Robust Formulations and Algorithms for Learning Problems under Distributional Ambiguity

Steve Wright (UW-Madison)

DRDS, August, 2022

1. Distributionally robust classification with Wasserstein ambiguity.

- **Nam Ho-Nguyen**
- “zero-one” classification
- perturbation robustness vs Wasserstein robustness
- robustness and risk
- specialization to linear classification, and “benign nonconvexity” of the resulting formulation.

2. Robust classification, generalized linear programming, and first-order min-max algorithms.

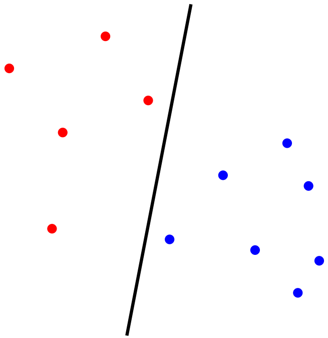
- **Ahmet Alacaoglu, Jelena Diakonikolas, Chaobing Song, Eric Lin**
- The convex-concave min-max paradigm and generalized LP
- Formulating robust classification
- Algorithms
    - Basics
    - PURE-CD
    - CLVR
    - Complexity

Sources

1. Ho-Nguyen, N. and Wright, S. J., “Adversarial classification via distributional robustness with Wasserstein ambiguity,” to *Mathematical Programming Series B*, 2022.
2. Alacaoglu, A., Cevher, V., and Wright, S. J., *On the Complexity of a Practical Primal-Dual Coordinate Method*, arXiv preprint arXiv:2201.07684 (2022).
3. Song, C., Lin, C. Y., Wright, S. J., and Diakonikolas, J., *Coordinate Linear Variance Reduction for Generalized Linear Programming*, arXiv preprint arXiv:2111.01842 (2021).

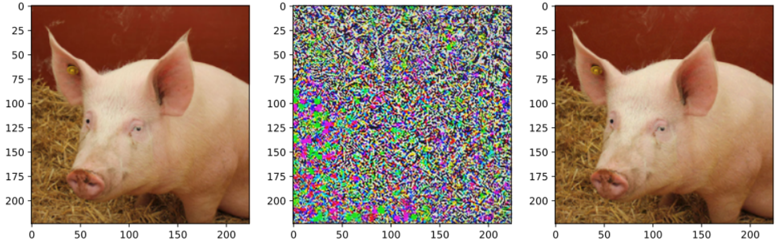
Classification

Consider the **classification** problem: find a **decision boundary** that separates the **red** and **blue** points.



Adversarial classification

Problem in image classification: small perturbations of images can change the classification![1]

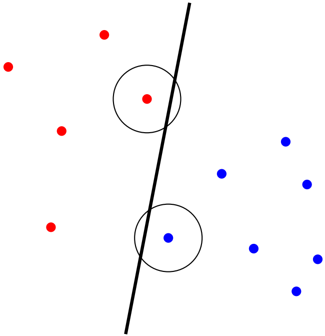


Left: image of pig, classified correctly.
Right: incorrectly classified (wombat) identical pig obtained by adding visually imperceptible noise **(middle)**.

[1] See `https://adversarial-ml-tutorial.org/introduction/` for full details.

Adversarial classification

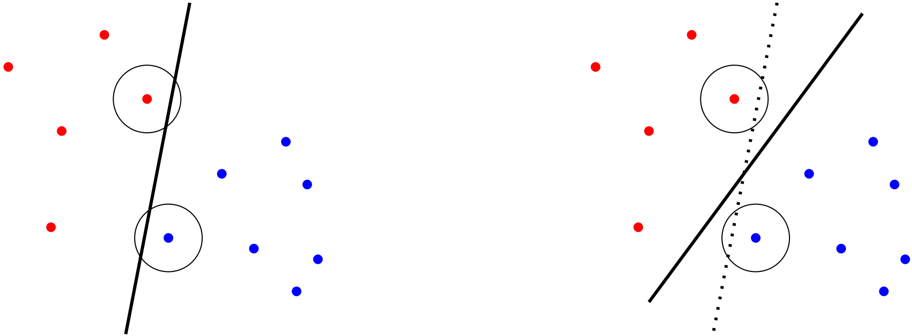
This phenomenon arises because the **points are too close** to the **decision boundary**.



Adversarial classification

This phenomenon arises because the **points are too close** to the **decision boundary**.

We prefer a decision boundary that is **“far away”** from the points.



Adversarial binary classification: Formalities

We have points $(x, y) \in X \times \{\pm 1\}$ distributed according to P.

Seek $f : X \to \mathbb{R}$ such that (**ideally**) $\text{sign}(f(x)) = y$.

- (x, y) is **misclassified** $\iff yf(x) \leq 0$.
- Want f such that $\mathbb{P}_{(x,y)\sim P}[yf(x) \leq 0]$ is small.

Adversarial binary classification: Formalities

We have points $(x, y) \in X \times \{\pm 1\}$ distributed according to P.

Seek $f : X \to \mathbb{R}$ such that (**ideally**) $\mathrm{sign}(f(x)) = y$.

- (x, y) is **misclassified** $\iff yf(x) \leq 0$.
- Want f such that $\mathbb{P}_{(x,y)\sim P}[yf(x) \leq 0]$ is small.

How to account perturbations of points?

Define the **margin** (or **distance to misclassification**)

$$\mathrm{dist}(x, y, f) := \min_{\Delta} \{\|\Delta\| : yf(x + \Delta) \leq 0\}$$

$$(\text{note: } yf(x) \leq 0 \iff \mathrm{dist}(x, y, f) = 0).$$

Choosing classifiers

- Fix $\epsilon > 0$ (defines "how much" we can perturb data points).
- Don't know P, but have i.i.d. samples $(x_i, y_i) \sim P$, $i \in [n]$.
- Let $\hat{\mathbf{P}}_\mathbf{n}$ be the empirical distribution based on these samples.

Perturbation-robust classifier:

$$\text{choose } f \in \mathcal{F} \text{ to minimize } \mathbb{P}_{(x,y)\sim\hat{\mathbf{P}}_\mathbf{n}}[\text{dist}(x, y, f) \le \epsilon].$$

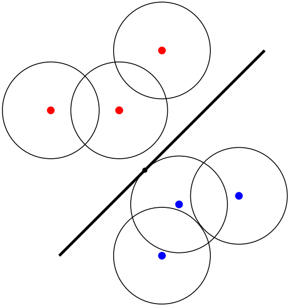
Distributionally robust classifier:

$$\text{choose } f \in \mathcal{F} \text{ to minimize } \max_{d(Q,\hat{\mathbf{P}}_\mathbf{n})\le\epsilon} \mathbb{P}_{(x,y)\sim Q}[yf(x) \le 0]$$

for some distance $d(\cdot, \cdot)$ between distributions.

Perturbation robustness

- Fix $f \in \mathcal{F}$. Perturbation robustness perturbs x_i by Δ_i to misclassify $y_i f(x_i + \Delta_i) \leq 0$.
- Subject to the constraints $\|\Delta_i\| \leq \epsilon$ for all $i \in [n]$.



