Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

ON SOLVING DISTRIBUTIONALLY ROBUST OPTIMIZATION FORMULATIONS EFFICIENTLY

Soumyadip Ghosh
Mark S. Squillante
Mathematical Sciences, IBM Research
Thomas J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10198, USA

Ebisa D. Wollega
Department of Engineering
Colorado State University-Pueblo
2200 Bonforte Boulevard
Pueblo, CO 81001, USA

ABSTRACT

In this paper we propose and investigate a new stochastic gradient descent (SGD) algorithm to efficiently solve distributionally robust optimization (DRO) formulations that arise across a wide range of applications. Our approach for the min-max formulations of DRO applies SGD to the outer minimization problem. Towards this end, the gradient of the inner maximization is estimated by a sample average approximation using a subset of the data in each iteration, where the subset size is progressively increased over iterations to ensure convergence. We rigorously establish convergence of our method for a broad class of models. For strongly convex models, we also determine the optimal support-size growth sequence that balances a fundamental tradeoff between stochastic error and computational effort. Empirical results demonstrate the significant benefits of our approach over previous work in solving these DRO formulations efficiently.

1 INTRODUCTION

We consider general formulations of the distributionally robust optimization (DRO) problem as follows. Let $\mathbb{X}$ denote a sample space, P a probability distribution on $\mathbb{X}$, and $\Theta \subseteq \mathbb{R}^d$ a parameter space. Define $L_P(\theta) := \mathbb{E}_P[l(\theta,\xi)]$ to be the expectation with respect to (w.r.t.) P of an objective function $l : \Theta \times \mathbb{X} \to \mathbb{R}$ that will also be called a *loss* function since we seek to minimize it over parameters $\theta \in \Theta$ given samples (data) $\xi \in \mathbb{X}$. Define the worst-case expected loss $R(\theta) := \mathbb{E}_{P^*(\theta)}[l(\theta,\xi)] = \sup_{P\in\mathscr{P}}\{L_P(\theta)\}$, which maximizes the loss L_P over a well-defined set of measures $\mathscr{P}$. This set typically takes the form $\mathscr{P} = \{P \,|\, D(P,P_b) \le \rho, \int dP(\xi) = 1, P(\xi) \ge 0\}$, where $D(\cdot,\cdot)$ is a measure of distance on a set or space of probability distributions on $\mathbb{X}$ and where the constraints limit the feasible candidates to be within a distance ρ of a base P_b. We then seek parameters $\theta^*_{rob} \in \Theta$ that, for a given $\mathscr{P}$, solve the DRO problem formulated as

$$\theta^*_{rob} = \arg\min_{\theta\in\Theta} \Big\{ R(\theta) \Big\} = \arg\min_{\theta\in\Theta} \Big\{ \sup_{P\in\mathscr{P}} \{L_P(\theta)\} \Big\}. \tag{1}$$

Nonparametric input model uncertainty is an important concern in simulation-based optimization. This uncertainty arises when only a finite set of observations $\mathscr{N} = \{\xi_n, n = 1,\ldots,N\}$ are available to characterize the inputs of the simulation model that estimates the loss function l. A confidence interval (CI) constructed for the expected loss using as an input model the equal-weight *empirical* distribution $U_N = \{1/N\}$ can provide poor coverage of the true value. A rich literature exists on constructing CIs using boot-strapping methods to incorporate the impact of input model uncertainty (Barton et al. 2014). Lam (2019) shows how ϕ-divergence balls centered at $P_b = U_N$ with appropriately chosen radius ρ can construct robust loss $R(\theta)$ as in (1) to obtain an asymptotically valid CI for $L_{P_0}(\theta)$ (for a fixed θ), where P_0 is the true unknown distribution generating the input samples $\mathscr{N}$. In terms of simulation optimization, the equal-weight empirical distribution U_N over $\mathscr{N}$ is the nonparametric maximum likelihood estimator (Owen 2001) of the (unknown)

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

distribution underlying the datasets, which motivates the standard practice of minimizing the *empirical loss* $L_{U_N}(\cdot)$ over Θ. The DRO philosophy seeks to extend the input model uncertainty analysis to model optimization by instead picking θ^*_{rob} as the best parameter. In practice, estimating with DRO formulations (1) amounts to dynamically re-weighing the data using the solution $P^*(\theta)$ to the inner maximization at each parameter $\theta \in \Theta$. Unlike the equal emphasis placed by $L_{U_N}(\theta)$ on all observed data, the $R(\theta)$ in DRO formulations sets these weights to emphasize data that experience high loss at θ. Hence, this approach explicitly treats the ambiguity in the identity of P_0, since in general $U_N \neq P_0$.

This problem is studied also in the statistical learning setting where the best model parameters θ of a statistical model is sought given only a finite *training* dataset $\mathscr{N}$ and this model is then used for inference over other test datasets, all of which are typically assumed to be identically distributed. In real-world settings, the training dataset and any dataset to which the trained model is applied are finite sets sampled from the same underlying distribution P_0. While popular model selection techniques, such as cross-validation (CV) (see, e.g., Stone (1974)), seek to improve the estimation error between training and testing datasets, they are often computationally prohibitive and lack rigorous guarantees. The DRO formulation (1) with the empirical distribution U_N over the finite training dataset as the base distribution P_b has been proposed as an alternative approach by Namkoong and Duchi (2016), Blanchet et al. (2019). Blanchet et al. (2019) show for Wasserstein distance metrics that, with an appropriately chosen value of constraint parameter ρ, $P_0 \in \mathscr{P}$ with high probability; and Namkoong and Duchi (2017) establish similar results for ϕ-divergence measures. The DRO approach therefore holds great promise.

Our primary focus is on efficiently obtaining solutions of (1), motivated to realize DRO as a viable statistical learning approach. The key obstacle is the min-max form, and specifically the inner maximization over probability sets $\mathscr{P}$. In some cases, its solution is explicitly available; e.g., $\mathscr{P}$ constrained by certain instances of Wasserstein distance, studied by Blanchet et al. (2019) and Sinha et al. (2018), admit an explicit characterization of the robust objective $\mathbb{E}_{P^*(\theta)}[l(\theta,\xi)]$. However, such reductions do not hold in general, and they require solving a convex nonlinear program (Esfahani and Kuhn 2018). Namkoong and Duchi (2017) show that the inner maximization with χ^2-divergence constraints can be efficiently solved. We therefore focus on the general ϕ-divergence distance function $D_\phi(P,P_b) = \mathbb{E}_{P_b}[\phi(\frac{dP}{dP_b})]$, where $\phi(s)$ is a nonnegative convex function taking a value of 0 only at $s = 1$. The modified χ^2 and Kullback-Leibler (KL) divergences are given by $\phi(s) = (s-1)^2$ and $\phi(s) = s\log s - s + 1$, respectively. Define the N-sized vector $P := (p_n)$ and set the base $P_b = U_N$. Then:

$$L_P(\theta) = \sum_{n=1}^{N} p_n l(\theta,\xi_n) \quad \text{and} \quad \mathscr{P} = \Big\{ P \,\Big|\, D_\phi(P,U_N) = \frac{1}{N}\sum_{n=1}^{N} \phi(Np_n) \leq \rho,\ \sum_{n=1}^{N} p_n = 1, p_n \geq 0, \forall n \Big\}.$$

For convex loss functions $l(\cdot,\xi_n)$, the DRO formulation (1) is convex in θ. Ben-Tal et al. (2013) describe the typical Lagrangian dual algorithm used for this convex-concave case (see, e.g., (3)), and apply classical stochastic gradient descent (SGD) to solve this reformulation to a standard stochastic optimization form. Such an approach fails when l is non-convex since the required strong duality does not hold; and the characteristics of certain dual variables can cause instability in the SGD (Namkoong and Duchi 2016). To address this, Namkoong and Duchi (2017) determine for the modified χ^2 case the optimal $P^*(\theta)$ that defines $R(\theta)$ exactly by solving (1) as a large deterministic problem that is well-defined for finite N. This full-gradient approach is feasible for specific choices of ϕ-divergences by reduction to two one-dimensional root-finding problems that can be solved via bisection search. This nevertheless requires an $O(N\log N)$ effort (Proposition 2) at each iteration, and hence the method can be expensive. Levy et al. (2020) and Ghosh and Squillante (2020) independently develop an algorithm based on multi-level Monte Carlo (Giles 2008; Blanchet and Glynn 2015) to alleviate the computational burden of the gradient estimation. While the algorithm holds great promise especially for large-scale datasets, our experiments reveal that it inherently induces an increase in the variance of the gradient estimator due to the added randomization, which can in turn adversely affect the computational requirements and solution quality of the overall approach.

