A Stochastic (Sub)gradient Method for Distributionally Robust and Risk-Averse Learning

Mert Gürbüzbalaban

Department of Management Science and Information Systems
Center for Theoretical Mathematics and Computer Science (DIMACS)
Department of Electrical and Computer Engineering (Affiliated)
Department of Statistics (Affiliated)

RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

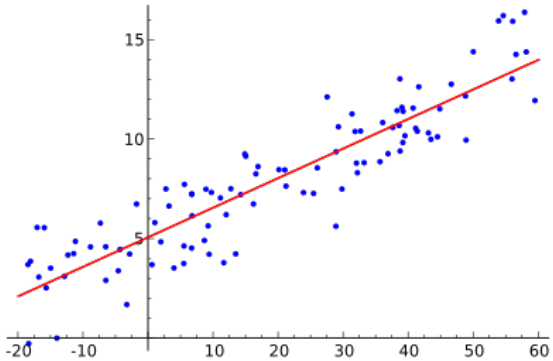
US-Mexico Optimization Workshop, January 9th, 2023
In honor of Steve Wright's 60th birthday

Optimization for Machine Learning

- Learning from labeled data: Risk minimization

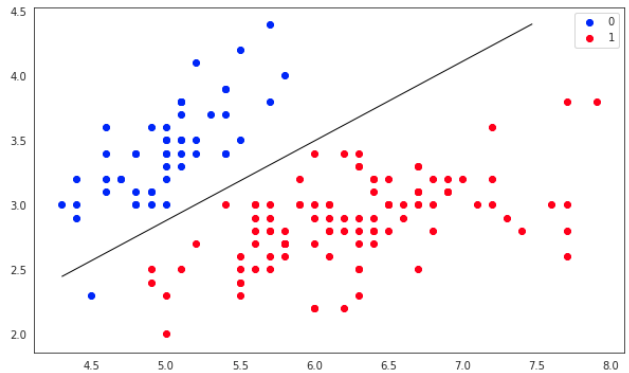
$$\min_{x \in X} \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)] \tag{1}$$

  where $D = (input, output)$ data, $x =$ model parameters.
- Classic examples:



(a) Linear regression: ℓ is convex & smooth

$$\ell(x, D) = (a^T x - b)^2$$
$$D = (a, b),\ X = \mathbb{R}^d.$$



(b) Classification with SVM: ℓ is convex & non-smooth

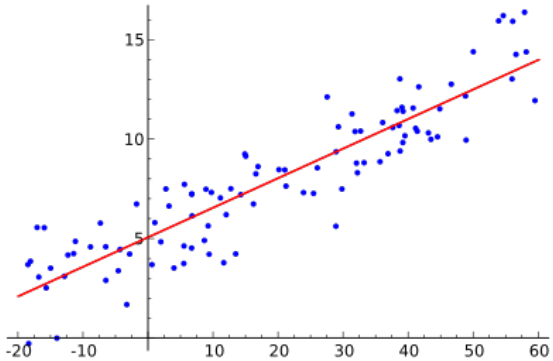
$$\ell(x, D) = \max(0, 1 - bx^T a) + \tau \|x\|^2$$

Optimization for Machine Learning

- Learning from labeled data: Risk minimization

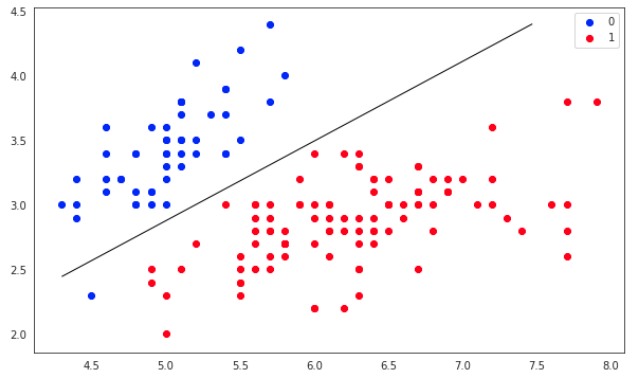
$$\min_{x \in X} \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)] \tag{1}$$

  where $D = (input, output)$ data, $x =$ model parameters.
- Classic examples:



(a) Linear regression: ℓ is convex & smooth

$$\ell(x, D) = (a^T x - b)^2$$
$$D = (a, b),\ X = \mathbb{R}^d.$$



(b) Classification with SVM: ℓ is convex & non-smooth

$$\ell(x, D) = \max(0, 1 - bx^T a) + \tau \|x\|^2$$

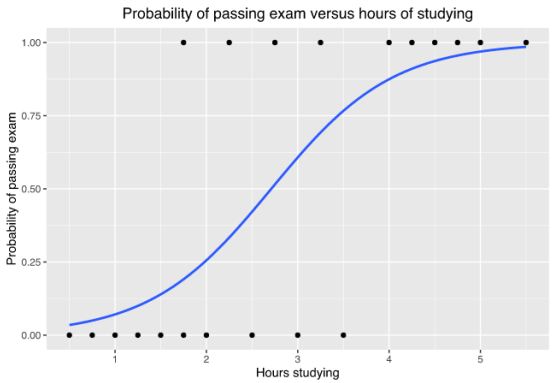
Optimization for Machine Learning

- Learning from labeled data: Risk minimization

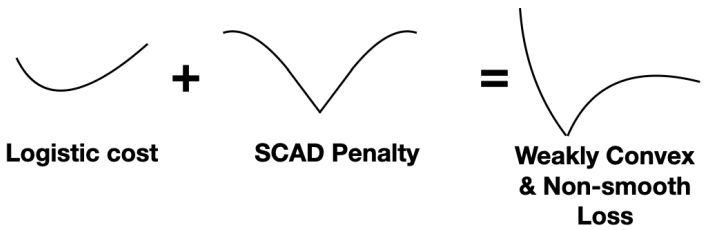
$$\min_{x \in X} \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)] \tag{2}$$

where $D = (input, output)$ data, $x =$ model parameters.

- Classic examples:



(a) SCAD-Regularized Logistic regression



(b) Loss is δ-weakly convex if loss $+ \delta\|x\|^2/2$ is convex.

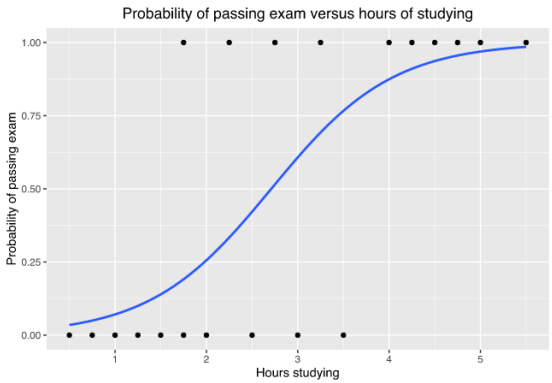
Optimization for Machine Learning

- Learning from labeled data: Risk minimization

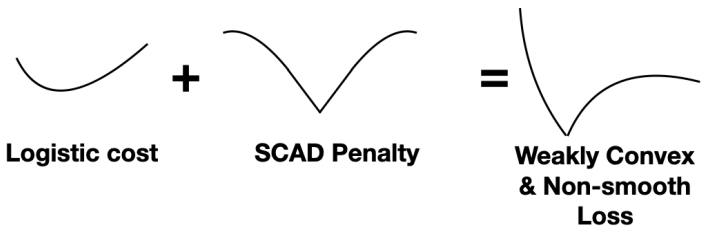
$$\min_{x \in X} \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)] \tag{2}$$

where $D = (input, output)$ data, $x =$ model parameters.

- Classic examples:

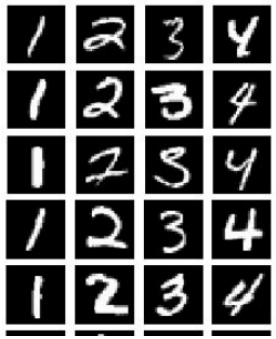


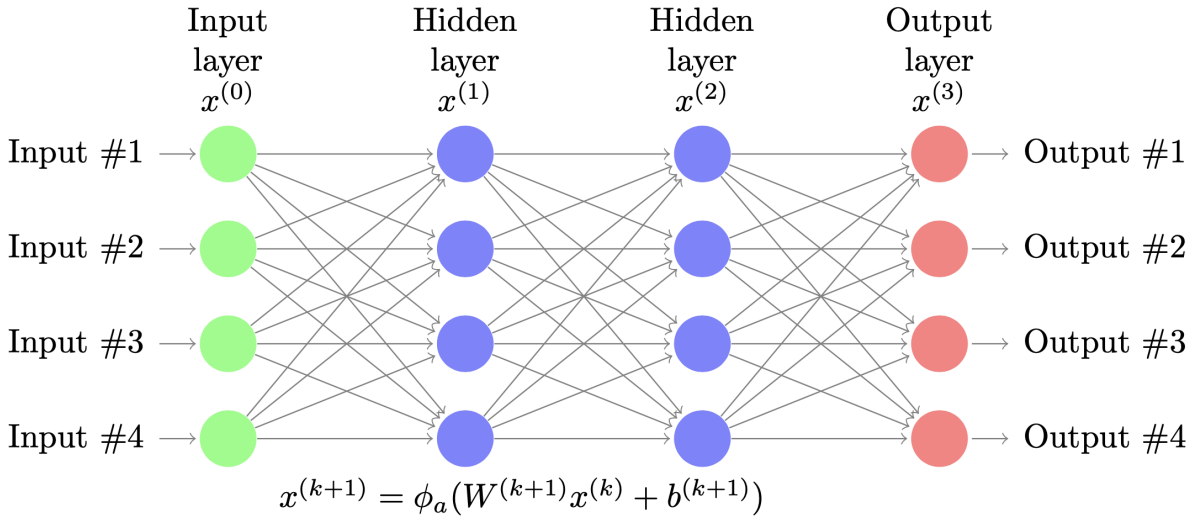
(a) SCAD-Regularized Logistic regression



(b) Loss is δ-weakly convex if loss $+ \delta\|x\|^2/2$ is convex.

Another example: Deep learning





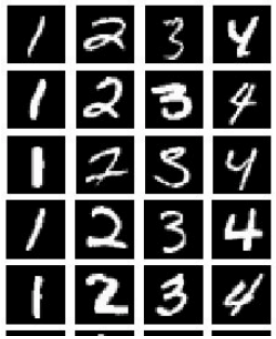
- Risk minimization:

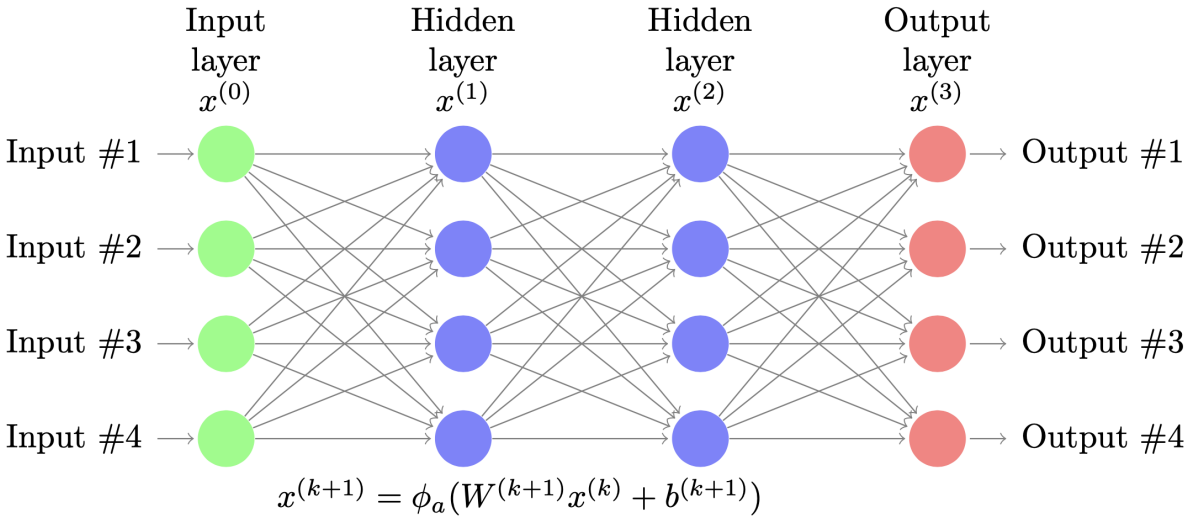
$$\min_{x \in X} f(x) := \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)]$$

where $D = (input, output)$, $x =$ network parameters $(\{W^{(k)}, b^{(k)}\}_k)$.

