An Online Method for A Class of Distributionally Robust Optimization with Non-convex Objectives

Qi Qi*', Zhishuai Guo*', Yi Xu", Rong Ji", Tianbao Yang'
*Equal Contribution
'Department of Computer Science, The University of Iowa
"Alibaba DAMO Academy

Content

Distributionally Robust Optimization

- Problem Formulation:

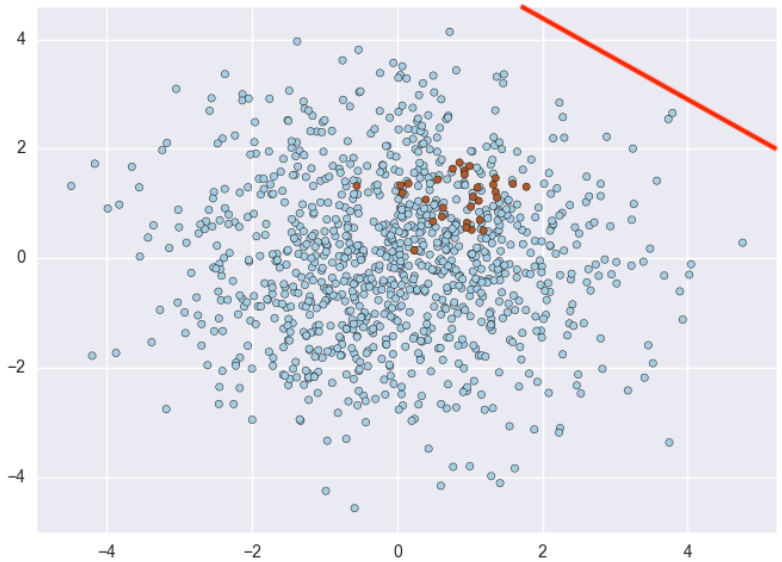
$$\min_{w \in R^d} \max_{p \in \Delta_n} \left\{ F_p(w) = \sum_{i=1}^{n} p_i \ell(w; z_i) - h(p, 1/n) + r(w) \right\}, \quad (1)$$

1. where $\Delta_n = \{p \in R^n : \sum_i p_i = 1, p_i \geq 0\}$.
2. $\ell(w; z)$: denotes a loss function on data $z = (x, y_I) \sim \mathcal{D}$,
3. $h(p, 1/n)$: a divergence measure
4. $r(w)$ is convex regularizer of w

Non-Convex Concave Min-Max Optimization

Applications in Deep Learning

- Data Imbalance (Rahimian, H., Mehrotra, S. (2019).)



- Label Noise (Li, Tian, et al. (2020).)

Optimization Methods

Primal-Dual algorithms:

- PG-SMD (Rafique et al. 2018), $O(1/\epsilon^4)$
- Epoch Primal-Dual SGD (Yan et al. 2020), $O(1/\epsilon^4)$
- Stoc-AGD (Yang et al. 2020)
  - $O(1/\mu^2\epsilon)$ by leveraging PL Condition
- PE-SGDA (Guo et al. 2020)
  - $O(1/\mu^2\epsilon)$ by leveraging PL Condition

Deficiencies:

- $O(n)$ computational cost per variable updates.

Note: Primal-Dual are quite slow compared with SGD for DL, it required additional computational cost for the dual variable p, which is depending on the data size n.

From Min-Max to Compositional Problem

- Min-Max Formulation:

$$\min_{w\in R^d}\max_{p\in\Delta_n}\left\{F_p(w)=\sum_{i=1}^{n}p_i\ell(w;z_i)-\lambda\sum_i p_i\log(np_i)+r(w)\right\},\quad(2)$$

where $\Delta_n=\{p\in R^n:\sum_i p_i=1,p_i\geq 0\}$

- Compositional Minimization Formulation

Online Setting

$$\min_{w\in R^d}\left\{F_{dro}(w)=\lambda\log\left(\mathbb{E}_z\exp\left(\ell(w;z)/\lambda\right)\right)+r(w)\right\}.\quad(3)$$