- Perturbation robust classifier: try to classify the balls of radius ϵ correctly (as much as possible).

Wasserstein robustness

Distributionally robust classifier:

$$\min_{f \in \mathcal{F}} \max_{d(Q, \hat{\mathbf{P}}_{\mathbf{n}}) \leq \epsilon} \mathbb{P}_{(x,y) \sim Q}[y f(x) \leq 0]$$

For distributionally robust classifiers, we claim that **Wasserstein distances** are a natural choice[2]:

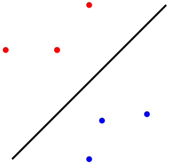
$$d_W(Q, P) = \min_{\Pi} \left\{ \mathbb{E}_{(x,x') \sim \Pi}[\|x - x'\|] : \Pi \text{ has marginals } P_X, Q_X \right\}.$$

[2] Formally this is the 1-Wasserstein distance defined with norm $\| \cdot \|$.

Wasserstein worst-case distributions

Fix a classifier $f \in \mathcal{F}$. We can characterize the **worst-case distribution**[3]

$$Q^* = \arg \max_{d_W(Q,\hat{P}_n) \le \epsilon} \mathbb{P}_{(x,y)\sim Q}[yf(x) \le 0].$$



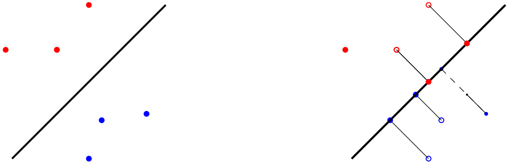
[3] Chen, Kuhn, and Wiesemann *Data-Driven Chance Constrained Programs over Wasserstein Balls* 2018

Wasserstein worst-case distributions

Fix a classifier $f \in \mathcal{F}$. We can characterize the **worst-case distribution**[3]

$$Q^* = \arg \max_{d_W(Q,\hat{P}_n)\le \epsilon} \mathbb{P}_{(x,y)\sim Q}[yf(x) \le 0].$$

- Q^* tries to perturb x_i by Δ_i to misclassify $y_i f(x_i + \Delta_i) \le 0$.
- Subject to constraint $\frac{1}{n}\sum_{i\in[n]} \|\Delta_i\| \le \epsilon$.
- If it cannot transport a whole point, it can “split” a point.



[3] Chen, Kuhn, and Wiesemann *Data-Driven Chance Constrained Programs over Wasserstein Balls* 2018

Why use Wasserstein robustness?

- There are similarities between Wasserstein and perturbation robustness.
- **Question:** are there advantages to Wasserstein robust classifiers over the more common perturbation robust classifiers?

Generalized maximum margin classifiers

- We say that the data $\{(x_i, y_i)\}_{i\in[n]}$ is **separable** if there exists $f \in \mathcal{F}$ such that $\min_{i\in[n]} \operatorname{dist}(x_i, y_i, f) > 0$. The **margin** of a classifier is

$$\gamma(f) := \min_{i\in[n]} \operatorname{dist}(x_i, y_i, f).$$

- The **maximum margin classifier** is the function f that solves

$$\max_{f\in\mathcal{F}} \gamma(f).$$

Generalized maximum margin classifiers

For **(potentially) non-separable data**, we generalize by using a **bilevel formulation**:

The **generalized maximum margin** is defined as follows:

$$\rho^* := \min_{f \in \mathcal{F}} \mathbb{P}_{(x,y)\sim\hat{P}_n}[yf(x) \le 0] \qquad \text{(optimal empirical classification level)}$$

$$\mathcal{F}^* := \arg\min_{f \in \mathcal{F}} \mathbb{P}_{(x,y)\sim\hat{P}_n}[yf(x) \le 0] \qquad \text{(optimal empirical classifiers)}$$

$$\mathcal{I}(f) := \{i \in [n] : \mathrm{dist}(x_i, y_i, f) > 0\} \qquad \text{(correctly classified points)}$$

$$\gamma(f) := \min_{i \in \mathcal{I}(f)} \mathrm{dist}(x_i, y_i, f) > 0 \qquad \text{(margin on correctly classified points)}$$

$$\gamma^* := \max_{f \in \mathcal{F}} \{\gamma(f) : f \in \mathcal{F}^*\}. \qquad \text{(max. margin on optimal classifiers)}$$

Wasserstein vs perturbation robustness

Minimizing Wasserstein worst-case error:

- When $\epsilon < \gamma^*/n$, classifier is **guaranteed** to maximize the generalized margin.
- Correctly classified points can be safely perturbed up to threshold γ^*.

Minimizing perturbation robust error:

- When $\epsilon < \gamma^*$, classifier is **guaranteed** to have margin $\gamma(f) \geq \epsilon$.
- Correctly classified points can be safely perturbed up to threshold $\epsilon < \gamma^*$.
- Need to choose ϵ **as close as possible** to γ^*.

For "small" ϵ, Wasserstein robustness is advantageous.

Robustness and risk

What happens when ϵ is not “small enough”? We can frame robustness in terms of **tail risk measures**.

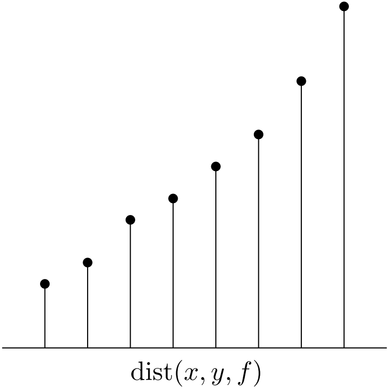
(Think of $n = 100$, and $\rho = 0.05$.)

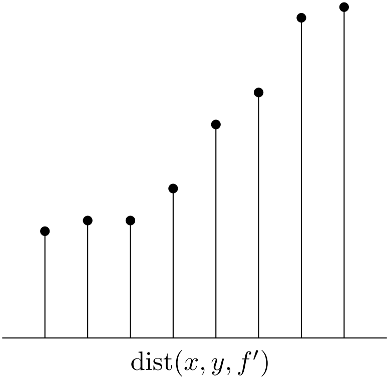
$$\begin{aligned}
&\mathrm{VaR}_\rho(\mathrm{dist}(x, y, f); \hat{P}_n) \\
&\quad := \sup_v \left\{ v : \mathbb{P}_{(x,y)\sim\hat{P}_n} [\mathrm{dist}(x, y, f) < v] \leq \rho \right\} \\
&\quad (= \text{the 5th-smallest margin } \mathrm{dist}(x_i, y_i, f))
\end{aligned}$$

$$\begin{aligned}
&\mathrm{CVaR}_\rho(\mathrm{dist}(x, y, f); \hat{P}_n) \\
&\quad := \sup_t \left\{ t + \frac{1}{\rho} \mathbb{E}_{(x,y)\sim\hat{P}_n} [\min\{0, \mathrm{dist}(x, y, f) - t\}] \right\} \\
&\quad (\approx \text{average of the 5 smallest margins}).
\end{aligned}$$

Robustness and risk

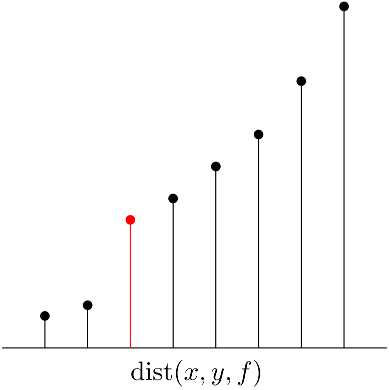
($n = 8$, $\rho = 3/8$.) Margins $\mathrm{dist}(x_i, y_i, f)$, $i \in [n]$. (Larger is better.)

