol{Z})) \in \boldsymbol{A}_3$ for almost all $\boldsymbol{Z} \in \pi(\boldsymbol{A}_3)$. Since the latter set has $\mathbf{P}$-measure 1, we conclude that this $\widehat{h}$ realizes the properties sought in the statement of Theorem 14, concluding the argument. □

Proof of Proposition 16. Consider any sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. Write $\gamma(h) = \gamma_{\mu_n}(h)$ for simplicity. Fix any $\varepsilon > 0$. By continuity of L, exists $\delta > 0$ where $\|h - h'\| \leq \delta$ implies

$$\max \left\{|l(h; \boldsymbol{z}_i) - \gamma(h) - l(h'; \boldsymbol{z}_i) + \gamma(h')|\right\}_{i=1}^{n} \leq \varepsilon.$$

Denote $s := s_h$ and $s' := s_{h'}$. Now assume $|s - s'| > \varepsilon$, say for concreteness that $s + \varepsilon < \widetilde{s} < s'$. This implies that for any $\widetilde{s}$ taken such that $s + \varepsilon < \widetilde{s} < s'$, we have

$$\frac{l(h'; \boldsymbol{z}_i) - \gamma(h')}{s'} < \frac{l(h'; \boldsymbol{z}_i) - \gamma(h')}{\widetilde{s}} < \frac{l(h; \boldsymbol{z}_i) - \gamma(h)}{s}, \quad i = 1, \ldots, n$$

and by the weak monotonicity of χ, and the definitions of the two roots s and s',

$$\begin{aligned} 0 = \sum_{i=1}^{n} \chi\left(\frac{l(h'; \boldsymbol{z}_i) - \gamma(h')}{s'}\right) &\leq \sum_{i=1}^{n} \chi\left(\frac{l(h'; \boldsymbol{z}_i) - \gamma(h')}{\widetilde{s}}\right) \\ &\leq \sum_{i=1}^{n} \chi\left(\frac{l(h; \boldsymbol{z}_i) - \gamma(h)}{s}\right) \\ &= 0, \end{aligned}$$

and thus the middle sum is in fact zero. This implies

$$\widetilde{s} \in \left\{s > 0 : \sum_{i=1}^{n} \chi((l(h'; \boldsymbol{z}_i) - \gamma(h'))/s) = 0\right\},$$

but since $\widetilde{s} < s'$, this is a contradiction of s' as the infimum of this set. An identical argument holds for the other case of $s' + \varepsilon < s$, and so $|s - s'| \leq \varepsilon$. We conclude for any $\varepsilon > 0$, there exists $\delta > 0$ such that $\|h - h'\| \leq \delta$ implies $|s_h - s_{h'}| \leq \varepsilon$. □

Proof of Theorem 17. The techniques are standard; see Fan et al. [20, 21] for example. First, note that

$$|\theta^* - \mathbf{E}_\nu\, x|^2 = \mathbf{E}_\nu((x - \theta^*)^2 - (x - \mathbf{E}_\nu\, x)^2).$$

Focusing on the right-hand side, let $f(u) := u^2$ and $f_s(u) := 2s^2\rho_s(u)$, where the rescaling is purely for technical reasons, noting $\theta^* = \arg\min_u \mathbf{E}_\nu\, f_s(x - u) = \arg\min_u \mathbf{E}_\nu\, \rho_s(x - u)$. The difference between f and f_s should grow small as $s \to \infty$. Write this difference as $d_s(u) := f(u) - f_s(u)$. With the optimality of θ^*, it follows that in our new notation

$$\mathbf{E}_\nu(f(x - \theta^*) - f(x - \mathbf{E}_\nu\, x)) \leq \mathbf{E}_\nu(d_s(x - \theta^*) - d_s(x - \mathbf{E}_\nu\, x)). \tag{3.17}$$

Fixing arbitrary x for now, by a first-order exact Taylor expansion [41, Chapter 5], one has that

$$d_s(x-\theta^*) = d_s(x - \mathbf{E}_\nu\, x) + d_s'(x-\widetilde{\theta})(\mathbf{E}_\nu\, x - \theta^*), \quad \widetilde{\theta} = t\theta^* + (1-t)\,\mathbf{E}_\nu\, x \tag{3.18}$$

for some $t \in [0,1]$. This t, and thus the value of $\widetilde{\theta}$, may depend on x and thus in taking expectations we will need some general results. Plugging (3.18) into (3.17), we see that

$$|\theta^* - \mathbf{E}_\nu\, x|^2 \leq \mathbf{E}_\nu\, d_s'(x-\widetilde{\theta})(\mathbf{E}_\nu\, x - \theta^*),$$

implying $|\theta^* - \mathbf{E}_\nu\, x| \leq \mathbf{E}_\nu\, d_s'(x-\widetilde{\theta})$.

To deal with this term, looking at higher order terms is fruitful due to the nice behaviour of ρ and its derivatives. For clean notation, let $g(u) := d_s'(u)$. Applying the Taylor expansion with integral remainder [5, Section 23], we have

$$g(b) = g(a) + g'(b-a) + \int_a^b \frac{(u-b)^2}{2} g^{(3)}(u)\,du.$$

Checking the terms, we have

$$\begin{aligned} g(u) &= 2u - 2s\psi_s(u) \\ g'(u) &= 2 - 2\eta_s(u) \\ g^{(3)}(u) &= -\frac{2}{s^2}\rho_s^{(4)}(u) \end{aligned}$$

and thus setting $a = 0$ in the above expansion, all terms but the right-most summand will clear, leaving us with

$$d_s'(x) = -\frac{1}{s^2}\int_0^x (u-x)^2 \rho_s^{(4)}(u)\,du$$

regardless of the value of x. Note that writing $M := \|\rho^{(4)}\|_\infty$, we have

$$\begin{aligned} |d_s'(x)| &\leq \frac{M}{s^2}\int_0^x (u-x)^2\,du \\ &= \frac{M}{3s^2}(x)^3. \end{aligned}$$

Setting this x value to our $x - \widetilde{\theta}$, it follows that for $c = M/3$ we have

$$d_s'(x-\widetilde{\theta}) \leq cs^{-2}|x-\widetilde{\theta}|^3$$

where x is arbitrarily selected. Note that

$$\mathbf{E}_\nu\, |x-\widetilde{\theta}|^3 \leq \mathbf{E}_\nu\, |x-\theta^*|^3 \vee \mathbf{E}_\nu\, |x - \mathbf{E}_\nu\, x|^3 < \infty,$$

and as s tends to infinity, since the two terms being maximized over will encroach upon each other, and using our finite third moment assumption, we can always find a value to bound this from above. This implies the result. □

Proof of Theorem 19. For $h \in \mathcal{H}$, write $x = l(h;\boldsymbol{z})$ and $\widehat{\theta} = \widehat{\theta}(h)$, $\theta^* = \theta^*(h)$ for simplicity. Let s be either a fixed positive constant, or be generated on a per-observation basis, i.e., $s_1, \ldots, s_n$ are independent positive random variables, where say $s = s(\boldsymbol{z})$ for $\boldsymbol{z} \sim \mu$. The existence of $\widehat{\theta}$ and θ^* is given by Lemma 12. For convenience denote $\psi_u := \psi((x-u)/s)$ and note that

$$\{\widehat{\theta} < u\} = \{\mathbf{E}_{\mu_n} \psi_u < 0\}, \quad \{\widehat{\theta} > u\} = \{\mathbf{E}_{\mu_n} \psi_u > 0\} \tag{3.19}$$

for any choice of u. Use the typical set lim inf definition, which is to say for any given sequence of sets A_m, let $\liminf_m A_m := \bigcup_{m=1}^{\infty} \bigcap_{k \geq m} A_k$. For arbitrary fixed $\varepsilon > 0$, we have

$$\begin{aligned}
\mathbf{P}\left\{\lim_n \widehat{\theta} < \theta^* + \varepsilon\right\} &= \mathbf{P} \liminf_n \{\widehat{\theta}_n < \theta^* + \varepsilon\} \\
&= \mathbf{P} \liminf_n \{\mathbf{E}_{\mu_n} \psi_{\theta^*+\varepsilon} < 0\} \\
&= \mathbf{P}\left\{\lim_n \mathbf{E}_{\mu_n} \psi_{\theta^*+\varepsilon} < 0\right\} \\
&\geq \mathbf{P}\left\{\lim_n \mathbf{E}_{\mu_n} \psi_{\theta^*+\varepsilon} = \mathbf{E}_{\mu} \psi_{\theta^*+\varepsilon}\right\} \\
&= 1.
\end{aligned}$$

The final equality holds via the strong law of large numbers, which is where we require $\mathbf{E}_{\mu} x^2 < \infty$ [9, Theorem 3.27]. The inequality prior to that holds since $\mathbf{E}_{\mu} \psi_{\theta^*+\varepsilon} < 0$, and the remaining equalities by lim inf definition and (3.19). An indentical argument can be used to show $\mathbf{P}\{\lim_n \widehat{\theta} > \theta^* - \varepsilon\} = 1$, which implies

$$\begin{aligned}
\mathbf{P}\left\{\lim_n |\widehat{\theta} - \theta^*| \geq \varepsilon\right\} &\leq \mathbf{P}\left\{\lim_n \widehat{\theta}_n \geq \theta^* + \varepsilon\right\} \cup \left\{\lim_n \widehat{\theta} \leq \theta^* - \varepsilon\right\} \\
&= 0.
\end{aligned}$$

This holds for any choice of $\varepsilon > 0$, and thus $|\widehat{\theta} - \theta^*| \to 0$ almost surely, yielding strong consistency. □

Proof of Lemma 20. If ρ specifies a robust objective, then ψ is a bounded measurable function, and can be uniformly approximated by a sequence of weighted indicators as follows. For concreteness, say $|\psi| \leq M < \infty$. Let sequence $\varepsilon_m \downarrow 0$, and for each $m \in \mathbb{N}$ partition the range $[-M, M]$ into $k_m := 2M/\varepsilon_m$ segments $A_j := \{t : a_{j-1} \leq \psi(t) < a_j\}$ defined by

$$a_0 = -M, \quad a_j = a_{j-1} + \varepsilon_m, \quad j = 1, \ldots, k_m.$$

The approximating function s_m is then defined as

$$s_m(u) := \sum_{j=1}^{k_m} a_j I_{A_j}(u), \quad u \in \mathbb{R}.$$

By strong convexity, there is no $u \in \mathbb{R}$ where $|\psi(u)| = M$, and thus the uniform approximation is immediate. That is, $|s_m(u) - \psi(u)| \leq \varepsilon_m$ holds uniformly in u. Note that each A_j can be given as an interval. Defining b_j to be the unique element in $\overline{\mathbb{R}}$ where $\psi(b_j) = a_j$, the marginal sets are $A_1 = (-\infty, b_1)$ and $A_{k_m} = [b_{k_m - 1}, \infty)$ respectively, and the remainder are half-closed real intervals $A_j = [b_{j-1}, b_j)$.

Denote $\mathbf{P}_n = \mathbf{E}_{\mu_n}$ and $\mathbf{E} = \mathbf{E}_{\mu}$ for clean notation. Our interest is with the quantity

$$\|\mathbf{P}_n \psi - \mathbf{E}\psi\| := \sup_{u,h,s} \left| \mathbf{P}_n \psi\left(\frac{l(h;\boldsymbol{z}) - u}{s}\right) - \mathbf{E}\psi\left(\frac{l(h;\boldsymbol{z}) - u}{s}\right) \right|$$

where $s > 0$, $h \in \mathcal{H}$, and $u \in \mathbb{R}$ when taking the supremum. For any observation $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ an application of the triangle inequality yields

$$\|\mathbf{P}_n \psi - \mathbf{E}\,\psi\| \leq \|\mathbf{P}_n \psi - \mathbf{P}_n s_m\| + \|\mathbf{P}_n s_m - \mathbf{E}\,s_m\| + \|\mathbf{E}\,s_m - \mathbf{E}\,\psi\| \tag{3.20}$$

where the $\|\cdot\|$ terms on the right-hand side denote taking the exact same suprema as on the left-hand side. The first and third terms are readily dealt with. Note for example that

$$\|\,\mathbf{E}(s_m - \psi)\| \leq \|s_m - \psi\|_\infty \leq \varepsilon_m \to 0$$

whenever we set index $m = m(n) \to \infty$ as $n \to \infty$. An identical argument holds for the first term. This convergence is deterministic, in the sense that it holds for arbitrary observations, and thus also holds almost surely.

The second term in (3.20) is slightly more involved but the approach is rather standard. To get started, denoting for convenience the events

$$E_j := \{l(h; \boldsymbol{z}) \in [sb_{j-1} + u, sb_j + u)\}, \quad j = 1, \ldots, k_m$$

with the understanding that for the index $j = 1$ the interval is $(-\infty, sb_1 + u)$ and $j = k_m$ it is $[sb_{k_m-1} + u, \infty)$. The obvious but important fact is that each event E_j, specified by s, h, u, and the b_j values, is naturally captured by a larger class of sets $\mathcal{C}$

$$\mathcal{C} := \left\{\{\boldsymbol{z} : l(h; \boldsymbol{z}) \in [a, b)\} : h \in \mathbb{R}^d, a, b \in \overline{\mathbb{R}}, a < b\right\}.$$

Note we are assuming $\mathcal{H}$ is specified by elements of d-dimensional Euclidean space. Since each $E_j \in \mathcal{C}$, we have that

$$\begin{aligned} \|\mathbf{P}_n s_m - \mathbf{E}\,s_m\| &= \sup \left| \sum_{j=1}^{k_m} a_j \left(\mathbf{P}_n I_{E_j}(\boldsymbol{z}) - \mathbf{P}\,E_j\right)\right| \\ &\leq M k_m \|\,\mathbf{P}_n I_C - \mathbf{P}\,C\|_{\mathcal{C}} \end{aligned} \tag{3.21}$$

where $\|\cdot\|_{\mathcal{C}}$ denotes taking the supremum over $C \in \mathcal{C}$. We will frequently use I_C to denote $I_C(\cdot)$, with domain $\mathbb{R}^{d+1}$. It remains to show the strong convergence to zero of the supremal factor in (3.21), with convergence rates to deal with the increasing k_m sequence.

A typical symmetrization inequality is of use next [48, Lemma 2]. Take an artificial sample $\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_n$, independent from $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, but identically distributed. For any $\varepsilon > 0$, whenever $n > 2/\varepsilon^2$, we have

$$\mathbf{P}\left\{\|\,\mathbf{P}_n I_C - \mathbf{P}\,C\|_{\mathcal{C}} > \varepsilon\right\} \leq 2\,\mathbf{P}\left\{\|\,\mathbf{P}_n I_C - \mathbf{P}'_n I_C\|_{\mathcal{C}} \geq \varepsilon/2\right\} \tag{3.22}$$

where $\mathbf{P}'_n$ analogously denotes μ'_n supported on the new sample. Next a randomization technique due to Pollard [37]. Let $\sigma_1, \ldots, \sigma_n$ be iid, and independent from both samples, with distribution $\mathbf{P}\{\sigma = -1\} = \mathbf{P}\{\sigma = 1\} = 1/2$. Checking cases one immediately confirms that for any $C \in \mathcal{C}$, the random quantities $I_C(z) - I_C(z')$ and $\sigma(I_C(z) - I_C(z'))$ have the same distribution. As such for any $\varepsilon > 0$,

$$\begin{aligned} \mathbf{P}\left\{\|\,\mathbf{P}_n I_C - \mathbf{P}'_n I_C\|_{\mathcal{C}} \geq \varepsilon\right\} &= \mathbf{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n} \sigma_i(I_C(z_i) - I_C(z'_i))\right\|_{\mathcal{C}} \geq \varepsilon\right\} \\ &\leq \mathbf{P}\left\{\|\,\mathbf{P}_n\,\sigma I_C\|_{\mathcal{C}} + \|\,\mathbf{P}'_n\,\sigma I_C\|_{\mathcal{C}} \geq \varepsilon\right\} \\ &\leq 2\,\mathbf{P}\left\{\|\,\mathbf{P}_n\,\sigma I_C\|_{\mathcal{C}} \geq \varepsilon/2\right\} \end{aligned}$$

where for the first inequality one leverages the triangle inequality, and for the second a union bound. We can conclude up to this point for large enough n that

$$\mathbf{P}\left\{\| \mathbf{P}_n I_C - \mathbf{P}\, C\|_{\mathcal{C}} > \varepsilon\right\} \leq 4\, \mathbf{P}\left\{\| \mathbf{P}_n\, \sigma I_C\|_{\mathcal{C}} \geq \varepsilon/4\right\}.$$

Fixing arbitrary sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, a combinatorial indicator of the complexity $\mathcal{C}$ is given by

$$\begin{aligned}\Delta_n(\mathcal{C}) &:= |\{C \cap \{z_1, \ldots, z_n\} : C \in \mathcal{C}\}| \\ &= |\{(I_C(z_1), \ldots, I_C(z_n)) \in \{0,1\}^n : C \in \mathcal{C}\}|.\end{aligned}$$

Naturally the number of distinct subsets captured by members of $\mathcal{C}$ is identical to the number of distinct n-length binary-valued vectors than can be built on the sample when indexing over $\mathcal{C}$. Trivially $\Delta_n(\mathcal{C}) \leq 2^n$. Again conditioning on a fixed sample, we can always take $C_1, \ldots, C_k \in \mathcal{C}$ such that all possible realizations of $\mathbf{P}_n\, \sigma I_C$ are captured by indexing over these $k = \Delta_n(\mathcal{C})$ sets. That is, denoting $\boldsymbol{Z} := (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$,

$$\begin{aligned}\mathbf{P}\left\{\| \mathbf{P}_n\, \sigma I_C\|_{\mathcal{C}} \geq \varepsilon; \boldsymbol{Z}\right\} &= \mathbf{P}\left\{\max_{1 \leq j \leq k} \left|\mathbf{P}_n\, \sigma I_{C_j}\right| \geq \varepsilon; \boldsymbol{Z}\right\} \\ &\leq \mathbf{P} \bigcup_{j=1}^{k} \left\{\left|\mathbf{P}_n\, \sigma I_{C_j}\right| \geq \varepsilon; \boldsymbol{Z}\right\} \\ &\leq \Delta_n(\mathcal{C}) \max_{1 \leq j \leq k} \mathbf{P}\left\{\left|\mathbf{P}_n\, \sigma I_{C_j}\right| \geq \varepsilon; \boldsymbol{Z}\right\}.\end{aligned}$$

The two multiplicands need to be controlled. Let us start with the former. When we do not fix $\boldsymbol{Z}$, naturally $\Delta_n(\mathcal{C})$ is a random quantity. Note that the possible forms any $C \in \mathcal{C}$ can take are characterized into three types as

$$\{\boldsymbol{z} : l \in [a, b)\}, \quad \{\boldsymbol{z} : l \in [a, \infty)\}, \quad \{\boldsymbol{z} : l \in (-\infty, b)\},$$

also $\{l \in (-\infty, \infty)\} = \mathbb{R}^{d+1}$, and setting $b \leq 0$ returns the empty set since $l \geq 0$. We have denoted $l(h; \boldsymbol{z})$ by l for simplicity. For concreteness consider $l(h; \boldsymbol{z}) = (y - h(\boldsymbol{x}))^2$ case, though the exact same argument clearly holds for other related losses. Take any $a, b \in \mathbb{R}$ where $a < b$. Then setting

$$\begin{aligned}G_1 &:= \left\{y - \boldsymbol{w}^T \boldsymbol{x} \geq \sqrt{|a|}\right\}, \quad G_2 := \left\{y - \boldsymbol{w}^T \boldsymbol{x} \leq -\sqrt{|a|}\right\} \\ G_1' &:= \left\{y - \boldsymbol{w}^T \boldsymbol{x} < \sqrt{|b|}\right\}, \quad G_2' := \left\{y - \boldsymbol{w}^T \boldsymbol{x} > -\sqrt{|b|}\right\}\end{aligned}$$

and recalling under the linear model assumption on $\mathcal{H}$, for any $h \in \mathcal{H}$ we have $h(\boldsymbol{x}) = w^T \boldsymbol{x}$ for some $\boldsymbol{w} \in \mathbb{R}^d$, thus clearly we have

$$\{l \in [a, b)\} = (G_1 \cup G_2) \cap G_1' \cap G_2'.$$

If one defines $g(z) := (y - \boldsymbol{w}^T \boldsymbol{x} - \sqrt{|a|})(-1)$, then $G_1 = \{g(\boldsymbol{z}) \leq 0\}$. Setting $g'(\boldsymbol{z}) := (y - \boldsymbol{w}^T \boldsymbol{x} - \sqrt{|b|})(-1)$, have $G_1' = \{g'(\boldsymbol{z}) \leq 0\}^c$ where the superscript denotes the complement. If our observations are $d+1$ dimension vectors of the form $\boldsymbol{z} = (x_1, \ldots, x_d, y)$, define functions $f_0(\boldsymbol{z}) := 1$ and $f_j(\boldsymbol{z}) := \pi_j(\boldsymbol{z})$ for $j = 1, \ldots, d+1$, where π_j denotes the jth coordinate projection. That is, e.g., $f_1(\boldsymbol{z}) = x_1$ and so forth. Construct a linear space of functions on $\mathbb{R}^{d+1}$ as

$$\mathcal{F} := \operatorname{span}\{f_0, \ldots, f_{d+1}\}.$$

One may check the linear independence of these functions, and thus the dimension of is precisely $\dim \mathcal{F} = d+2$. Note clearly that $g, g' \in \mathcal{F}$. From this one naturally induces two classes of sets, namely

$$\mathcal{G} := \{\{f(\boldsymbol{z}) \leq 0\} : f \in \mathcal{F}\}, \quad \mathcal{G}^c := \{G^c : G \in \mathcal{G}\}.$$

A classic result [45, 18] says that, using more modern parlance the class $\mathcal{G}$ has a VC dimension bounded by $\dim \mathcal{F}$. The fundamental property of classes with finite VC dimension is that the supremum of Δ_n taken over all samples is bounded by a polynomial in n. More precisely, for some constant c_0, for all n we have

$$\mathbf{E}\,\Delta_n(\mathcal{G}) \leq s_n(\mathcal{G}) := \sup_{\boldsymbol{Z}} \Delta_n(\mathcal{G}) \leq c_0 n^{d+2},$$

where the expectation is being taken over the sample $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$. It is then clear that

$$\{\boldsymbol{z} : l \in [a,b)\} \in (\mathcal{G} \cup \mathcal{G}) \cap \mathcal{G}^c \cap \mathcal{G}^c.$$

For all the other forms the sets $C \in \mathcal{C}$ take, it is clear that each is composed of sets from $\mathcal{G}$, $\mathcal{G}^c$, or $\{\mathbb{R}^{d+1}, \emptyset\}$. The zero function $g_0(\boldsymbol{z}) = 0$, is $g_0 \in \mathcal{F}$, and as such $\mathbb{R}^{d+1} = \{g_0(\boldsymbol{z}) \leq 0\} \in \mathcal{G}$. Also the basis function f_0 used in defining $\mathcal{F}$ is such that $\emptyset = \{f_0(\boldsymbol{z}) \leq 0\} \in \mathcal{G}$. It thus follows that $\emptyset, \mathbb{R}^{d+1} \in \mathcal{G}^c$ as well. We thus conclude

$$\mathcal{C} \subseteq \mathcal{G}^* := (\mathcal{G} \cup \mathcal{G}) \cap \mathcal{G}^c \cap \mathcal{G}^c.$$

Basic combinatorial arguments [38, Lemma 15] show that for a constant $c_1 > 0$ we have

$$s_n(\mathcal{G}^*) \leq s_n(\mathcal{G})^2 s_n(\mathcal{G}^c)^2 \leq c_1 n^{4d+8}$$

which implies $\mathbf{E}\,\Delta_n(\mathcal{C}) \leq c_1 n^{4d+8}$. This is the desired bound for the combinatorial parameter. As for the conditional probability term, note that with fixed $\boldsymbol{Z}$ and the σ_i left random, taking expectation with respect to σ we have for any $C \in \mathcal{C}$ that

$$\mathbf{E}\,\mathbf{P}_n\,\sigma I_C(\boldsymbol{z}) = (\mathbf{E}\,\sigma)\,\mathbf{P}_n\,I_C(\boldsymbol{z}) = 0,$$

and so $\mathbf{P}_n\,\sigma I_C(\boldsymbol{z})$ is a zero-mean sum of random variables taking values on $[-1/n, 1/n]$. Direct application of Hoeffding's inequality yields, with an application of the union bound to get two-sided inequalities,

$$\mathbf{P}\left\{|\mathbf{P}_n\,\sigma I_C| \geq \varepsilon | \boldsymbol{z}_{(n)}\right\} \leq 2\exp\left(\frac{-n\varepsilon^2}{2}\right)$$

for all n. Since the exact same bound holds regardless of $\boldsymbol{z}_{(n)}$ and choice of C, we connect things by integrating, noting for large enough n and constant $c_2 > 0$ we have

$$\begin{aligned}
\mathbf{P}\{\|\,\mathbf{P}_n\,I_C - \mathbf{P}\,C\|_{\mathcal{C}} > \varepsilon\} &\leq 4\,\mathbf{P}\{\|\,\mathbf{P}_n\,\sigma I_C\|_{\mathcal{C}} \geq \varepsilon/4\} \\
&= 4\,\mathbf{E}\left(\Delta_n(\mathcal{C}) \max_{1\leq j\leq k} \mathbf{P}\left\{\left|\mathbf{P}_n\,\sigma I_{C_j}\right| \geq \varepsilon; \boldsymbol{Z}\right\}\right) \\
&\leq c_2 n^{4d+8} \exp\left(\frac{-n\varepsilon^2}{32}\right).
\end{aligned}$$

Application of the root test immediately shows that summing the right-hand side of the final inequality over n, the series converges and thus

$$\sum_{n=1}^{\infty} \mathbf{P} \left\{ \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} > \varepsilon \right\} < \infty.$$

The Borel-Cantelli lemma then says that for any $\varepsilon > 0$,

$$\mathbf{P} \limsup_n \left\{ \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} > \varepsilon \right\} = 0$$

and since

$$\left\{ \lim_{n \to \infty} \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} = 0 \right\}^c = \bigcup_{k=1}^{\infty} \limsup_n \left\{ \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} > 1/k \right\},$$

using a union bound we have

$$\mathbf{P} \left\{ \lim_{n \to \infty} \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} = 0 \right\} \geq 1 - \sum_{k=1}^{k} \mathbf{P} \limsup_n \left\{ \| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} > 1/k \right\} = 1$$

which means $\| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} \to 0$ almost surely.

Returning to sequence k_m from (3.21), while we are free to make this grow as slow as we like, a convergence rate for the term converging to zero makes the argument more transparent. This is done applying Theorems 37 and the Approximation Lemma of Pollard [38, Ch. 2], using the fact that $\mathcal{C}$ has polynomial discrimination, which is precisely what was proved above. In particular, setting $k_{m(n)} = O(n^{1/3})$ is sufficient to imply $\| \, \mathbf{P}_n \, I_C - \mathbf{P} \, C \|_{\mathcal{C}} = o(k_{m(n)}^{-1})$ almost surely. Thus via (3.21) we have that $\| \mathbf{P}_n \, s_m - \mathbf{E} \, s_m \| \to 0$ almost surely, implying the desired result via (3.20). □

Proof of Theorem 22. We start by controlling the random sequence $\widehat{\theta}(h)$ from above. Fix any $h \in \mathcal{H}$. By the usual strong law of large numbers, for any fixed $\delta > 0$, the event

$$\boldsymbol{A} := \left\{ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \psi \left(\frac{l(h; \boldsymbol{z}_i) - (\theta^*(h) + \delta)}{s_h(\boldsymbol{z}_i)} \right) = \mathbf{E}_\mu \, \psi \left(\frac{l(h; \boldsymbol{z}) - (\theta^*(h) + \delta)}{s_h(\boldsymbol{z})} \right) \right\}$$

has $\mathbf{P}(\boldsymbol{A}) = 1$. Given arbitrary $\omega \in \boldsymbol{A}$, and any $\varepsilon > 0$, we can choose $N(\omega) < \infty$ where $n \geq N(\omega)$ implies $| \, \mathbf{E}_{\mu_n} \psi - \mathbf{E}_\mu \, \psi| \leq \varepsilon$, where this notation represents the absolute difference between the sample mean (pre-limit) and expectation taken in the definition of $\boldsymbol{A}$. By definition of $\theta^*(\cdot)$ and monotonicity of ψ, one has that

$$0 = \mathbf{E}_\mu \, \psi \left(\frac{l(h; \boldsymbol{z}) - \theta^*(h)}{s_h(\boldsymbol{z})} \right) > \mathbf{E}_\mu \, \psi \left(\frac{l(h; \boldsymbol{z}) - (\theta^*(h) + \delta)}{s_h(\boldsymbol{z})} \right).$$

It follows that there exists $\varepsilon' > 0$ such that

$$\frac{1}{n} \sum_{i=1}^{n} \psi \left(\frac{l(h; \boldsymbol{z}_i) - (\theta^*(h) + \delta)}{s_h(\boldsymbol{z}_i)} \right) \leq (-1)\varepsilon' < 0$$

eventually (in $n \in \mathbb{N}$), on this $\omega \in \boldsymbol{A}$, and similarly by the definition of $\widehat{\theta}(\cdot)$, we have

$$\widehat{\theta}(\mathcal{H}) \leq \widehat{\theta}(h) < \theta^* + \delta.$$

This shows us that $\boldsymbol{A} \subseteq \{\limsup_n \widehat{\theta}(\mathcal{H}) < \theta^*(h) + \delta\}$. Letting $\delta = 1/k$, for each $k = 1, 2, \ldots$ denote the corresponding convergence event $\boldsymbol{A}$ particularly as $\boldsymbol{A}_k$. Noting $\boldsymbol{A}_{k+1} \subseteq \boldsymbol{A}_k$, we have

$$\boldsymbol{A}_m \subseteq \bigcap_{k=1}^{m} \boldsymbol{A}_k, \quad m = 1, 2, \ldots$$

Basic continuity of measures gives us that

$$\mathbf{P}\left\{\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(h)\right\} = \lim_{m\to\infty} \mathbf{P} \bigcap_{k=1}^{m} \boldsymbol{A}_k = 1,$$

and the same result holds for arbitrary choice of $h \in \mathcal{H}$. Similarly, construct a sequence (h_m) of $h_m \in \mathcal{H}$ such that $\theta^*(h_m) \downarrow \theta^*(\mathcal{H})$. Clearly

$$\left\{\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(h_{m+1})\right\} \subseteq \left\{\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(h_m)\right\}$$

with each event occurring with probability 1. Again via measure continuity it follows that

$$\mathbf{P}\left\{\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(\mathcal{H})\right\} = \lim_{m\to\infty} \mathbf{P} \bigcap_{k=1}^{m} \left\{\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(h_m)\right\} = 1.$$

Thus we have that

$$\limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(\mathcal{H}) \coloneqq \inf_{h\in\mathcal{H}} \theta^*(h), \qquad \text{a.s.}$$

Now we look at the $\liminf$ side of the argument. At this point, we have

$$0 \leq \liminf_n \widehat{\theta}(\mathcal{H}) \leq \limsup_n \widehat{\theta}(\mathcal{H}) \leq \theta^*(\mathcal{H}),$$

which follows from the above argument and the fact that $L \geq 0$, so

$$\widehat{\theta}(h) \geq 0 \text{ and } |\liminf_n \widehat{\theta}(h)| < \infty$$

almost surely. Label the event

$$\boldsymbol{A}' \coloneqq \left\{\liminf_n \widehat{\theta}(\mathcal{H}) < \theta^*(\mathcal{H})\right\},$$

and start by assuming $\mathbf{P}\,\boldsymbol{A}' > 0$. On this event, we can fix a distance $\delta > 0$ such that taking n over $\mathbb{N}$, the sequence $\widehat{\theta}(\mathcal{H})$ drops more than δ below $\theta^*(\mathcal{H})$ infinitely often. To make this more concrete, fix $\theta_L \coloneqq \liminf_n \widehat{\theta}(\mathcal{H})$, and take any $\delta \in (0, \theta^*(\mathcal{H}) - \theta_L)$. Then for all $N < \infty$, can find index $n \geq N$ where $\widehat{\theta}(\mathcal{H}) < \theta^*(\mathcal{H}) - \delta < \theta^*(\mathcal{H})$. This gap, between $\widehat{\theta}(\mathcal{H})$ and $\theta^*(\mathcal{H})$, of at least δ, occurs infinitely often. For any such n, we have

$$\begin{aligned} \mathbf{E}_\mu\, \psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \widehat{\theta}(\mathcal{H})}{s_{h_n}(\boldsymbol{z})}\right) &> \mathbf{E}_\mu\, \psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \theta^*(\mathcal{H})}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) \\ &\geq \mathbf{E}_\mu\, \psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \theta^*(\widehat{h}_n)}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) \\ &= 0. \end{aligned}$$

The second inequality and the final inequality hold for all n, by the optimality of $\theta^*(\mathcal{H})$ and the definition of $\theta^*(\cdot)$. Depending on the $\omega \in \boldsymbol{A}'$, the actual value of this $\delta > 0$ will differ, but what matters is that such a δ-gap is fixed as we take n over $\mathbb{N}$. By the $\limsup$ bound shown above, taking any $\theta_U \in (\theta^*(\mathcal{H}), \infty)$, we have that $\widehat{\theta}(\mathcal{H}) \in [0, \theta_U]$ eventually. That is, there exists $N < \infty$ where $n \geq N$ implies $\widehat{\theta}(\mathcal{H}) \in [0, \theta_U]$. Using concavity, note for any $u^* > 0$, $u \in [0, u^* - \delta]$, $s > 0$ and $l \geq 0$, we have

$$\psi\left(\frac{l-u}{s}\right) - \psi\left(\frac{l-u^*}{s}\right) \geq \frac{\delta}{s}\psi'\left(\frac{l-u^*+\delta}{s}\right).$$

Set $l = l(h; \boldsymbol{z})$, $u^* = \theta^*(\mathcal{H})$, $s = s_h(\boldsymbol{z})$, and integrate with $\mathbf{E}_\mu$. Note that by assumption, there exists $\epsilon > 0$ such that

$$\delta\,\mathbf{E}_\mu \frac{1}{s_{\widehat{h}_n}(\boldsymbol{z})}\psi'\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \theta^*(\mathcal{H}) + \delta}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) \geq \frac{\delta}{s_2}\inf_{h \in \mathcal{H}} \mathbf{E}_\mu\,\psi'\left(\frac{l(h; \boldsymbol{z}) - \theta^*(\mathcal{H}) + \delta}{s_1}\right) \geq \epsilon$$

noting that ψ' is non-increasing on $\mathbb{R}_+$, by concavity of ψ. We thus have that on the event $\boldsymbol{A}'$, and any "bad index" n where $\widehat{\theta}(\mathcal{H}) < \theta^*(\mathcal{H}) - \delta$, we have

$$\mathbf{E}_\mu\,\psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \widehat{\theta}(\mathcal{H})}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) - \mathbf{E}_\mu\,\psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \theta^*(\mathcal{H})}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) \geq \epsilon > 0.$$

Since this occurs infinitely often as n ranges over $\mathbb{N}$ and ϵ is free of n, it implies

$$\boldsymbol{A}' \subseteq \left\{\lim_{n\to\infty} \mathbf{E}_\mu\,\psi\left(\frac{l(\widehat{h}_n; \boldsymbol{z}) - \widehat{\theta}(\mathcal{H})}{s_{\widehat{h}_n}(\boldsymbol{z})}\right) = 0\right\}^c,$$

which contradicts the strong convergence guaranteed by Corollary 21, noting $\widehat{\theta}(\widehat{h}_n) = \widehat{\theta}(\mathcal{H})$ by definition and Theorem 14. We conclude $\mathbf{P}\,\boldsymbol{A}' = 0$, which is to say that almost surely

$$\liminf_n \widehat{\theta}(\mathcal{H}) \geq \theta^*(\mathcal{H}) \geq \limsup_n \widehat{\theta}(\mathcal{H}).$$

We thus conclude that $\widehat{\theta}(\widehat{h}_n) = \widehat{\theta}(\mathcal{H}) \to \theta^*(\mathcal{H})$ as $n \to \infty$. □

Proof of Proposition 23. We verify the statements by adapting a standard comparison function technique [31, Lemma 7.7]. Fix arbitrary sample $x_1, \ldots, x_n$ where $x_i = l(h; \boldsymbol{z}_i)$ in the setting of this chapter. We consider the case of arbitrary s, where it may be completely determined by μ_n. Here defining two functions

$$g(\theta) := \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{x_i - \theta}{s}\right) s$$

$$\widetilde{g}(u; \theta) := g(\theta) + \frac{1}{2ns}\sum_{i=1}^{n}\left(\left(\psi\left(\frac{x_i - \theta}{s}\right)s - u\right)^2 - \psi\left(\frac{x_i - \theta}{s}\right)^2 s^2\right),$$

for any choice of $\theta, u \in \mathbb{R}$, we have a bound from above in $\widetilde{g}(u; \theta) \geq g(\theta + u)$. To see this, note that the difference function $d_\theta(u) := \widetilde{g}(u; \theta) - g(\theta + u)$ satisfies

$$d_\theta(0) = 0, \quad d_\theta'(0) = 0, \quad d_\theta'' \geq 0$$

for any choice of θ. The first two follow immediately from definitions, and the final inequality follows from $\rho'' \leq 1$ assuming we've standardized ρ such that ρ' is 1-Lipschitz, noting

$$d_\theta''(u) = \frac{1}{s}\left(1 - \frac{1}{n}\sum_{i=1}^{n}\psi'\left(\frac{x_i - (\theta + u)}{s}\right)\right),$$

which implies $d_{\theta}(u) \geq 0$ for all $u \in \mathbb{R}$, and also

$$g(\theta) - g(\theta + u) \geq g(\theta) - \widetilde{g}(u;\theta). \tag{3.23}$$

To make the best possible update to θ, we should set u to maximize the right-hand side, equivalently minimize $\widetilde{g}(u;\theta)$. Noting $\widetilde{g}'' = 1/s > 0$, and defining

$$u_0(\theta) := \frac{s}{n}\sum_{i=1}^{n} \psi\left(\frac{x_i - \theta}{s}\right)$$

we have $\widetilde{g}'(u_0(\theta)) = 0$ which is thus the unique minimum. Plugging $u_0(\theta)$ into (3.23), some algebra reveals

$$g(\theta) - g(\theta + u_0(\theta)) \geq \frac{1}{2s}u_0(\theta)^2 \geq 0. \tag{3.24}$$

Note that the right-hand side is zero iff $\theta = \widehat{\theta}(h)$, otherwise it is strictly positive. Defining $\widehat{\theta}_{(k)} := \widehat{\theta}_{(k-1)} + u_0(\widehat{\theta}_{(k-1)})$ is equivalent to the update (3.3). Looking at sequences taking $k \in \mathbb{N}$, $g(\widehat{\theta}_{(k)})$ is bounded and monotonic, and thus convergent. Since it is also Cauchy, this naturally implies $u_0(\widehat{\theta}_{[t]}) \to 0$ as well, from which it follows that $\widehat{\theta}_{(k)} \to \widehat{\theta}(h)$. To see this, assume $\widehat{\theta}_{(k)}$ is not Cauchy. Then there exists scale $\varepsilon_0 > 0$ at which for any $K < \infty$, there exist $k, k' \geq K$ such that $|\widehat{\theta}_{(k)} - \widehat{\theta}_{(k')}| > \varepsilon_0$. By the update definition and (3.24), for fixed sample $\boldsymbol{Z}$ the sequence $\widehat{\theta}_{(k)}$ is bounded. For concreteness, denote these bounds as $0 \leq \widehat{\theta}_{(k)} \leq \theta_U$. By strong monotonicity of ψ it then follows that defining

$$\varepsilon_1 := \inf_{\theta \in [0,\theta_U]} \left|\sum_{i=1}^{n}\left(\psi\left(\frac{x_i - \theta}{s}\right) - \psi\left(\frac{x_i - (\theta \pm \varepsilon_0)}{s}\right)\right)\right|$$

we have $\varepsilon_1 > 0$, with a value that is determined once the sample $\boldsymbol{Z}$ is observed and the update routine is intitialized. On the bad indices k, k' where $|\widehat{\theta}_{(k)} - \widehat{\theta}_{(k')}| > \varepsilon_0$, we always have

$$\left|\sum_{i=1}^{n}\left(\psi\left(\frac{x_i - \widehat{\theta}_{(k)}}{s}\right) - \psi\left(\frac{x_i - \widehat{\theta}_{(k')}}{s}\right)\right)\right| \geq \varepsilon_1$$

which would imply that $u_0(\widehat{\theta}_{(k)})$ is not Cauchy, contradicting $u_0(\widehat{\theta}_{(k)}) \to 0$. Thus $\widehat{\theta}_{(k)}$ is convergent. Using continuity of ψ, we have

$$u_0\left(\lim_{k\to\infty} \widehat{\theta}_{(k)}\right) = \lim_{k\to\infty} u_0(\widehat{\theta}_{(k)}) = 0,$$

implying $\widehat{\theta}_{(k)} \to \widehat{\theta}(h)$.

Shifting our focus to the scale result, consider χ as in Defn. 15, but with some additional restrictions. Similar to ψ in the location estimation setting, treat χ as a gradient of some convex objective to be minimized. The general form of the objective function is to be

$$g(s) := \mathbf{E}_{\mu_n}\left(r\left(\frac{x-\gamma}{s}\right) + \beta\right)s, \quad s > 0$$

where the function $r(\cdot)$ is assumed to be $r \geq 0$, convex and even, with a unique minimum at $r(0) = 0$. In addition, $r(u)/u$ should be concave on $\mathbb{R}_{+}$. The idea then is to construct χ using the gradient of this auxiliary objective, namely we seek that

$$g'(s) = (-1)\,\mathbf{E}_{\mu_n}\left(\chi\left(\frac{x-\gamma}{s}\right)\right). \tag{3.25}$$

To achieve this given a valid r, one need only set $\chi(u) \coloneqq r'(u)u - r(u) - \beta$.

A brief remark on constructing valid robust control functions of this form. Perhaps the simplest setting of r with the desired properties is $r(u) = u^{1+k}$, for $k \in (0,1]$. Clearly $r'' > 0$ on $\mathbb{R}_+$, and since $(r(u)/u)'' = k(k-1)u^{k-2} \leq 0$, we have the concavity desired. Furthermore, $\chi(u) = ku^{1+k} - \beta$, and

$$s = \left(\frac{k}{\beta} \mathbf{E}_{\mu_n} (x - \gamma)^{1+k} \right)^{\frac{1}{1+k}}$$

is the unique root of $\mathbf{E}_{\mu_n} \chi((x-\gamma)/s)$ in $s > 0$. There are no issues with zero-valued solutions given this formulation.

Returning to the main proof, a critical property of the update (3.4) is that for any $k = 1, 2, \ldots$ we have

$$g(s_{(k)}) - g(s_{(k+1)}) \geq \frac{\beta}{s_{(k)}} \left(s_{(k+1)} - s_{(k)} \right)^2. \tag{3.26}$$

To simplify notation even further, denote $l_i \coloneqq x_i - \gamma$ for $i = 1, \ldots, n$. We set $\chi(u) \coloneqq r'(u)u - r(u) - \beta$ as above, and denote $\widetilde{\chi}(u) \coloneqq \chi(u) + \beta$. Just as for the location case, a comparison function is introduced of the form

$$\widetilde{g}(u; s) \coloneqq g(s) + (u - s)\beta + \frac{1}{n} \sum_{i=1}^{n} \widetilde{\chi} \left(\frac{l_i}{s} \right) \left(\frac{s^2}{u} - s \right).$$

A few remarks regarding this form. First of all, we want $\widetilde{g}(s;s) = g(s)$, thus the need for the first constant. The second term ensures β appears in the first derivative of $\widetilde{g}$. The third term takes the form that it does such that in addition to $u = s$ implying $g = \widetilde{g}$, we also get that $g'(\cdot)$ and $\widetilde{g}'(\cdot; s)$ coincide when evaluated at s. With this form, it is immediate as

$$\widetilde{g}'(u; s) = \frac{1}{n} \sum_{i=1}^{n} \widetilde{\chi} \left(\frac{l_i}{s} \right) \frac{s^2}{u^2} (-1) + \beta$$

since we can note

$$\begin{aligned} g'(s) &= \frac{1}{n} \sum_{i=1}^{n} r' \left(\frac{l_i}{s} \right) \left(\frac{l_i}{s} \right) (-1) + \frac{1}{n} \sum_{i=1}^{n} r \left(\frac{l_i}{s} \right) + \beta \\ &= (-1) \frac{1}{n} \sum_{i=1}^{n} \chi \left(\frac{l_i}{s} \right) \\ &= \widetilde{g}'(s; s). \end{aligned}$$

Defining the difference function for pre-fixed arbitrary $s > 0$ by $d_s(u) \coloneqq \widetilde{g}(u;s) - g(u)$, we have that $d_s(s) = 0$, $d_s'(s) = 0$. Since we want to show $d_s(u) \geq 0$ for all $u > 0$, it remains to show that $d_s(\cdot)$ is convex. This is straightforward, if one notices that there are positive constants α_0 and α_1 which depend on s but are free of u such that

$$\begin{aligned} d_s(u) &= \alpha_0 + \frac{\alpha_1}{u} + \frac{-1}{n} \sum_{i=1}^{n} r \left(\frac{l_i}{u} \right) u \\ &= \alpha_0 + \alpha_1 \sigma + \frac{-1}{n} \sum_{i=1}^{n} r \left(l_i \sigma \right) \frac{1}{\sigma} \end{aligned}$$

when defining $\sigma \coloneqq 1/u$. The first two terms together form an affine function of σ, and by assumption $r(u)/u$ is a concave function on $\mathbb{R}_{+}$. Note that having l_i scaling this has no impact on convexity, since letting $f(u) \coloneqq r(u)/u$ and for any $\alpha \neq 0$ setting $\widetilde{f}(u) \coloneqq r(\alpha u)/(\alpha u)$, using first-order characterization of concavity, we have for any $u, v \geq 0$ that

$$\begin{aligned}\widetilde{f}(u) - \widetilde{f}(v) &= f(\alpha u) - f(\alpha v)\\ &\leq (u - v)\alpha f'(\alpha v)\\ &= (u - v)\widetilde{f}'(v),\end{aligned}$$

showing $\widetilde{f}$ is concave on $\mathbb{R}_{+}$ when f is. Thus the third summand in d_s is a convex function of $\sigma > 0$, and $d_s(1/\sigma) \geq 0$ for all $\sigma > 0$, implying $d_s(u) \geq 0$ for all $u > 0$ as desired, and $\widetilde{g}(u; s) \geq g(u)$ for all $u > 0$. Since we seek an update routine where g gets smaller, fixing $s > 0$ as the scale value from a previous iteration, we naturally seek that $g(s) - g(u)$ is maximized in u. Note that $\widetilde{g}(\cdot; s)$ has its unique critical point at

$$u_A = \left(\frac{1}{n\beta}\sum_{i=1}^{n}\widetilde{\chi}\left(\frac{l_i}{s}\right)s^2\right)^{1/2} = s\left(1 + \frac{1}{n\beta}\sum_{i=1}^{n}\chi\left(\frac{l_i}{s}\right)\right)^{1/2}$$

noting that the term inside the square root is non-negative as $\chi \geq -\beta$ by definition. Plugging u_A into $\widetilde{g}(\cdot; s)$ and some algebra then readily yields

$$\begin{aligned}g(s) - g(u_A) &\geq g(s) - \widetilde{g}(u_A; s)\\ &= \frac{\beta}{s}(u_A - s)^2\end{aligned}$$

and thus implying 3.26 by the update definition (3.4).

