

# Efficient Mirror Descent Ascent Methods for Nonsmooth Minimax Problems

Feihu Huang, Xidong Wu, Heng Huang

Department of Electrical and Computer Engineering, University of Pittsburgh, USA



**NeurIPS | 2021**

Thirty-fifth Conference on Neural  
Information Processing Systems

# Outline

- Background
- Mirror Descent Algorithm
- Our Mirror Descent Ascent Methods
- Theoretical Results
- Experimental Results
- Conclusions

- Background
- Mirror Descent Algorithm
- Our Mirror Descent Ascent Methods
- Theoretical Results
- Experimental Results
- Conclusions

# Background

**Minimax optimization** recently has attracted increased interest due to advance in machine learning applications such as generative adversarial networks (GANs), robust neural networks training, fair learning and federated learning. For example, **distributionally robust federated learning** can be represented a minimax optimization problem,

$$\min_{w \in \Omega} \max_{p \in \Pi} \sum_{i=1}^n p_i \mathbb{E}_{\xi \sim \mathcal{D}_i} [f_i(w; \xi)] - \lambda \psi(p),$$

where  $p_i \in (0, 1)$  denotes the proportion of  $i$ -th device in the entire model, and  $f_i(w; \xi)$  is the loss function on  $i$ -th device, and  $\lambda > 0$  is a tuning parameter, and  $\psi(p)$  is a strongly-convex regularization. Here  $\Pi = \{p \in \mathbb{R}_+^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\}$  is a  $n$ -dimensional simplex, and  $\Omega \subseteq \mathbb{R}^d$  is a nonempty convex set.

# Background

So recently many methods have been developed to solve these minimax optimization problems. For example, the **gradient descent ascent** method and its variants have been widely studied.

Although recently many methods have been proposed to solve these minimax problems, they suffer from a **high gradient complexity** and only focus on some specific minimax problems.

Thus, in the paper, we propose a class of efficient **mirror descent ascent** methods for solving nonconvex-strongly-concave minimax problems with nonsmooth regularization.

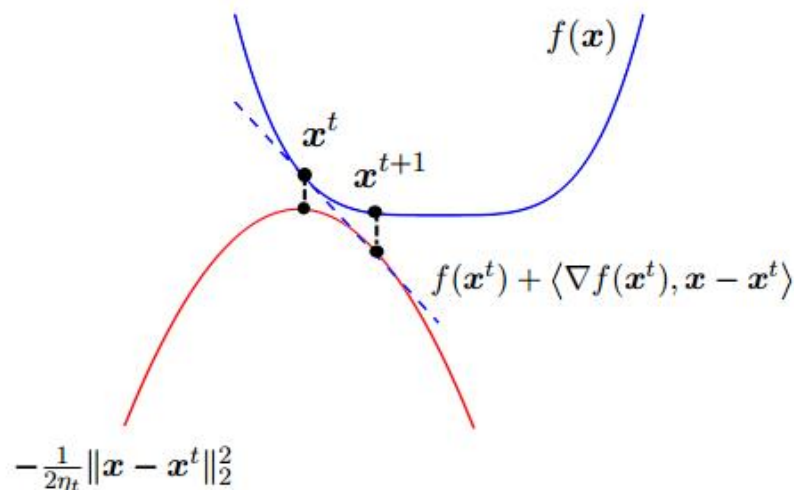
- Background
- **Mirror Descent Algorithm**
- Our Mirror Descent Ascent Methods
- Theoretical Results
- Experimental Results
- Conclusions

# Mirror Descent

## A proximal viewpoint to Projected Gradient Descent

$$\min_{x \in \mathcal{C}} f(x)$$

$$\mathbf{x}^{t+1} = \arg \min_{x \in \mathcal{C}} \left\{ \underbrace{f(\mathbf{x}^t) + \langle \nabla f(\mathbf{x}^t), \mathbf{x} - \mathbf{x}^t \rangle}_{\text{linear approximation}} + \underbrace{\frac{1}{2\eta_t} \|\mathbf{x} - \mathbf{x}^t\|_2^2}_{\text{proximity term}} \right\}$$

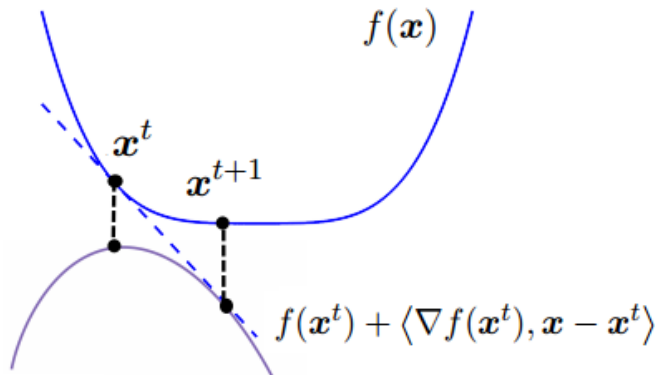


From Yuxin Chen' Slides

# Mirror Descent

This quadratic proximity term only can deal with some **homogeneous** local geometry of objective function, but not deal with some **inhomogeneous** or even **non-Euclidean** local geometry of objective function. Thus, the mirror descent method use the Bregman distance instead of Euclidean distance.