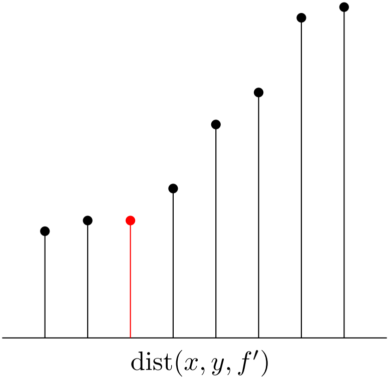




Robustness and risk

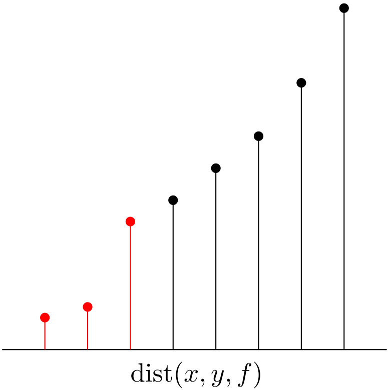
($n = 8$, $\rho = 3/8$.) VaR: third smallest margin. (Larger is better.)

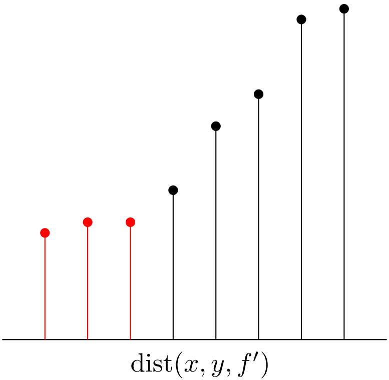




Robustness and risk

($n = 8$, $\rho = 3/8$.) CVaR: average of three smallest margins. (Larger is better.)





Robustness and risk

Lemma

For $\rho \in (0,1)$, $\epsilon > 0$,

$$\mathbb{P}_{(x,y)\sim\hat{P}_n}[\text{dist}(x,y,f) \le \epsilon] \le \rho$$
$$\iff \text{VaR}_{\rho}(\text{dist}(x,y,f); \hat{P}_n) \ge \epsilon$$

$$\max_{d_W(Q,\hat{P}_n)\le\epsilon} \mathbb{P}_{(x,y)\sim Q}[\text{dist}(x,y,f) = 0] \le \rho$$
$$\iff \rho\, \text{CVaR}_{\rho}(\text{dist}(x,y,f); \hat{P}_n) \ge \epsilon$$

Robustness and risk

Theorem

Fix $\rho \in (0,1)$, *set*

$$\epsilon_1 := \max_{f \in \mathcal{F}} \mathrm{VaR}_\rho(\mathrm{dist}(x,y,f); \hat{P}_n)$$

$$\epsilon_2 := \max_{f \in \mathcal{F}} \rho\, \mathrm{CVaR}_\rho(\mathrm{dist}(x,y,f); \hat{P}_n).$$

Then

$$\rho = \min_{f \in \mathcal{F}} \mathbb{P}_{(x,y)\sim\hat{P}_n}[\mathrm{dist}(x,y,f) \le \epsilon_1] \qquad \textit{(perturbation robustness)}$$

$$= \min_{f \in \mathcal{F}} \max_{d_W(Q,\hat{P}_n)\le\epsilon_2} \mathbb{P}_{(x,y)\sim Q}[\mathrm{dist}(x,y,f) = 0]. \qquad \textit{(Wasserstein robustness)}$$

The **type of robustness** (Wasserstein vs perturbation) simply **changes the risk measure**.

The **level of robustness** ϵ and the **risk level** ρ are closely related.

Wasserstein vs perturbation robustness

- For "small" ϵ, Wasserstein robustness is advantageous.
- For arbitrary ϵ:
  - Rigorous theory developed that shows:
    - minimizing perturbation robust error equivalent to maximizing value-at-risk of margin.
    - minimizing Wasserstein robust error equivalent to maximizing **conditional** value-at-risk of margin.
  - Do we only want high proportion of "safe" points? Use perturbation robustness.
  - Or do we want to make "potentially unsafe" points hard to perturb as well? Use Wasserstein robustness.

DRO: Reformulation for linear classifiers

We now consider the class of linear classifiers:

$$\mathcal{F} = \left\{ x \mapsto w^\top x + b : w \in \mathbb{R}^d, b \in \mathbb{R} \right\}.$$

Then

$$\operatorname{dist}(x, y, (w, b)) = \frac{\max\{0, y(w^\top x + b)\}}{\|w\|_*}.$$

Reformulation for linear classifiers

Theorem

We can reformulate[a]

$$\min_{(w,b)\in\mathcal{F}} \max_{d_W(Q,\hat{P}_n)\leq\epsilon} \mathbb{P}_{(x,y)\sim Q}[yf(x) \leq 0]$$

$$\equiv \min_{w,b} \left\{ \epsilon\|w\|_* + \frac{1}{n}\sum_{i\in[n]} L_R(y_i(w^\top x_i + b)) \right\}$$

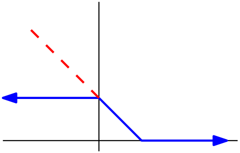
where $L_R(r) := \max\{0, 1-r\} - \max\{0, -r\}$ *is a non-convex ramp loss.*

[a] The proof uses the dual representation for the Wasserstein distance.

Blanchet and Murthy *Quantifying distributional model risk via optimal transport* 2019
Chen, Kuhn, and Wiesemann *Data-driven chance constrained programs over Wasserstein balls* 2018
Gao and Kleywegt *Distributionally robust stochastic optimization with Wasserstein distance* 2016
Xie *On distributionally robust chance constrained programs with Wasserstein distance* 2019

Solving for Wasserstein robust linear classifiers

$$\min_{w,b} \left\{ \epsilon \|w\|_* + \frac{1}{n} \sum_{i \in [n]} L_R(y_i(w^\top x_i + b)) \right\}.$$



- Solve by mixed-integer programming (highly non-scalable).
- First-order based approaches.