In this paper *our primary contributions* include a new SGD algorithm to efficiently solve large-scale DRO problems (Section 2), namely Alg. 1 which *subsamples* the support of the variable P from a (finite) observed dataset and estimates the robust loss (and its gradient) via a sample average approximation (SAA). While subsampling (mini-batching) typically works well with SGD because the gradient estimates are unbiased, our theoretical results (Theorem 3) show that subsampling in the DRO context induces a bias in the estimation of the robust loss gradient. This is due to a fixed mini-batch size resulting in a high chance that critical data which suffer high loss will be missed, leading to an optimistic estimation of the robust loss. Namkoong and Duchi (2017) sidestep this issue by assembling the full-data gradient, while the Giles estimator of Levy et al. (2020), Ghosh and Squillante (2020) randomizes the choice of the mini-batch size over the entire dataset. Alg. 1 reduces this bias by progressively growing the subsample size. We establish theoretical results (Theorem 5) showing that convergence is assured even for non-convex losses l as long as the subsamples grow at a certain minimal rate. For strongly convex losses l (Theorem 7), we further show how to optimally set the parameters of our algorithm so that the required computational effort is balanced with the desired level of accuracy, thus providing the fastest rate of convergence.

Our primary contributions also include empirical results (Section 3) that consider convex DRO formulations of binary classification problems, comparing the performance of our Alg. 1 against the methods referenced above and a regularized ERM formulation tuned via k-fold CV. Our results show that our algorithm can attain the same or better performance as k-fold CV and all other DRO methods, while also providing significant computational speedup over the other DRO methods and orders of magnitude reductions in the computation time required by k-fold CV. The key parameters of our algorithm do not require fine tuning and are set based on our theoretical results, whereas the key parameters of the other DRO approaches impact both the solution quality and the computation time and require fine tuning.

2 ALGORITHM AND ANALYSIS

Alg. 1 presents our progressively sampled subgradient descent algorithm comprising SGD-like iterations

$$\theta_{t+1} \;=\; \theta_t - \gamma \nabla_\theta \hat{R}_t(\theta_t) \;=\; \theta_t - \gamma G_t, \tag{2}$$

for the outer minimization in (1), where γ is the (fixed) step size, $\hat{R}_t(\cdot)$ is an SAA of the robust loss $R(\cdot)$ from the inner maximization over D_ϕ-constrained $\mathscr{P}$, and $G_t := \nabla_\theta \hat{R}_t(\theta_t)$. This view of (1) allows us to depart from the convex-concave formulations of Ben-Tal et al. (2013) and consider non-convex losses l, as long as the subgradient $\nabla_\theta \hat{R}_t(\cdot)$ approximates the gradient $\nabla_\theta R(\cdot)$ sufficiently well. Recall that $R(\theta)$ is the optimal value of the inner maximization problem. Define the set $\Theta_\varnothing := \{\theta : l(\theta,\xi_{n_1}) = l(\theta,\xi_{n_2}),\ \forall n_1,n_2\}$, and for a small $\varsigma > 0$ let the set $\Theta_{\varnothing,\varsigma} := \cup_{\theta_o \in \Theta_\varnothing}\{\theta : \|\theta - \theta_o\|_2 < \varsigma\}$ define the ς-neighborhood of $\Theta_\varnothing$. Proposition 1 assumes that the formulation (1) precludes $\Theta_{\varnothing,\varsigma}$ in order to avoid a degenerate inner maximization objective function that does not depend on the decision variables p_n, in which case the entire feasible set is optimal.

Proposition 1 Let the feasible region Θ be compact and assume $\Theta \subseteq \Theta^c_{\varnothing,\varsigma}$, for a small $\varsigma > 0$. Further suppose ϕ in the D_ϕ-constraint has strictly convex level sets. Then: (i) the optimal solution P^* of $R(\theta) = \sup_{P\in\mathscr{P}}\{L_P(\theta)\}$ is unique if $\rho < \left(1 - \frac{1}{N}\right)\phi\left(\frac{N}{N-1}\right) + \frac{1}{N}\phi(0)$, and the gradient is given by $\nabla_\theta R(\theta) := \sum_{n\in\mathscr{N}} p^*_n(\theta)\nabla_\theta l(\theta,\xi_n)$; and (ii) for all ρ, the $\nabla_\theta R(\theta)$ is a sub-gradient of $R(\theta)$.

Proof Sketch: For part (i), the detailed exposition in Ghosh et al. (2020) shows that the assumption on the constraint parameter ρ only admits feasible probability mass functions (pmfs) that assign nonzero mass to all support points. For strictly convex functions $\phi(\cdot)$, this then ensures that the problem (4) has a unique optimal solution P^* when combined with the assumption that the objective coefficients $l(\theta,\xi_n) \neq \ell$ for all n and some ℓ. Moreover, $D_\phi(P^*,U_N) = \rho$. For the form of the gradient, we can write the Lagrangian dual form (Luenberger 1969) of the inner maximization as

$$R(\theta^*_{rob}) = \min_{\theta\in\Theta}\max_{p_n\geq 0}\min_{\alpha\geq 0,\lambda}\left\{\mathscr{L}(\theta,\alpha,\lambda,P) \;:=\; L_P(\theta) + \alpha(\rho - D_\phi(P,U_N)) + \lambda\left(1 - \sum_{n=1}^{N} p_n\right)\right\}, \tag{3}$$

1: **procedure** PROGRESSIVESSD($\gamma, \{M_t\}, \theta_0, \rho$)
2:
3: **for** $t = 1, 2, \ldots, \mathscr{T}$ **do**
4: $\mathscr{M}_t \leftarrow \varnothing$
5: ▷ Sample subset $\mathscr{M}_t$
6: **for** $m = 1, \ldots, M_t$ **do**
7: $\xi_m \sim$ Uniform $(\mathscr{N} \setminus \mathscr{M}_t)$
8: $\mathscr{M}_t \leftarrow \mathscr{M}_t \cup \{\xi_m\}$
9:
10: Assemble $\mathscr{Z}_t \leftarrow \{l(\theta_t, \xi_m),\ \forall m \in \mathscr{M}_t\}$
11: $\rho_t \leftarrow \rho + c\left(\frac{1}{M_t} - \frac{1}{N}\right)^{(1-\delta)/2}$
12: $P_t^* \leftarrow$ `InnerMax`$(\mathscr{Z}_t, \mathscr{M}_t, \rho_t)$
13: Set $G_t \leftarrow \sum_{m \in \mathscr{M}_t} p_{t,m}^* \nabla_\theta l(\theta_t, \xi_m)$
14: Set $\theta_{t+1} \leftarrow \theta_t - \gamma G_t$
15: **return** $\theta_{\mathscr{T}}$
16: **end procedure**
17:

(a) Outer Minimization, where input parameters include step size γ, sample size sequence $\{M_t\}_{t=1}^{\mathscr{T}}$ with $M_{\mathscr{T}} = N$, initial iterate θ_0, D_ϕ constraint ρ.

