Machine Learning (2023) 112:2933–2951
https://doi.org/10.1007/s10994-021-06036-0

Diametrical Risk Minimization: theory and computations

Matthew D. Norton[1] · **Johannes O. Royset**[2]

Received: 21 July 2020 / Revised: 1 July 2021 / Accepted: 7 July 2021 /
Published online: 2 September 2021

Abstract
The theoretical and empirical performance of Empirical Risk Minimization (ERM) often suffers when loss functions are poorly behaved with large Lipschitz moduli and spurious sharp minimizers. We propose and analyze a counterpart to ERM called Diametrical Risk Minimization (DRM), which accounts for worst-case empirical risks within neighborhoods in parameter space. DRM has generalization bounds that are independent of Lipschitz moduli for convex as well as nonconvex problems and it can be implemented using a practical algorithm based on stochastic gradient descent. Numerical results illustrate the ability of DRM to find quality solutions with low generalization error in sharp empirical risk landscapes from benchmark neural network classification problems with corrupted labels.

Keywords Empirical risk minimization · Generalization error · Solution stability

1 Introduction

In stochastic optimization, the minimum value of the empirical risk exhibits a downward bias and the corresponding minimizers are therefore often poor in terms of their true (population) risk. Lipschitz continuity[1] is often brought in as a critical component in attempts to assess the quality of such minimizers, with the Lipschitz moduli of loss functions relative to model parameters (weights) entering in generalization bounds and other results for Empirical Risk Minimization (ERM) problems; see for example Bartlett et al. (2017), Bousquet and Elisseeff (2002), Charles and Papailiopoulos (2018),

[1] We recall that a function f is *Lipschitz continuous* on a set C if there is a positive constant κ, called the *Lipschitz modulus*, such that $|f(w) - f(w')| \leq \kappa \|w - w'\|$ for all $w, w' \in C$.

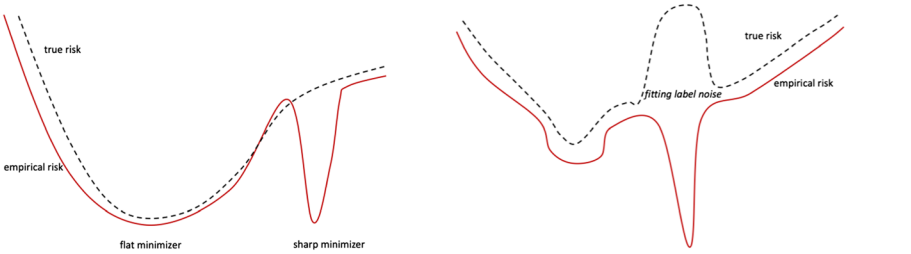
Editors: Daniel Fremont, Mykel Kochenderfer, Alessio Lomuscio, Dragos Margineantu, Cheng Soon-Ong.

✉ Johannes O. Royset
joroyset@nps.edu

Matthew D. Norton
mdnorto@gmail.com

[1] Target Corporation, Minneapolis, USA

[2] Naval Postgraduate School, Monterey, USA



(a) Sharp vs flat minimizers. **(b)** Sharp minimizer from label noise in training set.

Fig. 1 The empirical risk forms the empirical risk landscape and can be quite different than the true risk

Hardt et al. (2016) and Shalev-Shwartz et al. (2010). In this work, we propose a counterpart to ERM called Diametrical Risk Minimization (DRM) that possesses a generalization bound which is independent of Lipschitz moduli for convex as well as nonconvex loss functions. Preliminary simulations on benchmark Neural Network (NN) classification problems with MNIST and CIFAR-10 datasets support the hypothesis that when problems have large Lipschitz moduli, DRM is able to locate quality solutions with low generalization error, while ERM comparatively struggles.

The empirical risk as a function of model parameters in a learning problem has a graph, which we refer to as the *empirical risk landscape*; see the solid red line in Fig. 1a. The process of training the model then amounts to determining parameters that correspond to the bottom of a "valley" in this landscape. A large Lipschitz modulus tends to produce a *sharp* empirical risk landscape, where the empirical risk is highly variable with sudden "drops" of the kind labelled as a *sharp minimizer* in Fig. 1a. If the Lipschitz modulus is low, at least locally, then the empirical risk landscape is *flat* as to the left in Fig. 1a and the resulting minimizer is *flat*.

Instead of the empirical risk, DRM considers the *diametrical risk* at a point in the parameter space, which is given by the worst-case empirical risk in a neighborhood of the point. This provides DRM with a broader view of the empirical risk landscape than ERM and results in improved performance when the landscape is sharp.

Dealing with empirical risk landscapes that have large Lipschitz moduli and sharp minimizers is a challenge that has seen renewed attention in recent years under the heading of sharp vs flat minimizers in landscapes generated by NN. It is hypothesized that the landscape of NN problems are chaotic (Li et al., 2018; Nguyen & Hein, 2017) and that flat minimizers have better generalization properties compared to sharp ones (Chaudhari et al., 2017; Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Sagun et al., 2016). The potential effects can be seen in Fig. 1a, which also depicts the true risk in a learning problem (black dashed line). The spurious dip (right-most valley) of the empirical risk landscape is caused by a large Lipschitz modulus. Since ERM seeks out such dips, the resulting minimizer is poor when assessed using the true risk. We would have preferred that ERM found the left-most valley where any of its obtained minimizers would have had a low true risk. Although the figure is conceptual, it is believed that landscapes of NNs may exhibit similar behavior (Keskar et al., 2016). Thus, it has been a goal of many researchers to either locate the flat minimizers of such problems or to construct loss functions and/or NN architectures which do not have a large number of sharp minimizers (Chaudhari et al., 2017; Gouk et al., 2018; Keskar et al., 2016; Sagun et al., 2016).

Figure 1b illustrates a different kind of sharpness, which can be induced by introducing label noise as explored in Zhang et al. (2016) and Oymak et al. (2019). When a portion of labels are randomly flipped, it has been shown that NNs are capable of fitting the training data perfectly, achieving a zero-training error solution. However, this solution clearly will not generalize and lies near a sharp minimizer which is associated with the fitting of the incorrectly labeled training data. Nevertheless, it has been shown that, even in the presence of label noise, there still exist flat minimizers such as the left-most one in Fig. 1b. It is immediately clear that DRM may perform well and achieve flat minimizers even in landscapes with spurious sharp minimizers due its broader view of the empirical risk landscape.

Lipschitz moduli frequently play a direct role in statistical learning theory. For example, a paradigm in learning theory is the analysis of algorithmic stability (Bousquet & Elisseeff, 2002; Charles & Papailiopoulos, 2018; Hardt et al., 2016; Shalev-Shwartz et al., 2010). However, a majority of these learning bounds require some notion of smoothness in terms of either a Lipschitz continuous objective function and/or Lipschitz continuous gradient. The Lipschitz moduli then enter the resulting generalization bounds and influence the (theoretical) stability of the algorithm used to perform ERM. The reliance of these and other generalization bounds (Bartlett et al., 2017) on the magnitude of the Lipschitz moduli, as well as the growing support for the sharp-vs-flat hypothesis, have even given rise to research centered on Lipschitz regularization (Gouk et al., 2018; Oberman & Calder, 2018; Qian & Wegman, 2018) for improving the generalization of NNs. We provide generalization bounds, however, that do not rely upon the Lipschitz moduli: DRM can be applied to sharp empirical risk landscapes with resulting solutions having low generalization error. Eliminating the dependence on the Lipschitz modulus does, however, come with a cost. The provided bounds rely somewhat unfavorably on the dimension of the parameter space. Still, experiments indicate that this is a limitation of the proof approach and not a fundamental limitation of DRM in general. We carry out experiments in a NN setting where the dimension of the parameter space is larger than the number of training samples. Even in this setting, we find that DRM solutions generalize favorably compared to solutions found via ERM.

Although not studied in detail, DRM may also support training in the context of quantification where parameters are represented with lower precision (Dong et al., 2019). There, the explicit robustification in DRM against parameter perturbations could emerge as beneficial.