- Belongs to a genereal family of problems:

$$\min_{w\in R^d} f(\mathbb{E}_\xi[g(w,\xi)])+r(w)$$

Compositional Problem

Optimization Algorithms for Problem

- ASC-PG (Wang et al. 2017)
  - $O(1/\epsilon^{4.5})$
  - Polynomially decay stepsize.
- CIVR (Zhang et al. 2019)
  - $O(1/\epsilon^3)$ and $\widetilde{O}(1/\mu\epsilon)$ by leveraging PL condition.
  - Large minibatch size required $O(1/\epsilon)$

Question: *Can we design a better algorithm that is independent of data size n, is more practical for deep neural network training, and also has faster convergence rates? Yes*

Contribution

We propose a practical online algorithm, RECOVER, for solving a class of non-convex distributionally robust optimization (DRO) objectives:

Theory Contribution

It achieves:

1. General non-convex: $O(1/\epsilon^3)$
2. Polyak-Lojasiewicz (PL) condition: $O(1/\mu\epsilon)$

Best Rates

Lower Bounds

without any requirements on large mini batch or reference points.

Contribution

- Algorithm Comparison

Style	Algorithms	NC-SM	PL Condition	Batch	Geo η
Primal-Dual	Stoc-AGDA	-	$O(1/\mu^2\epsilon)$	$O(1)$	x
	PG-SMD2	$O(n/\epsilon^2 + 1/\epsilon^4)$	-	$O(n)$	x
	PE-SGDA	-	$O(1/\mu^2\epsilon)$	$O(1)$	✓
Compositional	ASC-PG	$O(1/\epsilon^{4.5})$	-	$O(1)$	x
	RCIVR	$O(1/\epsilon^3)$	$O(1/\mu\epsilon \log(1/\epsilon))$	$O(1/\epsilon)$	x
	RECOVER	$O(1/\epsilon^3)$	$O(1/\mu\epsilon)$	$O(1)$	✓

where *NC-SM* denotes the non-convex smooth objective without PL Condition.

Contribution

Practical Perspective

1. RECOVER is a duality-free algorithm that much faster than SOTA primal dual algorithms.
2. RECOVER resembles the practical stochastic Nesterov's method in several perspectives that are widely used for learning deep neural networks

Content

Assumptions

Assumption 1. *Let C_f, L_f, C_g and L_g be positive constants. Assume that*

(a) $f : \mathbb{R}^p \to \mathbb{R}$ *is a* C_f*-Lipschitz function and its gradient* ∇f *is* L_f*-Lipschitz.*

(b) $g_{\mathbf{z}} : \mathbb{R}^d \to \mathbb{R}^p$ *satisfies* $\mathbb{E}\|g_{\mathbf{z}}(\mathbf{w}_1) - g_{\mathbf{z}}(\mathbf{w}_2)\|^2] \leq C_g^2\|\mathbf{w}_1 - \mathbf{w}_2\|^2$ *for any* $\mathbf{w}_1, \mathbf{w}_2$ *and its Jacobian* $\nabla g_{\mathbf{z}}$ *satisfies* $\mathbb{E}[\|\nabla g_{\mathbf{z}}(\mathbf{w}_1) - \nabla g_{\mathbf{z}}(\mathbf{w}_2)\|^2] \leq L_g^2\|\mathbf{w}_1 - \mathbf{w}_2\|^2$.

(c) $r : R^d \to \mathbb{R} \cup \{\infty\}$ *is a convex and lower-semicontinuous function.*

(d) $F_* = \inf_{\mathbf{w}} F(\mathbf{w}) \geq -\infty$ *and* $F(\mathbf{w}_1) - F_* \leq \Delta_F$ *for some initial solution* $\mathbf{w}_1$.