- Thresholding at every layer: Non-smooth or smooth
- Loss: Generalized (Norkin) differentiable or smooth

Another example: Deep learning





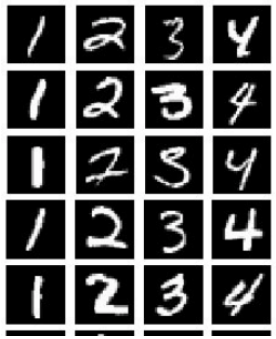
- Risk minimization:

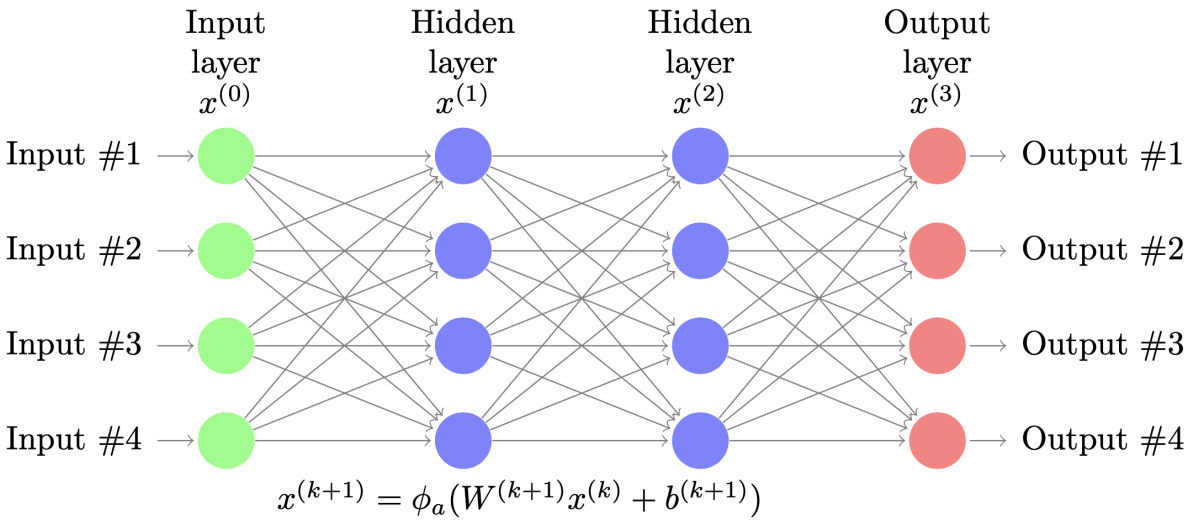
$$\min_{x \in X} f(x) := \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)]$$

where $D = (input, output)$, $x =$ network parameters $(\{W^{(k)}, b^{(k)}\}_k)$.

- Thresholding at every layer: Non-smooth or smooth
- Loss: Generalized (Norkin) differentiable or smooth

Another example: Deep learning

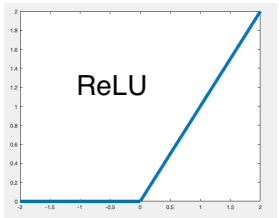


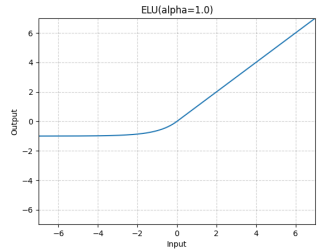


- Risk minimization:

$$\min_{x \in X} f(x) := \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)]$$

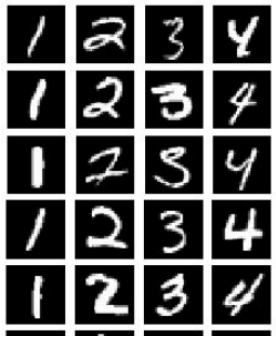
where $D = (input, output)$, $x =$ network parameters $(\{W^{(k)}, b^{(k)}\}_k)$.

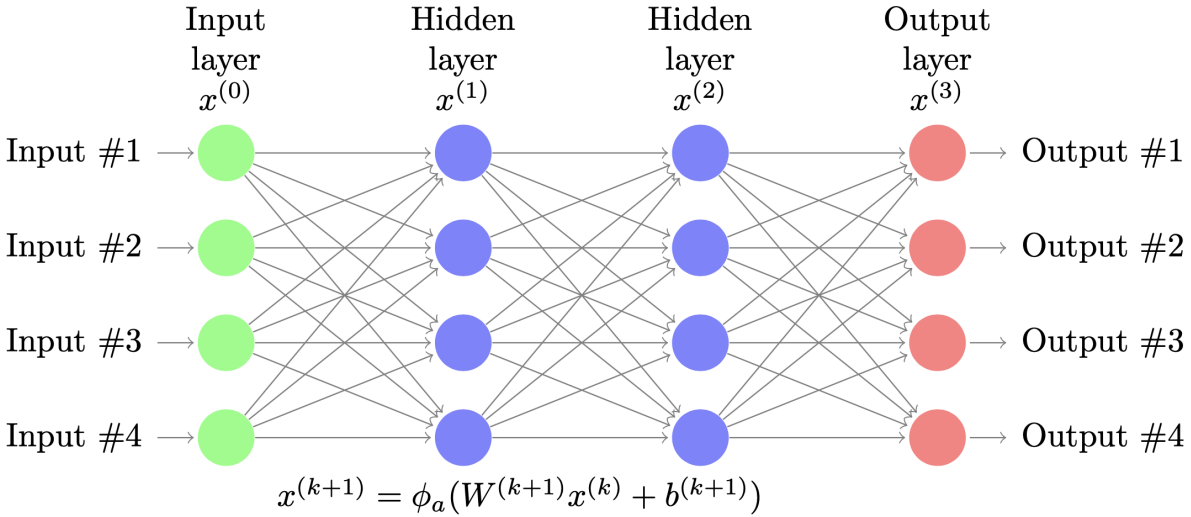
- Thresholding at every layer: Non-smooth 

or smooth 

- Loss: Generalized (Norkin) differentiable or smooth

Another example: Deep learning

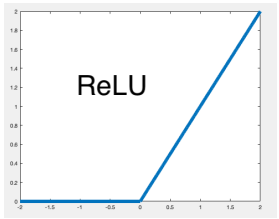


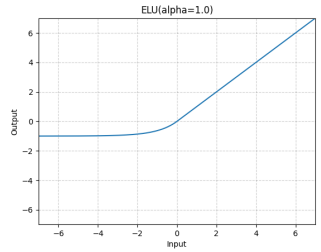


- Risk minimization:

$$\min_{x \in X} f(x) := \mathbb{E}_{D \sim \mathbb{P}}[\ell(x, D)]$$

where $D = (input, output)$, $x =$ network parameters $(\{W^{(k)}, b^{(k)}\}_k)$.

- Thresholding at every layer: Non-smooth 

or smooth 

- Loss: Generalized (Norkin) differentiable 
or smooth 

Deep Learning Applications

Figure: Computer Vision

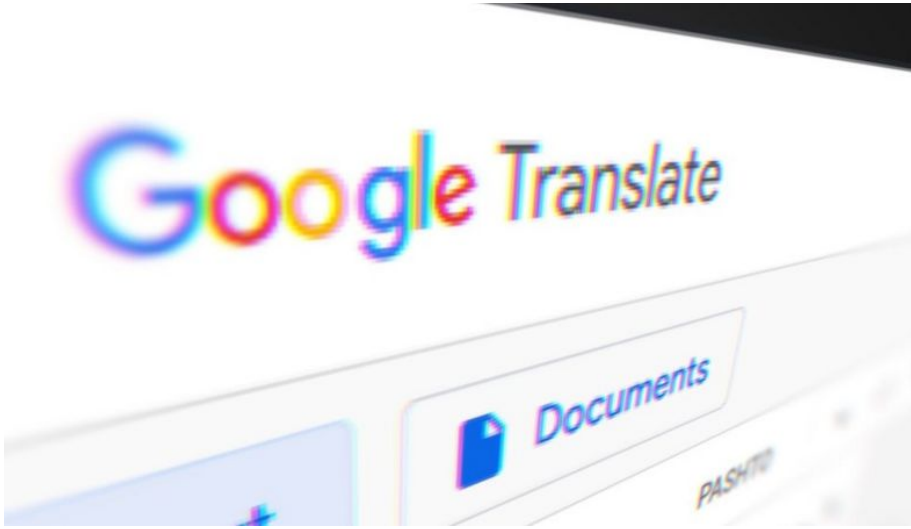


Figure: Machine Translation

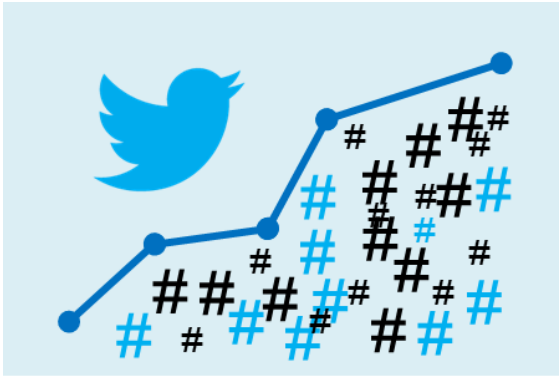


Figure: Predicting Social Media

AI Technique	Classifier	Accuracy	AUC	F1-Score
Machine learning	SVM	80.00%	–	–
Machine learning	SVM, RF	–	0.87	0.72
Machine learning	XGB	–	0.66	–
Deep learning	**CNNLSTM**	**92.30%**	**0.90**	**0.93**

Figure: Diagnosing Covid

Robustness to Statistical Changes in Input Data

- Risk minimization leads to fragile models.

Figure: Distributional shift in the input

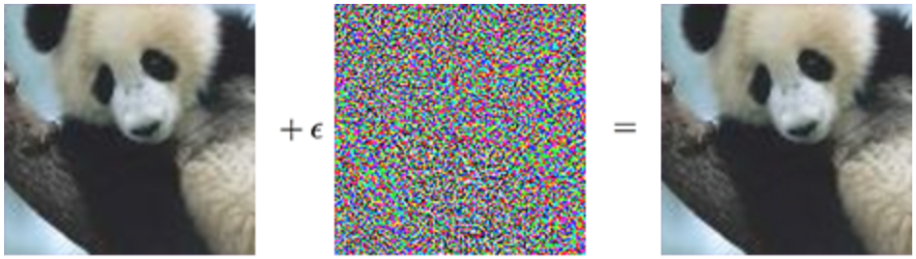


Figure: [Goodfellow et al. 2014] Robustness issue to attacks/perturbations

Distributionally robust statistical learning

- Ensuring distributional robustness:

$$\min_{x \in X} \max_{\mathbb{Q} \in \underbrace{\mathcal{M}(\mathbb{P})}_{\text{ambiguity set}}} \mathbb{E}_{D \sim \mathbb{Q}} \left[\ell(x, D) \right]$$

- Existing approaches to modelling $\mathcal{M}(\mathbb{P})$ include conditional value at risk [Takeda and Kanamori, 2009], f-divergence based sets [Duchi and Namkoong, 2018], Wasserstein distance/distance-based approaches [Ho-Nguyen & Wright, 2021], [Esfahani & Kuhn, 2018],[Gao & Kleywegt, 2016].
- $\mathcal{M}(\mathbb{P})$ is typically infinite-dimensional.

Distributionally robust statistical learning

- Ensuring distributional robustness:

$$\min_{x \in X} \max_{\mathbb{Q} \in \underbrace{\mathcal{M}(\mathbb{P})}_{\text{ambiguity set}}} \mathbb{E}_{D \sim \mathbb{Q}} \left[\ell(x, D) \right]$$

- Existing approaches to modelling $\mathcal{M}(\mathbb{P})$ include conditional value at risk [Takeda and Kanamori, 2009], f-divergence based sets [Duchi and Namkoong, 2018], Wasserstein distance/distance-based approaches [Ho-Nguyen & Wright, 2021], [Esfahani & Kuhn, 2018],[Gao & Kleywegt, 2016].
- $\mathcal{M}(\mathbb{P})$ is typically infinite-dimensional.

Existing work

Sample-based approximations to the ambiguity set: Finite-sum instead of expectation

- **Convex loss:** Finite-dimensional convex program formulations [Esfahani & Kuhn, 2018],[Abadeh et al., 2015], [Chen & Pashalidis 2018], bandit mirror descent Namkoong & Duchi, 2016], conic interior point solvers or gradient descent with backtracking Armijo line-searches [Duchi & Namkoong, 2021], convex and Lipschitz losses [Levy et al., 2020], SGD-based algorithm with $\mathcal{O}(1/\varepsilon^2)$ complexity for Lipschitz and smooth losses [Soma & Yoshida, 2020], SAPD alg. [Zhang et al., 2022],..
- **Smooth non-convex loss:** Wasserstein distance-based [Sinha et al. 2018], f-divergences/smooth Lipschitz losses [Jin et al. 2021], $\mathcal{O}(1/\varepsilon^6)$ complexity for smooth weakly convex losses [Zhang et al., 2022], (nonsmooth) weakly convex/strongly convex min-max approach of [Yan et al., 2020], CVaR-based approach with $\mathcal{O}(1/\varepsilon^6)$ complexity [Soma & Yoshida, 2020], ..
- **Non-smooth nonconvex loss:** For "zero-one loss" in linear classification, efficient algorithms for smoothed ramp loss [Ho-Nguyen, Wright, 2021].

For general non-smooth non-convex losses, no scalable algorithm with convergence guarantees to our knowledge.