Solving for Wasserstein robust linear classifiers

Approximate ramp loss by **smooth function** ψ_σ:

$$\begin{aligned} L_R(r) &= \max\{0, 1-r\} - \max\{0, -r\} \\ \approx \psi_\sigma(r) &:= \sigma \log\left(1 + \exp\left(\frac{1-r}{\sigma}\right)\right) - \sigma \log\left(1 + \exp\left(\frac{-r}{\sigma}\right)\right). \end{aligned}$$

Use ℓ_2-norm $\|\cdot\| = \|\cdot\|_2$, replace with squared norm:

$$\min_{w,b} \left\{ \tfrac{1}{2}\epsilon \|w\|_2^2 + \frac{1}{n} \sum_{i \in [n]} \psi_\sigma(y_i(w^\top x_i + b)) \right\}.$$

Numerical experiments

Data generation:

- $x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$ for $i \in [n]$.
- $x_i \sim N(0, 10I)$ or $N(0, \Sigma)$ $(\mathrm{cond}(\Sigma) = 10)$ or Laplace$(0, 10I)$.
- Fix some unit vector w^*, set $y_i = \text{sign}\left((w^*)^\top x_i\right)$ for all $i \in [n]$.
- **Adversarial perturbations:** generate separable data, replace $\kappa n/2$ points (x_i, y_i) with $(w^*)^T x_i > 0$ (label $y_i = +1$) with points further from the boundary with wrong label:

$$x_i' = x_i + \left((w^*)^\top x_i + 1\right) w^*, \quad y_i' = -1.$$

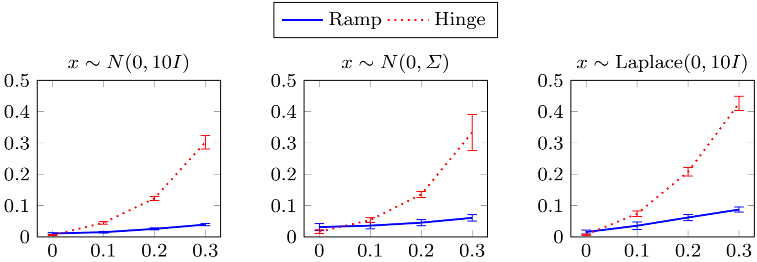
Objective function: set $\epsilon = 0.1$ (regularization), $\sigma = 0.05$ (smoothing).

Algorithms: nonlinear conjugate gradient (CG), L-BFGS, Newton's method with diagonal damping. Behavior was similar, so we display results for CG only.

Robustness: ramp vs hinge

We compare regularized ramp loss reformulation vs hinge loss classifiers $L_H(r) = \max\{0, 1-r\}$ (support vector machines).

- Fix $d = 10$. Generate $n = 10,000$ training points with different adversarial parameter κ (horizontal axis).
- Train both classifiers. Generate $100,000$ test points and compute the misclassification error (vertical axis).



Hinge (DRO formulation) much more robust to mislabelling.

Local minimizers – empirical

Empirically: As n grows, the number of local minimizers decreases.

For given d, how large does n have to be to eliminate multiple local minimizers?

Distribution \ d	5	10	20	40
$N(0, 10I)$	800	1600	1600	6400
$N(0, \Sigma)$	1600	1600	3200	6400
$\mathrm{Laplace}(0, 10I)$	1600	1600	6400	12800

Table Approximate training set size n for a problem with dimension d to have a single (global) minimizer, empirically determined.

Local minimizers – theory

Definition

We say that a random variable x is **spherically symmetric about** 0 if we can write $x = r \cdot s$, where r is a random variable on $\mathbb{R}_+$ and s is a uniform random variable on the unit sphere $\{s \in \mathbb{R}^d : \|s\|_2 = 1\}$, with r and s independent.

Spherically symmetric distributions include normal distributions, Student's t-distributions and Laplace distributions with identity covariance.

Local minimizers – theory

Let

$$\begin{aligned} F_\epsilon(w) &= \tfrac{1}{2}\epsilon\|w\|_2^2 + \mathbb{E}_{x,y}\left[L_R\left(y\left(w^\top x\right)\right)\right] \\ &= \tfrac{1}{2}\epsilon\|w\|_2^2 + \lim_{n\to\infty}\frac{1}{n}\sum_{i\in[n]} L_R\left(y_i\left(w^\top x_i\right)\right). \end{aligned}$$

Recall $y = \text{sign}((w^*)^\top x)$. When $w \neq 0$, we have

$$\nabla F(w) = \epsilon w - \mathbb{E}_{x\sim E}\left[\mathbf{1}(0 \le w^\top x \le 1, (w^*)^\top x \ge 0)x\right].$$

Want to show: $\nabla F(w) = 0$ only when w is a positive multiple of w^*.

Local minimizers – theory

Theorem

Suppose that x is spherically symmetric about 0, and $y = \mathrm{sign}((w^*)^\top x)$. Then for w that is not a **positive multiple** *of w^*, we have $\nabla F_\epsilon(w) \neq 0$. Furthermore, there is a* **single stationary point** *of the form $w(\epsilon) = \alpha(\epsilon) w^*$, for a unique $\alpha(\epsilon) > 0$.*

Note that at $w = 0$, $F_\epsilon(w)$ is non-smooth. We can show that there is a direction of descent in the w^*-direction.

Proved with a nice argument based on geometry of the region

$$\mathcal{R} = \left\{ x : 0 \le w^\top x \le 1, (w^*)^\top x \ge 0 \right\}.$$

Summary of Part 1

- Wasserstein robustness has **favourable properties** compared to perturbation robustness.
- Optimization is **essentially a regularized ramp loss empirical risk minimization** problem.
  - Previous links between regularization and robustness have been studied.[4]

  DRO reformulation gives rise to **loss-regularizer pairs**.
- **Non-convexity** of the ramp loss is **provably benign** for a class of distributions, meaning we can use first-order methods to find the global minimum.

[4] See, e.g.,
Xu and Mannor *Robustness and regularization of support vector machines* 2009
Bertsimas and Copenhaver *Characterization of the equivalence of robustification and regularization in linear and matrix regression* 2018
Shafieezadeh-Abadeh, Kuhn and Mohajerin Esfahani *Regularization via mass transportation* 2019

Part 2: Robust classification, generalized LP, first-order min-max algorithms

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} L(x,y) \tag{Min-Max}$$

where

$$\begin{aligned} L(x,y) &= \sum_{i=1}^{n} \left[\langle A_i x, y_i \rangle - h_i^*(y_i) \right] + g(x) \\ &= \langle Ax, y \rangle - h^*(y) + g(x), \end{aligned}$$

- $h_i^* : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is convex conjugate of h_i:
$$h_i^*(t) := \sup_s (st - h_i(s))$$
(convex and extended-valued);
- $h^*(y) = \sum_{i=1}^n h_i^*(y_i)$ (separable);
- $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ (convex and extended-valued);
- $A_i \in \mathbb{R}^d$ is a row vector; A is the $n \times d$ matrix with rows A_i.

More Specs

We consider cases in which A is dense and A is sparse.

In the case of sparse A, we assume for analysis that g is separable, that is,

$$g(x) = \sum_{j=1}^{d} g_j(x_j).$$

All algorithms make use of the prox-operator denoted for diagonal weighting matrix $\mathrm{T} \succ 0$ and function g by $\mathrm{prox}_{\mathrm{T},g}$ and defined

$$\begin{aligned}\mathrm{prox}_{\mathrm{T},g}(x) &:= \arg\min_u \tfrac{1}{2}\|u - x\|^2_{\mathrm{T}^{-1}} + g(u) \\ &= \arg\min_u \frac{1}{2} \sum_{i=1}^{d} \frac{(x_i - u_i)^2}{\mathrm{T}_{ii}} + g(u).\end{aligned}$$

Assume that we can compute prox-operators for g and h_i^* "easily."

