Machine Learning (2019) 108:1523–1560
https://doi.org/10.1007/s10994-019-05802-5

Efficient learning with robust gradient descent

Matthew J. Holland[1] · **Kazushi Ikeda**[2]

Received: 13 September 2018 / Accepted: 20 April 2019 / Published online: 25 June 2019

Abstract
Minimizing the empirical risk is a popular training strategy, but for learning tasks where the data may be noisy or heavy-tailed, one may require many observations in order to generalize well. To achieve better performance under less stringent requirements, we introduce a procedure which constructs a robust approximation of the risk gradient for use in an iterative learning routine. Using high-probability bounds on the excess risk of this algorithm, we show that our update does not deviate far from the ideal gradient-based update. Empirical tests using both controlled simulations and real-world benchmark data show that in diverse settings, the proposed procedure can learn more efficiently, using less resources (iterations and observations) while generalizing better.

Keywords Robust learning · Stochastic optimization · Statistical learning theory

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 $R(w) := \mathbf{E}\, l(w; z)$, induced by a loss l, where w is the parameter (vector, function, set, etc.) to be specified, and expectation is with respect to z, namely the underlying data distribution. Given data $z_1, \ldots, z_n$, if an algorithm outputs $\widehat{w}$ such that $R(\widehat{w})$ 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 $\widehat{w}$ we ever have in practice is one we can actually compute.

Empirical risk minimization (ERM), which admits any minimizer of $n^{-1} \sum_{i=1}^{n} l(\cdot; z_i)$, is the canonical strategy for machine learning problems, and there exists a rich body of literature on its generalization ability (Kearns and Schapire 1994; Bartlett et al. 1996; Alon et al. 1997; Bartlett and Mendelson 2003). In recent years, however, some severe limitations

Editors: Karsten Borgwardt, Po-Ling Loh, Evimaria Terzi, Antti Ukkonen.

✉ Matthew J. Holland
matthew-h@ar.sanken.osaka-u.ac.jp

[1] Osaka University, Mihogaoka 8-1, Ibaraki, Osaka, Japan

[2] Nara Institute of Science and Technology, Takayama-cho, Ikoma, Nara 8916-5, Japan

of this technique have come into light. ERM can be implemented by numerous methods, but its performance is sensitive to this implementation (Daniely and Shalev-Shwartz 2014; Feldman 2016), 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 (2016), 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 (Finkenstädt and Rootzén 2003), 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 paper, 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 (2016) 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 (Lerasle and Oliveira 2011; Minsker and Strawn 2017) to multi-dimensional and even functional models (Minsker 2015; Hsu and Sabato 2016; Lecué and Lerasle 2017). 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 $\widehat{R}(w)$ of the risk $R(w)$ for each 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. (2015), based on fundamental results due to Catoni (2009, 2012). 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 can be both unstable and costly in high dimensions.

Finally, conceptually the closest recent work to our research are those also analyzing novel "robust gradient descent" algorithms, namely steepest descent procedures which utilize a robust estimate of the gradient vector of the underlying (unknown) objective of interest. The first works in this line are due to Holland and Ikeda (2017a) (a preliminary version of our work) and Chen et al. (2017a) (later updated as Chen et al. (2017b)), which appeared as pre-prints almost simultaneously. This was later followed by more recent entries into the literature due to Prasad et al. (2018) and Lecué et al. (2018). We give a more formal description of these methods in Remark 1 at the end of Sect. 2, but describe key similarities and differences here. All of these cited works are based upon the "median of means" strategy for aggregating weak, independent estimators to produce a final estimate of the risk gradient to be plugged into a steepest descent routine. These routines differ from each other in terms of how this aggregation is executed. Both Chen et al. (2017a) and Prasad et al. (2018) compute *gradient* sample means on independent sub-samples (after partitioning), and then compute the high-dimensional geometric median of these means to aggregate them into a final estimate. On the other hand, Lecué et al. (2018) compute *loss* sample means on the sub-samples, identify the sub-sample which realizes the median loss, and then plugs in the gradient sample mean based on just this one sub-sample. These routines are fundamentally different from our approach in that we do no partitioning at all, and make no use of any median-like quantity in our sub-routines; using all data is used to construct an M-estimator which can be approximated easily using fixed-point iterative updates, which amounts to a parameter falling between the median and the mean. One key advantage to our approach is the ease of computation (see Remark 2 and "Appendix A.4"). While the geometric median used by Chen et al. (2017b) can indeed be computed using well-known iterative routines (Vardi and Zhang 2000), these suffer from substantial overhead in computing pairwise distances over all partitions at each iteration, and as mentioned above in reference to the work of Minsker (2015) and Hsu and Sabato (2016), can run into significant bias when the number of partitions cannot be made large enough.

To compare our problem setting with the setting of these works: we are considering the potentially heavy-tailed situation, in which we do not assume the higher-order moments of the observations are finite (thereby implying that sub-Gaussianity cannot be used for concentration inequalities). This heavy-tailed setting is shared by Prasad et al. (2018) and Lecué et al. (2018), although we mention that Prasad et al. (2018) also propose an entirely separate gradient-based algorithm for the case of data which may be subject to arbitrarily large contamination, a setting shared with Chen et al. (2017b). This setting is fundamentally different from the heavy-tailed scenario, and methods designed for estimation under adversarial outliers cannot be directly compared to those designed for the heavy-tailed setting, and thus in our subsequent theoretical and empirical comparisons with these works, we constrain our focus to methods proposed for the heavy-tailed setting, which allows for meaningful comparison with our proposal.

We also remark that while our proposed algorithm is designed to be easy to integrate into existing gradient-based learning algorithms, our chief focus is on improving the statistical error incurred when trying to estimate the risk gradient for determining a good update *direction*, and we do not attend to the problem of update distance, nor long-run convergence via decreasing step sizes. Especially in the situation of small mini-batches, it is perfectly plausible to apply algorithms with principled adaptive step sizes such as AdaGrad (Duchi et al. 2011) or Adam (Kingma and Ba 2014), where instead of the empirical mean of the gradient, one substitutes our robust gradient estimate (see Sects. 2, 3 for details).

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 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 general methodology.

- Using both tightly controlled simulations and real-world benchmarks, we compare our routine with ERM-GD and other cited methods, obtaining results that reinforce the practical utility and flexibility suggested by the theory.

Content overview In Sect. 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 Sect. 3, including a sketch of the proof technique and discussion of the main results. Empirical analysis follows in Sect. 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 Sect. 5. Proofs and extra information regarding computation is given in "Appendix A". Additional empirical test results are provided in "Appendix B".

2 Robust gradient descent

Before introducing the proposed algorithm in more detail, we motivate the practical need for a procedure which deals with the weaknesses of the traditional sample mean-based gradient descent strategy.

2.1 Why robustness?

Recall that since ERM admits any minima of $n^{-1}\sum_{i=1}^{n} l(\cdot; z_i)$, the simplest implementation of gradient descent (for $\widehat{w}_{(t)} \in \mathbb{R}^d$) results in the update

$$\widehat{w}_{(t+1)} = \widehat{w}_{(t)} - \alpha_{(t)} \frac{1}{n} \sum_{i=1}^{n} l'(\widehat{w}_{(t)}; z_i) \tag{1}$$

where $\alpha_{(t)}$ are scaling parameters. Taking the derivative under the integral we have $R'(\cdot) = \mathbf{E}\, l'(\cdot; z)$, 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 $l(w; z)$ for each w, it has become well-known that the sample mean fails to provide sharp estimates (Catoni 2012; Minsker 2015; Devroye et al. 2015; Lugosi and Mendelson 2016). 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 Fig. 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 $\widehat{w}_{(0)}$. With data $z = (x, y) \in \mathbb{R}^{d+1}$, losses are generated as $l(w; z_i) = (\langle w, x_i \rangle - y_i)^2/2$. We consider the case where the "noise" $\langle w, x_i \rangle - y_i$ 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 (1), we replace the summands $n^{-1}l'(\cdot; z_i)$ with $\omega_i\, l'(\cdot; z_i)$, where $0 \leq \omega_i \leq 1$, $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \omega_i = 1$. For example, we could set

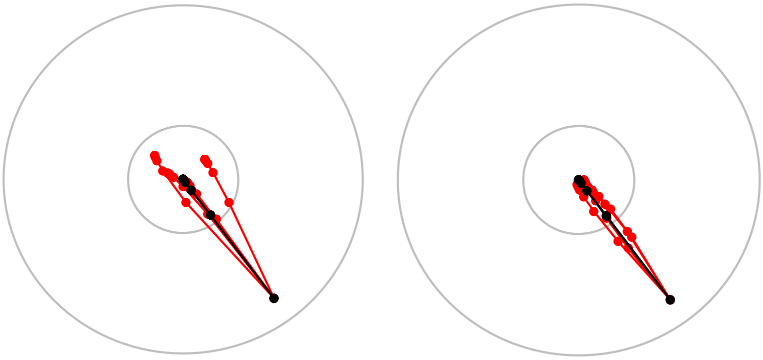

Fig. 1 A comparison of the minimizing sequence trajectories in a two-dimensional approximate risk minimization task, for the traditional ERM-based gradient descent (left) and a simple re-weighting procedure (right). Trajectories of the oracle update using R' (black) is pictured alongside the approximate methods (red). All procedures use $\alpha_{(t)} = 0.35$, $t = 0, \ldots, 9$ (Color figure online)

$$\omega_i := \frac{\widetilde{\omega}_i}{\sum_{k=1}^{n} \widetilde{\omega}_k}, \quad \widetilde{\omega}_i := \frac{\psi\left(\langle w, x \rangle - y_i\right)}{\left(\langle w, x \rangle - y_i\right)}$$

where ψ is an odd function of sigmoid form (see A.1 and A.4). The idea is that for observations 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 Fig. 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 idiosyncrasies 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.

2.2 Outline of proposed procedure

Were the risk to be known, we could update using

$$w^*_{(t+1)} := w^*_{(t)} - \alpha_{(t)} g(w^*_{(t)}) \tag{2}$$

where $g(w) := R'(w)$, an idealized procedure. Any learning algorithm in practice will not have access to R or g, and thus must approximate this update with

$$\widehat{w}_{(t+1)} := \widehat{w}_{(t)} - \alpha_{(t)} \widehat{g}(\widehat{w}_{(t)}), \tag{3}$$

where $\widehat{g}$ represents some sample-based estimate of g. Setting $\widehat{g}$ to the sample mean reduces to ERM-GD, and conditioned on $\widehat{w}_{(t)}$, $\mathbf{E}\,\widehat{g}(\widehat{w}_{(t+1)}) = g(\widehat{w}_{(t+1)})$, a property used throughout the literature (Rakhlin et al. 2012; Le Roux et al. 2012; Johnson and Zhang 2013; Shalev-Shwartz and Zhang 2013; Frostig et al. 2015; Murata and Suzuki 2016). While convenient from a technical standpoint, there is no conceptual necessity for $\widehat{g}$ to be unbiased. More realistically, as long as $\widehat{g}$ is sharply distributed around 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 1.

Algorithm 1 Robust gradient descent outline

```
inputs: $\widehat{w}_0, T > 0$
for $t = 0, 1, \ldots, T-1$ do
    $D_{(t)} \leftarrow \{l'(\widehat{w}_{(t)}; z_i)\}_{i=1}^n$          ▷ Update loss gradients.
    $\widehat{\sigma}_{(t)} \leftarrow$ RESCALE$(D_{(t)})$          ▷ Eq. (5).
    $\widehat{\theta}_{(t)} \leftarrow$ LOCATE$(D_{(t)}, \widehat{\sigma}_{(t)})$          ▷ Eqs. (4), (6).
    $\widehat{w}_{(t+1)} \leftarrow \widehat{w}_{(t)} - \alpha_{(t)}\widehat{\theta}_{(t)}$          ▷ Plug in to update.
end for
return: $\widehat{w}_{(T)}$
```

Let us flesh out the key sub-routines used in a single iteration, for the $w \in \mathbb{R}^d$ case. When the data is prone to outliers, a "soft" truncation of errant values is a prudent alternative to discarding valuable data. This can be done systematically using a convenient class of M-estimators of location and scale (van der Vaart 1998; Huber and Ronchetti 2009). The LOCATE sub-routine entails taking a convex, even function ρ, and for each coordinate, computing $\widehat{\theta} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_d)$ as

$$\widehat{\theta}_j \in \arg\min_{\theta \in \mathbb{R}} \sum_{i=1}^n \rho\left(\frac{l'_j(w; z_i) - \theta}{s_j}\right), \quad j = 1, \ldots, d. \tag{4}$$

Note that if $\rho(u) = u^2$, then $\widehat{\theta}_j$ reduces to the sample mean of $\{l'_j(w; z_i)\}_{i=1}^n$, 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 for each j using

$$\widehat{\sigma}_j \in \left\{\sigma > 0 : \sum_{i=1}^n \chi\left(\frac{l'_j(w; z_i) - \gamma_j}{\sigma}\right) = 0\right\}. \tag{5}$$

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 $l'_j(w; z)$ about a pivot point, say $\gamma_j = \sum_{i=1}^n l'_j(w; z_i)/n$. Second, we adjust this estimate based on the available sample size and desired confidence level, as

$$s_j = \widehat{\sigma}_j \sqrt{n/\log(2\delta^{-1})} \tag{6}$$

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 Sect. 3.1. This concludes all the steps[1] in one full iteration of Algorithm 1 on $\mathbb{R}^d$.