We now move on to the final step of this proof. Initialize using $s_{(k)} > 0$. Beginning with some basic facts, note that by (3.26), we have

$$g(s_{(0)}) \geq g(s_{(k)}) \geq g(s_{(k+1)}) \geq 0,$$

so $g(s_{(k)})$ is a bounded, monotone sequence, and thus the limit $\lim\limits_{k\to\infty} g(s_{(k)})$ certainly exists and is finite. As for the sequence $s_{(k)}$, note first that

$$\widetilde{\chi}'(u) = \chi'(u) = ur''(u) > 0, \quad \forall\, u > 0.$$

It follows that χ and $\widetilde{\chi}'$ are uniquely minimized at 0, meaning in particular that unless $l_1 = \cdots = l_n = 0$, we have $\mathbf{E}_{\mu_n}\widetilde{\chi}(l_i/s_{[0]}) > 0$. Assuming a continuous distribution function, this occurs with probability zero. Thus by definition of the update rule, $s_{(k)} > 0$ almost surely for all $k \in \mathbb{N}$. Henceforth we assume at least once $l_i \neq 0$. An upper bound is also simple to check. Taking $s \to \infty$, necessarily $g(s) \to \infty$, meaning $g(s) > g(s_{(0)})$ for s large enough. Since $g(s_{(k)}) > g(s_{(0)})$ is a contradiction, necessarily $s_{(k)}$ is bounded above as well. Regarding convergence, note that

$$g''(u) = \frac{1}{n}\sum_{i=1}^{n} r''\left(\frac{l_i}{u}\right)\left(\frac{l_i^2}{u^3}\right),$$

so by strong convexity of r, $g'' > 0$ on $\mathbb{R}_{+}$, and has a unique minimum. Denote this by $u_0 \coloneqq \arg\min g(u)$. Certainly either $u_0 = 0$ or $u_0 > 0$ are possible, but convergence is readily confirmed as follows. Since

$$\begin{aligned}g(s_{(k)}) = g(s_{(k+1)}) &\iff \frac{1}{n}\chi\left(\frac{l_i}{s_{(k)}}\right) = 0\\ &\iff g(s_{(k)}) = \min_{u} g(u),\end{aligned}$$

we have that $g(s_{(k)}) \to \min_u g(u) = g(u_0)$ as $t \to \infty$. Now say $s_{(k)}$ under update (A) is not Cauchy. Then, there exists some $\varepsilon_0 > 0$ such that for any $K \in \mathbb{N}$, we can find bad indices $k_1, k_2 \geq K$ such that $|s_{(k_1)} - s_{(k_2)}| > \varepsilon_0$. Note that by continuity and strong convexity of g, for any $\varepsilon > 0$, we can find a $\delta > 0$ such that $|s - u_0| > \delta \implies |g(s) - g(u_0)| > \varepsilon$. Taking ε arbitrarily small lets us take δ arbitrarily small. Choose $\varepsilon > 0$ such that $\delta \leq \varepsilon_0/2$. Since $g(s_{(k)}) \to g(u_0)$, exists K_0 such that $k \geq K_0$ implies $|g(s_{(k)}) - g(u_0)| \leq \varepsilon$. Taking $K \geq K_0$ and bad indices $k_1, k_2 \geq K$, we have $|s_{(k_1)} - s_{(k_2)}| > \varepsilon_0 \geq 2\delta$, but also $|g(s_{(k)}) - g(u_0)| \leq \varepsilon$ for both $t = k_1, k_2$. Taking k_1 for instance, note that $s_{(k_1)} \in [u_0 - \delta, u_0 + \delta]$. Looking at k_2 then, one sees

$$\begin{aligned} |s_{(k_1)} - s_{(k_2)}| > \varepsilon_0 &\implies s_{(k_2)} \notin [u_0 - \delta, u_0 + \delta] \\ &\implies |g(s_{(k_2)}) - g(u_0)| > \varepsilon, \end{aligned}$$

a contradiction since $k_2 \geq K \geq K_0$. We conclude that $s_{(k)}$ must be Cauchy and thus convergent to the unique minimizer u_0, implying the desired result. □

Remark 24. It should be noted that the convergence of (3.4) given by Proposition 23 is convergence to a *solution*, but it is possible that the solution may in fact be zero. This depends on the loss observations (and thus choice of $h \in \mathcal{H}$), the form of r, and the value of $\chi(0) < 0$ in a rather complex manner. For any given sample $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ and candidate h, the solution will be positive if and only if $\mathbf{E}_{\mu_n} \chi((l(h; \boldsymbol{z}) - \gamma)/s)$ can be made positive for small enough $s > 0$, and the natural control for this is to ensure $\chi(0)$ is far enough below zero. Thus if χ is built following (3.25) with a strictly convex r and small enough β, one can rest assured that the $s_{(k)}$ updates of 3.4 used as a sub-routine in Algorithm 1 will converge to a positive solution.

Bibliography

[1] Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series.* US National Bureau of Standards.

[2] Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. (1997). Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631.

[3] Ash, R. B. and Doléans-Dade, C. A. (2000). *Probability and Measure Theory.* Academic Press, 2nd edition.

[4] Audibert, J.-Y. and Catoni, O. (2011). Robust linear least squares regression. *Annals of Statistics*, 39(5):2766–2794.

[5] Bartle, R. G. (1964). *The Elements of Real Analysis.* John Wiley & Sons, 1st edition.

[6] Bartlett, P. L., Long, P. M., and Williamson, R. C. (1996). Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452.

[7] Bartlett, P. L. and Mendelson, S. (2006). Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334.

[8] Bartlett, P. L., Mendelson, S., and Neeman, J. (2012). ℓ_1-regularized linear regression: persistence and oracle inequalities. *Probability Theory and Related Fields*, 154(1-2):193–224.

[9] Breiman, L. (1968). *Probability.* Addison-Wesley.

[10] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[11] Brent, R. P. (1973). *Algorithms for minimization without derivatives.* Prentice-Hall.

[12] Brownlees, C., Joly, E., and Lugosi, G. (2015). Empirical risk minimization for heavy-tailed losses. *Annals of Statistics*, 43(6):2507–2536.

[13] Catoni, O. (2009). High confidence estimates of the mean of heavy-tailed real random variables. *arXiv preprint arXiv:0909.5366.*

[14] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148–1185.

[15] Cucker, F. and Smale, S. (2002). On the mathematical foundations of learning. *Bulletin (New Series) of the American Mathematical Society*, 39(1):1–49.

[16] Dellacherie, C. and Meyer, P.-A. (1978). *Probabilities and Potential*, volume 29 of *North-Holland Mathematics Studies.* North-Holland.

[17] Devroye, L., Lerasle, M., Lugosi, G., and Oliveira, R. I. (2015). Sub-Gaussian mean estimators. *arXiv preprint arXiv:1509.05845.*

[18] Dudley, R. M. (1978). Central limit theorems for empirical measures. *Annals of Probability*, 6(6):899–929.

[19] Dudley, R. M. (2014). *Uniform Central Limit Theorems.* Cambridge University Press, 2nd edition.

[20] Fan, J., Fan, Y., and Barut, E. (2014a). Adaptive robust variable selection. *Annals of Statistics*, 42(1):324.

[21] Fan, J., Li, Q., and Wang, Y. (2014b). Robust estimation of high-dimensional mean regression. *arXiv preprint arXiv:1410.2150v1*.

[22] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[23] Geman, D. and Reynolds, G. (1992). Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383.

[24] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741.

[25] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

[26] Hsu, D., Kakade, S. M., and Zhang, T. (2014). Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600.

[27] Hsu, D. and Sabato, S. (2014). Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML2014)*, pages 37–45.

[28] Hsu, D. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, 17(18):1–40.

[29] Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

[30] Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, 1st edition.

[31] Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*. John Wiley & Sons, 2nd edition.

[32] Kearns, M. J. and Schapire, R. E. (1994). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497.

[33] Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1):33–50.

[34] Lerasle, M. and Oliveira, R. I. (2011). Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914*.

[35] Lugosi, G. and Mendelson, S. (2016). Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757*.

[36] Minsker, S. (2015). Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335.

[37] Pollard, D. (1981). Limit theorems for empirical processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(2):181–195.

[38] Pollard, D. (1984). *Convergence of Stochastic Processes*. Springer-Verlag.

[39] R Core Team (2016). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

[40] Rousseeuw, P. and Yohai, V. (1984). Robust regression by means of S-estimators. In *Robust and Nonlinear Time Series Analysis*, volume 26 of *Lecture Notes in Statistics*, pages 256–272. Springer.

[41] Rudin, W. (1976). *Principles of Mathematical Analysis.* McGraw-Hill, 3rd edition.

[42] Salibian-Barrera, M. and Yohai, V. J. (2006). A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics*, 15(2):1–14.

[43] Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2010). Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670.

[44] Srebro, N., Sridharan, K., and Tewari, A. (2010). Smoothness, low noise and fast rates. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2199–2207.

[45] Steele, J. M. (1975). *Combinatorial entropy and uniform limit laws.* PhD thesis, Stanford University.

[46] Takeuchi, I., Le, Q. V., Sears, T. D., and Smola, A. J. (2006). Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264.

[47] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1):267–288.

[48] Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280.

[49] Vardi, Y. and Zhang, C.-H. (2000). The multivariate L_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426.

[50] Yu, Y., Aslan, Ö., and Schuurmans, D. (2012). A polynomial-time form of robust regression. In *Advances in Neural Information Processing Systems 25*, pages 2483–2491.

Chapter 4

Learning using robust gradients

4.1 Introduction

Any successful machine learning application depends both on procedures for reliable statistical inference, and a computationally efficient implementation of these procedures. This can be formulated using a risk $R(\boldsymbol{w}) := \mathbf{E}\, l(\boldsymbol{w}; \boldsymbol{z})$, induced by a loss l, where $\boldsymbol{w}$ is the parameter (vector, function, set, etc.) to be specified, and expectation is with respect to $\boldsymbol{z}$, namely the underlying data distribution. Given data $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, if an algorithm outputs $\widehat{\boldsymbol{w}}$ such that $R(\widehat{\boldsymbol{w}})$ is small with high probability over the random draw of the sample, this is formal evidence for good generalization, up to assumptions on the distribution. Performance-wise, the statistical side is important because R is always unknown, and the method of implementation is important since the only $\widehat{\boldsymbol{w}}$ we ever have in practice is one we can actually compute.

Empirical risk minimization (ERM), which admits any minimizer of $n^{-1} \sum_{i=1}^{n} l(\cdot; \boldsymbol{z}_i)$, is the canonical strategy for machine learning problems, and there exists a rich body of literature on its generalization ability [20, 4, 2, 5]. In recent years, however, some severe limitations of this technique have come into light. ERM can be implemented by numerous methods, but its performance is sensitive to this implementation [11, 13], showing sub-optimal guarantees on tasks as simple as multi-class pattern recognition, let alone tasks with unbounded losses. A related issue is highlighted in recent work by Lin and Rosasco [25], where we see that ERM implemented using a gradient-based method only has appealing guarantees when the data is distributed sharply around the mean in a sub-Gaussian sense. These results are particularly important due to the ubiquity of gradient descent (GD) and its variants in machine learning. They also carry the implication that ERM under typical implementations is liable to become highly inefficient whenever the data has heavy tails, requiring a potentially infinitely large sample to achieve a small risk.

Since tasks with such "inconvenient" data are common [14], it is of interest to investigate and develop alternative procedures which can be implemented as readily as the GD-based ERM (henceforth, ERM-GD), but which have desirable performance for a wider class of learning problems. In this chapter, we introduce and analyze an iterative routine which takes advantage of robust estimates of the risk gradient.

Review of related work Here we review some of the technical literature related to our work. As mentioned above, the analysis of Lin and Rosasco [25] includes the generalization of ERM-GD for sub-Gaussian observations. ERM-GD provides a key benchmark to be compared against; it is of particular interest to find a technique that is competitive with ERM-GD when it is optimal, but which behaves better under less congenial data distributions. Other researchers have investigated methods for distribution-robust learning. One notable line of work looks at generalizations of the "median of means" procedure, in which one constructs candidates on disjoint partitions of the data, and aggregates them such that anomalous candidates are effectively ignored. These methods can be implemented and have theoretical guarantees, ranging from the one-dimensional setting [24, 29] to multi-dimensional and even functional models [28, 17, 23]. Their main limitation is practical: when sample size n is small relative to the complexity of the model, very few subsets can be created, and robustness is poor; conversely, when n is large enough to make many candidates, cheaper and less sophisticated methods often suffice.

An alternative approach is to use all the observations to construct robust estimates $\widehat{R}(\boldsymbol{w})$ of the risk $R(\boldsymbol{w})$ for each $\boldsymbol{w}$ to be checked, and subsequently minimize $\widehat{R}$ as a surrogate. An elegant strategy using M-estimates of R was introduced by Brownlees et al. [7], based on fundamental results due to Catoni [8, 9]. While the statistical guarantees are near-optimal under very weak assumptions on the data, the proxy objective $\widehat{R}$ is defined implicitly, introducing many computational roadblocks. In particular, even if R is convex, the estimate $\widehat{R}$ need not be, and the non-linear optimization required by this method does not scale well to higher dimensions.

Our contributions To deal with these limitations of ERM-GD and its existing robust alternatives, the key idea here is to use robust estimates of the risk gradient, rather than the risk itself, and to feed these estimates into a first-order steepest descent routine. In doing so, at the cost of minor computational overhead, we get formal performance guarantees for a wide class of data distributions, while enjoying the computational ease of a gradient descent update. Our main contributions:

- A learning algorithm which addresses the vulnerabilities of ERM-GD, is easily implemented, and can be parallelized or adapted to stochastic sub-sampling for big problems.
- High-probability bounds on excess risk of this procedure, which hold under mild moment assumptions on the data distribution, and suggest a promising methodology.
- In a variety of learning settings, numerical tests comparing our routine with ERM-GD and other cited benchmarks reinforce the practical utility and flexibility suggested by the theory.

Content overview In section 4.2, we introduce the key components of the proposed algorithm, and provide an intuitive example meant to highlight the learning principles taken advantage of. Theoretical analysis of algorithm performance is given in section 4.3, including a sketch of the proof technique and discussion of the main results. Empirical analysis follows in section 4.4, in which we elucidate both the strengths and limits of the proposed procedure, through a series of tightly controlled numerical tests. Finally, concluding remarks and a look ahead are given in section 4.5.

4.2 Robust gradient descent

Before introducing the proposed algorithm in detail, we motivate the practical need for a procedure which deals with the weaknesses of the traditional sample mean-based gradient descent strategy.

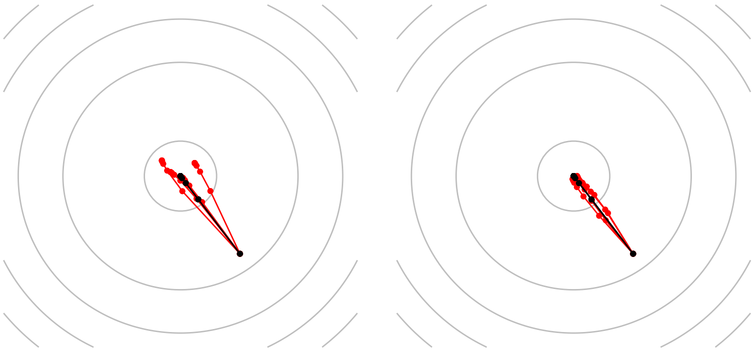

Figure 4.1: A comparison of the minimizing sequence trajectories in a two-dimensional approximate risk minimization task, for the traditional ERM-based gradient descent (left) and a simple re-weighting procedure (right). Trajectories of the oracle update using R' (black) is pictured alongside the approximate methods (red). All procedures use $\alpha_{(t)} = 0.35$, $t = 0, \ldots, 9$.

4.2.1 Why robustness?

Recall that since ERM admits any minima of $n^{-1} \sum_{i=1}^{n} l(\cdot; \boldsymbol{z}_i)$, the simplest implementation of gradient descent (for $\widehat{\boldsymbol{w}}_{(t)} \in \mathbb{R}^d$) results in the update

$$\widehat{\boldsymbol{w}}_{(t+1)} = \widehat{\boldsymbol{w}}_{(t)} - \alpha_{(t)} \frac{1}{n} \sum_{i=1}^{n} l'(\widehat{\boldsymbol{w}}_{(t)}; \boldsymbol{z}_i) \tag{4.1}$$

where $\alpha_{(t)}$ are scaling parameters. Taking the derivative under the integral we have $R'(\cdot) = \mathbf{E}\, l'(\cdot; \boldsymbol{z})$, meaning ERM-GD uses the sample mean as an estimator of each coordinate of R', in pursuit of a solution minimizing the unknown R. Without rather strong assumptions on the tails and moments of the distribution of $l(\boldsymbol{w}; \boldsymbol{z})$ for each $\boldsymbol{w}$, it has become well-known that the sample mean fails to provide sharp estimates [9, 28, 12, 27]. Intuitively, the issue is that we expect bad estimates to imply bad approximate minima. Does this formal sub-optimality indeed manifest itself in natural settings? Can principled modifications improve performance at a tolerable cost?

A simple example suggests affirmative answers to both questions. The plot on the left of Figure 4.1 shows contour lines of a strongly convex quadratic risk to be minimized, as well as the trajectory of 10 iterations of ERM-GD, given four independent samples from a common distribution, initiated at a common $\widehat{\boldsymbol{w}}_{(0)}$. With data $\boldsymbol{z} = (\boldsymbol{x}, y) \in \mathbb{R}^{d+1}$, losses are generated as $l(\boldsymbol{w}; \boldsymbol{z}_i) = (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - y_i)^2/2$. We consider the case where the "noise" $\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - y_i$ is heavy-tailed (log-Normal). Half of the samples saw relatively good solutions after ten iterations, and half saw rather stark deviation from the optimal procedure. When the sample contains errant observations, the empirical mean estimate is easily influenced by such points.

To deal with this, a classical idea is to re-weight the observations in a principled manner, and then carry out gradient descent as normal. That is, in the gradient estimate of (4.1), we replace the summands $n^{-1} l'(\cdot; \boldsymbol{z}_i)$ with $\omega_i l'(\cdot; \boldsymbol{z}_i)$, where $0 \leq \omega_i \leq 1$, $i \in [n]$ and $\sum_{i=1}^{n} \omega_i = 1$. For example, we could set

$$\omega_i := \frac{\widetilde{\omega}_i}{\sum_{k=1}^{n} \widetilde{\omega}_k}, \quad \widetilde{\omega}_i := \frac{\psi\left(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y_i\right)}{\left(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y_i\right)}$$

where ψ is an odd function of sigmoidal form (see 4.6.1). The idea is that for observations $\boldsymbol{z}_i$ that induce errors which are *inordinately* large, the weight ω_i will be correspondingly small, reducing the impact. In the right-hand plot of Figure 4.1, we give analogous results for this procedure, run under the exact same settings as ERM-GD above. The modified procedure at least appears to be far more robust to random idiosyncracies of the sample; indeed, if we run many trials, the average risk is far better than the ERM-GD procedure, and the variance smaller. The fragility observed here was in the elementary setting of $d = 2$, $n = 500$; it follows *a fortiori* that we can only expect things to get worse for ERM-GD in higher dimensions and under smaller samples. In what follows, we develop a robust gradient-based minimization method based directly on the principles illustrated here.

4.2.2 Algorithm introduction

Were the risk to be known, we could update using

$$\boldsymbol{w}^{*}_{(t+1)} := \boldsymbol{w}^{*}_{(t)} - \alpha_{(t)}\boldsymbol{g}(\boldsymbol{w}^{*}_{(t)}) \tag{4.2}$$

where $\boldsymbol{g}(\boldsymbol{w}) := R'(\boldsymbol{w})$, an ideal procedure. Any learning algorithm in practice will not have access to R or $\boldsymbol{g}$, and thus must approximate this update with

$$\widehat{\boldsymbol{w}}_{(t+1)} := \widehat{\boldsymbol{w}}_{(t)} - \alpha_{(t)}\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t)}), \tag{4.3}$$

where $\widehat{\boldsymbol{g}}$ represents some sample-based estimate of $\boldsymbol{g}$. Setting $\widehat{\boldsymbol{g}}$ to the sample mean reduces to ERM-GD, and conditioned on $\widehat{\boldsymbol{w}}_{(t)}$, $\mathbf{E}\,\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t+1)}) = \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t+1)})$, a property used throughout the literature [34, 22, 19, 35, 16, 30]. While convenient from a technical standpoint, there is no conceptual necessity for $\widehat{\boldsymbol{g}}$ to be unbiased. More realistically, as long as $\widehat{\boldsymbol{g}}$ is sharply distributed around $\boldsymbol{g}$, then an approximate first-order procedure should not deviate too far from the ideal, even if these estimators are biased. An outline of such a routine is given in Algorithm 2.

Algorithm 2 Outline of robust gradient descent (`rgd`)

```
for $t \in \{0, 1, \ldots, T-1\}$ do
    $D_{(t)} \leftarrow \{l'(\widehat{\boldsymbol{w}}_{(t)}; \boldsymbol{z}_i)\}_{i=1}^{n}$    ▷ Update loss gradients.
    $\widehat{\boldsymbol{\sigma}}_{(t)} \leftarrow$ RESCALE$(D_{(t)})$    ▷ Eqn. (4.5); scale for truncation.
    $\widehat{\boldsymbol{\theta}}_{(t)} \leftarrow$ LOCATE$(D_{(t)}, \widehat{\boldsymbol{\sigma}}_{(t)})$    ▷ Eqns. (4.4), (4.6); truncate losses, estimate risk gradient.
    $\widehat{\boldsymbol{w}}_{(t+1)} \leftarrow \widehat{\boldsymbol{w}}_{(t)} - \alpha_{(t)}\widehat{\boldsymbol{\theta}}_{(t)}$
end for
```

Let us flesh out the key sub-routines used in a single iteration, for the $\boldsymbol{w} \in \mathbb{R}^d$ case. When the data is prone to outliers, a "soft" truncation of errant values is a prudent alternative to discarding valuable data. We saw a rudimentary application of this maxim in section 4.2.1. This can be done systematically using a convenient class of M-estimators of location and scale [38, 18]. The LOCATE sub-routine entails taking a convex, even function ρ, and for each coordinate, computing $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_d)$ as

$$\widehat{\theta}_j \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \sum_{i=1}^{n} \rho\left(\frac{l'_j(\boldsymbol{w}; \boldsymbol{z}_i) - \theta}{s_j}\right), \quad j = 1, \ldots, d. \tag{4.4}$$

Note that if $\rho(u) = u^2$, then $\widehat{\theta}_j$ reduces to the sample mean of $\{l'_j(\boldsymbol{w}; \boldsymbol{z}_i)\}_{i=1}^{n}$, thus to reduce the impact of extreme observations, it is useful to take $\rho(u) = o(u^2)$ as $u \to \pm\infty$. Here the $s_j > 0$ factors are used to ensure that consistent estimates take place irrespective of the order of magnitude of the observations. We set the scaling factors in two steps. First is RESCALE, in which a rough dispersion estimate of the data is computed using

$$\widehat{\sigma}_j \in \left\{\sigma > 0 : \sum_{i=1}^{n} \chi\left(\frac{l'_j(\boldsymbol{w}; \boldsymbol{z}_i) - \gamma_j}{\sigma}\right) = 0, \quad j = 1, \ldots, d.\right\}. \tag{4.5}$$

Here $\chi : \mathbb{R} \to \mathbb{R}$ is an even function, satisfying $\chi(0) < 0$, and $\chi(u) > 0$ as $u \to \pm\infty$ to ensure that the resulting $\widehat{\sigma}_j$ is an adequate measure of the dispersion of $l'_j(\boldsymbol{w}; \boldsymbol{z})$ about a pivot point, say $\gamma_j = \sum_{i=1}^{n} l'_j(\boldsymbol{w}; \boldsymbol{z}_i)/n$. Second, we adjust this estimate based on the available sample size and desired confidence level, as

$$s_j = \widehat{\sigma}_j \sqrt{n/\log(2\delta^{-1})} \tag{4.6}$$

where $\delta \in (0, 1)$ specifies the desired confidence level $(1-\delta)$, and n is the sample size. This last step appears rather artificial, but can be derived from a straightforward theoretical argument, given in section 4.3.1. This concludes all the steps[1] in one full iteration of Algorithm 2 on $\mathbb{R}^d$.

In the remainder of this chapter, we shall investigate the learning properties of this procedure, through analysis of both a theoretical (section 4.3) and empirical (section 4.4) nature. As an example, in the strongly convex risk case, our formal argument yields excess risk bounds of the form

$$R(\widehat{\boldsymbol{w}}_{(T)}) - R^* \leq O\left(\frac{d\log(dT\delta^{-1})}{n}\right) + O\left((1-\alpha\beta)^T\right)$$

with probability no less than $1-\delta$, for small enough $\alpha_{(t)} = \alpha$ over T iterations. Here $\beta > 0$ is a constant that depends only on R, and analogous results hold without strong convexity. Of the underlying distribution, all that is assumed is a bound on the variance of $l'(\cdot; \boldsymbol{z})$, suggesting formally that the procedure should be competitive over a diverse range of data distributions.

4.3 Theoretical analysis

Here we analyze the performance of Algorithm 2 on hypothesis class $\mathcal{W} \subseteq \mathbb{R}^d$, as measured by the risk achieved, which we estimate using upper bounds that depend on key parameters of the learning task. A general sketch is given, followed by some key conditions, representative results, and discussion. All proofs are relegated to 4.6.

Notation For integer k, write $[k] := \{1, \ldots, k\}$ for all the positive integers from 1 to k. Let μ denote the data distribution, with $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ independent observations from μ, and $\boldsymbol{z} \sim \mu$ an independent copy. Risk is then $R(\boldsymbol{w}) := \mathbf{E}_\mu\, l(\boldsymbol{w}; \boldsymbol{z})$, its gradient $\boldsymbol{g}(\boldsymbol{w}) := R'(\boldsymbol{w})$, and $R^* := \inf_{\boldsymbol{w}\in\mathcal{W}} R(\boldsymbol{w})$. $\mathbf{P}$ denotes a generic probability measure, typically the product measure induced by the sample. We write $\|\cdot\|$ for the usual (ℓ_2) norm on $\mathbb{R}^d$. For function F on $\mathbb{R}^d$ with partial derivatives defined, write the gradient as $F'(\boldsymbol{u}) := (F'_1(\boldsymbol{u}), \ldots, F'_d(\boldsymbol{u}))$ where for short, we write $F'_j(\boldsymbol{u}) := \partial F(\boldsymbol{u})/\partial u_j$. In addition to asymptotic notation O and O_P [38], we use $\lesssim$ to suppress terms which are not of leading order.

4.3.1 Sketch of the general argument

The analysis here requires only two steps:

- A good estimate $\widehat{\boldsymbol{g}} \approx \boldsymbol{g}$ implies that approximate update (4.3) is near the optimal update.
- Under variance bounds, coordinate-wise M-estimation yields a good gradient estimate.

[1] For concreteness, in all empirical tests to follow we use the Gudermannian function [1], $\rho(u) = \int_0^u \psi(x)\,dx$ where $\psi(u) = 2\,\mathrm{atan}(\exp(u)) - \pi/2$, and $\chi(u) = u^2/(1+u^2) - c$, for a constant $c > 0$. General conditions on ρ, as well as standard methods for computing the M-estimates, namely the $\widehat{\theta}_j$ and $\widehat{\sigma}_j$, are given in 4.6.1.

We are then able to conclude that with enough samples and iterations (but not too many iterations), the output of Algorithm 2 can achieve an arbitrarily small excess risk. Here we spell out the key facts which motivate this approach.

For the first step, let $\boldsymbol{w}^{*} \in \mathbb{R}^{d}$ be a minimizer of R. When the risk R is strongly convex, then using well-established convex optimization theory [31], we can easily control $\|\boldsymbol{w}^{*}_{(t+1)} - \boldsymbol{w}^{*}\|$ as a function of $\|\boldsymbol{w}^{*}_{(t)} - \boldsymbol{w}^{*}\|$ for any step $t \geq 0$. Thus to control $\|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^{*}\|$, in comparing the approximate case and optimal case, all that matters is the difference between $\boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})$ and $\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t)})$ (Lemma 28). For the general case of convex R, since we cannot easily control the distance of the optimal update from any potential minimum, we instead directly compare the trajectories of $\widehat{\boldsymbol{w}}_{(t)}$ and $\boldsymbol{w}^{*}_{(t)}$ over $t = 0, 1, \ldots, T$, which once again amounts to a comparison of $\boldsymbol{g}$ and $\widehat{\boldsymbol{g}}$ (Lemma 30). This inevitably leads to more error propagation and thus a stronger dependence on T, but the essence of the argument is identical to the strongly convex case.

For the second step, since $\widehat{\boldsymbol{g}}$ is based on a random sample $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, we need an estimation technique which admits guarantees for any choice of $\boldsymbol{w}$, with high probability over the random draw of this sample. A basic requirement is that

$$\mathbf{P}\left\{\|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \leq \varepsilon\right\} \geq 1 - \delta, \quad \forall\, \boldsymbol{w} \in \mathcal{W}. \tag{4.7}$$

Then running Algorithm 2 for T steps, we can invoke (4.7) once for each step, and use a union bound to get a $1 - T\delta$ event on which $\widehat{\boldsymbol{w}}_{(T)}$ closely approximates the optimal GD output, up to the accuracy specified by ε. Naturally this ε will depend on confidence level δ, which implies that to get $1 - \delta$ confidence intervals, the upper bound in (4.7) will depend on T. As an example, assume we were to run ERM-GD, namely using an empirical mean estimate of the gradient. Using Chebyshev's inequality,[2] with probability $1 - \delta$ all we can guarantee is $\varepsilon \leq O(\sqrt{Td/(n\delta)})$. On the other hand, with a reasonable ancillary estimate of the gradient variance, one can readily construct a stronger estimator using a smooth truncation scheme as in the LOCATE sub-routine of Algorithm 2. One important property of ρ in (4.4) is that we can show

$$-\log(1 - u + Cu^{2}) \leq \rho'(u) \leq \log(1 + u + Cu^{2}), \quad \forall\, u \in \mathbb{R} \tag{4.8}$$

for a fixed $C > 0$, a simple generalization of the key property utilized by Catoni [9]. For the Gudermannian function (section 4.2.2 footnote), we can take $C \leq 2$, with the added benefit that ρ' is bounded and increasing. As to the quality of these estimates, note that they are distributed sharply around the risk gradient, as follows.

Lemma 25 (Concentration of M-estimates)**.** *For each coordinate $j \in [d]$, the estimates $\widehat{\theta}_j$ of (4.4) satisfy*

$$\mathbf{P}\left\{\frac{1}{2}|\widehat{\theta}_j - g_j(\boldsymbol{w})| \leq \frac{C\,\mathrm{var}_{\mu}\, l'_j(\boldsymbol{w};\boldsymbol{z})}{s_j} + \frac{s_j \log(2\delta^{-1})}{n}\right\} \geq 1 - \delta, \tag{4.9}$$

for large enough n and s_j.

To get the tightest possible confidence interval as a function of $s_j > 0$, we must set

$$s_j^{2} = \frac{Cn\,\mathrm{var}_{\mu}\, l'_j(\boldsymbol{w};\boldsymbol{z})}{\log(2\delta^{-1})},$$

from which we derive (4.6), with $\widehat{\sigma}_j^{2}$ corresponding to a computable estimate of $\mathrm{var}_{\mu}\, l'_j(\boldsymbol{w};\boldsymbol{z})$. If the variance over all choices of $\boldsymbol{w}$ is bounded by some $V < \infty$, then up to the variance estimates,

[2] In the statistics and probability literature, this is often called Markov's inequality.

we have $\|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \leq O(\sqrt{dV\log(2d\delta^{-1})/n})$, with $\widehat{\boldsymbol{g}} = \widehat{\boldsymbol{\theta}}$ from Algorithm 2, yielding a bound for (4.7) free of $\boldsymbol{w}$. This $\log(T/\delta)$ dependence provides an exponential improvement over the T/δ dependence in the case of ERM-GD, and an appealing formal motivation for using M-estimates of location as an alternative strategy.

4.3.2 Conditions and results

On the learning task, we make the following assumptions.

(A1) Risk $R(\cdot)$ is to be minimized over closed, convex $\mathcal{W} \subseteq \mathbb{R}^d$.

(A2) R is λ-smooth, convex, and continuously differentiable on $\mathcal{W}$.

(A3) There exists $\boldsymbol{w}^* \in \mathcal{W}$ at which $\boldsymbol{g}(\boldsymbol{w}^*) = 0$.

(A4) Distribution μ satisfies $\operatorname{var}_\mu l_j'(\boldsymbol{w};\boldsymbol{z}) \leq V < \infty$, for all $\boldsymbol{w} \in \mathcal{W}$, $j \in [d]$.

Algorithm 2 is run following (4.4), (4.5), and (4.6) as specified in section 4.2. For RESCALE, the choice of χ is only important insofar as the scale estimates (the $\widehat{\sigma}_j$) should be moderately accurate. To make the dependence on this accuracy precise, take constants $c_{min}, c_{max} > 0$ such that

$$c_{min}\sqrt{\operatorname{var}_\mu l_j'(\boldsymbol{w};\boldsymbol{z})} \leq \widehat{\sigma}_j \leq c_{max}\sqrt{\operatorname{var}_\mu l_j'(\boldsymbol{w};\boldsymbol{z})}, \quad j \in [d] \tag{4.10}$$

for all choices of $\boldsymbol{w} \in \mathcal{W}$, and write $c_0 := (c_{max} + C/c_{min})$. For $1-\delta$ confidence, we need a large enough sample; more precisely, for each $\boldsymbol{w}$, it is sufficient if

$$\frac{1}{4} \geq \frac{C\log(2\delta^{-1})}{n}\left(1 + \frac{C\operatorname{var}_\mu l_j'(\boldsymbol{w};\boldsymbol{z})}{\widehat{\sigma}_j^2}\right), \quad j \in [d]. \tag{4.11}$$

For simplicity, fix a small enough step size,

$$\alpha_{(t)} = \alpha, \forall\, t \in \{0,\ldots,T-1\}, \quad \alpha \in (0, 2/\lambda). \tag{4.12}$$

Dependence on initialization is captured by $R_0 := R(\boldsymbol{w}^*_{(0)}) - R^*$, and $D_0 := \|\boldsymbol{w}^*_{(0)} - \boldsymbol{w}^*\|$. Under this setup, we can control the estimation error.

Lemma 26 (Accuracy of gradient estimates)**.** *For each step $t = 0,\ldots,T-1$ of Algorithm 2, we have*

$$\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| \leq 2c_0\sqrt{\frac{dV\log(2d\delta^{-1})}{n}}$$

with probability no less than $1-\delta$.

Remark 27 (Projected descent case). The above analysis proceeds on the premise that $\widehat{\boldsymbol{w}}_{(t)} \in \mathcal{W}$ holds after all the updates, $t \in [T]$. To enforce this, a standard variant of Algorithm 2 is to update as

$$\widehat{\boldsymbol{w}}_{(t+1)} \leftarrow \pi_{\mathcal{W}}\left(\widehat{\boldsymbol{w}}_{(t)} - \alpha_{(t)}\widehat{\boldsymbol{\theta}}_{(t)}\right), \quad t \in \{0,\ldots,T-1\}$$

where $\pi_{\mathcal{W}}(\boldsymbol{u}) := \arg\min_{\boldsymbol{v}\in\mathcal{W}} \|\boldsymbol{u} - \boldsymbol{v}\|$. By (A1), this projection is well-defined [26, Sec. 3.12, Thm. 3.12]. Using this fact, it follows that $\|\pi_{\mathcal{W}}(\boldsymbol{u}) - \pi_{\mathcal{W}}(\boldsymbol{v})\| \leq \|\boldsymbol{u} - \boldsymbol{v}\|$ for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$, by which we can immediately show that Lemma 30 holds for the projected robust gradient descent version of Algorithm 2.

Under strongly convex risk In addition to assumptions (A1)–(A4), assume that R is κ-strongly convex. In this case, $\boldsymbol{w}^*$ in (A3) is the unique minimum. First, we control[3] the estimation error by showing that the approximate update (4.3) does not differ much from the optimal update (4.2).

Lemma 28 (Minimizer control)**.** *Assume that (4.7) holds with bound ε. Consider (4.3), with $\alpha_{(t)} = \alpha$ such that $0 < \alpha < 2/(\kappa + \lambda)$. Write $\beta := 2\kappa\lambda/(\kappa + \lambda)$. Then, with probability no less than $1 - T\delta$, we have*

$$\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*\| \leq (1 - \alpha\beta)^{T/2} D_0 + \frac{2\varepsilon}{\beta}.$$

With control over the distance of the robust GD approximation from the risk minimizer, the smoothness of R implies a risk bound.

Theorem 29 (Excess risk bounds)**.** *Write $\widehat{\boldsymbol{w}}_{(T)}$ for the output of Algorithm 2 after T iterations, run such that (4.11)–(4.12) hold, with step size $\alpha_{(t)} = \alpha$ for all $0 < t < T$, as in Lemma 28. It follows that*

$$R(\widehat{\boldsymbol{w}}_{(T)}) - R^* \leq \lambda(1 - \alpha\beta)^T D_0^2 + \frac{16\lambda c_0^2}{\beta^2} \frac{dV \log(2dT\delta^{-1})}{n}$$

with probability no less than $1 - \delta$.

Let us discuss these formal guarantees. There are two terms in the upper bound of Theorem 29: first an optimization term which decreases in T, and an estimation term which decreases in n, and increases in T. We intuitively expect that for any fixed n, taking T too large should result in overfitting. This is reflected by the tradeoff between optimization error and estimation error here. Notably, however, the impact of taking too large here (a $\log(T)$ factor) is much less than the case of running ERM-GD (a T factor), as we would expect. As the optimization term shows linear convergence (shrinks exponentially with T), T only needs to be large enough to make up for the initial error D_0, and thus we expect fast convergence. That said, even letting $T \to \infty$ as $n \to \infty$, as long as $T = o(\exp(n))$ holds, an arbitrarily small excess risk can be achieved. Furthermore, since $\alpha_{(t)}$ requirements are rather lenient, even a rather large pre-fixed step-size should be expected to perform well. Finally, we note that up to variance estimates, all we assume is finite second-order moments. If we assume finite kurtosis, the argument of Catoni [9] can be used to create analogous guarantees for an explicit scale estimation procedure. Most importantly, since the guarantees are available whether the data is sub-Gaussian or heavy-tailed with infinite higher-order moments, we have the important implication of robustness to the underlying distribution.

A related question is that of learning efficiency: are there important learning settings in which running only a few iterations is sufficient to match or outperform competing methods? Certainly, it would be desirable if it arrives at a better solution, faster, but this hinges on the estimate sharpness and initialization. How sensitive is this performance to initial $\widehat{\boldsymbol{w}}_{(0)}$ settings? Can c_0 be assumed small using dispersion-based estimates as in Algorithm 2? Are there problem settings in which the poor accuracy of ERM-GD makes a substantial impact on learning quality and efficiency? All of these points are addressed in the numerical experiments of section 4.4.

[3] This useful fact is a modified version of related results due to Chen et al. [10], whose preprint appeared after the preparation of our original manuscript. Using strong convexity makes for a much stronger argument than is possible in the convex case, as we discuss below.

Without strong convexity When the risk is not strongly convex, it becomes difficult to control the distance between $\widehat{\boldsymbol{w}}_{(t)}$ in (4.3) and any particular $\boldsymbol{w}^{*}$ minimizing R. Here we adopt the tactic of directly comparing the trajectories of our approximate $\widehat{\boldsymbol{w}}_{(t)}$ and the ideal procedure $\boldsymbol{w}^{*}_{(t)}$ given in (4.2), when initialized at a common point. Error propagates over iterations, but we can use a good gradient estimate to mitigate this effect.

Lemma 30 (Comparing trajectories)**.** *Comparing (4.2) and (4.3), assume that $\widehat{\boldsymbol{g}}$ satisfies (4.7). Setting $\alpha_{(t)} \in (0,1)$ for all $0 \leq t < T$, and initializing to $\widehat{\boldsymbol{w}}_{(0)} = \boldsymbol{w}^{*}_{(0)}$, with probability at least $1 - T\delta$, we have*

$$\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^{*}_{(T)}\| \leq \varepsilon \left(\prod_{t=0}^{T-1} (1 + \lambda\alpha_{(t)}) - 1 \right). \tag{4.13}$$

Using this property, we can obtain risk bounds analogous to those in the strong convexity scenario.

Corollary 31. *Run Algorithm 2 for T iterations as in Theorem 29, and fixing arbitrary constant $q \geq 1$, take step size as $\lambda\alpha \leq \exp(\log(q+1)/T) - 1$. We then have that*

$$R(\widehat{\boldsymbol{w}}_{(T)}) - R^{*} \lesssim 2\sqrt{2\lambda} c_0 q \left(\frac{CdV \log(2dT\delta^{-1}) \tau(T,\alpha)}{n} \right)^{1/2} + \tau(T,\alpha)$$

with probability no less than $1 - \delta$, where

$$\tau(T,\alpha) := \left(\frac{T\alpha(2 - \lambda\alpha)}{2D_0^2} + R_0 \right)^{-1}.$$

Here we discuss these results. The general form of the excess risk bound is similar to Theorem 29, though the impact of running too many iterations is decidely stronger. A more accurate approximation of $\widehat{\boldsymbol{w}}_{(t)} \approx \boldsymbol{w}^{*}_{(t)}$ at each step implies that we can run more iterations or take larger steps, which in turn controls how close the ideal $\boldsymbol{w}^{*}_{(T)}$ gets to a risk minimizer. How do settings of T, α, and the sample size n interact? As an illustrative example, perhaps the simplest setting is to fix $\alpha = 0.1$ for all n choices. Since this step size is rather large, error propagation may be significant, and one naturally expects that this algorithm should achieve its best performance after relatively few iterations. More precisely: to make use of a $\alpha = 0.1$ rate for any n, and achieve a $O(\log(n)/\sqrt{n})$ rate on the estimation error term, it is sufficient to set $T = 10\log(\log(n))$, since letting $q = \log(n) - 1$ then implies $\exp(\log(q+1)/T) - 1 \approx 0.105$. Needless to say, for pre-fixed α, T can only grow *very* slowly in n. On the other hand, if $\alpha = 1/\sqrt{n}$, the results hold for all $T \leq 1/\log_2(1 + n^{-1/2})$, a large relaxation. One would expect that taking T too large (stopping late) leads to overfitting, and these results support this intuition, with T taking the "regularizer" role as reported by Lin and Rosasco [25].

With all this understood, the particularly strong dependence on T is expected to be an artifact of the analysis technique, though the degree to which this can be improved is a matter of interest. The implications, however, are natural, as having only convexity makes minimizing R harder even when it is known, and thus *a fortiori* the approximate optimization required in risk minimization is also made substantially more difficult. If this dependence is not tight, is there much of a penalty for taking T too large? Since stopping conditions are non-obvious and problem-specific, one hopes that running until numerical convergence would not be sub-optimal. The key to this is whether our algorithm is accurate enough to "follow" the ideal algorithm closely for the first few iterations; if we can quickly get to a good region where $\|\boldsymbol{g}(\cdot)\|$ is very small, then we would not expect much damage from running the routine too long.

With prior information An interesting concept in machine learning is that of the relationship between learning efficiency, and the task-related prior information available to the learner. In the previous results, the learner is assumed to have virtually no information beyond the data available, and the ability to set a small enough step-size. What if, for example, just the gradient variance was known? A classic example from decision theory is the dominance of the estimator of James and Stein over the maximum likelihood estimator, in multivariate Normal mean estimation using prior variance information. In our more modern and non-parametric setting, the impact of rough, data-driven scale estimates was made explicit by the factor c_0. Here we give complementary results that show how partial prior information on the distribution μ can improve learning.