1: **procedure** INNERMAX($\mathscr{Z}, \mathscr{M}, \rho$)
2: $M \leftarrow |\mathscr{M}|$, base $P_b = \left\{\frac{1}{M}, \forall m \in \mathscr{M}\right\}$
3: $\bar{z} \leftarrow \max_m \{z_m \mid z_m \in \mathscr{Z}\}$
4: $\mathscr{M}' \leftarrow \{m \in \mathscr{M} : z_m = \bar{z}\}$ and $M' \leftarrow |\mathscr{M}'|$
5: $P' \leftarrow \left\{\frac{1}{M'}\mathbb{I}\{m \in \mathscr{M}'\},\ \forall m \in \mathscr{M}\right\}$
6: **If** $D_\phi(P^*, P_b) \le \rho$ **then**
7: $P^* \leftarrow P'$ and **return** P^*
8: **for** $\alpha \in [0, \bar{\alpha}]$ **do**
9: **for** $\lambda \in [\underline{\lambda}, \bar{\lambda}]$ **do**
10: $\mathscr{M}' \leftarrow \{m \mid \lambda \le z_m - \alpha\phi'(0)\}$
11: $P' \leftarrow \left\{\frac{1}{M}(\phi')^{-1}(\frac{z_m - \lambda}{\alpha}), m \in \mathscr{M}'\right\}$
12: **If** $\sum_m p'_m = 0$, **then**
13: $P^*(\alpha) \leftarrow P'$, and **break**
14: **If** $D_\phi(P^*(\alpha), P_b) = \rho$, **then**
15: $P^* \leftarrow P^*(\alpha)$ and **break**
16: **return** P^*
17: **end procedure**

(b) Inner Maximization, where input parameters include loss values $\mathscr{Z}$, subsampled support $\mathscr{M}$, D_ϕ constraint ρ.

Alg. 1: Progressively Sampled Subgradient Descent Algorithm

where $\mathscr{L}(\theta, \alpha, \lambda, P)$ is the Lagrangian objective. By Lagrangian duality principles (Lemma 2.1 in Shapiro (1985)), a corresponding unique pair (α^*, λ^*) exists. Let us collectively call the primal and dual variables $v^*(\theta) = (\alpha^*(\theta), \lambda^*(\theta), P^*(\theta))$, and hence $R(\theta) = \mathscr{L}(\theta, v^*(\theta))$ where the first term $L_{P^*}(\theta) = \sum_n p_n^*(\theta) l(\theta, \xi_n)$. Differentiating using the chain rule, we obtain $\nabla_\theta R(\theta) = \nabla_\theta L_{P^*(\theta)}(\theta) + \nabla_\theta v^*(\theta)\ \nabla_v \mathscr{L}(\theta, v^*(\theta)) = \sum_{n \in \mathscr{N}} p_n^*(\theta) \nabla_\theta l(\theta, \xi_n)$, where the second term in the first summation vanishes because $\nabla_v \mathscr{L}(\theta, v^*(\theta)) = 0$ by the first order optimality conditions of v^*. Part (ii) obtains the same result in the more general setting that allows for multiple solutions to the maximization problem; see Theorem 7.21, p. 352 Shapiro et al. (2009). □

We next construct the estimate $\hat{R}_t(\theta)$ in (2) from the inner maximization problem restricted only to a relatively small subset $\mathscr{M}_t$ of size $|\mathscr{M}_t| = M_t$ of the full dataset $\mathscr{N}$ of size N. Alg. 1a in lines 6-8 shows that this subset is sampled uniformly *without replacement* from the full dataset $\mathscr{N}$; the discussion following Theorem 3 (below) motivates this method of generation. Defining $P = (p_m)$ of dimension M_t and the objective coefficients $z_m = l(\theta, \xi_m)$, we have

$$\hat{R}_t(\theta) \ = \ \max_{P=(p_m)} \sum_{m \in \mathscr{M}_t} p_m z_m \qquad \text{s.t.} \ \sum_{m \in \mathscr{M}_t} \phi(M_t p_m) \le M_t \rho_t, \ \sum_{m \in \mathscr{M}_t} p_m = 1, \ p_m \ge 0, \tag{4}$$

where the uncertainty radius $\rho_t = \rho + \eta_t$ now changes with the subsample size M_t in iteration t, motivated by Theorem 3 discussed below, where $\eta_t = c(1/M_t - 1/N)^{(1-\delta)/2}$ for small positive constants c, δ. Suppose $P_t^*(\theta) = (p_{t,m}^*(\theta))$ is an optimal solution to (4). Then a valid subgradient for $\hat{R}_t(\theta_t)$ is obtained as an expression analogous to that in Proposition 1(i) under appropriate substitutions w.r.t. θ_t, P_t^* and $\mathscr{M}_t$. Alg. 1a uses a progressively increasing subsample support size M_t as $t \nearrow \mathscr{T}$. At iteration $\mathscr{T}$ the algorithm reaches the full support size $M_{\mathscr{T}} = N$, and the user may then optionally choose to switch to a deterministic optimization algorithm, such as the one due to Namkoong and Duchi (2017).

Alg. 1b presents the steps needed to obtain an exact solution to the inner maximization problem (4), by solving the Lagrangian dual form analogous to (3) of the subsampled problem (4) to obtain the optimal

primal and dual variables for various ϕ functions. Note that the '**for**' loops over dual variables α (line 8) and λ (line 9) can be implemented efficiently as bisection searches that seek the zeros of the gradient of $\mathscr{L}(\theta,\alpha,\lambda,P)$ w.r.t. these variables, which reduces to seeking values where the equality constraints $D_\phi(P,P_b)=\rho$ and $\sum_{m\in\mathscr{M}_t} p_m = 1$ are respectively satisfied. This observation leads to the next result which provides a worst-case bound on the computational effort required to obtain an ε-optimal solution to (4).

Proposition 2 For any ϕ-divergence, Alg. 1b finds a feasible primal-dual solution $(\tilde{\alpha}^*,\tilde{\lambda}^*,\tilde{P}_t^*)$ to (4) with an objective value $\tilde{R}_t^*$ such that $|\hat{R}_t^*(\theta)-\tilde{R}_t^*|<\varepsilon$ and with a worst-case computational effort bounded by $O(M_t \log M_t + (\log\frac{1}{\varepsilon})^2)$, where ε is a small precision parameter.

A complete proof, including the monotonicity of the left hand expressions in lines 12 and 14 and the existence of a root within the chosen bounds for α and λ in lines 8 and 9, are available in Ghosh et al. (2020). The machine-precision ε does not relate to any other parameter of the formulation or algorithm (e.g., M_t, N, ρ), and it is required because of the two one-dimension bisection searches in sequence to obtain α^* and λ^*, respectively. In the sequel we assume that ε is a fixed small value and Alg. 1b returns an exact solution $(\alpha^*,\lambda^*,P_t^*)$ to problem (4), and that the computational effort is bounded by $O(M_t \log M_t)$.

2.1 Bias in $\nabla_\theta \hat{R}(\theta)$ as Approximation of $\nabla_\theta R(\theta)$

We shall now assume, to simplify the exposition, that the conditions of Proposition 1 hold in order to assume that the inner maximization has a unique solution. Let the mass vector $P^*=(p_1^*,\ldots,p_N^*)$ be the optimal solution to the full-data version of (4), i.e., with $M_t=N$, and let the mass vector $P_t^*=(p_1^*,\ldots,p_{M_t}^*)$ be the optimal solution of (4) restricted to any subset $\mathscr{M}_t$. The classical literature on SAAs (Shapiro 2003) of stochastic optimization formulations indicates that the robust loss approximation $\hat{R}(\theta_t)$ assembled from P_t^* suffers a bias w.r.t. the true robust loss $R(\theta)$. In our setting, this bias arises because the approximate problem might miss support points ξ_n where loss is high when subsampling the dataset, and this leads to optimistic estimation of the robust loss $R(\theta)$. We provide in Theorem 3 a bound on the squared bias when using the sub-gradient expression in Proposition 1(i) to approximate $\nabla_\theta R(\theta_t)$ as a function of the sample size M_t. We start with an assumption on the growth of the ϕ-function underlying the D_ϕ divergence.