In the remainder of this paper, we shall investigate the learning properties of this procedure, through analysis of both a theoretical (Sect. 3) and empirical (Sect. 4) nature. As an example, in the strongly convex risk case, our formal argument yields excess risk bounds of the form

[1] For concreteness, in all empirical tests to follow we use the Gudermannian function (Abramowitz and Stegun 1964), $\rho(u) = \int_0^u \psi(x)\,dx$ 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 "Appendix A.1".

$$R(\widehat{w}_{(T)}) - R^* \leq O\left(\frac{d(\log(d\delta^{-1}) + d\log(n))}{n}\right) + O\left((1-\alpha\beta)^T\right)$$

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 (see "Appendix A.3"). Of the underlying distribution, all that is assumed is a bound on the variance of $l'(\cdot; z)$, suggesting formally that the procedure should be competitive over a diverse range of data distributions.

Remark 1 (Comparison with other robust gradient descent methods) Here we more formally describe the algorithms of other robust gradient descent methods in the literature (Chen et al. 2017b; Prasad et al. 2018; Lecué et al. 2018). The core idea is to partition the sample into k disjoint subsets $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_k = \{1, 2, \ldots, n\}$, generate inexpensive empirical estimators on each subset, and then robustly aggregate these estimators to output a final gradient estimate. In the case of both Chen et al. (2017b) and Prasad et al. (2018), the estimate vector $\widehat{g}$ is constructed as

$$\widehat{g}(w) = \arg\min_u \sum_{m=1}^{k} \|u - \widetilde{g}_m(w)\|$$

$$\widetilde{g}_m(w) = \frac{1}{|\mathcal{D}_m|} \sum_{i \in \mathcal{D}_m} l'(w; z_i), \quad m = 1, \ldots, k$$

and plugged in to the gradient update (3). On the other hand, the update of Lecué et al. (2018) uses

$$\widehat{g}(w) = \frac{1}{|\mathcal{D}_\star|} \sum_{i \in \mathcal{D}_\star} l'(w; z_i)$$

$$\star = \arg\min_{m \in [k]} \left|\widetilde{l}_m(w) - \bar{l}(w)\right|$$

$$\bar{l}(w) = \operatorname{med}\left\{\widetilde{l}_m(w) : m \in [k]\right\}$$

$$\widetilde{l}_m(w) = \frac{1}{|\mathcal{D}_m|} \sum_{i \in \mathcal{D}_m} l(w; z_i), \quad m = 1, \ldots, k.$$

Compare these updates to Eqs. (4) and (6), which determine our setting of $\widehat{g}(w)$ for use in the steepest descent update (3). Both median (or geometric median, as above) updates and M-estimator updates (as in our proposal) involve doing convex optimization as a sub-routine, but the nature of the respective objective functions is in general completely different.

Remark 2 (On the efficiency of RESCALE and LOCATE) Both of these sub-routines can be efficiently solved using fixed-point updates, as described in "Appendix A.4". Both routines have convergence guarantees, and can be confirmed empirically to converge rapidly irrespective of the underlying distribution. Formal proofs of convergence of the RESCALE and LOCATE sub-routines along with controlled empirical tests is given by Holland and Ikeda (2017b).

3 Theoretical analysis

Here we analyze the performance of Algorithm 1 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 Appendix A.2.

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 $z_1, \ldots, z_n$ independent observations from μ, and $z \sim \mu$ an independent copy. Risk is then $R(w) := \mathbf{E}_{\mu}\, l(w; z)$, its gradient $g(w) := R'(w)$, and $R^* := \inf_{w \in \mathcal{W}} R(w)$. $\mathbf{P}$ 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 $F'(u) := (F_1'(u), \ldots, F_d'(u))$ where for short, we write $F_j'(u) := \partial F(u)/\partial u_j$.

3.1 Sketch of the general argument

The analysis here requires only two steps: (i) A good estimate $\widehat{g} \approx g$ implies that approximate update (3) is near the optimal update. (ii) Under variance bounds, coordinate-wise M-estimation yields a good gradient estimate. We are then able to conclude that with enough samples and iterations, the output of Algorithm 1 can achieve an arbitrarily small excess risk. Here we spell out the key facts underlying this approach.

For the first step, let $w^* \in \mathbb{R}^d$ be a minimizer of R. When the risk R is strongly convex, then using well-established convex optimization theory (Nesterov 2004), we can easily control $\|w^*_{(t+1)} - w^*\|$ as a function of $\|w^*_{(t)} - w^*\|$ for any step $t \geq 0$. Thus to control $\|\widehat{w}_{(t+1)} - w^*\|$, in comparing the approximate case and optimal case, all that matters is the difference between $g(\widehat{w}_{(t)})$ and $\widehat{g}(\widehat{w}_{(t)})$ (Lemma 7). For the general case of convex R, since we cannot easily control the distance of the optimal update from any potential minimum, one can directly compare the trajectories of $\widehat{w}_{(t)}$ and $w^*_{(t)}$ over $t = 0, 1, \ldots, T$, which once again amounts to a comparison of g and $\widehat{g}$. 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 both $\widehat{g}$ and $\widehat{w}_{(t)}$ are based on a random sample $\{z_1, \ldots, z_n\}$, we need an estimation technique which admits guarantees for any step, with high probability over the random draw of this sample. A basic requirement is that

$$\mathbf{P}\left\{\max_{t \leq T} \|\widehat{g}(\widehat{w}_{(t)}) - g(\widehat{w}_{(t)})\| \leq \varepsilon(\delta)\right\} \geq 1 - \delta. \tag{7}$$

Of course this must be proved (see Lemmas 5 and 11), but if valid, then running Algorithm 1 for T steps, we can invoke (7) to get a high-probability event on which $\widehat{w}_{(T)}$ closely approximates the optimal GD output, up to the accuracy specified by ε. Naturally this $\varepsilon = \varepsilon(\delta)$ will depend on confidence level δ, which implies that to get $1 - \delta$ confidence intervals, the upper bound in (7) will increase as δ gets smaller.

In the LOCATE sub-routine of Algorithm 1, we construct a more robust estimate of the risk gradient than can be provided by the empirical mean, using an ancillary estimate of the gradient variance. This is conducted using a smooth truncation scheme, as follows. One important property of ρ in (4) is that for any $u \in \mathbb{R}$, one has

$$-\log(1 - u + Cu^2) \leq \rho'(u) \leq \log(1 + u + Cu^2) \tag{8}$$

for a fixed $C > 0$, a simple generalization of the key property utilized by Catoni (2012). For the Gudermannian function (Sect. 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 3 *(Concentration of M-estimates) For each coordinate* $j \in [d]$*, the estimates* $\widehat{\theta}_j$ *of (4) satisfy*

$$\frac{1}{2}|\widehat{\theta}_j - g_j(w)| \le \frac{C \operatorname{var}_\mu l'_j(w; z)}{s_j} + \frac{s_j \log(2\delta^{-1})}{n} \tag{9}$$

with probability no less than $1 - \delta$*, given large enough* n *and* s_j*.*

To get the tightest possible confidence interval as a function of $s_j > 0$, we must set

$$s_j^2 = \frac{Cn \operatorname{var}_\mu l'_j(w; z)}{\log(2\delta^{-1})},$$

from which we derive (6), with $\widehat{\sigma}_j^2$ corresponding to a computable estimate of $\operatorname{var}_\mu l'_j(w; z)$. If the variance over all choices of w is bounded by some $V < \infty$, then up to the variance estimates, we have $\|\widehat{g}(w) - g(w)\| \le O(\sqrt{dV \log(2d\delta^{-1})/n})$, with $\widehat{g} = \widehat{\theta}$ from Algorithm 1, yielding a bound for (7) free of w.

Remark 4 (Comparison with ERM-GD) As a reference example, assume we were to run ERM-GD, namely using an empirical mean estimate of the gradient. Using Markov's inequality, with probability $1 - \delta$ all we can guarantee is $\varepsilon \le O(\sqrt{d/(n\delta)})$. In the case of only assuming finite variance (matching our scenario), Catoni (2012, Proposition 6.2) derives a lower bound on the deviations of the empirical mean which scales with $1/\delta$. This implies that under a heavy-tailed setting such as ours, the upper bound given by Markov's inequality is in fact tight in terms of dependence on $1/\delta$. On the other hand, using the location estimate of Algorithm 1 provides guarantees with $\log(1/\delta)$ dependence on the confidence level, realizing an exponential improvement over the $1/\delta$ dependence of ERM-GD, and an appealing formal motivation for using M-estimates of location as a novel strategy.

3.2 Conditions and results

On the learning task, we make the following assumptions.

A1. $R(\cdot)$ is to be minimized over a closed, convex $\mathcal{W} \subset \mathbb{R}^d$ with diameter $\Delta < \infty$.
A2. $R(\cdot)$ and $l(\cdot; z)$ (for all z) are λ-smooth, convex, and continuously differentiable on $\mathcal{W}$.
A3. There exists $w^* \in \mathcal{W}$ at which $g(w^*) = 0$.
A4. Distribution μ satisfies $\operatorname{var}_\mu l'_j(w; z) \le V < \infty$, for all $w \in \mathcal{W}$, $j \in [d]$.

Algorithm 1 is run following (4), (5), and (6) as specified in Sect. 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

$$c_{min}^2 \le \frac{\widehat{\sigma}_j}{\operatorname{var}_\mu l'_j(w; z)} \le c_{max}^2, \quad j \in [d] \tag{10}$$

for all choices of $w \in \mathcal{W}$, and write $c_0 := (c_{max} + C/c_{min})$. For $1 - \delta$ confidence, we need a large enough sample; more precisely, for each w, it is sufficient if for each j,

$$\frac{1}{4} \ge \frac{C \log(2\delta^{-1})}{n} \left(1 + \frac{C \operatorname{var}_\mu l'_j(w; z)}{\widehat{\sigma}_j^2}\right). \tag{11}$$

For simplicity, fix a small enough step size,

$$\alpha_{(t)} = \alpha, \forall t \in \{0, \ldots, T-1\}, \quad \alpha \in (0, 2/\lambda). \tag{12}$$

Dependence on initialization is captured by two related factors $R_0 := R(w^*_{(0)}) - R^*$, and $D_0 := \|w^*_{(0)} - w^*\|$. Under this setup, we can control the estimation error.

Lemma 5 *(Uniform accuracy of gradient estimates) For all steps $t = 0, \ldots, T-1$ of Algorithm 1, we have*

$$\|\widehat{\theta}_{(t)} - g(\widehat{w}_{(t)})\| \leq \frac{\widetilde{\varepsilon}}{\sqrt{n}} := \frac{\lambda(\sqrt{d}+1)}{\sqrt{n}} + 2c_0\sqrt{\frac{dV(\log(2d\delta^{-1}) + d\log(3\Delta\sqrt{n}/2))}{n}}$$

with probability no less than $1-\delta$.

Remark 6 (On the technical assumptions) We remark here that the assumptions made above can be considered standard in the literature. Assuming at the very least bounded variance is essential (Devroye et al. 2015), and both Prasad et al. (2018, Section 7) and Lecué et al. (2018, Sections 2–3) utilize similar moment assumptions for controlling the statistical error. Convexity is generally needed to control the optimization error, and is utilized in all the robust gradient descent works we have cited (Chen et al. 2017b; Lecué et al. 2018; Prasad et al. 2018). We assume the smoothness of the loss (a Lipschitz condition on the gradient), whereas for example Lecué et al. (2018) work with Lipschitz losses. This excludes from their analysis the case of, for example, linear regression under the squared loss with unbounded noise, while this setting is valid under our assumptions.

Under strongly convex risk In addition to assumptions A1–A4, assume that R is κ-strongly convex. In this case, w^* in A3 is the unique minimum. First, we control the estimation error by showing that the approximate update (3) does not differ much from the optimal update (2).

Lemma 7 *(Minimizer control) Consider the general approximate GD update (3), with $\alpha_{(t)} = \alpha$ such that $0 < \alpha < 2/(\kappa+\lambda)$. Assume that (7) holds with bound ε. Write $\beta := 2\kappa\lambda/(\kappa+\lambda)$. Then, with probability no less than $1-\delta$, we have*

$$\|\widehat{w}_{(T)} - w^*\| \leq (1-\alpha\beta)^{T/2} D_0 + \frac{2\varepsilon}{\beta}.$$

Since Algorithm 1 indeed satisfies (7), as proved in Lemma 5, we can use the control over the parameter deviation provided by Lemma 7 and the smoothness of R to prove a finite-sample excess risk bound.

Theorem 8 *(Excess risk bounds) Write $\widehat{w}_{(T)}$ for the output of Algorithm 1 after T iterations, run such that (11)–(12) hold, with step size $\alpha_{(t)} = \alpha$ for all $0 < t < T$, as in Lemma 7. It follows that*

$$R(\widehat{w}_{(T)}) - R^* \leq \lambda(1-\alpha\beta)^T D_0^2 + \frac{4\lambda\widetilde{\varepsilon}}{\beta^2 n}$$

with probability no less than $1-\delta$, where $\widetilde{\varepsilon}$ is as given in Lemma 5.