Lemma 32 (Accuracy with variance information)**.** *Run Algorithm 2, with* $\widehat{\boldsymbol{\sigma}}_{(t)} = (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_d)$ *modified to satisfy* $\widehat{\sigma}_j^2 = C \operatorname{var}_\mu l'_j(\widehat{\boldsymbol{w}}_{(t)}; \boldsymbol{z})$*,* $j \in [d]$ *over each step* $t = 0, \ldots, T-1$*. It follows that*

$$\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| \leq 4 \left(\frac{C \operatorname{trace}(\Sigma_{(t)}) \log(2d\delta^{-1})}{n} \right)^{1/2}$$

with probability no less than $1-\delta$*, where* $\Sigma_{(t)}$ *is the covariance matrix of* $l'(\widehat{\boldsymbol{w}}_{(t)}; \boldsymbol{z})$*.*

One would expect that with sharp gradient estimates, the variance of the updates should be small with a large enough sample. Here we show that the procedure stabilizes quickly as the estimates get closer to an optimum.

Theorem 33 (Control of update variance)**.** *Run Algorithm 2 as in Lemma 32, with arbitrary step-size* $\alpha_{(t)}$*. Then, for any* $t < T$*, taking expectation with respect to the sample* $\{\boldsymbol{z}_i\}_{i=1}^n$*, conditioned on* $\widehat{\boldsymbol{w}}_{(t)}$*, we have*

$$\mathbf{E}\, \|\widehat{\boldsymbol{w}}_{(t+1)} - \widehat{\boldsymbol{w}}_{(t)}\|^2 \leq 2\alpha_{(t)}^2 \left(\frac{32Cd \operatorname{trace}(\Sigma_{(t)})}{n} + \|\boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\|^2 \right).$$

In addition to these results, one can prove an improved version of Theorem 29 and Corollary 31, in a perfectly analogous fashion, using Lemma 32.

4.3.3 Uniform confidence

In our general sketch of 4.3.1, recall that we considered the case of high-probability events *pointwise* in the parameter $\boldsymbol{w}$ to be determined. That is, we showed that our algorithm of interest satisfied (4.7). Certainly, it is natural to ask whether the same algorithm can satisfy a *uniform* version of such a result, namely whether we can show

$$\mathbf{P} \left\{ \sup_{\boldsymbol{w} \in \mathcal{W}} \|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \leq \varepsilon \right\} \geq 1 - \delta \tag{4.14}$$

holds for some ε that depends on the sample size and task dimension. If such a result was available for the analysis in 4.3.2, the dependence on T in the estimation error terms (the $\log(T)$ factors) could be removed, and in the strongly convex case this would remove all tradeoffs, allowing the algorithm to freely run until convergence regardless of the sample size.

Obtaining such a result is more involved than the pointwise case. For our Algorithm 2 in 4.2.2, we make use of an even function ρ with a derivative ρ' that is increasing on $\mathbb{R}_+$, though the slope of that derivative monotonically tapers off in the limit, i.e., $\rho''(u) \to 0$ as $u \to \pm\infty$.

The main potential problem can be illustrated as follows. If we place no restrictions on $\mathcal{W} \subseteq \mathbb{R}^d$ or μ, then for arbitrarily large constant $A > 0$, we can always find a "bad" $\boldsymbol{w} \in \mathcal{W}$ such that $\|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - l'(\boldsymbol{w}; \boldsymbol{z}_i)\| > A$ for a fixed fraction of the indices $i \in [n]$, with probability over 50%. This gets in the way of a high-probability uniform bound on $\sup_{\boldsymbol{w} \in \mathcal{W}} \|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\|$.

To deal with this, we must ensure the scaling parameters $s_j > 0$ used in LOCATE are large enough that for almost all samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, and each choice of $\boldsymbol{w} \in \mathcal{W}$, we have

$$|\widehat{\theta}_j(\boldsymbol{w}) - l'_j(\boldsymbol{w}; \boldsymbol{z}_i)| \leq s_j, \quad j \in [d]$$

for *at least* one $i \in [n]$. Here we have made the dependence of $\widehat{\theta}_j$ on $\boldsymbol{w}$ explicit for clarity. Note that we can still deal with an arbitrarily large fraction of errant data. As long as a certain fraction, however small, of the data is within the limits set by s_j, Algorithm 2 looks to behave as we would hope. This analysis is provided in the following section, which can be skipped by the reader not interested in chiefly results of a chiefly technical nature. For reference, we note that under this additional assumption, it follows that

$$\sup_{\boldsymbol{w} \in \mathcal{W}} \|\widehat{\boldsymbol{\theta}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \lesssim O\left(\frac{d^{3/2} \log(d\delta^{-1})\Delta}{\sqrt{n}}\right) + O_P(1) \tag{4.15}$$

with probability no less than $1 - \delta$. This result is given in Theorem 42. Here $\lesssim$ suppresses terms of $O(1/n)$ order, Δ is the diameter of $\mathcal{W}$, and $O_P(1)$ is stochastic asymptotic notation for a term bounded in probability, which can ordinarily be taken small. Further investigation of related algorithms and their uniform estimation error is interesting from both technical and conceptual standpoints.

Technical preparation Our generic data shall be denoted by $\boldsymbol{z} \in \mathcal{Z}$. Let μ denote a probability measure on $\mathcal{Z}$, equipped with an appropriate σ-field. Data samples shall be assumed independent and identically distributed (iid), written $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. We shall work with loss function $l : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}_+$ throughout, with $l(\cdot; \boldsymbol{z})$ assumed differentiable for each $\boldsymbol{z} \in \mathcal{Z}$. Write $\mathbf{P}$ for a generic probability measure, most commonly the product measure induced by the sample. Let $f : \mathcal{Z} \to \mathbb{R}$ be an measurable function. Expectation is written $\mathbf{E}_\mu f(\boldsymbol{z}) := \int f \, d\mu$, with variance $\operatorname{var}_\mu f(\boldsymbol{z})$ defined analogously.

Denote the $\mathcal{L}_p$ norms ($1 \leq p < \infty$) under μ by $\|f\|_p = (\mathbf{E}_\mu |f(\boldsymbol{z})|^p)^{1/p}$. For the $\mathcal{L}_\infty$ norm taken over some $Z \subseteq \mathcal{Z}$, write $\|f\|_Z := \sup_{\boldsymbol{z} \in Z} |f(\boldsymbol{z})|$. With these norms we specify the usual function spaces $\mathcal{L}_p(\mu) = \{f : \|f\|_p < \infty\}$. We shall also make use of metrics on subsets of these spaces, denoted by $d_p(f, g) := \|f - g\|_p$ and $d_\infty(f, g) := \|f - g\|_\infty$. For d-dimensional Euclidean space $\mathbb{R}^d$, the standard (ℓ_2) norm shall be denoted $\|\cdot\|$ unless otherwise specified. For function F on $\mathbb{R}^d$ with partial derivatives defined, write the gradient as $F'(\boldsymbol{u}) := (F'_1(\boldsymbol{u}), \ldots, F'_d(\boldsymbol{u}))$ where for short, we write $F'_j(\boldsymbol{u}) := \partial F(\boldsymbol{u}) / \partial u_j$. In addition to asymptotic notation O and O_P [38], we use $\lesssim$ to suppress terms which are not of leading order. For integer k, write $[k] := \{1, \ldots, k\}$ for all the positive integers from 1 to k. Risk shall be denoted $R(\boldsymbol{w}) := \mathbf{E}_\mu l(\boldsymbol{w}; \boldsymbol{z})$, and its gradient $\boldsymbol{g}(\boldsymbol{w}) := R'(\boldsymbol{w})$.

Smoothness and convexity of functions shall also be utilized. For convex function F on convex set $\mathcal{W}$, say that F is λ-*Lipschitz* if, for all $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$ we have $|F(\boldsymbol{w}_1) - F(\boldsymbol{w}_2)| \leq \lambda \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|$. We say that F is λ-*smooth* if F' is λ-Lipschitz. Finally, F is *strongly convex* with parameter $\kappa > 0$ if for all $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$,

$$F(\boldsymbol{w}_1) - F(\boldsymbol{w}_2) \geq \langle F'(\boldsymbol{w}_2), \boldsymbol{w}_1 - \boldsymbol{w}_2 \rangle + \kappa \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|^2$$

for any norm $\|\cdot\|$ on $\mathcal{W}$, though we shall be assuming $\mathcal{W} \subseteq \mathbb{R}^d$. If there exists $\boldsymbol{w}^* \in \mathcal{W}$ such that $F'(\boldsymbol{w}^*) = 0$, then it follows that $\boldsymbol{w}^*$ is the unique minimum of F on $\mathcal{W}$.

Regarding numerical constants, there are numerous constant coefficients which appear throughout different results, and to keep each result presentable, we shall re-use characters for representing these constants, and they should not be taken to be common across distinct results unless otherwise noted. Precise values of these constants are discussed in the proofs.

Desired error estimates We assume that we can differentiate under the integral sign in each coordinate [3, 37], namely that

$$\boldsymbol{g}(\boldsymbol{w}) = \left(\mathbf{E}_{\mu} \frac{\partial l(\boldsymbol{w};\boldsymbol{z})}{\partial w_1}, \ldots, \mathbf{E}_{\mu} \frac{\partial l(\boldsymbol{w};\boldsymbol{z})}{\partial w_d} \right). \tag{4.16}$$

This ensures that a good estimate of $\mathbf{E}_{\mu} l'(\boldsymbol{w};\boldsymbol{z})$ will imply a good estimate of $\boldsymbol{g}$. In this chapter, we take the naive approach of constructing robust estimates in a by-coordinate fashion. Denote the jth component of $\widehat{\boldsymbol{g}}(\boldsymbol{w})$ and $\boldsymbol{g}(\boldsymbol{w})$ by $\widehat{g}_j$ and g_j respectively. Note that if

$$\mathbf{P}\left\{ |\widehat{g}_j - g_j|_{\mathcal{W}} \leq \varepsilon \right\} \geq 1 - \delta/d$$

for arbitrary $j \in [d]$, an application of a union bound yields

$$\mathbf{P}\left\{ \|\widehat{\boldsymbol{g}} - \boldsymbol{g}\|_{\mathcal{W}} \leq \sqrt{d}\varepsilon \right\} \geq 1 - \delta. \tag{4.17}$$

Our objective, then, must be to control $|\widehat{g}_j(\cdot) - g_j(\cdot)|$ in the $\mathcal{L}_\infty$ norm on $\mathcal{W}$, a random quantity depending on the sample.

A useful class of estimators We shall leverage a special type of M-estimator here, built using the following convenient class of functions.

Definition 34 (Function class for location estimates)**.** Let $\rho : \mathbb{R} \to [0, \infty)$ be an even function ($\rho(u) = \rho(-u)$) with $\rho(0) = 0$ and the following properties. Denote $\psi(u) := \rho'(u)$.

1. $\rho(u) = O(u)$ as $u \to \pm\infty$.
2. $\rho(u)/(u^2/2) \to 1$ as $u \to 0$.
3. $\psi' > 0$, and for some $C > 0$, and all $u \in \mathbb{R}$,

$$-\log(1 - u + Cu^2) \leq \psi(u) \leq \log(1 + u + Cu^2).$$

Of particular importance in the proceeding analysis is the fact that $\psi = \rho'$ is bounded, monotonically increasing and Lipschitz on $\mathbb{R}$, plus the upper/lower bounds which let us generalize the technique of Catoni [9].

Example 35 (Valid ρ choices). In addition to the Gudermannian function (section 4.2.2 footnote), functions such as $2(\sqrt{1 + u^2/2} - 1)$ and $\log\cosh(u)$ are well-known examples that satisfy the desired criteria. Note that the wide/narrow functions of Catoni do not meet all these criteria, nor does the classic Huber function. See Appendix A.1 for more.

For random variable $x \sim \nu$, and iid sample $x_1, \ldots, x_n \in \mathbb{R}$, define

$$\theta^* := \arg\min_{\theta} \mathbf{E}_{\nu} \rho_s(x - \theta) \tag{4.18}$$

$$\widehat{\theta} := \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \rho_s(x_i - \theta) \tag{4.19}$$

where $\rho_s(u) := \rho(u/s)$, and $s > 0$ is a scaling parameter. Note that θ^* satisfies $\mathbf{E}_{\nu} \psi_s(x - \theta^*) = 0$ and analogously for $\widehat{\theta}$ under the empirical measure, thus root-finding and minimizing are equivalent here. Both θ^* and $\widehat{\theta}$ are concentrated around $\mathbf{E}_{\nu} x$. This was shown in Lemma 25, and we give a more general statement here.

Lemma 36. *Given an n-sized sample from $x \sim \nu$, and assuming $\mathbf{E}_\nu x^2 < \infty$, we have*

$$\mathbf{P}\left\{\frac{1}{2}|\widehat{\theta} - \mathbf{E}_\nu x| > \frac{C \operatorname{var}_\nu x}{s} + \frac{s\log(\delta^{-1})}{n}\right\} \leq 2\delta$$

whenever $1/4 \geq C^2(\operatorname{var}_\nu x)/s^2 + C\log(\delta^{-1})/n$. Furthermore, we have that

$$|\theta^* - \mathbf{E}_\nu x| \leq \frac{2C \operatorname{var}_\nu x}{s}.$$

For cleaner notation in the following results, we shall proceed without loss of generality taking $C = 1/2$.

Uniform control In what follows, we spend some time developing more general results of a technical nature. Doing so yields conditions on the scale s and distribution μ under which uniform control of the estimation error $\|\widehat{\boldsymbol{g}} - \boldsymbol{g}\|$ is possible. Let $\mathcal{F} \subseteq \mathcal{L}_2(\mu)$ be a general class of functions. To get started, denote

$$\theta_f := \arg\min_\theta \mathbf{E}_\mu (f(\boldsymbol{z}) - \theta)^2 = \mathbf{E}_\mu f(\boldsymbol{z}) \tag{4.20}$$

$$\theta_f^* := \arg\min_\theta \mathbf{E}_\mu \rho_s\left(f(\boldsymbol{z}) - \theta\right) \tag{4.21}$$

$$\widehat{\theta}_f := \arg\min_\theta \frac{1}{n}\sum_{i=1}^n \rho_s\left(f(\boldsymbol{z}_i) - \theta\right) \tag{4.22}$$

for each $f \in \mathcal{F}$. Note the dependence of θ^* and $\widehat{\theta}$ on s, though this is not made explicit in the notation. We will need to restrict $\mathcal{F}$ and μ to some degree, and this can be done with mild bounds on the low-order moments. More precisely, we shall require that there exist $v < \infty$ such that

$$\operatorname{var}_\mu f(\boldsymbol{z}) \leq v, \quad \forall f \in \mathcal{F}. \tag{4.23}$$

By design, $\rho_s(\cdot)$ closely approximates $(\cdot)^2/2$ as $s \to \infty$, which suggests that $\theta_f^* \approx \theta_f$ should be sharp irrespective of f, for a sufficiently regular function class. By Lemma 36,

$$(4.23) \implies |\theta^* - \theta|_{\mathcal{F}} \leq \frac{v}{s}. \tag{4.24}$$

Let us introduce some additional notions. The empirical process $\{X_f(\theta) : f \in \mathcal{F}\}$ with random variables defined by

$$X_f(\theta) := s\left(\frac{1}{n}\sum_{i=1}^n \psi_s\left(f(\boldsymbol{z}_i) - \theta\right) - \mathbf{E}_\mu \psi_s\left(f(\boldsymbol{z}) - \theta\right)\right), \quad \theta \in \mathbb{R}$$

is an object of general technical interest, and for *pre-fixed* θ, bounds on the increments of this process were obtained by Brownlees et al. [7]. Our focus is on a closely related new setting, in which the θ value is not pre-fixed, but also varies with f, though in a specific way. In particular, our focus is the new process $\{X_f^* : f \in \mathcal{F}\}$ with

$$X_f^* := X_f(\theta_f^*) = \frac{s}{n}\sum_{i=1}^n \psi_s\left(f(\boldsymbol{z}_i) - \theta_f^*\right).$$

We are interested in controlling $|X^*|_{\mathcal{F}} = \sup_{f\in\mathcal{F}} |X^*_f|$, which can be done up to finite precision as follows. Fixing any $f_0 \in \mathcal{F}$, we have

$$|X^*|_{\mathcal{F}} \leq |X^* - X^*_{f_0}|_{\mathcal{F}} + |X^*_{f_0}|. \tag{4.25}$$

On the event $\{|X^*|_{\mathcal{F}} \leq \varepsilon\}$, one expects that $\widehat{\theta}_f$ and θ^*_f will tend to be close for any choice of f, under proper scaling. The following result makes this explicit.

Lemma 37 (Control of uniform estimation error)**.** *Fix arbitrary $\varepsilon > 0$, and condition on $\{|X^*|_{\mathcal{F}} \leq \varepsilon\}$. Under appropriate scaling as specified by event $\mathcal{E}$ in (4.46), there exists a constant $c > 0$ such that*

$$|\widehat{\theta} - \theta^*|_{\mathcal{F}} \leq c\varepsilon.$$

Under the conditions of Lemma 37 then, using Lemma 36 and (4.23), we have with probability no less than $\mathbf{P}\{|X^*|_{\mathcal{F}} \leq \varepsilon\}$ that

$$|\widehat{\theta} - \theta|_{\mathcal{F}} \leq |\widehat{\theta} - \theta^*|_{\mathcal{F}} + |\theta^* - \theta|_{\mathcal{F}} \leq c\varepsilon + v/s \tag{4.26}$$

where $s > 0$ is noted to control the bias induced by using ρ_s instead of the squared error.

It only remains to control $|X^* - X^*_{f_0}|_{\mathcal{F}}$. Using the (deterministic) M-estimate θ^*_f, we introduce a metric d^*, defined by

$$d^*(f,g) \coloneqq |\theta^*_f - \theta^*_g|, \quad f,g \in \mathcal{F}. \tag{4.27}$$

By the properties of ρ (Defn. 34), for any $f \in \mathcal{F}$, the value θ^*_f is uniquely determined, and thus $f = g \implies d^*(f,g) = 0$. Furthermore, $d^*(f,h) \leq d^*(f,g) + d^*(g,h)$ for any $f,g,h \in \mathcal{F}$, and d^* is symmetric in its arguments. Strictly speaking, however, d^* is a quasi-metric, due to the fact that $d^*(f,g) = 0$ need not imply $f = g$. To see this, say for example that $f(\boldsymbol{z})$ has a symmetric distribution about 0. Let $g(\boldsymbol{z}) \coloneqq kf(\boldsymbol{z})$ for some $k \neq 0$. Then $\theta^*_g = k\,\mathbf{E}_\mu f(\boldsymbol{z}) = 0 = \theta^*_f$, but $f \neq g$. In any case, it has the properties we require. In particular, this metric is used to develop a Bernstein-type inequality for the increments of X^*_f.

Lemma 38 (Increment tails)**.** *For any $f,g \in \mathcal{F}$, we have*

$$\mathbf{P}\left\{|X^*_f - X^*_g| > t\right\} \leq 2\exp\left(\frac{-nt^2}{2\left(V(f,g) + tb(f,g)/3\right)}\right) \tag{4.28}$$

where $V(f,g) \coloneqq 2(d_2(f,g)^2 + d^*(f,g)^2)$ and $b(f,g) \coloneqq d_\infty(f,g) + d^*(f,g)$.

This inequality allows us access to metric entropy estimates, via a chaining argument in a form pioneered by Talagrand [36]. To give such a result concisely, some additional notation is required. Let $\Delta(\mathcal{F};D)$ denote the diameter of $\mathcal{F}$ in arbitrary metric D. We use $N(\varepsilon,\mathcal{F},D)$ to denote the covering number of $\mathcal{F}$ at radius ε in metric D [39, Section 2.1], and denote the covering integral by

$$E_\beta(D) \coloneqq \int_0^{\Delta(\mathcal{F};D)} \left(\log N(\varepsilon,\mathcal{F},D)\right)^{1/\beta}\, d\varepsilon \tag{4.29}$$

where dependence on $\mathcal{F}$ is suppressed in the notation.

Theorem 39 (Entropy-based bounds)**.** *Fixing any $f_0 \in \mathcal{F}$ and $\delta \in (0,1)$, there exists a positive constant c such that*

$$|X^* - X^*_{f_0}|_{\mathcal{F}} \leq c\log(\delta^{-1})\left(\frac{E_1(d_\infty) + E_1(d^*)}{n} + \frac{E_2(d_2) + E_2(d^*)}{\sqrt{n}}\right)$$

with probability no less than $1 - \delta$.

Returning to the risk With these general results in hand, let us return to the context of approximate risk minimization. Here the general function class $\mathcal{F}$ corresponds to

$$\mathcal{F} = \{l_j'(\boldsymbol{w}; \cdot) : \boldsymbol{w} \in \mathcal{W}\}, j \in [d].$$

This is a class of functions on $\mathcal{Z}$, indexed by the parameter space $\mathcal{W}$. Using this new index, for θ_f we write $\theta_{\boldsymbol{w}} = \mathbf{E}_{\mu} \, l_j'(\boldsymbol{w}; \cdot)$, and so forth for $\widehat{\theta}_{\boldsymbol{w}}$ and $\theta_{\boldsymbol{w}}^*$. The estimate $\widehat{\theta}_{\boldsymbol{w}}$ is used as our gradient $\widehat{g}_j$, and corresponds directly to the LOCATE step (4.4) given in our overview of the algorithm. To evaluate the $E_\beta(\cdot)$ terms in Theorem 39, note that the d_∞ on $\mathcal{F}$ in our setting is

$$d_\infty(\boldsymbol{w}_1, \boldsymbol{w}_2) = \sup_{\boldsymbol{z}} |l_j'(\boldsymbol{w}_1; \boldsymbol{z}) - l_j'(\boldsymbol{w}_2; \boldsymbol{z})|,$$

noting that d_∞ is a function on $\mathcal{F} \times \mathcal{F}$, despite the notation. Analogous statements hold for d_2 and $d^*(\boldsymbol{w}_1, \boldsymbol{w}_2) = |\theta_{\boldsymbol{w}_1}^* - \theta_{\boldsymbol{w}_2}^*|$. Assuming $l(\cdot; \boldsymbol{z})$ is smooth in its parameters, we have for any $\boldsymbol{z}$ that there exists a $c(\boldsymbol{z}) > 0$ such that for all $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$,

$$|l_j'(\boldsymbol{w}_1; \boldsymbol{z}) - l_j'(\boldsymbol{w}_2; \boldsymbol{z})| \leq c(\boldsymbol{z}) \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|.$$

Note that if $\sup_{\boldsymbol{z}} c(\boldsymbol{z}) \leq c < \infty$, it follows that

$$d_\infty(\boldsymbol{w}_1, \boldsymbol{w}_2) \leq c \|\boldsymbol{w}_1 - \boldsymbol{w}_2\| \tag{4.30}$$

and similarly if $\mathbf{E}_{\mu} \, c(\boldsymbol{z}) \leq c < \infty$, we have $d_2(\boldsymbol{w}_1, \boldsymbol{w}_2) \leq c\|\boldsymbol{w}_1, \boldsymbol{w}_2\|$, and d^* can be controlled by both of these distances. Such assumptions hold under very weak assumptions, as the following example shows.

Example 40 (Uniform Lipschitz property). Consider the agnostic regression task, where $\boldsymbol{z} = (y, \boldsymbol{x})$, and our function approximation is done with a linear model. Under quadratic loss, we have $l'(\boldsymbol{w}; \boldsymbol{z}) = -(y - \langle \boldsymbol{w}, \boldsymbol{x} \rangle)\boldsymbol{x}$, and thus in the Euclidean norm we have

$$\|\boldsymbol{x}(\langle \boldsymbol{w}_2 - \boldsymbol{w}_1, \boldsymbol{x} \rangle)\| \leq \|\boldsymbol{x}\|^2 \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|.$$

If $\|\boldsymbol{x}\|$ is bounded by M, then (4.30) is implied, and holds with $c \leq M^2$. It should be remarked that this holds without any assumptions on y. That is, the response (and consequently the noise distribution) need not be bounded or even sub-Gaussian.

For any metric space $\mathcal{F}$ with arbitrary metrics D_1 and D_2, if $D_1 \leq D_2$, then any ε-cover in metric D_2 is an ε-cover in metric D_1. It follows that $N(\varepsilon, \mathcal{F}, D_1) \leq N(\varepsilon, \mathcal{F}, D_2)$. In our setting, this implies that if $\mathcal{W}$ is contained in the unit ball in Euclidean metric $\|\cdot\|$, this amounts to a c-radius ball in the scaled metric of (4.30), and thus we have

$$N(\varepsilon, \mathcal{F}, d_\infty) \leq N(\varepsilon, \mathcal{W}, c\|\cdot\|) \leq \left(\frac{3c}{2\varepsilon}\right)^d \tag{4.31}$$

for all $\varepsilon \in (0, c)$, where the latter inequality is a basic property of d-dimensional Euclidean space [21]. One gets clean results for this case, as follows.

Corollary 41. *Let $\mathcal{F} = \{l_j'(\boldsymbol{w}; \cdot) : \boldsymbol{w} \in \mathcal{W}\}$, writing $\Delta := \Delta(\mathcal{W}; \|\cdot\|)$, and assume (4.30) holds for $\mathcal{W} \subset \mathbb{R}^d$. Then there exist positive constants a and c such that*

$$|X^* - X_{f_0}^*|_{\mathcal{F}} \leq cd \log(\delta^{-1}) \left(\frac{a\Delta^2 + \Delta \log(a)}{n} + \frac{\Delta}{\sqrt{nd}} \left(\sqrt{\log(a)} + \frac{a\sqrt{\pi}}{2} \right) \right)$$

with probability no less than $1 - \delta$.

Let us now connect all the steps to illustrate the property of interest. On the underlying learning task, we assume that $\Delta(\mathcal{W}; \|\cdot\|) < \infty$ and l' satisfies (4.30), without loss of generality assuming $c = 1$. On the learning procedure, we run Algorithm 2, with scaling $s = O(\sqrt{vn})$ such that the premise of Lemma 37 holds almost surely.

Theorem 42. *Under the above assumptions, the estimates of Algorithm 2 satisfy*

$$\sup_{\boldsymbol{w} \in \mathcal{W}} \|\widehat{\boldsymbol{\theta}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \lesssim d^{3/2} \log(d\delta^{-1}) O\left(\frac{\Delta}{\sqrt{n}}\right) + O_P(1)$$

with probability no less than $1 - \delta$.

We thus have high-probability bounds on the accuracy, up to an additive term bounded in probability. This extra term arises as a result of our analysis technique, and sharper bounds may be available using a different tactic. In any case, this result can be leveraged for any convex risk to prove risk bounds in the vein of Theorem 29 and Corollary 31, though convergence to the minimum risk naturally would only be guaranteed when the $O_P(1)$ term can be taken arbitrarily small.

4.4 Empirical analysis

The chief goal of our experiments is to elucidate the relationship between factors of the learning task (e.g., sample size, model dimension, initial value, underlying data distribution) and the behaviour of the robust gradient procedure proposed in Algorithm 2. We are interested in how these factors influence performance, both in an absolute sense and relative to the key competitors cited in section 4.1.

We have carried out two classes of experiments. The first considers a concrete risk minimization task given noisy function observations, and takes an in-depth look at how each experimental factor influences algorithm behaviour, in particular the trajectory of performance over time (as we iterate). The latter is an application of the proposed algorithm to the corresponding regression task under a large variety of data distributions, meant to rigorously evaluate the practical utility and robustness in an agnostic learning setting.

4.4.1 Noisy convex minimization

Experimental setup In this experiment, we construct a risk function taking a canonical quadratic form, setting $R(\boldsymbol{w}) = \langle \Sigma \boldsymbol{w}, \boldsymbol{w} \rangle / 2 + \langle \boldsymbol{w}, \boldsymbol{u} \rangle + c$, for pre-fixed constants $\Sigma \in \mathbb{R}^{d \times d}$, $\boldsymbol{u} \in \mathbb{R}^d$, and $c \in \mathbb{R}$. The task is to minimize $R(\cdot)$ without knowledge of R itself, but rather only access to n random function observations $r_1, \ldots, r_n$. These $r : \mathbb{R}^d \to \mathbb{R}$ are generated independently from a common distribution, satisfying the property $\mathbf{E}\, r(\boldsymbol{w}) = R(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathbb{R}^d$. In particular, here we generate observations $r_i(\boldsymbol{w}) = (\langle \boldsymbol{w}^* - \boldsymbol{w}, \boldsymbol{x}_i \rangle + \epsilon_i)^2/2$, $i \in [n]$, with $\boldsymbol{x}$ and ϵ independent of each other. Here $\boldsymbol{w}^*$ denotes the minimum, and we have that $\Sigma = \mathbf{E}\, \boldsymbol{x}\boldsymbol{x}^T$. The inputs $\boldsymbol{x}$ shall follow an isotropic d-dimensional Gaussian distribution thoughout all the following experiments, meaning Σ is positive definite, and R is strongly convex.

For the bulk of these tests, we run three procedures. First is ideal gradient descent, denoted `gd`, which assumes the objective function R known. This corresponds to (4.2). Second, as a standard approximate procedure (4.3), we use the ERM-GD routine, denoted `erm` and defined in (4.1), which approximates the optimal procedure using the empirical risk. Against these two benchmarks, we compare our Algorithm 2, denoted `rgd`, as a robust alternative for (4.3). In addition, we also look at the technique of Brownlees et al. [7], denoted `bjl`, using a general-purpose first-order optimizer, and directly compare it with `rgd`.

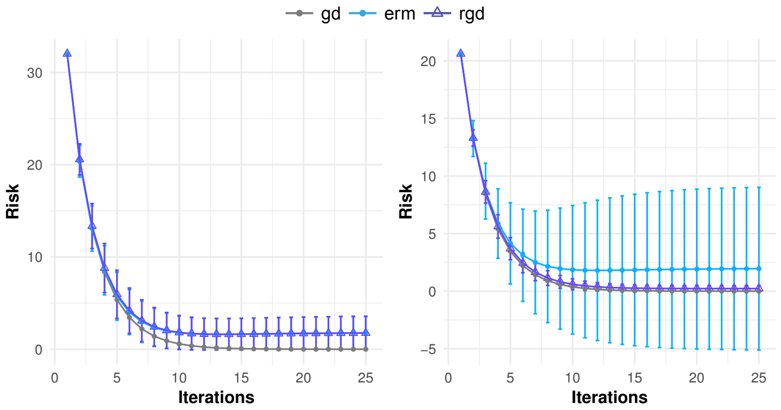

Figure 4.2: Risk as a function of iterations after a common initial point. Each trial corresponds to a new random sample, and all values on the vertical axis are averaged over 250 trials. Error bars are standard deviation over trials. Left is Normal noise, right is log-Normal noise. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

Several experimental parameters are systematically modified to explore their impact on algorithm performance. The distribution generating the data $\{r_i\}_{i=1}^{n}$, sample size n, task dimension d, and the initialization of $\widehat{\boldsymbol{w}}_{(0)}$ are key parameters of interest. Results for several settings are given below.

Results: light/heavy-tailed samples We begin with the simplest and most important question: are there natural learning settings in which `rgd` outperforms ERM-GD? Also, how does it fare in situations where ERM is optimal? Under Gaussian noise, ERM-GD is effectively optimal [25, Appendix C]. We thus consider the case of Gaussian noise (mean 0, standard deviation 20) as a baseline, and use centred log-Normal noise (log-location 0, log-scale 1.75) as an archetype of asymmetric heavy-tailed data. We have set $\boldsymbol{w}^* = (1, 1)$ and initialize as $\boldsymbol{w}^*_{(0)} = \boldsymbol{w}^* + (\text{Unif}[-5, 5], \text{Unif}[-5, 5])$. Risk results are given in Figures A.18–A.19, with corresponding sample error results in Figure A.20.

Several observations can be made immediately. In the situation favorable to `erm`, differences in performance are basically negligible. On the other hand, in the heavy-tailed setting, the performance of `rgd` is superior in terms of quality of the solution found and the variance of the estimates. Furthermore, we see that at least in the situation of small d and large n, taking T beyond numerical convergence has minimal negative effect on `rgd` performance; on the other hand `erm` is more sensitive. Comparing true risk with sample error, we see that while there is some unavoidable overfitting, in the heavy-tailed setting `rgd` is slower in the rate at which it departs from the ideal routine.

Results: impact of initialization At this point, we still have little more than a proof of concept, with rather arbitrary choices of n, d, noise distribution, and initialization method. We proceed to investigate how each of these experimental parameters impacts performance, starting with initialization. Given a fixed sample size, how does the quality of the initial guess impact estimates in the noisy convex minimization task? We consider three initializations of the form $\boldsymbol{w}^* + \text{Unif}[-\boldsymbol{\Delta}, \boldsymbol{\Delta}]$, with $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_d)$, values ranging over $\Delta_j \in \{0.5, 5, 10\}$, $j \in [d]$, where larger Δ_j naturally correspond to potentially worse initialization. Figure A.21 displays the risk achieved by the two competing techniques under the same settings, with log-Normal noise.

Some interesting observations can be made here. That `rgd` achieves a better solution on average in all settings is immediate. As well, its degradation in terms of departure from the

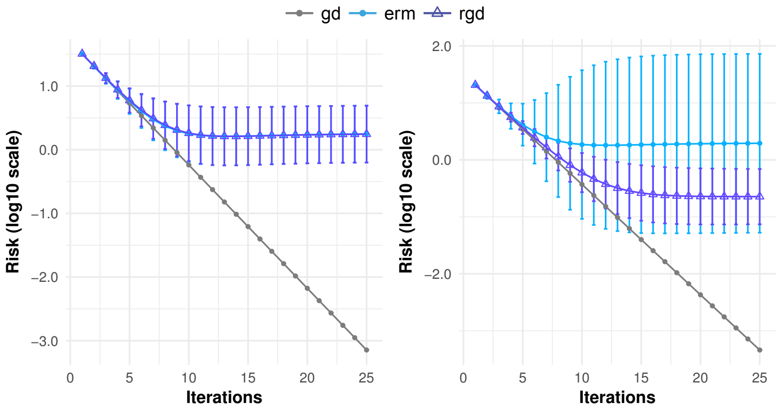

Figure 4.3: Results of Figure A.18, given in $\log_{10}$ scale. Error bars, say $\log_{10}(u) \pm \epsilon$, are computed as relative error, namely $\epsilon = \mathrm{sd}(u)/(u \log(10))$.

ideal routine (gd here) is much slower than erm, as we would expect from Lemmas 26 and 30. Finally, here we see impact of taking T large for both methods. When the initial value is very good ($\Delta_j = 0.5$ case), the poor estimates of the ERM-based routine only make the solution worse on average, while rgd gets no worse, and even can improve on this solution if stopped early enough. This of course suggests that the tradeoff induced by T in our estimation error bounds is not superfluous. Considering the discussion of section 4.3.3, it may be that Algorithm 2 as in section 4.2.2 satisfies (4.7), but not (4.14).

Results: impact of distribution The process generating the sample, namely the underlying data distribution, plays an equally critical role in algorithm performance. In the noisy convex minimization task here, we can observe performance changes that occur as we modify the data distribution, all while keeping the exact same risk function. This means the underlying optimization problem is the same, and all that changes is the statistical estimation side, highlighting the divergence between optimal oracle-based procedures and approximate procedures under less congenial data. Here we run the two algorithms of interest from common initial values as in the first experimental setting, and measure performance changes as the noise distribution is modified. We consider six situations, three for Normal noise, three for log-Normal noise. The location and scale parameters for the former are respectively $(0, 0, 0), (1, 20, 34)$; the log-location and log-scale parameters for the latter are respectively $(0, 0, 0), (1.25, 1.75, 1.9)$. Results are given in Figure A.22.

Looking first at the Normal case, where we expect ERM-based methods to perform well, we see that rgd is able to match erm in all settings, and even tends to out-perform it in the high-variance setting. In the log-Normal case, as our previous example suggested, the performance of erm degrades rather dramatically, and a clear gap in performance appears. This flexibility of rgd in dealing with both symmetric and asymmetric noise, both exponential and heavy tails, is indicative of the robustness suggested by the weak conditions of section 4.3.2. In addition, it suggests that our simple dispersion-based technique ($\widehat{\sigma}_j$ settings in 4.2.2) provides tolerable accuracy, implying a small enough c_0 factor, and reinforcing the insights from the proof of concept case in Figures A.18–A.20.

Results: impact of sample size Since the true risk is unknown, the size and quality of the sample $\{\boldsymbol{z}_i\}$ is critical to the output of all learners. To evaluate learning efficiency, we examine how performance depends on the available sample size, with dimension and all algorithm

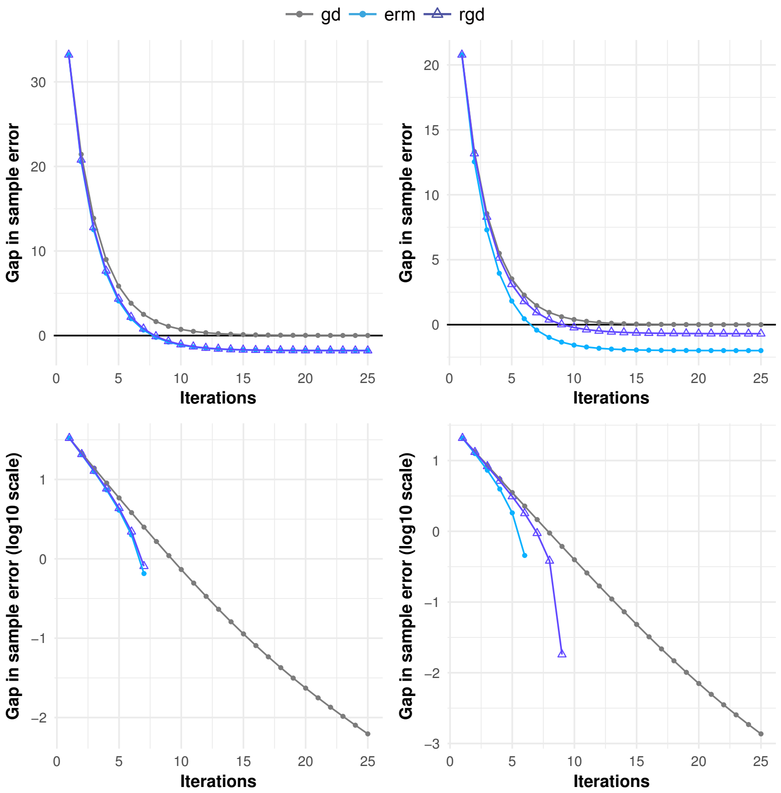

Figure 4.4: Sample error gap for the tests shown in Figure A.18, computed as $\widehat{R}(\widehat{\boldsymbol{w}}) - \widehat{R}(\boldsymbol{w}^{*})$, with $\widehat{R}(\boldsymbol{w}) = n^{-1}\sum_{i=1}^{n} l(\boldsymbol{w}; \boldsymbol{z}_i)$. First row is original coordinates, second row is $\log_{10}$ scale. The latter graphs stop once the gap (before log transform) becomes negative. Left is Normal noise, right is log-Normal noise.

parameters fixed. Figure A.23 gives the accuracy of `erm` and `rgd` in tests analogous to those above, using common initial values across methods, and $n \in \{25, 50, 100, 200, 400, 800\}$.

Both algorithms naturally show monotonic performance improvements as the sample size grows, but the most salient feature of these figures is the speed with which the performance of `erm` saturates, and its degradation as T gets large in the small sample case. On the other hand, under the exact same settings, `rgd` shows essentially monotonic improvement over iterations. It is also evident that the robust GD approach realizes better performance than ERM-GD with less samples, implying better learning efficiency in the heavy-tailed setting.

Results: impact of dimension The role of dimension d, namely the number of parameters to be determined, plays a key role in practice and in theory, as seen in the error bounds of section 4.3.2. Fixing the sample size and all algorithm parameters as above, we investigate the relative difficulty each algorithm has as the dimension increases. Figure A.24 shows the risk of `erm` and `rgd` in tests just as above, with $d \in \{2, 4, 8, 16, 32, 64\}$.

Here we see that both algorithms degrade monotonically in the dimension, just as the optimal `gd` does. We see that `rgd` maintains superiority over all d settings. In contrast to `erm`, whose performance hits bottom very quickly in high dimensions, `rgd` continues to improve for more iterations, presumably due to updates which are close to that of the optimal (4.2).

Results: against implicit robust loss minimizers Here we compare the performance of the robust GD given in Algorithm 2 with the procedure analyzed by Brownlees et al. [7] (denoted `bjl` henceforth). The former algorithm was designed using the exact same principles

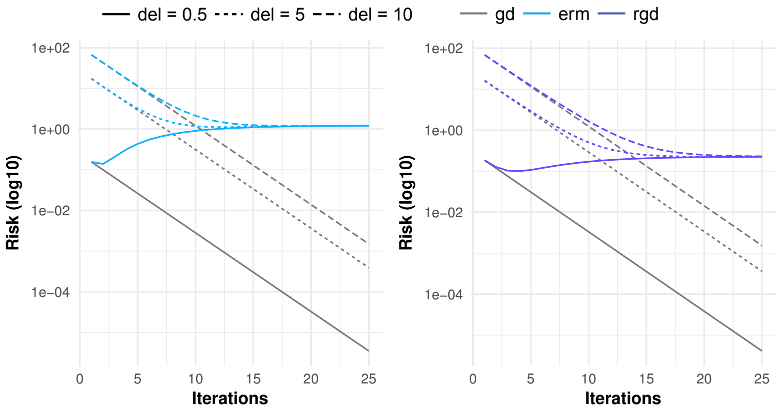

Figure 4.5: Risk as a function of iterations, in $\log_{10}$ scale. Values are averaged over 250 trials. The "del" refers to Δ_j. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

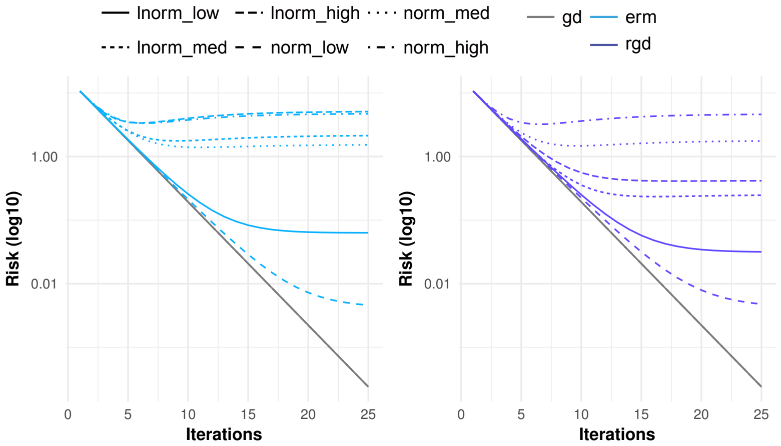

Figure 4.6: Risk as a function of iterations, in $\log_{10}$ scale. Values averaged over 250 trials. Here "lnorm" and "norm" denote log-Normal and Normal samples, with "low," "med," and "high" designating the three parameter settings. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

that makes their algorithm formally appealing, but with the merit of being far more computationally straightforward. Indeed, implementing their procedure as-is can turn a convex risk minimization into a non-convex minimization task, and requires general-purpose non-linear optimization. Such procedures can be sensitive to initial values, are at risk of getting stuck near sub-optimal local minima, and may require a large number of iterations to complete. To give a simple comparison between `bjl` and `rgd`, we run multiple trials of the same task, starting both routines at the same (random) initial value each time, generating a new sample, and repeating this process for $d \in \{2, 4, 8, 16, 32, 64\}$, as in the tests above.

To implement `bjl`, we use the conjugate gradient method of Fletcher and Reeves [15], as implemented in the `optim` function of the R `stats` library [33], using the recommended stopping conditions. This gives us a standard first-order technique for minimizing the `bjl` objective. To see how well our procedure can compete with a pre-fixed max iteration number, we set $T = 25$ for all settings. Computation time is computed using the `microbenchmark` R library, under which multiple runs of each trial are conducted internally; the median of these times is used as the representative time for each trial. Results are given in Figure A.25.

In the low-dimensional setting, the behaviour of both routines is similar, but as the number of parameters increases, `bjl` tends to deteriorate much faster, with a clear reduction in stability

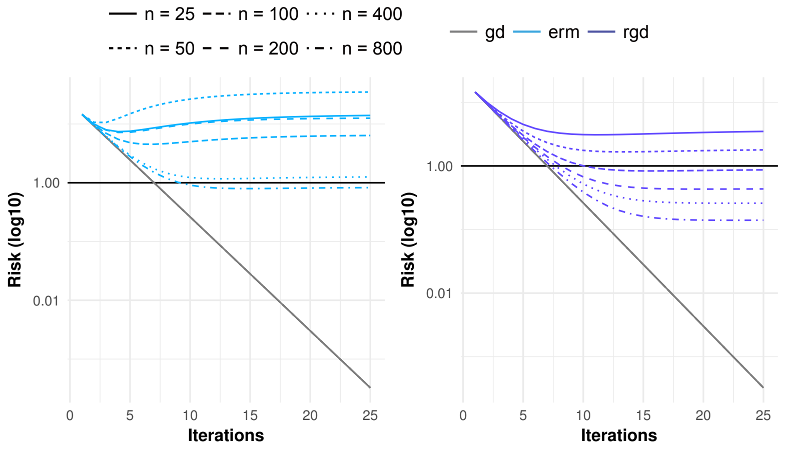

Figure 4.7: Risk as a function of iterations, in $\log_{10}$ scale. Values are averaged over 250 trials. The black horizontal rules are set for reference between the two plots, whose coordinates differ slightly. $d = 2, \alpha_{(t)} = 0.1$ for all t.