Existing work

Sample-based approximations to the ambiguity set: Finite-sum instead of expectation

- **Convex loss:** Finite-dimensional convex program formulations [Esfahani & Kuhn, 2018],[Abadeh et al., 2015], [Chen & Pashalidis 2018], bandit mirror descent Namkoong & Duchi, 2016], conic interior point solvers or gradient descent with backtracking Armijo line-searches [Duchi & Namkoong, 2021], convex and Lipschitz losses [Levy et al., 2020], SGD-based algorithm with $\mathcal{O}(1/\varepsilon^2)$ complexity for Lipschitz and smooth losses [Soma & Yoshida, 2020], SAPD alg. [Zhang et al., 2022],..
- **Smooth non-convex loss:** Wasserstein distance-based [Sinha et al. 2018], f-divergences/smooth Lipschitz losses [Jin et al. 2021], $\mathcal{O}(1/\varepsilon^6)$ complexity for smooth weakly convex losses [Zhang et al., 2022], (nonsmooth) weakly convex/strongly convex min-max approach of [Yan et al., 2020], CVaR-based approach with $\mathcal{O}(1/\varepsilon^6)$ complexity [Soma & Yoshida, 2020], ..
- **Non-smooth nonconvex loss:** For "zero-one loss" in linear classification, efficient algorithms for smoothed ramp loss [Ho-Nguyen, Wright, 2021].

For general non-smooth non-convex losses, no scalable algorithm with convergence guarantees to our knowledge.

Existing work

Sample-based approximations to the ambiguity set: Finite-sum instead of expectation

- **Convex loss:** Finite-dimensional convex program formulations [Esfahani & Kuhn, 2018],[Abadeh et al., 2015], [Chen & Pashalidis 2018], bandit mirror descent Namkoong & Duchi, 2016], conic interior point solvers or gradient descent with backtracking Armijo line-searches [Duchi & Namkoong, 2021], convex and Lipschitz losses [Levy et al., 2020], SGD-based algorithm with $\mathcal{O}(1/\varepsilon^2)$ complexity for Lipschitz and smooth losses [Soma & Yoshida, 2020], SAPD alg. [Zhang et al., 2022],..
- **Smooth non-convex loss:** Wasserstein distance-based [Sinha et al. 2018], f-divergences/smooth Lipschitz losses [Jin et al. 2021], $\mathcal{O}(1/\varepsilon^6)$ complexity for smooth weakly convex losses [Zhang et al., 2022], (nonsmooth) weakly convex/strongly convex min-max approach of [Yan et al., 2020], CVaR-based approach with $\mathcal{O}(1/\varepsilon^6)$ complexity [Soma & Yoshida, 2020], ..
- **Non-smooth nonconvex loss:** For "zero-one loss" in linear classification, efficient algorithms for smoothed ramp loss [Ho-Nguyen, Wright, 2021].

For general non-smooth non-convex losses, no scalable algorithm with convergence guarantees to our knowledge.

Modeling $\mathcal{M}(\mathbb{P})$ with mean semi-deviation risk I

- The mean–semideviation risk measure is defined as follows:

$$\rho[Z] = \mathbb{E}[Z] + \varkappa\, \mathbb{E}\big[\max\big(0, Z - \mathbb{E}[Z]\big)\big], \qquad \varkappa \in [0,1].$$

- It is known to be a coherent measure of risk.
- In particular, it has the dual representation

$$\rho[Z] = \max_{\mu \in \mathcal{A}} \int_\Omega Z(\omega)\mu(\omega)\, \mathbb{P}(d\omega) = \max_{\mathbb{Q}\,:\, \frac{d\mathbb{Q}}{d\mathbb{P}} \in \mathcal{A}} \int_\Omega Z(\omega)\, \mathbb{Q}(d\omega)$$
$$= \max_{\mathbb{Q}\,:\, \frac{d\mathbb{Q}}{d\mathbb{P}} \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}}[Z],$$

where $\mathcal{A}$ is a convex and closed set defined as follows:

$$\mathcal{A} = \big\{\mu = \mathbb{1} + \xi - \mathbb{E}[\xi] :\ \xi \in \mathcal{L}_\infty(\Omega, \mathcal{F}, \mathbb{P}),\ \|\xi\|_\infty \le \varkappa,\ \xi \ge 0\big\}.$$

Modeling $\mathcal{M}(\mathbb{P})$ with mean semi-deviation risk II

- After plugging $Z = \ell(x, D)$ into this formulation, we obtain

$$\min_{x \in X} \max_{\mathbb{Q} \in \mathcal{M}(\mathbb{P})} \mathbb{E}_{\mathbb{Q}}[\ell(x, D)] = \min_{x \in X} \mathbb{E}\Big[\ell(x, D) + \varkappa \max\big(0, \ell(x, D) - \mathbb{E}[\ell(x, D)]\big)\Big],$$

  with the perturbation set

$$\mathcal{M}(\mathbb{P}) = \{\mathbb{Q} : \frac{d\mathbb{Q}}{d\mathbb{P}} \in \mathcal{A}\}.$$

- Max over probability distributions is avoided.
- Robust binary linear classification with Wasserstein ambiguity is equivalent to unconstrained "ramp loss" [Ho Nguyen, Wright, 2021] or maximizing CVaR risk measure (of distance to misclassification) and minimizing without finite-support assumption.

Modeling $\mathcal{M}(\mathbb{P})$ with mean semi-deviation risk II

- After plugging $Z = \ell(x, D)$ into this formulation, we obtain

$$\min_{x \in X} \max_{\mathbb{Q} \in \mathcal{M}(\mathbb{P})} \mathbb{E}_{\mathbb{Q}}[\ell(x, D)] = \min_{x \in X} \mathbb{E}\Big[\ell(x, D) + \varkappa \max\left(0, \ell(x, D) - \mathbb{E}[\ell(x, D)]\right)\Big],$$

  with the perturbation set

$$\mathcal{M}(\mathbb{P}) = \{\mathbb{Q} : \frac{d\mathbb{Q}}{d\mathbb{P}} \in \mathcal{A}\}.$$

- Max over probability distributions is avoided.
- Robust binary linear classification with Wasserstein ambiguity is equivalent to unconstrained "ramp loss" [Ho Nguyen, Wright, 2021] or maximizing CVaR risk measure (of distance to misclassification) and minimizing without finite-support assumption.

A composition optimization problem

- This yields

$$\min_{x \in X} \; f(x, h(x)),$$

with the functions

$$f(x,u) = \mathbb{E}\Big[\ell(x,D) + \varkappa \max\big(0, \ell(x,D) - u\big)\Big],$$
$$h(x) = \mathbb{E}[\ell(x,D)].$$

- The main difficulty is that neither values nor (sub)gradients of $f(\cdot)$, $h(\cdot)$, and of their composition are available.
- Instead, we postulate access to their random estimates.

Contributions

- New modeling of the uncertainty set:
  - Uses mean-semideviation risk.
  - Computational advantage for the max step.
- New single-time scale (STS) stochastic subgradient algorithm
  - Works for all generalized differentiable losses
  - Scalable (cost is at most 2 times that of SGD).
  - With probability one convergence to a stationary point.
  - Can handle the streaming data setting.
- Iteration and sample complexity for (non-smooth or smooth) weakly convex losses

References:

- A Stochastic Subgradient Method for Distributionally Robust Non-Convex and Non-Smooth Learning [Gurbuzbalaban, Ruszczynski, Zhu; Journal of Optimization Theory and Applications, 2022]
- Distributionally Robust Learning with Weakly Convex Losses: Convergence and Finite-Sample Guarantees [Gurbuzbalaban, Ruszczynski and Zhu, 2023].

Contributions

- New modeling of the uncertainty set:
  - Uses mean-semideviation risk.
  - Computational advantage for the max step.
- New single-time scale (STS) stochastic subgradient algorithm
  - Works for all generalized differentiable losses
  - Scalable (cost is at most 2 times that of SGD).
  - With probability one convergence to a stationary point.
  - Can handle the streaming data setting.
- Iteration and sample complexity for (non-smooth or smooth) weakly convex losses

References:

- A Stochastic Subgradient Method for Distributionally Robust Non-Convex and Non-Smooth Learning [Gurbuzbalaban, Ruszczynski, Zhu; Journal of Optimization Theory and Applications, 2022]
- Distributionally Robust Learning with Weakly Convex Losses: Convergence and Finite-Sample Guarantees [Gurbuzbalaban, Ruszczynski and Zhu, 2023].

Contributions

- New modeling of the uncertainty set:
  - Uses mean-semideviation risk.
  - Computational advantage for the max step.
- New single-time scale (STS) stochastic subgradient algorithm
  - Works for all generalized differentiable losses
  - Scalable (cost is at most 2 times that of SGD).
  - With probability one convergence to a stationary point.
  - Can handle the streaming data setting.
- Iteration and sample complexity for (non-smooth or smooth) weakly convex losses

References:

- A Stochastic Subgradient Method for Distributionally Robust Non-Convex and Non-Smooth Learning [Gurbuzbalaban, Ruszczynski, Zhu; Journal of Optimization Theory and Applications, 2022]
- Distributionally Robust Learning with Weakly Convex Losses: Convergence and Finite-Sample Guarantees [Gurbuzbalaban, Ruszczynski and Zhu, 2023].

Assumptions

- We make the following assumptions.
  - (A1) The set $X \subset \mathbb{R}^n$ is convex and compact;
  - (A2) For almost every (a.e.) $\omega \in \Omega$, the function $\ell(\cdot, D(\omega))$ is differentiable in a generalized (Norkin) sense with the subdifferential $\partial_x \ell(x, D(\omega))$, $x \in \mathbb{R}^n$ and we can interchange the expectation with the subderivative.

Definition

Given $x \in \mathbb{R}^n$, by (A2), generalized subdifferential is well-defined:

$$G_F(x) = \operatorname{conv}\left\{ s \in \mathbb{R}^n : s = g_x + J^\top g_u, \ \begin{bmatrix} g_x \\ g_u \end{bmatrix} \in \partial f(x, h(x)), \ J \in \partial h(x) \right\}.$$

- We say $x^* \in X$ *stationary* if $0 \in G_F(x^*) + N_X(x^*)$.
- Stochastic estimates of subgradients and function values are "easy".

Assumptions

- We make the following assumptions.

  (A1) The set $X \subset \mathbb{R}^n$ is convex and compact;

  (A2) For almost every (a.e.) $\omega \in \Omega$, the function $\ell(\cdot, D(\omega))$ is differentiable in a generalized (Norkin) sense with the subdifferential $\partial_x \ell(x, D(\omega))$, $x \in \mathbb{R}^n$ and we can interchange the expectation with the subderivative.

Definition

Given $x \in \mathbb{R}^n$, by (A2), generalized subdifferential is well-defined:

$$G_F(x) = \operatorname{conv}\Big\{ s \in \mathbb{R}^n : s = g_x + J^\top g_u, \ \begin{bmatrix} g_x \\ g_u \end{bmatrix} \in \partial f(x, h(x)), \ J \in \partial h(x) \Big\}.$$

- We say $x^* \in X$ *stationary* if $0 \in G_F(x^*) + N_X(x^*)$.
- Stochastic estimates of subgradients and function values are "easy".

Assumptions

- We make the following assumptions.
  - (A1) The set $X \subset \mathbb{R}^n$ is convex and compact;
  - (A2) For almost every (a.e.) $\omega \in \Omega$, the function $\ell(\cdot, D(\omega))$ is differentiable in a generalized (Norkin) sense with the subdifferential $\partial_x \ell(x, D(\omega))$, $x \in \mathbb{R}^n$ and we can interchange the expectation with the subderivative.

Definition

Given $x \in \mathbb{R}^n$, by (A2), generalized subdifferential is well-defined:

$$G_F(x) = \operatorname{conv}\Big\{ s \in \mathbb{R}^n : s = g_x + J^\top g_u, \ \begin{bmatrix} g_x \\ g_u \end{bmatrix} \in \partial f(x, h(x)), \ J \in \partial h(x) \Big\}.$$

- We say $x^* \in X$ *stationary* if $0 \in G_F(x^*) + N_X(x^*)$.
- Stochastic estimates of subgradients and function values are "easy".

Our method

- For $k = 0, 1, 2, \ldots$, with stepsize[1] τ_k, any scalars $a, b, c > 0$;

$$\begin{aligned} y^k &= \operatorname*{argmin}_{y \in X} \left\{ \langle z^k, y - x^k \rangle + \frac{c}{2} \|y - x^k\|^2 \right\}, \\ x^{k+1} &= x^k + \tau_k (y^k - x^k). \end{aligned}$$

- Track subgradient and inner function with (exponential) averaging:

$$z^{k+1} = (1 - a\tau_k) z^k + a\tau_k \underbrace{\left(\tilde{g}_x^{k+1} + \left[\tilde{J}^{k+1} \right]^\top \tilde{g}_u^{k+1} \right)}_{\text{Stochastic subgradient}},$$

$$u^{k+1} = (1 - b\tau_k) u^k + + b\tau_k \underbrace{\tilde{h}^{k+1}}_{\text{loss estimate}} + \tau_k \underbrace{\tilde{J}^{k+1}(y^k - x^k)}_{\text{effect of updated solution}}$$

based on "cheap" stochastic estimates $\tilde{g}_x^{k+1}, \tilde{g}_u^{k+1}, \tilde{J}^{k+1}, \tilde{h}^{k+1}$.