Remark 9 (Interpretation of bounds) There are two terms in the upper bound of Theorem 8, an optimization term decreasing in T, and an estimation term decreasing in n. The optimization error decreases at the usual gradient descent rate, and due to the uniformity of the bounds obtained, the statistical error is not hurt by taking T arbitrarily large, thus with enough samples we can guarantee arbitrarily small excess risk. Finally, the most important assumption on the distribution is weak: finite second-order moments. If we assume finite kurtosis, the argument of Catoni (2012) can be used to create analogous guarantees for an explicit scale estimation procedure, yielding guarantees whether the data is sub-Gaussian or heavy-tailed an appealing robustness to the data distribution.

Remark 10 (Doing projected descent) The above analysis proceeds on the premise that $\widehat{w}_{(t)} \in \mathcal{W}$ holds after all the updates, $t \in [T]$. To enforce this, a standard variant of Algorithm 1 is to update as

$$\widehat{w}_{(t+1)} \leftarrow \pi_{\mathcal{W}}\left(\widehat{w}_{(t)} - \alpha_{(t)}\widehat{\theta}_{(t)}\right), \quad t \in \{0, \ldots, T-1\}$$

where $\pi_{\mathcal{W}}(u) := \arg\min_{v \in \mathcal{W}} \|u - v\|$. By A1, this projection is well-defined (Luenberger 1969, Sec. 3.12, Thm. 3.12). Using this fact, it follows that $\|\pi_{\mathcal{W}}(u) - \pi_{\mathcal{W}}(v)\| \leq \|u - v\|$ for all $u, v \in \mathcal{W}$, by which we can immediately show that Lemma 7 holds for the *projected robust gradient descent* version of Algorithm 1.

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 11 (Accuracy with variance information) *Conditioning on $\widehat{w}_{(t)}$ and running one scale-location sequence of Algorithm 1, with $\widehat{\sigma}_{(t)} = (\widehat{\sigma}_1, \ldots, \widehat{\sigma}_d)$ modified to satisfy $\widehat{\sigma}_j^2 = C \operatorname{var}_\mu l_j'(\widehat{w}_{(t)}; z)$, $j \in [d]$. It follows that*

$$\|\widehat{\theta}_{(t)} - g(\widehat{w}_{(t)})\| \leq 4\left(\frac{C \operatorname{trace}(\Sigma_{(t)}) \log(2d\delta^{-1})}{n}\right)^{1/2}$$

with probability no less than $1-\delta$, where $\Sigma_{(t)}$ is the covariance matrix of $l'(\widehat{w}_{(t)}; z)$.

Observe that in addition to being able to remove the c_0 factor, we also are able to replace the $d\,V$ factor observed in Lemma 5 with $\operatorname{trace}(\Sigma_{(t)})$. Certainly, if the variance in all coordinates is the same, then the trace will essentially scale with $d\,V$. However, if most coordinates have variance much smaller than the maximum coordinate, then this d coefficient can be reduced proportionally.

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 12 *(Control of update variance) Run Algorithm 1 as in Lemma 11, with arbitrary step-size $\alpha_{(t)}$. Then, for any $t < T$, taking expectation with respect to the sample $\{z_i\}_{i=1}^n$, conditioned on $\widehat{w}_{(t)}$, we have*

$$\mathbf{E}\,\|\widehat{w}_{(t+1)} - \widehat{w}_{(t)}\|^2 \leq 2\alpha_{(t)}^2\left(\frac{32Cd \operatorname{trace}(\Sigma_{(t)})}{n} + \|g(\widehat{w}_{(t)})\|^2\right).$$

In addition to these results, one can prove an improved version of Theorem 8 in a perfectly analogous fashion, using Lemma 11.

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 behavior of the robust gradient procedure proposed in Algorithm 1. We are interested in how these factors influence performance, both in an absolute sense and relative to the key competitors cited in Sect. 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 behavior, 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.

4.1 Controlled tests

Experimental setup Our first set of controlled numerical experiments uses a "noisy convex minimization" model, designed as follows. We construct a risk function taking a canonical quadratic form, setting $R(w) = \langle \Sigma w, w\rangle/2 + \langle w, u\rangle + c$, for pre-fixed constants $\Sigma \in \mathbb{R}^{d\times d}$, $u \in \mathbb{R}^d$, 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 $\mathbf{E}\, r(w) = R(w)$ for all $w \in \mathbb{R}^d$. In particular, here we generate observations $r_i(w) = (\langle w^* - w, x_i\rangle + \epsilon_i)^2/2$, $i \in [n]$, with x and ϵ independent of each other. Here w^* denotes the minimum, and we have that $\Sigma = \mathbf{E}\, xx^T$. The inputs x shall follow an isotropic d-dimensional Gaussian distribution throughout all the following experiments, meaning Σ is positive definite, and R is strongly convex.

We consider three main performance metrics in this section: the average excess empirical risk (based on the losses $r_1, \ldots, r_n$), the average excess risk (based on true risk R), and the variance of the risk. Averages and variances are computed over trials, with each trial corresponding to a new independent random sample. For all tests, the number of trials is 250.

For these first tests, we run three procedures. First is ideal gradient descent, denoted `oracle`, which has access to the true objective function R. This corresponds to (2). Second, as a standard approximate procedure (3) when R is unknown, we use ERM-GD, denoted `erm` and discussed at the start of Sect. 2, which approximates the optimal procedure using the empirical risk. Against these two benchmarks, we compare our Algorithm 1, denoted `rgd`, as a robust alternative for (3).

Impact of heavy-tailed noise Let us examine the results. We begin with a simple question: are there natural learning settings in which `rgd` outperforms ERM-GD? How does the same algorithm fare in situations where ERM is optimal? Under Gaussian noise, ERM-GD is effectively optimal (Lin and Rosasco 2016, Appendix C). We thus consider the case of Gaussian noise (mean 0, standard deviation 20) as a baseline, and use centered log-Normal noise (log-location 0, log-scale 1.75) as an archetype of asymmetric heavy-tailed data. Risk results for the two routines are given alongside training error in Fig. 2.

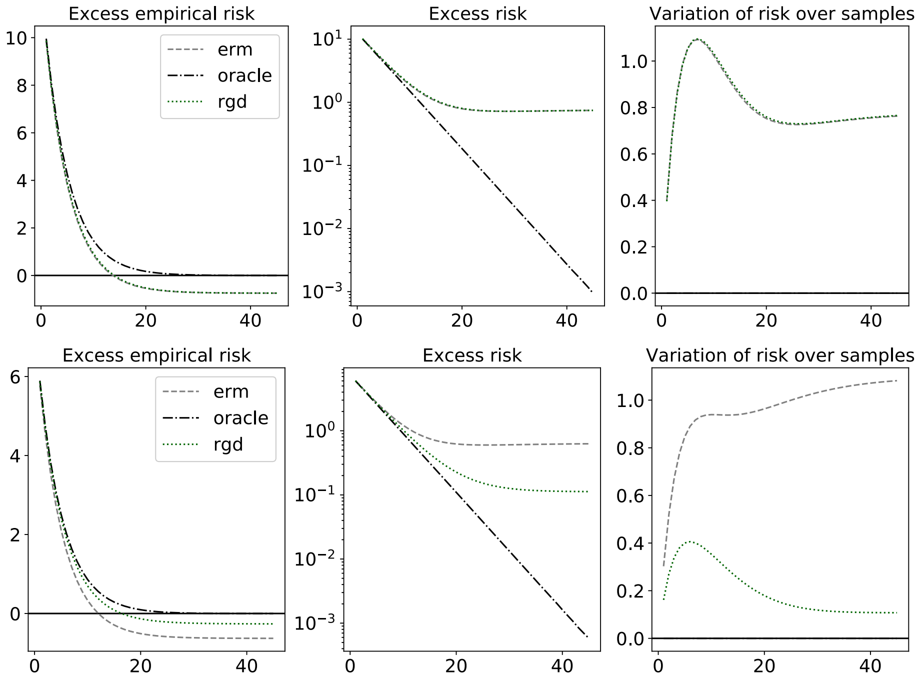

Fig. 2 Performance metrics as a function of iterative updates. Top row: Normal noise. Bottom row: log-Normal noise. Settings: $n = 500$, $d = 2$, $\alpha_{(t)} = 0.1$ for all t

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 departs from the ideal routine at a slower rate, a desirable trait.

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 independently impacts performance.

Impact of initialization Given a fixed data distribution and sample size, how does the quality of the initial guess impact learning performance? We consider three initializations of the form $w^* + \mathrm{Unif}[-\Delta, \Delta]$, with $\Delta = (\Delta_1, \ldots, \Delta_d)$, values ranging over $\Delta_j \in \{2.5, 5.0, 10.0\}$, $j \in [d]$, where larger Δ_j naturally correspond to potentially worse initialization. Relevant results are displayed in Fig. 3.

Some interesting observations can be made. That rgd matches erm when the latter is optimal is clear, but more importantly, we see that under heavy-tailed noise, rgd is far more robust to poor initial value settings. Indeed, while a bad initialization leads to a much worse solution in the limit for erm, we see that rgd is able to achieve the same performance as if it were initialized at a better value.

Impact of distribution It is possible for very distinct distributions to have exactly the same risk functions. Learning efficiency naturally depends heavily on the process generating the sample; the underlying optimization problem is the same, but the statistical inference task

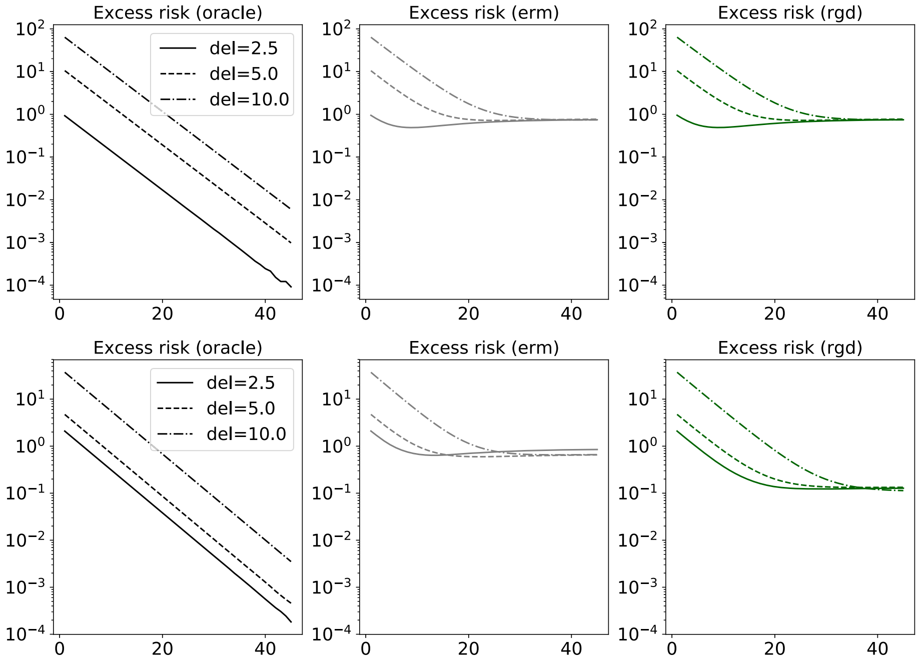

Fig. 3 Performance over iterations, under strong/poor initialization. Here del refers to Δ_j. Top row: Normal noise. Bottom row: log-Normal noise. Settings: $n = 500$, $d = 2$, $\alpha_{(t)} = 0.1$ for all t

changes. 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 Fig. 4.

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. In the log-Normal case, as our previous example suggested, the performance of erm degrades rather dramatically, and a clear gap in performance appears, which grows wider as the variance increases. 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 Sect. 3.2. In addition, it suggests that our simple dispersion-based technique ($\widehat{\sigma}_j$ settings in Sect. 2.2) provides tolerable accuracy, implying a small enough c_0 factor, and reinforcing the insights from the proof of concept case seen in Fig. 2.

Impact of sample size Since the true risk is unknown, the size and quality of the sample $\{z_i\}_{i=1}^n$ 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 parameters fixed. Figure 5 gives the accuracy of erm and rgd in tests analogous to those above, using common initial values across methods, and $n \in \{10, 40, 160, 640\}$.