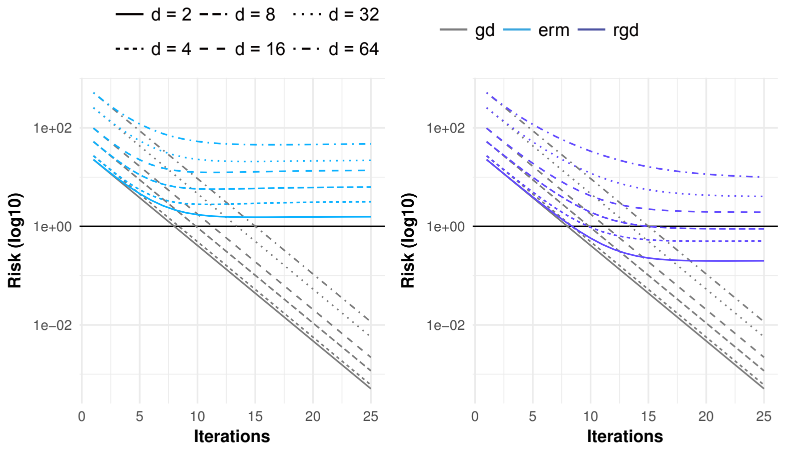

Figure 4.8: Risk as a function of iterations, in $\log_{10}$ scale. Values averaged over 250 trials. The black horizontal rules are set for reference between the two plots, whose coordinates differ slightly. $n = 500, \alpha_{(t)} = 0.1$ for all t.

(increased variance) and a large increase in computation time. On the other hand, even with very few iterations and a pre-fixed stopping rule, the simple `rgd` achieves smaller risk and greater stability. Let us remark that since this `rgd` is a stable prototype implementation in R, a "production" version can naturally be made much more efficient.

4.4.2 Regression task

Experimental setup In this experiment, we apply our algorithm to a general regression task, under a wide variety of data distributions, and compare its performance against standard regression algorithms, both classical and modern. For each experimental condition, and for each trial, we generate n training observations of the form $y_i = \boldsymbol{x}^T \boldsymbol{w}^* + \epsilon_i, i \in [n]$. Distinct experimental conditions are delimited by the setting of (n, d) and μ. Inputs $\boldsymbol{x}$ are assumed to follow a d-dimensional isotropic Gaussian distribution, and thus our setting of μ will be determined by the distribution of noise ϵ. In particular, we look at several families of distributions, and within each family look at 15 distinct *noise levels*. Each noise level is simply a particular parameter setting, designed such that $\mathrm{sd}_\mu(\epsilon)$ monotonically increases over the range 0.3–20.0,

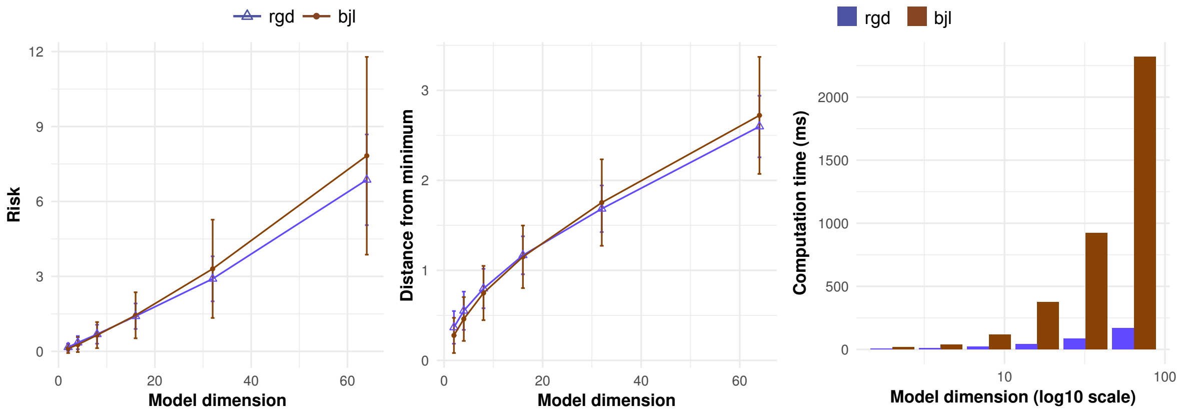

Figure 4.9: Performance as a function of the number of parameters to optimize. Left: excess risk. Centre: ℓ_2 distance from the minimum. Right: computation times. Performance values and computation times are averaged over 250 trials. $n = 500, \alpha_{(t)} = 0.1$ for all t.

approximately linearly over the levels (cf. Appendix A.5).

To ensure a variety of signal/noise ratios are captured, for each trial, $\boldsymbol{w}^* \in \mathbb{R}^d$ is randomly generated as follows. Defining the sequence $w_k := \pi/4 + (-1)^{k-1}(k-1)\pi/8, k = 1, 2, \ldots$ and uniformly sampling $i_1, \ldots, i_d \in [d_0]$ with $d_0 = 500$, we set $\boldsymbol{w}^* = (w_{i_1}, \ldots, w_{i_d})$. Computing $\mathrm{SN}_\mu = \|\boldsymbol{w}^*\|_2^2 / \operatorname{var}_\mu(\epsilon)$, we have $0.2 \leq SN_\mu \leq 1460.6$. Regarding the noise distribution families, the tests described above were run for 27 different families, but as space is limited, here we provide results for four representative families: log-logistic (denoted `llog` in figures), log-Normal (`lnorm`), Normal (`norm`), and symmetric triangular (`tri_s`). Even with just these four, we capture both bounded and unbounded sub-Gaussian noise, and heavy-tailed data both with and without finite higher-order moments.

In the figures here, we call the risk *prediction error*, computed as follows. For each condition and each trial, an independent test set of m observations is generated identically to the corresponding n-sized training set. All competing methods use common sample sets for training and are evaluated on the same test data, for all conditions/trials. For each method, in the kth trial, some estimate $\widehat{\boldsymbol{w}}(k)$ is determined. To approximate the ℓ_2-risk, compute root mean squared test error $e_k(\widehat{\boldsymbol{w}}) := (m^{-1} \sum_{i=1}^m (\widehat{\boldsymbol{w}}^T \boldsymbol{x}_{k,i} - y_{k,i})^2)^{1/2}$, and output prediction error as the average of normalized errors $e_k(\widehat{\boldsymbol{w}}(k)) - e_k(\boldsymbol{w}^*(k))$ taken over all K trials. While n values vary, in all experiments we fix $K = 250$ and test size $m = 1000$.

Finally, we summarize the competing methods used. Classical choices are ordinary least squares (ℓ_2-ERM, denoted `ols`) and least absolute deviations (ℓ_1-ERM, `lad`). We also look at two very recent methods of practical and theoretical importance described in section 4.1, namely the robust regression routines of Hsu and Sabato [17] (`hs`) and Minsker [28] (`geomed`). For the former, we used the source published online by the authors. For the latter, on each subset the `ols` solution is computed, and solutions are aggregated using the geometric median (in ℓ_2 norm), computed using the well-known algorithm of Vardi and Zhang [40, Eqn. 2.6], and the number of partitions is set to $\max(2, \lfloor n/(2d) \rfloor)$. For comparison to this, we also initialize `rgd` to the `ols` solution, with confidence $\delta = 0.005$, and $\alpha_{(t)} = 0.1$ for all iterations. Maximum number of iterations is $T \leq 100$; the routine finishes after hitting this maximum or when the absolute value of the gradient falls below 0.001. These settings are used across all trials of all subsequent experiments.

Results: impact of noise levels In Figure 4.10, we look at performance over noise settings, from negligible noise to significant noise with potentially infinite higher-order moments. We

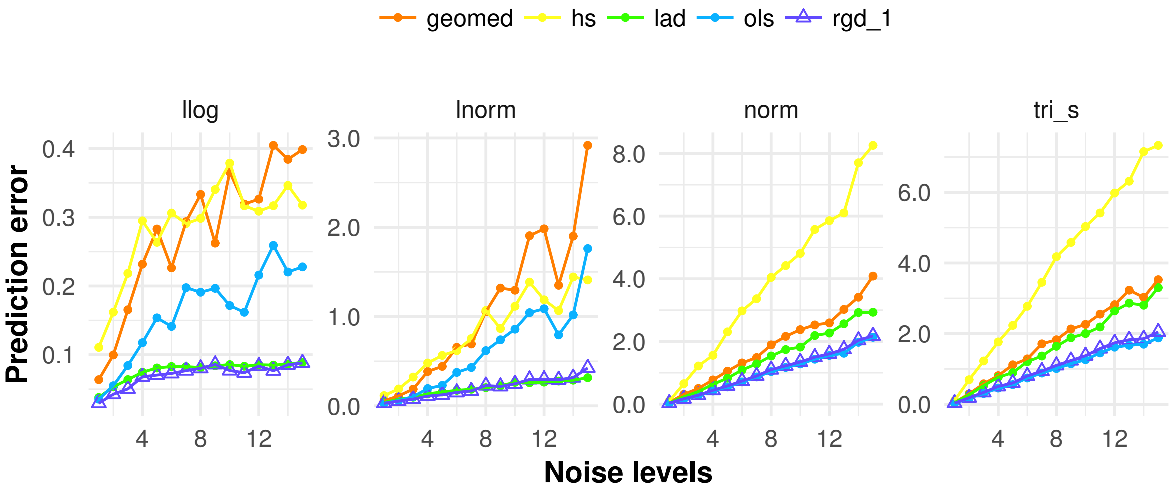

Figure 4.10: Prediction error over noise levels, for $n = 30, d = 5$.

see that `rgd` generalizes well, in a manner which is effectively uniform across the distinct noise families. We note that even in such diverse settings with pre-fixed step-size and iteration numbers, very robust performance is shown. It appears that under small sample size, `rgd` reduces the variance due to errant observations, while incurring a smaller bias than the other robust methods. When `ols` (effectively ERM-GD) is optimal, note that `rgd` follows it closely, with virtually negligible bias. When the former breaks down, `rgd` remains stable.

Results: impact of n and d First we fix the model dimension d, and evaluate performance as sample size n ranges from very small to quite large. For a fixed noise level, prediction error is displayed in Figure 4.11. We see that regardless of distribution, `rgd` effectively matches the optimal convergence of OLS in the `norm` and `tri_s` cases, and is resiliant to the remaining two scenarios where `ols` breaks down. There are clear issues with the median of means based methods at very small sample sizes, though the geometric median based method does eventually at least surpass OLS in the `llog` and `lnorm` cases. Essentially the same trends can be observed at all noise levels.

Next we fix the ratio n/d and look at the role played by increasingly large dimension. Prediction error as a function of d at a fixed noise level is given in Figure 4.12, and we see that for all distributions, the performance of `rgd` is essentially constant. This coincides with the theory of section 4.3.2, and our intuition since Algorithm 2 is run in a by-coordinate fashion. On the other hand, competing methods show sensitivity to the number of free parameters, especially in the case of asymmetric data with heavy tails.

4.5 Concluding remarks

In this work, we introduced and analyzed a learning algorithm which takes advantage of robust estimates of the unknown risk gradient, integrating statistical estimation and practical implementation into a single routine. Doing so allows us to deal with the statistical vulnerabilities of ERM-GD and partition-based methods, while circumventing computational issues posed by minimizers of robust surrogate objectives. The price to be paid is new computational overhead and potentially biased estimates. Is this price worth paying? Bounds on the excess risk are available under very weak assumptions on the data distribution, and we find empirically that

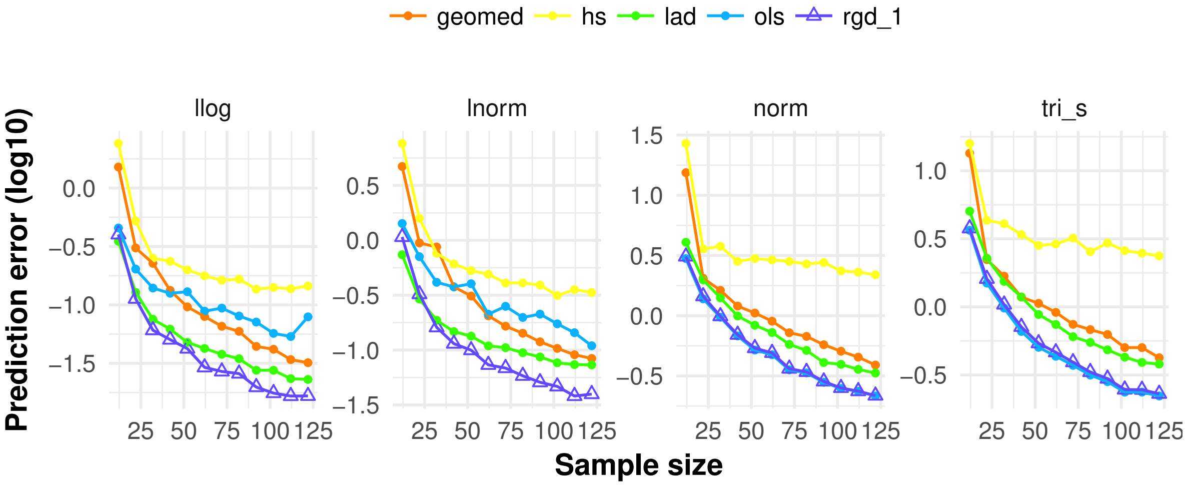

Figure 4.11: Prediction error over sample size $12 \leq n \leq 122$, fixed $d = 5$, noise level $= 8$.

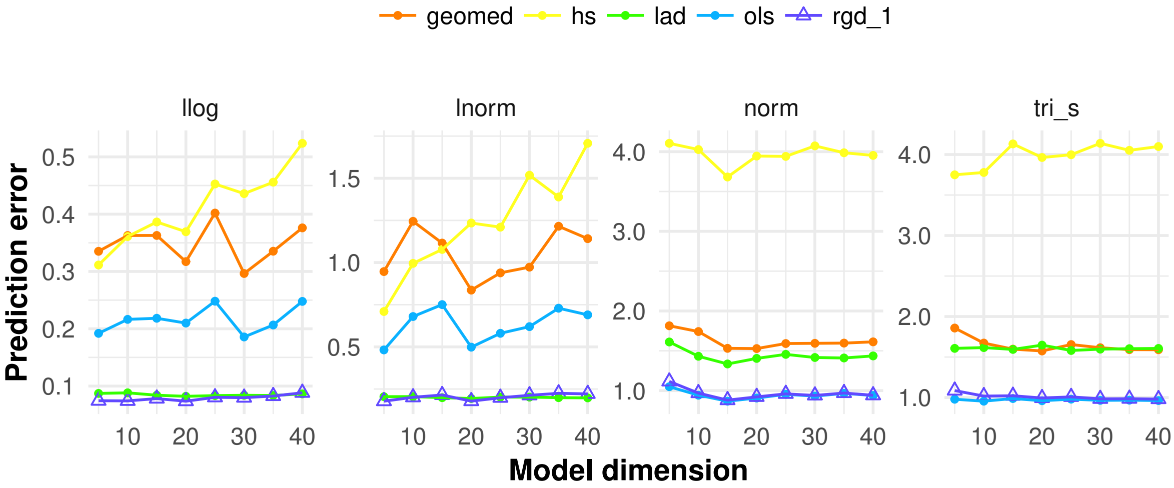

Figure 4.12: Prediction error over dimensions $5 \leq d \leq 40$, with ratio $n/d = 6$ fixed, and noise level $= 8$.

the proposed algorithm has desirable learning efficiency, in that it can competitively generalize, with less samples, over more distributions than its competitors.

Moving forward, a more careful analysis of the role that prior knowledge can play on learning efficiency, starting with the first-order optimizer setting, is of significant interest. Characterizing the learning effiency enabled by sharper estimates could lead to useful insights in the context of larger-scale problems, where a small overhead might save countless iterations and dramatically reduce budget requirements, while simultaneously leading to more consistent performance across samples. Another natural line of work is to look at alternative strategies which operate on the data vector as a whole (rather than coordinate-wise), integrating information across coordinates, in order to infer more efficiently.

4.6 Proofs

Basic facts: smooth convex functions Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable, convex, λ-smooth function.

$$f(\boldsymbol{u}) - f(\boldsymbol{v}) \leq \frac{\lambda}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2 + \langle f'(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v}\rangle \tag{4.32}$$

$$\frac{1}{2\lambda}\|f'(\boldsymbol{u}) - f'(\boldsymbol{v})\|^2 \leq f(\boldsymbol{u}) - f(\boldsymbol{v}) - \langle f'(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v}\rangle \tag{4.33}$$

for all $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$.

Basic facts: Orlicz norms For ψ assumed convex and non-decreasing, satisfying $\psi(0) = 0$, one defines the Orlicz norm of random variable z as

$$\|z\|_\psi := \inf\{c > 0 : \mathbf{E}\,\psi(|z|/c) \leq 1\}. \tag{4.34}$$

Setting $\psi(u) := u^q$ yields the special case of the $\mathcal{L}_q$ norm, and $\psi_q(u) := \exp(u^q) - 1$ is used in Theorem 39. The ψ_q version of this norm is very sensitive to the distribution tails, and thus when finite implies much stronger tail control than is possible with $\mathcal{L}_q$ bounds. Most simply, if we fix $q \geq 1$ and $\delta \in (0, 1)$, whenever $\|z\|_{\psi_q} < \infty$, one has

$$\mathbf{P}\left\{|z| \leq \|z\|_{\psi_q}(\log \delta^{-1})^{1/q}\right\} \geq 1 - 2\delta. \tag{4.35}$$

Furthermore, note that

$$\|z\|_{\psi_q} \leq \|z\|_{\psi_p}(\log 2)^{1/q-1/p}, \quad q \leq p. \tag{4.36}$$

For more background, van der Vaart and Wellner [39], Pollard [32] provide a good start.

Basic facts: Evaluating metric entropy integrals Some basic facts for fixed $A > 0$ and $a < b$:

$$\int_a^b \log(A/\varepsilon)\,d\varepsilon = A(b-a) + b\log(A/b) - a\log(A/a) \tag{4.37}$$

$$\begin{aligned}\int_a^b \sqrt{\log(A/\varepsilon)}\,d\varepsilon = \\ b\sqrt{\log(A/b)} - a\sqrt{\log(A/a)} - \frac{A\sqrt{\pi}}{2}\left(\operatorname{erf}\left(\sqrt{\log(A/b)}\right) - \operatorname{erf}\left(\sqrt{\log(A/a)}\right)\right)\end{aligned} \tag{4.38}$$

It follows that taking $a \to 0$ and setting $b = \Delta := \Delta(\mathcal{W}; \|\cdot\|)$,

$$\begin{aligned}E_1(d^*) &\leq E_1(d_\infty) \leq A\Delta + \Delta\log(A/\Delta) \\ E_2(d^*) &\leq E_2(d_\infty) \leq \Delta\sqrt{\log(A/\Delta)} - \frac{A\sqrt{\pi}}{2}\left(\operatorname{erf}\left(\sqrt{\log(A/\Delta)}\right) - 1\right) \\ &\leq \Delta\sqrt{\log(A/\Delta)} + \frac{A\sqrt{\pi}}{2}.\end{aligned}$$

Identical inequalities hold for the $\mathcal{L}_2$ distance as $E_\beta(d_2) \leq E_\beta(d_\infty)$, since $d_2 \leq d_\infty$. To see this, just note $d_2(f,g)^2 = \int (f(\boldsymbol{z}) - g(\boldsymbol{z}))^2\,d\mu(\boldsymbol{z}) \leq d_\infty(f,g)^2 \int d\mu(\boldsymbol{z}) = d_\infty(f,g)^2$. To see that $d^* \leq d_\infty$ holds for d^* defined in (4.27) is straightforward. Writing $\epsilon(\boldsymbol{z}) := f(\boldsymbol{z}) - g(\boldsymbol{z})$ for arbitrary functions f and g in the class of interest, first make the trivial observation that

$$0 = \mathbf{E}_\mu\,\psi_s(g(\boldsymbol{z}) - \theta_g^*) = \mathbf{E}_\mu\,\psi_s(f(\boldsymbol{z}) - (\theta_g^* + \epsilon(\boldsymbol{z}))). \tag{4.39}$$

Now, say $|\theta_f^* - \theta_g^*| > d_\infty(f,g)$. Note that $d_\infty(f,g) \geq |\epsilon(\boldsymbol{z})|$, and writing $\delta := |\theta_f^* - \theta_g^*|$, say WLOG that $\theta_f^* = \theta_g^* + \delta$. Then, we have

$$\begin{aligned}
0 &= \mathbf{E}_\mu\, \psi_s(f(\boldsymbol{z}) - \theta_f^*) \\
&= \mathbf{E}_\mu\, \psi_s(f(\boldsymbol{z}) - (\theta_g^* + \delta)) \\
&< \mathbf{E}_\mu\, \psi_s(f(\boldsymbol{z}) - (\theta_g^* + \epsilon(\boldsymbol{z}))) \\
&= 0.
\end{aligned}$$

The strict inequality follows from the strict monotonicity of ψ and the fact that $\delta > \epsilon(\boldsymbol{z})$ almost surely. The last equality comes directly from (4.39). Since this is a contradiction, we conclude that $|\theta_f^* - \theta_g^*| \leq d_\infty(f,g)$.

Pointwise error setting

Proof of Lemma 25. For cleaner notation, write $x_1, \ldots, x_n \in \mathbb{R}$ for our iid observations. Here ρ is assumed to satisfy the conditions of Defn. 34. A high-probability concentration inequality follows by direct application of the specified properties of ρ and $\psi := \rho'$, following the general technique laid out by Catoni [8, 9]. For $u \in \mathbb{R}$ and $s > 0$, writing $\psi_s(u) := \psi(u/s)$, and taking expectation over the random draw of the sample,

$$\begin{aligned}
\mathbf{E} \exp\left(\sum_{i=1}^n \psi_s(x_i - u)\right) &\leq \left(1 + \frac{1}{s}(\mathbf{E}\, x - u) + \frac{C}{s^2}\,\mathbf{E}(x^2 + u^2 - 2xu)\right)^n \\
&\leq \exp\left(\frac{n}{s}(\mathbf{E}\, x - u) + \frac{Cn}{s^2}(\operatorname{var} x + (\mathbf{E}\, x - u)^2)\right).
\end{aligned}$$

The inequalities above are due to an application of the upper bound on ψ, and the inequality $(1+u) \leq \exp(u)$. Now, letting

$$\begin{aligned}
A &:= \frac{1}{n}\sum_{i=1}^n \psi_s(x_i - u) \\
B &:= \frac{1}{s}(\mathbf{E}\, x - u) + \frac{C}{s^2}(\operatorname{var} x + (\mathbf{E}\, x - u)^2)
\end{aligned}$$

we have a bound on $\mathbf{E}\exp(nA) \leq \exp(nB)$. By Chebyshev's inequality, we then have

$$\begin{aligned}
\mathbf{P}\{A > B + \varepsilon\} &= \mathbf{P}\{\exp(nA) > \exp(nB + n\varepsilon)\} \\
&\leq \frac{\mathbf{E}\exp(nA)}{\exp(nB + n\varepsilon)} \\
&\leq \exp(-n\varepsilon).
\end{aligned}$$

Setting $\varepsilon = \log(\delta^{-1})/n$ for confidence level $\delta \in (0,1)$, and for convenience writing

$$b(u) := \mathbf{E}\, x - u + \frac{C}{s}(\operatorname{var} x + (\mathbf{E}\, x - u)^2),$$

we have with probability no less than $1 - \delta$ that

$$\frac{s}{n}\sum_{i=1}^n \psi_s(x_i - u) \leq b(u) + \frac{s\log(\delta^{-1})}{n}. \tag{4.40}$$

The right hand side of this inequality, as a function of u, is a polynomial of order 2, and if

$$1 \geq D := 4\left(\frac{C^2 \operatorname{var} x}{s^2} + \frac{C \log(\delta^{-1})}{n}\right),$$

then this polynomial has two real solutions. In the hypothesis, we stated that the result holds "for large enough n and s_j." By this we mean that we require n and s to satisfy the preceding inquality (for each $j \in [d]$ in the multi-dimensional case). The notation D is for notational simplicity. The solutions take the form

$$u = \frac{1}{2}\left(2\,\mathbf{E}\,x + \frac{s}{C} \pm \frac{s}{C}\,(1-D)^{1/2}\right).$$

Looking at the smallest of the solutions, noting $D \in [0,1]$ this can be simplified as

$$\begin{aligned} u_{+} &:= \mathbf{E}\,x + \frac{s}{2C}\frac{(1-\sqrt{1-D})(1+\sqrt{1-D})}{1+\sqrt{1-D}} \\ &= \mathbf{E}\,x + \frac{s}{2C}\frac{D}{1+\sqrt{1-D}} \\ &\leq \mathbf{E}\,x + sD/2C, \end{aligned} \tag{4.41}$$

where the last inequality is via taking the $\sqrt{1-D}$ term in the previous denominator as small as possible. Now, writing $\widehat{x}$ as the M-estimate using s and ρ as in (4.4), note that $\widehat{x}$ equivalently satisfies $\sum_{i=1}^{n} \psi_s(\widehat{x} - x_i) = 0$. Using (4.40), we have

$$\frac{s}{n}\sum_{i=1}^{n} \psi_s(x_i - u_{+}) \leq b(u_{+}) + \frac{s\log(\delta^{-1})}{n} = 0,$$

and since the left-hand side of (4.40) is a monotonically decreasing function of u, we have immediately that $\widehat{x} \leq u_{+}$ on the event that (4.40) holds, which has probability at least $1-\delta$. Then leveraging (4.41), it follows that on the same event,

$$\widehat{x} - \mathbf{E}\,x \leq sD/2C.$$

An analogous argument provides a $1-\delta$ event on which $\widehat{x} - \mathbf{E}\,x \geq -sD/2C$, and thus using a union bound, one has that

$$|\widehat{x} - \mathbf{E}\,x| \leq 2\left(\frac{C \operatorname{var} x}{s} + \frac{s\log(\delta^{-1})}{n}\right) \tag{4.42}$$

holds with probability no less than $1-2\delta$. Setting the x_i to $l'_j(\boldsymbol{w};\boldsymbol{z}_i)$ for $j \in [d]$ and some $\boldsymbol{w} \in \mathbb{R}^d$, $i \in [n]$, and $\widehat{x}$ to $\widehat{\theta}_j$ corresponds to the special case considered in this Lemma. Dividing δ by two yields the $(1-\delta)$ result. □

Proof of Lemma 26. For each $t = 0, \ldots, T-1$, and $j \in [d]$, note that

$$\begin{aligned} |\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)})| \leq \varepsilon_j &:= 2\left(\frac{C \operatorname{var}_{\mu} l'_j(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z})}{s_j} + s_j \log(2\delta^{-1})\right) \\ &= 2\sqrt{\frac{\log(2\delta^{-1})}{n}}\left(\frac{C \operatorname{var}_{\mu} l'_j(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z})}{\widehat{\sigma}_j} + \widehat{\sigma}_j\right) \\ &\leq \varepsilon^{*} := 2\sqrt{\frac{V\log(2\delta^{-1})}{n}}c_0 \end{aligned} \tag{4.43}$$

holds with probability no less than $1-\delta$. The first inequality holds via direct application of Lemma 25, which holds under (4.11) and using ρ which satisfies (4.8). The equality follows immediately from (4.6). The final inequality follows from (A4) and (4.10), along with the definition of c_0. In d dimensions, writing $\widehat{\boldsymbol{\theta}}_{(t)} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_d)$, we then have for any $\varepsilon > 0$

$$\begin{aligned}\mathbf{P}\left\{\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| > \varepsilon\right\} &= \mathbf{P}\left\{\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\|^2 > \varepsilon^2\right\} \\ &\leq \sum_{j=1}^{d} \mathbf{P}\left\{|\widehat{\theta}_j - \boldsymbol{g}_j(\widehat{\boldsymbol{w}}_{(t)})| > \frac{\varepsilon}{\sqrt{d}}\right\}.\end{aligned}$$

Using the notation of ε_j and ε^* from (4.43), filling in $\varepsilon = \sqrt{d}\varepsilon^*$, we thus have

$$\begin{aligned}\mathbf{P}\left\{\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| > \sqrt{d}\varepsilon^*\right\} &\leq \sum_{j=1}^{d} \mathbf{P}\left\{|\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)})| > \varepsilon^*\right\} \\ &\leq \sum_{j=1}^{d} \mathbf{P}\left\{|\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)})| > \varepsilon_j\right\} \\ &\leq d\delta.\end{aligned}$$

The second inequality is because $\varepsilon_j \leq \varepsilon^*$ for all $j \in [d]$, and the final inequality is due to (4.43). Setting the per-coordinate confidence to δ/d, and plugging into these inequalities, we get the desired result. □

Proof of Lemma 28. Given $\widehat{\boldsymbol{w}}_{(t)}$, running the approximate update (4.3), we have

$$\begin{aligned}\|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^*\| &= \|\widehat{\boldsymbol{w}}_{(t)} - \alpha\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{w}^*\| \\ &\leq \|\widehat{\boldsymbol{w}}_{(t)} - \alpha\boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{w}^*\| + \alpha\|\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\|.\end{aligned}$$

The first term looks at the distance from the target given an optimal update, using $\boldsymbol{g}$. Using the κ-strong convexity of R, via Nesterov [31, Thm. 2.1.5] it follows that

$$\|\widehat{\boldsymbol{w}}_{(t)} - \alpha\boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{w}^*\|^2 \leq \left(1 - \frac{2\alpha\kappa\lambda}{\kappa + \lambda}\right) \|\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{w}^*\|^2.$$

Writing $\beta \coloneqq 2\kappa\lambda/(\kappa + \lambda)$, the coefficient becomes $(1 - \alpha\beta)$.

To control the second term is immediate using ε via (4.7), and thus we have

$$\|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^*\| \leq \sqrt{1 - \alpha\beta}\|\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{w}^*\| + \alpha\varepsilon$$

with probability no less than $1 - \delta$. For notational ease, write $a \coloneqq \sqrt{1 - \alpha\beta}$ and $\Delta_k \coloneqq \|\widehat{\boldsymbol{w}}_{(k)} - \boldsymbol{w}^*\|$ for each $0 < k \leq t$. To unfold the recursion, we must apply (4.7) an additional t times, for a total of $t + 1$ applications, yielding

$$\begin{aligned}\|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^*\| &\leq a^{t+1}\Delta_0 + \alpha\varepsilon\left(1 + a + a^2 + \cdots + a^t\right) \\ &= a^{t+1}\Delta_0 + \alpha\varepsilon\frac{(1 - a^{t+1})}{1 - a}\end{aligned}$$

with probability no less than $1 - (t + 1)\delta$. To clean up the second summand,

$$\begin{aligned}\alpha\varepsilon\frac{(1 - a^{t+1})}{1 - a} &\leq \frac{\alpha\varepsilon(1 + a)}{(1 - a)(1 + a)} \\ &= \frac{\alpha\varepsilon(1 + \sqrt{1 - \alpha\beta})}{\alpha\beta} \\ &\leq \frac{2\varepsilon}{\beta}.\end{aligned}$$

Taking this to the original inequality yields the desired result. □

Proof of Theorem 29. Using strong convexity and (4.32), we have that

$$\begin{aligned} R(\widehat{\boldsymbol{w}}_{(T)}) - R^* &\leq \frac{\lambda}{2}\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*\|^2 \\ &\leq \lambda(1-\alpha\beta)^T D_0^2 + \frac{4\lambda\varepsilon^2}{\beta^2}. \end{aligned}$$

The latter inequality holds by direct application of Lemma 28, followed by the elementary fact $(a+b)^2 \leq 2(a^2+b^2)$. Controlling the gradient accuracy with ε is done using Lemma 26, which implies the desired result. □

Proof of Lemma 30. For arbitrary step t, comparing the results of updates (4.2) and (4.3) with common step size $\alpha_{(t)}$, we have

$$\begin{aligned} \|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^*_{(t+1)}\| &\leq \|\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{w}^*_{(t)}\| + |\alpha_{(t)}| \left(\|\widehat{\boldsymbol{g}}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| + \|\boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)}) - \boldsymbol{g}(\boldsymbol{w}^*_{(t)})\| \right) \\ &\leq \|\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{w}^*_{(t)}\| \left(1 + \lambda\alpha_{(t)}\right) + \alpha_{(t)}\varepsilon. \end{aligned} \tag{4.44}$$

The latter inequality follows from the ε-accuracy and λ-smoothness in the hypothesis. Next, note that for any $t \geq 1$, if we have

$$\|\widehat{\boldsymbol{w}}_{(t)} - \boldsymbol{w}^*_{(t)}\| \leq \frac{\varepsilon}{\lambda}\left(\prod_{k=0}^{t-1} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right),$$

then using (4.44), it follows that in the next iteration

$$\begin{aligned} \|\widehat{\boldsymbol{w}}_{(t+1)} - \boldsymbol{w}^*_{(t+1)}\| &\leq \frac{\varepsilon}{\lambda}\left(\prod_{k=0}^{t-1} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right) \left(1 + \lambda\alpha_{(t)}\right) + \alpha_{(t)}\varepsilon \\ &= \frac{\varepsilon}{\lambda}\left(\prod_{k=0}^{t} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right). \end{aligned}$$

Finally noting that we have the base case

$$\|\widehat{\boldsymbol{w}}_{(1)} - \boldsymbol{w}^*_{(1)}\| \leq \alpha_{(0)}\varepsilon = \frac{\varepsilon}{\lambda}\left((1 + \lambda\alpha_{(0)}) - 1 \right),$$

taking the form assumed in the induction step. The desired result follows by mathematical induction. □

Proof of Corollary 31. Begin by controlling the risk, using (A2) and (4.32):

$$\begin{aligned} R(\widehat{\boldsymbol{w}}_{(T)}) - R^* &= R(\widehat{\boldsymbol{w}}_{(T)}) - R(\boldsymbol{w}^*_{(T)}) + R(\boldsymbol{w}^*_{(T)}) - R^* \\ &\leq \frac{\lambda}{2}\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)}\|^2 + \langle \boldsymbol{g}(\boldsymbol{w}^*_{(T)}), \widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)} \rangle + R(\boldsymbol{w}^*_{(T)}) - R^* \\ &\leq \frac{\lambda}{2}\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)}\|^2 + \|\boldsymbol{g}(\boldsymbol{w}^*_{(T)})\|\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)}\| + R(\boldsymbol{w}^*_{(T)}) - R^*. \end{aligned}$$

Furthermore, using $\boldsymbol{g}(\boldsymbol{w}^*) = 0$ and (4.33), we have

$$\begin{aligned} \|\boldsymbol{g}(\boldsymbol{w}^*_{(T)})\|^2 &= \|\boldsymbol{g}(\boldsymbol{w}^*_{(T)}) - \boldsymbol{g}(\boldsymbol{w}^*)\|^2 \\ &\leq 2\lambda \left(R(\boldsymbol{w}^*_{(T)}) - R(\boldsymbol{w}^*) - \langle \boldsymbol{g}(\boldsymbol{w}^*), \boldsymbol{w}^*_{(T)} - \boldsymbol{w}^* \rangle \right) \\ &= 2\lambda \left(R(\boldsymbol{w}^*_{(T)}) - R(\boldsymbol{w}^*) \right). \end{aligned}$$

By convexity and (A3), we have $R^* = R(\boldsymbol{w}^*)$. Writing $A := \|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)}\|^2$ and $B := R(\boldsymbol{w}^*_{(T)}) - R(\boldsymbol{w}^*)$, it follows that

$$R(\widehat{\boldsymbol{w}}_{(T)}) - R^* \leq \frac{\lambda A}{2} + \sqrt{2\lambda AB} + B.$$

Control of the estimation error A can be done using a direct application of Lemmas 26 and 30, which naturally yield

$$\|\widehat{\boldsymbol{w}}_{(T)} - \boldsymbol{w}^*_{(T)}\| \leq 2c_0\sqrt{\frac{dV\log(2d\delta^{-1})}{n}}\left((1+\lambda\alpha)^T - 1\right)$$

with probability at least $1-\delta$, and taking $\lambda\alpha \leq \exp(\log(1+q)/T) - 1$ implies $(1+\lambda\alpha)^T - 1 \leq q$.

As for the optimization error B, this can be controlled using Theorem 2.1.14 of Nesterov [31], as

$$B \leq \frac{2R_0 D_0^2}{2D_0^2 + T\alpha(2-\lambda\alpha)R_0} = \left(\frac{T\alpha(2-\lambda\alpha)}{2D_0^2} + R_0\right)^{-1}$$

which is valid using (A2) and (4.12). Plugging these into A and B above, we have

$$\begin{aligned} R(\widehat{\boldsymbol{w}}_{(T)}) - R^* \leq & 2\lambda(c_0 q)^2\frac{CdV\log(2dT\delta^{-1})}{n} + \left(\frac{T\alpha(2-\lambda\alpha)}{2D_0^2} + R_0\right)^{-1} \\ & + \sqrt{2\lambda}2c_0 q\left(\frac{CdV\log(2dT\delta^{-1})}{n}\right)^{1/2}\left(\frac{T\alpha(2-\lambda\alpha)}{2D_0^2} + R_0\right)^{-1/2}. \end{aligned}$$

The result stated gives this inequality, using $\lesssim$ to suppress the term of order $O(n^{-1})$ for readability. □

Proof of Lemma 32. As in the result statement, we write

$$\Sigma_{(t)} := \mathbf{E}_\mu\left(l'(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z}) - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\right)\left(l'(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z}) - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\right)^T, \quad \boldsymbol{w} \in \mathcal{W}.$$

Running this modified version of Algorithm 2, we are minimizing the bound in Lemma 25 as a function of scale s_j, $j \in [d]$, which immediately implies that the estimates $\widehat{\boldsymbol{\theta}}_{(t)} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_d)$ at each step t satisfy

$$\mathbf{P}\left\{|\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}})| > 4\left(\frac{C\operatorname{var}_\mu l'_j(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z})\log(2\delta^{-1})}{n}\right)^{1/2}\right\} \leq \delta. \tag{4.45}$$

For clean notation, let us also denote

$$A := 4\left(\frac{C\log(2\delta^{-1})}{n}\right)^{1/2}, \quad \varepsilon^* := A\sqrt{\operatorname{trace}(\Sigma_{(t)})}.$$

For the vector estimates then, we have

$$\begin{aligned}
\mathbf{P}\left\{\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| > \varepsilon^{*}\right\} &= \mathbf{P}\left\{\sum_{j=1}^{d}\frac{(\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)}))^2}{A^2} > \operatorname{trace}(\Sigma_{(t)})\right\}\\
&= \mathbf{P}\left\{\sum_{j=1}^{d}\left(\frac{(\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)}))^2}{A^2} - \operatorname{var}_{\mu} l_j'(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z})\right) > 0\right\}\\
&\leq \mathbf{P}\bigcup_{j=1}^{d}\left\{\frac{(\widehat{\theta}_j - g_j(\widehat{\boldsymbol{w}}_{(t)}))^2}{A^2} > \operatorname{var}_{\mu} l_j'(\widehat{\boldsymbol{w}}_{(t)};\boldsymbol{z})\right\}\\
&\leq d\delta.
\end{aligned}$$

The first inequality uses a union bound, and the second inequality follows from (4.45). Plugging in A and taking confidence δ/d implies the desired result. □

Proof of Theorem 33. From Lemma 32, the estimation error has exponential tails, as follows. Writing

$$A_1 := 2d, \quad A_2 := 4\left(\frac{C\operatorname{trace}(\Sigma_{(t)})}{n}\right)^{1/2},$$

for each iteration t we have

$$\mathbf{P}\{\|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| > \varepsilon\} \leq A_1 \exp\left(-\left(\frac{\varepsilon}{A_2}\right)^2\right).$$

Controlling moments using exponential tails can be done using a fairly standard argument. For random variable $X \in \mathcal{L}_p$ for $p \geq 1$, we have the classic equality

$$\mathbf{E}\,|X|^p = \int_0^\infty \mathbf{P}\{|X|^p > t\}\,dt$$

as a starting point. Setting $X = \|\widehat{\boldsymbol{\theta}}_{(t)} - \boldsymbol{g}(\widehat{\boldsymbol{w}}_{(t)})\| \geq 0$, and using substitution of variables twice, we have

$$\begin{aligned}
\mathbf{E}\,|X|^p &= \int_0^\infty \mathbf{P}\{X > t^{1/p}\}\,dt\\
&= \int_0^\infty \mathbf{P}\{X > t\} p t^{p-1}\,dt\\
&\leq A_1 p \int_0^\infty \exp\left(-(t/A_2)^2\right) t^{p-1}\,dt\\
&= \frac{A_1 A_2^p p}{2}\int_0^\infty \exp(-t) t^{p/2-1}\,dt.
\end{aligned}$$

The last integral on the right-hand side, written $\Gamma(p/2)$, is the usual Gamma function of Euler evaluated at $p/2$. Setting $p = 2$, we have $\Gamma(1) = 0! = 1$, and plugging in the values of A_1 and A_2 yields the desired result. □

Uniform error setting

Proof of Lemma 36. See proof of Lemma 25. The deterministic version follows basically the same strategy Brownlees et al. [7, Lemma 3], looking at θ^{*} rather that $\widehat{\theta}$. □

Proof of Lemma 37. By our definition of ρ, while ρ' is increasing on $\mathbb{R}_+$, its slope becomes arbitrarily small as one gets far away from the origin, i.e., $\rho''(u) \to 0$ as $u \to \pm\infty$. To alleviate this issue, we require that for most samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, at least one of the $\boldsymbol{z}_i$ observations is such that the gap $|f(\boldsymbol{z}) - \widehat{\theta}_f|$ is not too large, for any choice of $f \in \mathcal{F}$. For example, if $\mathcal{F}$ and μ are totally unrestricted, then for arbitrarily large $A > 0$ and say confidence $1/2$, one could always find a f such that $f(\boldsymbol{z})$ satisfies $|f(\boldsymbol{z}_i) - \widehat{\theta}_f| > A$ for a fixed fraction of the indices $i \in [n]$ with probability no less than $1/2$, which would make a uniform bound on $|\widehat{\theta}_f - \theta^*_f|$ impossible. A natural solution, then, is to constrain $\mathcal{F}$ and μ as follows. Fixing $B > 0$, for each $f \in \mathcal{F}$, define a "good index" $I_f \subseteq [n]$ by the property

$$i \in I_f \implies |f(\boldsymbol{z}_i) - \widehat{\theta}_f| \leq B$$

where $\widehat{\theta}_f$ is defined by (4.22) using ρ_B. The event we need is

$$\mathcal{E} := \left\{ \sup_{f \in \mathcal{F}} \max_{i \in I_f} |f(\boldsymbol{z}_i) - \widehat{\theta}_f| \leq B \right\} \tag{4.46}$$

which implicitly requires $\inf\{|I_f| : f \in \mathcal{F}\} > 0$. Taking any $\varepsilon_0 > 0$, by continuity there exists $a > 0$ such that $\psi(1+a) - \psi(1) = \varepsilon_0$. Take slope $K := \rho''(1+a)$. On the event $\mathcal{E}$, for any $f \in \mathcal{F}$ and $i \in I_f$, we have

$$\left| \psi\left(\frac{f(\boldsymbol{z}_i) - \widehat{\theta}_f}{B} - \frac{\theta}{B} \right) - \psi\left(\frac{f(\boldsymbol{z}_i) - \widehat{\theta}_f}{B} \right) \right| > \varepsilon_0 \tag{4.47}$$

as soon as θ is such that $|\theta| \geq B\varepsilon_0/K$ holds.

With this in place, the final argument can be made easily. Denote

$$x_f(\theta) := \frac{B}{n} \sum_{i=1}^{n} \psi_B\left(f(\boldsymbol{z}_i) - \theta\right)$$

and given ε in the hypothesis, write

$$r := \inf\{|I_f| : f \in \mathcal{F}\}$$

and set $\varepsilon_0 = \varepsilon/(Br)$. Say $|\widehat{\theta} - \theta^*|_{\mathcal{F}} > \varepsilon/(Br)$. Then there exists an $f \in \mathcal{F}$ such that

$$|x_f(\widehat{\theta}_f - \theta) - x_f(\widehat{\theta}_f)| > \frac{s|I_f|\varepsilon_0}{n} \geq \varepsilon$$

using (4.47) and $r \leq |I_f|/n$. From this we see

$$\begin{aligned} |X^*_f| &= |x_f(\theta^*_f) - x_f(\widehat{\theta}_f)| \\ &= |x_f(\widehat{\theta}_f - \theta) - x_f(\widehat{\theta}_f)| \\ &> \varepsilon \end{aligned}$$

which is a contradiction. Thus on the event in the hypothesis, we have $|\widehat{\theta} - \theta^*|_{\mathcal{F}} \leq \varepsilon/(Br)$. □

Proof of Lemma 38. Fix any $f, g \in \mathcal{F}$, and write

$$x_i(f,g) := s\left(\psi_s(f(\boldsymbol{z}_i) - \theta^*_f) - \psi_s(g(\boldsymbol{z}_i) - \theta^*_g)\right), \quad i \in [n].$$

Note that $n(X_f^* - X_g^*) = \sum_{i=1}^n x_i(f,g)$. First we bound the x_i using the 1-Lipschitz property of ψ, observing

$$x_i(f,g) \leq |f(\boldsymbol{z}_i) - g(\boldsymbol{z}_i)| + d^*(f,g).$$

Similarly for the second moment, we have

$$\begin{aligned}\mathbf{E}_\mu\, x_i(f,g)^2 &\leq \mathbf{E}_\mu \left(f(\boldsymbol{z}_i) - g(\boldsymbol{z}_i) + \theta_g^* - \theta_f^*\right)^2 \\ &\leq 2\left(d_2(f,g)^2 + d^*(f,g)^2\right).\end{aligned}$$

The result follows from $\mathbf{E}_\mu\, x_i(f,g) = 0$,

$$\mathbf{P}\{|X_f^* - X_g^*| \geq t\} = \mathbf{P}\left\{|\sum_{i=1}^n x_i(f,g)| \geq nt\right\},$$

and a direct application of the inequality of Bernstein [6, Section 2.8]. □

Proof of Theorem 39. Using the quantities γ_β of Talagrand [36] and the generic chaining protocol of Brownlees et al. [7, Theorem 12], via our Lemma 38, there exists a constant $c > 0$ such that for any $f_0 \in \mathcal{F}$, we have

$$H := \left\| |X^* - X_{f_0}^*|_{\mathcal{F}} \right\|_{\psi_1} \leq c\left(\frac{\gamma_1(\mathcal{F}, D_1)}{n} + \frac{\gamma_2(\mathcal{F}, D_2)}{\sqrt{n}}\right) \tag{4.48}$$

where $\psi_1(u) = \exp(u) - 1$ is the function which induces an Orlicz norm (see appendix); this notation is standard, and has nothing to do with the ψ in our Definition 34. The metrics are

$$\begin{aligned}D_1(f,g) &= \frac{1}{3}\left(d_\infty(f,g) + d^*(f,g)\right) \\ D_2(f,g) &= \sqrt{2}\left(d_2(f,g)^2 + d^*(f,g)^2\right)^{1/2}\end{aligned}$$

and the quantity γ_β is defined as follows. Let D be a quasi-metric for $\mathcal{F}$. Let $\mathcal{A} = (\mathcal{A}_k)_{k=1}^\infty$ denote a sequence of partitions of $\mathcal{F}$. That is, each $\mathcal{A}_k = \{A_1, \ldots, A_m\}$ for some $0 < m < \infty$ and $\cup A_i = \mathcal{F}$, $A_i \cap A_j = \emptyset$ for $i \neq j$. A sequence of partitions is "admissible" if it does not become too fine too quickly, namely if $|\mathcal{A}_k| \leq 2^{2^k}$ for all k.

$$\begin{aligned}\gamma_\beta(\mathcal{F}, D) &:= \inf_{\mathcal{A}} \sup_{f \in \mathcal{F}} \sum_{k=0}^\infty 2^{k/\beta} \Delta(A_k(f)) \\ &\leq c' E_\beta(D)\end{aligned}$$

for some constant $c' > 0$, where the inf is taken over all admissible sequences $\mathcal{A}$, and $A_k(f) \in \mathcal{A}_k$ denotes the element of partition $\mathcal{A}_k$ including f, which is uniquely determined by the definition of partition.