The downward bias associated with ERM has been known since the early days of stochastic optimization and *M*-estimators. Traditional remedies include a variety of regularization schemes, focused on alteration of the objective function (Bertsimas & Copenhaver, 2018; Bousquet & Elisseeff, 2002; Liu et al., 2019) or the optimization procedure itself with, for example, early stopping (Hardt et al., 2016; Royset, 2013; Royset & Szechtman, 2013). Another remedy is to replace ERM by the problem of minimizing the distributionally worst-case empirical risk; see for example Bertsimas et al. (2018), Duchi et al. (2018), Royset and Wets (2017), Wiesemann et al. (2014), Zhang et al. (2016) and references therein. Typically, the worst-case is defined in terms of a ball in some metric on a space of probability measures centered on the empirical distribution generated by the available data. Adversarial training (Madry et al., 2018) is a closely related approach where the worst-case is computed by perturbing the data directly as for classical robust M-estimators in statistics; see also Carmon et al. (2019), Cohen et al. (2019), Liao et al. (2018) and Zhang et al. (2019) for other efforts to achieve NNs that are robust against data perturbations. DRM is distinct from these approaches by

perturbing the parameter vector instead of the distribution governing the data or the data itself.

Perturbation of a parameter vector as a means to account for "implementation error" of a decision or design specified by the vector is included as a motivation for Robust Optimization (Ben-Tal & Nemirovski, 1998); see Stinstra and den Hertog (2008) for application in the context of meta-models and Men et al. (2014) for fabrication problems. The latter reference as well as Lewis (2002) and Lewis and Pang (2010) lay theoretical and computational foundations for minimizing functions of the form $w \mapsto \sup_{v \in V} f(w+v)$ that include establishing Lipschitz continuity even if f is rather general. In particular, the minimization of such sup-functions can be achieved by semidefinite programming (Lewis, 2002; Luo et al., 2004) when f is convex and quadratic. Examples of "robustification" by considering a worst-case parameter vector are also found in signal processing (Luo, 2003; Pinar & Arikan, 2004). Concurrent to the present work, Wu et al. (2020) develops an approach based on perturbation of parameters *and* data with strong empirical performance. The theoretical results are limited to a PAC-Bayes bound where an assumption on the distribution of parameters allows one to conclude that the approach has a robust generalization bound that involves the expectation of the "flatness" of the empirical risk landscape. Though, details of the argument are omitted. A more detailed theoretical analysis is carried in Tsai et al. (2021) for the specific class of feed-forward neural networks with non-negative monotone activation functions against norm-bounded parameter perturbations. In contrast, we consider nearly arbitrary neural networks and in fact stochastic optimization more generally as well.

The remainder of this paper is organized as follows. Section 2 introduces DRM and illustrates the difficulties faced by ERM when loss functions are poorly behaved. In Sect. 3, we provide a theoretical analysis of DRM that includes generalization bounds independent of Lipschitz moduli. In Sect. 4, we propose a practical algorithm for performing DRM based on stochastic gradient descent (SGD). We then provide an experimental study in Sect. 5, with a focus on supporting the idea that DRM can find good solutions to problems with sharp empirical risk landscapes. Code for the experiments is available online.[2]

2 Diametrical Risk Minimization

For a loss function $\ell : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ and sample $S = \{z_1, \ldots, z_m\} \subset \mathbb{R}^d$, it is well known that the *ERM problem*

$$\underset{w \in \mathbb{R}^n}{\text{minimize}}\, R_m(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w, z_i)$$

can be a poor surrogate for the actual problem of minimizing the *true risk* $R(w) = \mathbb{E}_z[\ell(w, z)]$. Here, $R_m(w)$ is the *empirical risk* of parameter vector w. Specifically, a (near-)minimizer $w_m^\star$ of R_m tends to have true risk $R(w_m^\star)$ significantly higher than that stipulated by $R_m(w_m^\star)$; cf. the difference between the solid red line illustrating R_m and the dashed black line illustrating R in Fig. 1a. The effect worsens when the loss $\ell(w, z)$ varies dramatically under changing parameters w, which is the case when $\ell(\cdot, z)$ has a large Lipschitz modulus.

[2] https://github.com/matthew-norton/Diametrical_Learning.

In this section, we propose an alternative that we coin *Diametrical Risk Minimization (DRM)*. In contrast to common robustification strategies based on perturbing the data set, DRM perturbs the parameters and thereby obtains stability even for poorly behaved loss functions. Instead of minimizing R_m directly as in ERM, any learned parameter vector w is "diametrically" modified before the empirical risk is minimized.

Definition 2.1 For a loss function $\ell : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ and sample $S = \{z_1, \ldots, z_m\} \subset \mathbb{R}^d$, the *diametrical risk* of a parameter vector $w \in \mathbb{R}^n$ is given as

$$R_m^\gamma(w) = \sup_{\|v\| \le \gamma} R_m(w + v) = \sup_{\|v\| \le \gamma} \frac{1}{m} \sum_{i=1}^{m} \ell(w + v, z_i),$$

where $\gamma \in [0, \infty)$ is the *diametrical risk radius*.

We see that the diametrical risk of parameter vector $w \in \mathbb{R}^n$ is the worst possible empirical risk in a neighborhood of w. Any norm $\|\cdot\|$ can be used to define the neighborhood. Trivially, $R_m^0(w) = R_m(w)$, but generally $R_m^\gamma(w) \ge R_m(w)$.

For some set $W \subset \mathbb{R}^n$ of permissible parameter vectors, the *DRM problem* amounts to

$$\underset{w \in W}{\text{minimize}}\, R_m^\gamma(w)$$

and results in a solution w_m^γ, which might be a global minimizer, a local minimizer, a stationary point, or some other parameter vector with "low" diametrical risk.

As we show in Theorem 3.3, under mild assumptions,

$$R(w_m^\gamma) - R_m^\gamma(w_m^\gamma) \le \beta m^{-1/2} \tag{1}$$

with high probability for some constant β regardless of the exact nature of w_m^γ. In particular, w_m^γ generalizes even if obtained after aggressive minimization of the diametrical risk; DRM is inherently resistant to overfitting.

Two examples illustrate the challenge faced by ERM when loss functions have large Lipschitz moduli (with respect to parameters w). In both examples, we will see that the generalization error for DRM is dramatically smaller than for ERM. For $\kappa \in (1, \infty)$ and $\gamma \in (0, 1)$, let

$$\ell(w, z) = \begin{cases} \kappa w/\gamma + \kappa & \text{if } w \in [-\gamma, 0), z = 0 \\ -\kappa w/\gamma - \kappa & \text{if } w \in [-\gamma, 0), z = 1 \\ -\kappa w/\gamma + \kappa & \text{if } w \in [0, \gamma), z = 0 \\ \kappa w/\gamma - \kappa & \text{if } w \in [0, \gamma), z = 1 \\ 0 & \text{otherwise.} \end{cases}$$

If z takes the values 0 and 1, each with probability $\frac{1}{2}$, then $R(w) = \mathbb{E}_z[\ell(w, z)] = 0$ for all $w \in \mathbb{R}$. In contrast,

$$R_m(w) = \begin{cases} \frac{1}{m}\rho_m(\kappa w/\gamma + \kappa) & \text{if } w \in [-\gamma, 0) \\ \frac{1}{m}\rho_m(-\kappa w/\gamma + \kappa) & \text{if } w \in [0, \gamma) \\ 0 & \text{otherwise,} \end{cases}$$

where ρ_m is the number of zeros minus the number of ones in the data $\{z_1, \ldots, z_m\}$. Viewing the data as random, we obtain that with probability nearly $\frac{1}{2}$, $\rho_m < 0$ and thus $w_m^\star = 0$ minimizes R_m for such outcomes of the data and $R_m(w_m^\star) = \rho_m \kappa / m$. Also with probability near $\frac{1}{2}$, $\rho_m \geq 0$ and then $w_m^\star = 1$ minimizes R_m and $R_m(w_m^\star) = R(w_m^\star) = 0$. Consequently, $R_m(w_m^\star)$ has a downward bias. Although

$$R(w_m^\star) - R_m(w_m^\star) \leq \max\{0, -\rho_m\}\kappa / m \tag{2}$$

with probability one, the right-hand side includes the constant κ, which is proportional to the Lipschitz modulus κ/γ of $\ell(\cdot, z)$. This illustrates the well-known fact that generalization tends to be poor for loss functions with large Lipschitz moduli. However, considering diametrical risk, we have that $R(w_m^\gamma) - R_m^\gamma(w_m^\gamma) \leq 0$ with probability one.