Assumption 1 The ϕ-divergence satisfies uniformly for all s and $\zeta<\zeta_0$ the continuity condition (for constants $\zeta_0,\kappa_1,\kappa_2>0$): $|\phi(s(1+\zeta))-\phi(s)|\le\kappa_1\zeta\phi(s)+\kappa_2\zeta$.

This condition (Shapiro et al. 2009) only allows for (local) linear growth in ϕ, and it can be verified for many common ϕ-divergences of interest including the modified χ^2-divergence metric and the KL-divergence metric. Let $\mathbb{E}_t$ and $\mathbb{P}_t$ respectively denote expectation and probability w.r.t. the random set $\mathscr{M}_t$.

Theorem 3 Suppose Assumption 1 and the assumptions of Proposition 1 hold, and further assume that the loss functions $l(\cdot,\xi)$ are Lipschitz continuous in θ for all $\xi\in\mathscr{N}$. Define $\eta_t = c(\frac{1}{M_t}-\frac{1}{N})^{(1-\delta)/2}$ for small constants $c,\delta>0$, and set the D_ϕ-target in (4) to be $\rho_t=\rho+\eta_t$. Then, there exists a small positive M' of order $o(N)$ such that, for all $M_t\ge M'$, the subgradient $\nabla_\theta\hat{R}_t(\theta)$ and full-gradient $\nabla_\theta R(\theta)$ satisfy for any $C<\infty$ and $1-\bar{\tau}_t=O(\eta_t^{2\delta/(1-\delta)})$: $\mathbb{P}_t(\eta_t^{-2}\|\nabla_\theta\hat{R}_t(\theta)-\nabla_\theta R(\theta)\|_2^2\le C)\ge\bar{\tau}_t$.

The proof of this result (sketched below) makes apparent that the squared bias in Theorem 3 actually drops, via η_t, as a function of the number $|\mathscr{M}_t|$ of unique support points in the mini-batch. If M_t support points are sampled *with replacement*, the set $\mathscr{M}_t$ will have an expected number $\mathbb{E}|\mathscr{M}_t| = 1+1/2+1/3+\ldots = O(\log M_t)$ of unique values, obtained by adding the frequency of additional samples needed to see a new support sample. Thus, sampling with replacement results in a slow reduction in bias, which motivates the choice in Alg. 1a of assembling the subset $\mathscr{M}_t$ by sampling uniformly *without replacement* M_t values from the complete observed dataset.

Sampling without replacement differs from the standard with-replacement approach in the stochastic optimization literature, even though it is preferred by practitioners in statistical learning. A short comment on the probability measures $\mathbb{P}_t$ generated by sampling finite sets without replacement is in order. Let $\{x_1,\ldots,x_N\}$ be a set of N values with mean $\mu=\frac{1}{N}\sum_n x_n$ and variance $\sigma^2=\frac{1}{N-1}\sum_n(x_n-\mu)^2$. Suppose we

sample $M < N$ of these points uniformly without replacement to construct the set $\mathscr{M} = \{X_1, \ldots, X_M\}$. The probability of choosing any particular set of M subsamples is given by $\binom{(N-M)!}{N!}$. Let $\bar{X} = \frac{1}{M}\sum_{m=1}^{M} X_m$ and $\bar{S}^2 = \frac{1}{M-1}\sum_{m=1}^{M}(X_m - \bar{X})^2$ represent the sample mean and sample variance, respectively. The expectation of the sample mean $\mathbb{E}[\bar{X}] = \mu$ and of the sample variance $\mathbb{E}_{\mathscr{M}}[\bar{S}^2] = \sigma^2$ are both unbiased; refer to Wilks (1962). Moreover, the variance of the sample mean is $\mathbb{E}_{\mathscr{M}}[(\bar{X} - \mu)^2] = \left(\frac{1}{M} - \frac{1}{N}\right)\sigma^2$, which reduces to zero as $M \to N$. Note that this expression relates to the form of η_t in Theorem 3. The complete proof of Theorem 3, including exact expressions for the corresponding constants such as M' and $\bar{\tau}_t$ and their dependence on the constant C and on the magnitude of $\nabla_\theta R(\theta)$, are made explicit in Ghosh et al. (2020).

Proof Sketch of Theorem 3: The proof starts by constructing $\tilde{P}^*$, a restriction of the (unique) optimal solution P^* of the full-data version of the inner maximization (4) onto the (random) subset $\mathscr{M}_t$ in the restricted problem (4), where $\tilde{p}^*_m \propto p^*_m$, $\forall m \in \mathscr{M}_t$. The assumed minimum number of unique support points M' ensures with probability at least $\bar{\tau}_t$ that $\tilde{p}^*_m \neq \mathbf{0}$ and that $\tilde{P}^*$ is also a feasible solution to (4) when ρ_t is inflated as assumed. Next, we establish that its objective value is within η_t of the optimum with high probability, showing that $\mathbb{P}_t\left({\eta_t}^{-1}|\hat{R}_t(\theta) - R(\theta)| \leq c_1\right) \geq \bar{\tau}_t$. Proposition 1 provides that $\nabla_\theta R(\theta) = \sum_{n\in\mathscr{N}} \nabla_\theta l(\theta, \xi_n) p^*_n$ and our approximation $\nabla_\theta \hat{R}_t(\theta) = \sum_{m\in\mathscr{M}_t} \nabla_\theta l(\theta, \xi_m)\hat{p}_m$. The mean-value theorem of calculus yields $(\nabla_\theta l(\theta, \xi_n))_u = \frac{\partial l(\theta,\xi_n)}{\partial \theta_u} = \frac{1}{h_{u,n}}(l(\theta + h_{u,n}\mathbf{e}_u, \xi_n) - l(\theta, \xi_n))$, where $h_{u,n}$ is a small positive value that depends on the component θ_u and on the sample ξ_n, with $\mathbf{e}_u$ the unit-vector in the u-th coordinate. Letting $\underline{\mathrm{h}} = \min_{u,n} h_{u,n}$, we then have $\left|(\nabla_\theta \hat{R}_t(\theta) - \nabla_\theta R(\theta))_u\right| \leq \frac{1}{\underline{\mathrm{h}}}\left|\left[\sum_n l(\theta + h_{u,n}\mathbf{e}_u, \xi_n)^T(p^*_n - \hat{p}_n)\right]\right| + \frac{1}{\underline{\mathrm{h}}}\left|\left[l(\theta, \xi_n)^T(p^*_n - \hat{p}_n)\right]\right|$. Using the Lipschitz assumption on $l(\cdot, \xi)$ reduces the first term to a term similar to the second. Then, applying the same arguments as those used in deriving the high-probability bound on $\mathbb{P}_t\left({\eta_t}^{-1}|\hat{R}_t(\theta) - R(\theta)| \leq c_1\right)$, when squared and combined over all u, renders the desired final result. □

In the sequel, we use a corollary of Theorem 3 that follows from Theorem 17.4 in Jacod and Protter (2004).

Corollary 4 If the conditions for Theorem 3 are satisfied, then $\|\mathbb{E}_t[\nabla_\theta \hat{R}_t(\theta)] - \nabla_\theta R(\theta)\|_2^2 = O(\eta_t^2)$.

Thus, the bias in the estimator of the robust loss gradient vanishes only as $M_t \nearrow N$ as $t \to \infty$. Since fixed bias violates a basic requirement for standard SGD (see Section 4.3 in Bottou et al. (2018)) that the gradient estimator is bounded above and below as $\mathbb{E}[\nabla_\theta \hat{R}_t(\theta)] = \Theta(\nabla_\theta R(\theta))$, then the convergence of (2) cannot be guaranteed when $M_t = M$ for all t, where $M < N$. Namkoong and Duchi (2017) circumvent this bias issue by only working with the full-data gradient $\nabla_\theta R(\theta)$. Levy et al. (2020) and Ghosh and Squillante (2020) randomize the subset size M_t such that all data have a (small) positive probability of being included in $\mathscr{M}_t$, doing so in a manner that eliminates this bias at the expense of added variance. In contrast, Alg. 1 chooses to grow the mini-batch size M_t progressively with t to eliminate such bias. Since this also provides a decrease in noise as a consequence, it is no longer necessary to diminish the step size as iterations t grow; indeed, doing so negates the benefits of the extra work in computing gradients using a larger M_t. Alg. 1 therefore takes fixed-length steps γ. Fixed step-lengths and increasing sample (or mini-batch) sizes have also been proposed in the context of unbiased gradient estimators to temper the variance observed in the iterations of standard SGD; see Section 5.1 in Bottou et al. (2018).