Using (4.48) and our D_1, D_2 above, it requires only some algebraic manipulations to show

$$H \leq 192\sqrt{2}\log(2)\left(\frac{\gamma_1(\mathcal{F}, d_\infty) + \gamma_1(\mathcal{F}, d^*)}{n} + \frac{\gamma_2(\mathcal{F}, d_2) + \gamma_2(\mathcal{F}, d^*)}{\sqrt{n}}\right).$$

Using the metric entropy bound on γ_β, this implies there is a constant c'' such that

$$H \leq c''\left(\frac{E_1(d_\infty) + E_1(d^*)}{n} + \frac{E_2(d_2) + E_2(d^*)}{\sqrt{n}}\right).$$

Note by (4.35), we have that

$$\mathbf{P}\{|X^* - X^*_{f_0}|_{\mathcal{F}} > H\log(\delta^{-1})\} \leq \delta$$

which implies the desired result, with constant $c = c''$. □

Proof of Corollary 41. This follows immediately from Theorem 39, using equations (4.37), (4.38), and the exposition following them to evaluate the definite integrals $E_\beta(D)$ for $\beta \in \{1,2\}$ and $D \in \{d_\infty, d_2, d^*\}$. Since $\mathcal{W}$ is always contained in a $\|\cdot\|$-ball of radius Δ, we can take $A = 3\Delta/2$, and thus the a in the stated result satisfies $a \leq 3/2$, using (4.31), from which the d factor arises. The constant c is precisely the one that appears in Theorem 39. □

Proof of Theorem 42. Setting arbitrary coordinate $j \in [d]$ we have $\mathcal{F}$ as in Corollary 41. The ε in (4.17) is controlled by $\varepsilon \leq (|X^* - X^*_{f_0}|_{\mathcal{F}} + |X^*_{f_0}|)$. Fixing an arbitrary $\boldsymbol{u} \in \mathcal{W}$ to set $f_0 = l'_j(\boldsymbol{u};\cdot)$, the term $|X^*_{f_0}|$ in (4.25) is $O_P(1)$ taking $s \to \infty$ as $O(\sqrt{vn})$. Using $\lesssim$ to suppress terms of $O(1/n)$ order, we get

$$|\widehat{\theta} - \theta^*|_{\mathcal{F}} \lesssim c_1 \left(c_2 d \log(d\delta^{-1}) O\left(\frac{\Delta}{\sqrt{n}}\right) + O_P(1) \right)$$

with probability no less than $1 - \delta/d$, via Corollary 41. Here c_1 is the constant from Lemma 37, while c_2 is that from Corollary 41. Setting $|\widehat{g}_j - g_j|_{\mathcal{W}} = |\widehat{\theta} - \theta|_{\mathcal{F}}$, analogous results hold for each coordinate $j \in [d]$, though constant may differ. Absorbing these into the asymptotic notation, it follows from (4.17) that

$$\|\widehat{\boldsymbol{g}}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{w})\| \lesssim d^{3/2} \log(d\delta^{-1}) O\left(\frac{\Delta}{\sqrt{n}}\right) + O_P(1)$$

with probability no less than $1 - \delta$. □

4.6.1 Computational methods

Here we discuss precisely how to compute the implicitly-defined M-estimates of (4.4) and (4.6). Assuming $s > 0$ and real-valued observations $x_1, \ldots, x_n$, we first look at the program

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \rho_s(x_i - \theta)$$

assuming ρ is as specified in Defn. 34, with $\psi = \rho'$. Write $\widehat{\theta}$ for this unique minimum, and note that it satisfies

$$\frac{s}{n} \sum_{i=1}^{n} \psi_s\left(x_i - \widehat{\theta}\right) = 0.$$

Indeed, by monotonicity of ψ, this $\widehat{\theta}$ can be found via ρ minimization or root-finding. The latter yields standard fixed-point iterative updates, such as

$$\widehat{\theta}_{(k+1)} = \widehat{\theta}_{(k)} + \frac{s}{n} \sum_{i=1}^{n} \psi_s\left(x_i - \widehat{\theta}_{(k)}\right).$$

Note the right-hand side has a fixed point at the desired value. In our routines, we use the Gudermannian function

$$\rho(u) := \int_0^u \psi(x)\,dx, \quad \psi(u) := 2\,\mathrm{atan}(\exp(u)) - \pi/2$$

which can be readily confirmed to satisfy all requirements of Defn. 34.

For the dispersion estimate to be used in re-scaling, we introduce function χ, which is even, non-decreasing on $\mathbb{R}_{+}$, and satisfies

$$0 < \left| \lim_{u \to \pm\infty} \chi(u) \right| < \infty, \quad \chi(0) < 0.$$

In practice, we take dispersion estimate $\widehat{\sigma} > 0$ as any value satisfying

$$\frac{1}{n} \sum_{i=1}^{n} \chi\left(\frac{x_i - \gamma}{\widehat{\sigma}}\right) = 0$$

where $\gamma = n^{-1} \sum_{i=1}^{n} x_i$, computed by the iterative procedure

$$\widehat{\sigma}_{(k+1)} = \widehat{\sigma}_{(k)} \left(1 - \frac{1}{\chi(0)n} \sum_{i=1}^{n} \chi\left(\frac{x_i - \gamma}{\widehat{\sigma}_{(k)}}\right)\right)^{1/2}$$

which has the desired fixed point, as in the location case. Our routines use the quadratic Geman-type χ, defined

$$\chi(u) := \frac{u^2}{1 + u^2} - c$$

with parameter $c > 0$, noting $\chi(0) = -c$. Writing the first term as χ_0 so $\chi(u) = \chi_0(u) - c$, we set $c = \mathbf{E}\,\chi_0(x)$ under $x \sim N(0,1)$. Computed via numerical integration, this is $c \approx 0.34$.

Bibliography

[1] Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. US National Bureau of Standards.

[2] Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. (1997). Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631.

[3] Ash, R. B. and Doleans-Dade, C. (2000). *Probability and Measure Theory*. Academic Press.

[4] Bartlett, P. L., Long, P. M., and Williamson, R. C. (1996). Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452.

[5] Bartlett, P. L. and Mendelson, S. (2003). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

[6] Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: a nonasymptotic theory of independence*. Oxford University Press.

[7] Brownlees, C., Joly, E., and Lugosi, G. (2015). Empirical risk minimization for heavy-tailed losses. *Annals of Statistics*, 43(6):2507–2536.

[8] Catoni, O. (2009). High confidence estimates of the mean of heavy-tailed real random variables. *arXiv preprint arXiv:0909.5366*.

[9] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148–1185.

[10] Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *arXiv preprint arXiv:1705.05491*.

[11] Daniely, A. and Shalev-Shwartz, S. (2014). Optimal learners for multiclass problems. In *27th Annual Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 287–316.

[12] Devroye, L., Lerasle, M., Lugosi, G., and Oliveira, R. I. (2015). Sub-Gaussian mean estimators. *arXiv preprint arXiv:1509.05845*.

[13] Feldman, V. (2016). Generalization of ERM in stochastic convex optimization: The dimension strikes back. In *Advances in Neural Information Processing Systems 29*, pages 3576–3584.

[14] Finkenstädt, B. and Rootzén, H., editors (2003). *Extreme Values in Finance, Telecommunications, and the Environment*. CRC Press.

[15] Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. *Computer Journal*, 7(2):149–154.

[16] Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. (2015). Competing with the empirical risk minimizer in a single pass. *arXiv preprint arXiv:1412.6606*.

[17] Hsu, D. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, 17(18):1–40.

[18] Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics.* John Wiley & Sons, 2nd edition.

[19] Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323.

[20] Kearns, M. J. and Schapire, R. E. (1994). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497.

[21] Kolmogorov, A. N. (1993). ε-entropy and ε-capacity of sets in functional spaces. In Shiryayev, A. N., editor, *Selected Works of A. N. Kolmogorov, Volume III: Information Theory and the Theory of Algorithms*, pages 86–170. Springer.

[22] Le Roux, N., Schmidt, M., and Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2663–2671.

[23] Lecué, G. and Lerasle, M. (2017). Learning from MOM's principles. *arXiv preprint arXiv:1701.01961.*

[24] Lerasle, M. and Oliveira, R. I. (2011). Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914.*

[25] Lin, J. and Rosasco, L. (2016). Optimal learning for multi-pass stochastic gradient methods. In *Advances in Neural Information Processing Systems 29*, pages 4556–4564.

[26] Luenberger, D. G. (1969). *Optimization by Vector Space Methods.* John Wiley & Sons.

[27] Lugosi, G. and Mendelson, S. (2016). Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757.*

[28] Minsker, S. (2015). Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335.

[29] Minsker, S. and Strawn, N. (2017). Distributed statistical estimation and rates of convergence in normal approximation. *arXiv preprint arXiv:1704.02658.*

[30] Murata, T. and Suzuki, T. (2016). Stochastic dual averaging methods using variance reduction techniques for regularized empirical risk minimization problems. *arXiv preprint arXiv:1603.02412.*

[31] Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course.* Springer.

[32] Pollard, D. (2002). *A User's Guide to Measure Theoretic Probability.* Cambridge University Press.

[33] R Core Team (2016). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

[34] Rakhlin, A., Shamir, O., and Sridharan, K. (2012). Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning*, pages 449–456.

[35] Shalev-Shwartz, S. and Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599.

[36] Talagrand, M. (2014). *Upper and lower bounds for stochastic processes: modern methods and classical problems*. Springer.

[37] Talvila, E. (2001). Necessary and sufficient conditions for differentiating under the integral sign. *American Mathematical Monthly*, 108(6):544–548.

[38] van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press.

[39] van der Vaart, A. W. and Wellner, J. A. (1996). *Weak Convergence and Empirical Processes*. Springer.

[40] Vardi, Y. and Zhang, C.-H. (2000). The multivariate L_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426.

Chapter 5

Conclusion

To conclude the main body of this thesis, we recall the chief message that was foreshadowed early on in chapter 1:

> *Paying a small computational price for better feedback can lead to substantial payoffs in terms of the stability and robustness of learning.*

In chapters 3 and 4, we proposed novel learning algorithms which invest additional computational resources into carrying out better statistical inference, with the express goal of providing the learning machine with more useful feedback. Through both analysis of the theoretical performance guarantees that can be made, as well as comprehensive numerical tests of off-sample generalization and sensitivity to task parameters, we verified that the aforementioned strategy can lead to a significant increase in learning efficiency. That is, by taking on some additional computational overhead in exchange for better feedback, when compared with more traditional empirical risk minimizers and other robust methods, the two main algorithms proposed in this thesis were able to achieve better solutions, faster. The "fast RLM" algorithm of chapter 3 can be applied to any regression problem where the loss is observable, and the core idea is sufficiently flexible that extending the tool to tasks aside from regression should be straightforward. On the other hand, the "robust gradient descent" algorithm of chapter 4 requires gradient information for the loss, but it has the appeal of being applicable to any learning task where slope information is available, and enjoys much stronger theoretical guarantees. We believe the chief contribution of this thesis to be methodological, since we have demonstrated the practical and formal appeal of a new approach to designing learning algorithms. That said, both algorithms are easy to implement, sufficiently fast, and flexible enough to be deployed in a wide variety of machine learning problems.

To close, we put forward some interesting lines of potential research work that are related to both the core ideas and technical tools that appear throughout this thesis.

Optimizer-centric robustification Perhaps the chief appeal of the robust gradient descent strategy laid out in chapter 4 is that there is virtually no gap between the algorithm being analyzed formally, and the algorithm that a practioner will use in practice. This is due to the fact that we *start* with a computational routine (namely steepest descent), and re-assign computational resources to more effectively approximate the unknown parameters of interest, from the perspective of that routine. Plugging more sophisticated statistical estimators into the routine results in a simple routine which is analytically tractable, all without inducing computationally intractable sub-routines that need to be circumvented (as in chapter 3). Such a strategy should be valid for a much wider class of optimizers, and we are actively pursuing additional research in this direction. Routines which use second-order approximations, constrained situations which make use of projections or Frank-Wolfe type estimates, non-smooth losses, high-dimensional sparse settings, important learning scenarios with respectively distinct optimizers. To more fully demonstrate the merits of this new methodology, it is necessary to prove its efficacy in a wider range of tasks.

Efficiency through robust feedback One of the chief empirical findings of this thesis was that the proposed algorithms could reach superior solutions faster than rival routines, with less observations, and with smaller variance over the draw of the sample. The numerical experiments, however, did not capture the ultra-high dimensional situation, where a tremendous number of parameters need to be culled, and a certain subset need to be precisely determined. This computationally intensive task can be thought to benefit greatly from more efficient search of the parameter space, a task which depends directly on the quality of the feedback provided to the learner. Using small mini-batches to forge very rough estimates and save on per-iteration complexity has been ubiquitous in machine learning applications, but as we discussed in 2.5, speeding this up with sophisticated learning rate strategies has proved challenging. On the other hand, putting more resources into achieving substantially better per-iteration estimates may ease the difficulty of setting learning rates, and indeed allow for much larger steps, since the likelihood of getting off track is decreased. Two scenarios of interest are the high-dimensional sparse setting, and the distributed high-dimensional setting, in particular the "federated learning" scenario [6].

When does ERM fail to be economical? In this thesis, one of the chief problems raised with existing techniques was the fact that the sample mean does not always furnish a reliable estimate of the desired parameter, and consequently that only very weak guarantees are possible for ERM-based algorithms. It is easy to say that "without sub-Gaussianity, ERM fails," but of course this is false. Traditionally, sub-Gaussian assumptions on the loss have provided a convenient scenario in which sharp confidence bounds on the risk are readily available. Simply lacking a guarantee of good performance assuredly does not imply bad performance, but when such guarantees are technically extremely challenging to make, it does suggest that such guarantees may not be possible. That said, in more recent theoretical efforts, new tools have been developed to better characterize the class of data distributions for which appealing performance guarantees are available for ERM. Notable work is due to Mehta and Williamson [3], Grünwald and Mehta [2], Dinh et al. [1], Mendelson [4, 5]. These works provide new, weaker conditions for upper bounds on the excess risk of ERM that decrease at a desirable rate, and represent an important technical contribution to a sub-domain which was stalled for a number of years. That said, from our perspective, the more important question is not when the ERM strategy as a whole fails to have satisfactory guarantees, but when it cannot be guaranteed to be *economical*. Consideration of this facet of the problem necessarily requires explicit consideration of the method by which ERM is implemented, and applying the new technical tools that have recently appeared in the literature to analyze what implementions fail/succeed to be economical, and when, is an important and interesting direction to be pursued.

Non-risk performance metrics Finally, the reader has assuredly noticed that our focus, and the focus of virtually all the cited work, has been on creating learning machines which "generalize" in the extremely narrow sense of achieving small *risk*. While we have touched on this point at other locations in this thesis, it must be understood that the expected value simply one parameter of the distribution of the loss, among countless others, and has obvious limitations. For example, depending on the application, "above average" losses might be particularly undesirable, in which case a quantile of the distribution larger than the risk would likely provide a better *performance* metric, strictly speaking. In other cases, sample-wise variation in performance might be highly undesirable, in which case both the risk and the variance of the loss should be simulateously minimized. The requirements of the system can be encoded directly in the loss(es) used, or can be reflected in the choice of ideal parameters to be optimized, which subsequently guides the algorithm design and selection process. The basic motivation of using non-risk "targets" for algorithm design is the core idea at the heart of the "fast RLM" routine proposed in chapter 3, but the literature on multi-parameter and non-risk learning models is extremely sparse. We believe this to be a potentially extremely fruitful direction, both in terms of creating a richer theory of learning machines, but also in engineering learning algorithms which are better able to reflect the needs and desires of the end-users of the systems to which they are applied.

Bibliography

[1] Dinh, V. C., Ho, L. S., Nguyen, B., and Nguyen, D. (2016). Fast learning rates with heavy-tailed losses. In *Advances in Neural Information Processing Systems 29*, pages 505–513.

[2] Grünwald, P. D. and Mehta, N. A. (2016). Fast rates with unbounded losses. *arXiv preprint arXiv:1605.00252*.

[3] Mehta, N. A. and Williamson, R. C. (2014). From stochastic mixability to fast rates. In *Advances in Neural Information Processing Systems 27*, pages 1197–1205.

[4] Mendelson, S. (2015). Learning without concentration. *Journal of the ACM*, 62(3):1–25.

[5] Mendelson, S. (2017). Extending the small-ball method. *arXiv preprint arXiv:1709.00843*.

[6] Wang, S., Roosta-Khorasani, F., Xu, P., and Mahoney, M. W. (2017). GIANT: Globally improved approximate newton method for distributed optimization. *arXiv preprint arXiv:1709.03528*.

Appendix A

Supplementary resources

A.1 Functions for M-estimation

In Definitions 10 and 34, we made use of a very particular class of M-estimators of location, which played a central role in the algorithms where they were applied. In addition, a more auxiliary role was played by M-estimators of scale. Here we provide some supplementary information regarding this class of estimators, along with some references that provide historical context.

A.1.1 Location

Starting with the location estimator, recall that we made use of a function ρ with the following properties:

- $\rho : \mathbb{R} \to [0, \infty)$ is even
- $\rho(u) = O(u)$ as $u \to \pm\infty$.
- $\rho(u)/u^2 \to K$ as $u \to 0$, for some $K > 0$.
- $\rho'' > 0$, and for some $C > 0$, and all $u \in \mathbb{R}$,

$$-\log(1 - u + Cu^2) \leq \rho'(u) \leq \log(1 + u + Cu^2)$$

The uniform bounds on ρ' are a generalization of the key property utilized in the analysis of Catoni [4, 5]. While other forms for these bounds are assuredly plausible, this form is convenient for finding confidence intervals, and gives us a tool to build robust estimators with a controllable bias, as explored in chapters 3–4. The value of $C > 0$ determines the range over which ρ is effectively quadratic. Figure A.1 shows the impact of this value clearly.

Next we explore some examples of ρ which satisfy the above conditions, all of which appear in the statistics and/or machine learning literature with varying frequencies. Strong convexity is immediate, and the limiting properties can often be checked directly, or by using an application of L'Hôpital's rule. Checking the bounds on ρ' formally can be rather tedious, but is usually an elementary exercise (e.g., Lemma 43). The names in parentheses are the abbreviations used throughout our source code implementing numerical tests considered later in this chapter. Denote $\psi := \rho'$, and $\eta := \rho''$.

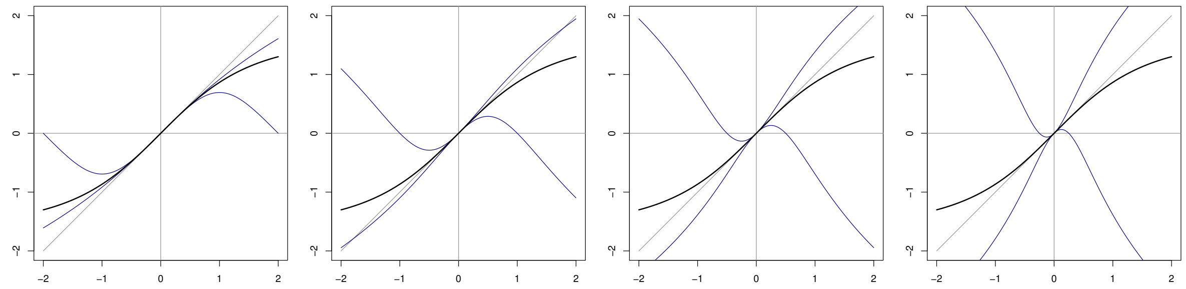

Figure A.1: Upper and lower bounds for a sigmoidal function. Left to right, $C = 1/2, 1, 2, 4$.

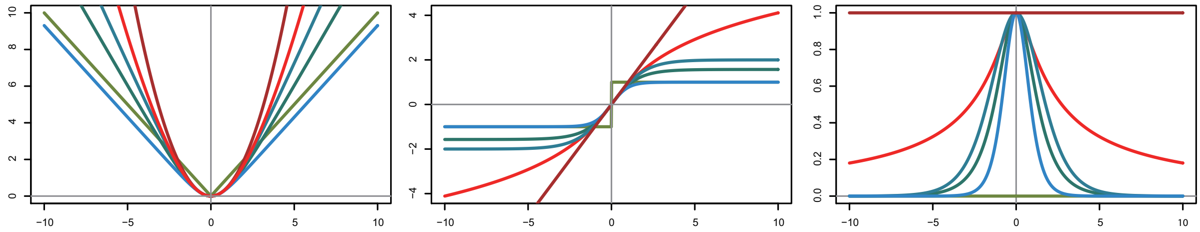

Figure A.2: Six examples of ρ (left) and their corresponding derivatives ψ (middle) and η (right). Three are valid, three are not. Crimson: quadratic. Red: wide Catoni. Green-blue: logistic, Gudermannian, and $\log\cosh$. Olive green: absolute.

Simple algebraic function While there are numerous possible choices, a rather appealing choice of algebraic function with the desired properties is the simple function

$$\rho(u) := 2\left(\sqrt{1+u^2/2} - 1\right),$$

with derivatives

$$\psi(u) = \frac{u}{\sqrt{1+u^2/2}}, \quad \eta(u) = \frac{1}{\sqrt{1+u^2/2}}\left(1 - \frac{u^2}{2+u^2}\right).$$

Inverse tangent function Another familiar choice based on inverse trigonometric functions is

$$\rho(u) := u\,\mathrm{atan}(u) - \log(1+u^2)/2,$$

with derivatives

$$\psi(u) = \mathrm{atan}(u), \quad \eta(u) = \frac{1}{1+u^2}.$$

Gudermannian function This function is specified implicitly by a sigmoidal function as follows.

$$\rho(u) := \int_0^u \psi(x)\,dx, \quad \psi(u) := 2\,\mathrm{atan}\left(\exp\left(u\right)\right) - \pi/2, \quad \eta(u) = \frac{2\exp(u)}{\exp(2u)+1}. \tag{A.1}$$

Historically, ψ here is known as the Gudermannian function [1, Ch. 4], named after C. Gudermann[1]. As an illustrative result, we show that this function is bounded for small C.

[1] Christoph Gudermann (1798–1852). Well-known as a teacher of K. Weierstrass, and for foundational work on elliptic functions [16].

Lemma 43. *There exists a constant C such that for all $u \in \mathbb{R}$,*

$$-\log(1 - u + Cu^2) < 2\,\mathrm{atan}\,(\exp(u)) - \pi/2 < -\log(1 + u + Cu^2) \tag{A.2}$$

Proof of Lemma 43. Let ψ denote the Gudermannian function. Start with the upper bound on $\mathbb{R}_+$, denoting $\psi_U(u) := \log(1 + u + Cu^2)$, with derivative

$$\psi_U'(u) = \frac{1 + 2Cu}{1 + u + Cu^2}.$$

Clearly $\psi_U'(0) = 1$ and $\psi_U'(u) > 1 \iff 2C > 1 + Cu$. Taking $C = 1/2 + \varepsilon$, this will occur for all $0 < u < 2\varepsilon/(\varepsilon + (1/2))$. Note that

$$\psi'(u) = \frac{2\exp(u)}{\exp(2u) + 1},$$

and so we have $\psi'(u) \leq 1$ as long as $2 \leq \exp(u) + (1/\exp(u))$, which holds for all $u \geq 0$. With this and $\psi'(0) = 1$, the bound $\psi \leq \psi_U$ holds on the ε-specified window. That it holds on all of $\mathbb{R}_+$ is checked easily.

Next, for the lower bound, denote $\psi_L(u) := -\log(1 - u + Cu^2)$. Again the derivative is

$$\psi_L'(u) = \frac{1 - 2Cu}{1 - u + Cu^2}.$$

Looking at the inequality $\psi_L'(u) \leq \psi'(u)$, one requires

$$A := 2\exp(u) + 2Cu\exp(2u) - 2u\exp(u) + 2Cu^2\exp(u) \geq B := \exp(2u) + 1 - 2Cu.$$

Defining $A' := 2\exp(u) + 2Cu^2\exp(u)$ and using the fact that for $C \geq 1$, $2Cu\exp(2u) \geq 2u\exp(u)$, it follows that $A \geq A'$. Then, noting that $A' - B = 0$ at $u = 0$, and that taking C large enough, we can clearly get $d/du(A' - B) > 0$ on $u \in (0, \delta)$ for some positive δ. We thus have $A \geq B$ on this window, and similarly $\psi_L(u) := -\log(1 - u + Cu^2)$ as desired. To show that once separated these two functions do not meet is again readily checked. We thus have the upper bound on all of $\mathbb{R}_+$. Symmetry of ψ and the fact that $-\psi_L(u) = \psi_U(-u)$ implies the remaining results for $\mathbb{R}_-$, concluding the proof. □

Log-hyperbolic cosine function A well-known function which grows slowly, and which was featured rather prominently in tests for the FastICA algorithm [14].

$$\rho(u) := \log(\cosh(u)), \quad \psi(u) = \tanh(u), \quad \eta(u) = \frac{1}{\cosh^2(u)}.$$

Logistic function Another implicitly-defined ρ, based on the well-known "logistic function," properly shifted and re-scaled with parameters $\kappa_1, \kappa_2 > 0$, as

$$\rho(u) := \int_0^u \psi(x)\,dx, \quad \psi(u) := \frac{\kappa_1}{1 + \exp(-\kappa_2 u)} - \frac{\kappa_1}{2}, \quad \eta(u) = \frac{\kappa_1\kappa_2\exp(-\kappa_2 u)}{(1 + \exp(-\kappa_2 u))^2}.$$

Setting $\kappa_1 = 4$, $\kappa_2 = 1$, we get $K = 1/2$ in the conditions for ρ. The history of this function in the sciences is long. It appears in classical work of P.F. Verhulst[2] from the 1840s, as the solution

[2] Pierre François Verhulst (1804–1849), mathematician and pupil of A. Quetelet, who introduced the term "logistic" (*logistique*) in work published between 1838–1847 [7].

to a differential equation modeling population growth (in continuous time). More recently, the "logistic equation" featured famously in Li and Yorke [15] is a discrete-time variant of this model, which we note also featured prominently in an article by May [17].

In contrast with the above examples, we now provide some particularly well-known examples of ρ which do not meet the criteria given, from literature both classical and modern. For more, see references such as Andrews et al. [2], Rey [18], Hampel et al. [12].

Sub-quadratic function The following function behaves as desired around the origin, but fails to satisfy the $\rho(u) = O(u)$ requirement.

$$\rho(u) := |u| \log(1 + |u|), \quad \psi(u)|_{u \neq 0} = \operatorname{sign}(u) \left(\log(1 + |u|) + \frac{|u|}{1 + |u|} \right) \tag{A.3}$$

Catoni function Defined implicitly, from Catoni [5] comes the widest member of his proposed class,

$$\psi(u) := \begin{cases} \log(1 + u + u^2/2) & u \geq 0 \\ -\log(1 - u + u^2/2) & u < 0 \end{cases} \qquad \eta(u) = \begin{cases} \frac{1+u}{1+u+u^2/2} & u \geq 0 \\ \frac{1-u}{1-u+u^2/2} & u < 0 \end{cases}$$

and the narrowest member

$$\psi(u) := \begin{cases} \log(2) & u > 1 \\ -\log(1 - u + u^2/2) & u \in (0, 1] \\ \log(1 + u + u^2/2) & u \in [-1, 0] \\ -\log(2) & u < -1. \end{cases} \qquad \eta(u) = \begin{cases} 0 & u > 1 \\ \frac{1-u}{1-u+u^2/2} & u \in (0, 1] \\ \frac{1+u}{1+u+u^2/2} & u \in [-1, 0] \\ 0 & u < -1. \end{cases}$$

The wide version is not bounded above by a linear function, and the narrow version does not have strong convexity.

Fair function A choice whose practical utility was emphasized by Rey [18, Section 6.4.5], with parameter $c > 0$, is defined as

$$\rho(u) := c^2 \left(\frac{|u|}{c} - \log \left(1 + \frac{|u|}{c} \right) \right)$$

with derivatives

$$\psi(u) = \frac{u}{1 + \frac{|u|}{c}}, \quad \eta(u) = \frac{1 - \frac{|u|}{c}}{1 + \frac{|u|}{c}}.$$

and default value $c = 1.3998$. This function is not convex over $\mathbb{R}$.

Huber function The classic function originally proposed in the seminal work of Huber [13] is

$$\rho(u) := \begin{cases} c^2 \left(\frac{|u|}{c} - \frac{1}{2} \right) & |u| > c \\ u^2/2 & |u| \leq c \end{cases} \tag{A.4}$$

with associated functions

$$\psi(u) = \begin{cases} c \operatorname{sign}(u) & |u| > c \\ u & |u| \leq c \end{cases}, \quad \eta(u) = \begin{cases} 0 & |u| > c \\ 1 & |u| \leq c \end{cases}$$

and default setting of $c = 1.3450$ for consistency at the Normal model. Satisfies both asymptotic conditions, but does not have strong convexity.

Smooth Huber function A nice modification of the Huber function in order to secure continuous second-order derivatives is from Rey [18, Section 6.4.4], and defined by

$$\rho(u) := \begin{cases} c|u| + c^2(1-\pi/2) & |u| > c\pi/2 \\ c^2(1-\cos(u/c)) & |u| \leq c\pi/2 \end{cases}$$

with derivatives

$$\psi(u) = \begin{cases} c\,\mathrm{sign}(u) & |u| > c\pi/2 \\ c\sin(u/c) & |u| \leq c\pi/2 \end{cases}, \quad \eta(u) = \begin{cases} 0 & |u| > c\pi/2 \\ \cos(u/c) & |u| \leq c\pi/2 \end{cases}$$

all implemented as `hmod`. Default is $c = 1.2107$. Once again, strong convexity does not hold on $\mathbb{R}$ in this case.

A.1.2 Scale

The issue of "re-scaling" is a rather subtle one, though the precise operations carried out are typically very simple. Note that in setting $s > 0$ based on either prior knowledge or the observed data, assuming valid ρ, the impact of s on the resulting estimate is clear, since we have

$$\arg\min_{u\in\mathbb{R}} \mathbf{E}\,\rho\left(\frac{x-u}{s}\right) = \arg\min_{u\in\mathbb{R}} \mathbf{E}\,\rho\left(\frac{x}{s}-u\right)s. \tag{A.5}$$

It is worth observing that this reflects some implicit assumptions about the scale, the underlying distribution, and the function ρ. Evidently, it assumes that the location of the distribution of x can be readily approximated by re-scaling the ρ-based location estimate of a "baseline" observation, namely the original normalized by a factor of $1/s$. More concretely, if we denote

$$\widehat{x} := \arg\min_{u\in\mathbb{R}} \mathbf{E}\,\rho\left(\frac{x-u}{s}\right), \quad \widehat{x}_0 := \arg\min_{u\in\mathbb{R}} \mathbf{E}\,\rho\left(\frac{x}{s}-u\right) \tag{A.6}$$

it assumes that

- the location of observation x scales with it;
- $\widehat{x}_0$ is a *good* estimate of the location of x/s to begin with.

If both of these assumptions hold, then since $\widehat{x}_0$ effectively reflects location, and as this location scales, then $\widehat{x} = s\widehat{x}_0$ must be a good location estimate of $x = (x/s)s$. The first assumption is by no means trivial in general, but in the special case of seeking the mean directly, since $\mathbf{E}(x/s) = (\mathbf{E}\,x)/s$ by linearity of the integral, it is satisfied. The second assumption is the chief challenge here, and naturally depends on what is meant by location. As to actually measuring the scale, an analogous class of M-estimators can be readily used:

$$\widehat{s} = \inf\left\{s > 0 : \frac{1}{n}\sum_{i=1}^{n}\chi\left(\frac{x_i}{s}\right) = 0\right\} \tag{A.7}$$

Here $\chi : \mathbb{R} \to \mathbb{R}$ is typically non-decreasing on $\mathbb{R}_+$, even, and takes on both positive and negative values over its range. This intuitively forces the scale estimate not to be too large or too small, given the values taken by the observations $x_1, \ldots, x_n$ used in this measurement. This approach has roots dating back well over a half-century [13, 20, 19], though we have made several modifications in chapters 3–4 when making use of this approach for our tasks of interest. To make things more concrete, we give some useful examples of χ below. The value $\beta > 0$ is a constant parameter.

Andrews function

$$\chi(u) := \begin{cases} 2c^2\left(1-\cos\left(u/c\right)\right)-\beta & |u| \leq c\pi \\ 4c^2-\beta & |u| \geq c\pi. \end{cases} \tag{A.8}$$

This function was featured in the famous Princeton study on robust statistics in the early 1970s, summarized in [2, Section 2C3]. The precise form given here is from Rey [18, Section 6.4.9], with default parameter value $c = 1.3387$. Alternate versions using $\cos^2$ have also appeared in the computer vision literature [3].

Dennis-Welsh function

$$\chi(u) := c^2\left(1-\exp\left(-u^2/c^2\right)\right)-\beta \tag{A.9}$$

Default parameter value $c = 2.9846$ [9, 18]. Since this is just a re-scaled and shifted negative Normal density, this kind of function has also been featured prominently in nonparametric statistical literature [23].

Geman-type functions

$$\chi(u) := \frac{|u|}{1+|u|}-\beta \tag{A.10}$$

$$\chi(u) := \frac{u^2}{1+u^2}-\beta \tag{A.11}$$

These originate in highly-cited image processing literature [11, 10], are included in the comprehensive robust computer vision methods outlined by Black and Rangarajan [3], and more recently in the context of fast robust regression by Yu et al. [24].

Huber's proposal 2 Originally given in Huber [13, Section 11] in the context of simultaneous estimation of both location and scale parameters, one has

$$\chi(u) := \min(u^2, c^2) \tag{A.12}$$

with a default parameter of $c = 1.5$ [22, Chapter 5].

Median absolute devations One important classical choice is

$$\chi(u) := \operatorname{sign}(|u|-1) \tag{A.13}$$

which for arbitrary γ_μ, induces the explicit scale estimate

$$\widehat{s} = \operatorname{med}_\mu |x-\gamma_\mu|, \tag{A.14}$$

namely the median absolute deviations about γ_μ. In particular, the case where $\gamma_\mu = \mu^{-1}(0.5)$, that is the MAD about the median, was discussed in detail by Andrews et al. [2], and hailed as a very useful robust scale estimate for use in more general location problems. In our tests this is `madmed`, while the MAD about the mean is `madmean`.

Transformed robust loss derivative (quad, lquad) Let ρ be an even function, and assume that it is valid according to the conditions of the previous sub-section. Then defining

$$\chi(u) := \psi^2(u) - \beta \tag{A.15}$$

$$\chi(u) := \log(1 + \psi^2(u)) - \beta \tag{A.16}$$

we have a general-purpose class of scale estimates, the first of which is a generalization of "proposal 2" discussed above.

Biweight antiderivative (tukey)

$$\chi(u) := \begin{cases} \frac{c^2}{6} - \beta & |u| \geq c \\ \frac{x^6}{6c^4} - \frac{x^4}{2c^2} + \frac{x^2}{2} - \beta & |u| < c \end{cases} \tag{A.17}$$

The derivative of this function is the well-known function known in the literature as Tukey's bi-weight. Default value $c = 1.547$ [19, p. 261].

Additional choices Here we include some miscellaneous choices for looking at dispersion, aside from the χ-based M-estimation framework. Some are classical, some are rather modern. Note that in our use of pivot term γ_μ above, given the sample we shall have to compute γ_{μ_n}, an initial estimate of "central tendency," and then measure dispersion of the data using this point as a reference. Of course the standard deviation $s_\mu = (\mathbf{E}_\mu(x - \mathbf{E}_\mu\, x)^2)^{1/2}$ is the canonical example, and so is MAD given above. On the other hand, some methods do not require such an ancillary estimate, and instead look at the range spanned by "most" of the data. Perhaps the most ubiquitous ideal estimate comes from the interquartile range, defined here as

$$s := \mu^{-1}(0.75) - \mu^{-1}(0.25). \tag{A.18}$$

Naturally this can be generalized to arbitrary range as $\mu^{-1}(p_2) - \mu^{-1}(p_1)$ with $0 < p_1 < p_2 < 1$. This is a simple and useful tool for roughly capturing the range over which large portions of the data are captured, which only requires a sort of our n observations, and can be done efficiently [6, Section 2]. Two more recent empirical proposals come from Rousseeuw and Croux [21], focusing on pairwise deviations. The first, called S_n in their work, is

$$\widehat{s} := c\,\mathrm{med}\{\mathrm{med}\{|x_i - x_j| : j \in [n]\} : i \in [n]\} \tag{A.19}$$

with default value $c = 1.1926$. The second, called Q_n in their work, is

$$\widehat{s} := c\{|x_i - x_j| : i < j\}_{(k)}, \tag{A.20}$$

where (k) here denotes the kth order statistic of this set of pairwise deviations, and default factor $c = 2.2219$. For both of these computations, algorithms requiring just $O(n \log(n))$ time have been proposed [8], with additional bias corrections for very small values of n.

A.2 Figures (Ch. 3): Prediction error for all noise classes

This section includes supplementary figures for chapter 3, including results for additional families of noise distributions.

A.2.1 Over noise levels

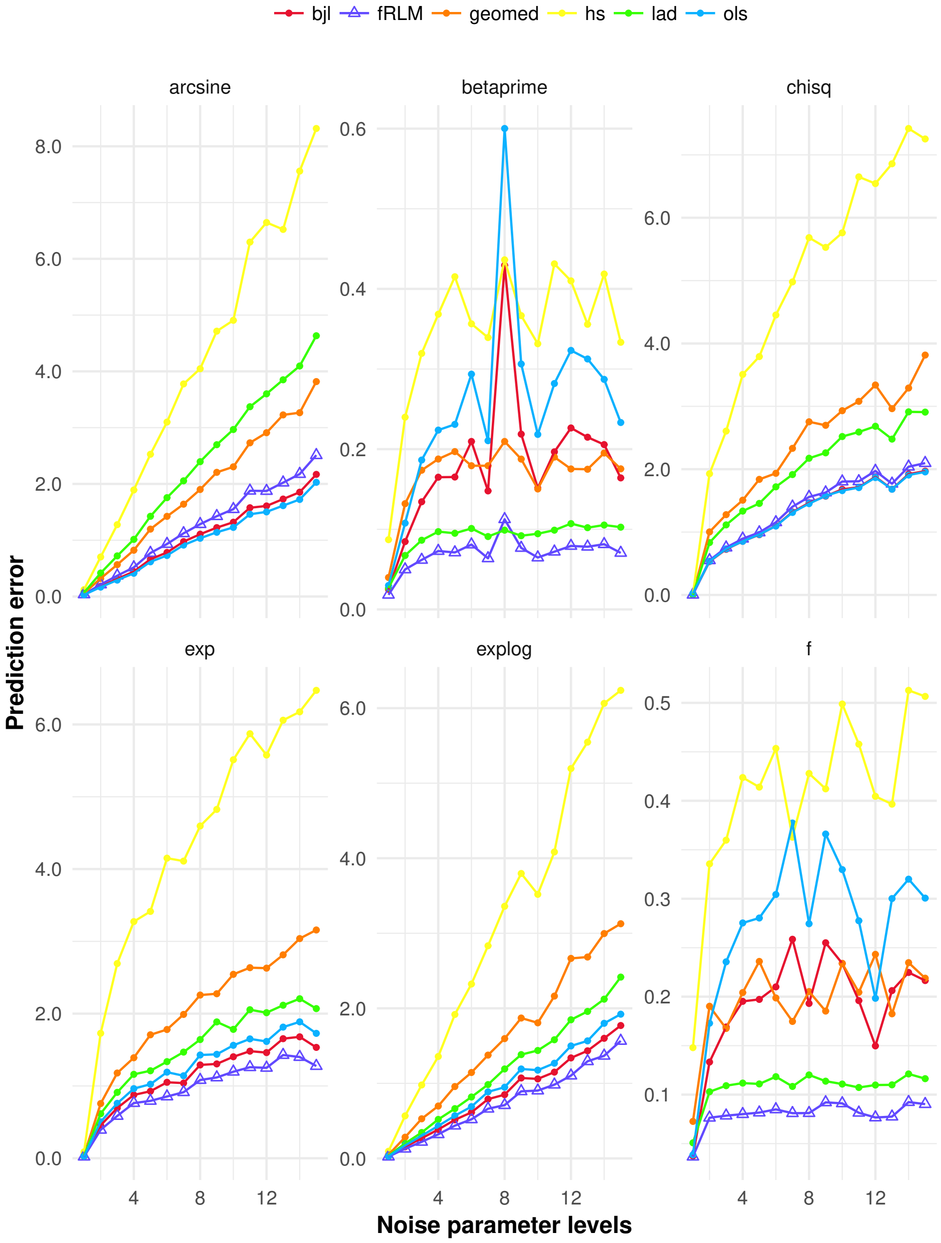

Figure A.3: Average error, over levels, for $n = 30, d = 5$.

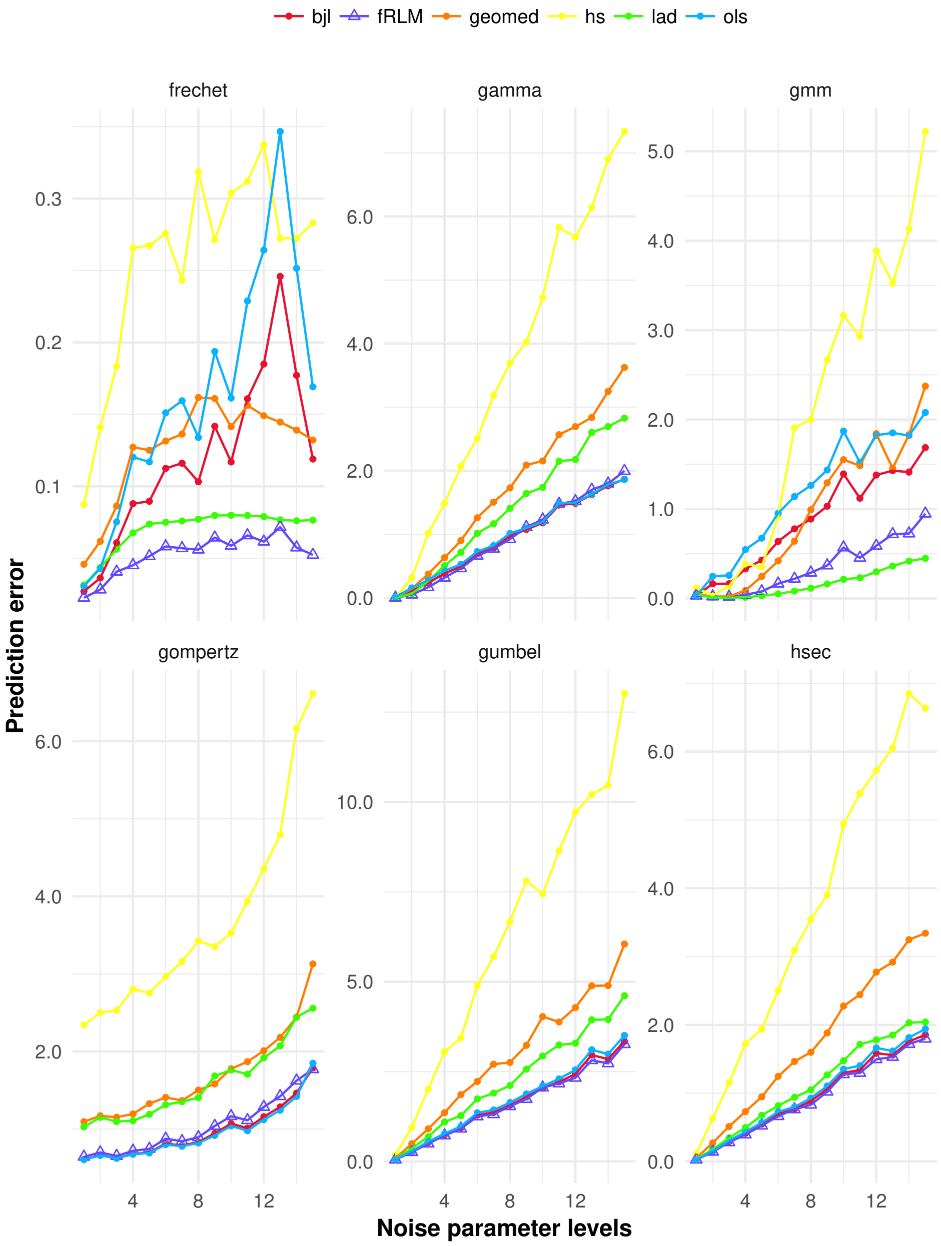

Figure A.4: Average error, over levels, for $n = 30, d = 5$.