The situation deteriorates further when the loss function is not Lipschitz continuous. Let

$$\ell(w, z) = \begin{cases} 1/w & \text{if } w \in (0, \infty), z = 0 \\ -1/w & \text{if } w \in (0, \infty), z = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Again, with z and ρ_m as above, $R(w) = \mathbb{E}_z[\ell(w, z)] = 0$ for all $w \in \mathbb{R}$ and

$$R_m(w) = \begin{cases} \frac{1}{m}\rho_m / w & \text{if } w \in (0, \infty) \\ 0 & \text{otherwise.} \end{cases}$$

Then, $\inf_{w \in \mathbb{R}} R_m(w) = -\infty$ when $\rho_m < 0$, which takes place with probability nearly $\frac{1}{2}$. The downward bias is now unbounded. Considering diametrical risk, we have the much more favorable bound $R(w_m^\gamma) - R_m^\gamma(w_m^\gamma) \leq \beta \gamma^{-1} m^{-1/2}$ with high probability for some constant β.

From these simple examples, it is clear that ERM can lead to arbitrarily slow learning when the loss function is poorly behaved. In the first example above, DRM has a generalization error equal to zero and thus is independent of the Lipschitz modulus. In the second example, DRM reduces the unbounded generalization error encountered by ERM to a quantity proportional to $m^{-1/2}$ as we also see in the following section.

3 Rates of convergence

We begin by formalizing the setting and recall that $f : \mathbb{R}^n \times \Omega \to \mathbb{R}$ is a *Caratheodory function* relative to a probability space $(\Omega, \mathcal{A}, P)$ if for all $\omega \in \Omega$, $f(\cdot, \omega)$ is continuous and for all $w \in \mathbb{R}^n$, $f(w, \cdot)$ is $\mathcal{A}$-measurable. In the following, we assume that the data comprises d-dimensional random vectors generated by independent sampling according to the distribution $\mathbb{P}$ and thus consider the m-fold product probability space $(Z^m, \mathcal{Z}^m, \mathbb{P}^m)$ constructed from a probability space $(Z, \mathcal{Z}, \mathbb{P})$, with $Z \subset \mathbb{R}^d$. If $\ell : \mathbb{R}^n \times Z \to \mathbb{R}$ is a Caratheodory function relative to $(Z, \mathcal{Z}, \mathbb{P})$, then R_m, now viewed as a function on $\mathbb{R}^n \times Z^m$, is a Caratheodory function relative to $(Z^m, \mathcal{Z}^m, \mathbb{P}^m)$; see for example Rockafellar and Wets (1998, Prop. 14.44; Ex. 14.29). Likewise, we view R_m^γ as a function on $\mathbb{R}^n \times Z^m$. It is real-valued by virtue of being the maximum value of the continuous R_m over a compact set.[3] For all $w \in \mathbb{R}^n$, $R_m^\gamma(w)$ is $\mathcal{Z}^m$-measurable when R_m is a Caratheodory

[3] A set $C \subset \mathbb{R}^n$ is compact if it is closed and bounded.

function (Rockafellar & Wets, 1998, Thm. 14.37). Since R_m^γ is continuous (in w) for all $(z_1, \ldots, z_m) \in \mathcal{Z}^m$ by Rockafellar and Wets (1998, Thm. 1.17), we conclude that R_m^γ is a Caratheodory function relative to $(Z^m, \mathcal{Z}^m, \mathbb{P}^m)$. In view of Rockafellar and Wets (1998, Thm. 14.37; Ex. 14.32), $\sup_{w\in W} R_m^\gamma(w)$ is $\mathcal{Z}^m$-measurable as long as $W \subset \mathbb{R}^n$ is closed. In effect, any concern about measurability in the below statements are put to rest if ℓ is a Caratheodory function and W is closed.

We denote by $\mathbb{E}$ the expectation with respect to $\mathbb{P}$ so that for $w \in \mathbb{R}^n$, $R(w) = \mathbb{E}[\ell(w, z)] = \int \ell(w, z) d\mathbb{P}(z)$. When $\ell : \mathbb{R}^n \times Z$ is a Caratheodory function, we say it is *locally sup-integrable* if for all $\bar{w} \in \mathbb{R}^n$, there exists $\rho > 0$ such that $\int \max\{0, \sup\{\ell(w, z) \mid \|w - \bar{w}\| \le \rho\}\} d\mathbb{P}(z) < \infty$. It is *locally inf-integrable* if "max-sup" is replaced by "min-inf" in the above statement. The *moment-generating function* of a random variable X is $\tau \mapsto E[\exp(\tau X)]$.

We start with a preliminary fact, which follows from Fatou's Lemma.

Proposition 3.1 *If $\ell : \mathbb{R}^n \times Z \to \mathbb{R}$ is a locally inf-integrable Caratheodory function, then R is lower semicontinuous. If locally inf-integrable is replaced by locally sup-integrable, then R is upper semicontinuous.*

Proof Suppose that $w^\nu \to \bar{w}$. Since ℓ is a inf-integrable Caratheodory function, Fatou's Lemma establishes that $\operatorname{liminf}\mathbb{E}[\ell(w^\nu, z)] \ge \mathbb{E}[\operatorname{liminf}\ell(w^\nu, z)] = \mathbb{E}[\ell(\bar{w}, z)]$. Thus, R is lower semicontinuous. A similar argument confirms the claim about upper semicontinuity; see Royset and Wets (2021, Propositions 8.54, 8.55) for details. □

The first main result bounds the amount the diametrical risk can fall below the true risk.

Theorem 3.2 (generalization error in DRM). *Suppose that $W \subset \mathbb{R}^n$ is compact, $\ell : \mathbb{R}^n \times Z \to \mathbb{R}$ is a locally sup-integrable Caratheodory function, and for all $w \in W$, the moment generating function of $\ell(w, \cdot) - R(w)$ is real-valued in a neighborhood of zero. Then, for any $\varepsilon, \gamma > 0$ and m, there exist $\eta, \beta > 0$ (independent of m) such that*

$$\mathbb{P}^m\Big(\sup_{w\in W}\big\{R(w) - R_m^\gamma(w)\big\} \le \varepsilon\Big) \ge 1 - \eta e^{-\beta m}.$$

Proof By Proposition 3.1, R is upper semicontinuous. Let $\{W_k \subset \mathbb{R}^n,\ k = 1, \ldots, N\}$ be a finite cover of W consisting of closed balls with radius $\gamma/2$. Without loss of generality, suppose that $W_k \cap W \neq \emptyset$ for all $k = 1, \ldots, N$. Let $w^k \in \operatorname{argmax}_{w\in W_k\cap W} R(w)$, which exists for all $k = 1, \ldots, N$ because $W_k \cap W$ is nonempty and compact, and R is upper semicontinuous.

For $k = 1, \ldots, N$, let $\tau \mapsto M_k(\tau)$ be the moment generating function of $R(w^k) - \ell(w^k, \cdot)$ and $I_k(\varepsilon) = \sup_{\tau\in\mathbb{R}}\{\varepsilon\tau - \log M_k(\tau)\}$, which is positive since M_k is real-valued in a neighborhood of zero. Then, by the upper bound in Cramer's Large Deviation Theorem (see for example Shapiro et al., 2009, Sec. 7.2.8)

$$\mathbb{P}^m\big(R(w^k) - R_m(w^k) \ge \varepsilon\big) \le e^{-mI_k(\varepsilon)}.$$

Moreover, with $\beta = \min_{k=1,\ldots,N} I_k(\varepsilon)$,

$$\mathbb{P}^m\Big(\max_{k=1,\ldots,N}\big\{R(w^k) - R_m(w^k)\big\} \ge \varepsilon\Big) \le \sum_{k=1}^N e^{-mI_k(\varepsilon)} \le Ne^{-\beta m}.$$