2.2 Convergence of Alg. 1

We next analyze the convergence of (2) under the following additional assumptions.

Assumption 2 (i) A lower bound $R_{\inf}$ exists for the robust loss function $R(\theta) \geq R_{\inf}$, $\forall \theta \in \Theta$.
(ii) The variance of $\nabla \hat{R}_t(\theta)$ with subsample size M obeys $\mathbb{E}[\|\nabla \hat{R}(\theta) - \mathbb{E}[\nabla \hat{R}(\theta)]\|_2^2] \leq C(\frac{1}{M} - \frac{1}{N})$.
(iii) The robust loss objective $R(\theta)$ has L-Lipschitz gradients $\nabla_\theta R(\theta)$.
(iv) The loss functions $l(\theta, \xi_n)$ are c-strongly convex.

Assumption 2(ii) ensures that the variance of $\nabla \hat{R}_t(\theta)$ follows as expected for sampling without replacement from any finite set (Wilks 1962). Combining this with the bias (see Corollary 4) will allow us

to progressively decrease the mean squared error. Since N is finite, any M_t strictly increasing as t grows will eventually end at an iterate $\mathscr{T} < \infty$ where $M_{\mathscr{T}} = N$. Seen in this light, Alg. 1 is not guaranteed to converge by the $\mathscr{T}$-th iteration. We can nevertheless provide a guarantee on the performance of the method over any loss function that satisfies Assumptions 2(i)-(iii).

Theorem 5 Suppose the loss functions satisfy Assumption 2(i)-(iii), and the conditions of Theorem 3 hold. Further, let the constant step size satisfy $\gamma \leq \frac{1}{2L}$. Then, at termination, we have

$$\sum_{t=1}^{\mathscr{T}} \|\nabla_\theta R(\theta_t)\|_2^2 \;\leq\; \frac{R(\theta_0) - R_{\text{inf}}}{\frac{\gamma}{2}(2 - L\gamma)} \;+\; C\frac{L\gamma + 1}{2 - L\gamma} \sum_{t=1}^{\mathscr{T}} \eta_t^2. \tag{5}$$

Proof Sketch: For any θ and a set $\mathscr{M}_t$ sampled to have M_t support points, Theorem 3 and Assumption 2(ii) show that the mean squared error $\mathbb{E}_t\left[\|\nabla_\theta \hat{R}_t(\theta) - \nabla_\theta R(\theta)\|_2^2\right] = O(\eta_t^2)$ since the slower rate of decrease in the bias prevails. Elementary algebraic manipulations yield the following two implications: $\mathbb{E}_t[\|\nabla_\theta \hat{R}_t(\theta)\|_2^2] \leq C\eta_t^2 + \|\nabla_\theta R(\theta)\|_2^2$ and $-\mathbb{E}_t[(\nabla_\theta \hat{R}_t(\theta))^T \nabla_\theta R(\theta)] \leq C\eta_t^2 - \|\nabla_\theta R(\theta)\|_2^2 - \mathbb{E}_t[\|\nabla_\theta \hat{R}_t(\theta)\|_2^2]$. We can therefore bound the expected robust loss at step $(t+1)$ using (see Ghosh et al. (2020))

$$\begin{aligned} R_{\text{inf}} \leq \mathbb{E}_t[R(\theta_{t+1})] &\leq \mathbb{E}_t[R(\theta_t)] - \gamma \mathbb{E}_t[\nabla_\theta R(\theta_t)^T \nabla_\theta \hat{R}_t(\theta_t)] + \frac{L\gamma^2}{2} \mathbb{E}_t\left[\|\nabla_\theta \hat{R}(\theta_t)\|_2^2\right] \\ &\leq \mathbb{E}_t[R(\theta_t)] + \frac{C\gamma\eta_t^2}{2}(L\gamma + 1) - \frac{\gamma}{2}(2 - L\gamma)\|\nabla_\theta R(\theta_t)\|_2^2. \end{aligned} \tag{6}$$

The desired final result is obtained by rearranging (6) and telescoping back to the initial iterate θ_0. □

Theorem 5 establishes that the sum of the gradients of $R(\theta_t)$ at iterates visited by the algorithm is bounded above, in particular by $\sum_{t=1}^{\mathscr{T}} (\frac{1}{M_t} - \frac{1}{N})^{(1-\delta)}$. If this summation remains finite as $\mathscr{T} \to \infty$, then the upper bound of (5) remains finite, and hence the gradients $\|\nabla_\theta R(\theta_t)\|_2^2$ at the iterates converge to 0; in other words, the algorithm converges to a local optimal solution. The summation can converge for M_t increasing moderately, such as at a polynomial rate. Thus, our algorithm comes substantially close to a local minimizer in $\mathscr{T}$ iterations when the sample set sizes M_t are chosen to satisfy a minimum-growth condition.

Theorem 5 assumes that the robust loss gradient $\nabla_\theta R(\theta)$ is Lipschitz continuous (Assumption 2(iii)). Gradients of such extreme value functions are in general not Lipschitz even if the objective function $l(\theta, \xi)$ is Lipschitz. For example, a linear objective $l(\theta, \xi_n) = \theta^t \xi_i$ leads to $R(\theta) = \max_p \sum_i p_i \theta^t \xi_i$ and when maximized over a *polyhedral* constraint set it will not preserve the 0-Lipschitzness of the objective functions, because the optimal solutions P^* are picked from the discrete set of vertices of the polyhedron and hence $\nabla_\theta R(\theta)$ is piecewise discontinuous. Our Proposition 1 assumptions yield an inner maximization with a nonzero linear objective over a *strictly convex* feasible set. The desired smoothness can then be obtained with some additional conditions on the loss functions $l(\theta, \xi_i)$. Proposition 6 provides one such condition where the Lipschitzness of $\nabla_\theta R(\theta)$ follows from bounds on the Hessian of the individual losses $l(\theta, \xi)$.

Proposition 6 Assume the conditions in Proposition 1 hold. Further suppose that the Hessians $\nabla_\theta^2 l(\theta, \xi_n)$ exist, $\forall \theta$ and each ξ_n, and are bounded in Frobenius norm $\|\nabla_\theta^2 l(\theta, \xi_n)\|_F \leq L$, $\forall \theta, n$. Then, the robust loss also follows $\|\nabla_\theta^2 R(\theta)\|_F \leq M$ for some positive $M < \infty$.

The proof follows by differentiating the expression for the gradient $\nabla_\theta R(\theta)$ in the proof sketch of Proposition 1, and analyzing each resultant term; refer to Ghosh et al. (2020) for details. Our convergence analysis can thus hold for $l(\theta, \xi)$ that are Lipschitz continuous (required by Theorem 3) and possess Lipschitz gradients, but may otherwise be non-convex. This is often satisfied by common statistical learning losses, e.g., log-logistic and squared losses of linear models over compact spaces, thus allowing our algorithm to be used in important cases when $l(\cdot, \xi_n)$ are non-convex, such as training deep learning models.