Assumption 2. *Let* σ_g *and* $\sigma_{g'}$ *be positive constants and* $\sigma^2 = \sigma_g^2 + \sigma_{g'}^2$. *Assume that*

$$\mathbb{E}_{\mathbf{z}}[\|g_{\mathbf{z}}(\mathbf{w}) - g(\mathbf{w})\|^2] \leq \sigma_g^2, \ \mathbb{E}_{\mathbf{z}}[\|\nabla g_{\mathbf{z}}(\mathbf{w}) - \nabla g(\mathbf{w})\|^2] \leq \sigma_{g'}^2.$$

Assumption 3. $F(\mathbf{w})$ *satisfies* μ*-PL condition if there exists* $\mu > 0$ *such that*

PL Condition

$$2\mu(F(\mathbf{w}) - \min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w})) \leq \|\nabla F(\mathbf{w})\|^2.$$

Algorithm: COVER

Algorithm 1: COVER $(\mathbf{w}_0, \mathbf{u}_0, \mathbf{v}_0, \{\eta_t\}, T, \text{PL} = \text{False})$

Let $a_t = c\eta_t^2$
if not PL **then**
  Draw a samples $\mathbf{z}$ and construct the estimates: $\mathbf{u}_0 = g_{\mathbf{z}}(\mathbf{w}_0),\ \mathbf{v}_0 = \nabla g_{\mathbf{z}}(\mathbf{w}_0)$
end if
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{w}_{t+1} \leftarrow \mathbf{prox}_r^{\eta_t}(\mathbf{w}_t - \eta_t \mathbf{v}_t^\top \nabla f(\mathbf{u}_t))$
  Draw a samples $\mathbf{z}_{t+1}$, and update

$$\mathbf{u}_{t+1} = g_{\mathbf{z}_{t+1}}(\mathbf{w}_{t+1}) + (1 - a_{t+1})(\mathbf{u}_t - g_{\mathbf{z}_{t+1}}(\mathbf{w}_t))$$
$$\mathbf{v}_{t+1} = \nabla g_{\mathbf{z}_{t+1}}(\mathbf{w}_{t+1}) + (1 - a_{t+1})(\mathbf{v}_t - \nabla g_{\mathbf{z}_{t+1}}(\mathbf{w}_t))$$

end for
Return: $(\mathbf{w}_\tau, \mathbf{u}_\tau, \mathbf{v}_\tau)$ for randomly selected $\tau \in \{1, \ldots, T\}$.

Theorem 2. *Assume the Assumption 1 and 2, for any* $C > 0$, $k = \frac{C\sigma^{2/3}}{L}$, $c = 128L + \sigma^2/(7Lk^3)$, $w = \max((16Lk^3), 2\sigma^2, (\frac{ck}{4L})^3)$, *and* $\eta_t = k/(w + \sigma^2 t)^{1/3}$. *The output of COVER satisfies*

$$\mathbb{E}[\|\mathcal{G}_{\eta_{t^*}}(\mathbf{w}_{t_*})\|^2] \leq \widetilde{O}\left(\frac{\Delta_F}{T^{2/3}} + \frac{\sigma^2}{T^{2/3}}\right). \tag{5}$$

where t_* *is sampled from* $\{1, \ldots, T\}$.

Key Steps of the Algorithm COVER

$$\min_{\mathrm{w}} f(\mathbb{E}_{\xi}[g(\mathrm{w};\xi)])$$

$$\begin{aligned}
\mathrm{u}_{t+1} &= g_{\xi}(w_{t+1}) + (1 - a_{t+1})(u_t - g_{\xi}(w_t)) \\
\mathrm{v}_{t+1} &= \nabla g_{\xi}(w_{t+1}) + (1 - a_{t+1})(v_t - \nabla g_{\xi}(w_t)) \\
\mathrm{w}_{t+1} &= \mathrm{w}_t - \eta \mathrm{v}_{t+1}^{\top} \nabla f(u_{t+1})
\end{aligned}$$

Algorithm: RECOVER

- Stagewise Restarting COVER Algorithm

Algorithm 2: RECOVER$(\mathbf{w}_0, \epsilon_0, c)$

1: **Initialization**: Draw a sample ξ_0 and construct the estimates $\mathbf{u}_0 = g_{\xi_0}(\mathbf{w}_0)$, $\mathbf{v}_0 = \nabla g_{\xi_0}(\mathbf{w}_0)$
2: **for** $k = 1, \ldots, K$ **do**
3: $(\mathbf{w}_k, \mathbf{u}_k, \mathbf{v}_k) =$ COVER$(\mathbf{w}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}, \eta_k, T_k, \text{True})$
4: change η_k, T_k according to Theorem 3
5: **end for**
6: **Return:** $\mathbf{w}_K$