[1] $\lim_{k \to \infty} \tau_k = 0$, $\sum_{k=0}^{\infty} \tau_k = \infty$, $\sum_{k=0}^{\infty} \mathbb{E}[\tau_k^2] < \infty$, $\tau_k \in (0, \min(1, 1/a))$

Our method

- For $k = 0, 1, 2, \ldots$, with stepsize[1] τ_k, any scalars $a, b, c > 0$;

$$\begin{aligned} y^k &= \operatorname*{argmin}_{y \in X} \left\{ \langle z^k, y - x^k \rangle + \frac{c}{2} \|y - x^k\|^2 \right\}, \\ x^{k+1} &= x^k + \tau_k (y^k - x^k). \end{aligned}$$

- Track subgradient and inner function with (exponential) averaging:

$$z^{k+1} = (1 - a\tau_k) z^k + a\tau_k \underbrace{\left(\tilde{g}_x^{k+1} + \left[\tilde{J}^{k+1} \right]^\top \tilde{g}_u^{k+1} \right)}_{\text{Stochastic subgradient}},$$

$$u^{k+1} = (1 - b\tau_k) u^k + + b\tau_k \underbrace{\tilde{h}^{k+1}}_{\text{loss estimate}} + \tau_k \underbrace{\tilde{J}^{k+1}(y^k - x^k)}_{\text{effect of updated solution}}$$

based on "cheap" stochastic estimates $\tilde{g}_x^{k+1}, \tilde{g}_u^{k+1}, \tilde{J}^{k+1}, \tilde{h}^{k+1}$.

[1] $\lim_{k \to \infty} \tau_k = 0$, $\sum_{k=0}^{\infty} \tau_k = \infty$, $\sum_{k=0}^{\infty} \mathbb{E}[\tau_k^2] < \infty$, $\tau_k \in (0, \min(1, 1/a))$

Our method

- For $k = 0, 1, 2, \ldots$, with stepsize[1] τ_k, any scalars $a, b, c > 0$;

$$\begin{aligned} y^k &= \operatorname*{argmin}_{y \in X} \left\{ \langle z^k, y - x^k \rangle + \frac{c}{2} \|y - x^k\|^2 \right\}, \\ x^{k+1} &= x^k + \tau_k (y^k - x^k). \end{aligned}$$

- Track subgradient and inner function with (exponential) averaging:

$$z^{k+1} = (1 - a\tau_k) z^k + a\tau_k \underbrace{\left(\tilde{g}_x^{k+1} + \left[\tilde{J}^{k+1} \right]^\top \tilde{g}_u^{k+1} \right)}_{\text{Stochastic subgradient}},$$

$$u^{k+1} = (1 - b\tau_k) u^k + + b\tau_k \underbrace{\tilde{h}^{k+1}}_{\text{loss estimate}} + \tau_k \underbrace{\tilde{J}^{k+1}(y^k - x^k)}_{\text{effect of updated solution}}$$

based on "cheap" stochastic estimates $\tilde{g}_x^{k+1}, \tilde{g}_u^{k+1}, \tilde{J}^{k+1}, \tilde{h}^{k+1}$.

[1] $\lim_{k \to \infty} \tau_k = 0$, $\sum_{k=0}^{\infty} \tau_k = \infty$, $\sum_{k=0}^{\infty} \mathbb{E}[\tau_k^2] < \infty$, $\tau_k \in (0, \min(1, 1/a))$

Stochastic estimates

- Draw a second independent sample D_2^{k+1} only if the loss based on the first sample D_1^{k+1} looks "bad".

$$\tilde{g}_x^{k+1} \in \begin{cases} \partial_x \ell(x^{k+1}, D_1^{k+1}) & \text{if } \ell(x^{k+1}, D_1^{k+1}) < u^k, \\ (1+\varkappa)\partial_x \ell(x^{k+1}, D_1^{k+1}) & \text{if } \ell(x^{k+1}, D_1^{k+1}) \geq u^k, \end{cases}$$

$$\tilde{g}_u^{k+1} = \begin{cases} 0 & \text{if } \ell(x^{k+1}, D_1^{k+1}) < u^k, \\ -\varkappa & \text{if } \ell(x^{k+1}, D_1^{k+1}) \geq u^k, \end{cases}$$

$$\tilde{h}^{k+1} = \ell(x^{k+1}, D_1^{k+1}),$$

$$\tilde{J}^{k+1} \in \begin{cases} \{\tilde{g}_x^{k+1}\} & \text{if } \ell(x^{k+1}, D_1^{k+1}) < u^k, \\ \partial_x \ell(x^{k+1}, D_2^{k+1}) & \text{if } \ell(x^{k+1}, D_1^{k+1}) \geq u^k. \end{cases}$$

Convergence result

Theorem (Informal)

If the assumptions (A1)–(A2) *are satisfied, and stochastic subgradients have (conditionally) bounded variance, then with probability 1 every accumulation point* $\hat{x}$ *of the sequence* $\{x^k\}$ *is stationary,* $\lim_{k\to\infty}(u^k - h(x^k)) = 0$, *and the sequence* $\{F(x^k)\}$ *is convergent.*

2

- Step 1: The Limiting Dynamical System is a "Differential Inclusion".
- Step 2: Descent Along a Path through our Lyapunov function.
- Step 3: Analysis of the Limit Points.

[2] Assuming the set of optimal values do not contain an interval of positive length.

Convergence result

Theorem (Informal)

If the assumptions (A1)–(A2) *are satisfied, and stochastic subgradients have (conditionally) bounded variance, then with probability 1 every accumulation point* $\hat{x}$ *of the sequence* $\{x^k\}$ *is stationary,* $\lim_{k\to\infty}(u^k - h(x^k)) = 0$, *and the sequence* $\{F(x^k)\}$ *is convergent.*

2

- Step 1: The Limiting Dynamical System is a "Differential Inclusion".
- Step 2: Descent Along a Path through our Lyapunov function.
- Step 3: Analysis of the Limit Points.

[2] Assuming the set of optimal values do not contain an interval of positive length.

Convergence result

Theorem (Informal)

If the assumptions (A1)–(A2) *are satisfied, and stochastic subgradients have (conditionally) bounded variance, then with probability 1 every accumulation point* $\hat{x}$ *of the sequence* $\{x^k\}$ *is stationary,* $\lim_{k\to\infty}(u^k - h(x^k)) = 0$, *and the sequence* $\{F(x^k)\}$ *is convergent.*

[2]

- Step 1: The Limiting Dynamical System is a "Differential Inclusion".
- Step 2: Descent Along a Path through our Lyapunov function.
- Step 3: Analysis of the Limit Points.

[2] Assuming the set of optimal values do not contain an interval of positive length.

Convergence result

Theorem (Informal)

If the assumptions (A1)–(A2) *are satisfied, and stochastic subgradients have (conditionally) bounded variance, then with probability 1 every accumulation point* $\hat{x}$ *of the sequence* $\{x^k\}$ *is stationary,* $\lim_{k\to\infty}(u^k - h(x^k)) = 0$, *and the sequence* $\{F(x^k)\}$ *is convergent.*

[2]

- Step 1: The Limiting Dynamical System is a "Differential Inclusion".
- Step 2: Descent Along a Path through our Lyapunov function.
- Step 3: Analysis of the Limit Points.

[2] Assuming the set of optimal values do not contain an interval of positive length.

Deep learning experiment

- We consider a fully-connected network on two benchmark datasets: MNIST and CIFAR10, where the model has the depth (the number of layers) of 3 and the width (the number of neurons per hidden layer) of 100.
- In both MNIST and CIFAR10 datasets, the output variable y to be predicted is an integer valued from 0 to 9.
- We distort the distributions of MNIST and CIFAR10 training datasets by deleting almost all the data points with a y value equal to 0.
- If the training data are not contaminated at all, we have observed in our experiments that STS generates a similar or slightly worse solution than SGD.
- When the data contains distributional shifts, we see a clear advantage of the STS method over the SGD method.

MNIST dataset

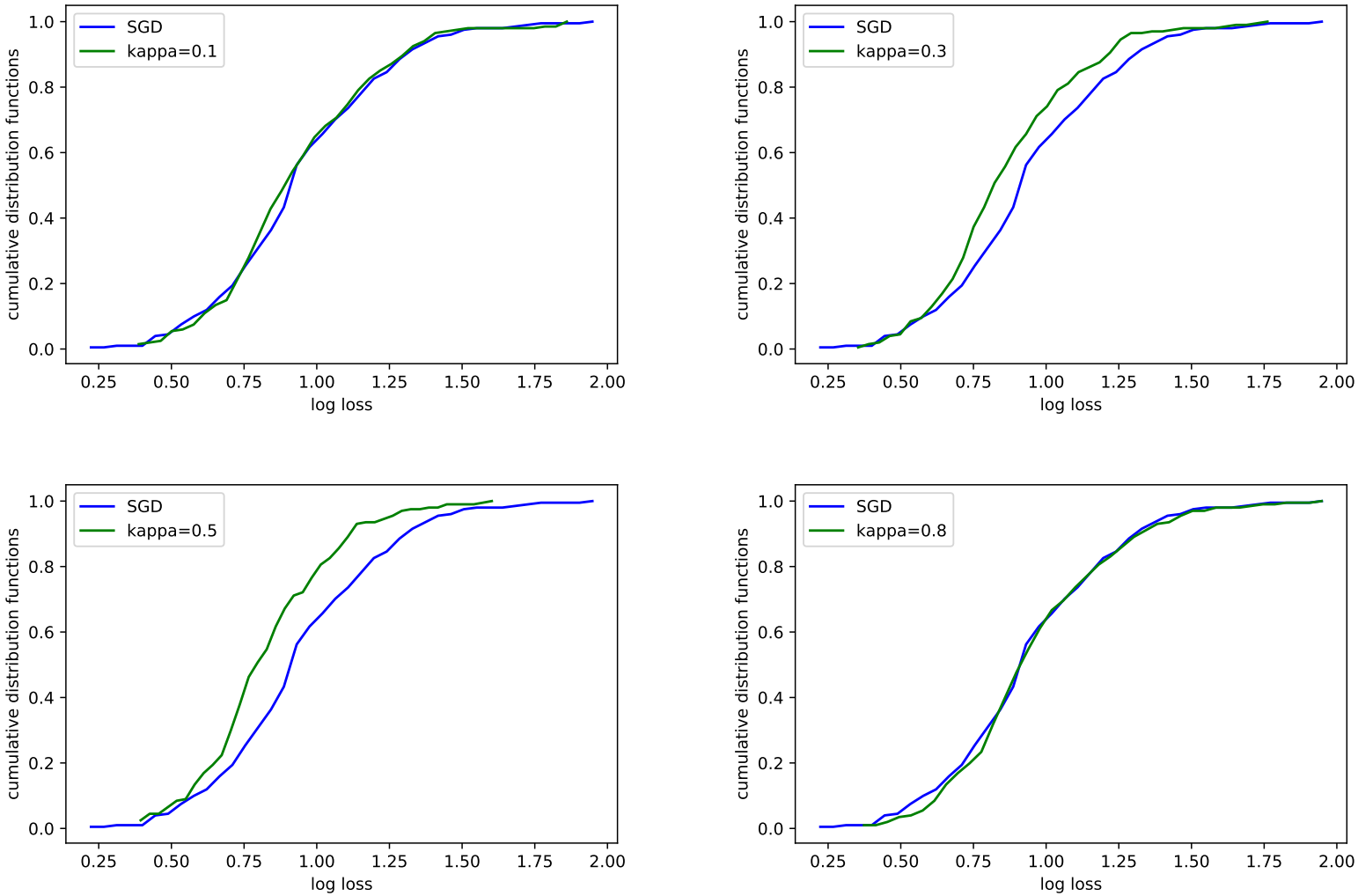


Figure: The CDFs of the SGD solution and the STS solutions.