Replace the quadratic proximity  $\|x - x^t\|_2^2$  with distance-like metric  $D_\varphi$


$$x^{t+1} = \arg \min_{x \in \mathcal{C}} \left\{ f(x^t) + \langle \nabla f(x^t), x - x^t \rangle + \frac{1}{\eta_t} \underbrace{D_\varphi(x, x^t)}_{\text{Bregman divergence}} \right\}$$

where  $D_\varphi(x, z) := \varphi(x) - \varphi(z) - \langle \nabla \varphi(z), x - z \rangle$  for convex and differentiable  $\varphi$



- Background
- Mirror Descent Algorithm
- **Our Mirror Descent Ascent Methods**
- Theoretical Results
- Experimental Results
- Conclusions

# Mirror Descent Ascent Methods

In the paper, we study the following nonsmooth nonconvex-strongly-concave minimax problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} F(x, y) = \{f(x, y) + g(x) - h(y)\}, \quad (1)$$

where the function  $f(x, y) : \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}$  is smooth and possibly nonconvex in  $x \in \mathbb{R}^d$  and  $\mu$ -strongly concave in  $y \in \mathbb{R}^p$ , and the functions  $g(x)$  and  $h(y)$  are convex and possibly nonsmooth. Here the constraint sets  $\mathcal{X} \subseteq \mathbb{R}^d$  and  $\mathcal{Y} \subseteq \mathbb{R}^p$  are compact and convex. In many machine learning problems,  $f(x, y)$  generally represents loss function and is a stochastic form, *i.e.*,  $f(x, y) = \mathbb{E}_\xi[f(x, y; \xi)]$ , where the random variable  $\xi$  follows an unknown distribution. Here both  $g(x)$  and  $h(y)$  frequently denote the nonsmooth regularization such as  $g(x) = \lambda \|x\|_1$ .

---

## Algorithm 1 (Stochastic) Mirror Descent Ascent Algorithm

---

- 1: **Input:**  $T$ , stepsizes  $\{\gamma_t > 0, \lambda_t > 0, \eta_t \in (0, 1]\}_{t=1}^T$ , mini-batch size  $b$ ;
  - 2: **initialize:**  $x_1 \in \mathcal{X}$  and  $y_1 \in \mathcal{Y}$ ;
  - 3: **for**  $t = 1, 2, \dots, T$  **do**
  - 4:   **MDA:** Compute partial derivatives  $v_t = \nabla_x f(x_t, y_t)$  and  $w_t = \nabla_y f(x_t, y_t)$ ;
  - 5:   **SMDA:** Generate randomly mini-batch samples  $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$  with  $|\mathcal{B}_t| = b$ , and compute stochastic partial derivatives  $v_t = \nabla_x f_{\mathcal{B}_t}(x_t, y_t)$  and  $w_t = \nabla_y f_{\mathcal{B}_t}(x_t, y_t)$ ;
  - 6:   Given the mirror functions  $\psi_t$  and  $\phi_t$ ;
  - 7:    $x_{t+1} = \arg \min_{x \in \mathcal{X}} \{ \langle v_t, x \rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t) \}$ ;
  - 8:    $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$  where  $\tilde{y}_{t+1} = \arg \max_{y \in \mathcal{Y}} \{ \langle w_t, y \rangle - h(y) - \frac{1}{\lambda_t} D_{\phi_t}(y, y_t) \}$ ;
  - 9: **end for**
  - 10: **Output:**  $x_\zeta$  and  $y_\zeta$  chosen uniformly random from  $\{x_t, y_t\}_{t=1}^T$ .
-