A key consideration, given termination at $t = \mathscr{T}$, is then to obtain $\theta_{\mathscr{T}}$ as close as possible to the minimizer θ_{rob} that attains R_{inf}. The tradeoff in (5) suggests that increasing M_t aggressively will lead to

smaller gradients at termination, but this will also increase the per-iteration computational cost. We therefore seek *good* values for the step-length γ and the sample growth sequence $\{M_t\}$ that obtain an optimal balance between the added computational burden of each iteration and the expected reduction in the optimality gap (as represented by the norm of the gradient of $R(\theta_{\mathscr{T}})$). Such analysis requires establishing the rate at which the deterministic error inherent in the (full-data) optimization problem drops, which is well characterized for the case of strong-convex loss functions $l(\theta,\xi)$. This motivates our study of the corresponding tradeoff in Theorem 7 under Assumption 2(iv), which we will show yields the strong convexity of $R(\theta)$. Extending these results to the convex and non-convex cases are subjects of our ongoing research.

Our notion of efficiency will be developed w.r.t. the total computational effort W_t that is expended up until iterate t, which is the sum of the amount of individual work w_s in each iterate $s \leq t$. From the discussion following Proposition 2, we have $w_t = O(M_t \log M_t)$. Define $\nu_t := M_{t+1}/M_t$ as the *growth factor* of the sequence $\{M_t\}$, and in this notation $\mathscr{T} = \inf\{t : N = M_0 \prod_{s=1}^{t} \nu_s\}$. Our final result below characterizes the rate at which the expected optimality gap $E_{t+1} := \mathbb{E}_t[R(\theta_{t+1})] - R(\theta_{\text{rob}})$ decreases as $M_t \nearrow N$.

Theorem 7 Suppose all the conditions of Theorem 5 are satisfied and Assumption 2(iv) holds. Then the function $R(\theta)$ is c-strongly convex. Further suppose that $\gamma \leq \min\{\frac{1}{4L}, 4c\}$ and let $r = 1 - \frac{\gamma}{4c}$. We also have:
(i) If $M_t = M_0 \nu^t$ with parameter $1 < \nu < r^{-1/(1-\delta)}$, then $W_t E_{t+1} \leq K_1 t \nu^{t\delta}$ for $t \leq \mathscr{T}$ and a constant K_1;
(ii) If $M_t = M_0 \nu^t$ with parameter $\nu \geq r^{-1/(1-\delta)}$, then $W_t E_{t+1} \leq K_2 t (r\nu)^t$ for $t \leq \mathscr{T}$ and a constant K_2; and
(iii) If $M_t = M_0 (\prod_s \nu_s)$ where $\nu_s \searrow 1$ as $s \uparrow$, then $E_{t+1} = o(W_t^{-1})$.

Recall that δ defines the parameter η_t in Theorem 3. The proof of Theorem 7 in Ghosh et al. (2020) includes exact expressions for the constants such as K_1, K_2. It starts by establishing under the stated assumptions that if the full batch-gradient method is applied in each iteration ($M_t = N$), then a strongly-convex objective R would enjoy a linear (i.e., constant factor) reduction of size r in the error $R(\theta_t) - R(\theta_{\text{rob}})$ for a step-size γ chosen to satisfy the conditions of Theorem 7. The average optimality gap can be written, like (5), as a sum of this deterministic error and an additional term representing the stochastic error induced by the subsampling of the support, all expressed in terms of γ, ν_s and the other parameters of the algorithm.

Theorem 7 considers two important cases of subsample support size growth. The first two parts consider *constant*-factor growth with $\nu_t = \nu > 1$, $\forall t$, namely geometric or exponential growth of the sample size, with $\mathscr{T} = \log(N/M_0)/\log \nu$ total iterations. They show that constant factor sequences can attain a good trade off between the rate of reduction in stochastic error and the drop in deterministic error. Specifically, parameter values $\nu \in (1, r^{-1/(1-\delta)})$ in Theorem 7(i) produce the best balance between the optimality gap E_{t+1} and the computational effort W_t, with the upper bound on their product growing at the slowest of the three cases at a near-linear rate w.r.t. the iteration count t (recall δ is arbitrarily small). In contrast, if $\nu \geq r^{-1/(1-\delta)}$ in Theorem 7(ii), the deterministic error now drops slower than the stochastic error, and such that the product $E_{t+1} W_t$ escapes to infinity at a geometric rate w.r.t. t which increases with ν. Thus, a small growth-factor geometrically increasing sample size sequence may yield the best performance. However, since the parameter r depends on c and L, the growth-factor bound is difficult to identify in practice.

Theorem 7(iii) studies slowly growing sequences where the growth factors ν_t are *diminishing* with $\nu_t \searrow 1$ as $t \uparrow$, e.g., the polynomial growth of $\nu_t = 1 + \frac{1}{t}$, leading to a much larger number $\mathscr{T}$ of iterations. The result establishes that any diminishing-factor growth of M_t will lead to the stochastic error decreasing to zero much slower than the geometric drop in the deterministic error. Consequently, the optimality gap E_{t+1} reduces suboptimally w.r.t. the total computational effort W_t.

3 EXPERIMENTAL RESULTS

Numerous experiments were conducted to empirically evaluate our progressively sampled subgradient descent (Progressive) Algorithm 1 in solving the DRO formulation (1) against the full-support gradient (Full-grad.) algorithm of Namkoong and Duchi (2017) and the multi-level Monte Carlo (Giles) algorithm of Levy et al. (2020) and Ghosh and Squillante (2020). We also consider in Fig. 1(center) the standard SGD (fixed minibatch $M_t = M$) method to gauge the impact of the bias shown in Theorem 3. To summarize, these

experiments support our theoretical results and show that Progressive produces equal or better performance than all the methods considered herein with significantly less computational effort, orders of magnitude less effort in many cases, without the burdensome fine tuning of parameters required by these other methods.

Binary Classification Formulation. We compare the algorithms over statistically training binary classification models that seek to correctly determine the class $y = \pm 1$ of any input data point x with d features. Following Namkoong and Duchi (2017), all examples use a logistic binary classification loss $l(\theta,(x,y)) = \log(1+\exp(-y\theta^t x))$, where the training samples ξ consists of (x,y) pairs. Our experimental results are based on 8 publicly available binary classification datasets (listed in Table 1) that were obtained from UCI* (Dua, D. and C. Graff 2017), OpenML† (Feurer et al. 2019), and SKLearn‡ (Lewis et al. 2004) with sizes ranging from $O(10^2)$ to $O(10^6)$. For instance, the *HIV-1 Protease Cleavage* dataset (hiv1) predicts whether the HIV-1 protease will cleave a protein sequence in its central position ($y = 1$) or not ($y = -1$). This dataset has $N = 5830$ samples of $d = 160$ feature vectors, of which 991 are cleaved and 4839 non-cleaved.

Table 1: Comparison of the DRO methods and regularized ERM over 8 publicly available datasets.

Dataset	$\sqrt{\frac{d}{N}}$	**Test Misclassified (%)**				**CPU Time (secs)**			
		Full-grad.	Giles	Progressive	ERM	Full-grad.	Giles	Progressive	ERM
adult*	0.051	17.1±0.0	16.7±0.1	16.6±0.1	16.7±0.1	45	214	36	2542
gina_prior†	0.475	13.7±0.4	14.3±1.0	**12.7±0.3**	14.6±0.7	34	38	31	1147
hiv1*	0.166	5.9±0.1	6.3±0.2	5.8±0.0	6.1±0.1	41	45	35	1012
IMDB.drama†	0.091	36.1±0.1	37.0±0.1	36.2±0.0	36.2±0.1	176	865	89	19436
la1s.wc†	2.029	9.3±0.0	8.3±0.3	8.2±0.1	9.0±0.1	17	47	12	2456
OVA_Breast†	2.660	3.2±0.1	3.8±0.4	**3.0±0.1**	3.4±0.2	140	23	37	4310
rcv1‡	0.242	5.7±0.0	5.3±0.0	**5.1±0.0**	6.3±0.0	2628	1271	543	701843
riccardo†	0.463	4.9±0.1	2.0±0.1	**1.5±0.1**	1.7±0.1	259	201	120	86575

The uncertainty radius ρ for the set of measures $\mathscr{P}$ forms a key paremeter in (1). Blanchet et al. (2019) and Namkoong and Duchi (2017) provide as a broad guideline that $\rho = O(\sqrt{d/N})$ for binary classification with logistic models. Table 1 includes the quantity $\sqrt{d/N}$ for each dataset. The experiments are therefore based on setting ρ to be the same order of magnitude as $\sqrt{d/N}$ for all DRO methods.