CIFAR10 dataset

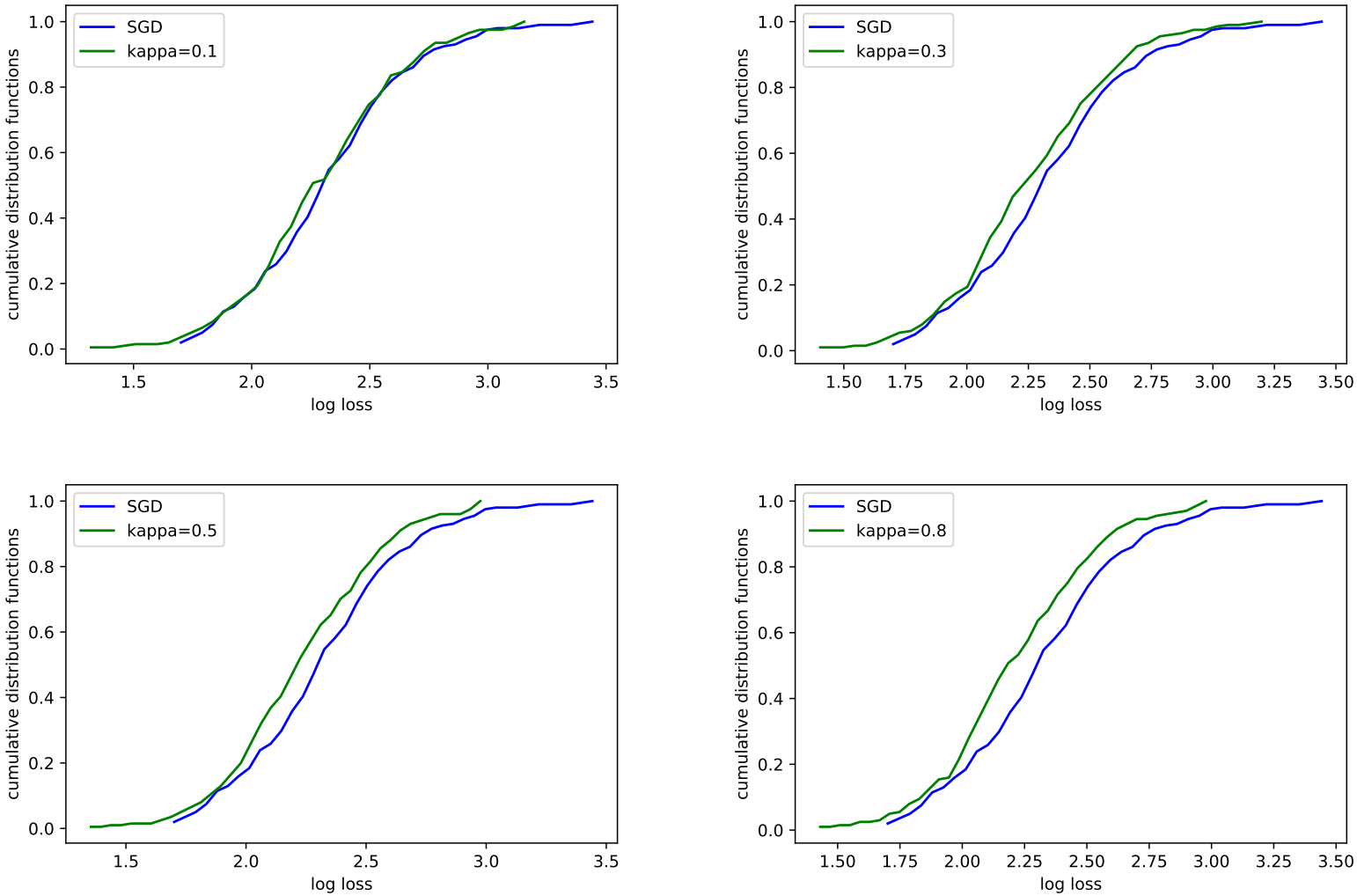


Figure: The CDFs of the SGD solution and the STS solutions.

Logistic regression experiment

- We consider binary logistic regression on the Adult dataset where the loss function has the form $\ell(x, D) = \left[\log(1 + \exp(-b\, a^T x))\right]$.
- We follow a similar methodology as before, where we distort the training data by deleting 80% of the data points with the corresponding income below \$50,000.
- We trained our model with STS and another state-of-the-art method Bandit Mirror Descent (BMD).
- We see that STS results in smaller errors and conclude that our method has desirable robustness properties with respect to perturbations in the input distribution.

Adult dataset

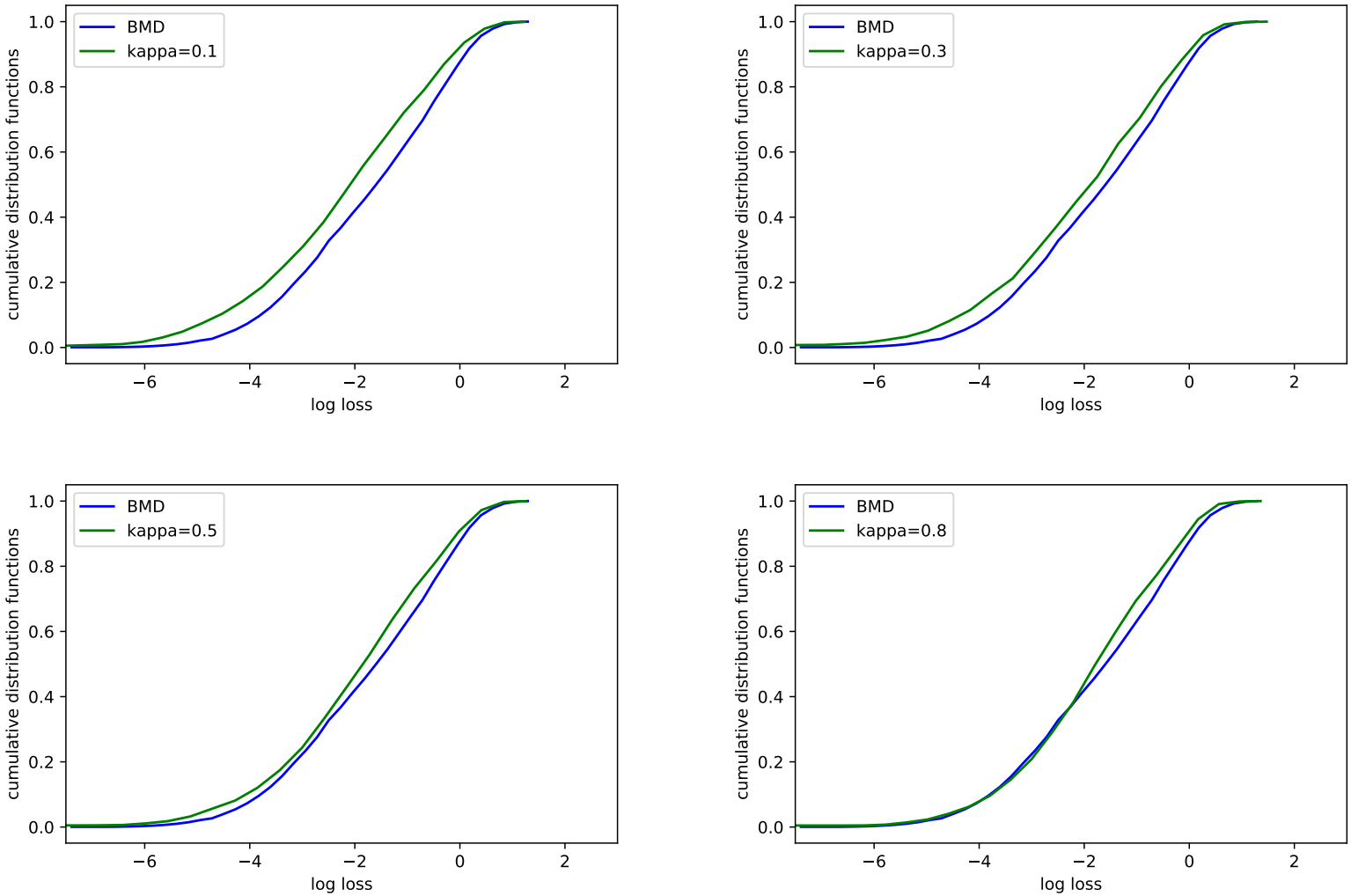


Figure: The CDFs of the BMD solution and the STS solutions.

Smooth weakly convex problems I

- When assuming a smooth loss function, we may adapt the STS method to a projected subgradient descent framework, and use gradient of the Moreau envelope as our new metric.
- Consider an alternative formulation of the main problem:

$$\min_{x \in \mathbb{R}^n} \ \varphi(x) := F(x) + r(x),$$

where $F(x) = f(x, h(x))$ and $r(x)$ is the indicator function of a convex and compact feasible set $X \subset \mathbb{R}^n$.

- The Moreau envelope and the proximal map are defined as:

$$\varphi_\lambda(x) := \min_y \{\varphi(y) + \frac{1}{2\lambda}\|y - x\|^2\},$$

$$prox_{\lambda\varphi}(x) := \operatorname{argmin}_y \{\varphi(y) + \frac{1}{2\lambda}\|y - x\|^2\},$$

Smooth weakly convex problems II

- A δ-weakly convex function $h(x) = \mathbb{E}[\ell(x, D)]$ has the following property: at every point $x \in \mathbb{R}^n$ a vector $g \in \mathbb{R}^n$ exists such that

$$h(y) \geq h(x) + \langle g, y - x \rangle - \frac{\delta}{2}\|y - x\|^2, \qquad \forall y \in \mathbb{R}^n.$$

- $\varphi_\lambda(x)$ is smooth when $\lambda \in (0, \rho^{-1})$. It has a gradient given by

$$\nabla \varphi_\lambda(x) = \lambda^{-1}(x - \mathit{prox}_{\lambda\varphi}(x)).$$

- It can also be shown that the quantity $\|\nabla \varphi_\lambda(x)\|$ is a measure of stationarity, i.e. when $\|\nabla \varphi_\lambda(x)\|$ is small, x will be near some *nearly stationary point* $\hat{x}$, which in turn, has the subdifferential close to 0:

$$\begin{cases} \|\hat{x} - x\| = \lambda\|\nabla \varphi_\lambda(x)\|, \\ \varphi(\hat{x}) \leq \varphi(x), \\ \operatorname{dist}(0; \partial\varphi(\hat{x})) \leq \|\nabla \varphi_\lambda(x)\|. \end{cases}$$

The stochastic compositional subgradient (SCS) method

- The algorithm can be summarized as

$$x^{k+1} = \Pi_X\left(x^k - \tau\left(\tilde{g}_{fx}^k + \tilde{g}_{fu}^k \tilde{g}_h^k\right)^T\right),$$
$$u^{k+1} = u^k + \tau\left(\tilde{h}^k - u^k\right) + \tilde{J}^k\left(x^{k+1} - x^k\right).$$

- And we also update our statistical estimates

$$G^k \in \partial_x \ell(x^k, D_1^{k+1}),$$
$$\tilde{g}_{fx}^k = \begin{cases} 0 & \text{if } \ell(x^k, D_1^{k+1}) < u^k, \\ \varkappa G^k & \text{if } \ell(x^k, D_1^{k+1}) \geq u^k, \end{cases}$$
$$\tilde{g}_{fu}^k = \begin{cases} 1 & \text{if } \ell(x^k, D_1^{k+1}) < u^k, \\ 1 - \varkappa & \text{if } \ell(x^k, D_1^{k+1}) \geq u^k, \end{cases}$$
$$\tilde{g}_h^k \in \partial_x \ell(x^k, D_2^{k+1}), \qquad \tilde{J}^k \in \partial_x \ell(x^k, D_3^{k+1}),$$
$$\tilde{h}^k = \frac{1}{3}(\ell(x^k, D_1^{k+1}) + \ell(x^k, D_2^{k+1}) + \ell(x^k, D_3^{k+1})).$$

Assumptions of SCS

(B1) The set $X \subset \mathbb{R}^n$ is convex and compact.

(B2) For all x in a neighborhood of the set X:

- The function $\ell(x, \cdot)$ is integrable;
- The function $\ell(\cdot, D)$ is continuously differentiable and integrable constants $\tilde{\Delta}_h(D)$ and $\tilde{\delta}(D)$ exist such that

$$\|\nabla \ell(x, D)\| \leq \tilde{\Delta}_h(D), \quad \forall\, D \in \mathbb{R}^d,$$

and

$$\|\nabla \ell(x, D) - \nabla \ell(y, D)\| \leq \tilde{\delta}(D)\|x - y\|, \quad \forall\, x, y \in X, \quad \forall\, D \in \mathbb{R}^d.$$

(B3) The stochastic estimates are unbiased and have finite error variances.

Convergence rate for smooth weakly convex losses

Theorem

Suppose Assumptions (B1)–(B3) hold. For any given iteration budget N, consider the trajectory $\{x^k\}_{k=0}^{N-1}$ of SCS. We have

$$\mathbb{E}[\|\nabla \varphi_{1/\bar{\rho}}(x^R)\|^2] \leq 2\frac{C_1 + NC_2\tau^{3/2}}{N\tau},$$

where $\bar{\rho}$, C_1 and C_2 are constants determined by the loss function and our choice of $\varkappa$, the expectation is taken with respect to the trajectory generated by SCS and the random variable R that is uniformly sampled from $\{0, 1, ..., N-1\}$ independently of the trajectory.

- If we choose $\tau = cN^{-2/3}$ for some constant $c > 0$, this theorem indicates the sample complexity of SCS is $\mathcal{O}(\varepsilon^{-3})$.

Nonsmooth weakly convex problems

- If we only assume a weakly convex loss function, instead of a smooth one, we can use the SPIDER estimator (a variant of SARAH [Nguyen et al. 2017]) to estimate the expectation of the loss function:

$$u^k = \ell_{\mathcal{B}^k}(x^k), \quad \|\mathcal{B}^k\| = B, \quad \text{if } k \bmod T == 0,$$
$$u^k = u^{k-1} + \ell_{\mathcal{B}^k}(x^k) - \ell_{\mathcal{B}^k}(x^{k-1}), \quad \|\mathcal{B}^k\| = b, \quad \text{otherwise.}$$

  where T is the SPIDER cycle length.
- Now the assumptions become

(B4) For all x in a neighborhood of the set X, the function $\ell(x, \cdot)$ is integrable; the function $\ell(\cdot, D)$ is weakly convex with an integrable constant $\tilde{\delta}(D)$.