# Mirror Descent Ascent Methods

In Algorithm [1](#), we use (stochastic) mirror descent to update the parameter  $x$ , and simultaneously use (stochastic) mirror ascent to update the parameter  $y$ . When choose the mirror functions  $\psi_t(x) = \frac{1}{2}\|x\|^2$  and  $\phi_t(y) = \frac{1}{2}\|y\|^2$  for all  $t \geq 1$ , we have  $D_{\psi_t}(x, x_t) = \frac{1}{2}\|x - x_t\|^2$  and  $D_{\phi_t}(y, y_t) = \frac{1}{2}\|y - y_t\|^2$ . Under this case, Algorithm [1](#) will reduce the standard (stochastic) proximal gradient descent ascent algorithm. When choose the mirror functions  $\psi_t(x) = \frac{1}{2}x^T H_t x$  and  $\phi_t(y) = \frac{1}{2}y^T G_t y$  for all  $t \geq 1$ , we have  $D_{\psi_t}(x, x_t) = \frac{1}{2}(x - x_t)^T H_t (x - x_t)$  and  $D_{\phi_t}(y, y_t) = \frac{1}{2}(y - y_t)^T G_t (y - y_t)$ , where  $H_t \succeq \rho I_d$  and  $G_t \succeq \rho I_p$ . For example, given  $\alpha \in (0, 1)$  and  $\rho > 0$ , we can generate the matrices  $H_t$  and  $G_t$  like as in Adam-type algorithms [\[20, 19\]](#), defined as

$$\tilde{v}_0 = 0, \tilde{v}_t = \alpha \tilde{v}_{t-1} + (1 - \alpha) \nabla_x f(x_t, y_t; \xi_t)^2, \quad H_t = \text{diag}(\sqrt{\tilde{v}_t} + \rho), \quad t \geq 1 \quad (8)$$

$$\tilde{w}_0 = 0, \tilde{w}_t = \alpha \tilde{w}_{t-1} + (1 - \alpha) \nabla_y f(x_t, y_t; \xi_t)^2, \quad G_t = \text{diag}(\sqrt{\tilde{w}_t} + \rho), \quad t \geq 1 \quad (9)$$

Under this case, our SMDA algorithm will reduce a novel adaptive gradient descent ascent algorithm.

# Mirror Descent Ascent Methods

---

**Algorithm 2** Accelerated Stochastic Mirror Descent Ascent (VR-SMDA) Algorithm

---

- 1: **Input:**  $T, q$ , stepsizes  $\{\gamma_t > 0, \lambda_t > 0, \eta_t \in (0, 1]\}_{t=1}^T$ , mini-batch sizes  $b$  and  $b_1$ ;
- 2: **initialize:**  $x_1 \in \mathcal{X}$  and  $y_1 \in \mathcal{Y}$ ;
- 3: **for**  $t = 1, 2, \dots, T$  **do**
- 4:   **if**  $\text{mod}(t, q) = 0$  **then**
- 5:     Randomly generate mini-batch samples  $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$  with  $|\mathcal{B}_t| = b$ ;
- 6:     Compute stochastic partial derivatives  $v_t = \nabla_x f_{\mathcal{B}_t}(x_t, y_t)$  and  $w_t = \nabla_y f_{\mathcal{B}_t}(x_t, y_t)$ ;
- 7:   **else**
- 8:     Randomly generate mini-batch samples  $\mathcal{I}_t = \{\xi_t^i\}_{i=1}^{b_1}$  with  $|\mathcal{I}_t| = b_1$ ;
- 9:     Compute stochastic partial derivatives

$$v_t = \nabla_x f_{\mathcal{I}_t}(x_t, y_t) - \nabla_x f_{\mathcal{I}_t}(x_{t-1}, y_{t-1}) + v_{t-1}, \quad (14)$$

$$w_t = \nabla_y f_{\mathcal{I}_t}(x_t, y_t) - \nabla_y f_{\mathcal{I}_t}(x_{t-1}, y_{t-1}) + w_{t-1}; \quad (15)$$

- 10:   **end if**
  - 11:   Given the mirror functions  $\psi_t$  and  $\phi_t$ ;
  - 12:    $x_{t+1} = \arg \min_{x \in \mathcal{X}} \left\{ \langle v_t, x \rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t) \right\}$ ;
  - 13:    $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$  where  $\tilde{y}_{t+1} = \arg \max_{y \in \mathcal{Y}} \left\{ \langle w_t, y \rangle - h(y) - \frac{1}{\lambda_t} D_{\phi_t}(y, y_t) \right\}$ ;
  - 14: **end for**
  - 15: **Output:**  $x_\zeta$  and  $y_\zeta$  chosen uniformly random from  $\{x_t, y_t\}_{t=1}^T$ .
-

- Background
- Mirror Descent Algorithm
- Our Mirror Descent Ascent Methods
- **Theoretical Results**
- Experimental Results
- Conclusions