Both algorithms naturally show monotonic performance improvements as the sample size grows, but the most salient feature of these figures is the performance of rgd under heavy-tailed noise, especially when sample sizes are small. When our data may be heavy-tailed, this

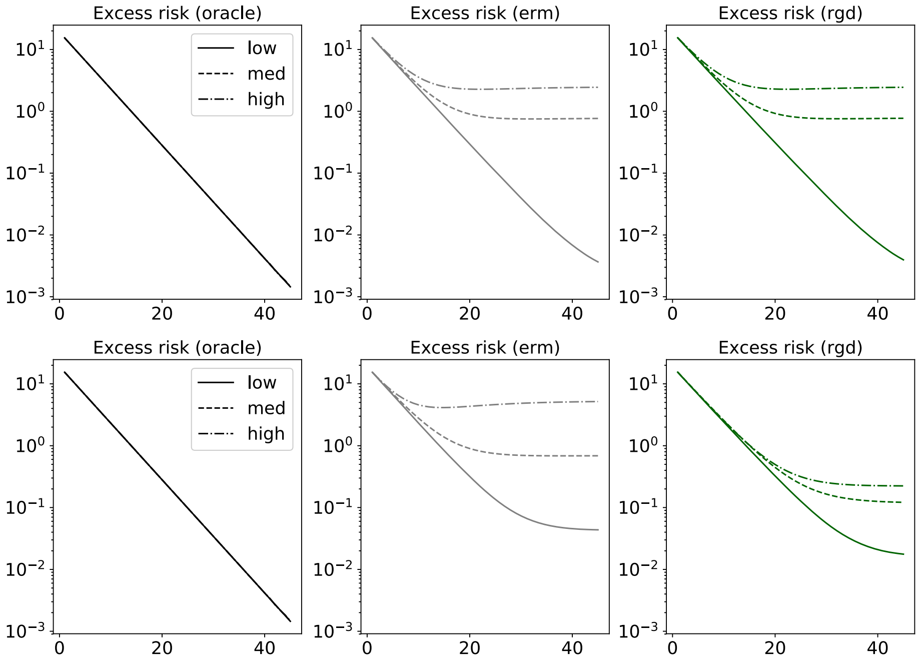

Fig. 4 Performance over iterations, under varying noise intensities. Here `low`, `med`, and `high` refer to the three noise distribution settings described in the main text. Settings: $n = 500$, $d = 2$, $\alpha_{(t)} = 0.1$ for all t

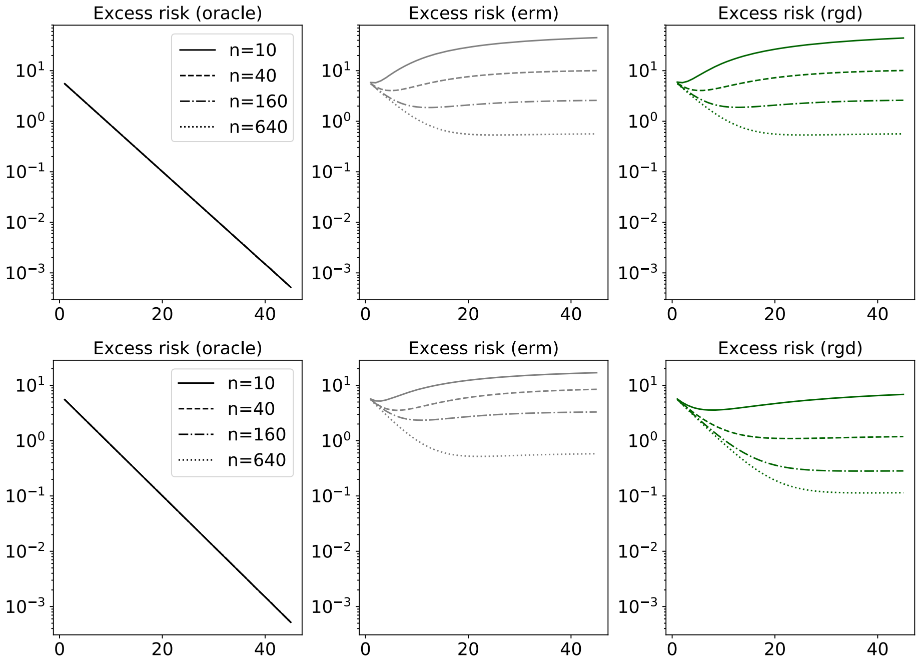

Fig. 5 Performance over iterations, under different sample sizes. Settings: $d = 2$, $\alpha_{(t)} = 0.1$ for all t

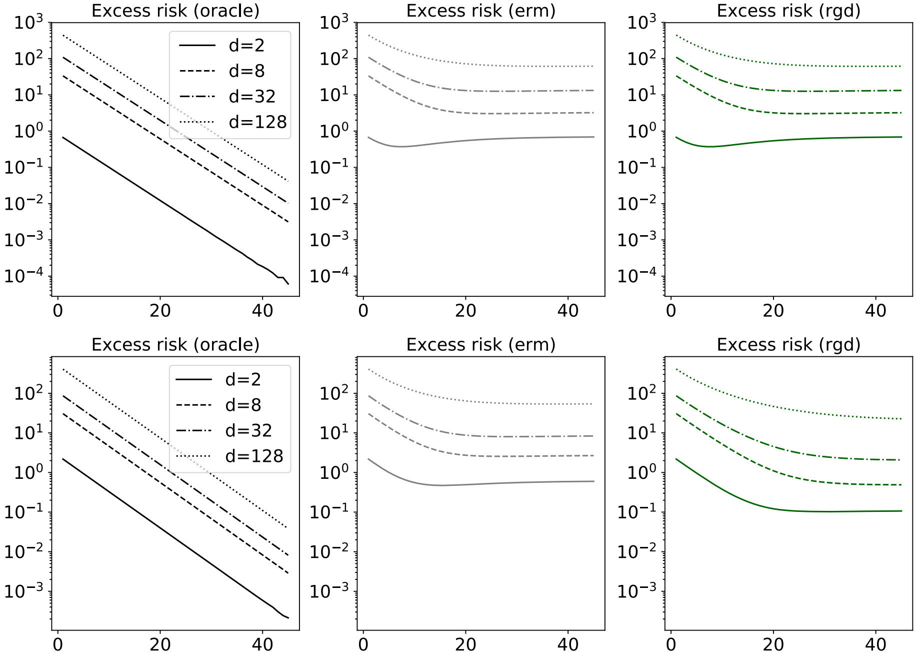

Fig. 6 Performance over iterations, under increasing dimension. Settings: $n = 500$, $\alpha_{(t)} = 0.1$ for all t

provides clear evidence that the proposed RGD methods can achieve better generalization than ERM-GD with less data, in less iterations.

Impact of dimension Given a fixed number of observations, the role of dimension d, namely the number of parameters to be determined, plays an important from both practical and theoretical standpoints, as seen in the error bounds of Sect. 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 6 shows the risk of `erm` and `rgd` in tests just as above, with $d \in \{2, 8, 32, 128\}$.

As the dimension increases, since the sample size is fixed, both non-oracle algorithms tend to require more iterations to converge. The key difference is that under heavy tails, the excess risk achieved by our proposed method is clearly superior to ERM-GD over all d settings, while matching it in the case of Gaussian noise. In particular, `erm` hits bottom very quickly in higher dimensions, while `rgd` continues to improve for more iterations, presumably due to updates which are closer to that of the optimal (2).

Comparison with robust loss minimizer Another interesting question: instead of paying the overhead to robustify gradient estimates (d dimensions to handle), why not just make robust estimates of the risk itself, and use those estimates to fuel an iterative optimizer? Just such a procedure is analyzed by Brownlees et al. (2015) (denoted `bjl` henceforth). To compare our gradient-centric approach with their loss-centric approach, we implement `bjl` using the non-linear conjugate gradient method of Polak and Ribière (Nocedal and Wright 1999), which is provided by `fmin_cg` in the `optimize` module of the SciPy scientific computation library (default maximum number of iterations is $200d$). This gives us a standard first-order general-purpose optimizer 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

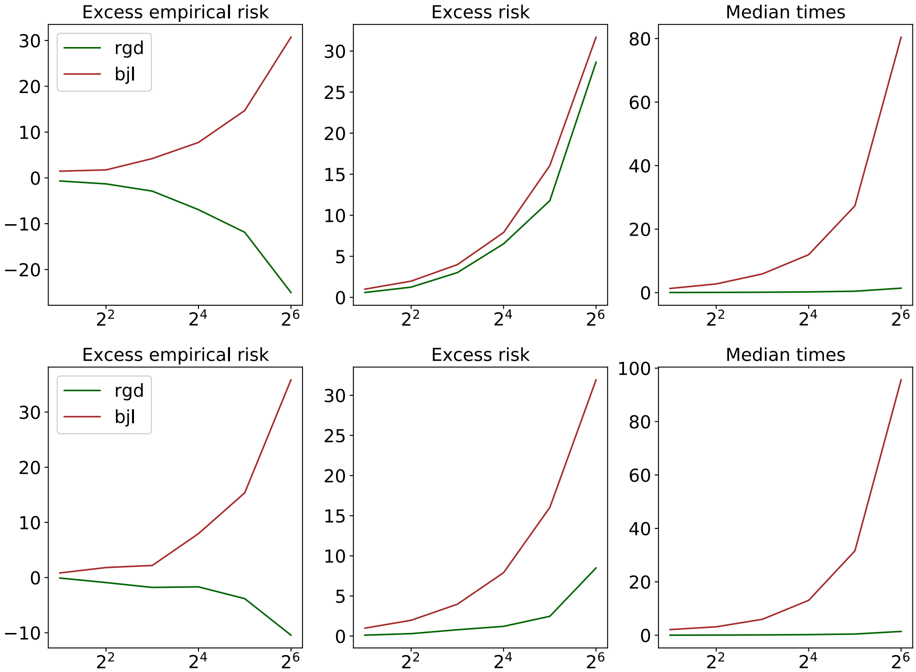

Fig. 7 Comparison of our robust gradient-based approach with the robust objective-based approach. Top: Normal noise. Bottom: log-Normal noise. Performance is given as a function of the number of d, the number of parameters to optimize, given in $\log_2$ scale. Settings: $n = 500$, $\alpha_{(t)} = 0.1$ for all t (Color figure online)

time is computed using the Python `time` module. To give a simple comparison between `bjl` and `rgd`, we run multiple independent trials of the same task, starting both routines at the same (random) initial value each time, generating a new sample, and repeating the whole process for different settings of $d = 2, 4, 8, 16, 32, 64$. Median times taken over all trials (for each d setting) are recorded, and presented in Fig. 7 alongside performance results.

From the results, we can see that while the performance of both methods is similar in low dimensions and under Gaussian noise, in higher dimensions and under heavy-tailed noise, our proposed `rgd` realizes much better performance in much less time. Regarding excess empirical risk, random deviations in the sample cause the minimum of the empirical risk function to deviate away from w^*, causing the `rgd` solution to be closer to the ERM solution in higher dimensions. On the other hand, `bjl` is minimizing a different objective function. It should be noted that there are assuredly other ways of approaching the `bjl` optimization task, but all of which require minimizing an implicitly defined objective which need not be convex. We believe that `rgd` provides a simple and easily implemented alternative, while still utilizing the same statistical principles.

Regression application 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 $y_i = x_i^T w^* + \epsilon_i$, $i \in [n]$. Distinct experimental conditions are delimited by the setting of (n, d) and μ. Inputs 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, namely parameter settings designed such that $\mathrm{sd}_{\mu}(\epsilon)$ monotonically increases over the range 0.3–20.0, approximately linearly over the levels.

To capture a range of signal/noise ratios, for each trial, $w^{*} \in \mathbb{R}^{d}$ 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 $w^{*} = (w_{i_1}, \ldots, w_{i_d})$. Computing $\mathrm{SN}_{\mu} = \|w^{*}\|_2^2 / \mathrm{var}_{\mu}(\epsilon)$, we have $0.2 \leq SN_{\mu} \leq 1460.6$. Noise families: log-logistic (denoted `llog` in figures), log-Normal (`lnorm`), Laplace (`lap`), and arcsine (`asin`). Even with just these four, we capture bounded, sub-Gaussian (arcsine), and sub-exponential (Laplace) noise, and heavy-tailed data both with (log-Normal) and without (log-logistic) finite higher-order moments. Results for many more noise distributions are given in “Appendix B”.

Our key performance metric of interest is off-sample prediction error, here computed as excess RMSE on an independent large testing set, averaged over trials. 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 $\widehat{w}(k)$ is determined. To approximate the ℓ_2-risk, compute root mean squared test error $e_k(\widehat{w}) := (m^{-1}\sum_{i=1}^{m}(\widehat{w}^{T}x_{k,i} - y_{k,i})^2)^{1/2}$, and output prediction error as the average of normalized errors $e_k(\widehat{w}(k)) - e_k(w^{*}(k))$ taken over all K trials. While n values vary, in all experiments we fix $K = 250$ and test size $m = 1000$.

Regarding the competing methods, classical choices are ordinary least squares (ℓ_2-ERM, denoted `OLS`) and least absolute deviations (ℓ_1-ERM, `LAD`). We also look at four recent methods of practical and theoretical importance described in Sect. 1, namely the robust regression routines of Hsu and Sabato (2016) (`HS`) and Minsker (2015) (`Minsker`), and the robust gradient descent methods of Prasad et al. (2018) (`Pras18`) and Lecué et al. (2018) (`Lec18`). For `HS`, we used the source published online by the authors. For `Minsker`, 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 (2000, Eqn. 2.6), and the number of partitions is set to $\max(2, \lfloor n/(2d) \rfloor)$. For `Pras18`, we follow their Algorithm 3 and partition into $1 + \lfloor 3.5 \log(\delta^{-1}) \rfloor$ blocks. This same setting is used for the partitioning done in `Lec18`. Both of these robust gradient descent methods are initialized to the `OLS` solution. For comparison to these methods, 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 for all conditions. Illustrative results are given in Fig. 8.