Geometrically Decreases

Lower Bounds of Sample Complexity

Main Theorem: RECOVER

By setting $b = O(1)$, $\eta_k = \frac{\sqrt{\mu\epsilon_k}}{25L^2\sigma}$, $T_k = \max\{\frac{L^2\sigma}{\mu^{3/2}\sqrt{\epsilon_k}}, \frac{L^2\sigma^2}{\mu\epsilon_k}\}$, in order to have $\mathbb{E}[F(w_K) - F_*] \leq \epsilon$. The sample complexity is $O(\frac{1}{\mu\epsilon})$.

Constant Mini-batch Size

Theoretical Verification of PL Assumption

Assumption 3. *$F(\mathbf{w})$ satisfies μ-PL condition if there exists $\mu > 0$ such that*

PL Condition

$$2\mu(F(\mathbf{w}) - \min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w})) \leq \|\nabla F(\mathbf{w})\|^2.$$

Lemma 1. *Let $F_{\mathbf{p}}(\mathbf{w}) = \sum_{i=1}^n p_i \ell(\mathbf{w}; \mathbf{z}_i)$. If for any $\mathbf{p} \in \Delta_n$, $F_{\mathbf{p}}(\mathbf{w})$ satisfies a μ-PL condition, then $F_{dro}(\mathbf{w}) = \lambda \log(\frac{1}{n}\sum_i \exp(\ell(\mathbf{w}; \mathbf{z}_i)/\lambda))$ satisfies μ-PL condition.*

Lemma 2. *Assume that input $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ satisfies $\|\mathbf{x}_i\| = 1$ and $\|\mathbf{x}_i - \mathbf{x}_j\| \geq \delta$, where $y_i \in \mathbb{R}^d$. Consider a deep neural network with $h_{i,0} = \phi(A\mathbf{x}_i), h_{i,l} = \phi(W_l h_{i,l-1}), l = 1, \ldots, L, \hat{y}_i = B h_{i,L}$ where $W_l \in \mathbb{R}^{d'\times d'}$, ϕ is the ReLU activation function, and $\ell(W; \mathbf{z}_i) = (\hat{y}_i - y_i)^2$ is a square loss. Suppose that for any W, $p_i^* = \exp(\ell(W; \mathbf{z}_i)/\lambda)/\sum_{i=1}^n \exp(\ell(W; \mathbf{z}_i)/\lambda) \geq p_0 > 0$, then with a high probability over randomness of W_0, A, B for every W with $\|W - W_0\| \leq O(1/poly(n, L, p_0^{-1}, \delta^{-1})$, there exists a small $\mu > 0$ such that $\|\nabla F_{dro}(W)\|_F^2 \geq \mu(F_{dro}(W) - \min_W F_{dro}(W))$.*

Content

Three Experiments

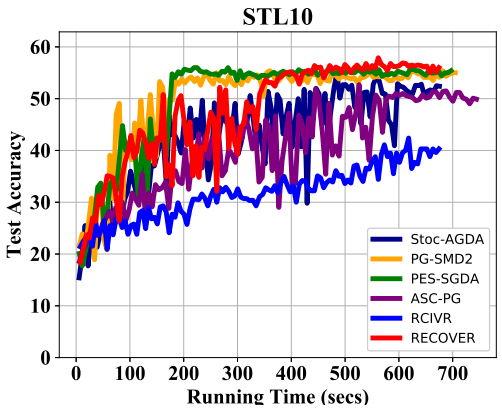
- Compare RECOVER with five State-Of-The-Art (SOTA) baselines from two categories: (i) primal-dual algorithms for solving the primal-dual formulation of DRO, and (ii) algorithms that are designed for the stochastic compositional formulation of DRO.
- Verify the advantages of DRO over Emperical Risk Minimization (ERM) for imbalanced data problems
- Show the RECOVER is also an effective fine-tuning algorithm for large-scale imbalanced data training.