In statistical learning, a practical heuristic to improve the classification performance of ERM-fitted models on unseen data (namely, generalization) is regularization, where the ERM loss objective is augmented with a penalty term $\lambda\|\theta\|_2^2$ and a fixed-batch SGD (size 10) is used to solve the formulation. An appropriate value for the penalty coefficient λ may sufficiently regularize the chosen optimal model parameter to improve generalization. However, no clear theory exists that guides the choice of a value for λ, and in practice a heuristic called k-fold CV is used (Stone 1974). We therefore consider the 10-*fold CV* procedure that partitions the full training dataset into 10 equal parts and trains a *regularized* model over each dataset formed by holding out one part as the *validation* dataset. An enumeration over λ of the average misclassification performance of each of the 10 models on the held-out validation data is used to determine the best λ.

Discussion of Results. Table 1 presents a comparison across the 8 datasets of the test misclassification produced by the DRO and the regularized ERM algorithms at termination. (The parameter settings for each algorithm are detailed below.) The 95% CIs are calculated over 10 permutations of the datasets into training (80%) and testing (20%) sets, where every algorithm elicits a model over the training data for each permutation, which is subsequently scored with the percentage of data misclassified by the model on the testing set, presented in the middle columns of Table 1. For each dataset, the method that produces generalization error (within the specified CIs) that is clearly better than the rest is highlighted in bold. The rightmost columns in Table 1 provide the average CPU time in seconds over the 10 permutations.

Among the DRO methods, Progressive takes the least time to solve the problem for all datasets (with one exception) while providing the same or better quality of performance. As expected, Full-grad. takes more time to solve the problem than Progressive for all datasets. Giles takes more time to solve the problem than Progressive, by an order of magnitude in several cases, for all but the OVA_Breast dataset. These longer computation times are due in large part to the added variability experienced by the Giles method arising from the Monte Carlo randomization. This added variability is also evident in the consistently wider CIs of the misclassification errors for Giles, as well as in the wider range of CPU times with an isolated case terminating the fastest (OVA_breast) and several cases terminating the slowest among the DRO methods.

We also observe from the table that at least one DRO method – always including Progressive – produces models of equal or better quality as the regularized ERM formulation for all datasets, with Progressive providing a significant improvement over ERM in 6 of the 8 cases. Recall that the DRO methods provide this level of performance by solving a single instance of the DRO formulation (1), hence avoiding the burdensome 10-fold CV enumeration. The average time taken by the ERM 10-fold regularization in solving its formulation multiple times to identify the best regularization parameter λ exceeds that taken on average to solve the DRO formulation by 1 to 2 orders of magnitude. The DRO methods thus gain significant computational savings by eliminating the expensive hyper-parameter tuning step.

Figure 1(left) presents comparisons of the empirical DRO runs over time, where the sample paths of all 10 runs for each method are plotted along with their average shown in bold. The binary classification formulation is being solved for riccardo with $\rho = 0.1$, as per the $\sqrt{d/N}$ recommendation. The sample paths for Progressive (green) exhibit relatively low variability overall with fast initial reductions in $\hat{R}(\theta)$ leading to quicker convergence. In strong contrast, Giles (purple) exhibit much higher variability across the iterations, and are computationally more expensive due to the multiple inner maximizations solved in each iteration. Although Full-grad. (red) also exhibits low variability, it is more expensive computationally.

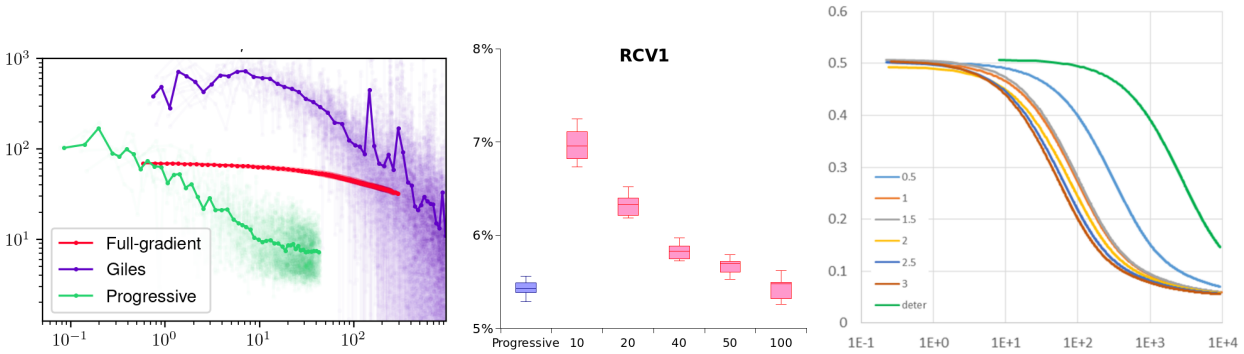

Figure 1: (left) Comparison of Progressive (green), Full-grad. (red), Giles (purple) on robust loss estimate $\hat{R}(\theta)$ (log-scale y-axis) vs. cumulative CPU time (log-scale x-axis) over riccardo for $\rho = 0.1$; (center) Comparison of Progressive (blue) and plain SGD (red) on percentage of misclassification vs. fixed minibatch sizes $M_t = M$ in testing over rcv1; (right) Comparison of Full-grad. (green) and Progressive on fraction of misclassification vs. cumulative CPU time (log-scale x-axis) over rcv1 for $\nu = 1.001$ and various γ.

Theorem 3 has important implications for DRO solvers, providing the relationship between the bias suffered by a fixed batch size SGD ($M_t = M$) method in solving (1) and the batch size M. The important impact of this bias is illustrated by Fig. 1(center), for the dataset rcv1. The test misclassification results for the standard SGD method at termination exhibits for small mini-batch sizes significant bias in comparison to Progressive, with the bias vanishing only as $M_t \to 100$. Recall that this bias arises because a small fixed subsample size in each iteration can easily miss those elements ξ of the training dataset that suffer from high loss $l(\theta_t, \xi)$ at the current iterate θ_t, hence yielding an optimistic estimate of the robust loss $R(\theta_t)$ and prematurely terminating the search for a robust solution to (1). Under a small growth factor of $\nu = 1.001$, Progressive iterations initially enjoy the benefits of fast objective value reduction similar to

standard SGD, but then eventually eliminates the introduced bias. Progressive therefore avoids the expense of the hyper-parameter tuning of the batch size of standard SGD for bias reduction.

Broadly, one expects the Progressive parameters of sample size growth factor ν and step length γ to impact its performance. (Theorem 7 establishes a bound on them for the iterates to converge geometrically with computational effort.) For the rcv1 dataset, Figure 1(right) contrasts the average test misclassification performance over cumulative CPU time (secs) for Progressive while keeping $\nu = 1.001$ fixed and varying γ; in addition, the figure provides the Full-grad. method (green). Although γ has some effect, these results show that Progressive is relatively insensitive to the chosen step length beyond $\gamma = 0.75$. Analogous results in Ghosh et al. (2020) show that Progressive is also insensitive to ν for values smaller than 1.001.

Algorithm Parameters. All of the algorithms sampled the initial θ_0 uniformly from the hypercube $[-1,1]^d$. Each DRO algorithm solved the inner-maximization formulation to within ε-accuracy where $\varepsilon = 10^{-7}$. The methods assemble and monitor the (robust or ERM) loss objective estimate and stop if the loss does not improve more than 1% in comparison with the average of the previous 10 such evaluations. All experiments were implemented in Python 3.7 and run on a 16-core 2.6GHz Intel Xeon processor with 128GB memory.