First we fix the model dimension d, and evaluate performance as sample size n ranges from very small to quite large (top row of Fig. 8). We see that regardless of distribution, `rgd` effectively matches the optimal convergence of OLS in the `norm` and `tri_s` cases, and is resilient 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 look at performance over noise settings, from negligible noise to significant noise with potentially infinite higher-order moments (middle row of Fig. 8). We 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

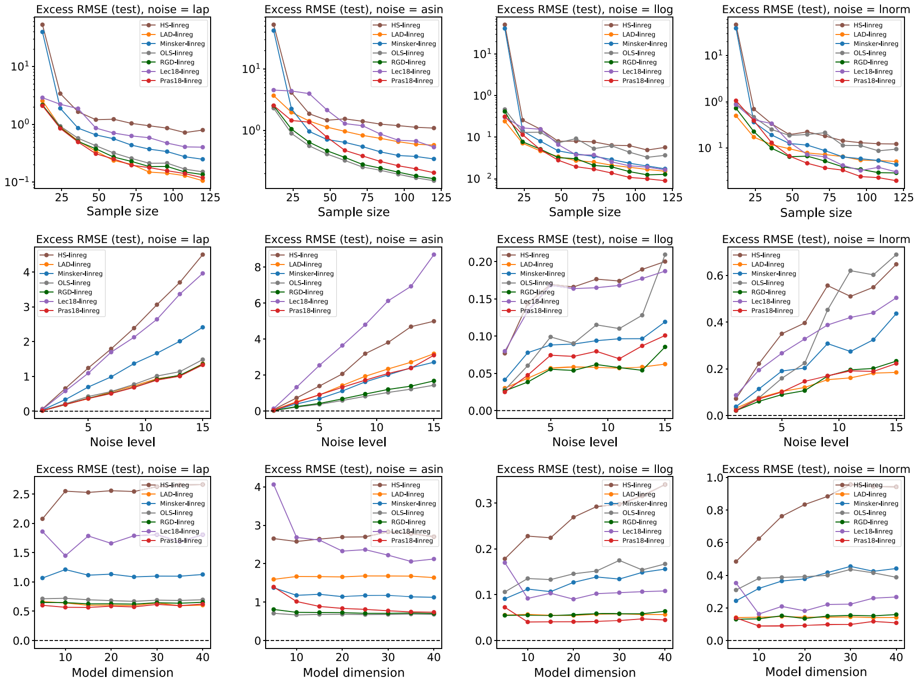

Fig. 8 Top row: Prediction error over sample size $12 \leq n \leq 122$, fixed $d = 5$, noise level = 8. Center row: Prediction error over noise levels, for $n = 30, d = 5$. Bottom row: Prediction error over dimensions $5 \leq d \leq 40$, with ratio $n/d = 6$ fixed, and noise level = 8. Each column corresponds to a distinct noise family (Color figure online)

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.

Finally, we fix the ratio n/d and look at the role played by increasingly large dimension (bottom row of Fig. 8). We see that for all distributions, the performance of `rgd` is essentially constant. This coincides with the theory of Sect. 3.2, and our intuition since Algorithm 1 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.

4.2 Application to real-world benchmarks

To close out this section, and to gain some additional perspective on algorithm performance, we shift our focus to some nascent applications to real-world benchmark data sets.

Having already paid close attention to regression models in the previous section, here we consider applications of robust gradient descent to classification tasks, under both binary and multi-class settings. The model assumed is standard multi-class logistic regression: if the number of classes is C, and the number of input features is F, then the total number of parameters to be determined is $d = (C - 1)F$. The loss function is convex in the parameters, and its partial derivatives all exist, so the model aligns well with our problem setting of interest. In addition, a squared ℓ_2-norm regularization term $a\|w\|^2$ is added to the loss, with a varying over datasets (see below). All learning algorithms are given a fixed budget of gradient computations, set here to $20n$, where n is the size of the training set made available to the learner.

We use three well-known data sets for benchmarking: the CIFAR-10 data set of tiny images,[2] the MNIST data set of handwritten digits,[3] and the protein homology dataset made popular by its inclusion in the KDD Cup.[4] For all data sets, we carry out 10 independent trials, with training and testing tests randomly sampled as will be described shortly. For all datasets, we normalize the input features to the unit interval $[0, 1]$ in a per-dimension fashion. For CIFAR-10, we average the RGD color channels to obtain a single greyscale channel. As a result, $F = 1024$. There are ten classes, so $C = 10$, meaning $d = (C - 1)F = 9216$. We take a sample size of $n = 4d = 36864$ for training, with the rest for testing, and set $a = 0.001$. For MNIST, we have $F = 784$ and once again $C = 10$, so $d = 7056$. As with the previous dataset, we set $n = 4d = 28224$, and $a = 0.0001$. Note that both of these datasets have all classes in equal proportions, so with uniform random sampling, class frequencies are approximately equal in each trial. On the other hand, the protein homology dataset (binary classification) has highly unbalanced labels, with only 1296 positive labels out of over 145,000 observations. We thus take random samples such that the training and test sets are balanced. For each trial, we randomly select 296 positively labeled examples, and the same amount of negatively labeled examples, yielding a test set of 592 examples. As for the training set size, we use the rest of the positive labels (1000 examples) plus a random selection of 1000 negatively labeled examples, so $n = 2000$, and with $C = 2$ and $F = 74$, we have $d = 74$. Regularization parameter a is 0.001. For all datasets, the parameter weights are initialized uniformly over the interval $[-0.05, 0.05]$.

Regarding the competing methods used, we test out a random mini-batch version of robust gradient descent given in Algorithm 1, with mini-batch sizes ranging over $\{5, 10, 15, 20\}$, roughly on the order of $n^{-1/4}$ for the largest datasets. We also consider a mini-batch in the sense of randomly selecting coordinates to robustify: select $\min(100, d)$ indices randomly at each iteration, and run the RGD sub-routine on just these coordinates, using the sample mean for all the rest. Furthermore, we considered several minor alterations to the original routine, including using $\log\cosh(\cdot)$ instead of the Gudermannian function for ρ, updating the scale much less frequently (compared to every iteration), and different choices of χ for re-scaling. We compare our proposed algorithm with stochastic gradient descent (SGD), and stochastic variance-reduced gradient descent (SVRG) proposed by Johnson and Zhang (2013). For each method, pre-fixed step sizes ranging over $\{0.0001, 0.001, 0.01, 0.05, 0.10, 0.15, 0.20\}$ are tested. SGD uses mini-batches of size 1, as does the inner loop of SVRG. The outer loop of SVRG continues until the budget is spent, with the inner loop repeating $n/2$ times.

Representative results are given in Fig. 9. For each of the three methods of interest, and each dataset, we chose the top two performance settings, displayed as `*_1` and `*_2` respectively. Here "top performance" is measured by the median value of the last five iterations. We see that in general, robust gradient descent is competitive with the best settings of these well-known routines, has minimal divergence between the performance of its first- and second-best settings, and in the case of smaller data sets (protein homology), indeed significantly outperforms the competitors. While these are simply nascent applications of RGD, the strong initial performance suggests that further investigation of efficient strategies under high-dimensional data is a promising direction.

[2] http://www.cs.toronto.edu/~kriz/cifar.html.

[3] http://yann.lecun.com/exdb/mnist/.

[4] http://www.kdd.org/kdd-cup/view/kdd-cup-2004/Tasks.

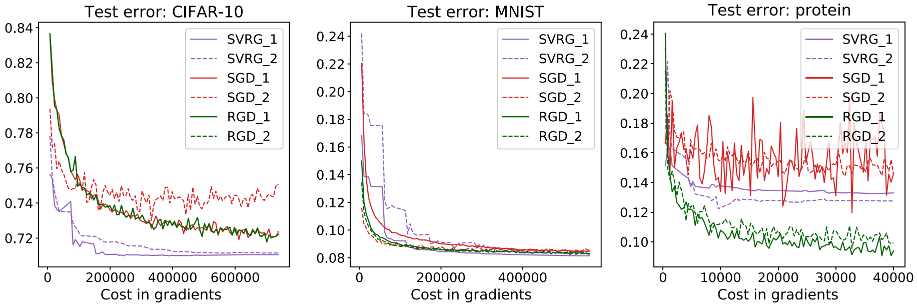

Fig. 9 Test error (misclassification rate) over budget spent, as measured by gradient computations, for the top two performers within each method class. Each plot corresponds to a distinct dataset (Color figure online)

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 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 efficiency 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.

Acknowledgements This work was partially supported by the Grant-in-Aid for JSPS Research Fellows.

A Technical appendix

A.1 Preliminaries

Our generic data shall be denoted by $z \in \mathcal{Z}$. 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 $z_1, \ldots, z_n$. We shall work with loss function $l : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}_+$ throughout, with $l(\cdot; z)$ assumed differentiable for each $z \in \mathcal{Z}$. 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 a measurable function. Expectation is written $\mathbf{E}_\mu f(z) := \int f \, d\mu$, with variance $\text{var}_\mu f(z)$ defined analogously. For d-dimensional Euclidean space $\mathbb{R}^d$, the usual (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 $F'(u) := (F_1'(u), \ldots, F_d'(u))$ where for short, we write $F_j'(u) := \partial F(u)/\partial u_j$. For integer k, write $[k] := \{1, \ldots, k\}$ for all the positive integers from 1 to k. Risk shall be denoted $R(w) := \mathbf{E}_\mu \, l(w; z)$, and its gradient $g(w) := R'(w)$. We make a running assumption that we can differentiate under the integral sign in each coordinate (Ash and Doleans-Dade 2000; Talvila 2001), namely that

$$g(w) = \left(\mathbf{E}_\mu \frac{\partial l(w; z)}{\partial w_1}, \ldots, \mathbf{E}_\mu \frac{\partial l(w; z)}{\partial w_d} \right). \tag{13}$$

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 $w_1, w_2 \in \mathcal{W}$ we have $|F(w_1) - F(w_2)| \leq \lambda \|w_1 - w_2\|$. We say that F is λ-*smooth* if F' is λ-Lipschitz. Finally, F is *strongly convex* with parameter $\kappa > 0$ if for all $w_1, w_2 \in \mathcal{W}$,

$$F(w_1) - F(w_2) \geq \langle F'(w_2), w_1 - w_2 \rangle + \frac{\kappa}{2} \|w_1 - w_2\|^2$$

for any norm $\|\cdot\|$ on $\mathcal{W}$, though we shall be assuming $\mathcal{W} \subseteq \mathbb{R}^d$. If there exists $w^* \in \mathcal{W}$ such that $F'(w^*) = 0$, then it follows that w^* is the unique minimum of F on $\mathcal{W}$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable, convex, λ-smooth function. The following basic facts will be useful: for any choice of $u, v \in \mathbb{R}^d$, we have

$$f(u) - f(v) \leq \frac{\lambda}{2} \|u - v\|^2 + \langle f'(v), u - v \rangle \tag{14}$$

$$\frac{1}{2\lambda} \|f'(u) - f'(v)\|^2 \leq f(u) - f(v) - \langle f'(v), u - v \rangle. \tag{15}$$

Proofs of these results can be found in any standard text on convex optimization, e.g. (Nesterov 2004).

We shall leverage a special type of M-estimator here, built using the following convenient class of functions.

Definition 13 (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 (2012).

Example 14 (Valid ρ choices) In addition to the Gudermannian function (Sect. 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.

A.2 Proofs

Proof of Lemma 3 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 Definition 13. 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 (2009, 2012). 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 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 Markov'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

$$\frac{s}{n}\sum_{i=1}^{n}\psi_s(x_i-u) \leq b(u)+\frac{s\log(\delta^{-1})}{n}. \tag{16}$$

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 inequality (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

$$\begin{aligned}u_+ &:= \mathbf{E}\,x+\frac{s}{2C}\frac{(1-\sqrt{1-D})(1+\sqrt{1-D})}{1+\sqrt{1-D}}\\ &= \mathbf{E}\,x+\frac{s}{2C}\frac{D}{1+\sqrt{1-D}}\\ &\leq \mathbf{E}\,x+sD/2C,\end{aligned} \tag{17}$$

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), note that $\widehat{x}$ equivalently satisfies $\sum_{i=1}^{n} \psi_s(\widehat{x} - x_i) = 0$. Using (16), 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 (16) is a monotonically decreasing function of u, we have immediately that $\widehat{x} \leq u_+$ on the event that (16) holds, which has probability at least $1-\delta$. Then leveraging (17), 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

$$|\widehat{x} - \mathbf{E}\, x| \leq 2\left(\frac{C \operatorname{var} x}{s} + \frac{s \log(\delta^{-1})}{n}\right) \tag{18}$$

holds with probability no less than $1-2\delta$. Setting the x_i to $l'_j(w; z_i)$ for $j \in [d]$ and some $w \in \mathbb{R}^d$, $i \in [n]$, 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 5 For any fixed w and $j \in [d]$, note that

$$\begin{aligned} |\widehat{\theta}_j - g_j(w)| &\leq \varepsilon_j \\ &:= 2\left(\frac{C \operatorname{var}_\mu l'_j(w; z)}{s_j} + s_j \log(2\delta^{-1})\right) \\ &= 2\sqrt{\frac{\log(2\delta^{-1})}{n}}\left(\frac{C \operatorname{var}_\mu l'_j(w; z)}{\widehat{\sigma}_j} + \widehat{\sigma}_j\right) \qquad (19) \\ &\leq \varepsilon^* := 2\sqrt{\frac{V \log(2\delta^{-1})}{n}} c_0 \qquad (20) \end{aligned}$$

holds with probability no less than $1-\delta$. The first inequality holds via direct application of Lemma 3, which holds under (11) and using ρ which satisfies (8). The equality follows immediately from (6). The final inequality follows from A4 and (10), along with the definition of c_0.