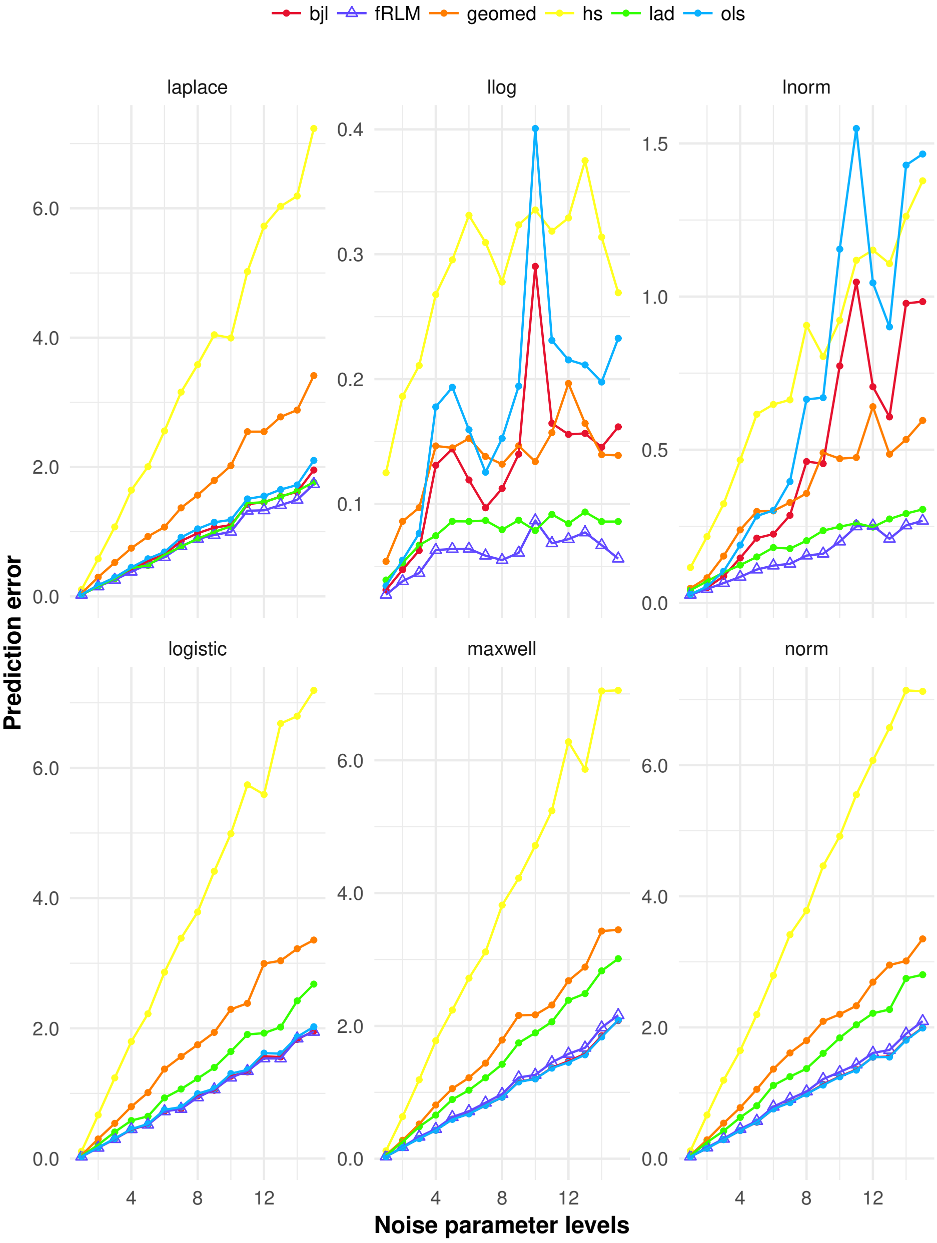

Figure A.5: Average error, over levels, for $n = 30, d = 5$.

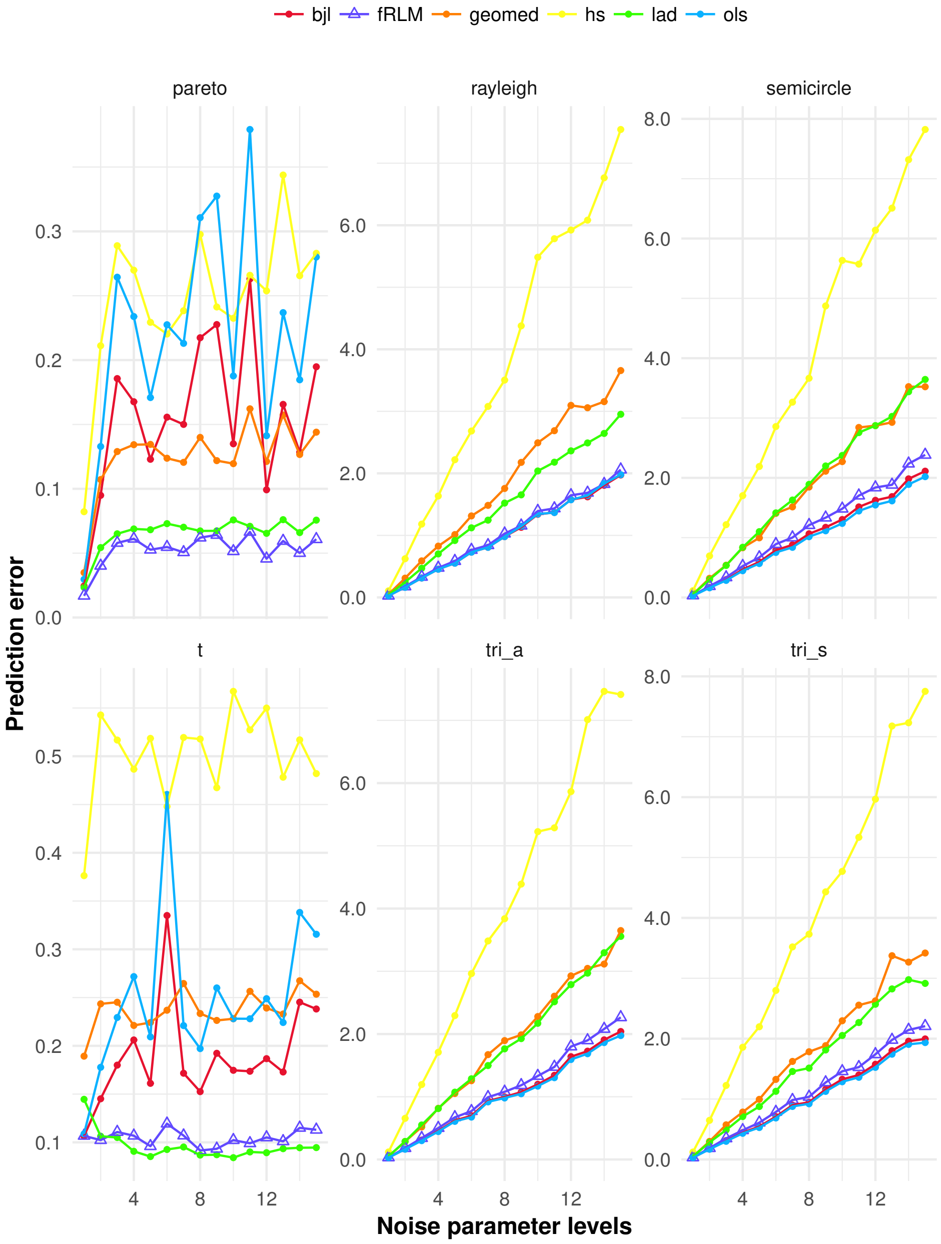

Figure A.6: Average error, over levels, for $n = 30, d = 5$.

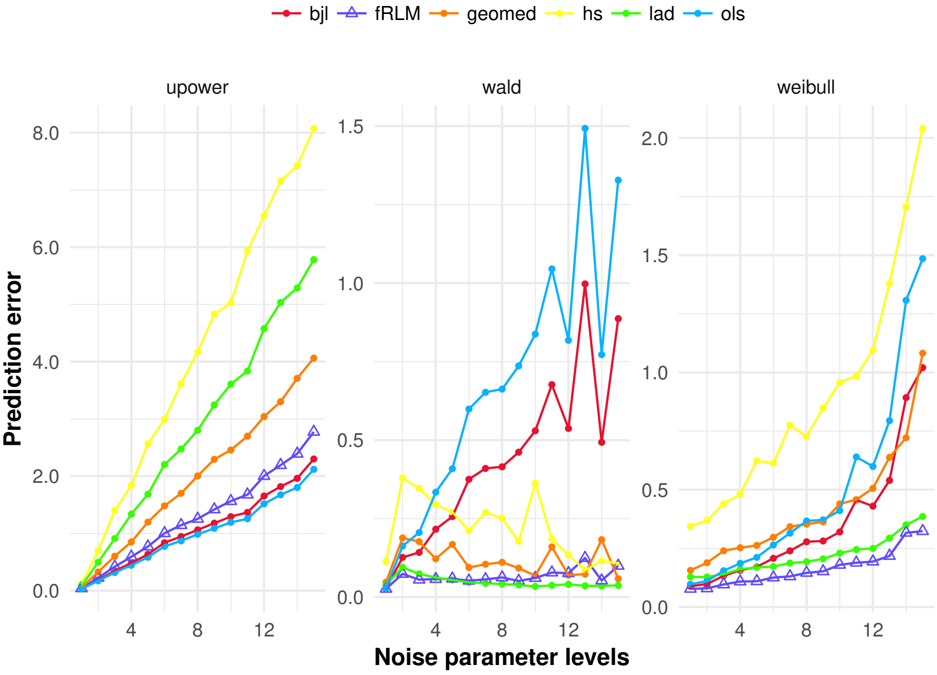

Figure A.7: Average error, over levels, for $n = 30, d = 5$.

A.2.2 Over sample size

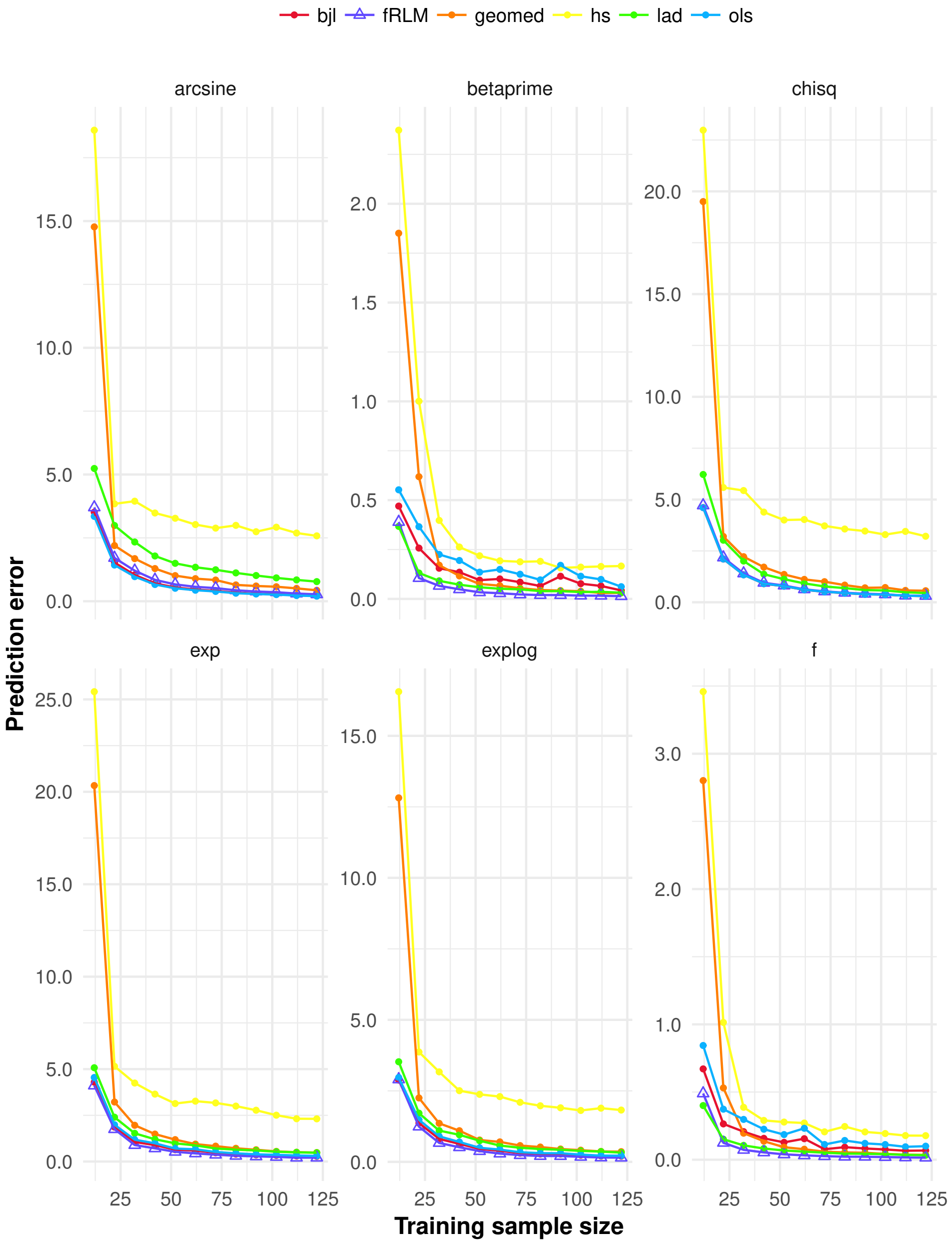

Figure A.8: Average error over n sizes, all methods. Level = 8.

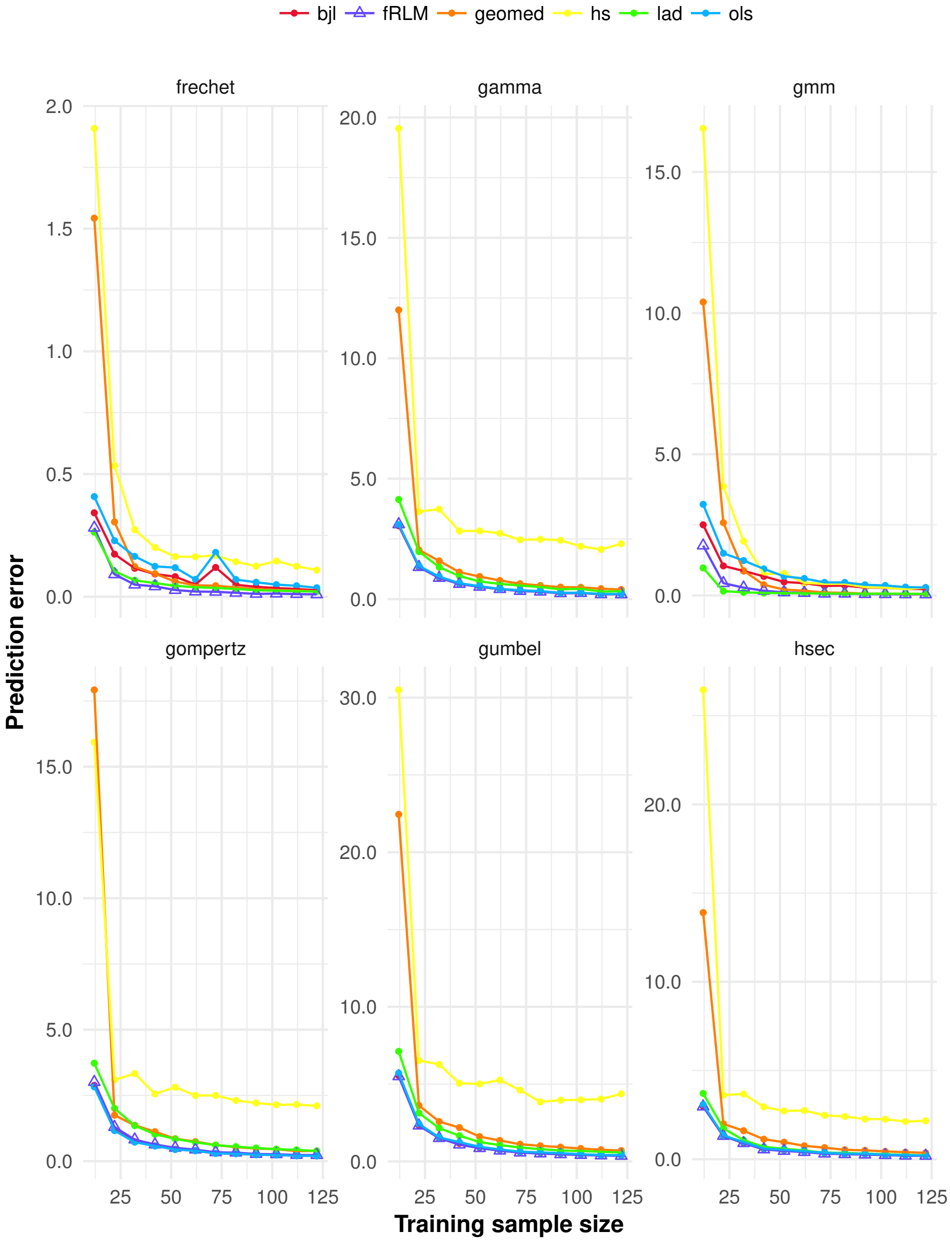

Figure A.9: Average error over n sizes, all methods. Level = 8.

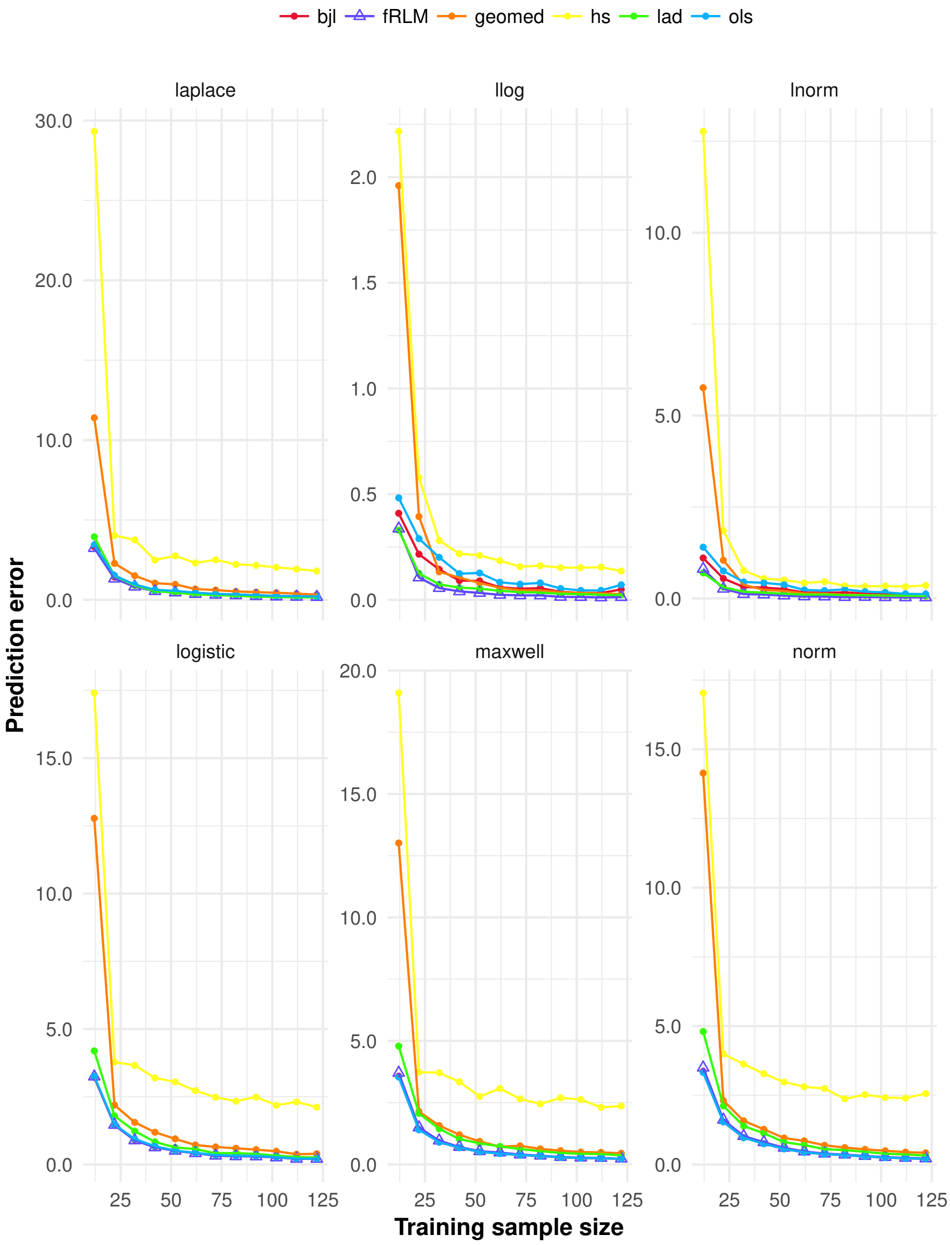

Figure A.10: Average error over n sizes, all methods. Level = 8.

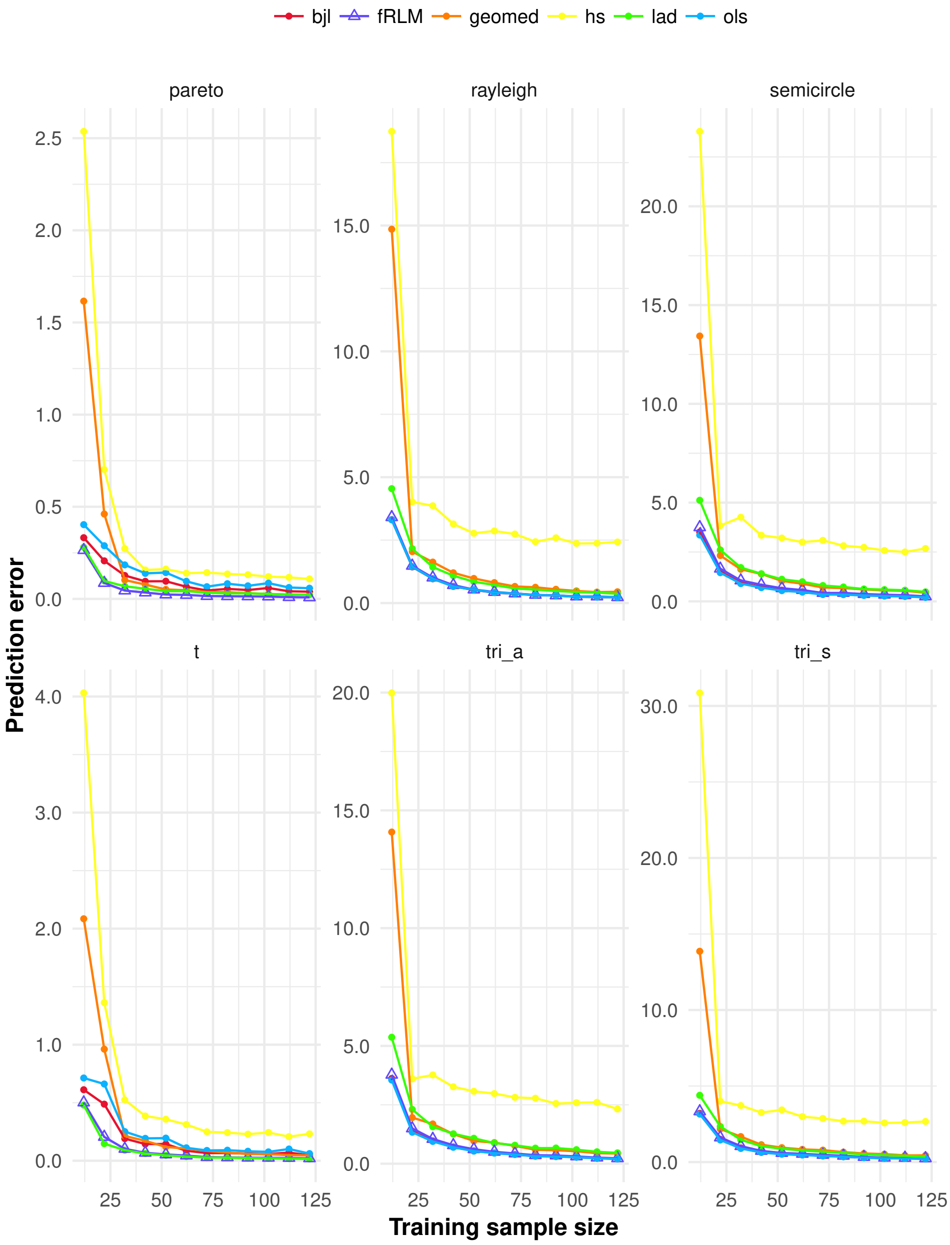

Figure A.11: Average error over n sizes, all methods. Level = 8.

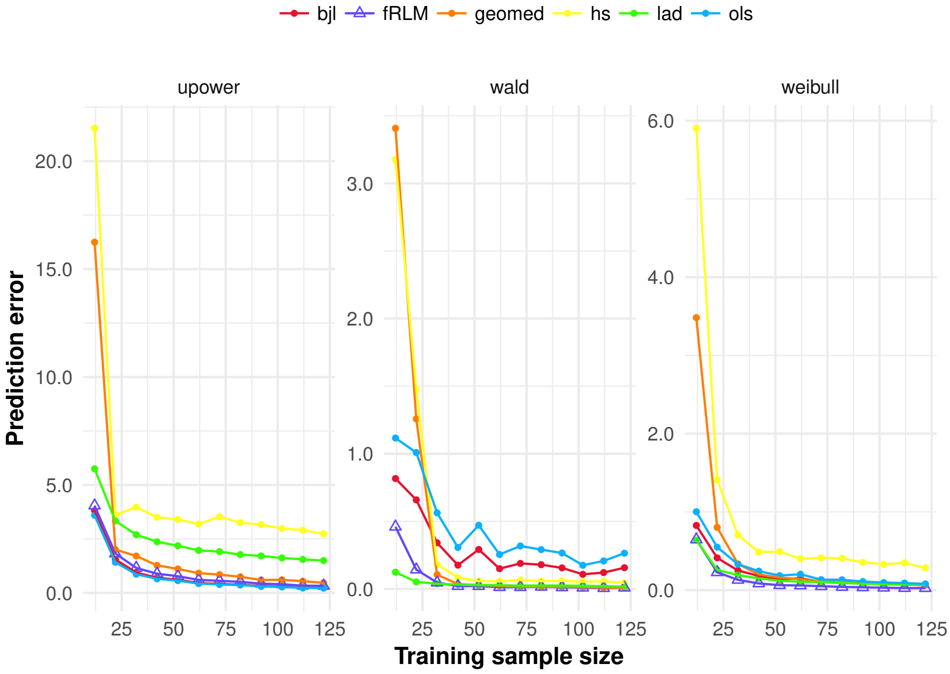

Figure A.12: Average error over n sizes, all methods. Level = 8.

A.2.3 Over dimension

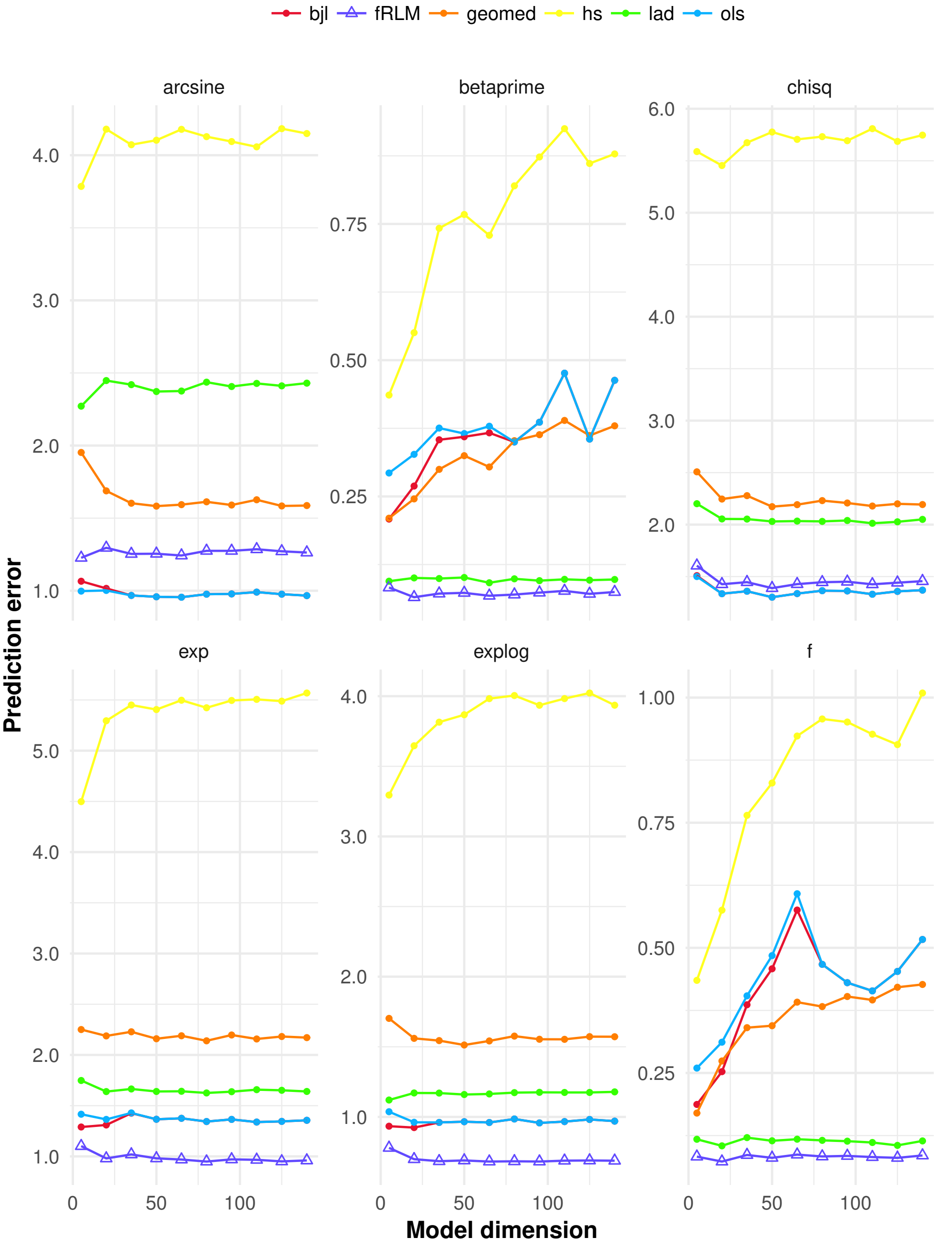

Figure A.13: Average error over dimension d, all methods. Level = 8.

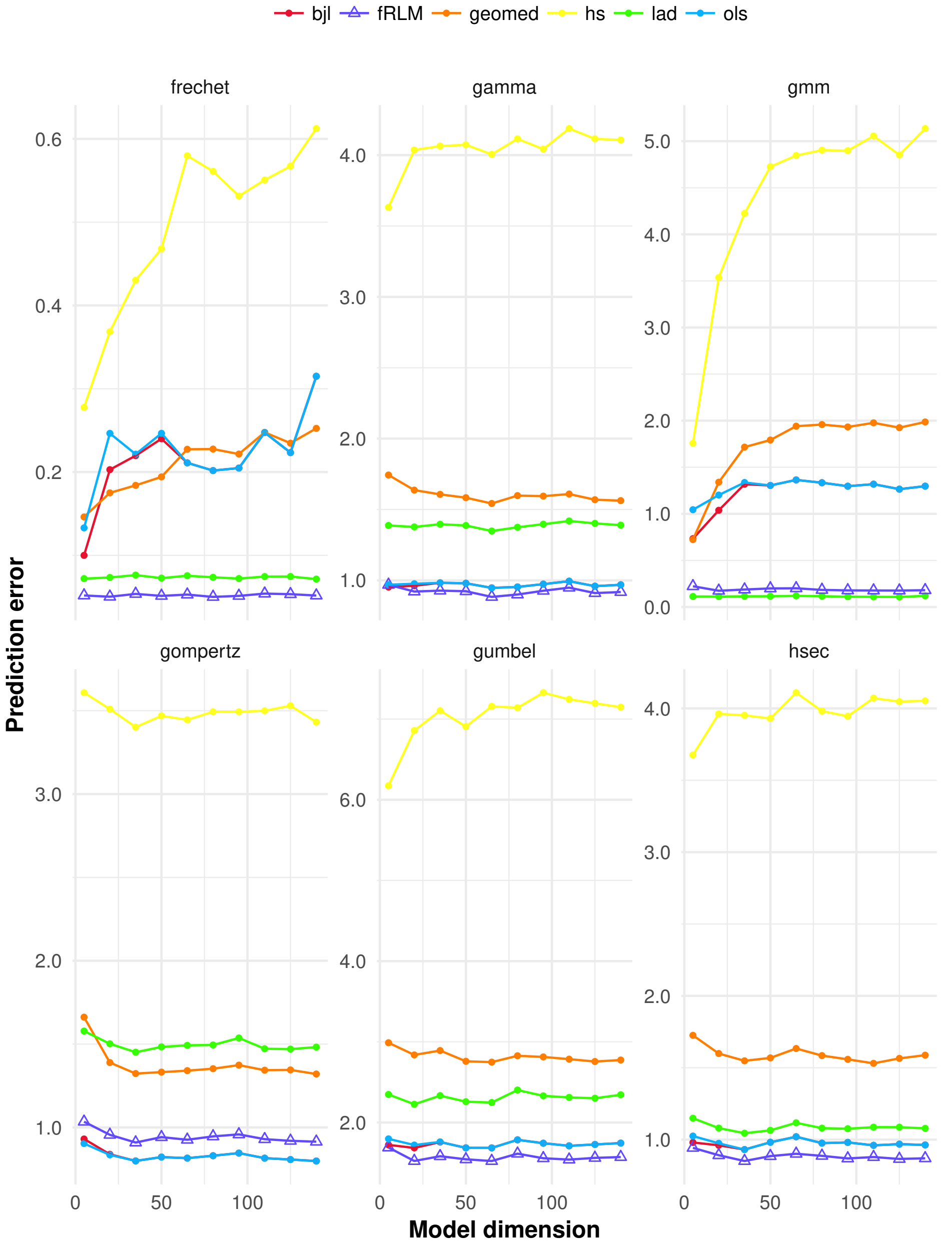

Figure A.14: Average error over dimension d, all methods. Level = 8.

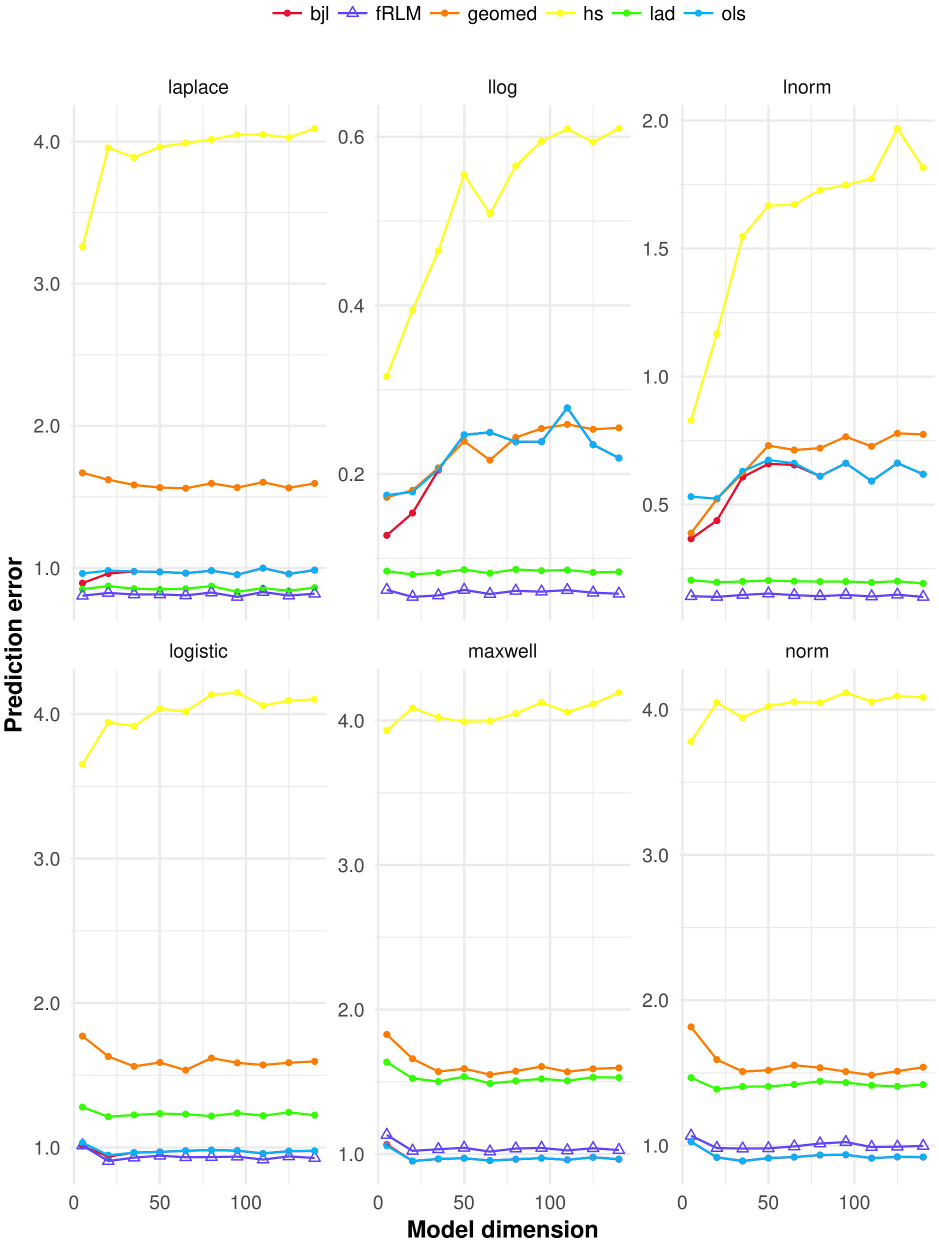

Figure A.15: Average error over dimension d, all methods. Level = 8.

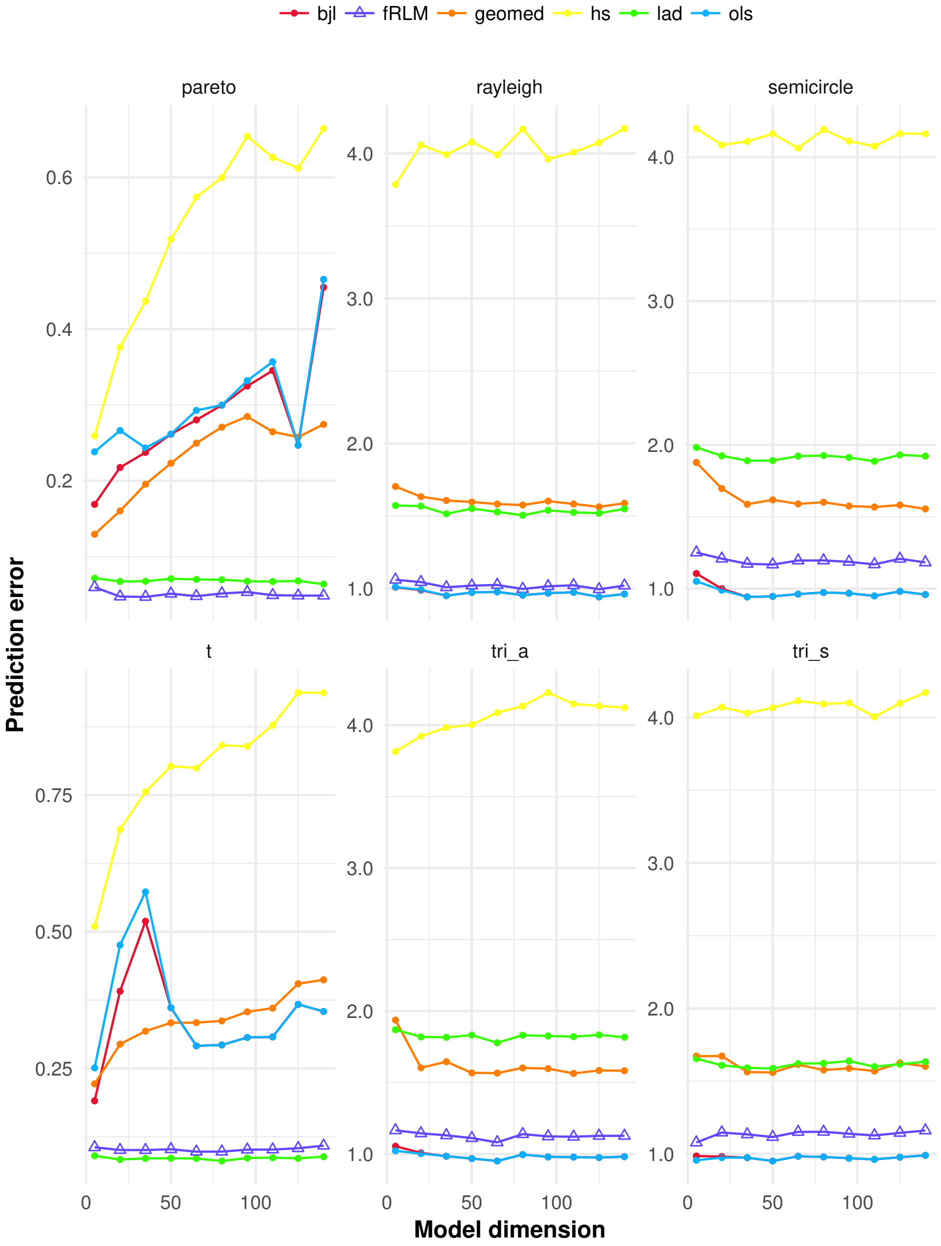

Figure A.16: Average error over dimension d, all methods. Level = 8.

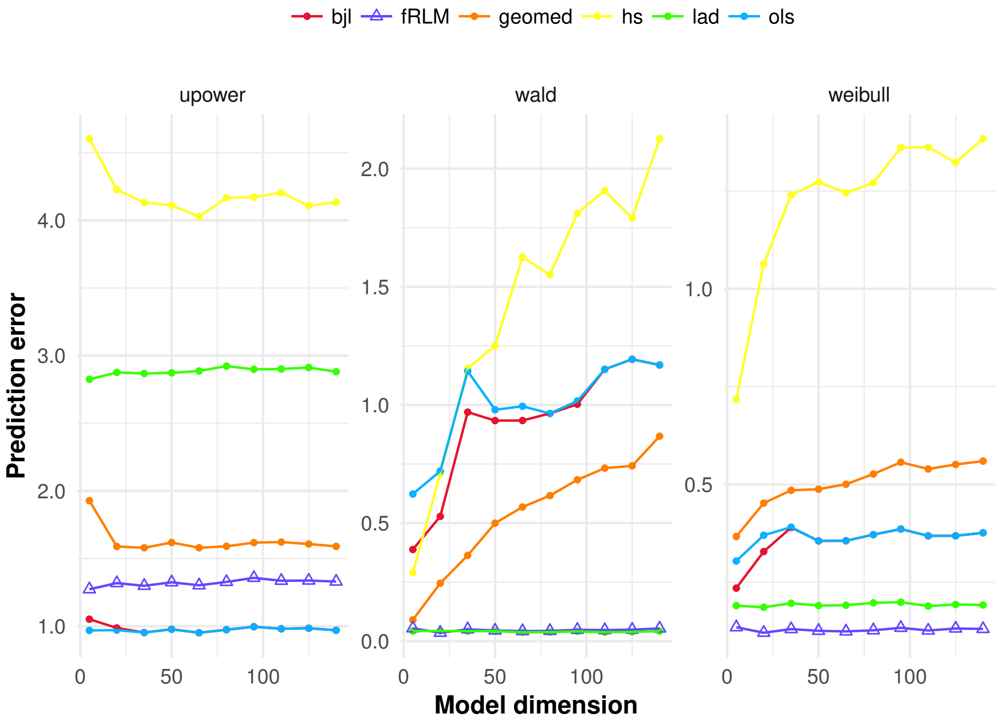

Figure A.17: Average error over dimension d, all methods. Level = 8.

A.3 Figures (Ch. 4): Risk trajectory and task parameters

This section includes figures given in chapter 4 in larger size, as well as additional figures that were excluded for readability.