Consider an event for which $\max_{k=1,\dots,N}\{R(w^k) - R_m(w^k)\} \le \varepsilon$, which takes place with probability at least $1 - Ne^{-\beta m}$, and let $\bar{w} \in W$. There exists $\bar{k} \in \{1, \dots, N\}$ such that $\bar{w} \in W_{\bar{k}}$. Since $R_m^\gamma(\bar{w}) = \sup_{\|v\|\le\gamma} R_m(\bar{w} + v) \ge R_m(w^{\bar{k}})$,

$$R(\bar{w}) - R_m^\gamma(\bar{w}) \le R(\bar{w}) - R_m(w^{\bar{k}}) \le R(\bar{w}) - R(w^{\bar{k}}) + \varepsilon \le R(w^{\bar{k}}) - R(w^{\bar{k}}) + \varepsilon = \varepsilon,$$

where the last inequality follows by the fact that $R(w^{\bar{k}}) = \sup_{w\in W_{\bar{k}}} R(w)$. The conclusion then follows with $\eta = N$ because $\bar{w}$ is arbitrary. □

The theorem furnishes a uniform bound on R, which implies in particular that

$$R(w_m^\gamma) \le R_m^\gamma(w_m^\gamma) + \varepsilon \text{ with high probability}$$

for any parameter vector w_m^γ produced by DRM. Thus, there is a strong justification for minimizing R_m^γ: lower values of the diametrical risk ensure better guarantees on the true risk. The goal now becomes to develop good methods for producing w_m^γ with low $R_m^\gamma(w_m^\gamma)$. The issue of overfitting is largely removed: it is unlikely that a parameter vector w_m^γ with a low diametrical risk, i.e., low $R_m^\gamma(w_m^\gamma)$, has a high true risk $R(w_m^\gamma)$.

The assumptions in the theorem are generally mild: $\ell(\cdot, z)$ only needs to be continuous and the condition on the moment generating function is just checked pointwise. The requirement about locally sup-integrable amounts to determine an integrable random variable at every $\bar{w}$ that upper bounds ℓ in a neighborhood of $\bar{w}$. The constant β depends on ε, while η is a function of γ and the diameter of W, i.e., $\sup_{w,w'\in W} \|w - w'\|$, in the norm underpinning the diametrical risk. In particular, β and η are independent of Lipschitz moduli of ℓ, which may not even be finite.

If the value of the moment generating function of $\ell(w, \cdot) - R(w)$ can be quantified near zero, then we can examine the effect as the error ε vanishes as seen next. We recall that a random variable X is *subgaussian* with variance proxy σ^2 if its mean $E[X] = 0$ and its moment generating function satisfies $E[\exp(\tau X)] \le \exp(\frac{1}{2}\sigma^2\tau^2)$ for all $\tau \in \mathbb{R}$.

Theorem 3.3 (rate of convergence in DRM). *Suppose that $W \subset \mathbb{R}^n$ is compact, $\ell : \mathbb{R}^n \times Z \to \mathbb{R}$ is a locally sup-integrable Caratheodory function, and for all $w \in W$, $\ell(w, \cdot) - R(w)$ is subgaussian. Then, for any $\alpha \in (0, 1)$, $\gamma > 0$, and m, there exists $\beta > 0$ (independent of m) such that*

$$\mathbb{P}^m\Big(\sup_{w\in W}\big\{R(w) - R_m^\gamma(w)\big\} \le \beta m^{-1/2}\Big) \ge 1 - \alpha.$$

Proof By Proposition 3.1, R is upper semicontinuous. Let w^k be as in the proof of Theorem 3.2. There exists $\xi \in (0, \infty)$, which may depend on n, such that the number of closed balls N of radius $\gamma/2$ required to cover W is no greater than $(\xi/\gamma)^n$. Since $\ell(w^k, \cdot) - R(w^k)$ is subgaussian, say with variance proxy σ_k^2, we have by Bernstein's inequality that

$$\mathbb{P}^m\big(R(w^k) - R_m(w^k) > \varepsilon\big) \le \exp\big(-\frac{1}{2}m\varepsilon^2/\sigma_k^2\big) \quad \text{for all } \varepsilon \in [0, \infty).$$

Let $\sigma = \max_{k=1,\dots,N} \sigma_k$. Thus,

$$\mathbb{P}^m\Big(\max_{k=1,\dots,N}\big\{R(w^k) - R_m(w^k)\big\} > \varepsilon\Big) \le N\exp\big(-\frac{1}{2}m\varepsilon^2/\sigma^2\big) \le \alpha$$

provided that $\varepsilon \ge \beta m^{-1/2}$ and

$$\beta = \sigma\sqrt{2n\log(\xi/\gamma) - 2\log\alpha}.$$

Consider the event for which $\max_{k=1,\dots,N}\{R(w^k) - R_m(w^k)\} \le \varepsilon$, which takes place with probability at least $1-\alpha$ for such ε. The arguments in the proof of Theorem 3.2 establishes that $\sup_{w\in W}\{R(w) - R_m^\gamma(w)\} \le \varepsilon$ for this event and the conclusion follows. □

The constant β in the theorem is given in the proof and depends on the largest variance proxy, denoted by σ^2, for $\ell(w,\cdot) - R(w)$ at a finite number of different w. It also depends on a parameter ξ given by the diameter of W. For example, if R_m^γ is defined in terms of the sup-norm, then the balls $W_1, \dots, W_N$ in the proof can be constructed according to that norm and the number required is simply[4] $N = \lceil\delta/\gamma\rceil^n$, where $\delta = \sup_{w,\bar w\in W}\|w - \bar w\|_\infty$. This leads to

$$\beta = \sigma\sqrt{2n\log\lceil\delta/\gamma\rceil - 2\log\alpha}.$$

The constant β in the theorem depends unfavorably on n. One can attempt to reduce the effect of n by enlarging γ as n increases. For example, under the sup-norm one may set $\gamma = \delta[\exp(\zeta n^{-\nu})]^{-1}$ for positive constants ζ and ν. Then, assuming that δ/γ is an integer (which can be achieved by enlarging δ when needed),

$$\beta = \sigma\sqrt{2n\log\lceil\delta/\gamma\rceil - 2\log\alpha} = \sigma\sqrt{2\zeta n^{1-\nu} - 2\log\alpha}.$$

For example, if $\nu = 1$, then β becomes independent of n at the expense of having to grow γ rather quickly as n increase. A compromise could be $\nu = \frac{1}{2}$, in which case β grows only as $n^{1/4}$ and γ grows somewhat slower too. Still, in the limit as $n\to\infty$, R_m^γ involves maximization over all of W, which of course leads to an upper bound.

For fixed n, we may also ask what is the right value of γ? Since a large value might imply additional computational burden and also lead to overly conservative upper bounds, it would be ideal to select it as small as possible without ruining the rate (significantly). One possibility could be to set γ proportional to m^{-1} because then the rate deteriorate only with a logarithmic factor from $m^{-1/2}$ to $\sqrt{m^{-1}\log m}$.

It is clear from the proof of the theorem that the assumptions about independent sampling and subgaussian random variables can be relaxed. We only needed that the error in $R_m(w)$ relative to $R(w)$ can be bounded for a finite number of w; see Boucheron et al. (2016) and Oliveira and Thompson (2017) for possible extensions.

It may be of interest to determine the error of an obtained parameter vector w_m^γ relative to the set of actually optimal parameters $\operatorname{argmin}_{w\in W} R(w)$. Theorem 3.3 yields immediately that for any $\delta\in\mathbb{R}$,

$$\left\{w\in W \mid R_m^\gamma(w) \le \delta\right\} \subset \left\{w\in W \mid R(w) \le \delta + \beta m^{-1/2}\right\} \text{ with probability at least } 1-\alpha.$$

We now examine the harder question of confidence regions for "good" parameter vectors. For two sets $A, B\subset\mathbb{R}^n$, we denote the *excess* of A over B by

$$\operatorname{exs}(A;B) = \sup_{w\in A}\inf_{\bar w\in B}\|w-\bar w\| \text{ for nonempty } A, B;$$

with the convention that $\operatorname{exs}(A;B) = \infty$ if $A\neq\emptyset$ and $B=\emptyset$; $\operatorname{exs}(A;B) = 0$ otherwise. Below, the sets of interest are lower level-sets of R_m, possible with a random level. Arguing

[4] We note by $\lceil c\rceil$ be lowest integer at least as high as the scalar c.

by means of Prop. 14.33, Thm. 14.37, and Ex. 14.32 in Rockafellar and Wets (1998), we see that the excess involving such sets is measurable.