Making the dependence on w explicit with $\widehat{\theta}_j = \widehat{\theta}_j(w)$, an important question to ask is how sensitive this estimator is to a change in w. Say we perturb w to $\widetilde{w}$, so that $\|w - \widetilde{w}\| = a > 0$. By A2, for any sample we have

$$\|l'(w; z_i) - l'(\widetilde{w}; z_i)\| \leq \lambda \|w - \widetilde{w}\| = \lambda a, \quad i \in [n]$$

which immediately implies $|l'_j(w; z_i) - l'_j(\widetilde{w}; z_i)| \leq \lambda a$ for all $j \in [d]$ as well. That is, the maximum that any data point can move in either direction is λa. Given a sample of $n \geq 1$ points, consider the impact that the a-sized shift from w to $\widetilde{w}$ has on $\widehat{\theta}_j(w)$ shifting to $\widehat{\theta}_j(\widetilde{w})$. Without loss of generality, say all the points shifted to the right by the maximum amount, that is, $l'_j(\widetilde{w}; z_i) - l'_j(w; z_i) = \lambda a$ for all $i \in [n]$. Then note that

$$\begin{aligned}\frac{s}{n}\sum_{i=1}^{n}\psi_s\left(\widehat{\theta}_j(w)+\lambda a-l'_j(\widetilde{w};z_i)\right) &= \frac{s}{n}\sum_{i=1}^{n}\psi_s\left(\widehat{\theta}_j(w)+\lambda a-(l'_j(w;z_i)+\lambda a)\right)\\ &= \frac{s}{n}\sum_{i=1}^{n}\psi_s\left(\widehat{\theta}_j(w)-l'_j(w;z_i)\right)\\ &= 0\end{aligned}$$

by definition of $\widehat{\theta}_j(w)$. It thus follows that $\widehat{\theta}_j(\widetilde{w})=\widehat{\theta}_j(w)+\lambda a$. Next, modify our assumption on the data to allow for at least one point to have moved to the right less than the maximum amount, namely that $l'_j(\widetilde{w};z_i)-l'_j(w;z_i)\leq \lambda a$ for some i. In this case, by monotonicity of ψ, it follows that $\widehat{\theta}_j(\widetilde{w})\leq\widehat{\theta}_j(w)+\lambda a$. An identical argument can be made for movement in the negative direction, implying that no matter how the points are distributed after the perturbation from w to $\widetilde{w}$, the change from $\widehat{\theta}_j(w)$ to $\widehat{\theta}_j(\widetilde{w})$ can be no more than λa. That is, smoothness of the loss function immediately implies a Lipschitz property of the estimator:

$$|\widehat{\theta}_j(w)-\widehat{\theta}_j(\widetilde{w})|\leq\lambda\|w-\widetilde{w}\|.$$

Considering the vector of estimates $\widehat{\theta}(w):=(\widehat{\theta}_1(w),\ldots,\widehat{\theta}_d(w))$, we then have

$$\|\widehat{\theta}(w)-\widehat{\theta}(\widetilde{w})\|\leq\sqrt{d}\lambda\|w-\widetilde{w}\|. \tag{21}$$

This will be useful for proving uniform bounds on the estimation error shortly.

First, let's use these one-dimensional results for statements about the vector estimator of interest. In d dimensions, using $\widehat{\theta}(w)$ just defined for any pre-fixed w, then for any $\varepsilon>0$ we have

$$\begin{aligned}\mathbf{P}\left\{\|\widehat{\theta}(w)-g(w)\|>\varepsilon\right\} &= \mathbf{P}\left\{\|\widehat{\theta}(w)-g(w)\|^2>\varepsilon^2\right\}\\ &\leq\sum_{j=1}^{d}\mathbf{P}\left\{|\widehat{\theta}_j(w)-g_j(w)|>\frac{\varepsilon}{\sqrt{d}}\right\}.\end{aligned}$$

Using the notation of ε_j and ε^* from (19), filling in $\varepsilon=\sqrt{d}\varepsilon^*$, we thus have

$$\begin{aligned}\mathbf{P}\left\{\|\widehat{\theta}(w)-g(w)\|>\sqrt{d}\varepsilon^*\right\} &\leq\sum_{j=1}^{d}\mathbf{P}\left\{|\widehat{\theta}_j(w)-g_j(w)|>\varepsilon^*\right\}\\ &\leq\sum_{j=1}^{d}\mathbf{P}\left\{|\widehat{\theta}_j(w)-g_j(w)|>\varepsilon_j\right\}\\ &\leq d\delta.\end{aligned}$$

The second inequality is because $\varepsilon_j\leq\varepsilon^*$ for all $j\in[d]$. It follows that the event

$$\mathcal{E}(w):=\left\{\|\widehat{\theta}(w)-g(w)\|>2\sqrt{\frac{dV\log(2d\delta^{-1})}{n}}c_0\right\}$$

has probability $\mathbf{P}\,\mathcal{E}(w)\leq\delta$. In practice, however, $\widehat{w}_{(t)}$ for all $t>0$ will be random, and depend on the sample. We seek uniform bounds using a covering number argument. By A1, $\mathcal{W}$ is closed and bounded, and thus compact, and it requires no more than $N_\epsilon:=\lfloor(3\Delta/2\epsilon)^d\rfloor$ balls of ϵ radius to cover $\mathcal{W}$, where Δ is the diameter of $\mathcal{W}$.[5] Write the centers of these ϵ

[5] This is a basic property of covering numbers for compact subsets of Euclidean space (Kolmogorov 1993).

balls by $\{\widetilde{w}_1, \ldots, \widetilde{w}_{N_\epsilon}\}$. Given $w \in \mathcal{W}$, denote by $\widetilde{w} = \widetilde{w}(w)$ the center closest to w, which satisfies $\|w - \widetilde{w}\| \leq \epsilon$. Estimation error is controllable using the following new error terms:

$$\|\widehat{\theta}(w) - g(w)\| \leq \|\widehat{\theta}(w) - \widehat{\theta}(\widetilde{w})\| + \|g(w) - g(\widetilde{w})\| + \|\widehat{\theta}(\widetilde{w}) - g(\widetilde{w})\|. \tag{22}$$

The goal is to be able to take the supremum over $w \in \mathcal{W}$. We bound one term at a time. The first term can be bounded, for any $w \in \mathcal{W}$, by (21) just proven. The second term can be bounded by

$$\|g(w) - g(\widetilde{w})\| \leq \lambda \|w - \widetilde{w}\| \tag{23}$$

which follows immediately from A2. Finally, for the third term, fixing any $w \in \mathcal{W}$, $\widetilde{w} = \widetilde{w}(w) \in \{\widetilde{w}_1, \ldots, \widetilde{w}_{N_\epsilon}\}$ is also fixed, and can be bounded on the δ event $\mathcal{E}(\widetilde{w})$ just defined. The important fact is that

$$\sup_{w \in \mathcal{W}} \left\|\widehat{\theta}(\widetilde{w}(w)) - g(\widetilde{w}(w))\right\| = \max_{k \in [N_\epsilon]} \left\|\widehat{\theta}(\widetilde{w}_k) - g(\widetilde{w}_k)\right\|.$$

We construct a “good event” naturally as the event in which the bad event $\mathcal{E}(\cdot)$ holds for no center on our ϵ-net, namely

$$\mathcal{E}_+ = \left(\bigcap_{k \in [N_\epsilon]} \mathcal{E}(\widetilde{w}_k) \right)^c.$$

Taking a union bound, we can say that with probability no less than $1 - \delta$, for all $w \in \mathcal{W}$, we have

$$\|\widehat{\theta}(\widetilde{w}(w)) - g(\widetilde{w}(w))\| \leq 2\sqrt{\frac{dV \log(2dN_\epsilon \delta^{-1})}{n}} c_0. \tag{24}$$

Taking the three new bounds together, we have with probability no less than $1 - \delta$ that

$$\sup_{w \in \mathcal{W}} \|\widehat{\theta}(w) - g(w)\| \leq \lambda\epsilon(\sqrt{d} + 1) + 2\sqrt{\frac{dV \log(2dN_\epsilon \delta^{-1})}{n}} c_0.$$

Setting $\epsilon = 1/\sqrt{n}$ we have

$$\sup_{w \in \mathcal{W}} \|\widehat{\theta}(w) - g(w)\| \leq \frac{\lambda(\sqrt{d} + 1)}{\sqrt{n}} + 2c_0\sqrt{\frac{dV(\log(2d\delta^{-1}) + d\log(3\Delta\sqrt{n}/2))}{n}}.$$

Since every step of Algorithm 1 (with orthogonal projection if required) has $\widehat{w}_{(t)} \in \mathcal{W}$, the desired result follows from this uniform confidence interval. □

Proof of Lemma 7 Given $\widehat{w}_{(t)}$, running the approximate update (3), we have

$$\begin{aligned}\|\widehat{w}_{(t+1)} - w^*\| &= \|\widehat{w}_{(t)} - \alpha\widehat{g}(\widehat{w}_{(t)}) - w^*\| \\ &\leq \|\widehat{w}_{(t)} - \alpha g(\widehat{w}_{(t)}) - w^*\| + \alpha\|\widehat{g}(\widehat{w}_{(t)}) - g(\widehat{w}_{(t)})\|.\end{aligned}$$

The first term looks at the distance from the target given an optimal update, using g. Using the κ-strong convexity of R, via Nesterov (2004, Thm. 2.1.15) it follows that

$$\|\widehat{w}_{(t)} - \alpha g(\widehat{w}_{(t)}) - w^*\|^2 \leq \left(1 - \frac{2\alpha\kappa\lambda}{\kappa + \lambda}\right) \|\widehat{w}_{(t)} - w^*\|^2.$$

Writing $\beta := 2\kappa\lambda/(\kappa + \lambda)$, the coefficient becomes $(1 - \alpha\beta)$.

To control the second term simply requires unfolding the recursion. By hypothesis, we can leverage (7) to bound the statistical estimation error by ε for every step, all on the same $1-\delta$ "good event." For notational ease, write $a := \sqrt{1-\alpha\beta}$. On the good event, we have

$$\begin{aligned}\|\widehat{w}_{(t+1)} - w^*\| &\leq a^{t+1}\|\widehat{w}_{(0)} - w^*\| + \alpha\varepsilon\left(1 + a + a^2 + \cdots + a^t\right)\\ &= a^{t+1}\|\widehat{w}_{(0)} - w^*\| + \alpha\varepsilon\frac{(1-a^{t+1})}{1-a}.\end{aligned}$$

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 8 Using strong convexity and (14), we have that

$$\begin{aligned}R(\widehat{w}_{(T)}) - R^* &\leq \frac{\lambda}{2}\|\widehat{w}_{(T)} - w^*\|^2\\ &\leq \lambda(1-\alpha\beta)^T D_0^2 + \frac{4\lambda\varepsilon^2}{\beta^2}.\end{aligned}$$

The latter inequality holds by direct application of Lemma 7, followed by the elementary fact $(a+b)^2 \leq 2(a^2+b^2)$. The particular value of ε under which Lemma 7 is valid (i.e., under which (7) holds) is given by Lemma 5. Filling in ε with this concrete setting yields the desired result. □

Proof of Lemma 11 As in the result statement, we write

$$\Sigma_{(t)} := \mathbf{E}_\mu \left(l'(\widehat{w}_{(t)};z) - g(\widehat{w}_{(t)})\right)\left(l'(\widehat{w}_{(t)};z) - g(\widehat{w}_{(t)})\right)^T, \quad w \in \mathcal{W}.$$

Running this modified version of Algorithm 1, by minimizing the bound on the right-hand side of the inequality in Lemma 3 as a function of scale s_j, $j \in [d]$, plugging in the optimal scale setting to Lemma 3 yields that the estimates $\widehat{\theta}_{(t)} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_d)$ at each step t incur an error of

$$|\widehat{\theta}_j - g_j(\widehat{w})| > 4\left(\frac{C \operatorname{var}_\mu l'_j(\widehat{w}_{(t)};z)\log(2\delta^{-1})}{n}\right)^{1/2} \tag{25}$$

with probability no greater than δ. 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

$$\begin{aligned}
&\mathbf{P}\left\{\|\widehat{\theta}_{(t)} - g(\widehat{w}_{(t)})\| > \varepsilon^{*}\right\} \\
&\qquad = \mathbf{P}\left\{\sum_{j=1}^{d} \frac{(\widehat{\theta}_j - g_j(\widehat{w}_{(t)}))^2}{A^2} > \operatorname{trace}(\Sigma_{(t)})\right\} \\
&\qquad = \mathbf{P}\left\{\sum_{j=1}^{d} \left(\frac{(\widehat{\theta}_j - g_j(\widehat{w}_{(t)}))^2}{A^2} - \operatorname{var}_{\mu} l_j'(\widehat{w}_{(t)}; z)\right) > 0\right\} \\
&\qquad \leq \mathbf{P} \bigcup_{j=1}^{d} \left\{\frac{(\widehat{\theta}_j - g_j(\widehat{w}_{(t)}))^2}{A^2} > \operatorname{var}_{\mu} l_j'(\widehat{w}_{(t)}; z)\right\} \\
&\qquad \leq d\delta.
\end{aligned}$$