A.3.1 Light/heavy-tailed samples

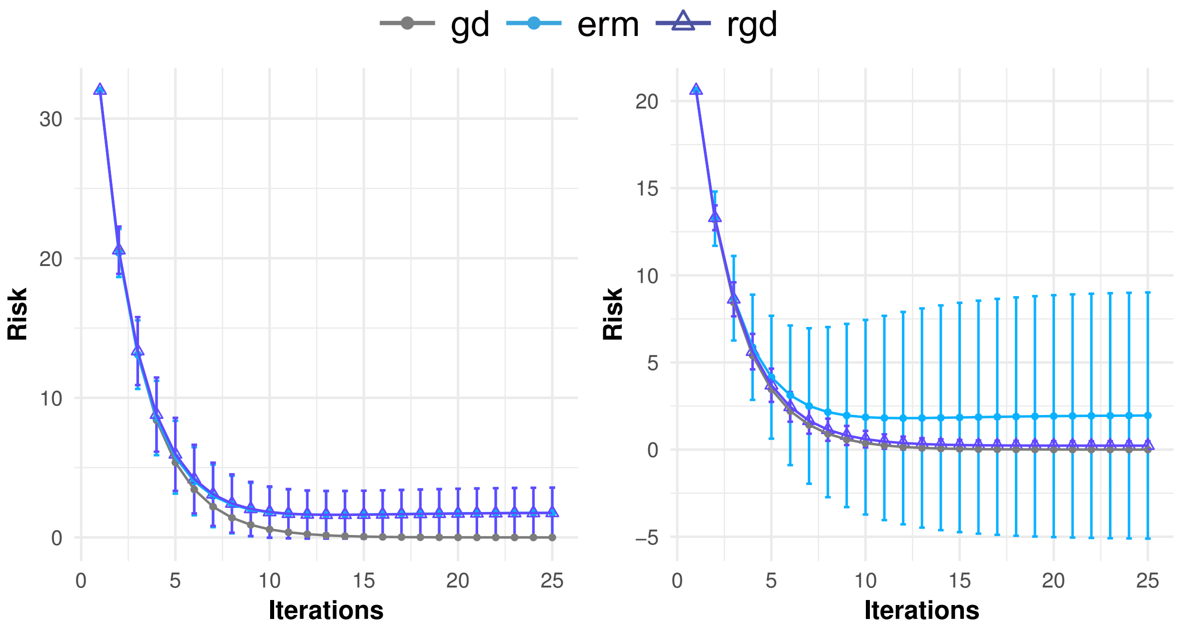

Figure A.18: Risk as a function of iterations after a common initial point. Each trial corresponds to a new random sample, and all values on the vertical axis are averaged over 250 trials. Error bars are standard deviation over trials. Left is Normal noise, right is log-Normal noise. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

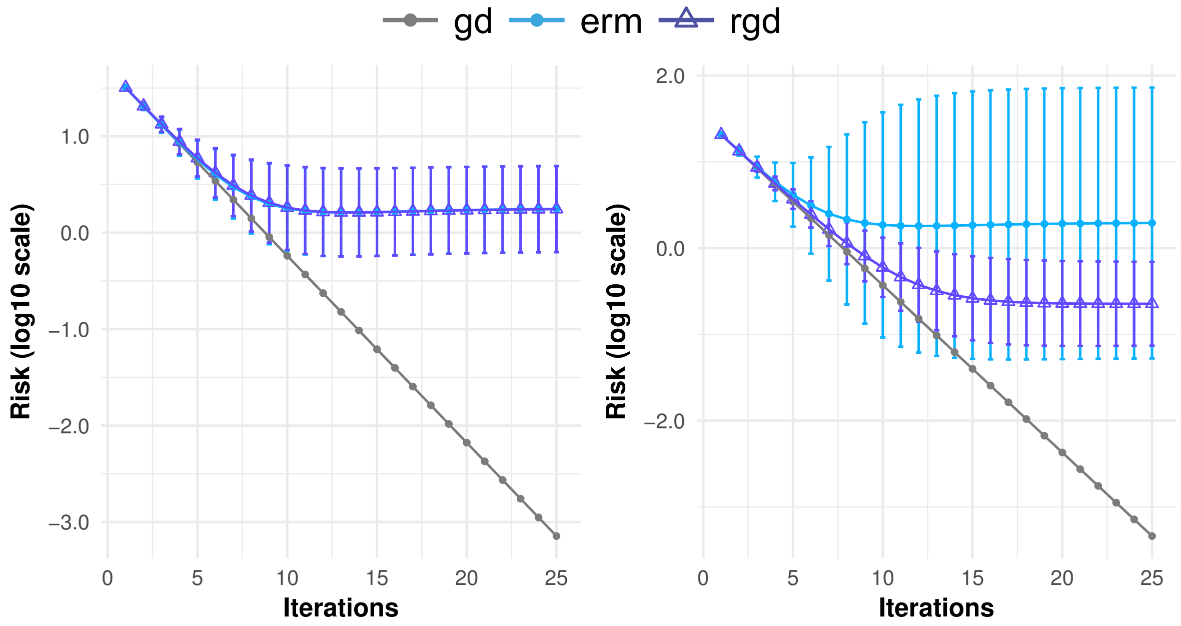

Figure A.19: Results of Figure A.18, given in $\log_{10}$ scale. Error bars, say $\log_{10}(u) \pm \delta$, are computed as "relative error," namely $\delta = \mathrm{sd}(u)/(u \log(10))$.

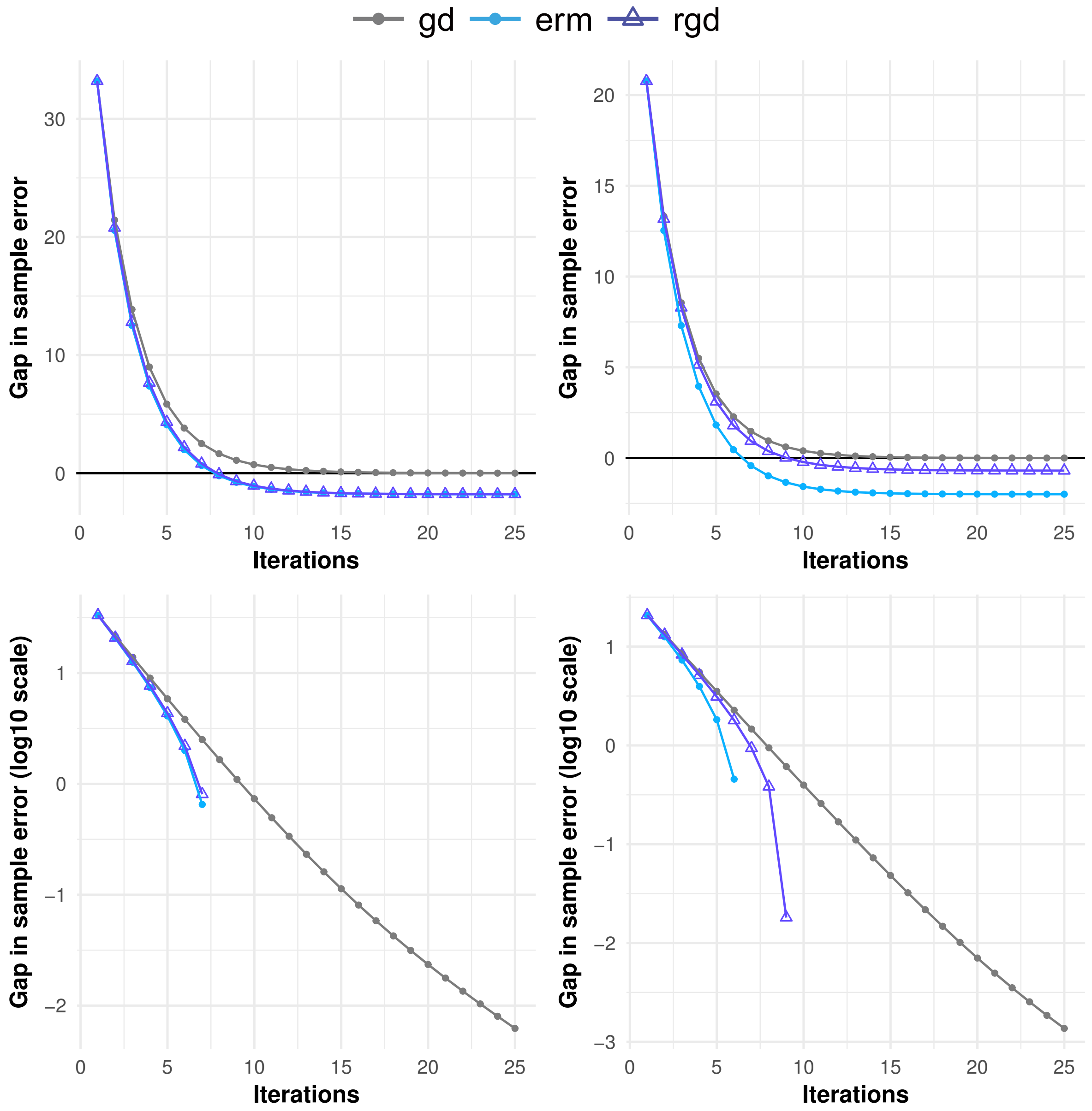

Figure A.20: Sample error gap for the tests shown in Figure A.18, computed as $\widehat{R}(\widehat{\boldsymbol{w}}) - \widehat{R}(\boldsymbol{w}^{*})$, with $\widehat{R}(\boldsymbol{w}) = n^{-1}\sum_{i=1}^{n} l(\boldsymbol{w};\boldsymbol{z}_i)$. First row is original coordinates, second row is $\log_{10}$ scale. The latter graphs stop once the gap (before log transform) becomes negative. Left is Normal noise, right is log-Normal noise.

A.3.2 Impact of initialization

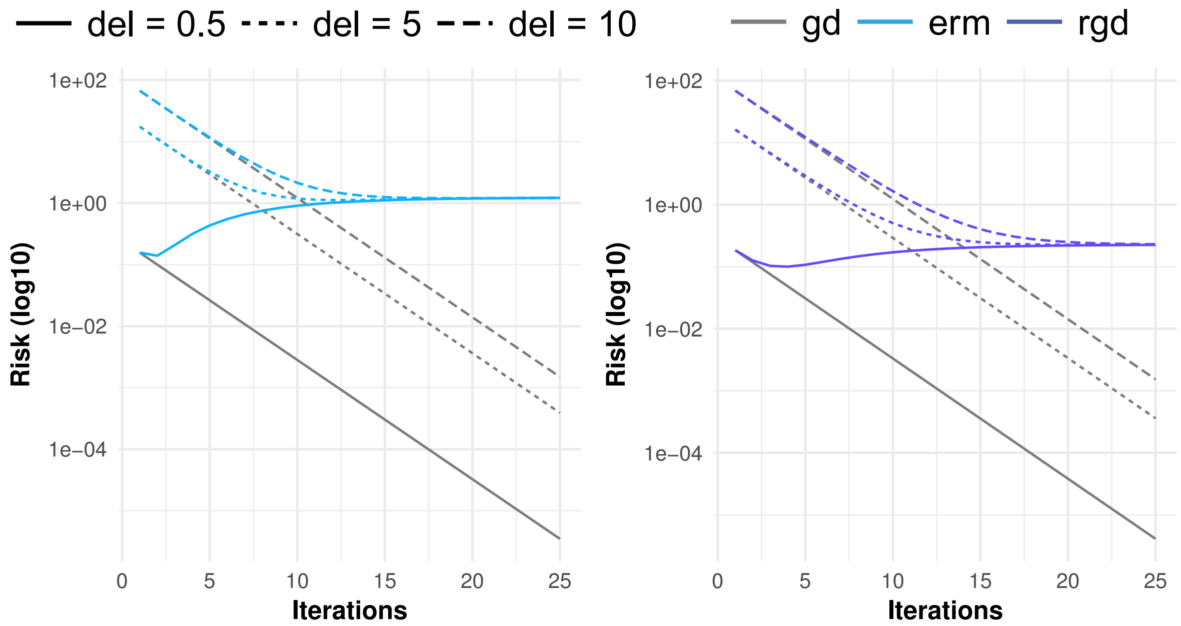

Figure A.21: Risk as a function of iterations, in $\log_{10}$ scale. Values are averaged over 250 trials. The "del" refers to Δ_j. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

A.3.3 Impact of distribution

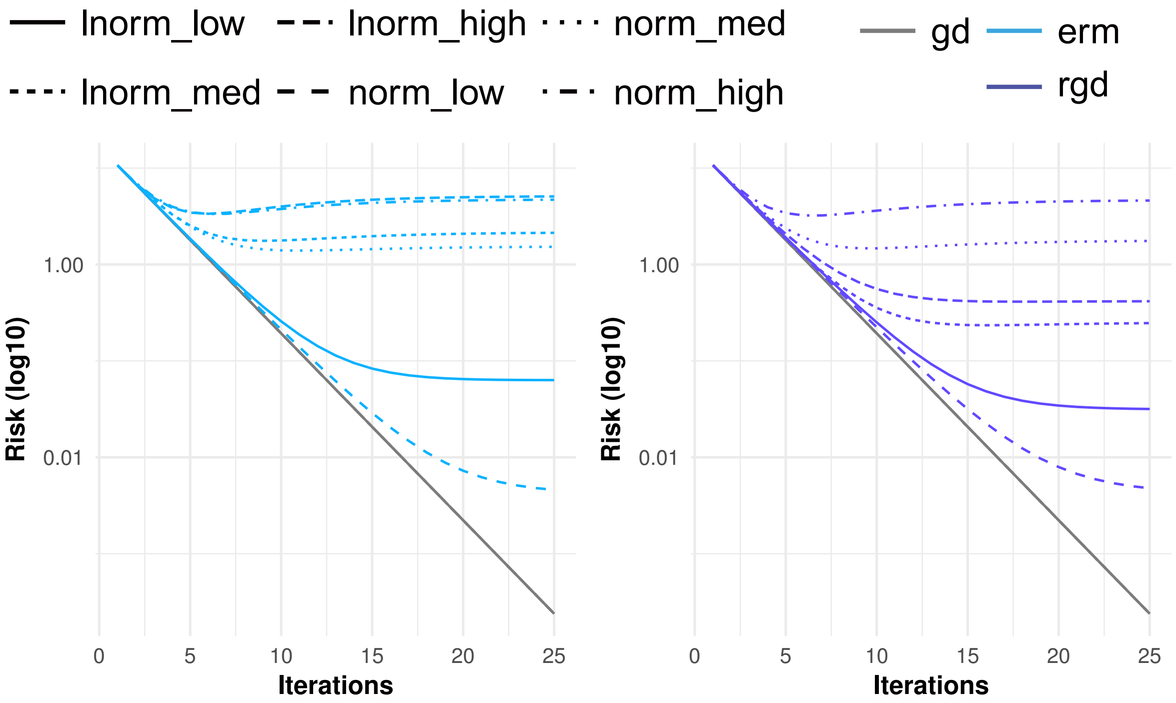

Figure A.22: Risk as a function of iterations, in $\log_{10}$ scale. Values averaged over 250 trials. Here “lnorm” and “norm” denote log-Normal and Normal samples, with “low,” “med,” and “high” designating the three parameter settings. $n = 500, d = 2, \alpha_{(t)} = 0.1$ for all t.

A.3.4 Impact of sample size

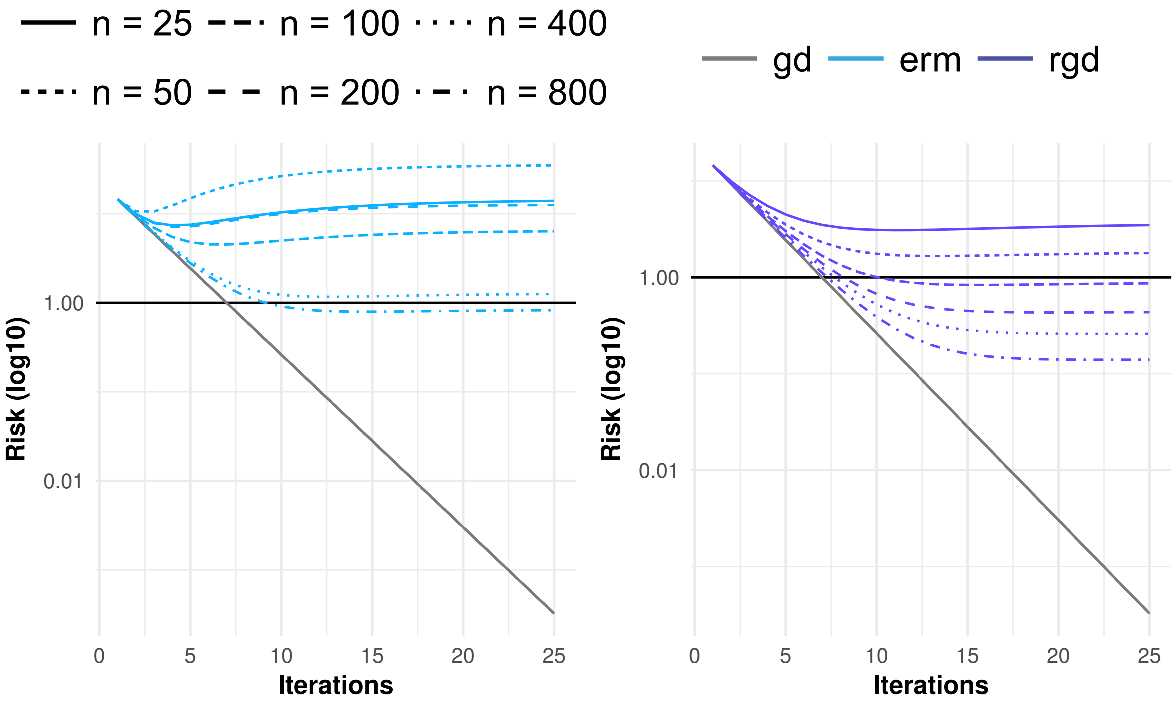

Figure A.23: Risk as a function of iterations, in $\log_{10}$ scale. Values are averaged over 250 trials. The black horizontal rules are set for reference between the two plots, whose coordinates differ slightly. $d = 2, \alpha_{(t)} = 0.1$ for all t.

A.3.5 Impact of dimension

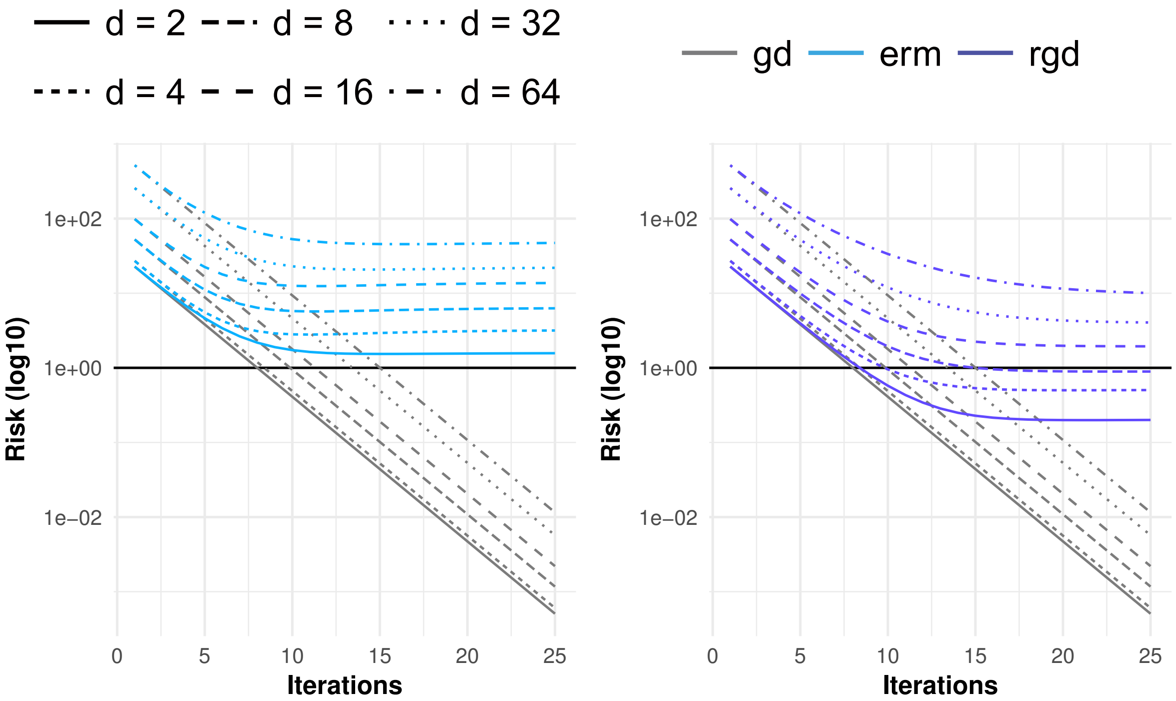

Figure A.24: Risk as a function of iterations, in $\log_{10}$ scale. Values averaged over 250 trials. The black horizontal rules are set for reference between the two plots, whose coordinates differ slightly. $n = 500, \alpha_{(t)} = 0.1$ for all t.

A.3.6 Against implicit robust loss minimizers

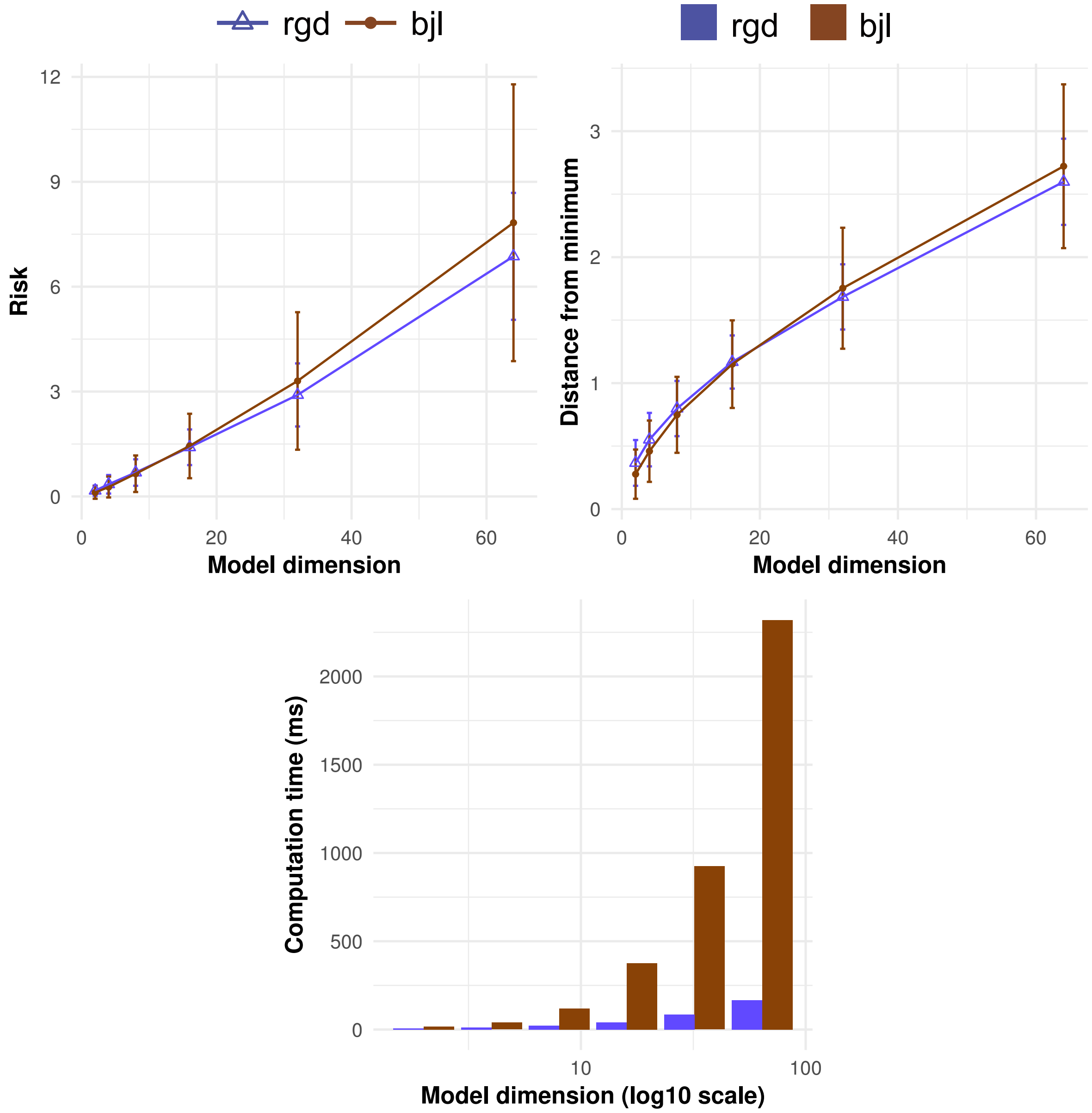

Figure A.25: Performance as a function of the number of parameters to optimize. Top: excess risk and ℓ_2 distance from the minimum. Bottom: computation times. Performance values and computation times are averaged over 250 trials. $n = 500, \alpha_{(t)} = 0.1$ for all t.

A.4 Figures (Ch. 4): Prediction error for all noise classes

This section includes supplementary figures for chapter 4, including results for additional families of noise distributions.

A.4.1 Over noise levels

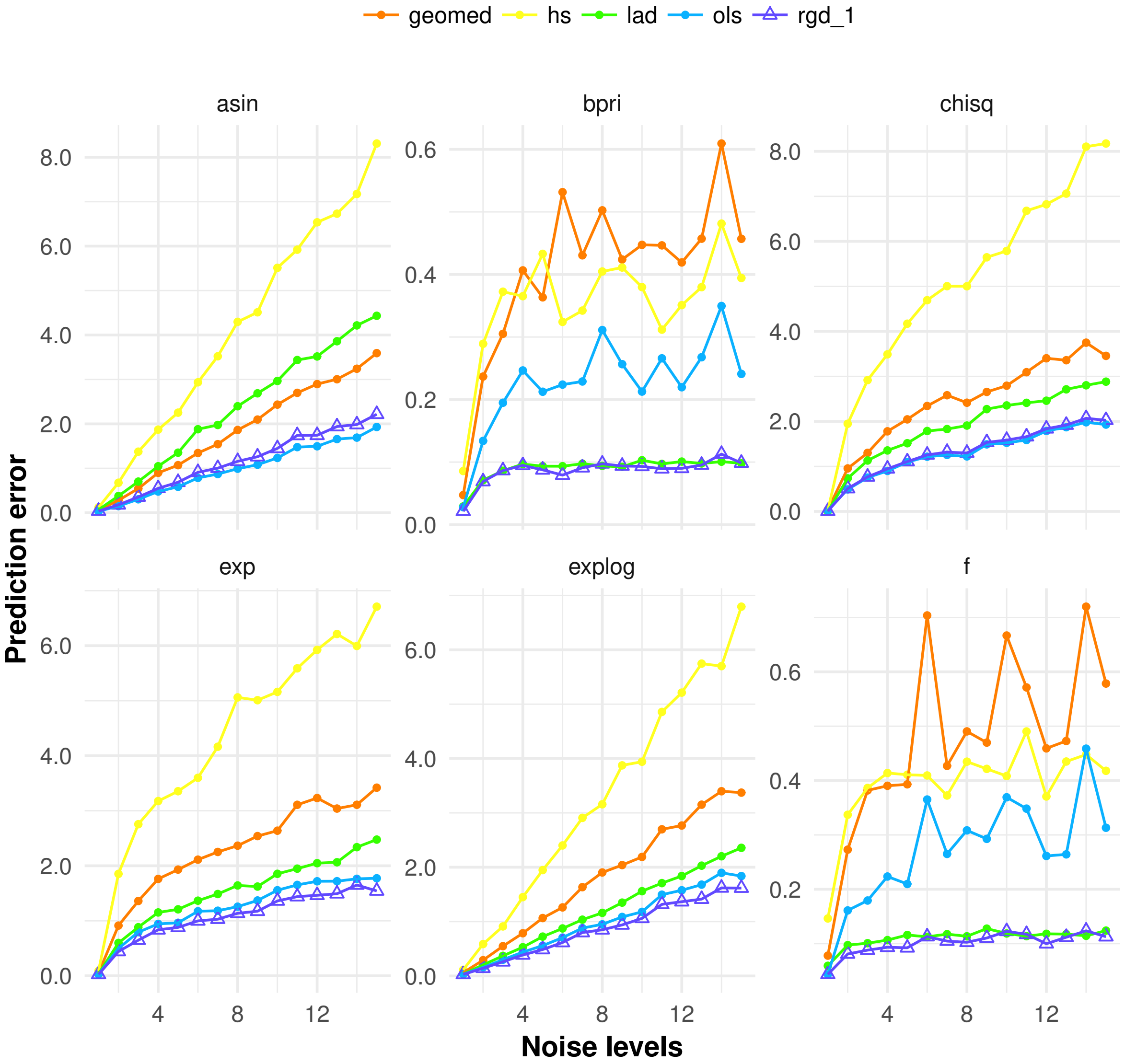

Figure A.26: Average error, over levels, for $n = 30, d = 5$.

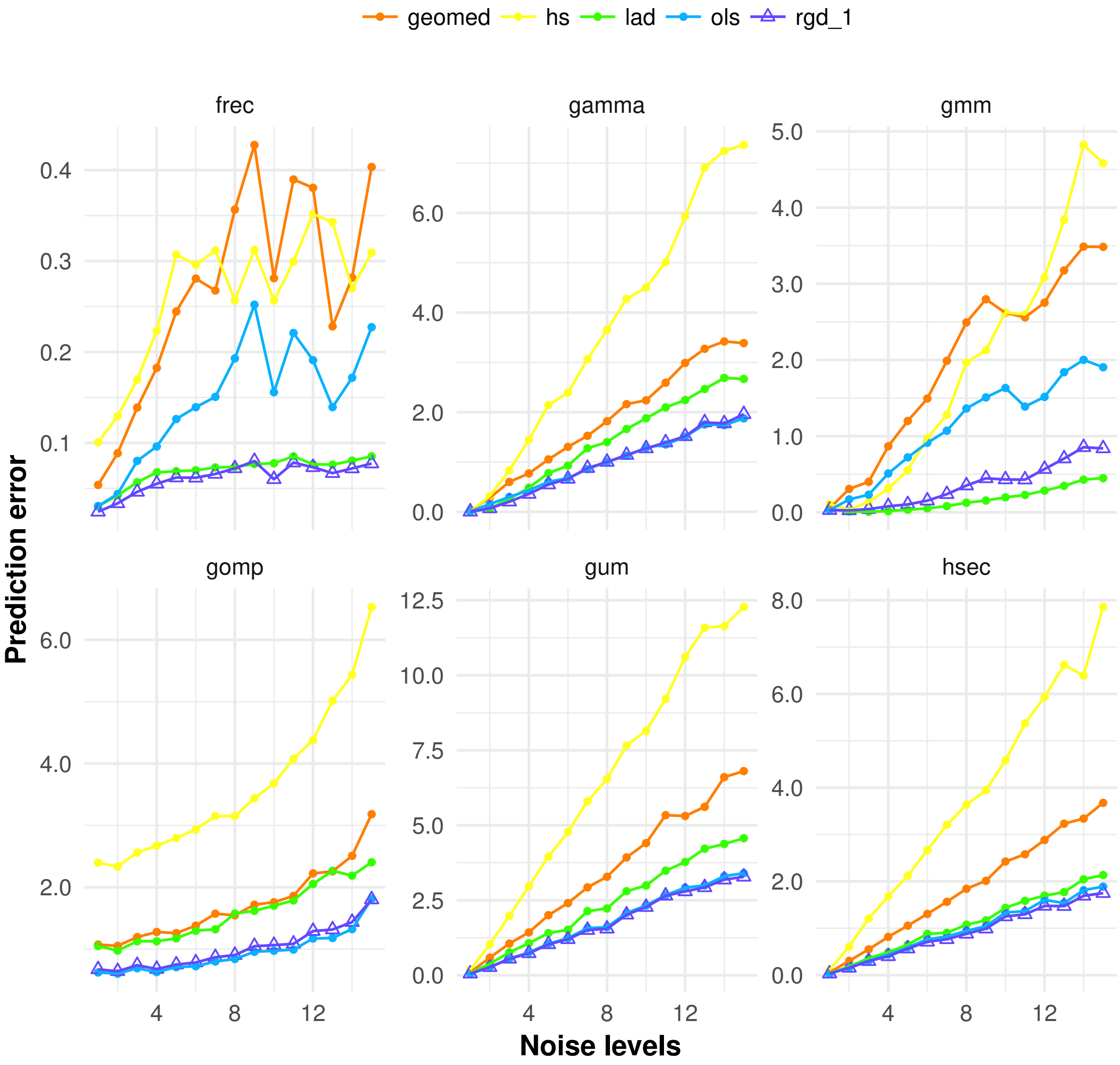

Figure A.27: Average error, over levels, for $n = 30, d = 5$.

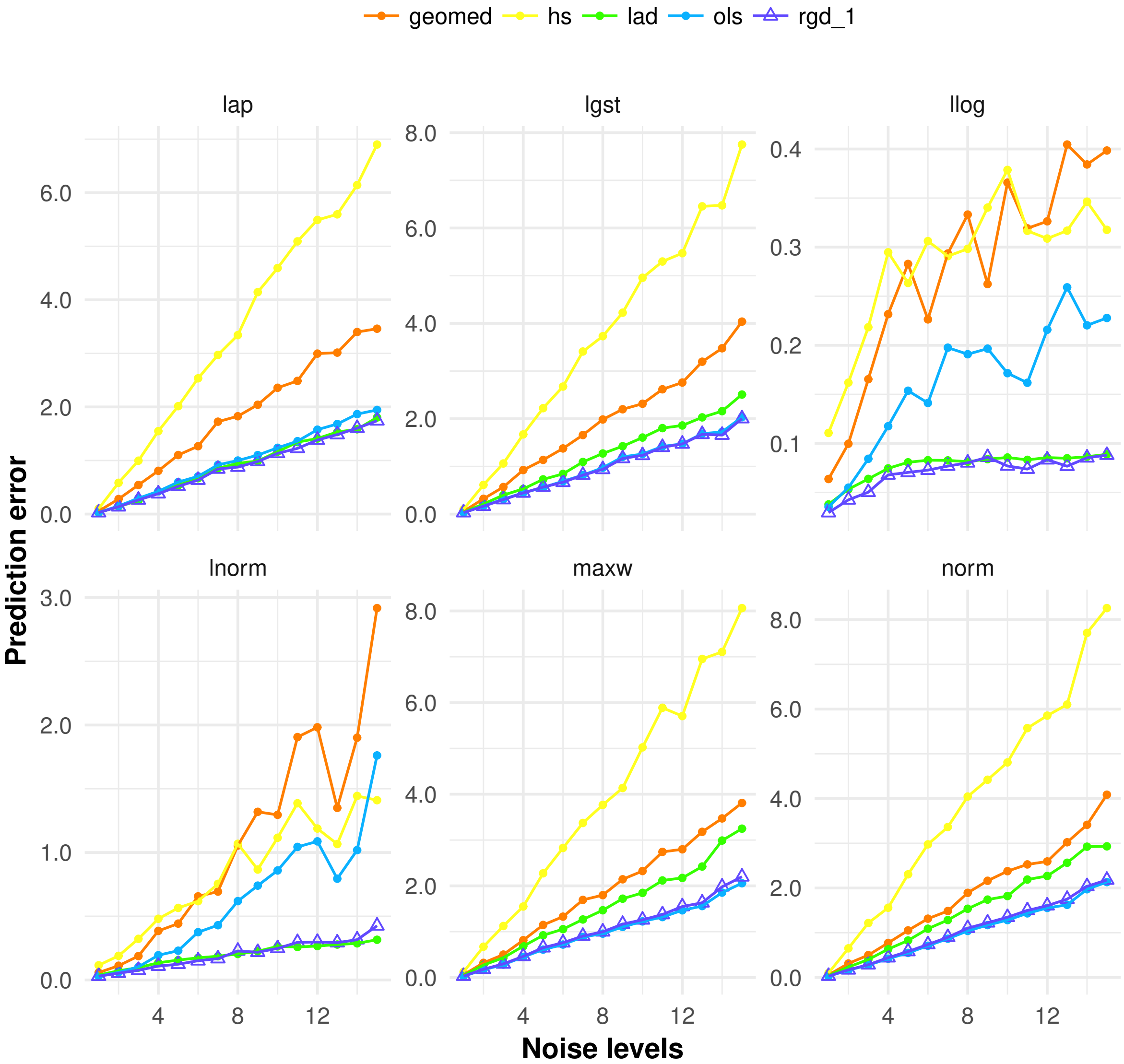

Figure A.28: Average error, over levels, for $n = 30, d = 5$.

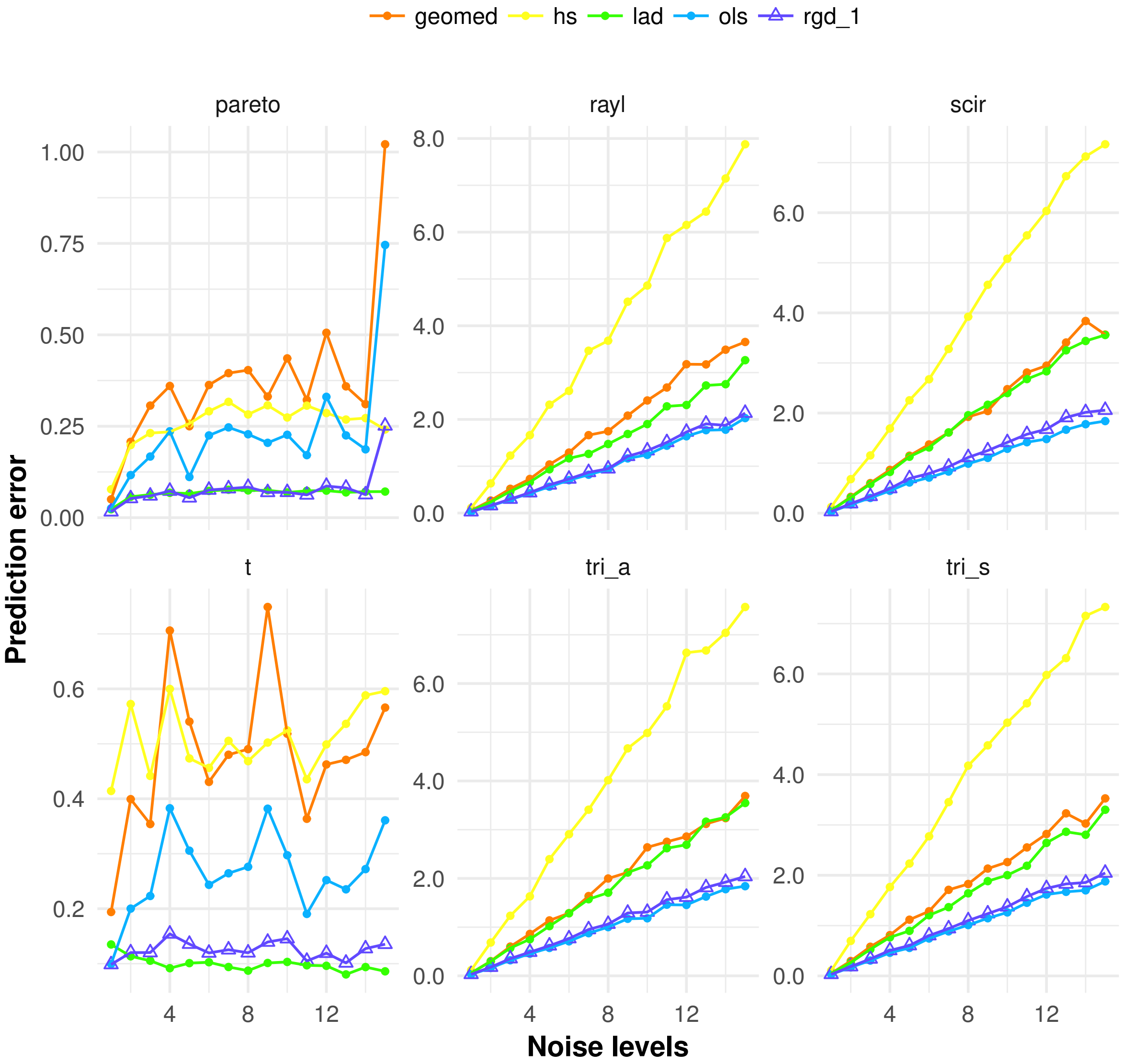

Figure A.29: Average error, over levels, for $n = 30, d = 5$.

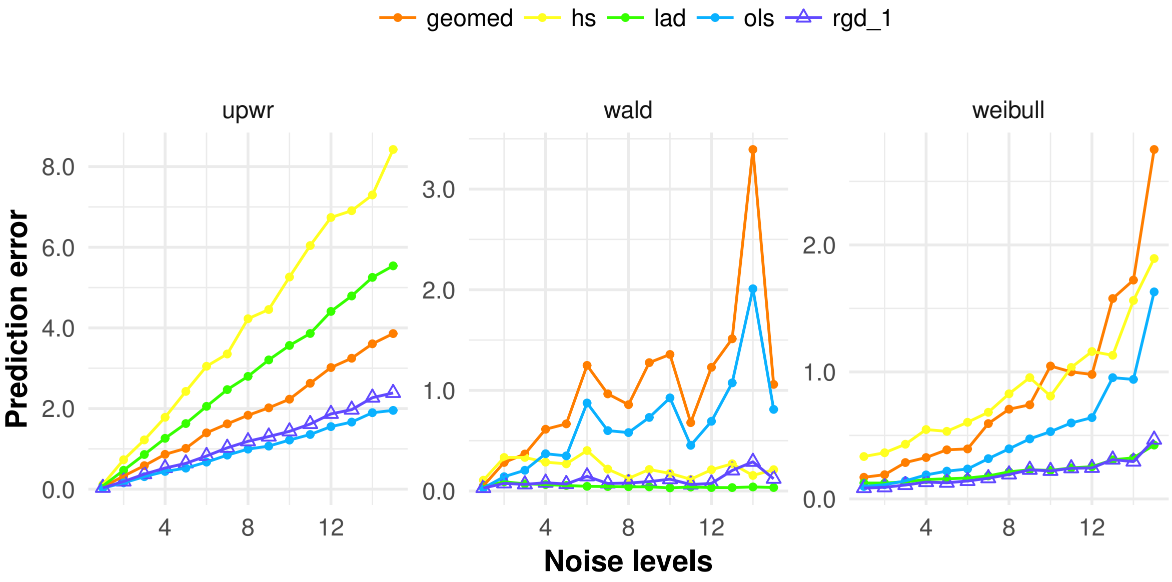

Figure A.30: Average error, over levels, for $n = 30, d = 5$.

A.4.2 Over sample size

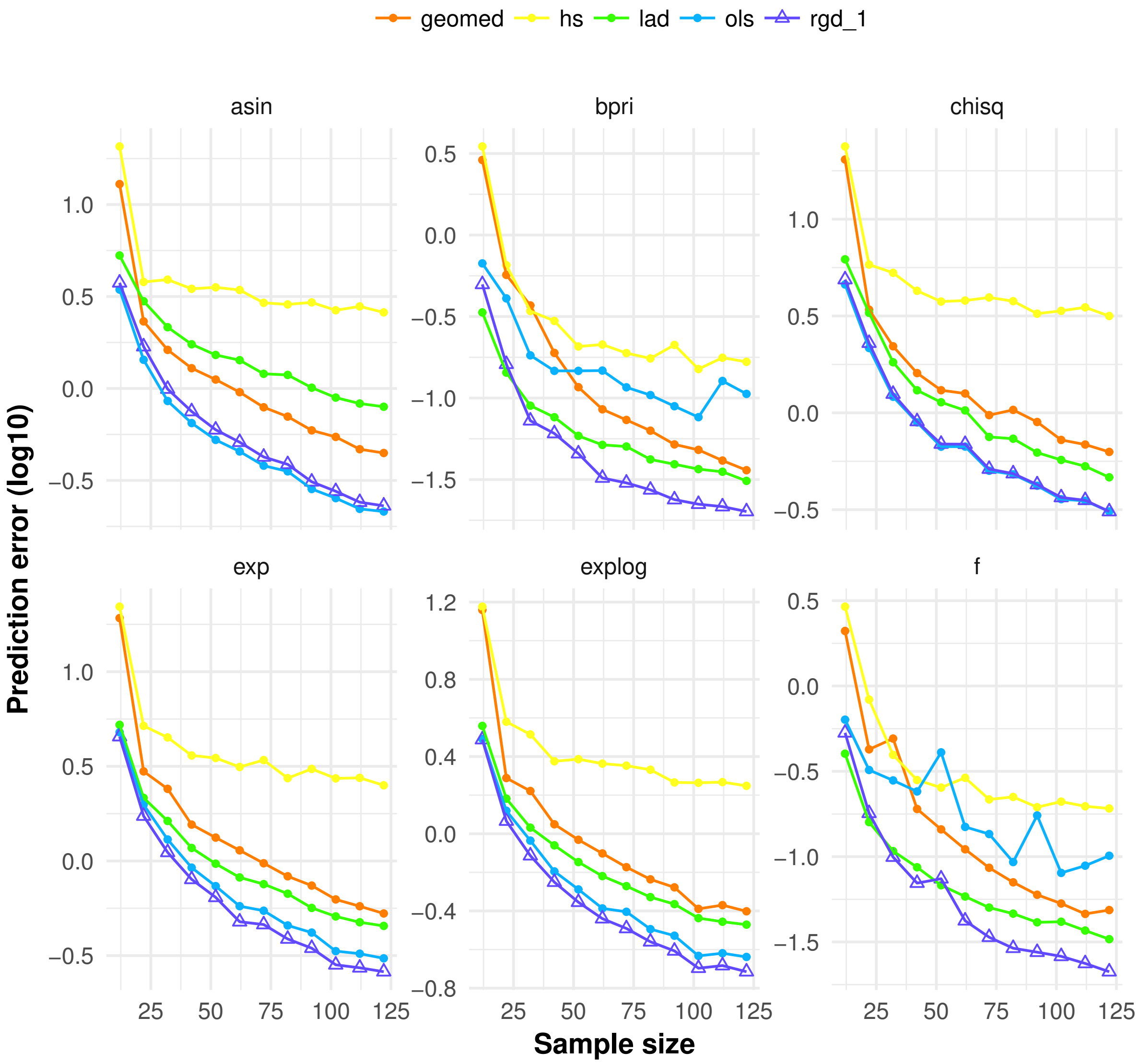

Figure A.31: Average error over n sizes, all methods. Level = 8.

