Towards a First-Order Algorithmic Framework for Wasserstein Distributionally Robust Optimization

Jiajin Li[12]

[1]The Chinese University of Hong Kong (CUHK)

[2]Stanford University

[Joint work with Caihua Chen, Anthony Man-Cho So, Sen Huang]

INFORMS 2021 Annual Meeting

Outline

Introduction and Motivation

Tractable Conic Reformulation

ADMM-based First-Order Algorithmic Framework

Conclusion and Future Directions

Empirical Risk Minimization

- Training dataset: i.i.d. input-output pairs $\{(\hat{x}_i, \hat{y}_i)\}_{i=1}^N$ drawn from the distribution $\mathbb{P}$;

Empirical Risk Minimization

- Training dataset: i.i.d. input-output pairs $\{(\hat{x}_i, \hat{y}_i)\}_{i=1}^N$ drawn from the distribution $\mathbb{P}$;
- As the true distribution $\mathbb{P}$ is typically not known, one considers the empirical risk minimization (ERM) problem

$$\inf_{\beta} \left\{ \mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N} [\ell(f_\beta(x), y)] = \frac{1}{N} \sum_{i=1}^N \ell(f_\beta(\hat{x}_i), \hat{y}_i) \right\},$$

where

$$\hat{\mathbb{P}}_N := \frac{1}{N} \sum_{i=1}^N \delta_{(\hat{x}_i, \hat{y}_i)}$$

is the empirical distribution associated with the training dataset.

Typical Example

- Logistic regression (LR) — classification problem

Typical Example

- Logistic regression (LR) — classification problem
  - input-output data: $x \in \mathbb{R}^n, y \in \{-1, +1\}$;

Typical Example

- Logistic regression (LR) — classification problem
  - input-output data: $x \in \mathbb{R}^n, y \in \{-1, +1\}$;
  - family of linear functions: $x \to f_\beta(x) := \beta^T x$;

Typical Example

- Logistic regression (LR) — classification problem
  - input-output data: $x \in \mathbb{R}^n, y \in \{-1, +1\}$;
  - family of linear functions: $x \to f_\beta(x) := \beta^T x$;
  - log-loss: $\ell(\mu, \nu) = \log(1 + \exp(-\mu\nu))$;

Typical Example

- Logistic regression (LR) — classification problem
  - input-output data: $x \in \mathbb{R}^n, y \in \{-1, +1\}$;
  - family of linear functions: $x \to f_\beta(x) := \beta^T x$;
  - log-loss: $\ell(\mu, \nu) = \log(1 + \exp(-\mu\nu))$;
  - ERM problem:

$$\inf_{\beta} \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-\hat{y}_i \beta^T \hat{x}_i));$$

Overfitting and Regularization

- A well-known issue with ERM is overfitting.

Overfitting and Regularization

- A well-known issue with ERM is overfitting.
  - estimator $\beta^\star$ works well on training dataset but generalizes poorly $\rightarrow \mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_{\beta^\star}(x),y)]$ and $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f_{\beta^\star}(x),y)]$ are far apart.

Overfitting and Regularization

- A well-known issue with ERM is overfitting.
  - estimator $\beta^\star$ works well on training dataset but generalizes poorly → $\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_{\beta^\star}(x),y)]$ and $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f_{\beta^\star}(x),y)]$ are far apart.
- A standard approach to deal with this is regularization:

$$\inf_{\beta}\left\{\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_\beta(x),y)]+\epsilon R(f_\beta)\right\}.$$

Overfitting and Regularization

- A well-known issue with ERM is overfitting.
  - estimator $\beta^\star$ works well on training dataset but generalizes poorly → $\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_{\beta^\star}(x),y)]$ and $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f_{\beta^\star}(x),y)]$ are far apart.
- A standard approach to deal with this is regularization:

$$\inf_{\beta}\left\{\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_\beta(x),y)]+\epsilon R(f_\beta)\right\}.$$

  - hard to choose the hyperparameter ϵ;

Overfitting and Regularization

- A well-known issue with ERM is overfitting.
  - estimator $\beta^\star$ works well on training dataset but generalizes poorly $\to \mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_{\beta^\star}(x),y)]$ and $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f_{\beta^\star}(x),y)]$ are far apart.
- A standard approach to deal with this is regularization:

$$\inf_{\beta}\left\{\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_\beta(x),y)] + \epsilon R(f_\beta)\right\}.$$

  - hard to choose the hyperparameter ϵ;
  - justification often relies on additional assumptions [Kakade et al., 2009].