# Convergence Results

We first introduce a useful convergence metric  $\mathbb{E}\|\mathcal{G}_t\|$  to measure convergence properties of our algorithms as in [24]. Given the generated parameters  $x_t$  at  $t$ -th iteration by our algorithms, we define a gradient mapping as

$$\mathcal{G}_t = \frac{1}{\gamma_t}(x_t - x_{t+1}^+), \quad (18)$$

$$x_{t+1}^+ = \arg \min_{x \in \mathcal{X}} \left\{ \langle \nabla \Phi(x_t), x \rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t) \right\}, \quad (19)$$

where  $\Phi(x) = f(x, y^*(x)) - h(y^*(x)) = \max_{y \in \mathcal{Y}} \{f(x, y) - h(y)\}$ . When  $\mathcal{X} = \mathbb{R}^d$  and  $g(x)$  is a constant, and  $\psi_t(x) = \frac{1}{2}\|x\|^2$ , we have  $\mathcal{G}_t = \nabla \Phi(x) = \nabla_x f(x, y^*(x))$ . Under this case, our convergence metric  $\mathbb{E}\|\mathcal{G}_t\| = \mathbb{E}\|\nabla_x f(x, y^*(x))\|$  is a common convergence metric used in [26].

# Convergence Results

**Assumption 1.** (*Smoothness*) For our deterministic and mini-batch stochastic algorithms (MDA and SMDA), we assume that the function  $f(x, y)$  has an  $L_f$ -Lipschitz gradient, i.e., for all  $x_1, x_2 \in \mathcal{X}$  and  $y_1, y_2 \in \mathcal{Y}$ , we have

$$\|\nabla f(x_1, y_1) - \nabla f(x_2, y_2)\| \leq L_f \|(x_1, y_1) - (x_2, y_2)\|. \quad (4)$$

For our variance-reduced stochastic algorithm (VR-SMDA), we assume that each component function  $f(x, y; \xi)$  has an  $L_f$ -Lipschitz gradient, i.e., for all  $x_1, x_2 \in \mathcal{X}$  and  $y_1, y_2 \in \mathcal{Y}$ , we have

$$\|\nabla f(x_1, y_1; \xi) - \nabla f(x_2, y_2; \xi)\| \leq L_f \|(x_1, y_1) - (x_2, y_2)\|, \quad \forall \xi. \quad (5)$$

In Assumption 1, the inequality (4) is commonly used in the minimax optimization [26, 2, 4, 8]. While the inequality (5) is frequently used in the variance-reduced stochastic optimization [28, 17].

**Assumption 2.** Each component function  $f(x, y; \xi)$  has an unbiased stochastic gradient with bounded variance  $\sigma^2$ , i.e.,

$$\mathbb{E}[\nabla f(x, y; \xi)] = \nabla f(x, y), \quad \mathbb{E}\|\nabla f(x, y; \xi) - \nabla f(x, y)\|^2 \leq \sigma^2. \quad (6)$$

**Assumption 3.** The function  $f(x, y)$  is  $\mu$ -strongly concave w.r.t  $y$ , i.e., for all  $x \in \mathcal{X}$  and  $y_1, y_2 \in \mathcal{Y}$ , we have  $\|\nabla_y f(x, y_1) - \nabla_y f(x, y_2)\| \geq \mu\|y_1 - y_2\|$ . Then the following inequality holds

$$f(x, y_1) \leq f(x, y_2) + \langle \nabla_y f(x, y_2), y_1 - y_2 \rangle - \frac{\mu}{2} \|y_1 - y_2\|^2. \quad (7)$$



# Convergence Results

**Assumption 4.** *The functions  $g(x)$  and  $h(y)$  are convex but possibly nonsmooth.*

Assumption 3 shows that the function  $f(x, y)$  is  $\mu$ -strongly concave w.r.t  $y$ . Assumption 4 shows that the function  $h(y)$  is convex. Thus, the function  $\{f(x, y) - h(y)\}$  is strongly concave in  $y \in \mathcal{Y}$ , there exists a unique solution to the problem  $\max_{y \in \mathcal{Y}} \{f(x, y) - h(y)\}$  for any  $x$ . Let  $y^*(x) = \arg \max_{y \in \mathcal{Y}} \{f(x, y) - h(y)\}$ , and we define a function  $\Phi(x) = f(x, y^*(x)) - h(y^*(x)) = \max_{y \in \mathcal{Y}} \{f(x, y) - h(y)\}$ .

**Assumption 5.** *For any  $\alpha \in \mathbb{R}$ , the sub-level set  $\{x : \Phi(x) + g(x) \leq \alpha\}$  is compact. The function  $\Phi(x) + g(x)$  is bounded below in  $\mathcal{X}$ , i.e.,  $F^* = \inf_{x \in \mathcal{X}} \{\Phi(x) + g(x)\} > -\infty$ .*

Assumption 5 is frequently used in nonsmooth minimax optimization [8]. In fact, when  $h(y) = c$  where  $c$  is a constant, we can only assume the function  $\Phi(x) + g(x)$  is bounded below in  $\mathcal{X}$  instead of Assumption 5.