Theorem 3.4 (confidence region in DRM). *Suppose that $W \subset \mathbb{R}^n$ is compact, $\ell : \mathbb{R}^n \times Z \to \mathbb{R}$ is a locally inf-integrable Caratheodory function, and for all $w \in W$, $\ell(w,\cdot) - R(w)$ is subgaussian. Then, for any $\alpha \in (0,1)$, $\gamma > 0$, $\delta \in \mathbb{R}$, and m, there exists $\beta > 0$ (independent of m) such that*

$$\mathbb{P}^m\Big(\operatorname{exs}\Big(\big\{w \in W \;\big|\; R(w) \le \delta\big\}; \big\{w \in W \;\big|\; R_m(w) \le \delta + \beta m^{-1/2}\big\}\Big) \le \gamma,$$
$$\operatorname{exs}\Big(\operatorname{argmin}_{w\in W} R(w); \big\{w \in W \;\big|\; R_m(w) \le \inf_{\bar{w}\in W} R_m^{\gamma}(\bar{w}) + 2\beta m^{-1/2}\big\}\Big) \le \gamma\Big) \ge 1-\alpha.$$

Proof By Proposition 3.1, R is lower semicontinuous. From the compactness of $W \subset \mathbb{R}^n$, we obtain $\xi \in (0,\infty)$, which may depend on n, such that the number of closed balls N with radius $\gamma/2$ required to cover W is no greater than $(\xi/\gamma)^n$. Suppose that $\{W_k \subset \mathbb{R}^n,\ k = 1,\dots,N\}$ is a collection of such balls with $W_k \cap W \neq \emptyset$ for all $k = 1,\dots,N$. Let $w^k \in \operatorname{argmin}_{w\in W_k\cap W} R(w)$, which exists for all $k = 1,\dots,N$ because $W_k \cap W$ is nonempty and compact, and R is lower semicontinuous.

Since $\ell(w^k,\cdot) - R(w^k)$ is subgaussian, say with variance proxy σ_k^2, we have by Bernstein's inequality that

$$\mathbb{P}^m\Big(\big|R_m(w^k) - R(w^k)\big| > \varepsilon\Big) \le 2\exp(-\frac{1}{2}m\varepsilon^2/\sigma_k^2) \quad \text{for all } \varepsilon \in [0,\infty).$$

Let $\sigma = \max_{k=1,\dots,N} \sigma_k$. Thus,

$$\mathbb{P}^m\Big(\max_{k=1,\dots,N} \big|R_m(w^k) - R(w^k)\big| > \varepsilon\Big) \le 2N\exp(-\frac{1}{2}m\varepsilon^2/\sigma^2) \le \alpha$$

provided that $\varepsilon \ge \beta m^{-1/2}$ and

$$\beta = \sigma\sqrt{2n\log(\xi/\gamma) - 2\log(\alpha/2)}.$$

Consider the event for which $\max_{k=1,\dots,N} |R_m(w^k) - R(w^k)| \le \varepsilon$. Let $\bar{w} \in W$ satisfy $R(\bar{w}) \le \delta$. Then, there exists $\bar{k}$ such that $\bar{w} \in W_{\bar{k}}$ and because $R(w_{\bar{k}}) = \inf_{w\in W_{\bar{k}}\cap W} R(w)$ we obtain that

$$R_m(w_{\bar{k}}) \le R(w_{\bar{k}}) + \varepsilon \le R(\bar{w}) + \varepsilon \le \delta + \varepsilon.$$

Since $\|\bar{w} - w_{\bar{k}}\| \le \gamma$, we conclude that

$$\operatorname{exs}\big(\{w \in W \mid R(w) \le \delta\}; \{w \in W \mid R_m(w) \le \delta + \varepsilon\}\big) \le \gamma.$$

We next turn to the result for $\operatorname{argmin}_{w\in W} R(w)$ and let $w^\star$ be a point in that set. Then, there is $k^\star$ such that $w^\star \in W_{k^\star}$ and

$$R_m(w^{k^\star}) \le R(w^{k^\star}) + \varepsilon \le R(w^\star) + \varepsilon = \inf_{w\in W} R(w) + \varepsilon.$$

Moveover, let $\bar{w} \in \operatorname{argmin}_{w\in W} R_m^{\gamma}(w)$. Then, there is $\bar{k}$ such that $\bar{w} \in W_{\bar{k}}$ and

$$\inf_{w\in W} R_m^{\gamma}(w) = \sup_{\|v\|\le\gamma} R_m(\bar{w} + v) \ge R_m(w^{\bar{k}}) \ge R(w^{\bar{k}}) - \varepsilon \ge \inf_{w\in W} R(w) - \varepsilon.$$

Combining these inequalities, we obtain that

$$R_m(w^{k^\star}) \leq \inf_{w\in W} R_m^\gamma(w) + 2\varepsilon.$$

Since $\|w^\star - w^{k^\star}\| \leq \gamma$, we have established that

$$\operatorname{exs}\big(\operatorname{argmin}_{w\in W} R(w);\, \{w \in W \mid R_m(w) \leq \inf_{\bar{w}\in W} R_m^\gamma(\bar{w}) + 2\varepsilon\}\big) \leq \gamma$$

and the conclusion follows. □

Since the constant β is nearly of the same form as in Theorem 3.3, the discussion following that theorem carries over to the present context. In particular, we note that a lower-level set of R_m, enlarged with γ, covers $\operatorname{argmin}_{w\in W} R(w)$ with high probability.

4 Algorithms for Diametrical Risk Minimization

Although there are some computational challenges associated with DRM, most of the existing optimization procedures for ERM can be adapted. Significantly, if the empirical risk function R_m is convex, then the diametrical risk function R_m^γ is also convex. Moreover, regardless of convexity,

$$\frac{1}{m}\sum_{i=1}^{m} \nabla_w \ell(\bar{w} + \bar{v}, z_i)$$

is a subgradient (in the general sense, cf. Rockafellar & Wets, 1998, Ch. 8) of R_m^γ at $\bar{w}$ under weak assumptions, where $\bar{v} \in \operatorname{argmax}_{\|v\|\leq\gamma} R_m^\gamma(\bar{w} + v)$; see for example Rockafellar and Wet (1998, Cor. 10.9). This implies that standard (sub)gradient methods apply provided that $\bar{v}$ can be computed. Since γ might very well be small, this could be within reach, at least approximately, by carrying one iteration of gradient *ascent*. However, this could become costly as computation of such subgradients need to access all data points. This challenge is similar to the one faced by adversarial training (Madry et al., 2018), but there the gradient ascent is carried out relative to the data; see also Gong et al. (2020), Wong et al. (2020) and Zheng et al. (2018).

We utilize a less costly approach based on the application of SGD to an outer approximation formed via sampling. In short, we approximate the inner maximization by maximizing over a finite set of random points inside the γ-neighborhood at the current solution w^t of each iteration. We find this approach to be effective, even when working with problems involving NNs where the dimension of w is in the millions. In these applications an outer approximation of R_m^γ using as little as 10-20 samples from $\{v \mid \|v\| = \gamma\}$ suffices to achieve improvement over ERM.

We observe also that DRM is related to but distinct from ERM with early termination. If from a minimizer $w_m^\star$ of R_m^γ the process of maximizing $R_m(w_m^\star + v)$ subject $\|v\| \leq \gamma$ follows the trajectory along which the algorithm approached $w_m^\star$ in the first place, then DRM would be equivalent to ERM that terminates a distance γ from a minimizer. However, this equivalence will only take place when $w_m^\star$ is approached along such direction. It appears that this will occur only occasionally.

4.1 Gradient based algorithm

We propose two variations of an SGD-based algorithm for DRM which we denote by Simple-SGD-DRM and SGD-DRM. We start with a simple version of the main algorithm that is easier to follow and then introduce the full algorithm with minor alterations aimed toward improving efficiency. In the following, let $\text{prj}_W(w)$ denote the projection of w on W and let $R_{B_t}(w) = \frac{1}{|B_t|}\sum_{z\in B_t}\ell(w,z)$ denote the empirical risk over a batch $B_t \subset S$.