(B5) The Lipschitz constant $\tilde{L}(D)$ of the loss function $\ell(x, D)$ with respect to x is square-integrable:

$$L^2 \equiv \mathbb{E}[\tilde{L}^2(D)] < +\infty.$$

Convergence rate for nonsmooth weakly convex losses

Theorem

Suppose Assumptions (B3)–(B5) hold. For any given iteration budget N, consider the trajectory $\{x^k\}_{k=0}^{N-1}$ of SCS with SPIDER. We have

$$\mathbb{E}[\|\nabla \varphi_{1/\bar{\rho}}(x^R)\|^2] \leq 2\frac{C_3 + NC_4\tau^{3/2}}{N\tau},$$

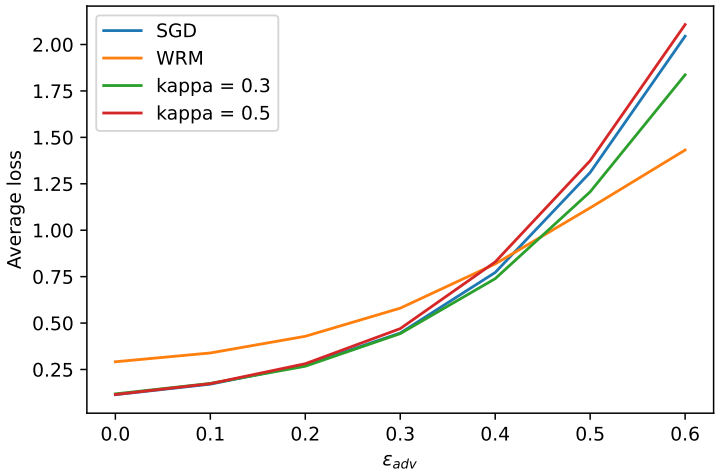
where $\bar{\rho}$, C_3 and C_4 are constants determined by the loss function and our choice of $\varkappa$, the expectation is taken with respect to the trajectory generated by SCS and the random variable R that is uniformly sampled from $\{0, 1, ..., N-1\}$ independently of the trajectory.

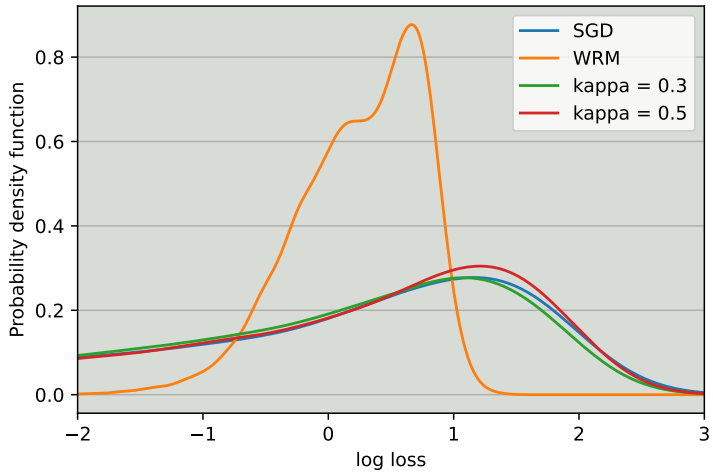
- SPIDER estimator has a lower tracking error bound, but requires an extra data batch, eventually the sample complexity is still $\mathcal{O}(\varepsilon^{-3})$.

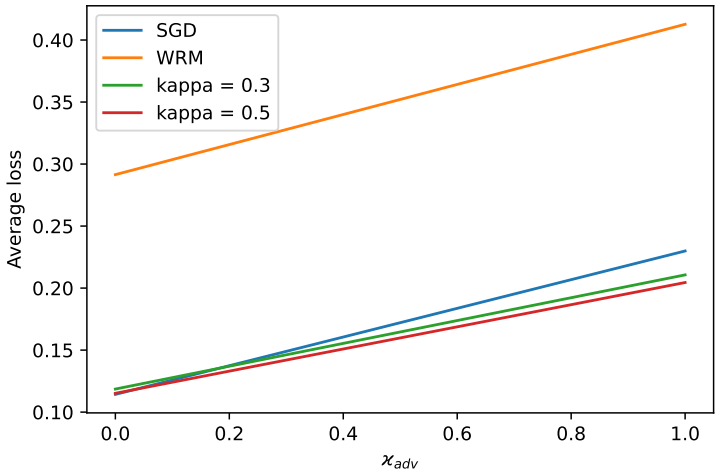
Deep learning

- We consider a convolutional neural network applied to the MNIST data set. The network consists of three convolutional layers followed by a dense layer. All the hidden layers have ELU activations, and the output layer has the softmax activation.
- We train the CNN with different optimizers, namely SGD, SCS and another state-of-the-art method Wasserstein Robust Method (WRM).
- To investigate the robustness of the trained networks, we consider two types of (adversarial attacks) perturbations to the test dataset: the PGM attacks and the semi-deviation attacks.
- The training data is the original (uncontaminated) MNIST data, whereas the models are tested with the contaminated data subject to PGM attacks and semi-deviation attacks.

Deep Learning







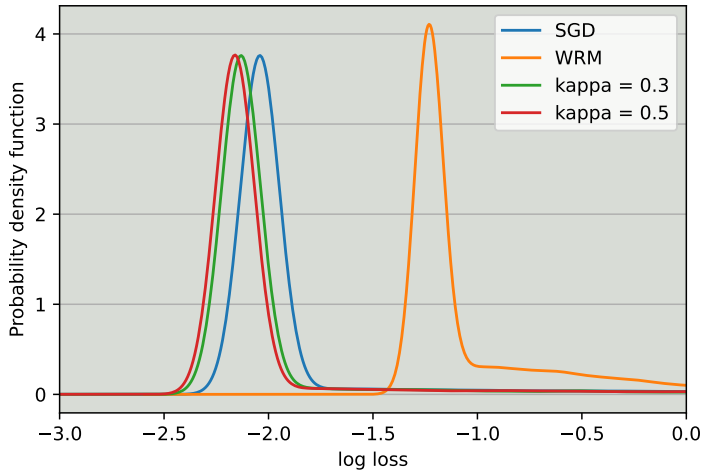


Figure: Test losses under PGM attacks (top) and semi-deviation attacks (bottom).

Nonconvex penalties

- We consider a regression task on the Blog Feedback data set.
- The loss function has the form $\ell(x, D) = |a^T x - b| + r(x)$ where $D = (a, b)$ is the input data, and $r(x)$ is the regularization term.
- Lasso:

$$r(x) = \lambda|x|,$$

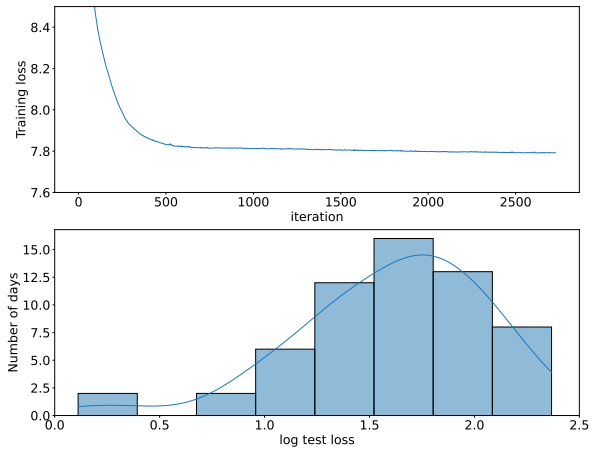
- SCAD:

$$r(x) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ \frac{\gamma\lambda|x| - 0.5(x^2+\lambda^2)}{\gamma-1} & \text{if } \lambda < |x| \leq \lambda\gamma, \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |x| > \lambda\gamma\,, \end{cases}$$

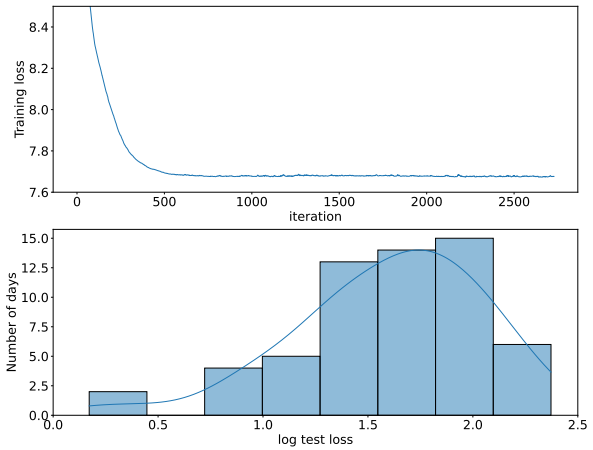
- MCP:

$$r(x) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma} & \text{if } |x| \leq \lambda\gamma, \\ \frac{\lambda^2\gamma}{2} & \text{if } |x| > \lambda\gamma, \end{cases}$$

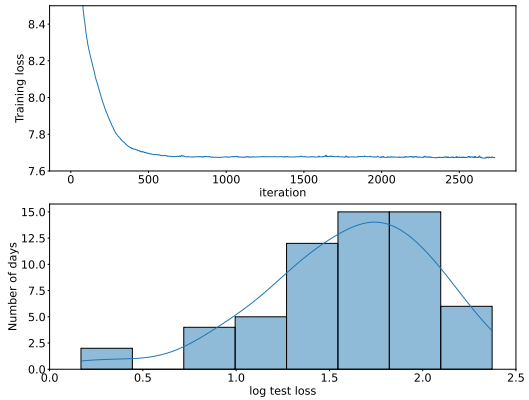
Nonconvex penalties



(a) Las



(b) SCAD penalty



(c) MCP penalty

Figure: Top: training loss vs iterations, bottom: distribution of the log test loss.

Relevant work: stochastic composite optimization

- In [Wang et al., 2017], the authors analyzed stochastic gradient algorithms with different assumptions on the objective, and prove sample complexities $\mathcal{O}(\varepsilon^{-3.5})$, $\mathcal{O}(\varepsilon^{-1.25})$ for smooth convex problems, smooth strongly convex problems respectively. These rates can be further improved with proper regularization [Wang et al., 2017].
- In [Ghadimi et al., 2020], the authors propose a single time-scale Nested Averaged Stochastic Approximation (NASA) method for smooth nonconvex composition optimization problems and prove the sample complexity of $\mathcal{O}(\varepsilon^{-2})$.
- For higher-level (more than two) problems, [Ruszczynski, 2021] establishes asymptotic convergence of a stochastic subgradient method by analyzing a system of differential inclusions, along with a sample complexity of $\mathcal{O}(\varepsilon^{-2})$ when smoothness is assumed.

Related Work: Robustness to hyperparameters

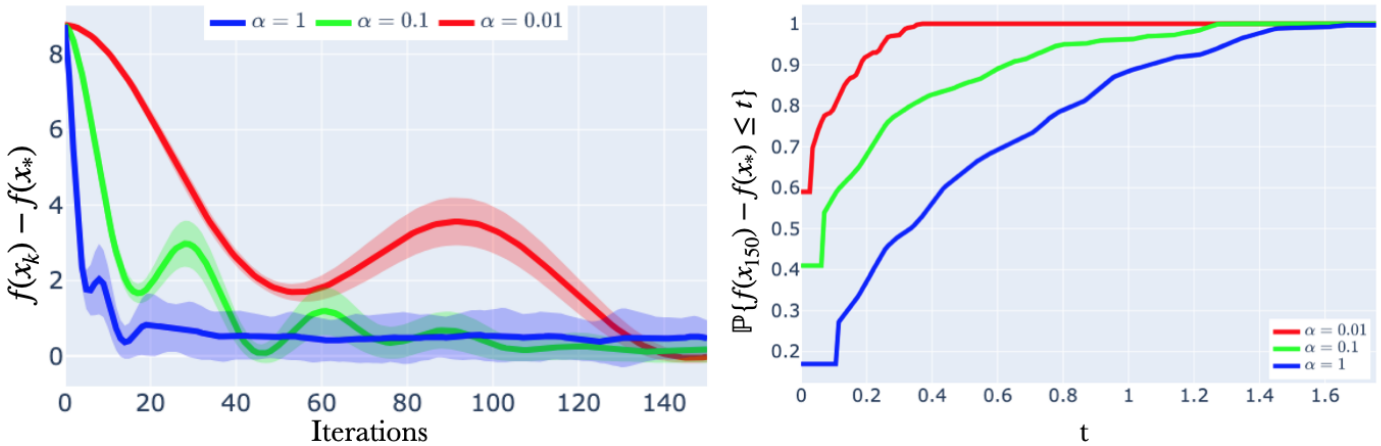


Figure: AGD algorithm with $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$ where the noise on the gradient is $\mathcal{N}(0, 16I_3)$ and the objective is quadratic function with $L = 10$ and $\mu = 0.01$. **Left:** The expected suboptimality and standard deviation from mean, **Right:** The CDF of $f(x_{150}) - f(x_*)$.

- **Our Idea:** For stochastic optimization, find stepsize and momentum parameters to minimize the risk $\rho(f(x_k) - f(x_*))$.
- Trade-offs between risk and convergence rates.
- For entropic risk $\rho(Z) := \mathbb{E}[e^{\theta Z}]$
  - Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022]
  - Generalizes risk-neutral case: Robust Accelerated Gradient Methods for Smooth Strongly Convex Functions [Aybat, Fallah, Gurbuzbalaban , Ozdaglar, SIOPT 2020].
- Min-max setting [Laguel, Aybat, Gurbuzbalaban, In preparation].