Overfitting and Regularization

- A well-known issue with ERM is overfitting.
  - estimator $\beta^\star$ works well on training dataset but generalizes poorly $\rightarrow \mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_{\beta^\star}(x),y)]$ and $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f_{\beta^\star}(x),y)]$ are far apart.
- A standard approach to deal with this is regularization:

$$\inf_{\beta}\left\{\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_N}[\ell(f_\beta(x),y)] + \epsilon R(f_\beta)\right\}.$$

  - hard to choose the hyperparameter ϵ;
  - justification often relies on additional assumptions [Kakade et al., 2009].
- Distributionally robust optimization (DRO) — a fresh perspective on regularization [Shafieezadeh-Abadeh et al., 2019, Namkoong and Duchi, 2017, Gao et al., 2017];

DRO Formulation

- Instead of ERM, consider minimizing the worst-case expected loss

$$\inf_{\beta} \sup_{\mathbb{Q} \in B_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\ell(f_\beta(x), y)], \qquad (\star)$$

where $B_\epsilon(\hat{\mathbb{P}}_N)$, the so-called ambiguity set, is a set of distributions around $\hat{\mathbb{P}}_N$. That is,

$$B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_N) \le \epsilon\},$$

where $D(\cdot, \cdot)$ is a certain probability discrepancy.

DRO Formulation

- Instead of ERM, consider minimizing the worst-case expected loss

$$\inf_{\beta} \sup_{\mathbb{Q} \in B_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\ell(f_\beta(x), y)], \quad (\star)$$

  where $B_\epsilon(\hat{\mathbb{P}}_N)$, the so-called ambiguity set, is a set of distributions around $\hat{\mathbb{P}}_N$. That is,

$$B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_N) \le \epsilon\},$$

  where $D(\cdot,\cdot)$ is a certain probability discrepancy.

- How to choose the probability metric $D(\cdot,\cdot)$? E.g., moment-based/ f-divergence/ Wasserstein distance.

DRO Formulation

- Instead of ERM, consider minimizing the worst-case expected loss

$$\inf_{\beta} \sup_{\mathbb{Q} \in B_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\ell(f_\beta(x), y)], \qquad (\star)$$

  where $B_\epsilon(\hat{\mathbb{P}}_N)$, the so-called ambiguity set, is a set of distributions around $\hat{\mathbb{P}}_N$. That is,

$$B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_N) \le \epsilon\},$$

  where $D(\cdot,\cdot)$ is a certain probability discrepancy.

- How to choose the probability metric $D(\cdot,\cdot)$? E.g., moment-based/ f-divergence/ Wasserstein distance.
  - asymptotic consistency;

DRO Formulation

- Instead of ERM, consider minimizing the worst-case expected loss

$$\inf_{\beta} \sup_{\mathbb{Q} \in B_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\ell(f_\beta(x), y)], \qquad (\star)$$

  where $B_\epsilon(\hat{\mathbb{P}}_N)$, the so-called ambiguity set, is a set of distributions around $\hat{\mathbb{P}}_N$. That is,

$$B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_N) \le \epsilon\},$$

  where $D(\cdot,\cdot)$ is a certain probability discrepancy.