# Convergence Results

**Theorem 1.** Suppose the sequence  $\{x_t, y_t\}_{t=1}^T$  be generated from Algorithm [1](#) using stochastic partial derivatives (i.e., SMDA algorithm). Let  $0 < \eta = \eta_t \leq 1$ ,  $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$  and  $0 < \lambda = \lambda_t \leq \frac{1}{6L_f}$ , we have

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}} + \frac{4\sqrt{2}\Delta_1}{\sqrt{3T\gamma\rho}} + \frac{10\sigma}{\sqrt{3b\rho}} + \frac{20\sigma\sqrt{\eta\lambda}}{3\sqrt{\gamma\rho\mu b}}, \quad (20)$$

where  $\kappa = L_f/\mu$ ,  $L = L_f(1 + \kappa)$ ,  $\tilde{F}(x) = \Phi(x) + g(x)$  and  $\Delta_1 = \|y_1 - y^*(x_1)\|$ .

**Remark 1.** Given  $0 < \eta \leq 1$ ,  $\lambda = O(\frac{1}{L_f})$ ,  $\gamma = \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$  and  $\rho = O(L_f^\nu)$ , ( $\nu \geq 0$ ), we have  $\gamma = O(\kappa^{\nu-3})$  and  $\gamma\rho = O(\kappa^{2\nu-3})$ . Thus, our SMDA algorithm has a convergence rate of  $O(\sqrt{\frac{\kappa^{3-2\nu}}{T}} + \sqrt{\frac{\kappa^{-2\nu}}{b}} + \sqrt{\frac{\kappa^{3-2\nu}}{b}})$ . When let  $\nu = \frac{1}{2}$  (i.e.,  $\rho = O(\sqrt{L_f})$ ),  $b = T/\kappa$  and  $\sqrt{\frac{\kappa^2}{T}} = \epsilon/3$ , we have  $T = O(\kappa^2\epsilon^{-2})$  and  $b = O(\kappa\epsilon^{-2})$ . Since our SMDA algorithm requires  $b$  samples to estimate the stochastic partial directives  $v_t$  and  $w_t$  at each iteration, and needs  $T$  iterations, it has a sample complexity of  $bT = O(\kappa^3\epsilon^{-4})$  for finding an  $\epsilon$ -stationary point, the same complexity as in [\[25\]](#). When let  $\nu = 5/6$  (i.e.,  $\rho = L_f^{4/3}$ ),  $b = T/\kappa^{1/3}$  and  $\sqrt{\frac{\kappa^{1/3}}{T}} = \epsilon/3$ , we have  $T = O(\kappa^{1/3}\epsilon^{-2})$  and  $b = O(\epsilon^{-2})$ . Thus, our SMDA algorithm has a near optimal sample complexity of  $bT = O(\kappa^{1/3}\epsilon^{-4})$ , the same complexity as in [\[22\]](#).

# Convergence Results

**Theorem 2.** Suppose the sequence  $\{x_t, y_t\}_{t=1}^T$  be generated from Algorithm [1](#) using the deterministic partial derivatives (i.e., MDA algorithm). Let  $0 < \eta = \eta_t \leq 1$ ,  $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$  and  $0 < \lambda = \lambda_t \leq \frac{1}{6L_f}$ , we have

$$\frac{1}{T} \sum_{t=1}^T \|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}} + \frac{4\sqrt{2}\Delta_1}{\sqrt{3T\gamma\rho}}, \quad (25)$$

where  $\kappa = L_f/\mu$ ,  $L = L_f(1 + \kappa)$ ,  $\tilde{F}(x) = \Phi(x) + g(x)$  and  $\Delta_1 = \|y_1 - y^*(x_1)\|$ .

**Remark 2.** Without loss of generality, let  $L_f \geq \frac{1}{\mu}$ . Given  $0 < \eta \leq 1$ ,  $\lambda = O(\frac{1}{L_f})$ ,  $\gamma = \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$  and  $\rho = O(L_f^{(\frac{1}{2}+\nu)})$  ( $\nu \geq 0$ ), we have  $\frac{1}{\gamma\rho} = O(\kappa^{(2-2\nu)})$ . Under this case, our MDA algorithm has a sample complexity of  $T = O(\kappa^{(2-2\nu)}\epsilon^{-2})$  for finding an  $\epsilon$ -stationary point. When let  $\nu = 0$ , our MDA algorithm has a sample complexity of  $T = O(\kappa^2\epsilon^{-2})$ , the same complexity as in [\[4\]](#). When let  $\nu = 1/2$ , our MDA algorithm has a lower sample complexity of  $T = O(\kappa\epsilon^{-2})$  than the sample complexity of [\[4\]](#) [\[2\]](#). In solving the problem [\(1\)](#) without the nonsmooth regularization, when let  $\nu = 3/4$ , our MDA algorithm has a near optimal sample complexity of  $T = O(\sqrt{\kappa}\epsilon^{-2})$ , the same complexity as in [\[27\]](#).