Algorithm 1: Simple-SGD-DRM

Step 0. Initialize $w^0 \in W$, $r \in \mathbb{N}$, $t = 0$. Initialize sequence of batches $B_t \subset S$ and learning rates $\lambda_t > 0$ for $t = 1, \ldots, T$.
Step 1. Sample r random perturbations (directions): $U = \{u_1, \ldots, u_r \mid \|u\| = \gamma\}$.
Step 2. Select $u^\star \in \text{argmax}_{u\in U}\frac{1}{|B_t|}\sum_{z\in B_t}\ell(w^t + u, z)$.
Step 3. Compute $w^{t+1} = \text{prj}_W\big(w^t - \lambda_t \nabla_w R_{B_t}(w^t + u^\star)\big)$.
Step 4. If $t = T$, stop. Else, $t \leftarrow t + 1$ and return to Step 1.

The Simple-SGD-DRM algorithm, at each iteration t, performs an SGD update towards minimizing the approximating objective function

$$w \mapsto \max_{u\in U}\frac{1}{|B_t|}\sum_{z\in B_t}\ell(w+u, z_i).$$

The algorithm does so by first forming a set of r random directions (vectors) $U = \{u_1, \ldots, u_r\}$ with norm equal to γ. Then, it determines the more critical $u \in U$, i.e., $u^\star \in \text{argmax}_{u\in U}\frac{1}{|B_t|}\sum_{z\in B_t}\ell(w^t + u, z)$. A subgradient of the approximating objective function is then $\nabla_w R_{B_t}(w^t + u^\star)$.

This algorithm, however, does have drawbacks. First, sampling r new vectors in Step 1 at every iteration can be computationally expensive. It may be enough, as we will see in the experiments, to only perform Step 1 intermittently.[5] Second, it may be beneficial to reuse one or more of the sampled vectors from Step 1 in future iterations, particularly if we decide to perform sampling only intermittently.

The following algorithm, which we simply call SGD-DRM, includes these options explicitly. One will notice that it can be made equivalent to simple-SGD-DRM with particular choices, and is thus an extension with more options to save computation by limiting sampling.

Algorithm 2: SGD-DRM.

Step 0. Initialize $V_{-1} = \{\}$, $w^0 \in W$, $q \in \mathbb{N}$, $r \in \mathbb{N}$, $p \in [0, 1]$, $t = 0$. Initialize sequence of batches $B_t \subset S$ and learning rates $\lambda_t > 0$ for $t = 1, \ldots, T$.

[5] In experiments, we sample only every 5th iteration.

Step 1. Sample r random perturbations (directions): $U = \{u_1, \ldots, u_r \mid \|u\| = \gamma\}$.
Step 2. Select $u^\star \in \operatorname{argmax}_{u \in U} \frac{1}{|B_t|} \sum_{z \in B_t} \ell(w^t + u, z)$.
Step 3. Let $V_t = V_{t-1} \cup \{u^\star\}$. If $|V_t| > q$, remove oldest element from V_t.
Step 4. Select $v^\star \in \operatorname{argmax}_{v \in V_t} \frac{1}{|B_t|} \sum_{z \in B_t} \ell(w^t + v, z)$.
Step 5. Compute $w^{t+1} = \operatorname{prj}_W\left(w^t - \lambda_t \nabla_w R_{B_t}(w^t + v^\star)\right)$.
Step 6. If $t = T$, stop. Else, $t \leftarrow t + 1$ and with probability p, return to Step 1; with probability $1 - p$ return to Step 4 with $V_t = V_{t-1}$.

The primary difference between this and the former algorithm is within Steps 3–4 and Step 6. Step 6 allows one to skip the expensive sampling in Step 1 at some frequency represented by p. The new set V_t is introduced in Steps 3–4 to allow the reuse of one or more vectors u from previous iterations. As the algorithm progresses, the set V_t acts like a queue with maximal size q.[6] Every time the sampling of Step 1 is not skipped, V_t will be equal to the set V_{t-1} with its oldest element replaced by $u^\star$. For iterations $t \leq q$, the oldest element need not be removed since the queue has not reached its maximum length of q.

4.2 Implementation for neural networks

In the following experiments, we consider NN classifiers. Because of the structure of NNs we implement the perturbations $w + v$ and $w + u$ by considering the groupings of parameters that correspond to the structure of each NN layer. For example, a two layer NN might have parameter matrices $\{W_1, W_2\}$, each belonging to the separate network layers. Because of this, we select perturbations in Step 1 such that the *Frobenius norm*[7] of each layer-wise perturbation matrix is equal to γ. So, for the network with separable parameters $\{W_1, W_2\}$, a single sample from Step 1 would look like $\{U_1, U_2\}$ with $\|U_1\|_F = \gamma$, $\|U_2\|_F = \gamma$. Additionally, in the implementation, we first sample each component from a standard normal distribution, then normalize the resulting vector (or matrix) to have norm equal to γ.

Additionally, in the experiments we implement the coin flip (based on p) from Step 6 deterministically. We only perform Step 2 and 3 at every 5th step. Otherwise, we let $V_t = V_{t-1}$. This allows us to save computation time, particularly since the sampling of U can be expensive for NNs with millions of parameters.

5 Experiments

We aim to illustrate that DRM is resistant to overfitting and that its solutions have different local characteristics compared to those from ERM. In particular, we hypothesize that minimizers found by DRM lie in flat portions of the empirical risk landscape. To illustrate these aspects, we focus on the problem of classification with NNs in the presence of label noise, which is suitable because it is prone to overfitting as ERM settles into sharp minimizers; see Oymak et al. (2019) and Zhang et al. (2016). These works show that NNs have the unique ability to perfectly fit (with zero training error) both a data set with label noise as

[6] In experiments, we set $q = 1$.

[7] The Frobenius norm of a matrix A is defined as $(\sum_{i,j} a_{ij}^2)^{1/2}$, where a_{ij} are the elements of A.

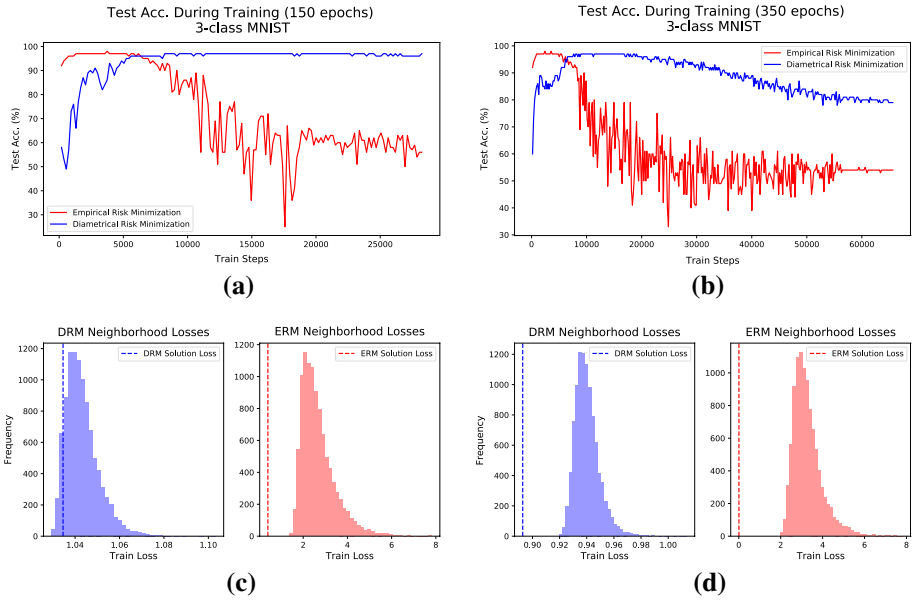


Fig. 2 MNIST experiments: **a** test accuracy when trained for 150 epochs. **b** Test accuracy when trained for 350 epochs. **c** Distribution of empirical risk for 10,000 random points in γ-neighborhood of final 150 epoch DRM and ERM solution. **d** Distribution of empirical risk for 10,000 random points in γ-neighborhood of final 350 epoch DRM and ERM solution

well as the same data with correct labels, with large generalization error in the former but small generalization error in the latter.