The first inequality uses a union bound, and the second inequality follows from (25). Plugging in A and taking confidence δ/d implies the desired result. □

Proof of Theorem 12 From Lemma 11, 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

$$\mathbf{P}\{\|\widehat{\theta}_{(t)} - g(\widehat{w}_{(t)})\| > \varepsilon\} \leq A_1 \exp\left(-\left(\frac{\varepsilon}{A_2}\right)^2\right).$$

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 inequality

$$\mathbf{E}\,|X|^p = \int_0^\infty \mathbf{P}\{|X|^p > t\}\,dt$$

as a starting point. Setting $X = \|\widehat{\theta}_{(t)} - g(\widehat{w}_{(t)})\| \geq 0$, 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. □

A.3 Guarantees in the case without strong convexity

Lemma 15 *(Comparing trajectories) Comparing* (2) *and* (3)*, assume that* $\widehat{g}$ *satisfies (7). Setting* $\alpha_{(t)} \in (0,1)$ *for all* $0 \le t < T$*, and initializing to* $\widehat{w}_{(0)} = w^*_{(0)}$*, with probability at least* $1 - T\delta$*, we have*

$$\|\widehat{w}_{(T)} - w^*_{(T)}\| \le \varepsilon \left(\prod_{t=0}^{T-1} (1 + \lambda\alpha_{(t)}) - 1 \right). \tag{26}$$

Proof of Lemma 15 For arbitrary step t, comparing the results of updates (2) and (3) with common step size $\alpha_{(t)}$, we have

$$\begin{aligned}\|\widehat{w}_{(t+1)} - w^*_{(t+1)}\| &\le \|\widehat{w}_{(t)} - w^*_{(t)}\| + |\alpha_{(t)}| \left(\|\widehat{g}(\widehat{w}_{(t)}) - g(\widehat{w}_{(t)})\| + \|g(\widehat{w}_{(t)}) - g(w^*_{(t)})\| \right) \\ &\le \|\widehat{w}_{(t)} - w^*_{(t)}\| \left(1 + \lambda\alpha_{(t)}\right) + \alpha_{(t)}\varepsilon. \end{aligned} \tag{27}$$

The latter inequality follows from the ε-accuracy and λ-smoothness in A2. Next, note that for any $t \ge 1$, if we have

$$\|\widehat{w}_{(t)} - w^*_{(t)}\| \le \frac{\varepsilon}{\lambda} \left(\prod_{k=0}^{t-1} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right),$$

then using (27), it follows that in the next iteration

$$\begin{aligned}\|\widehat{w}_{(t+1)} - w^*_{(t+1)}\| &\le \frac{\varepsilon}{\lambda} \left(\prod_{k=0}^{t-1} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right) \left(1 + \lambda\alpha_{(t)}\right) + \alpha_{(t)}\varepsilon \\ &= \frac{\varepsilon}{\lambda} \left(\prod_{k=0}^{t} \left(1 + \lambda\alpha_{(k)}\right) - 1 \right).\end{aligned}$$

Finally noting that we have the base case

$$\|\widehat{w}_{(1)} - w^*_{(1)}\| \le \alpha_{(0)}\varepsilon = \frac{\varepsilon}{\lambda} \left((1 + \lambda\alpha_{(0)}) - 1 \right),$$

taking the form assumed in the induction step. The desired result follows by mathematical induction. □

Without strong convexity, control of the risk becomes slightly more cumbersome, but weaker guarantees can be derived in a straightforward manner. Using A2 and (14):

$$\begin{aligned} R(\widehat{w}_{(T)}) - R^* &= R(\widehat{w}_{(T)}) - R(w^*_{(T)}) + R(w^*_{(T)}) - R^* \\ &\le \frac{\lambda}{2}\|\widehat{w}_{(T)} - w^*_{(T)}\|^2 + \langle g(w^*_{(T)}), \widehat{w}_{(T)} - w^*_{(T)} \rangle + R(w^*_{(T)}) - R^* \\ &\le \frac{\lambda}{2}\|\widehat{w}_{(T)} - w^*_{(T)}\|^2 + \|g(w^*_{(T)})\| \|\widehat{w}_{(T)} - w^*_{(T)}\| + R(w^*_{(T)}) - R^*. \end{aligned}$$

Furthermore, using $g(w^*) = 0$ and (15), we have

$$\begin{aligned} \|g(w^*_{(T)})\|^2 &= \|g(w^*_{(T)}) - g(w^*)\|^2 \\ &\le 2\lambda \left(R(w^*_{(T)}) - R(w^*) - \langle g(w^*), w^*_{(T)} - w^* \rangle \right) \\ &= 2\lambda \left(R(w^*_{(T)}) - R(w^*) \right). \end{aligned}$$

By convexity and A3 , we have $R^{*} = R(w^{*})$. Writing $A := \|\widehat{w}_{(T)} - w^{*}_{(T)}\|^{2}$ and $B := R(w^{*}_{(T)}) - R(w^{*})$, it follows that

$$R(\widehat{w}_{(T)}) - R^{*} \leq \frac{\lambda A}{2} + \sqrt{2\lambda A B} + B.$$

Control of the estimation error A can be done using a direct application of Lemmas 5 and 15, which naturally yield

$$\|\widehat{w}_{(T)} - w^{*}_{(T)}\| \leq \frac{\widetilde{\varepsilon}}{\sqrt{n}} \left((1+\lambda\alpha)^{T} - 1\right)$$

with probability at least $1-\delta$.

As for the optimization error B, this can be controlled using Theorem 2.1.14 of Nesterov (2004), 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 (12). Plugging these in as upper bounds on A and B in the risk control inequality gives the desired result.

A.4 Computational methods

Here we discuss precisely how to compute the implicitly-defined M-estimates of (4) and (6). Assuming $s > 0$ and real-valued observations $x_1, \ldots, x_n$, we first look at the program

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \rho_s\left(x_i - \theta\right)$$

assuming ρ is as specified in Definition 13, 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 Definition 13.

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 $c \approx 0.34$.

B Additional test results

In this section, we provide some additional experimental results obtained via the tests of Sect. 4. In particular, we consider the regression application at the end of Sect. 4.1, where due to space limitations, we only showed results for four distinct families of noise distributions. Here, we consider all of the following distribution families: Arcsine (asin), Beta Prime (bpri), Chi-squared (chisq), Exponential (exp), Exponential-Logarithmic (explog), Fisher's F (f), Fréchet (frec), Gamma (gamma), Gompertz (gomp), Gumbel (gum), Hyperbolic Secant (hsec), Laplace (lap), Log-Logistic (llog), Log-Normal (lnorm), Logistic (lgst), Maxwell (maxw), Pareto (pareto), Rayleigh (rayl), Semicircle (scir), Student's t (t), Triangle (asymmetric tri_a, symmetric tri_s), U-Power (upwr), Wald (wald), Weibull (weibull).

The content of this section is as follows:

- Figs. 10–11: performance as a function of sample size n.
- Figs. 12–13: performance over noise levels, with fixed n and d.
- Figs. 14–15: performance as a function of d, with fixed n/d ratio and noise level.

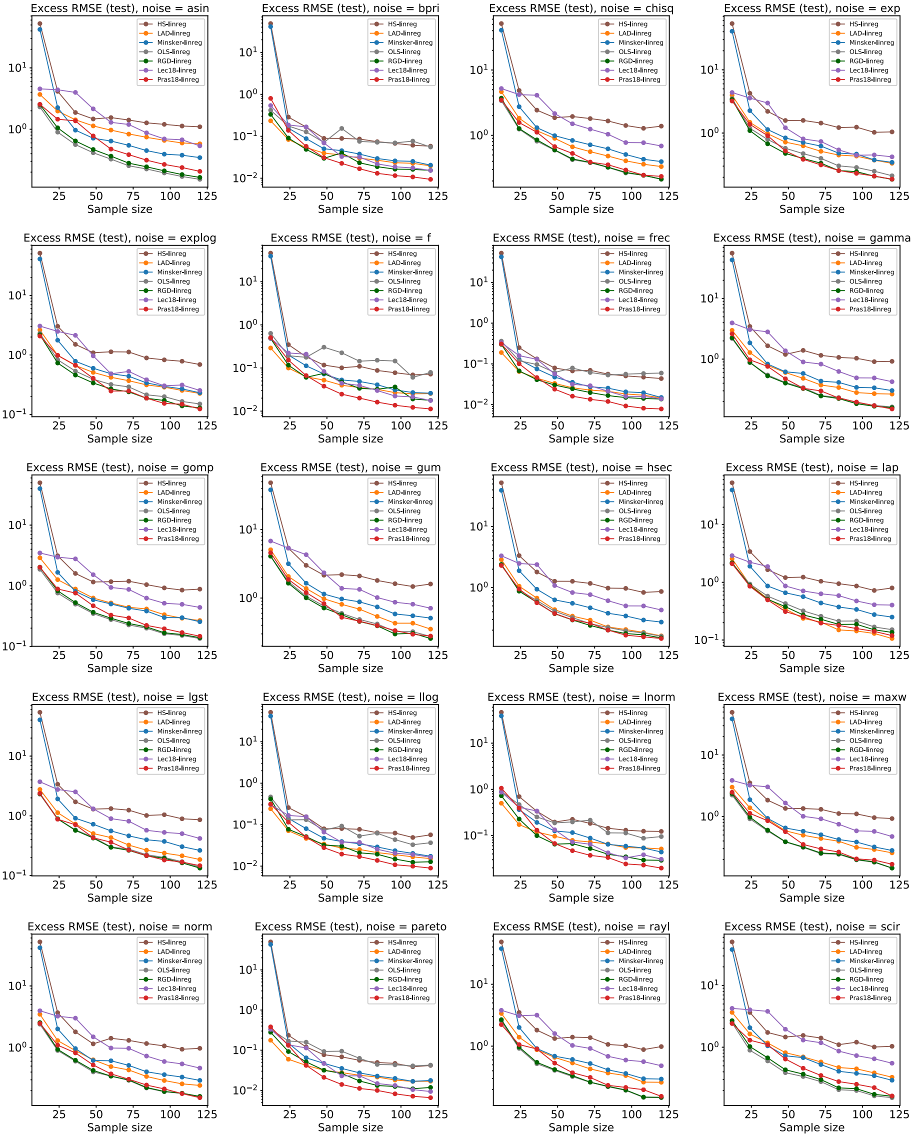

Fig. 10 Prediction error over sample size $12 \leq n \leq 122$, fixed $d = 5$, noise level = 8. Each plot corresponds to a distinct noise distribution (Color figure online)

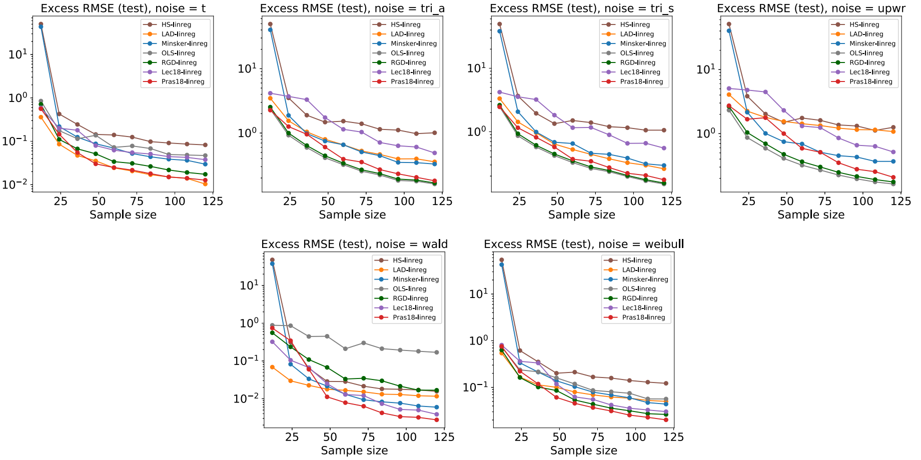

Fig. 11 Prediction error over sample size $12 \leq n \leq 122$, fixed $d = 5$, noise level = 8. Each plot corresponds to a distinct noise distribution (Color figure online)