The Progressive DRO Algorithm 1 uses a *geometrically* growing sample size sequence with initial sample size $M_0 = 1$ and constant-growth factor $\nu = 1.001$, along with fixed step length $\gamma = 0.75$ in (2). The γ is chosen as per the results of Fig. 1(right), and ν is fixed based on analogous results in Ghosh et al. (2020). Although the parameter δ appears prominently in the inflation of ρ to individual ρ_t (Theorem 3), δ only needs to be a small positive constant; since the results are not sensitive to δ, we set $\delta = 0.01$. The Full-grad. DRO algorithm determines step lengths of each iteration using the LBFGS-B algorithm with a maximum of 0.5, which our experiments over many datasets show to be the best choice. The Giles DRO algorithm uses the geometric Giles randomizer to determine mini-batch sizes with a minimum size of 5, chosen after careful study over multiple datasets, and a stepsize sequence of $\gamma_t = 0.5 * (5000/(5000+t))$ (replacing γ in (2)) for all experiments. We refer to Ghosh et al. (2020) for additional plots and tables of such experimental results, as well as complex interactions with the DRO parameter ρ.

The Regularized ERM algorithm finds the optimal λ by enumerating *average* misclassification performance over the 10 held-out validation datasets on a grid of 20 points in the range $[10^{-6}, 10^{6}]$, starting with 10^6 and backtracking until the performance does not improve for 3 λ enumerations. Note that the computational benefits of Progressive would be even larger if all λ values over all grid points were enumerated.

REFERENCES

Barton, R. R., B. L. Nelson, and W. Xie. 2014. "Quantifying Input Uncertainty via Simulation Confidence Intervals". *INFORMS Journal on Computing* 26(1):74–87.

Ben-Tal, A., D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. 2013. "Robust Solutions of Optimization Problems Affected by Uncertain Probabilities". *Management Science* 59(2):341–357.

Blanchet, J., Y. Kang, and K. Murthy. 2019. "Robust Wasserstein Profile Inference and Applications to Machine Learning". *Journal of Applied Probability* 56(3):830–857.

Blanchet, J. H., and P. W. Glynn. 2015. "Unbiased Monte Carlo for Optimization and Functions of Expectations via Multi-Level Randomization". In *Proceedings of the 2015 Winter Simulation Conference (WSC)*, edited by L. Yilmaz, V. Chan, I. Moon, T. Roeder, C. Macal, and M. Rossetti, 3656–3667. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bottou, L., F. E. Curtis, and J. Nocedal. 2018. "Optimization Methods for Large-Scale Machine Learning". *SIAM Review* 60(2):223–311.

Dua, D. and C. Graff 2017. "UCI Machine Learning Repository". http://archive.ics.uci.edu/ml. Accessed: Aug 2nd.

Esfahani, P., and D. Kuhn. 2018. "Data-Driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations". *Math. Programming* 171:115–166.

Feurer, M., J. N. van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Mueller, J. Vanschoren, and F. Hutter. 2019. "OpenML-Python: An Extensible Python API for OpenML". *arXiv* 1911.02490.

Ghosh, S., and M. S. Squillante. 2020. "Unbiased Gradient Estimation for Distributionally Robust Learning". *arXiv e-prints* 2012.12367.

Ghosh, S., M. S. Squillante, and E. Wollega. 2020. "Efficient Stochastic Gradient Descent for Learning with Distributionally Robust Optimization". *arXiv e-prints* 1805.08728.

Giles, M. B. 2008. "Multilevel Monte Carlo Path Simulation". *Operations Research* 56(3):607–617.

Jacod, J., and P. Protter. 2004. *Probability Essentials*. Universitext. Springer Berlin Heidelberg.

Lam, H. 2019. "Recovering Best Statistical Guarantees via the Empirical Divergence-Based Distributionally Robust Optimization". *Operations Research* 67(4):1090–1105.

Levy, D., Y. Carmon, J. C. Duchi, and A. Sidford. 2020. "Large-Scale Methods for Distributionally Robust Optimization". In *Advances in Neural Information Processing Systems*, Volume 33, 8847–8860.

Lewis, D., Y. Yang, T. Rose, and F. Li. 2004. "RCV1: A New Benchmark Collection for Text Categorization Research". *Journal of Machine Learning Research* 5:361–397.

Luenberger, D. G. 1969. *Optimization by Vector Space Methods*. John Wiley & Sons.

Namkoong, H., and J. C. Duchi. 2016. "Stochastic Gradient Methods for Distributionally Robust Optimization with f-Divergences". In *Advances in Neural Information Processing Systems 29*, 2208–2216.

Namkoong, H., and J. C. Duchi. 2017. "Variance-based Regularization with Convex Objectives". In *Advances in Neural Information Processing Systems 30*, 2971–2980.

Owen, A. B. 2001. *Empirical Likelihood*. Chapman & Hall.

Shapiro, A. 1985. "Second-Order Derivatives of Extremal-Value Functions and Optimality Conditions for Semi-Infinite Programs". *Mathematics of Operations Research* 10:207–219.

Shapiro, A. 2003. "Monte Carlo Sampling Methods". In *Stochastic Programming*, Volume 10 of *Handbooks in Operations Research and Management Sciences*, 353 – 425. Elsevier.

Shapiro, A., D. Dentcheva, and A. Ruszczynski. 2009. *Lectures on Stochastic Programming Modeling and Theory*. Philadelphia: SIAM.

Sinha, A., H. Namkoong, and J. Duchi. 2018. "Certifying Some Distributional Robustness with Principled Adversarial Training". In *2018 International Conference on Learning Representations*.

Stone, M. 1974. "Cross-Validatory Choice And Assessment Of Statistical Predictions". *Journal of the Royal Statistical Society* 36:111–147.

Wilks, S. S. 1962. *Mathematical Statistics*. John Wiley & Sons.

AUTHOR BIOGRAPHIES

SOUMYADIP GHOSH is a member of IBM Research in the mathematics of AI division at Yorktown Heights. His research focuses on stochastic optimization and decision making under uncertainty, with applications in training large statistical learning models with improved generalization in speech recognition and natural language processing, and previously on Smarter Grid and energy analytics, cloud infrastructure, and supply chain management. He is an associate editor of Stochastic Models and has served as a committee member for INFORMS / WSC. He holds a PhD from Cornell University (Ithaca, NY, USA), an MS from Univ of Michigan (Ann Arbor, MI, USA) and a B Tech from Indian Institute of Technology (Chennai, TN, INDIA). His email address is ghoshs@us.ibm.com. Further details can be found at the URL https://researcher.watson.ibm.com/researcher/view.php?person=us-ghoshs.

MARK S. SQUILLANTE is a Distinguished Research Staff Member and the Manager of Foundations of Probability, Dynamics and Control within Mathematical Sciences at IBM Research (Yorktown Heights, NY, USA). His research interests broadly concern mathematical foundations of the analysis, modeling and optimization of the design and control of stochastic systems, and their applications across a wide range of domains in computing, communications, science, engineering, and business. He is an elected Fellow of ACM, IEEE, INFORMS and AIAA, and currently serves as Editor-in-Chief of Stochastic Models. He received a PhD from the University of Washington (Seattle, WA, USA). His email address is mss@us.ibm.com. Further details can be found at the URL https://researcher.watson.ibm.com/researcher/view.php?person=us-mss.

EBISA WOLLEGA is an associate professor of engineering and the director of industrial engineering bachelor of science program at Colorado State University Pueblo. His research areas include large scale optimization and artificial intelligence algorithms applied to energy systems, healthcare, retail operations. He is an INFORMS member and has served as a board member at multiple levels in Industrial Engineering Division of the American Society for Engineering Education. He received a PhD and an MS from the University of Oklahoma in industrial engineering; BS in industrial engineering from Mekelle University, Ethiopia. His email address is ebisa.wollega@csupueblo.edu. Further details can be found at the URL https://www.csupueblo.edu/profile/ebisa-wollega/index.html.