Generalized LP

$$\min\; c^T x + r(x) \;\text{ s.t. } Ax = b,\; x \in \mathcal{X}, \qquad \text{(GLP)}$$

which can be written in min-max form as

$$\min_{x\in\mathcal{X}} \max_{y\in\mathbb{R}^n} L(x,y) = \langle Ax, y\rangle + c^T x + r(x) - b^T y,$$

which is unconstrained and linear in y.

$\mathcal{X} \subset \mathbb{R}^d$ is closed and convex, r is convex. We assume that the following modified prox-operator is easy to compute:

$$\text{prox}_{\mathcal{X},r}(\hat{x}) := \arg\min_{z\in\mathcal{X}} \tfrac{1}{2}\|z-\hat{x}\|_2^2 + r(z).$$

- Ordinary LP: $\mathcal{X} = \mathbb{R}^d_{\geq 0}$ and $r(\cdot) = 0$
- Approximate Dynamic Programming [De Farias and Van Roy, 2003]
- Optimal Transport [Villani, 2009]
- DRO (f-divergence, Wasserstein) (see below)
- relaxed Neural Net verification [Liu et al., 2020].

GLP formulation of the DRO: Wasserstein 1-norm

Setup: sample vectors $\{a_1, a_2, \dots, a_n\}$ in $\mathbb{R}^d$ with labels $\{b_1, b_2, \dots, b_n\}$, where $b_i \in \{1, -1\}$. Usual ERM problem is

$$\min_w \frac{1}{n} \sum_{i=1}^{n} h(b_i a_i^T w)$$

where $h : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is convex (e.g. hinge loss).

- Wasserstein metric defines a distance between distributions $\mathbb{P}$ and $\mathbb{Q}$ over $\mathbb{R}^d \times \{-1, 1\}$, based on cost

  $$\zeta((a,b),(a',b')) = \|a - a'\|_1 + \kappa |b - b'|$$

  for some $\kappa > 0$;
- $\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{(a_i,b_i)}$ is the empirical distribution defined by the data;
- Seek sup of the objective over the ball of radius ϵ around $\mathbb{P}_n$ (in space of distributions over (a, b)) defined by the Wasserstein metric:

$$\min_{w \in \mathbb{R}^d} \sup_{\mathrm{dist}(\mathbb{P},\mathbb{P}_n) \le \epsilon} \mathbb{E}_{\mathbb{P}}[h(ba^T w)].$$

GLP formulation of the DRO: Wasserstein 1-norm

$$
\begin{aligned}
\min_{w,\lambda,u,v,s,t} \quad & \epsilon\lambda + \frac{1}{n}\sum_{i=1}^{n} s_i \\
\text{s.t.} \quad & u_i = b_i a_i^T w, && i = 1, 2, \dots, n, \\
& v_i = -u_i, && i = 1, 2, \dots, n, \\
& t_i = 2\kappa\lambda + s_i, && i = 1, 2, \dots, n, \\
& h(u_i) \le s_i, && i = 1, 2, \dots, n, \\
& h(v_i) \le t_i, && i = 1, 2, \dots, n, \\
& \|w\|_\infty \le \lambda / M.
\end{aligned}
$$

$\mathcal{X}$ is defined by the last 3 constraints. The corresponding prox operation is separable so can be implemented easily.

See [Song et al., 2021a, Appendix C.2].

GLP formulation of DRO: f-divergence

$$\min_{x\in\mathcal{X}} \sup_{p\in\mathcal{P}_{\epsilon,n}} \sum_{i=1}^{n} p_i g(b_i(a_i^T x)),$$

where

- $\mathcal{P}_{\epsilon,n} = \left\{p \in \mathbb{R}_+^n : \sum_{i=1}^n p_i = 1,\ D_f(p\|\mathbf{1}/n) \le \frac{\epsilon}{n}\right\}$ is the ambiguity set,
- g is a convex loss function,
- D_f is an f-divergence defined by $D_f(p\|q) = \sum_{i=1}^n q_i f(p_i/q_i)$ with $p, q \in \left\{p \in \mathbb{R}_+^n : \sum_{i=1}^n p_i = 1\right\}$ and f being a convex function [Namkoong and Duchi, 2016].

GLP formulation of DRO: f-divergence

When $\mathcal{X}$ is a (simple) compact convex set, the DRO problem with f-divergence is equivalent to the following generalized LP:

$$
\begin{aligned}
\min_{x,u,v,w,\mu,q,\gamma} \quad & \left\{\gamma + \frac{\epsilon\mu_1}{n} + \frac{1}{n}\sum_{i=1}^{n} \mu_i f^*\left(\frac{q_i}{\mu_i}\right)\right\} & \\
s.t. \quad & w + v - \frac{q}{n} - \gamma \mathbf{1}_n = \mathbf{0}_n, & \\
& u_i = b_i a_i^T x, & i = 1, 2, \dots, n, \\
& \mu_1 = \mu_2 = \cdots = \mu_n, & \\
& g(u_i) \le w_i, & i = 1, 2, \dots, n, \\
& q_i \in \mu_i \operatorname{dom}(f^*), & i = 1, 2, \dots, n, \\
& v_i \ge 0, \ \mu_i \ge 0, & i = 1, 2, \dots, n, \\
& x \in \mathcal{X}. &
\end{aligned}
$$

See [Song et al., 2021a, Section 4].

Basic Algorithms

$$\begin{aligned}\bar{x}^{k+1} &= \text{prox}_{\tau,g}(\bar{x}^k - \tau A^\top \bar{y}^k)\\ \bar{y}^{k+1} &= \text{prox}_{\sigma,h^*}(\bar{y}^k + \sigma A \bar{x}^{k+1}),\end{aligned} \tag{GDA}$$

for positive step sizes τ and σ.

Primal-Dual Hybrid Gradient (PDHG) [Chambolle and Pock, 2011] uses extrapolation in the x step:

$$\begin{aligned}\bar{x}^{k+1} &= \text{prox}_{\tau,g}(\bar{x}^k - \tau A^\top (2\bar{y}^k - \bar{y}^{k-1}))\\ \bar{y}^{k+1} &= \text{prox}_{\sigma,h^*}(\bar{y}^k + \sigma A \bar{x}^{k+1}),\end{aligned} \tag{PDHG}$$

Equivalent form of PDHG:

$$\bar{x}^{k+1} = \text{prox}_{\tau,g}(\hat{x}^k - \tau A^\top \bar{y}^k) \tag{1a}$$

$$\bar{y}^{k+1} = \text{prox}_{\sigma,h^*}(\bar{y}^k + \sigma A \bar{x}^{k+1}) \tag{1b}$$

$$\hat{x}^{k+1} = \bar{x}^{k+1} - \tau A^\top (\bar{y}^{k+1} - \bar{y}^k). \tag{1c}$$

Related to to Douglas-Rachford, Extrapolated gradient, ADMM.