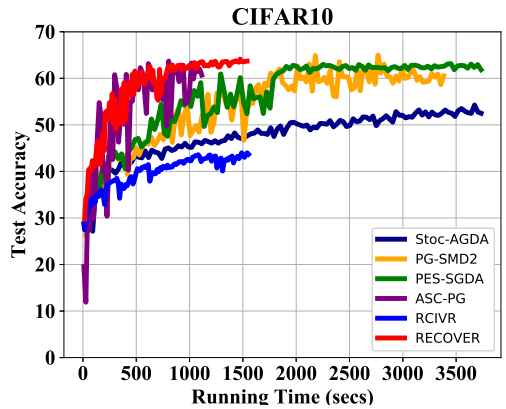
Compare RECOVER with SOTA Baselines

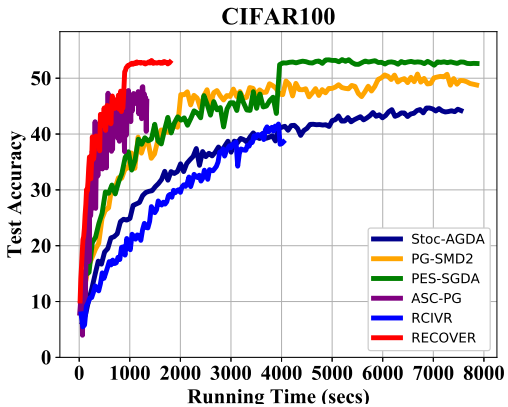
- Evaluation: **Testing accuracy**, **GPU time** and **sample complexity** .
- $\lambda = 5$
- $\ell(w, z)$ cross entropy loss.
- Datasets (Imbalanced Multi-Classification Tasks):

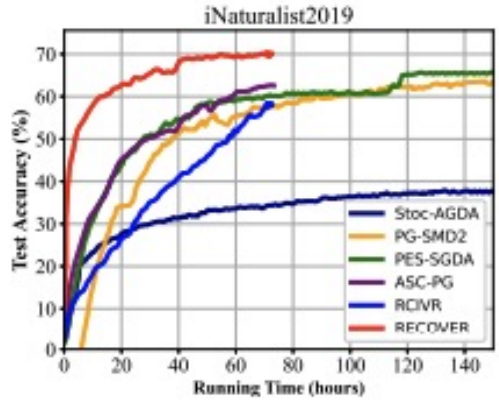
	First Half	Last Half	batch	Classes	Size	Network Arch
STL10	100	500	32	10	5000	Resnet20
Clifar10	100	5000	128	10	50000	Resnet20
CIFAR100	100	500	128	100	50000	Resnet20
iNaturalist 2019	Practical Imbalanced Datatset		64	1010	265,213	Inception-V3

Testing Accuracy vs Running Time









Conclusion and Observation:

1. RECOVER is much faster than primal-dual algorithms while achieving comparable results on compositional results.
2. It could save days of training times on iNaturalist2019.