- How to choose the probability metric $D(\cdot,\cdot)$? E.g., moment-based/ f-divergence/ Wasserstein distance.
  - asymptotic consistency;
  - support of the worst-case distribution $\mathbb{Q}$;

DRO Formulation

- Instead of ERM, consider minimizing the worst-case expected loss

$$\inf_{\beta} \sup_{\mathbb{Q}\in B_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{(x,y)\sim\mathbb{Q}}[\ell(f_\beta(x), y)], \qquad (\star)$$

  where $B_\epsilon(\hat{\mathbb{P}}_N)$, the so-called ambiguity set, is a set of distributions around $\hat{\mathbb{P}}_N$. That is,

$$B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_N) \le \epsilon\},$$

  where $D(\cdot,\cdot)$ is a certain probability discrepancy.

- How to choose the probability metric $D(\cdot,\cdot)$? E.g., moment-based/ f-divergence/ Wasserstein distance.
  - asymptotic consistency;
  - support of the worst-case distribution $\mathbb{Q}$;
  - tractability;

Wasserstein Distance

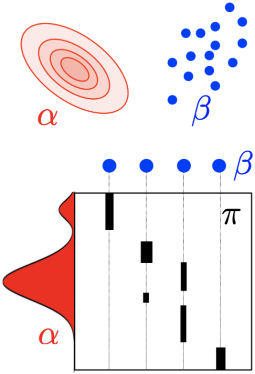
$$W(\alpha, \beta) := \inf_{\pi: z \sim \alpha, z' \sim \beta} \mathbb{E}_{(z,z') \sim \pi}[d(z, z')],$$

where

- $z = (x, y)$ is the input-output pair;
- $d(z, z')$ is the transport cost between z and z';
- π is a joint distribution (z, z').

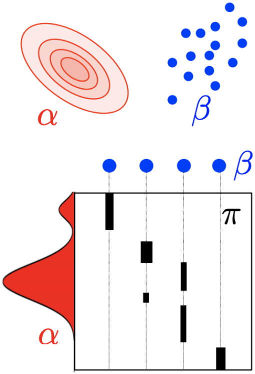
Specifically, we have

- $B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : W(\mathbb{Q}, \hat{\mathbb{P}}_N) \leq \epsilon\}$.



Wasserstein Distance

$$W(\alpha, \beta) := \inf_{\pi: z \sim \alpha, z' \sim \beta} \mathbb{E}_{(z,z') \sim \pi}[d(z, z')],$$



where

- $z = (x, y)$ is the input-output pair;
- $d(z, z')$ is the transport cost between z and z';
- π is a joint distribution (z, z').

Specifically, we have

- $B_\epsilon(\hat{\mathbb{P}}_N) = \{\mathbb{Q} : W(\mathbb{Q}, \hat{\mathbb{P}}_N) \leq \epsilon\}$.

Remarks

The worst-case distribution $\mathbb{Q}$ may have different support from $\hat{\mathbb{P}}_N$ and is capable of generating new examples within small perturbation.

Connect with Regularization and Adversarial Robustness

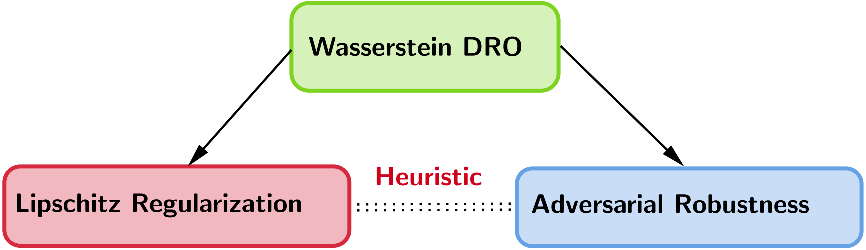


Figure 1: Connections among Wasserstein DRO, Generalized Lipschitz Regularization [Cranko et al., 2021], and Adversarial Robustness [Sinha et al., 2018]