Algorithms: Additional Features

Theoretical convergence / complexity properties of these algorithms can be improved (in some cases, including strong convexity / concavity and sparsity) by adding extra features.

- Coordinate descent: e.g. update random element(s) of y in (1b) instead of the whole vector.
- Variance Reduction: Adjust the update formula for x to account for noise arising from coordinate update of y.
- Dual Averaging: At step k, use a gradient term that is a weighted average over all previous iterations.
- Importance sampling: Apply different weights to different components of each update (e.g. weight matrix T in definition of prox).
- Iterate averaging: Output a weighted average of iterates, rather than the final iterate for x.

Some are used by PURE-CD, VRPDA^2, and CLVR.

Complexity Analysis

Find upper bounds on the number of flops needed to reduce (expected) measures of "primal-dual gap" below a given threshold $\varepsilon > 0$. Particularly interested in dependence on ε as well as

- Dimensions d (for primal x) and n (for dual y);
- size of A: e.g. $\|A\|$, $\max_{i=1,2,\dots,n} \|A_i\|$, or $\sum_{i=1}^n \|A_i\|$;
- $\mathrm{nnz}(A)$ (for sparse A);
- Distance between (x^0, y^0) and the optimum $(x^\star, y^\star)$.

Some algorithms (e.g. stochastic PDHG [Chambolle et al., 2018]) have less impressive bounds yet perform well for some types of problems.

PURE-CD: Sparse A [Alacaoglu et al., 2020]

Define notation $J(i) := \{j \in [d] \colon A_{i,j} \neq 0\}$

Assume that g is separable: $g(x) = \sum_{j=1}^{d} g_j(x_j)$.

```
Initialize $x_0 \in \mathrm{dom}\, g, y_0 \in \mathrm{dom}\, h^*$;
for $k \geq 0$ do
    Pick $i_k \in [n]$ with $\Pr(i_k = i) = \frac{1}{n}$
    $\left[\bar{x}^{k+1} = \mathrm{prox}_{\tau^k, g}\left(x^k - \tau^k (A^\top y^k)\right)\right]_{J(i_k)}$
    $\left[y^{k+1} = \mathrm{prox}_{\sigma^k, h^*}(y^k + \sigma^k A \bar{x}^{k+1})\right]_{i_k}$;    $\left[y^{k+1} = y^k\right]_{\setminus i_k}$
    $\left[x^{k+1} = \bar{x}^{k+1} - \tau^k \theta^k A_{i_k}^T (y_{i_k}^{k+1} - y_{i_k}^k)\right]_{J(i_k)}$; $\left[x^{k+1} = x^k\right]_{\setminus J(i_k)}$
end for
```

Notation:

- $[\cdot]_J$ means that the formula is executed on only the components indexed by the set J.
- $[\cdot]_{\setminus J}$ means that the formula is executed on all components *except* those indexed by the set J.

PURE-CD: Sparse A [Alacaoglu et al., 2020]

Define notation $J(i) := \{j \in [d] \colon A_{i,j} \neq 0\}$

Assume that g is separable: $g(x) = \sum_{j=1}^{d} g_j(x_j)$.

```
Initialize $x_0 \in \mathrm{dom}\, g, y_0 \in \mathrm{dom}\, h^*$;
for $k \geq 0$ do
    Pick $i_k \in [n]$ with $\Pr(i_k = i) = \frac{1}{n}$
    $\left[\bar{x}^{k+1} = \mathrm{prox}_{\tau^k, g}\left(x^k - \tau^k (A^\top y^k)\right)\right]_{J(i_k)}$
    $\left[y^{k+1} = \mathrm{prox}_{\sigma^k, h^*}(y^k + \sigma^k A\bar{x}^{k+1})\right]_{i_k}$;    $\left[y^{k+1} = y^k\right]_{\setminus i_k}$
    $\left[x^{k+1} = \bar{x}^{k+1} - \tau^k \theta^k A_{i_k}^T (y_{i_k}^{k+1} - y_{i_k}^k)\right]_{J(i_k)}$; $\left[x^{k+1} = x^k\right]_{\setminus J(i_k)}$
end for
```

Notation:

- $[\cdot]_J$ means that the formula is executed on only the components indexed by the set J.
- $[\cdot]_{\setminus J}$ means that the formula is executed on all components *except* those indexed by the set J.

PURE-CD: Sparse A [Alacaoglu et al., 2020]

Define notation $J(i) := \{j \in [d] \colon A_{i,j} \neq 0\}$

Assume that g is separable: $g(x) = \sum_{j=1}^{d} g_j(x_j)$.

```
1: Initialize $x_0 \in \operatorname{dom} g, y_0 \in \operatorname{dom} h^*$;
2: for $k \geq 0$ do
3:     Pick $i_k \in [n]$ with $\Pr(i_k = i) = \frac{1}{n}$
4:     $\left[\bar{x}^{k+1} = \operatorname{prox}_{\tau^k, g}\left(x^k - \tau^k (A^\top y^k)\right)\right]_{J(i_k)}$
5:     $\left[y^{k+1} = \operatorname{prox}_{\sigma^k, h^*}(y^k + \sigma^k A\bar{x}^{k+1})\right]_{i_k}$;     $\left[y^{k+1} = y^k\right]_{\setminus i_k}$
6:     $\left[x^{k+1} = \bar{x}^{k+1} - \tau^k \theta^k A_{i_k}^T (y_{i_k}^{k+1} - y_{i_k}^k)\right]_{J(i_k)}$; $\left[x^{k+1} = x^k\right]_{\setminus J(i_k)}$
7: end for
```

Notation:

- $[\cdot]_J$ means that the formula is executed on only the components indexed by the set J.
- $[\cdot]_{\setminus J}$ means that the formula is executed on all components *except* those indexed by the set J.

PURE-CD: Sparse A [Alacaoglu et al., 2020]

Define notation $J(i) := \{j \in [d] \colon A_{i,j} \neq 0\}$

Assume that g is separable: $g(x) = \sum_{j=1}^{d} g_j(x_j)$.

```
Initialize $x_0 \in \operatorname{dom} g, y_0 \in \operatorname{dom} h^*$;
for $k \geq 0$ do
    Pick $i_k \in [n]$ with $\Pr(i_k = i) = \frac{1}{n}$
    $\left[\bar{x}^{k+1} = \operatorname{prox}_{\tau^k, g}\left(x^k - \tau^k (A^\top y^k)\right)\right]_{J(i_k)}$
    $\left[y^{k+1} = \operatorname{prox}_{\sigma^k, h^*}(y^k + \sigma^k A \bar{x}^{k+1})\right]_{i_k}$;    $\left[y^{k+1} = y^k\right]_{\setminus i_k}$
    $\left[x^{k+1} = \bar{x}^{k+1} - \tau^k \theta^k A_{i_k}^T (y_{i_k}^{k+1} - y_{i_k}^k)\right]_{J(i_k)}$; $\left[x^{k+1} = x^k\right]_{\setminus J(i_k)}$
end for
```

Notation:

- $[\cdot]_J$ means that the formula is executed on only the components indexed by the set J.
- $[\cdot]_{\setminus J}$ means that the formula is executed on all components *except* those indexed by the set J.