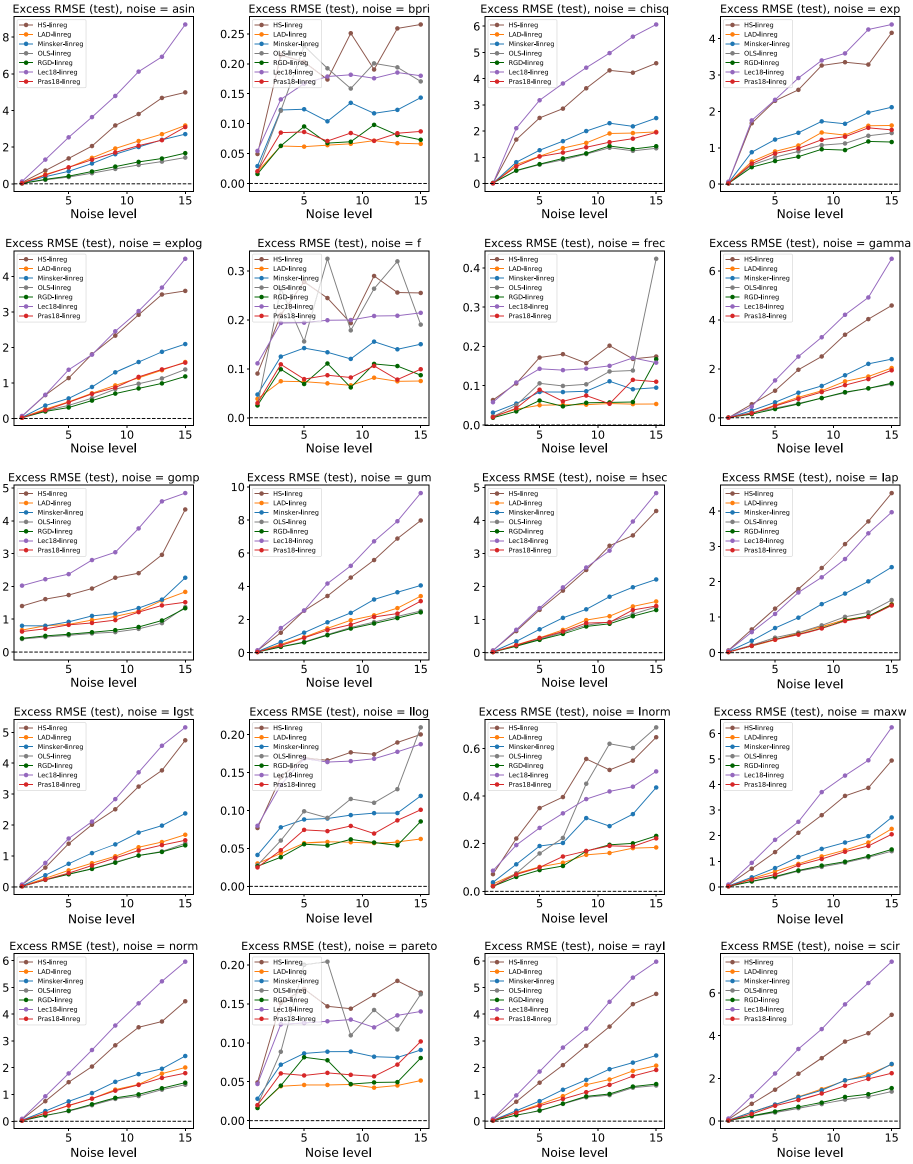

Fig. 12 Prediction error over noise levels, for $n = 30, d = 5$. Each plot corresponds to a distinct noise distribution (Color figure online)

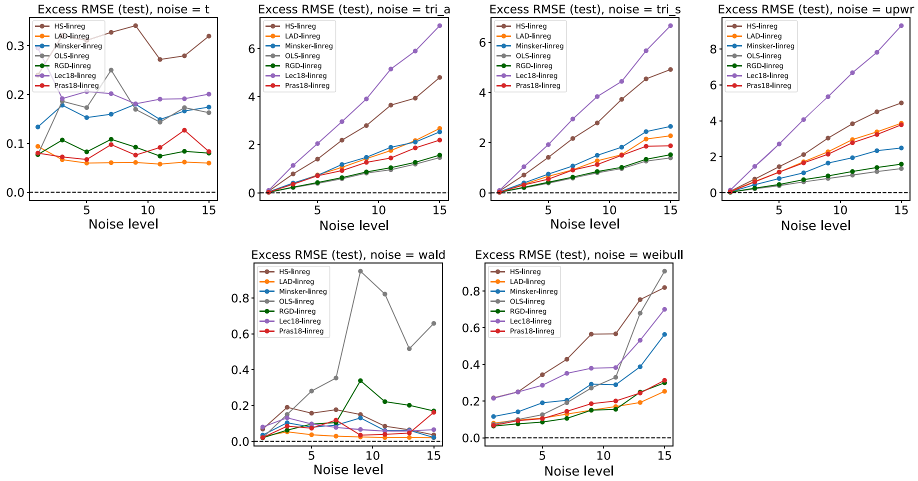

Fig. 13 Prediction error over noise levels, for $n = 30, d = 5$. Each plot corresponds to a distinct noise distribution (Color figure online)

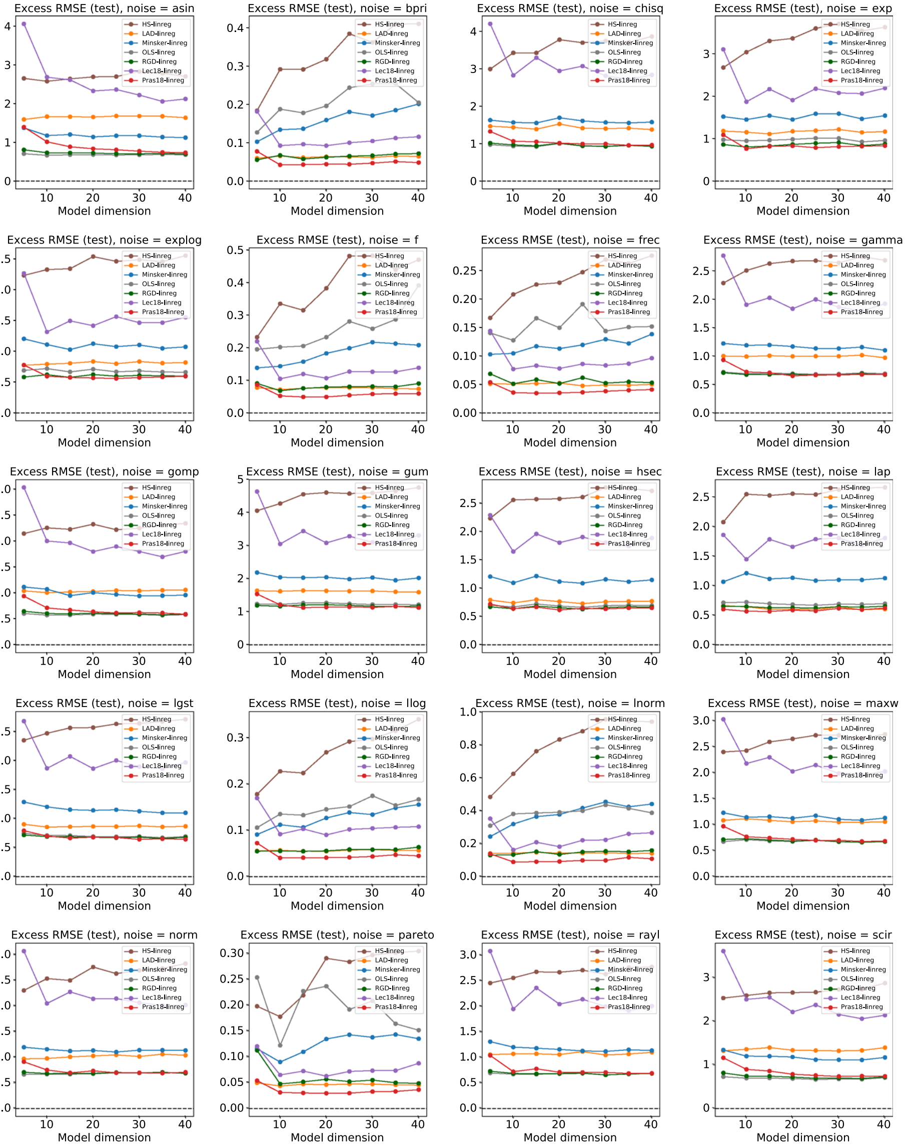

Fig. 14 Prediction error over dimensions $5 \leq d \leq 40$, with ratio $n/d = 6$ fixed, and noise level = 8. Each plot corresponds to a distinct noise distribution (Color figure online)

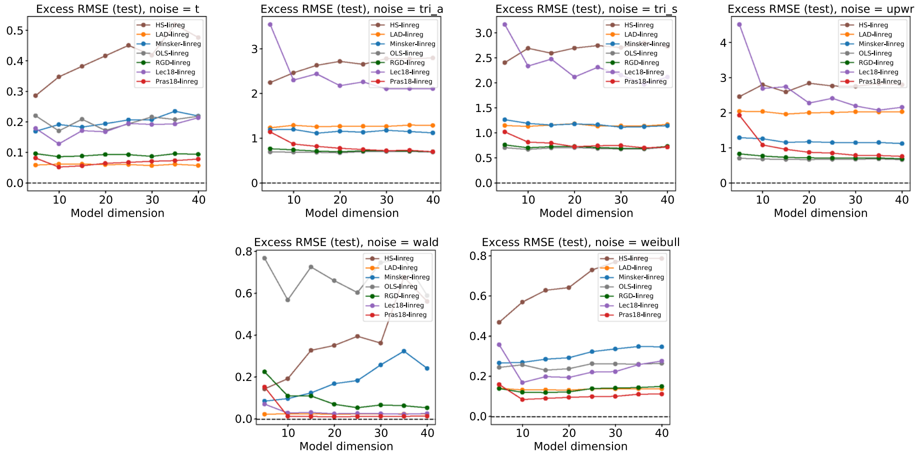

Fig. 15 Prediction error over dimensions $5 \leq d \leq 40$, with ratio $n/d = 6$ fixed, and noise level = 8. Each plot corresponds to a distinct noise distribution (Color figure online)

References

Abramowitz, M., & Stegun, I. A. (1964). Handbook of mathematical functions with formulas, graphs, and mathematical tables, National Bureau of Standards Applied Mathematics Series, vol 55. US National Bureau of Standards.

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

Ash, R. B., & Doleans-Dade, C. (2000). *Probability and measure theory*. Cambridge: Academic Press.

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

Bartlett, P. L., & Mendelson, S. (2003). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*, 463–482.

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

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

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.

Chen, Y., Su, L., & Xu, J. (2017a). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. arXiv preprint arXiv:1705.05491.

Chen, Y., Su, L., & Xu, J. (2017b). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *1*(2), 44.

Daniely, A., & Shalev-Shwartz, S. (2014). Optimal learners for multiclass problems. In *27th annual conference on learning theory, proceedings of machine learning research* (vol. 35, pp. 287–316).

Devroye, L., Lerasle, M., Lugosi, G., & Oliveira, R. I. (2015). Sub-Gaussian mean estimators. arXiv preprint arXiv:1509.05845.

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

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

Finkenstädt, B., & Rootzén, H. (Eds.). (2003). *Extreme values in finance, telecommunications, and the environment*. Boca Raton: CRC Press.

Frostig, R., Ge, R., Kakade, S. M., & Sidford, A. (2015). Competing with the empirical risk minimizer in a single pass. arXiv preprint arXiv:1412.6606.

Holland, M. J., & Ikeda, K. (2017a). Efficient learning with robust gradient descent. arXiv preprint arXiv:1706.00182.

Holland, M. J., & Ikeda, K. (2017b). Robust regression using biased objectives. *Machine Learning*, *106*(9), 1643–1679. https://doi.org/10.1007/s10994-017-5653-5.

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

Huber, P. J., & Ronchetti, E. M. (2009). *Robust statistics* (2nd ed.). New York: Wiley.

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

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

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kolmogorov, A. N. (1993). ε-entropy and ε-capacity of sets in functional spaces. In A. N. Shiryayev (Ed.), *Selected works of A. N. Kolmogorov, volume III: Information theory and the theory of algorithms* (pp. 86–170). Berlin: Springer.

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

Lecué, G., & Lerasle, M.(2017). Learning from MOM's principles. arXiv preprint arXiv:1701.01961.

Lecué, G., Lerasle, M., & Mathieu, T. (2018). Robust classification via MOM minimization. arXiv preprint arXiv:1808.03106.

Lerasle, M., & Oliveira, R. I. (2011). Robust empirical mean estimators. arXiv preprint arXiv:1112.3914.

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

Luenberger, D. G. (1969). *Optimization by vector space methods*. New York: Wiley.

Lugosi, G., & Mendelson, S. (2016). Risk minimization by median-of-means tournaments. arXiv preprint arXiv:1608.00757.

Minsker, S., & Strawn, N. (2017). Distributed statistical estimation and rates of convergence in normal approximation. arXiv preprint arXiv:1704.02658.

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

Murata, T., & Suzuki, T. (2016). Stochastic dual averaging methods using variance reduction techniques for regularized empirical risk minimization problems. arXiv preprint arXiv:1603.02412.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Berlin: Springer.

Nocedal, J., & Wright, S. (1999). *Numerical optimization.*, Springer Series in Operations Research Berlin: Springer.

Prasad, A., Suggala, A. S., Balakrishnan, S., & Ravikumar, P. (2018). Robust estimation via robust gradient estimation. arXiv preprint arXiv:1802.06485.

Rakhlin, A., Shamir, O., & Sridharan, K. (2012). Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th international conference on machine learning* (pp. 449–456).

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

Talvila, E. (2001). Necessary and sufficient conditions for differentiating under the integral sign. *American Mathematical Monthly*, *108*(6), 544–548.

van der Vaart, A. W. (1998). *Asymptotic statistics*. Cambridge: Cambridge University Press.

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.