Related Work: Robustness to hyperparameters

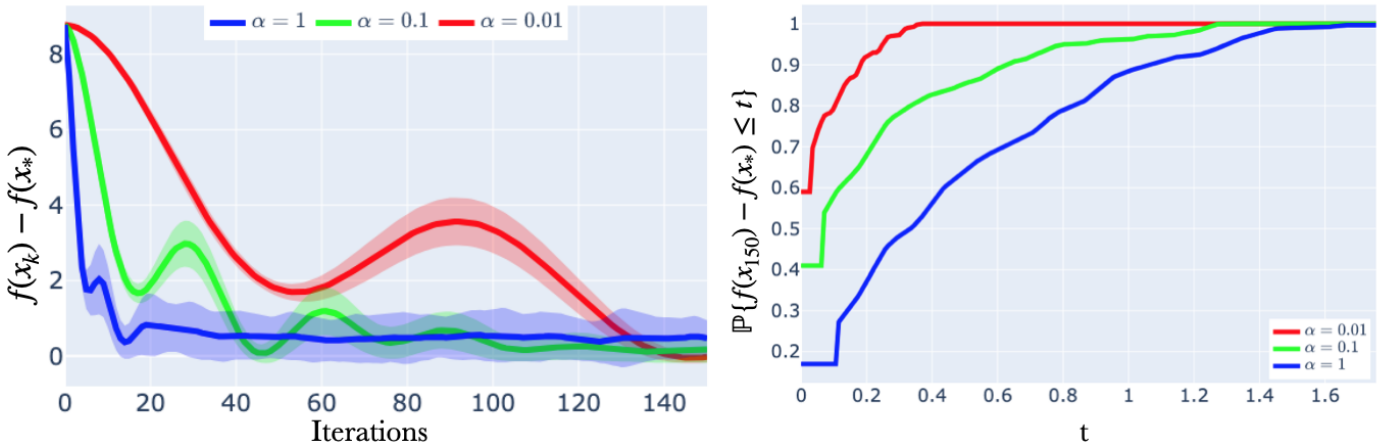


Figure: AGD algorithm with $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$ where the noise on the gradient is $\mathcal{N}(0, 16I_3)$ and the objective is quadratic function with $L = 10$ and $\mu = 0.01$. **Left:** The expected suboptimality and standard deviation from mean, **Right:** The CDF of $f(x_{150}) - f(x_*)$.

- **Our Idea:** For stochastic optimization, find stepsize and momentum parameters to minimize the risk $\rho(f(x_k) - f(x_*))$.
- Trade-offs between risk and convergence rates.
- For entropic risk $\rho(Z) := \mathbb{E}[e^{\theta Z}]$
  - Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022]
  - Generalizes risk-neutral case: Robust Accelerated Gradient Methods for Smooth Strongly Convex Functions [Aybat, Fallah, Gurbuzbalaban , Ozdaglar, SIOPT 2020].
- Min-max setting [Laguel, Aybat, Gurbuzbalaban, In preparation].

Related Work: Robustness to hyperparameters

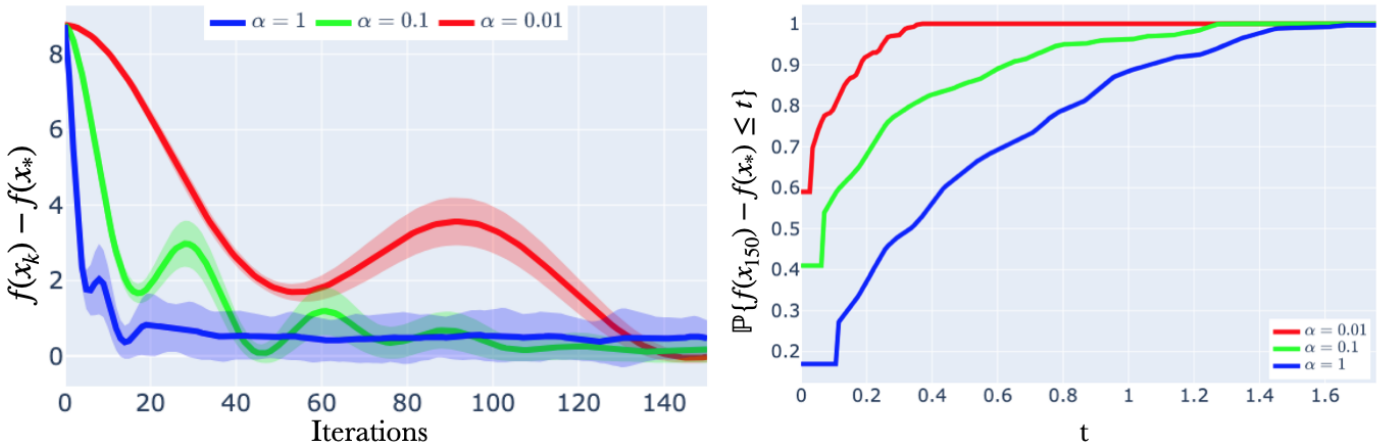


Figure: AGD algorithm with $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$ where the noise on the gradient is $\mathcal{N}(0, 16I_3)$ and the objective is quadratic function with $L = 10$ and $\mu = 0.01$. **Left:** The expected suboptimality and standard deviation from mean, **Right:** The CDF of $f(x_{150}) - f(x_*)$.

- **Our Idea:** For stochastic optimization, find stepsize and momentum parameters to minimize the risk $\rho(f(x_k) - f(x_*))$.
- Trade-offs between risk and convergence rates.
- For entropic risk $\rho(Z) := \mathbb{E}[e^{\theta Z}]$
  - Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022]
  - Generalizes risk-neutral case: Robust Accelerated Gradient Methods for Smooth Strongly Convex Functions [Aybat, Fallah, Gurbuzbalaban , Ozdaglar, SIOPT 2020].
- Min-max setting [Laguel, Aybat, Gurbuzbalaban, In preparation].

Related Work: Robustness to hyperparameters

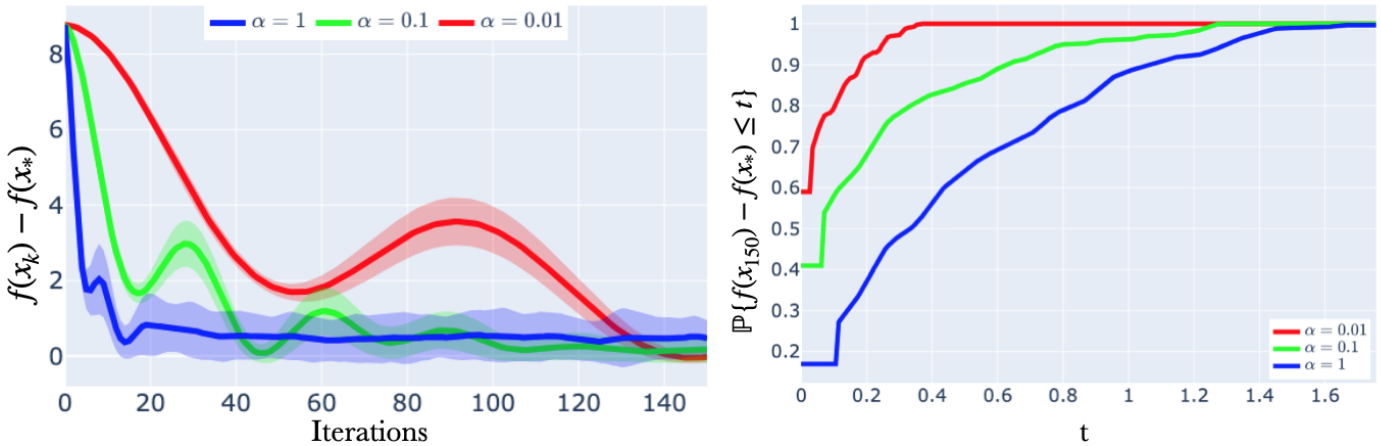


Figure: AGD algorithm with $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$ where the noise on the gradient is $\mathcal{N}(0, 16I_3)$ and the objective is quadratic function with $L = 10$ and $\mu = 0.01$. **Left:** The expected suboptimality and standard deviation from mean, **Right:** The CDF of $f(x_{150}) - f(x_*)$.

- **Our Idea:** For stochastic optimization, find stepsize and momentum parameters to minimize the risk $\rho(f(x_k) - f(x_*))$.
- Trade-offs between risk and convergence rates.
- For entropic risk $\rho(Z) := \mathbb{E}[e^{\theta Z}]$
  - Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022]
  - Generalizes risk-neutral case: Robust Accelerated Gradient Methods for Smooth Strongly Convex Functions [Aybat, Fallah, Gurbuzbalaban , Ozdaglar, SIOPT 2020].
- Min-max setting [Laguel, Aybat, Gurbuzbalaban, In preparation].

Risk-averse Momentum Methods

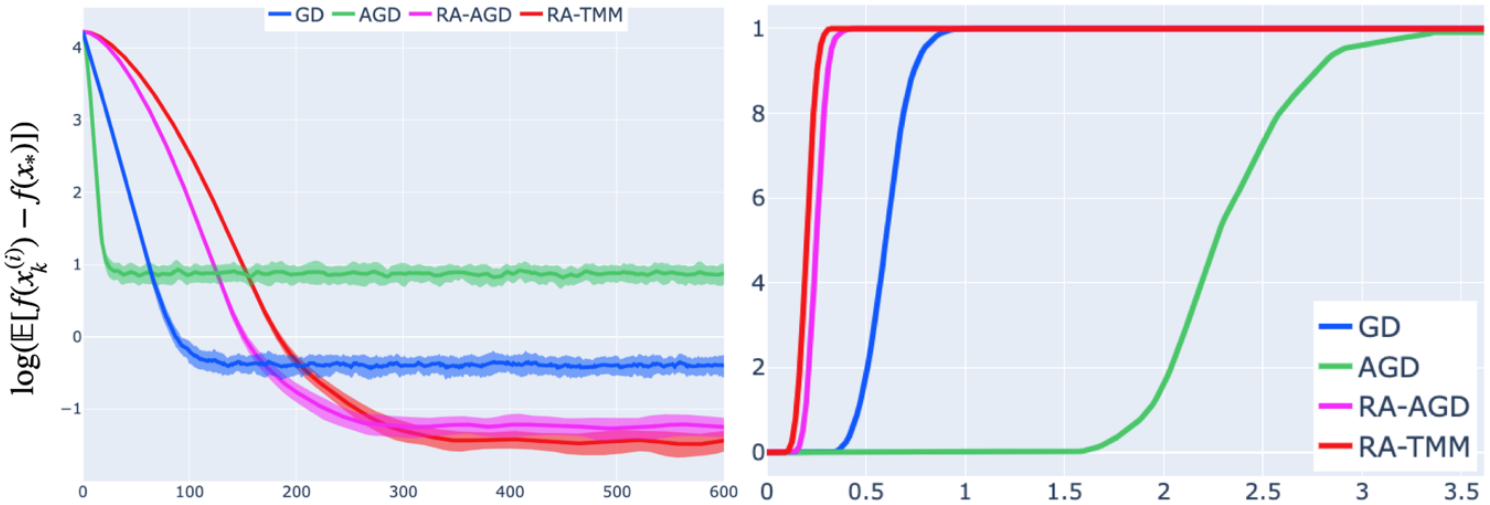


Figure: (Left) The expected suboptimality versus iterations for GD, AGD, RA-AGD and RA-TMM. **(Right)** The cumulative distribution of the suboptimality of the last iterates for GD, AGD, RA-AGD and RA-TMM after $k = 600$ iterations on logistic regression where the noise is $\mathcal{N}(0, I_{100})$.

- We plot the average $(\bar{f}_1, \ldots, \bar{f}_{300})$ where $\bar{f}_k := \frac{1}{50} \sum_{i=1}^{50} f(x_k^{(i)}) - f(x_*)$ over the samples $\{x_k^{(i)}\}_{i=1}^{50}$.
- We highlight the region between $(\bar{f}_0 \pm \sigma_0^f, \ldots, \bar{f}_{600} \pm \sigma_{600}^f)$ where $\sigma_k^f := \left(\frac{1}{50} \sum_{i=1}^{50} |f(x_k^{(i)}) - f(x_*)|^2\right)^{1/2}$.

Summary

- Our stochastic subgradient methods for distributionally robust learning
  - Admit probability one guarantees to a stationary point.
  - Only method that applies to ReLU.
  - Finite-sample guarantees for weakly convex and smooth problems.
- For convex problems, we developed robust/risk-averse triple momentum methods to gradient noise.
  - Optimal performance trading convergence rate and tail probabilities.

Main References:

- Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022].
- A Stochastic Subgradient Method for Distributionally Robust Non-Convex and Non-Smooth Learning [Gurbuzbalaban, Ruszczynski, Zhu; Journal of Optimization Theory and Applications, 2022]
- Distributionally Robust Learning with Weakly Convex Losses: Convergence Rates and Finite-Sample Guarantees [Gurbuzbalaban, Ruszczynski and Zhu, 2023].

Summary

- Our stochastic subgradient methods for distributionally robust learning
  - Admit probability one guarantees to a stationary point.
  - Only method that applies to ReLU.
  - Finite-sample guarantees for weakly convex and smooth problems.
- For convex problems, we developed robust/risk-averse triple momentum methods to gradient noise.
  - Optimal performance trading convergence rate and tail probabilities.

Main References:

- Entropic Risk-Averse Generalized Momentum Methods [Can, Gurbuzbalaban; Submitted, 2022].
- A Stochastic Subgradient Method for Distributionally Robust Non-Convex and Non-Smooth Learning [Gurbuzbalaban, Ruszczynski, Zhu; Journal of Optimization Theory and Applications, 2022]
- Distributionally Robust Learning with Weakly Convex Losses: Convergence Rates and Finite-Sample Guarantees [Gurbuzbalaban, Ruszczynski and Zhu, 2023].