Testing Accuracy vs # of processed training example

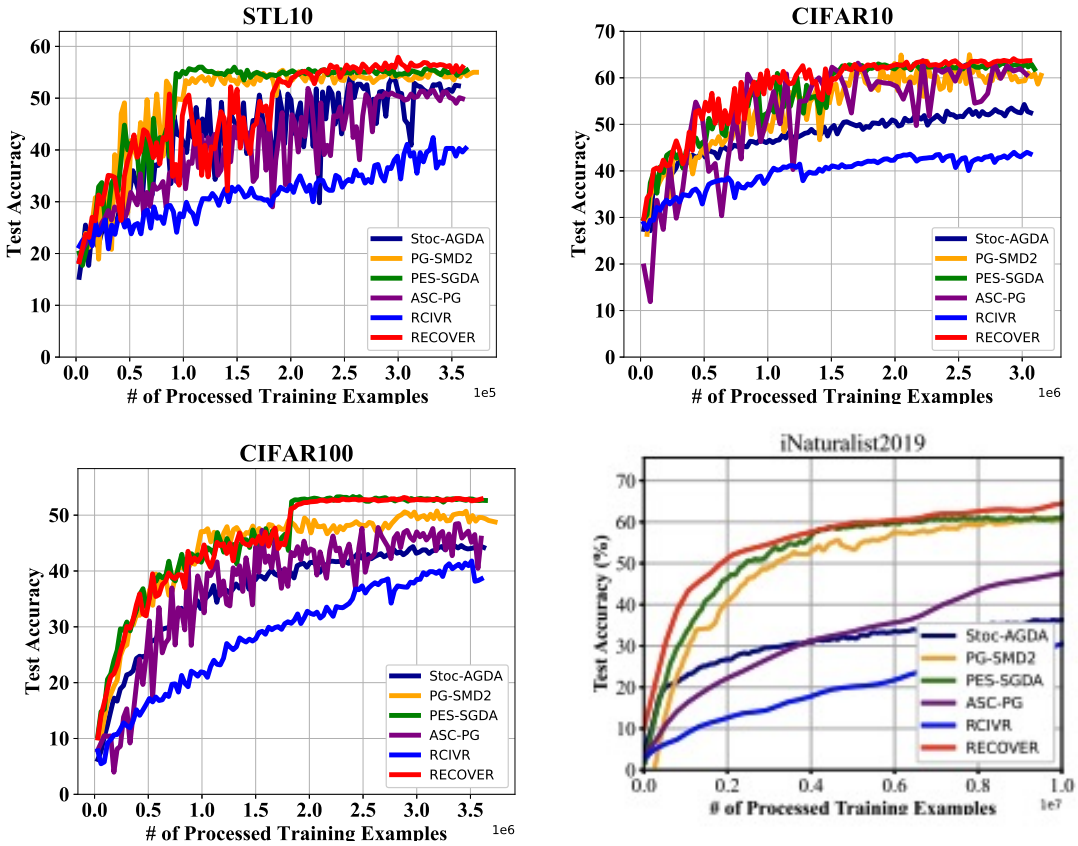


Figure: Testing accuracy vs # of processed training examples

Conclusion and Observation:

1. RECOVER has the same number of samples with other algorithms.

DRO with RECOVER vs ERM with SGD

- We compare the test accuracy learned by optimizing DRO using RECOVER and optimizing ERM using SGD on the imbalanced datasets: STL10, CIFAR10, CIFAR100, with four imbalance ratio $\rho = \{0.02, 0.05, 0.1, 0.2\}$.

Table 2: Test accuracy (%), mean (variance), of SGD for ERM and RECOVER for DRO. Bold numbers represent better performance.

IMRATIO	STL10		CIFAR10		CIFAR100	
	SGD	RECOVER	SGD	RECOVER	SGD	RECOVER
0.02	37.97 (0.62)	**38.08 (0.35)**	65.36(0.41)	**66.14 (0.24)**	38.99 (0.39)	**39.45 (0.32)**
0.05	41.12 (0.89)	**42.68 (0.37)**	74.74 (0.51)	**75.90 (0.11)**	45.79 (0.48)	**44.47 (0.44)**
0.1	46.03 (0.93)	**48.94 (0.74)**	79.32 (0.18)	**80.93 (0.09)**	49.45 (0.25)	**50.84 (0.74)**
0.2	51.75 (1.31)	**56.06 (1.59)**	84.84 (0.27)	**85.93 (0.02)**	55.80 (0.55)	**56.90 (0.18)**

Imbalance Ratio: The number of positive samples vs the number of negative samples. The smaller the ratio, the harder the task.

Effective as Fine-tune Methods

- ImageNet-LT, Places-LT
- ResNet50, ResNet152 Pretrained models.

Table 3: Test accuracy (%) of finetuned models by different methods.

Model	ImageNet-LT	Places-LT
Pretrained	40.50	23.28
CE (SGD)	41.29 (3e-3)	27.47 (1e-3)
Focal (SGD)	41.10 (2e-2)	27.64 (6e-3)
DRO (RECOVER)	**42.30** (4e-4)	**28.75** (4e-5)

- Thanks & Questions