Connect with Regularization and Adversarial Robustness

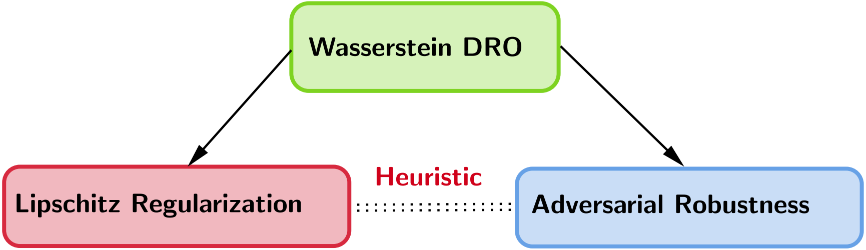


Figure 1: Connections among Wasserstein DRO, Generalized Lipschitz Regularization [Cranko et al., 2021], and Adversarial Robustness [Sinha et al., 2018]

Wasserstein DRO is a quite powerful modeling tool!

Main Question

Can we address Wasserstein DRO in a tractable way?

Main Question

Can we address Wasserstein DRO in a tractable way?

Outline

Introduction and Motivation

Tractable Conic Reformulation

ADMM-based First-Order Algorithmic Framework

Conclusion and Future Directions

Wasserstein DRO with Linear Hypothesis Space

- Binary classification problem $x \in \mathbb{R}^n$ and $y \in \{+1, -1\}$;
- Generalized linear model, $f_\beta(x) = \beta^T x$;
- Convex Lipschitz continuous loss, e.g., log-loss, hinge loss, smooth hinge loss;
- $d((x,y),(x',y')) = \|x - x'\|_p + \frac{\kappa}{2}|y - y'|$ with $\kappa > 0$ and $\|\cdot\|$ denotes the ℓ_p-norm on $\mathbb{R}^n$ where $p = \{1, 2, +\infty\}$;
- The parameter κ can be viewed as the reliability of the labels.

Wasserstein DRO with Linear Hypothesis Space

- Binary classification problem $x \in \mathbb{R}^n$ and $y \in \{+1, -1\}$;
- Generalized linear model, $f_\beta(x) = \beta^T x$;
- Convex Lipschitz continuous loss, e.g., log-loss, hinge loss, smooth hinge loss;
- $d((x,y),(x',y')) = \|x - x'\|_p + \frac{\kappa}{2}|y - y'|$ with $\kappa > 0$ and $\|\cdot\|$ denotes the ℓ_p-norm on $\mathbb{R}^n$ where $p = \{1, 2, +\infty\}$;
- The parameter κ can be viewed as the reliability of the labels.

Wasserstein DRO $(\star)$ admits a tractable conic reformulation!

Tractable Convex Reformulation

Theorem

(cf. Theorem 14 (ii) in [Shafieezadeh-Abadeh et al., 2019]) If $f_\beta(x) = \beta^T x$ and $\ell(\cdot,\cdot)$ is Lipschitz continuous, Problem $(\star)$ is equivalent to

$$\begin{aligned} \inf_{\lambda,\beta,s} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\ \text{s.t.} \quad & \ell(\beta^T \hat{x}_i, \hat{y}_i) \le s_i,\ i \in [N], \\ & \ell(\beta^T \hat{x}_i, -\hat{y}_i) - \lambda\kappa \le s_i,\ i \in [N], \\ & \mathit{Lip}(\ell)\|\beta\|_q \le \lambda. \end{aligned} \qquad (\Delta)$$

Here, $\frac{1}{p} + \frac{1}{q} = 1$ for $p \in \{1, 2, +\infty\}$.

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;
  - error free — $\kappa = +\infty$;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;
  - error free — $\kappa = +\infty$;
  - smoothness assumption for loss functions;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;
  - error free — $\kappa = +\infty$;
  - smoothness assumption for loss functions;
  - strong convexity for the transport cost;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;
  - error free — $\kappa = +\infty$;
  - smoothness assumption for loss functions;
  - strong convexity for the transport cost;

Algorithm Design for Wasserstein DRO

- Rely on general-purpose solvers (i.e., Gurobi, Mosek, YALMIP) — scalability issue ☹;
- Stochastic gradient descent (SGD) type algorithms for DRO problems [Sinha et al., 2018] — strong assumptions ☹;
  - error free — $\kappa = +\infty$;
  - smoothness assumption for loss functions;
  - strong convexity for the transport cost;

Our Target

How can we develop **provably** **efficient** algorithms tailored to a broad class of Wasserstein DRO problems?