In these experiments, we train NNs on a subset of the MNIST and CIFAR-10 datasets with large amounts of label noise (50% of training labels flipped to an incorrect class). Using standard SGD for ERM (labelled SGD-ERM), the NNs indeed settle into solutions with high generalization error. On the other hand, SGD-DRM is remarkably resistant to overfitting, finding solutions with dramatically lower generalization error. Additionally, we find empirical evidence that the SGD-DRM solution lies in a flat portion of the empirical risk landscape compared to the SGD-ERM solution which appears in a sharper portion.

Code associated with these experiments and a PyTorch[8] based implementation of SGD-DRM is available online.[9] The computational resources is a single Tesla V100 GPU, 16 core Xeon processor and 64 GB memory. The operating system is Ubuntu. Generally, training time for DRM is approximately 3–5 times longer than those for ERM with the same architecture and data. We conjecture that the training times can be reduced by adapting the ideas from fast adversarial training; see for example Gong et al. (2020), Wong et al. (2020) and Zheng et al. (2018).

[8] https://pytorch.org/.

[9] https://github.com/matthew-norton/Diametrical_Learning.

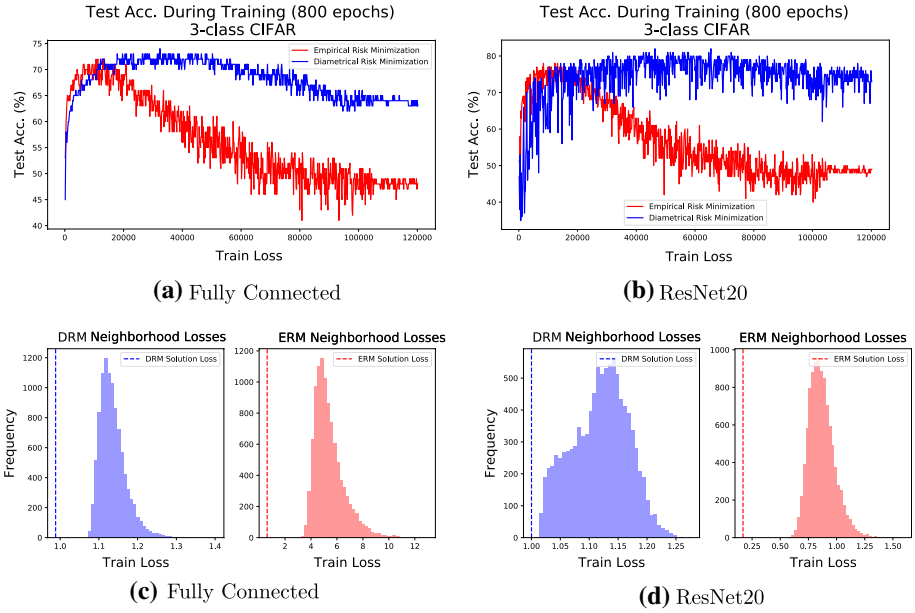
5.1 MNIST

In the first set of experiments, we train a fully connected, 3 layer NN with hidden units per layer being (320, 320, 200) with ReLU nonlinearities and an additional fully connected output layer feeding into a 3-class softmax negative log-likelihood objective function. For the MNIST dataset, we use only the handwritten digits zero, one, and two so that SGD-ERM achieves nearly zero training error. We flip 50% of the training labels to an incorrect class. For both SGD-ERM and SGD-DRM, we use standard SGD updates with batch size 100 and learning rate .01 until the last 50 epochs when decreased to .001. For SGD-DRM, the Step 1 perturbations are treated on a layer-wise basis (see Sect. 4.2) with $\gamma = 10$. We also use $r = 20$ and $q = 1$ for the size of U and V, respectively. We implement Step 6 deterministically with Step 1 being performed every 5th iteration (batch).

We train twice. First, we train the network for 150 epochs total with learning rate .01 for the first 100 and .001 for the final 50. Test accuracy can be seen in Fig. 2a. We first note the behavior of SGD-ERM. It begins by finding a good solution that generalizes, but then continues minimizing the empirical risk and settles into a solution that fits the training data with incorrect labels and thus suffers from a sharp decline in test accuracy (a sharp increase in generalization error). In contrast, SGD-DRM resists overfitting. Once it finds a good solution that generalizes, it is able to stay there, resisting the fall into a poor solution. As mentioned earlier, it is hypothesized that solutions in flat portions of the empirical risk landscape generalize better than those in sharp portions. Figure 2c plots the distribution of empirical risk for 10,000 random points in the γ-neighborhood of the final 150-epoch DRM and ERM solutions and illustrates that, indeed, the DRM solution is in a much flatter portion of the empirical risk landscape than the ERM solution. The dashed line represents the value of the empirical risk at the found solution and the rest of the plot is the distribution of empirical risk at points surrounding the solution.[10] Figure 2c also illustrates that, while the empirical risk of the SGD-ERM solution is lower (the red dashed line) than that of SGD-DRM it has much larger diametrical risk.

To allow SGD-ERM enough time to achieve nearly zero training error, we also train for 350 epochs with learning rate .01 for the first 300 and .001 for the final 50. Figure 2b depicts test accuracy. Again, we see the same behavior for SGD-ERM, as it chaotically falls into a poor solution that does not generalize. SGD-DRM remains resistant to overfitting. While it does experience some degradation in test accuracy, it still ends at a much better solution and its path is not nearly as chaotic; see the smooth vs choppy lines in Fig. 2b. We also see, again, that the SGD-DRM solution lies in a flatter portion of the empirical risk landscape than that from SGD-ERM. Figure 2d illustrates that, while the empirical risk of the SGD-ERM solution is lower (the dashed line at zero) it has (approximately) much larger diametrical risk equal to around 6. The SGD-DRM solution, on the other hand, has higher empirical risk ($\approx$.89) but much smaller diametrical risk ($\approx$.98).

[10] The points are sampled in the same way as for U in Step 1 of SGD-DRM, with neighborhood points sampled on a layer-wise basis as $\{w^* + u \mid \|u\| = \gamma\}$. Additionally, we use the same set of points u for approximating the neighborhood of ERM and DRM solutions.



(a) Fully Connected **(b)** ResNet20

(c) Fully Connected **(d)** ResNet20

Fig. 3 CIFAR-10 experiments: **a**, **b** Test accuracy when trained for 800 epochs. **c**, **d** Distribution of empirical risk for 10,000 random points in γ-neighborhood of final 800 epoch DRM and ERM solution

5.2 CIFAR-10

We perform similar experiments on CIFAR-10 using subclasses airplane, bird, and car and two network architectures: a fully connected and ResNet20. First, we use the same fully connected architecture as before with hidden 3 layers of size (320, 320, 200) with ReLU nonlinearities and an additional fully connected output layer feeding into a 3-class softmax negative log-likelihood objective function. We train the network on a dataset with 50% of training labels flipped to an incorrect class. For both SGD-ERM and SGD-DRM, we use standard SGD updates with batch size 100 and learning rate .01 for 750 epochs and .001 for the final 50 epochs. For SGD-DRM, the Step 1 perturbations are treated on a layer-wise basis (see Sect. 4.2) with $\gamma = 5$. We adopt $r = 20$, $q = 1$, and a deterministic implementation of Step 6 as above.

Results are nearly identical to those for MNIST; see Fig. 3a. SGD-ERM begins by finding a good solution that generalizes, but then continues minimizing the empirical risk and settles into a solution that fits the training data with incorrect labels and thus suffers from a sharp decline in test accuracy. SGD-DRM is again resistant to overfitting and is able to find a good solution that generalizes. Figure 3c illustrates that the DRM solution is in a much flatter portion of the empirical risk landscape than the ERM solution and achieves much smaller diametrical risk at the expense of larger empirical risk. The figure plots the distribution of empirical risk for 10,000 random points in the γ-neighborhood of the final 800-epoch DRM and ERM solutions. The dashed line represents the value of the empirical risk at the found solution and the rest of the plot is the distribution of empirical risk at points surrounding the solution.