# Convergence Results

**Theorem 3.** Suppose the sequence  $\{x_t, y_t\}_{t=1}^T$  be generated from Algorithm 2. Let  $b_1 = q$ ,  $0 < \eta = \eta_t \leq 1$ ,  $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{\eta\mu\lambda\rho}{38L_f^2}, \frac{3\rho}{19L_f^2\eta}, \frac{\rho\eta}{8}, \frac{9\rho\eta\mu\lambda}{400\kappa^2})$  and  $0 < \lambda = \lambda_t \leq \min(\frac{1}{6L_f}, \frac{9\mu}{100\eta^2L_f^2})$ , we have

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}} + \frac{4\sqrt{2}\Delta_1}{\sqrt{3T\gamma\rho}} + \frac{2\sqrt{2}\sigma}{\sqrt{\gamma\rho\eta bL_f}}, \quad (22)$$

where  $\kappa = L_f/\mu$ ,  $L = L_f(1 + \kappa)$ ,  $\tilde{F}(x) = \Phi(x) + g(x)$  and  $\Delta_1 = \|y_1 - y^*(x_1)\|$ .

**Remark 3.** Given  $0 < \eta \leq 1$ ,  $\lambda = O(\frac{1}{\kappa L_f})$ ,  $\gamma = \min(\frac{3\rho}{4L}, \frac{\eta\mu\lambda\rho}{38L_f^2}, \frac{3\rho}{19L_f^2\eta}, \frac{\rho\eta}{8}, \frac{9\rho\eta\mu\lambda}{400\kappa^2})$  and  $\rho = O(L_f^{1+\nu})$  ( $\nu \geq 0$ ), we have  $\frac{1}{\gamma\rho} = O(\kappa^{2-2\nu})$ . Thus, our VR-SMDA algorithm has a convergence rate of  $O(\sqrt{\frac{\kappa(2-2\nu)}{T}} + \sqrt{\frac{\kappa(1-2\nu)}{b}})$ . When let  $\nu = 0$  (i.e.,  $\rho = O(L_f)$ ),  $b = T/\kappa$  and  $\sqrt{\frac{\kappa^2}{T}} = \epsilon/2$ , we have  $T = O(\kappa^2\epsilon^{-2})$ . Further let  $b_1 = q = O(\kappa\epsilon^{-1})$  and  $b = O(\kappa\epsilon^{-2})$ . Since Algorithm 2 requires  $b$  samples to estimate the stochastic directives  $v_t$  and  $w_t$  at each iteration when  $\text{mod}(t, q) = 0$ , otherwise needs  $b_1$  samples, and need  $T$  iterations, it has a sample complexity of  $b_1T + bT/q = O(\kappa^3\epsilon^{-3})$  for finding an  $\epsilon$ -stationary point, the same complexity as in [27]. When  $\nu = 5/6$  (i.e.,  $\rho = L_f^{11/6}$ ),  $b = T/\kappa^{1/3}$  and  $\sqrt{\frac{\kappa^{1/3}}{T}} = \epsilon/2$ , we have  $T = O(\kappa^{1/3}\epsilon^{-2})$ . At the same time, let  $b_1 = q = O(\epsilon^{-1})$  and  $b = O(\epsilon^{-2})$ . Thus our VR-SMDA algorithm has a lower sample complexity of  $b_1T + bT/q = O(\kappa^{1/3}\epsilon^{-3})$ .

- Background
- Mirror Descent Algorithm
- Our Mirror Descent Ascent Methods
- Theoretical Results
- Experimental Results
- Conclusions