PURE-CD Sparse: Complexity Results for Min-Max

Focus on results where strong convexity is present in g and/or h^* (both separable functions).

- Each g_j has modulus of convexity $\mu_g \geq 0$;
- Each h_i^* has modulus of convexity $\mu_h \geq 0$,

Results are for last iterates x^K and/or y^K, not averaged iterates.

When $\mu_g > 0$ and $\mu_h > 0$, we have $\mathbb{E}\left[\|x^K - x^\star\|^2 + \|y^K - y^\star\|^2\right] \leq \varepsilon$ with expected complexity [5]

$$\tilde{O}\left(\mathrm{nnz}(A)\frac{\max_i \|A_i\|}{\sqrt{\mu_h \mu_g}} \log \varepsilon^{-1}\right).$$

Choices of Θ_k, σ_i^k, T_k do not depend on k, but require knowledge of μ_g and μ_h

[5] assuming $\max_i \|A_i\| \geq \sqrt{\mu_h \mu_g}$

PURE-CD Sparse: Complexity Results for Min-Max

When $\mu_g > 0$ but possibly $\mu_h = 0$ (strong convexity in g only) can make a (complicated) choice of parameters to ensure that $\mathbb{E}\left[\|x^K - x^\star\|^2\right] \le \varepsilon$ with expected complexity

$$O\left(\mathrm{nnz}(A)\sqrt{\frac{D_\star}{\varepsilon}}\frac{\max_i \|A_i\|}{\mu_g}\right),$$

When $\mu_h > 0$ but possibly $\mu_g = 0$ (strong convexity in h only) a different (still complicated) choice of parameters σ_j^k, τ_j^k, Θ_k ensures that $\mathbb{E}\left[\|y^K - y^\star\|^2\right] \le \varepsilon$ with expected complexity

$$O\left(\mathrm{nnz}(A)\sqrt{\frac{D_\star}{\varepsilon}}\frac{\max_i \|A_i\|}{\mu_h}\right),$$

Here $D_\star$ depends on (x^0, y^0) and (x^*, y^*).

Complexity Comparisons

The PURE-CD complexity bounds are compared with various other algorithms for Min-Max, or special cases of it:

- PDHG [Chambolle and Pock, 2011]
- SPDHG [Chambolle et al., 2018]
- VRPDA [Song et al., 2021b]
- CLVR [Song et al., 2021a]
- SPDAD [Tan et al., 2020]
- VRVI [Carmon et al., 2019, Alacaoglu and Malitsky, 2022]
- Katyusha [Allen-Zhu, 2017]
- SPDC [Zhang and Lin, 2015]

In each case, PURE-CD matches or improves the complexities of these alternatives, in terms of their dependence on n, d, measures of A, ε.

A typical improvement is $\|A\| \to \max_i \|A_i\|$ – a factor of up to $\sqrt{n}$.

Comments on Proofs

The proofs of these complexity results are extremely technical, involving mostly elementary manipulation of inequalities.

Telescoping sums over iterations $k = 1, 2, \ldots, K$ is used often, and convexity is essential.

But considerable expertise is needed to choose the algorithmic parameters T_k, σ_i^k, Θ_k to achieve the desired cancellations.

CLVR Algorithm for GLP [Song et al., 2021a]

$$\min\ c^T x + r(x) \ \text{ s.t. } Ax = b,\ x \in \mathcal{X}. \qquad \text{(GLP)}$$

Partition A into m row blocks – index partition $\{S^1, S^2, \ldots, S^m\}$.

```
**Input:** $x^0 \in \mathcal{X}, y^0 \in \mathbb{R}^n, z^0 = A^T y^0,\ \gamma > 0, \hat{L} > 0, \sigma \geq 0, K.$
$a_1 = B_1 = \frac{1}{2\hat{L}m}, q^0 = a_1(z^0 + c).$
**for** $k = 1, 2, \ldots, K$ **do**
    $x^k = \text{prox}_{\frac{1}{\gamma}B_k r}(x^0 - \frac{1}{\gamma} q^{k-1}).$
    Pick $j_k$ uniformly at random in $\{1, 2, \ldots, m\}.$
    $[y^k = y^{k-1}]_{\setminus S^{j_k}};\quad [y^k = y^{k-1} + \gamma m a_k(Ax^k - b)]_{S^{j_k}};$
    $a_{k+1} = \frac{\sqrt{1+\sigma B_k/\gamma}}{2\hat{L}m},\ B_{k+1} = B_k + a_{k+1}.$
    $z^k = z^{k-1} + A^T_{S^{j_k}}(y^k_{S^{j_k}} - y^{k-1}_{S^{j_k}}).$
    $q^k = q^{k-1} + a_{k+1}\big(z^k + c\big) + m a_k(z^k - z^{k-1}).$
**end for**
**return** weighted averages $x^K$ and $y^K$.
```

CLVR: Notes and Complexity

Again related to PDHD but with variations. Exploits the fact that the Min-Max formulation is linear and unconstrained in y.

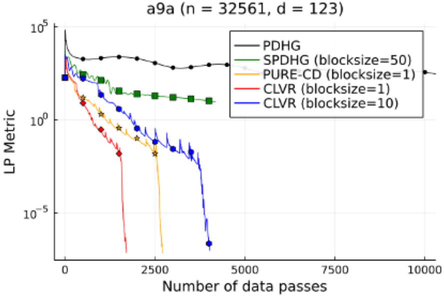
- Averaged gradients in x, block coordinate descent in y.
- Recall that specialized prox-operator involves constraint set $\mathcal{X}$.
- Can be implemented in a way that exploits sparsity in A
  - ....but this involves intermediate vectors and is more complicated than in Sparse PURE-CD.
- No special initialization required (unlike VRPDA^2).

Expected complexity for $\mathbb{E}G(x^K, y^K, x^\star, y^\star) < \varepsilon$ in Sparse CLVR is

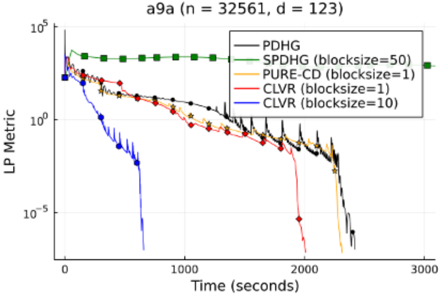
$$O\left(\frac{\mathrm{nnz}(A)\max_{i=1,2,\dots,m}\|A_{S^i}\|}{\varepsilon}\right).$$

Computational Results: Wasserstein DRO

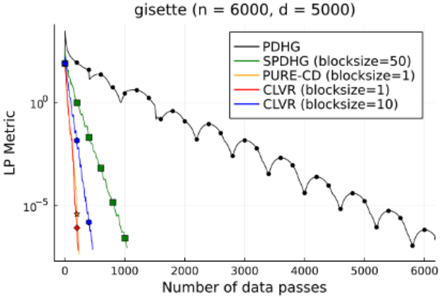
- Wasserstein DRO described above, with ℓ_1 norm and hinge loss.
- Several standard ML datasets (LIBSVM).
- Implemented in **Julia**. Use **SparseArrays** to support sparse vectors and matrices.
- CLVR uses blocks to improve utilization of multiple cores.