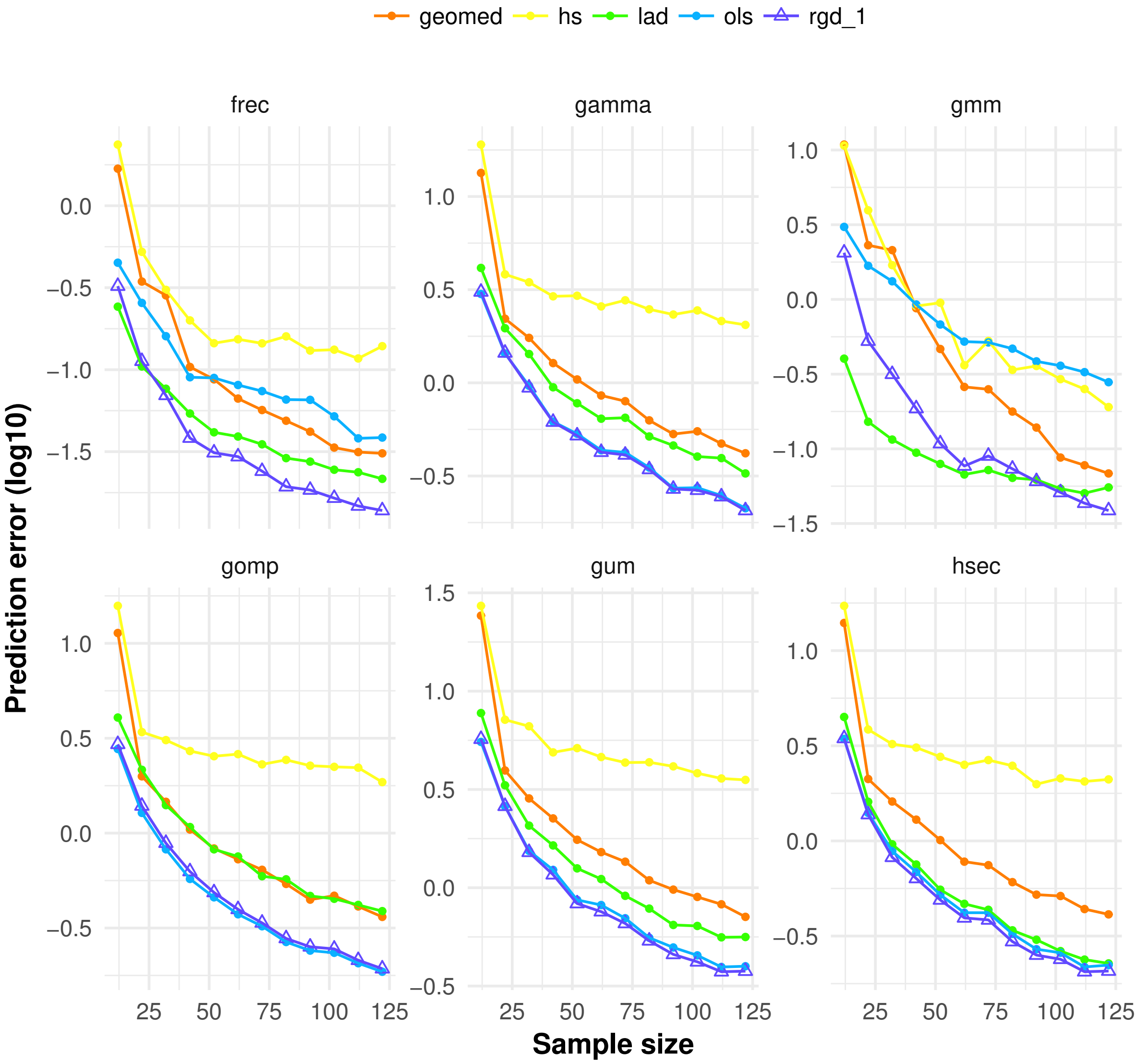

Figure A.32: Average error over n sizes, all methods. Level = 8.

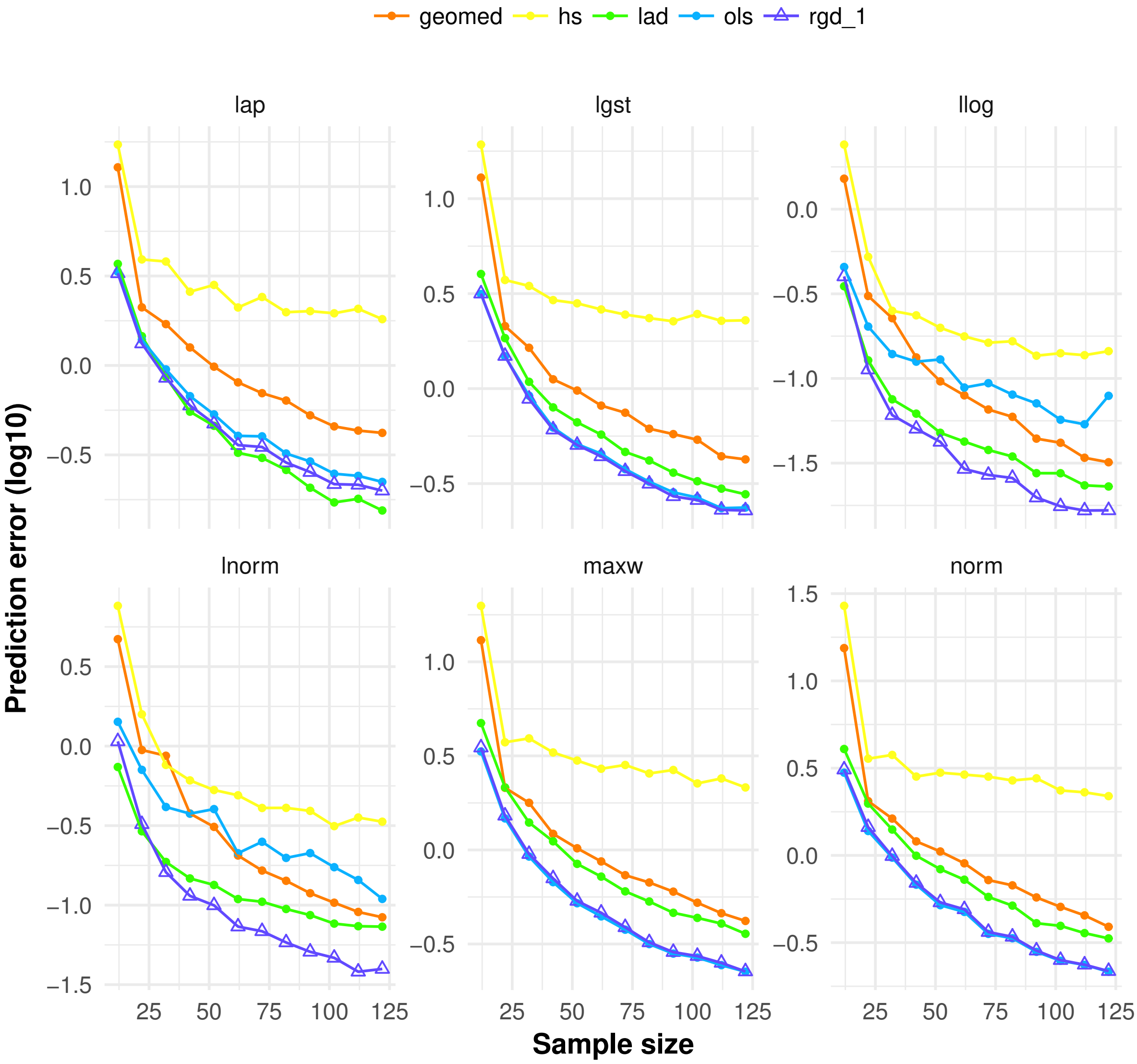

Figure A.33: Average error over n sizes, all methods. Level = 8.

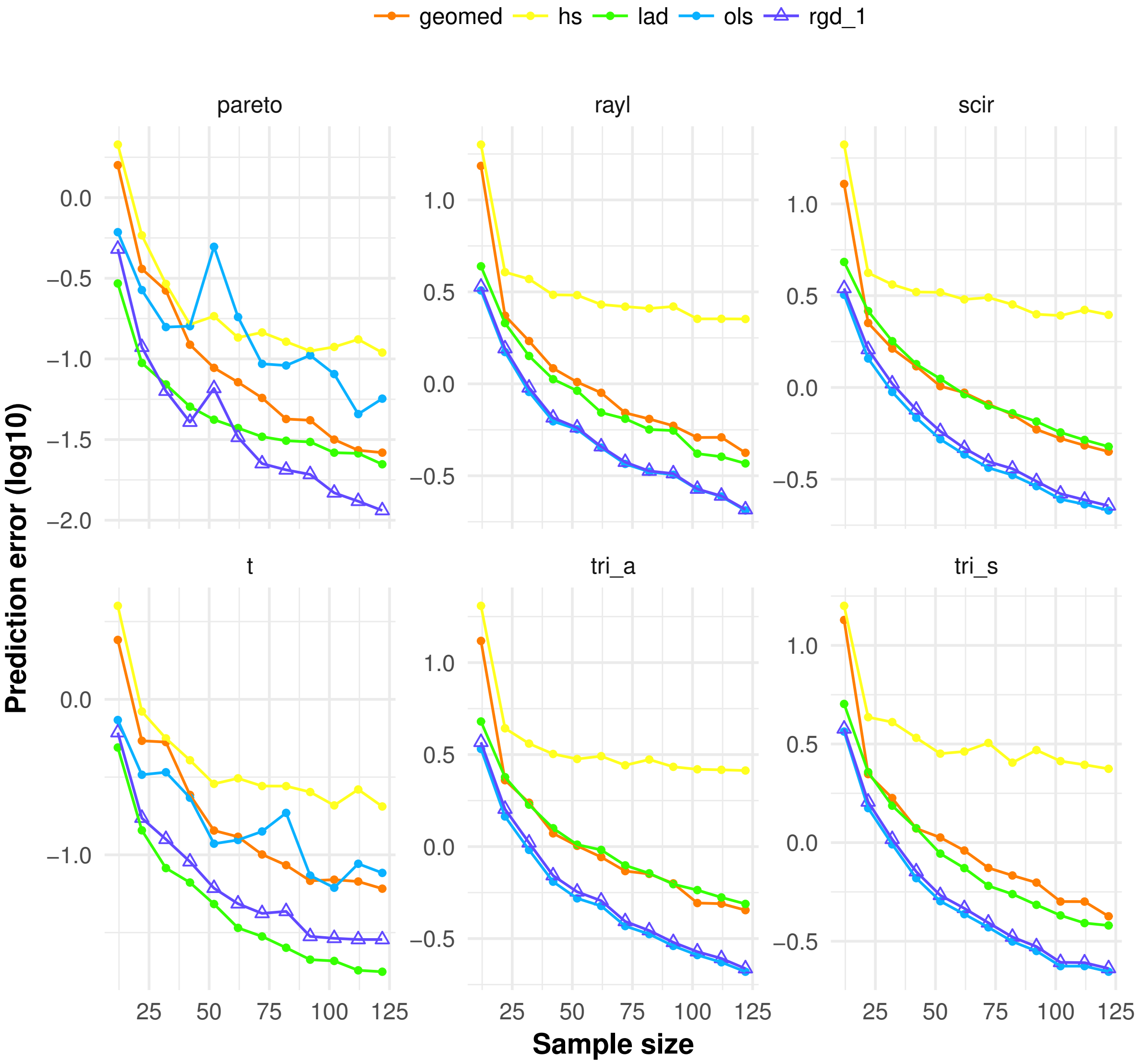

Figure A.34: Average error over n sizes, all methods. Level = 8.

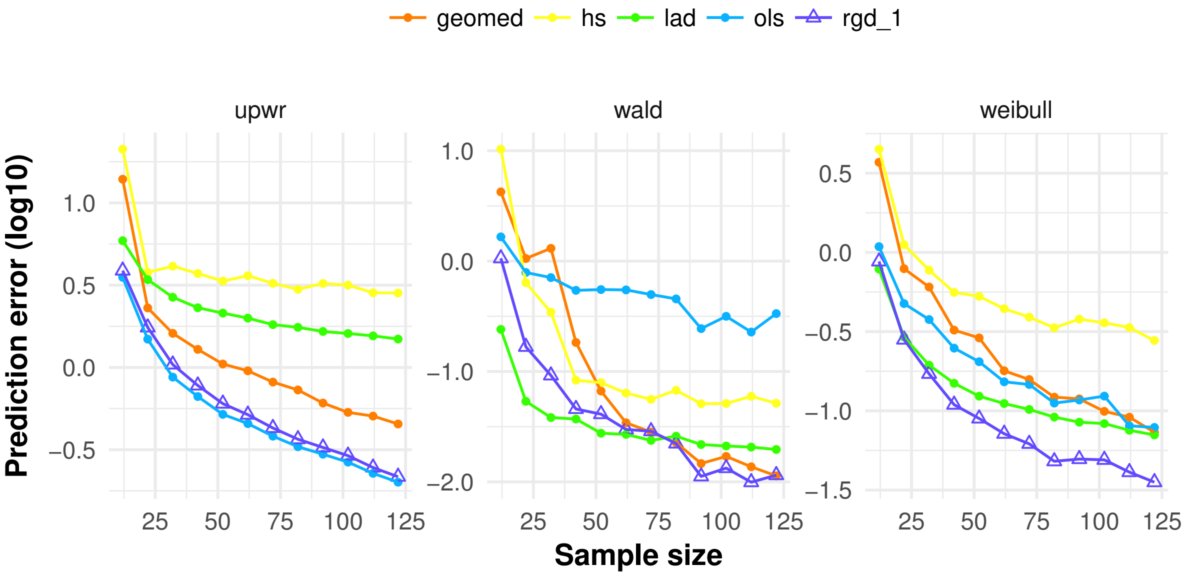

Figure A.35: Average error over n sizes, all methods. Level = 8.

A.4.3 Over dimension

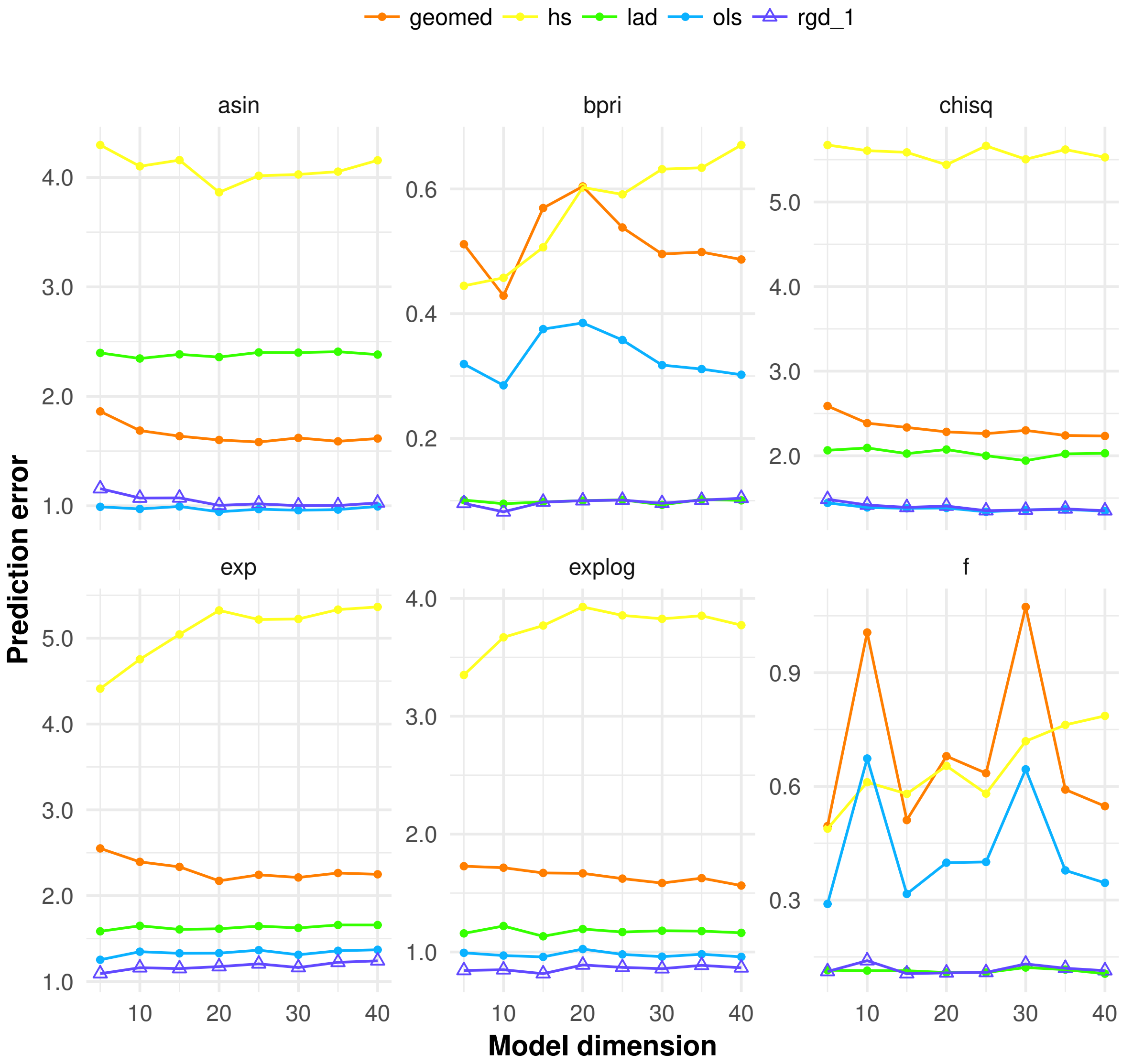

Figure A.36: Average error over dimension d, all methods. Level = 8.

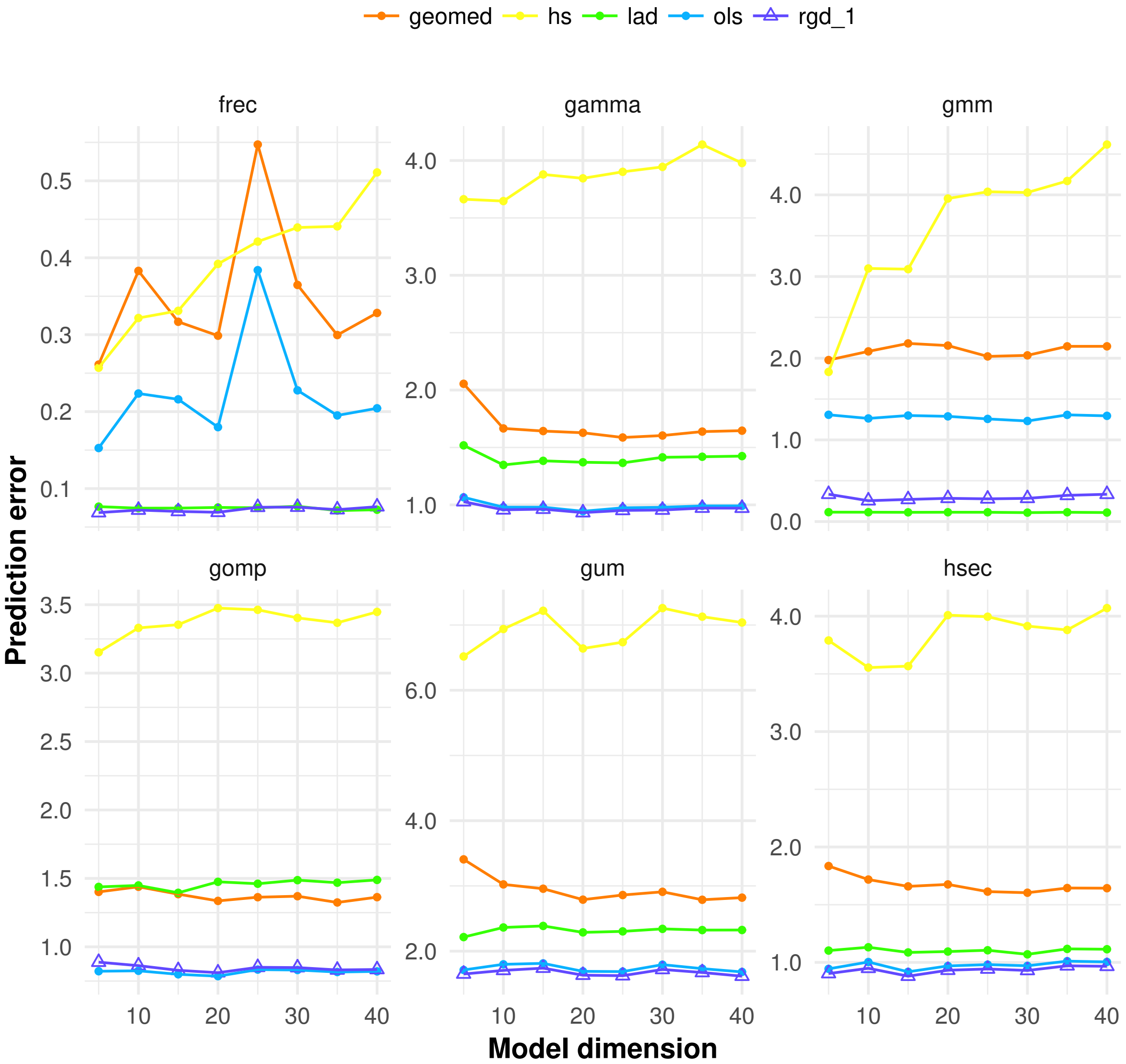

Figure A.37: Average error over dimension d, all methods. Level = 8.

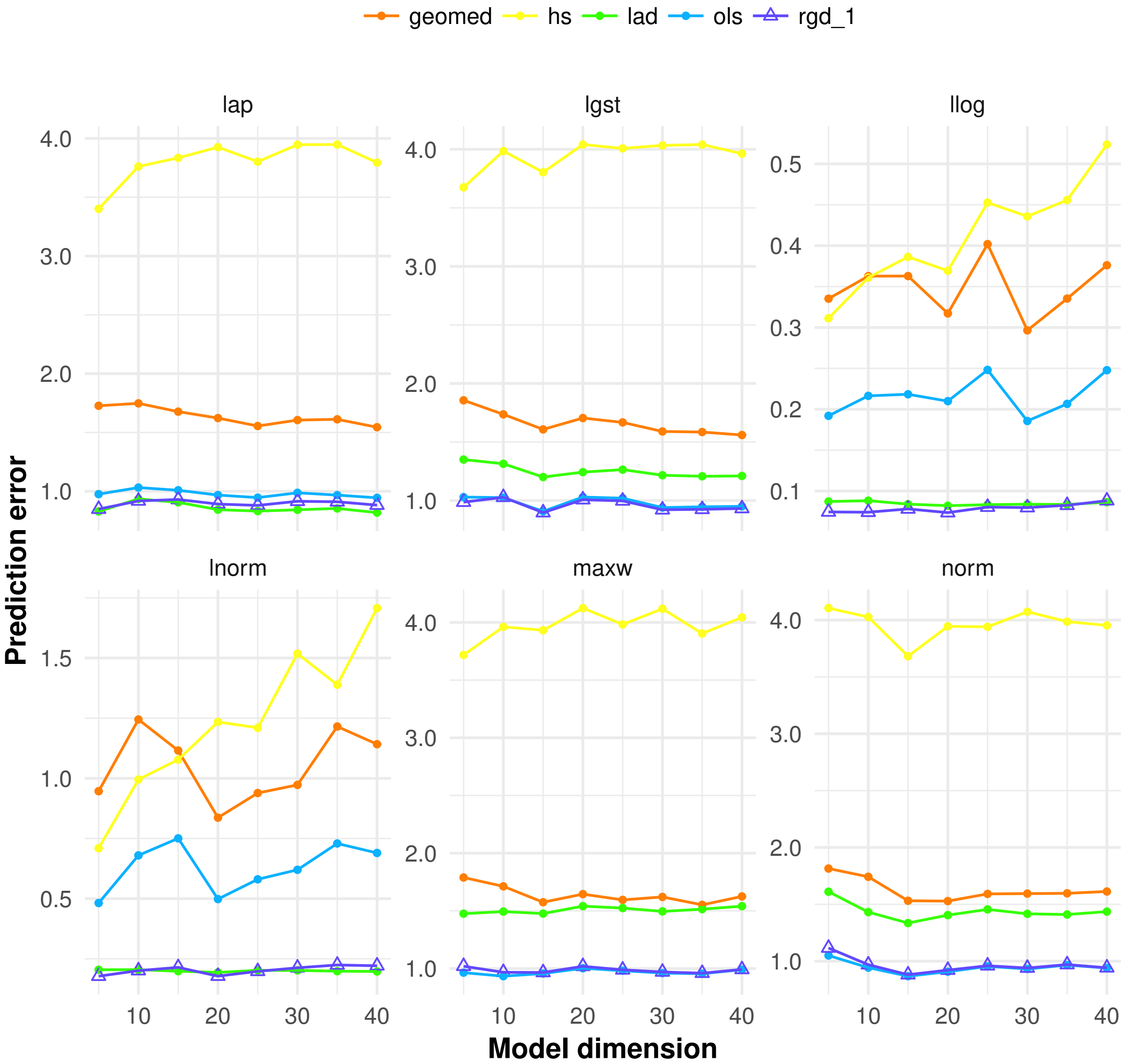

Figure A.38: Average error over dimension d, all methods. Level = 8.

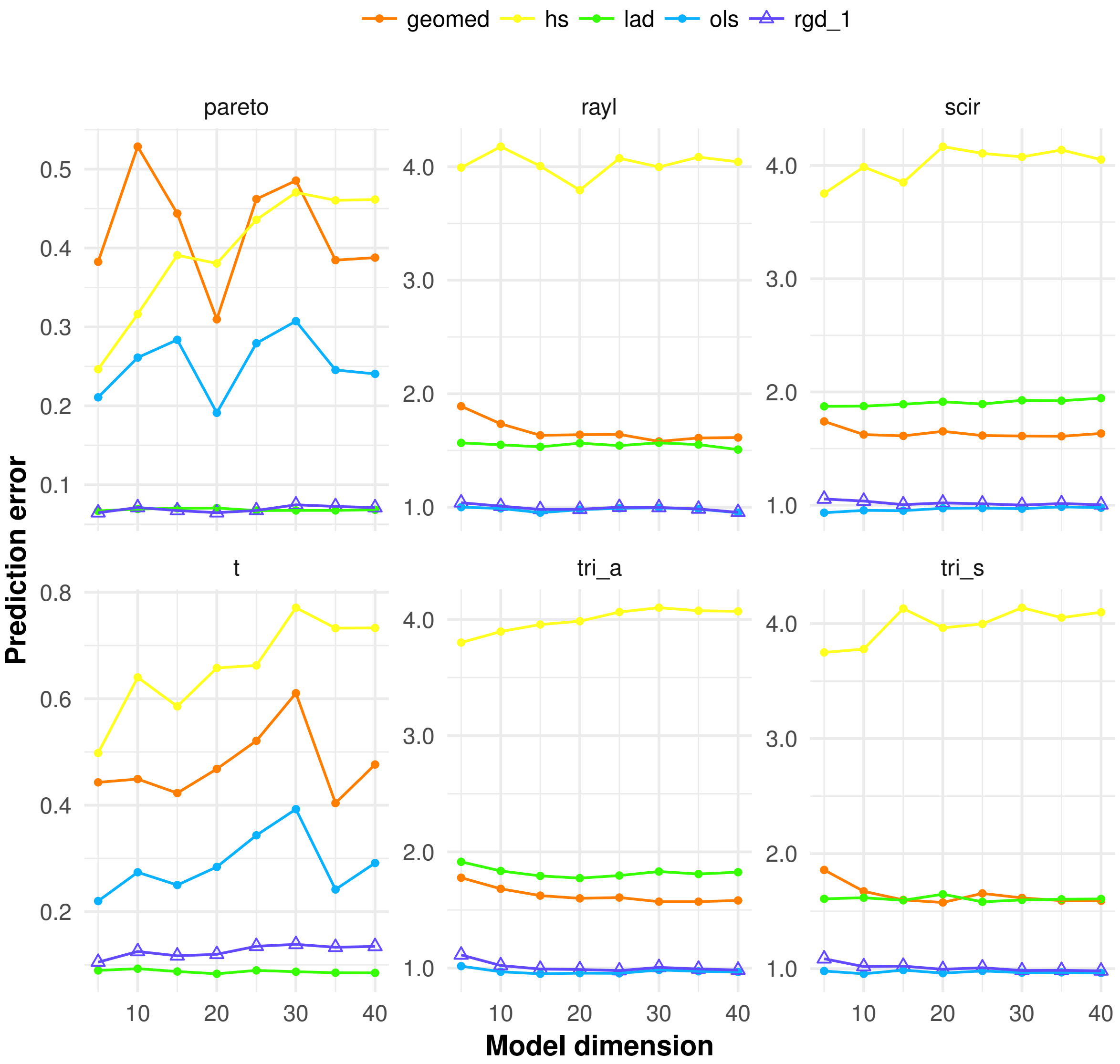

Figure A.39: Average error over dimension d, all methods. Level = 8.

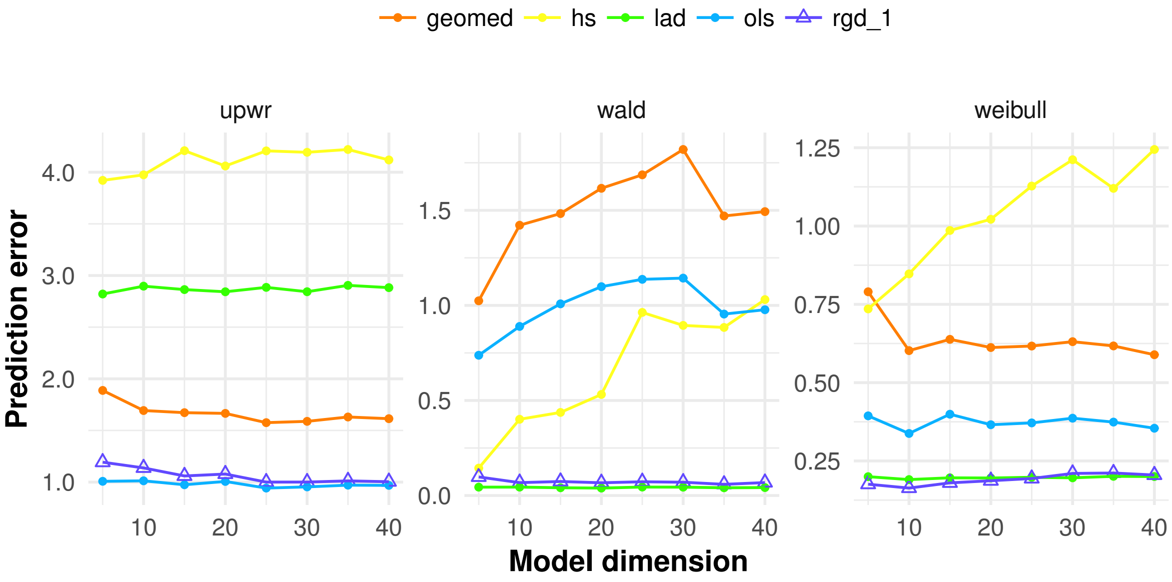

Figure A.40: Average error over dimension d, all methods. Level = 8.

A.5 Parameter settings for simulations

```
$asin
   shift      scale
1      0  0.8485281
2      0  4.8285292
3      0  8.8085302
4      0 12.7885312
5      0 16.7685322
6      0 20.7485333
7      0 24.7285343
8      0 28.7085353
9      0 32.6885363
10     0 36.6685374
11     0 40.6485384
12     0 44.6285394
13     0 48.6085404
14     0 52.5885415
15     0 56.5685425
```

```
$bpri
   shape1   shape2
1     1.5 6.164485
2     1.5 2.614809
3     1.5 2.266684
4     1.5 2.147522
5     1.5 2.092674
6     1.5 2.063206
7     1.5 2.045686
8     1.5 2.034483
9     1.5 2.026910
10    1.5 2.021564
11    1.5 2.017656
12    1.5 2.014715
13    1.5 2.012448
14    1.5 2.010665
15    1.5 2.009238
```

```
$chisq
         df
1    0.0450
2   14.3275
3   28.6100
4   42.8925
5   57.1750
6   71.4575
7   85.7400
8  100.0225
9  114.3050
10 128.5875
11 142.8700
12 157.1525
13 171.4350
14 185.7175
15 200.0000
```

```
$exp
         rate
1  3.33333333
2  0.18680986
3  0.13219836
4  0.10796782
5  0.09351514
6  0.08364907
7  0.07636481
8  0.07070272
9  0.06613820
10 0.06235706
11 0.05915814
12 0.05640588
13 0.05400516
14 0.05188700
15 0.05000000
```

```
$explog
   shape     scale
1   0.95  0.310000
2   0.95  1.717143
3   0.95  3.124286
4   0.95  4.531429
5   0.95  5.938571
6   0.95  7.345714
7   0.95  8.752857
8   0.95 10.160000
9   0.95 11.567143
10  0.95 12.974286
11  0.95 14.381429
12  0.95 15.788571
13  0.95 17.195714
14  0.95 18.602857
15  0.95 20.010000
```

```
$f
   df1        df2
1   15 111.935617
2   15   5.801913
3   15   4.698612
4   15   4.371894
5   15   4.229641
6   15   4.155263
7   15   4.111698
8   15   4.084079
9   15   4.065511
10  15   4.052448
11  15   4.042919
12  15   4.035759
13  15   4.030246
14  15   4.025913
15  15   4.022445
```

```
$frec
   shift scale  shape
1      0     1 5.5000
2      0     1 4.0000
3      0     1 3.0000
4      0     1 2.5000
5      0     1 2.3000
6      0     1 2.2500
7      0     1 2.2000
8      0     1 2.1500
9      0     1 2.1000
10     0     1 2.0500
11     0     1 2.0250
12     0     1 2.0125
13     0     1 2.0100
14     0     1 2.0075
15     0     1 2.0050
```

```
$gamma
   scale      shape
1      5  0.0036000
2      5  0.1165735
3      5  0.3879510
4      5  0.8177327
5      5  1.4059184
6      5  2.1525082
7      5  3.0575020
8      5  4.1209000
9      5  5.3427020
10     5  6.7229082
11     5  8.2615184
12     5  9.9585327
13     5 11.8139510
14     5 13.8277735
15     5 16.0000000
```

```
$gmm
   mean1 mean2 sd1 sd2     wt1     wt2 k
1    -15    15 0.3  43 0.99999 0.00001 2
2    -15    15 0.3  43 0.99750 0.00250 2
3    -15    15 0.3  43 0.99500 0.00500 2
4    -15    15 0.3  43 0.98250 0.01750 2
5    -15    15 0.3  43 0.97000 0.03000 2
6    -15    15 0.3  43 0.95750 0.04250 2
7    -15    15 0.3  43 0.94500 0.05500 2
8    -15    15 0.3  43 0.93250 0.06750 2
9    -15    15 0.3  43 0.92000 0.08000 2
10   -15    15 0.3  43 0.90750 0.09250 2
11   -15    15 0.3  43 0.89500 0.10500 2
12   -15    15 0.3  43 0.88250 0.11750 2
13   -15    15 0.3  43 0.87000 0.13000 2
14   -15    15 0.3  43 0.85750 0.14250 2
15   -15    15 0.3  43 0.84500 0.15500 2
```

```
$gomp
        shape scale
1  1.00000000    15
2  0.92857857    15
3  0.85715714    15
4  0.78573571    15
5  0.71431429    15
6  0.64289286    15
7  0.57147143    15
8  0.50005000    15
9  0.42862857    15
10 0.35720714    15
11 0.28578571    15
12 0.21436429    15
13 0.14294286    15
14 0.07152143    15
15 0.00010000    15
```

```
$gum
   shift     scale
1      0  0.414593
2      0  2.359231
3      0  4.303870
4      0  6.248508
5      0  8.193147
6      0 10.137785
7      0 12.082424
8      0 14.027062
9      0 15.971701
10     0 17.916339
11     0 19.860978
12     0 21.805616
13     0 23.750255
14     0 25.694893
15     0 27.639532
```

```
$hsec
   shift     scale
1      0  0.300000
2      0  1.707143
3      0  3.114286
4      0  4.521429
5      0  5.928571
6      0  7.335714
7      0  8.742857
8      0 10.150000
9      0 11.557143
10     0 12.964286
11     0 14.371429
12     0 15.778571
13     0 17.185714
14     0 18.592857
15     0 20.000000
```

```
$lap
   shift     scale
1      0  0.212132
2      0  1.207132
3      0  2.202133
4      0  3.197133
5      0  4.192133
6      0  5.187133
7      0  6.182134
8      0  7.177134
9      0  8.172134
10     0  9.167134
11     0 10.162135
12     0 11.157135
13     0 12.152135
14     0 13.147135
15     0 14.142136
```

```
$llog
    shape scale
1  5.5000     1
2  4.0000     1
3  3.0000     1
4  2.5000     1
5  2.3000     1
6  2.2500     1
7  2.2000     1
8  2.1500     1
9  2.1000     1
10 2.0500     1
11 2.0250     1
12 2.0125     1
13 2.0100     1
14 2.0075     1
15 2.0050     1
```

```
$lnorm
   meanlog sdlog
1        0 0.300
2        0 0.500
3        0 0.750
4        0 1.000
5        0 1.100
6        0 1.200
7        0 1.300
8        0 1.400
9        0 1.500
10       0 1.550
11       0 1.600
12       0 1.625
13       0 1.650
14       0 1.700
15       0 1.730
```

```
$lgst
   shift      scale
1      0  0.1653987
2      0  0.9411972
3      0  1.7169957
4      0  2.4927942
5      0  3.2685927
6      0  4.0443913
7      0  4.8201898
8      0  5.5959883
9      0  6.3717868
10     0  7.1475853
11     0  7.9233838
12     0  8.6991824
13     0  9.4749809
14     0 10.2507794
15     0 11.0265779
```

```
$maxw
        scale
1   0.4454742
2   2.5349606
3   4.6244469
4   6.7139332
5   8.8034195
6  10.8929059
7  12.9823922
8  15.0718785
9  17.1613648
10 19.2508511
11 21.3403375
12 23.4298238
13 25.5193101
14 27.6087964
15 29.6982827
```

```
$norm
   shift     scale
1      0  0.300000
2      0  1.707143
3      0  3.114286
4      0  4.521429
5      0  5.928571
6      0  7.335714
7      0  8.742857
8      0 10.150000
9      0 11.557143
10     0 12.964286
11     0 14.371429
12     0 15.778571
13     0 17.185714
14     0 18.592857
15     0 20.000000
```

```
$pareto
          a b
1  5.239266 1
2  2.414201 1
3  2.164561 1
4  2.086463 1
5  2.052701 1
6  2.035287 1
7  2.025208 1
8  2.018877 1
9  2.014651 1
10 2.011694 1
11 2.009547 1
12 2.007939 1
13 2.006704 1
14 2.005736 1
15 2.004963 1
```

```
$rayl
        scale
1   0.4579199
2   2.6057824
3   4.7536449
4   6.9015074
5   9.0493699
6  11.1972324
7  13.3450949
8  15.4929574
9  17.6408199
10 19.7886824
11 21.9365449
12 24.0844074
13 26.2322699
14 28.3801324
15 30.5279949
```

```
$scir
   center       rad
1       0  0.600000
2       0  3.414286
3       0  6.228571
4       0  9.042857
5       0 11.857143
6       0 14.671429
7       0 17.485714
8       0 20.300000
9       0 23.114286
10      0 25.928571
11      0 28.742857
12      0 31.557143
13      0 34.371429
14      0 37.185714
15      0 40.000000
```

```
$t
          df
1  11.523810
2   2.399800
3   2.148810
4   2.078362
5   2.048485
6   2.032991
7   2.023912
8   2.018132
9   2.014224
10  2.011457
11  2.009427
12  2.007893
13  2.006705
14  2.005767
15  2.005013
```

```
$tri_a                     $tri_s                     $upwr
   vert shift      scale      vert shift      scale      shape shift       scale
1   0.9     0  1.334249  1   0.5     0  1.469694  1       2     0  0.3549648
2   0.9     0  7.592511  2   0.5     0  8.363258  2       2     0  2.0199187
3   0.9     0 13.850773  3   0.5     0 15.256822  3       2     0  3.6848726
4   0.9     0 20.109035  4   0.5     0 22.150386  4       2     0  5.3498264
5   0.9     0 26.367297  5   0.5     0 29.043950  5       2     0  7.0147803
6   0.9     0 32.625559  6   0.5     0 35.937514  6       2     0  8.6797342
7   0.9     0 38.883821  7   0.5     0 42.831078  7       2     0 10.3446881
8   0.9     0 45.142083  8   0.5     0 49.724642  8       2     0 12.0096420
9   0.9     0 51.400345  9   0.5     0 56.618206  9       2     0 13.6745958
10  0.9     0 57.658608  10  0.5     0 63.511770  10      2     0 15.3395497
11  0.9     0 63.916870  11  0.5     0 70.405334  11      2     0 17.0045036
12  0.9     0 70.175132  12  0.5     0 77.298898  12      2     0 18.6694575
13  0.9     0 76.433394  13  0.5     0 84.192462  13      2     0 20.3344114
14  0.9     0 82.691656  14  0.5     0 91.086026  14      2     0 21.9993653
15  0.9     0 88.949918  15  0.5     0 97.979590  15      2     0 23.6643191

$wald                                  $weibull
   mean        shape                      scale shape
1     1 11.111111111                   1       1 0.990
2     1  0.343131248                   2       1 0.900
3     1  0.103105799                   3       1 0.800
4     1  0.048915743                   4       1 0.700
5     1  0.028451154                   5       1 0.650
6     1  0.018582972                   6       1 0.600
7     1  0.013082575                   7       1 0.550
8     1  0.009706617                   8       1 0.525
9     1  0.007486848                   9       1 0.500
10    1  0.005949806                   10      1 0.475
11    1  0.004841725                   11      1 0.450
12    1  0.004016656                   12      1 0.425
13    1  0.003385827                   13      1 0.400
14    1  0.002892729                   14      1 0.375
15    1  0.002500000                   15      1 0.350
```

A.6 Image credits

Several digital images were used for illustrative purposes. Detailed references are provided here. In Figure 1.1, digital images of paintings are displayed. All images were downloaded from the online repository of the United States National Gallery of Art (NGA), and are designated as *Open Access* images, determined by the Gallery to be in the public domain. The original images were resized (reduction) with original height/width ratios preserved, and cropped to a common size. The four paintings (left to right) are

```
Artist: Paul Cezanne
Title: Still Life with Apples and Peaches (c. 1905)

Artist: Paul Cezanne
Title: Still Life with Milk Jug and Fruit (c. 1900)
```

```
Artist: Auguste Renoir
Title: Peaches on a Plate (1902/1905)

Artist: Paul Cezanne
Title: The Peppermint Bottle (1893/1895)
```

In Figure 1.2, we have two digital images of classical Japanese texts, respectively of *Tosa Nikki*[3] (left) and *Man'yōshū*[4] (right). The original images are provided by the Center for Open Data in the Humanities, and the original texts are from the National Institute of Japanese Literature (NIJL), shared under a Attribution-ShareAlike 4.0 International license. Low-resolution screenshots of the original images were cropped to a common size.

[3] From image 3 of 34 (NIJL ID: 200010982).

[4] From image 13 of 1070 (NIJL ID: 200015542).

Bibliography

[1] Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series.* US National Bureau of Standards.

[2] Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972). *Robust estimates of location: survey and advances.* Princeton University Press.

[3] Black, M. J. and Rangarajan, A. (1996). On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91.

[4] Catoni, O. (2009). High confidence estimates of the mean of heavy-tailed real random variables. *arXiv preprint arXiv:0909.5366.*

[5] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148–1185.

[6] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms.* MIT Press, 3rd edition.

[7] Cramer, J. S. (2003). The origins and development of the logit model. In *Logit Models from Economics and Other Fields*, pages 149–157. Cambridge University Press.

[8] Croux, C. and Rousseeuw, P. J. (1992). Time-efficient algorithms for two highly robust estimators of scale. In *Computational Statistics, Volume 1: Proceedings of the 10th Symposium on Computational Statistics*, pages 411–428. Springer.

[9] Dennis, J. E. and Welsch, R. E. (1978). Techniques for nonlinear least squares and robust regression. *Communications in Statistics - Simulation and Computation*, 7(4):345–359.

[10] Geman, D. and Reynolds, G. (1992). Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383.

[11] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741.

[12] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions.* John Wiley & Sons.

[13] Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

[14] Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634.

[15] Li, T.-Y. and Yorke, J. A. (1975). Period three implies chaos. *American Mathematical Monthly*, 82(10):985–992.

[16] Manning, K. R. (1975). The emergence of the Weierstrassian approach to complex analysis. *Archive for History of Exact Sciences*, 14(4):297–383.

[17] May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261:459–467.

[18] Rey, W. J. J. (1983). *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer.

[19] Rousseeuw, P. and Yohai, V. (1984). Robust regression by means of S-estimators. In *Robust and Nonlinear Time Series Analysis*, volume 26 of *Lecture Notes in Statistics*, pages 256–272. Springer.

[20] Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

[21] Rousseeuw, P. J. and Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283.

[22] Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer.

[23] Yao, W., Lindsay, B. G., and Li, R. (2012). Local modal regression. *Journal of Nonparametric Statistics*, 24(3):647–663.

[24] Yu, Y., Aslan, Ö., and Schuurmans, D. (2012). A polynomial-time form of robust regression. In *Advances in Neural Information Processing Systems 25*, pages 2483–2491.