For the ResNet20 architecture, we utilize identical settings except that $\gamma = 1$ due to the smaller number of parameters (per layer) and we omit perturbations to the batchnorm layer parameters. Figure 3b reports test accuracy. Unlike for the fully connect architecture,

SGD-DRM suffers nearly zero degradation of test accuracy as training progresses. It also achieves, and maintains, higher test accuracy than is achieved by SGD-ERM. Furthermore, Fig. 3d illustrates again the flatness of the landscape surrounding the SGD-SRM solution. While the diametrical risk is similar for both solutions, there is still a significant gap between the empirical risk of the SGD-ERM solution and points within its neighborhood. For SGD-DRM, this gap is much smaller, indicating a flatter landscape.

5.3 Discussion

While these experiments support the proposition that minimization of diametrical risk leads to good generalization and can be used to handle problems with large Lipschitz moduli, there is still much to be explored with DRM. In particular, SGD-DRM can be improved by considering different policies for choosing hyperparameters γ, λ, q, d, batch size, and p. The choice of diametrical risk radius γ, for example, could be chosen adaptively at every iteration along with the step size λ, mimicking adaptive SGD implementations such as Adam. Also, different choices of γ could be made for different groupings of parameters corresponding to the layers in a NN.

An additional algorithmic component that could be made more efficient is the estimation of the diametrical risk, which currently is a bottleneck. First, the sampling in Step 1 can be expensive, especially if done at every iteration or for large value of d. Second, if d and/or q is large, the memory requirements of storing U and V can be large. Third, the maximization in Steps 2 and 4 can also be expensive, particularly if the batch size is large and the loss is expensive to compute. Many of these issues, however, can be reduced with parallel implementation. For example, independent workers can each produce a single sample u_i and calculate the value of the objective function. Then, Step 2 can be performed by considering only the collection of d function values produced by the set of workers. We leave these tasks to future work, however, and use the presented results to encourage more work in this direction.

Acknowledgements This work is supported in part by AFOSR under F4FGA08272G001.

References

Bartlett, P., Foster, D., & Telgarsky, M. (2017). Spectrally-normalized margin bounds for neural networks. In *Advances in neural information processing systems* (pp. 6240–6249).

Ben-Tal, A., & Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research, 23*(4), 769–805.

Bertsimas, D., & Copenhaver, M. (2018). Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research, 270*(3), 931–942.

Bertsimas, D., Gupta, V., & Kallus, N. (2018). Robust sample average approximation. *Mathematical Programming, 171*(1), 217–282.

Boucheron, S., Lugosi, G., & Massart, P. (2016). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford: Oxford University Press.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research, 2,* 499–526.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., & Duchi, J. C. (2019). Unlabeled data improves adversarial robustness. arXiv:1905.13736.

Charles, Z., & Papailiopoulos, D. (2018). Stability and generalization of learning algorithms that converge to global optima. In *International conference on machine learning* (pp. 744–753).

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., et al. (2017). Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment, 2019*(12), 124018.

Cohen, J. M., Rosenfeld, E., & Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th international conference on machine learning (ICML)*.

Dong, Z., Yao, Z., Gholami, A., Keutzer, K., & Mahoney, M. W. (2019). HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the international conference on computer vision (ICCV)*.

Duchi, J., Glynn, P., & Namkoong, H. (2018). Statistics of robust optimization: A generalized empirical likelihood approach. arXiv:1610.03425.

Gong, C., Ren, T., Ye, M., & Liu, Q. (2020). Maxup: A simple way to improve generalization of neural network training. arXiv:2002.09024.

Gouk, H., Frank, E., Pfahringer, B., & Cree, M. (2018). Regularisation of neural networks by enforcing Lipschitz continuity. arXiv:1804.04368.

Hardt, M., Recht, B., & Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning* (pp. 1225–1234).

Hochreiter, S., & Schmidhuber, J. (1997). Flat minima. *Neural Computation, 9*(1), 1–42.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. arXiv:1609.04836.

Lewis, A. (2002). *Robust regularization*. Technical report, School of ORIE, Cornell University, Ithaca, NY.

Lewis, A., & Pang, C. (2010). Lipschitz behavior of the robust regularization. *SIAM Journal of Control and Optimization, 48*(5), 3080–3104.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems* (pp. 6389–6399).

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1778–1787).

Liu, H., Wang, X., Yao, T., Li, R., & Ye, Y. (2019). Sample average approximation with sparsity-inducing penalty for high-dimensional stochastic programming. *Mathematical Programming, 178,* 69–108.

Luo, Z.-Q. (2003). Applications of convex optimization in signal processing and digital communication. *Mathematical Programming, 97*(1–2), 177–207.

Luo, Z.-Q., Sturm, J., & Zhang, S. (2004). Multivariate nonnegative quadratic mappings. *SIAM Journal of Optimization, 14*(4), 1140–1162.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *Proceedings of the international conference on learning representations (ICLR)*.

Men, H., Freund, R., Nguyen, N., Saa-Seoane, J., & Peraire, J. (2014). Fabrication-adaptive optimization with an application to photonic crystal design. *Operations Research, 62*(2), 418–434.

Nguyen, Q., & Hein, M. (2017). The loss surface of deep and wide neural networks. In *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 2603–2612). JMLR.org.

Oberman, A. M., & Calder, J. (2018). Lipschitz regularized deep neural networks converge and generalize. arXiv:1808.09540.

Oliveira, R., & Thompson, P. (2017). Sample average approximation with heavier tails i: non-asymptotic bounds with weak assumptions and stochastic constraints. arXiv:1705.00822.

Oymak, S., Fabian, Z., Li, M., & Soltanolkotabi, M. (2019). Generalization guarantees for neural networks via harnessing the low-rank structure of the Jacobian. arXiv:1906.05392.

Pinar, M., & Arikan, O. (2004). On robust solutions to linear least squares problems affected by data uncertainty and implementation errors with application to stochastic signal modeling. *Linear Algebra and its Applications, 391,* 223–243.

Qian, H., & Wegman, M. N. (2018). L2-nonexpansive neural networks. arXiv:1802.07896.

Rockafellar, R. T., & Wets, R.J.-B. (1998). *Variational analysis, volume 317 of Grundlehren der Mathematischen Wissenschaft* (3rd printing-2009 edition). Berlin: Springer.

Royset, J. O. (2013). On sample size control in sample average approximations for solving smooth stochastic programs. *Computational Optimization and Applications, 55*(2), 265–309.

Royset, J. O., & Szechtman, R. (2013). Optimal budget allocation for sample average approximation. *Operations Research, 61,* 762–776.

Royset, J. O., & Wets, R.J.-B. (2017). Variational theory for optimization under stochastic ambiguity. *SIAM Journal of Optimization, 27*(2), 1118–1149.

Royset, J. O., & Wets, R.J.-B. (2021). *An optimization primer. Springer series in operations research and financial engineering*. Berlin: Springer.
Sagun, L., Bottou, L., & LeCun, Y. (2016). Eigenvalues of the hessian in deep learning: Singularity and beyond. arXiv:1611.07476.
Shalev-Shwartz, S., Shamir, O., Srebro, N., & Sridharan, K. (2010). Learnability, stability and uniform convergence. *Journal of Machine Learning Research, 11,* 2635–2670.
Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2009). *Lectures on stochastic programming: Modeling and theory*. Philipadia: SIAM.
Stinstra, E., & den Hertog, D. (2008). Robust optimization using computer experiments. *European Journal of Operations Research, 191,* 816–837.
Tsai, Y.-L., Hsu, C.-Y., Yu, C.-M., & Chen, P.-Y. (2021). Formalizing generalization and robustness of neural networks to weight perturbations. arXiv:2103.02200.
Wiesemann, W., Kuhn, D., & Sim, M. (2014). Distributionally robust convex optimization. *Operations Research, 62*(6), 1358–1376.
Wong, E., Rice, L., & Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *Proceedings of the international conference on learning representations (ICLR)*.
Wu, D., Xia, S.-T., & Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, *33*.
Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv:1611.03530.
Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the international conference on machine learning (ICML)* (pp. 7472–7482).
Zhang, J., Xu, H., & Zhang, L. (2016). Quantitative stability analysis for distributionally robust optimization with moment constraints. *SIAM Journal of Optimization, 26*(3), 1855–1882.
Zheng, T., Chen, C., & Ren, K. (2018). Is pgd-adversarial training necessary? Alternative training via a soft-quantization network with noisy-natural samples only. arXiv:1810.05665.