# Experimental Results

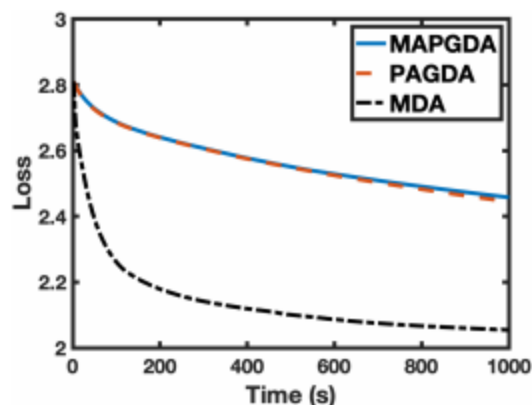
In this section, we perform two tasks (i.e., fair classifier and robust neural network training) to validate efficiency of our algorithms. Specifically, we conduct these tasks on the Fashion-MNIST dataset as in [33] as well MNIST dataset and CIFAR-10 dataset. Fashion-MNIST dataset and MNIST dataset consist of  $28 \times 28$  arrays of grayscale pixel images classified into 10 categories, and includes 60,000 training images and 10,000 testing images. CIFAR-10 dataset includes 60,000  $32 \times 32$  colour images (50,000 training images and 10,000 testing images). We compare our algorithms (MDA, SMDA and VR-SMDA) with the existing proximal gradient descent ascent algorithms (MAPGDA [2], PAGDA [4] and PASGDA [4]) for solving these nonsmooth nonconvex minimax problems. Note that the Proximal-GDA algorithm in [8] only is a non-accelerated version of MAPGDA algorithm [2], so we omit it in the comparison methods. The experiments are run on CPU machines with 2.3 GHz Intel Core i9 as well as NVIDIA Tesla P40 GPU.

# Experimental Results

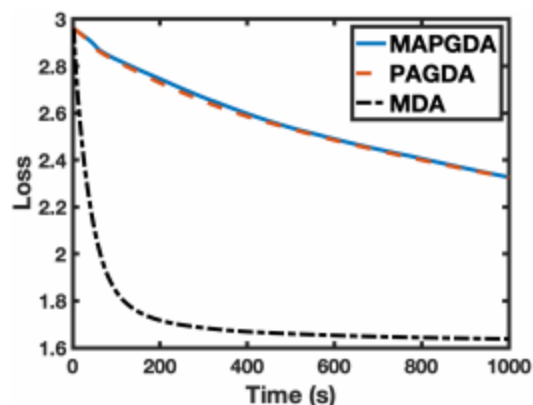
## 1) Fair Classification

$$\min_w \max_{y \in \mathcal{Y}} \left\{ \sum_{i=1}^3 y_i \mathcal{L}_i(w) + g(w) - h(y) \right\} \quad \text{s.t. } \mathcal{Y} = \left\{ y \mid y_i \geq 0, \sum_{i=1}^3 y_i = 1 \right\}, \quad (23)$$

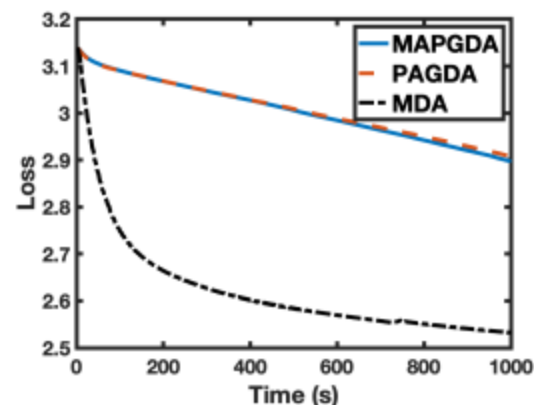
where  $w$  denotes the CNN model parameters, and  $\mathcal{L}_1$ ,  $\mathcal{L}_2$  and  $\mathcal{L}_3$  are the loss functions corresponding to the samples in three different categories. Here we let  $g(w) = \nu_1 \|w\|_1$  and  $h(y) = \nu_2 \|y\|_2^2$ , where



(a) Fashion-MNIST



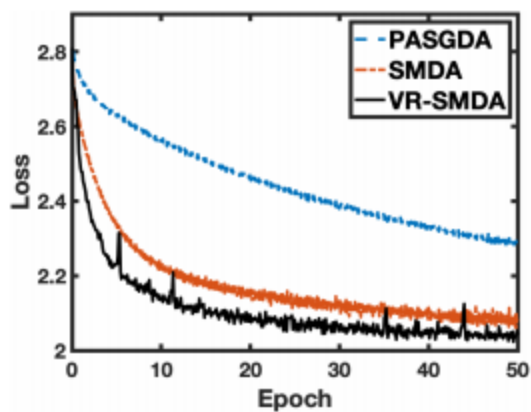
(b) MNIST



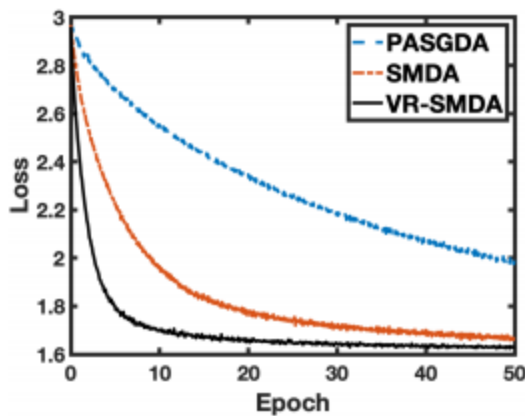
(c) CIFAR-10

Figure 1: Results of different deterministic methods on the fair classifier task.