Thanks

Sensitivity to noise/hyperparameters

Figure: Standard AGD with $\alpha = 1/L$ and $\beta = (1 - \sqrt{1/\kappa})/(1 + \sqrt{1/\kappa})$ on quadratic objective under the various noise levels: $\sigma = 0$ (left) and $\sigma \gg 1$ (right)

- Momentum methods are sensitive to persistent noise in the gradients [d'Aspremont, 2008],[Devolder, 2013], may even diverge [Flammarion & Bach, 2015].
- Stochastic gradients: Trade-offs between averaging and acceleration [Flammarion & Bach, 2015].

Stationary points and the multifunction Γ

- For a point $x \in \mathbb{R}^n$, we define the set:

$$G_F(x) = \operatorname{conv}\{s \in \mathbb{R}^n : s = g_x + J^\top g_u,\ g \in \partial f(x, h(x)),\ J \in \partial h(x)\}.$$

- We call a point $x^* \in X$ *stationary* for the risk minimization problem, if

$$0 \in G_F(x^*) + N_X(x^*),$$

- Consider the multifunction $\Gamma : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \rightrightarrows \mathbb{R}^n \times \mathbb{R}$:

$$\Gamma(x, z, u) = \{(R, v) : \exists g \in \partial f(x, u), \exists J_1, J_2 \in \partial h(x),$$
$$v = J_1(\bar{y}(x, z) - x) + b(h(x) - u),\ R = a(g_x + J_2^\top g_u - z)\}.$$

- With this notation,

$$\begin{bmatrix} z^{k+1} \\ u^{k+1} \end{bmatrix} \in \begin{bmatrix} z^k \\ u^k \end{bmatrix} + \tau_k \Gamma(x^{k+1}, z^k, u^k) + \tau_k \theta^{k+1} + \tau_k \alpha^{k+1}$$

with higher-order terms θ^{k+1} and α^{k+1}.

Stationary points and the multifunction Γ

- For a point $x \in \mathbb{R}^n$, we define the set:

$$G_F(x) = \operatorname{conv}\big\{s \in \mathbb{R}^n : s = g_x + J^\top g_u,\ g \in \partial f(x, h(x)),\ J \in \partial h(x)\big\}.$$

- We call a point $x^* \in X$ *stationary* for the risk minimization problem, if

$$0 \in G_F(x^*) + N_X(x^*),$$

- Consider the multifunction $\Gamma : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \rightrightarrows \mathbb{R}^n \times \mathbb{R}$:

$$\Gamma(x, z, u) = \big\{(R, v) : \exists g \in \partial f(x, u), \exists J_1, J_2 \in \partial h(x),$$
$$v = J_1\big(\bar{y}(x, z) - x\big) + b(h(x) - u),\ R = a\big(g_x + J_2^\top g_u - z\big)\big\}.$$

- With this notation,

$$\begin{bmatrix} z^{k+1} \\ u^{k+1} \end{bmatrix} \in \begin{bmatrix} z^k \\ u^k \end{bmatrix} + \tau_k \Gamma(x^{k+1}, z^k, u^k) + \tau_k \theta^{k+1} + \tau_k \alpha^{k+1}$$

with higher-order terms θ^{k+1} and α^{k+1}.

Proof of convergence I

Lemma

The multifunction Γ is compact and convex valued.

- Take two points from the output set. Consider the convexity of the input sets and the procedures to generate an arbitrary point in the output set.

Lemma

The sequences $\{z^k\}$ and $\{u^k\}$ are bounded with probability 1.

Relevant work: robust learning with smooth losses

- The authors in [Sinha et al., 2018] formulate $\mathcal{M}(\mathbb{P})$ as a ρ-neighborhood of the probability law $\mathbb{P}$ under the Wasserstein metric. They show that for a smooth loss and small enough robustness level ρ, the stochastic gradient descent (SGD) method can achieve the same rate of convergence as that in the standard smooth non-convex optimization.
- In [Jin et al., 2021], the authors consider smooth and Lipschitz non-convex losses and use a soft penalty term based on f-divergence. They analyzed the mini-batch normalized SGD with momentum and proved a $\mathcal{O}(\varepsilon^{-4})$ sample complexity.
- In [Soma & Yoshida, 2020], the authors proposed a conditional value-at-risk (CVaR) formulation. They show that for convex, Lipschitz and smooth losses their SGD-based algorithm has a complexity of $\mathcal{O}(1/\varepsilon^2)$, whereas for non-convex, smooth and Lipschitz losses, the authors obtain a complexity of $\mathcal{O}(1/\varepsilon^6)$.

Relevant work: robust learning with convex losses

- If formulated as finite-dimensional convex programs [Esfahani & Kuhn, 2018],[Abadeh et al., 2015], [Chen & Pashalidis 2018], the distributionally robust problem can be solved in polynomial time.
- When $\mathcal{M}(\mathbb{P})$ is defined via the f-divergences and the loss is convex and smooth, a sample-based approximation can be solved with a bandit mirror descent algorithm [Namkoong & Duchi, 2016] with the number of iterations comparable to that of the SGD.
- For convex losses in the same formulation, conic interior point solvers or gradient descent with backtracking Armijo line-searches [Duchi & Namkoong, 2021] can be used but can be computationally expensive.
- When the uncertainty set $\mathcal{M}(\mathbb{P})$ is based on the empirical distribution of the data and is defined via the χ^2-divergence or CVaR, and the loss is convex and Lipschitz, [Levy et al., 2020] proposed algorithms that achieve an optimal $\mathcal{O}(\varepsilon^{-2})$ rate which is independent of the training dataset size and the number of parameters.

Stochastic Momentum Methods

- Three-parameter momentum methods for minimizing $f(x)$:

$$\begin{aligned} x_{k+1} &= x_k + \beta(x_k - x_{k-1}) - \alpha\tilde{\nabla} f(y_k) \\ y_{k+1} &= x_k + \gamma(x_k - x_{k-1}) \end{aligned}$$

- Particular choice of parameters (triple momentum methods) without noise is studied in [Hu & Lessard, 2017],[Scoy et al., 2018],[Cyrus et al., 2018].
- Generalizes many methods:
  - $\gamma = \beta = 0 \implies$ Stochastic Gradient
  - $\gamma = 0 \implies$ Stochastic Heavy Ball (HB)
  - $\gamma = \beta \implies$ Stochastic Accelerated Gradient Descent (AGD)

Sensitivity to noise/hyperparameters

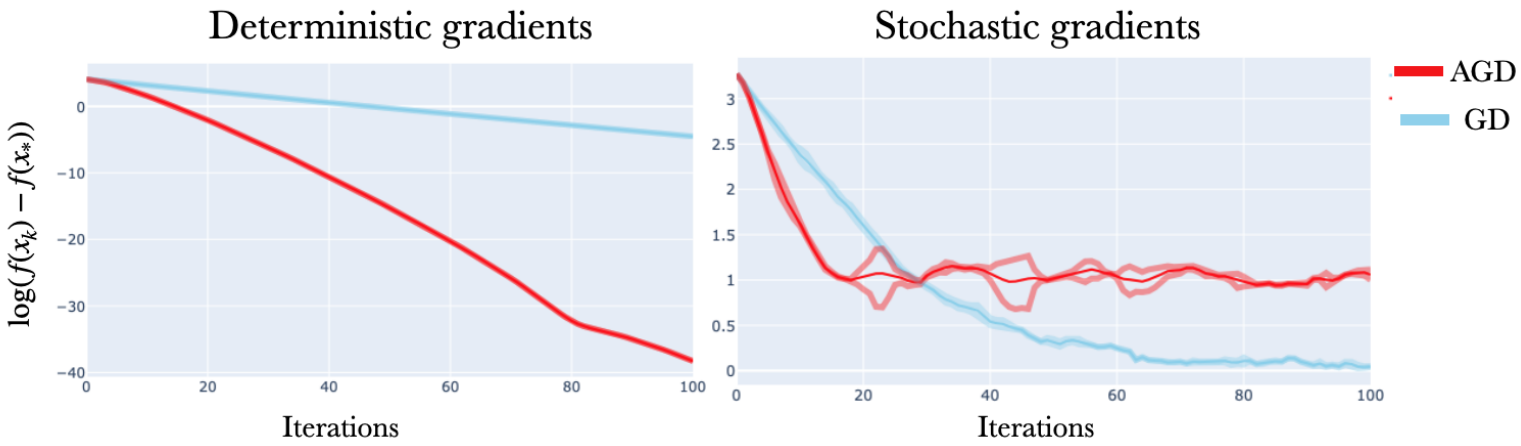


Figure: Standard AGD with $\alpha = 1/L$ and $\beta = (1 - \sqrt{1/\kappa})/(1 + \sqrt{1/\kappa})$ on quadratic objective under the various noise levels: $\sigma = 0$ (left) and $\sigma \gg 1$ (right)

- Momentum methods are sensitive to persistent noise in the gradients [d'Aspremont, 2008],[Devolder, 2013], may even diverge [Flammarion & Bach, 2015].
- Stochastic gradients: Trade-offs between averaging and acceleration [Flammarion & Bach, 2015].

Sensitivity to noise/hyperparameters

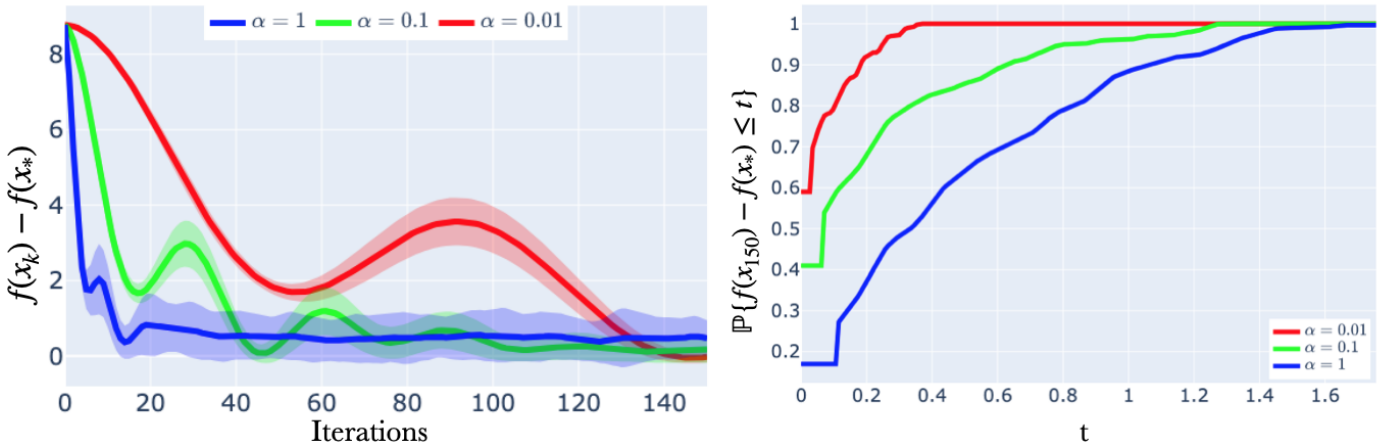


Figure: AGD algorithm with $\beta = (1 - \sqrt{\alpha\mu})/(1 + \sqrt{\alpha\mu})$ where the noise on the gradient is $\mathcal{N}(0, 16I_3)$ and the objective is quadratic function with $L = 10$ and $\mu = 0.01$. **Left:** The expected suboptimality and standard deviation from mean, **Right:** The CDF of $f(x_{150}) - f(x_*)$.

- A stochastic dominance effect based on the choice of parameter.
- The performance can be really bad unless the parameters are finely tuned!
- How to control the tail probabilities and deviation from mean as a function of parameters?

Entropic risk

- Finite-horizon entropic risk at a given risk averseness $\theta > 0$:

$$r_{k,\sigma^2}(\theta) = \frac{2\sigma^2}{\theta} \log \mathbb{E}[e^{\frac{\theta}{2\sigma^2} f(x_k) - f(x_*)}]$$

- Infinite-horizon entropic risk:

$$r_{\sigma^2}(\theta) = \limsup_{k \to \infty} r_{k,\sigma^2}(\theta)$$

- First-order expansion in θ:

$$r_{k,\sigma^2}(\theta) = \mathbb{E}[f(x_k) - f(x_*)] + \frac{\theta}{4\sigma^2} \mathbb{E}[|f(x_k) - f(x_*)|^2] + o(\theta)$$

- Chernoff bound

$$\mathbb{P}\left\{ f(x_k) - f(x_*) \geq \mathbf{r_{k,\sigma^2}}(\theta) + \frac{\mathbf{2\sigma^2}}{\theta} \log(\mathbf{1}/\zeta) \right\} \leq \zeta$$

where $\zeta \in (0, 1)$ is the confidence level.

Results

- We invent a new Lyapunov function.
- First-time fast deterministic rates $1 - \Theta(\sqrt{\alpha})$ for heavy ball
- First-time rate, entropic risk, tail probability bounds for triple momentum methods for general choice of parameters.
- Show that there are trade-offs between convergence rate and asymptotic risk level.
- We optimally trade-off asymptotic risk and convergence rate.