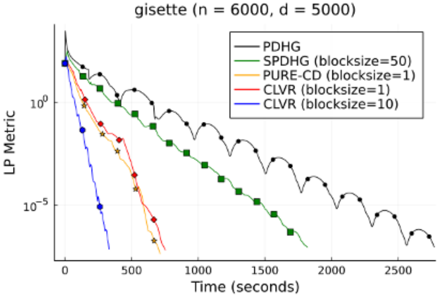
a9a (n = 32561, d = 123)
LP Metric
10^5
10^0
10^{-5}
0
2500
5000
7500
10000
Number of data passes
PDHG
SPDHG (blocksize=50)
PURE-CD (blocksize=1)
CLVR (blocksize=1)
CLVR (blocksize=10)



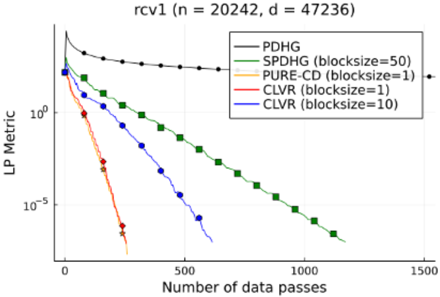
a9a (n = 32561, d = 123)
LP Metric
10^5
10^0
10^{-5}
0
1000
2000
3000
Time (seconds)
PDHG
SPDHG (blocksize=50)
PURE-CD (blocksize=1)
CLVR (blocksize=1)
CLVR (blocksize=10)

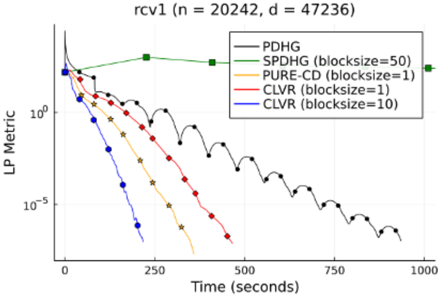


gisette (n = 6000, d = 5000)

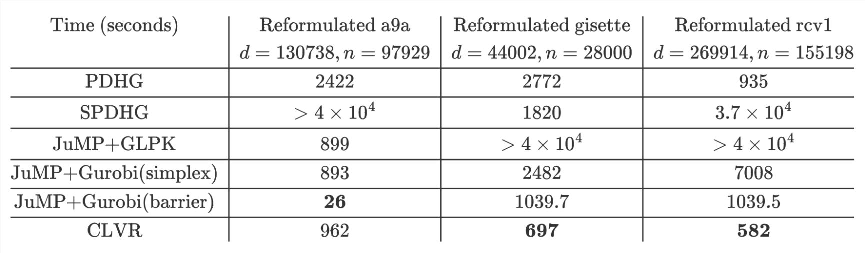


gisette (n = 6000, d = 5000)


rcv1 (n = 20242, d = 47236)
LP Metric
10^0
10^{-5}
0
500
1000
1500
Number of data passes
PDHG
SPDHG (blocksize=50)
PURE-CD (blocksize=1)
CLVR (blocksize=1)
CLVR (blocksize=10)


rcv1 (n = 20242, d = 47236)
LP Metric
10^0
10^{-5}
0
250
500
750
1000
Time (seconds)
PDHG
SPDHG (blocksize=50)
PURE-CD (blocksize=1)
CLVR (blocksize=1)
CLVR (blocksize=10)

Comparing with General LP solvers (times)



Time (seconds)	Reformulated a9a	Reformulated gisette	Reformulated rcv1
	$d = 130738, n = 97929$	$d = 44002, n = 28000$	$d = 269914, n = 155198$
PDHG	2422	2772	935
SPDHG	$> 4 \times 10^4$	1820	3.7×10^4
JuMP+GLPK	899	$> 4 \times 10^4$	$> 4 \times 10^4$
JuMP+Gurobi(simplex)	893	2482	7008
JuMP+Gurobi(barrier)	**26**	1039.7	1039.5
CLVR	962	**697**	**582**

Summary of Part 2

- Generalized LP is a nice framework for DRO classificaition with linear models.
- Generalized LP are a special case of convex-concave saddle point problems with bilinear coupling, therefore admit the use of powerful first-order methods such as PURE-CD and CLVR.
- The resulting computational approach may be advantageous on problems of extreme scale.

References I

Alacaoglu, A., Fercoq, O., and Cevher, V. (2020).
Random extrapolation for primal-dual coordinate descent.
In *International Conference on Machine Learning*, pages 191–201. PMLR.

Alacaoglu, A. and Malitsky, Y. (2022).
Stochastic variance reduction for variational inequality methods.
In *Conference on Learning Theory*, pages 778–816. PMLR.

Allen-Zhu, Z. (2017).
Katyusha: The first direct acceleration of stochastic gradient methods.
The Journal of Machine Learning Research, 18(1):8194–8244.

Carmon, Y., Jin, Y., Sidford, A., and Tian, K. (2019).
Variance reduction for matrix games.
Advances in Neural Information Processing Systems.

Chambolle, A., Ehrhardt, M. J., Richtárik, P., and Schonlieb, C.-B. (2018).
Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications.
SIAM Journal on Optimization, 28(4):2783–2808.

Chambolle, A. and Pock, T. (2011).
A first-order primal-dual algorithm for convex problems with applications to imaging.
Journal of Mathematical Imaging and Vision, 40(1):120–145.

References II

De Farias, D. P. and Van Roy, B. (2003).
The linear programming approach to approximate dynamic programming.
Operations research, 51(6):850–865.

Fercoq, O. and Bianchi, P. (2019).
A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions.
SIAM Journal on Optimization, 29(1):100–134.

Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M. J., et al. (2020).
Algorithms for verifying deep neural networks.
Foundations and Trends® in Optimization, 4.

Namkoong, H. and Duchi, J. C. (2016).
Stochastic gradient methods for distributionally robust optimization with f-divergences.
In *Proc. NIPS'16*.

Nesterov, Y. (2004).
Introductory Lectures on Convex Optimization: A Basic Course.
Springer Science and Business Media, New York.

Song, C., Lin, C. Y., Wright, S. J., and Diakonikolas, J. (2021a).
Coordinate linear variance reduction for generalized linear programming.
arXiv preprint arXiv:2111.01842.

References III

Song, C., Wright, S. J., and Diakonikolas, J. (2021b).
Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums.
In *International Conference on Machine Learning*, pages 9824–9834. PMLR.

Tan, C., Qian, Y., Ma, S., and Zhang, T. (2020).
Accelerated dual-averaging primal–dual method for composite convex minimization.
Optimization Methods and Software, 35(4):741–766.

Villani, C. (2009).
Optimal transport: old and new, volume 338.
Springer.

Zhang, Y. and Lin, X. (2015).
Stochastic primal-dual coordinate method for regularized empirical risk minimization.
In *International Conference on Machine Learning*, pages 353–361. PMLR.