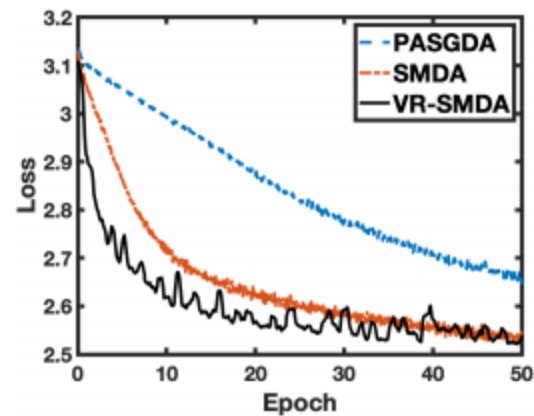
# Experimental Results



(a) Fashion-MNIST



(b) MNIST



(c) CIFAR-10

Figure 2: Results of different stochastic methods on the fair classifier task.

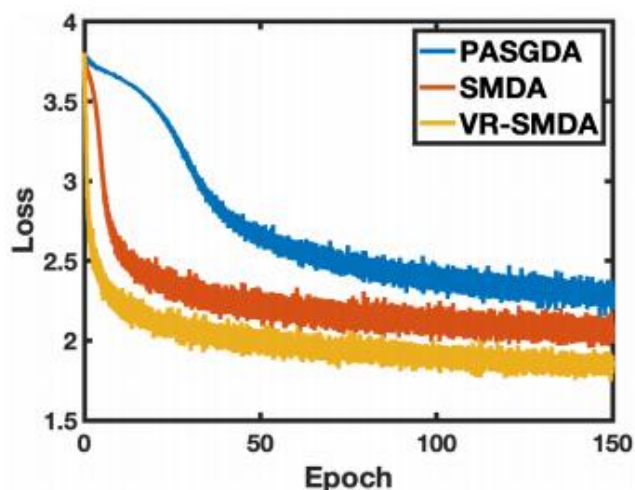


# Experimental Results

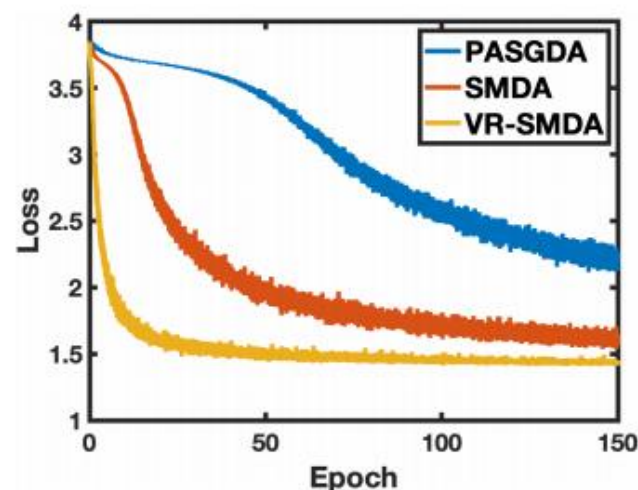
## 2) Robust Neural Network Training

$$\min_w \sum_{i=1}^n \max_{y_i \in \mathcal{Y}} \mathcal{L}(f(a_i + y_i; w), b_i), \quad \mathcal{Y} = \{y \mid \|y_i\|_\infty \leq \varepsilon, i \in [n]\} \quad (24)$$

where  $(a_i, b_i)$  denotes the  $i$ -th data point, and  $w$  is the parameter of NN, and  $y_i$  denotes is the perturbation added to the  $i$ -th data point. Following [33], we approximate the inner maximization



(a) Fashion-MNIST



(b) MNIST

Figure 3: Results of different stochastic methods on the robust NN training task at Fashion-MNIST and MNIST datasets.



- Background
- Mirror Descent Algorithm
- Our Mirror Descent Ascent Methods
- Theoretical Results
- Experimental Results
- Conclusions

# Conclusions

- 1) We proposed a class of efficient mirror descent ascent methods for solving non-smooth non-convex minimax problems;
- 2) We provided a convergence analysis framework for the proposed methods, and proved our methods reach a near optimal gradient (or sample) complexity than the existing methods.

**Thanks!**

**Q&A**