The Principle of Algorithm Design

- Close the **theoretical** and **computational** gap
  - provable algorithms but slow practical implementations ☹;
  - practically fast algorithms without theoretical guarantees ☹;

The Principle of Algorithm Design

- Close the **theoretical** and **computational** gap
  - provable algorithms but slow practical implementations ☹;
  - practically fast algorithms without theoretical guarantees ☹;

Guideline for Algorithm Design

The practical efficiency indeed relies on how the algorithm exploits the problem-specific structure.

Outline

Introduction and Motivation

Tractable Conic Reformulation

ADMM-based First-Order Algorithmic Framework

Conclusion and Future Directions

Identify the Key Structures

Reformulate (Δ) as a compact form[1],

$$\begin{aligned} \min_{\beta, \lambda \geq 0} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} \max\left\{L(\beta^T \hat{z}_i), L(-\beta^T \hat{z}_i) - \lambda\kappa\right\} \\ \text{s.t.} \quad & \text{Lip}(L)\|\beta\|_q \leq \lambda. \end{aligned} \tag{1}$$

- Training data $\hat{z}_i = \hat{y}_i \cdot \hat{x}_i$;

[1]For simplicity, $L(\beta^T \hat{z}_i) = \ell(\hat{y}_i, \beta^T \hat{x}_i)$.

Identify the Key Structures

Reformulate (Δ) as a compact form[1],

$$\begin{aligned} \min_{\beta,\lambda\geq 0} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N}\max\left\{L(\beta^T\hat{z}_i), L(-\beta^T\hat{z}_i) - \lambda\kappa\right\} \\ \text{s.t.} \quad & \text{Lip}(L)\|\beta\|_q \leq \lambda. \end{aligned} \tag{1}$$

- Training data $\hat{z}_i = \hat{y}_i \cdot \hat{x}_i$;
- λ: one-dimensional search?

[1] For simplicity, $L(\beta^T\hat{z}_i) = \ell(\hat{y}_i, \beta^T\hat{x}_i)$.

Identify the Key Structures

Reformulate (Δ) as a compact form[1],

$$\begin{aligned} \min_{\beta,\lambda\geq 0} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} \max\left\{L(\beta^T\hat{z}_i), L(-\beta^T\hat{z}_i) - \lambda\kappa\right\} \\ \text{s.t.} \quad & \text{Lip}(L)\|\beta\|_q \leq \lambda. \end{aligned} \tag{1}$$

- Training data $\hat{z}_i = \hat{y}_i \cdot \hat{x}_i$;
- λ: one-dimensional search?
- β-subproblem: two non-separable non-smooth terms ☹;

[1] For simplicity, $L(\beta^T\hat{z}_i) = \ell(\hat{y}_i, \beta^T\hat{x}_i)$.

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;
- iLP-ADMM for the resulting β-subproblem

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;
- iLP-ADMM for the resulting β-subproblem
  - operator splitting — prototypical form

$$
\begin{aligned}
\min_{\beta,\mu} \quad & f(\mu) + g(\mu) + p(\beta) \\
\text{s.t.} \quad & Z\beta - \mu = 0
\end{aligned}
\tag{2}
$$

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;
- iLP-ADMM for the resulting β-subproblem
  - operator splitting — prototypical form

$$\begin{aligned} \min_{\beta,\mu} \quad & f(\mu) + g(\mu) + p(\beta) \\ \text{s.t.} \quad & Z\beta - \mu = 0 \end{aligned} \tag{2}$$

  - $f(\cdot)$: convex and gradient Lipschitz continuous $L_f > 0$;

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;
- iLP-ADMM for the resulting β-subproblem
  - operator splitting — prototypical form

$$\begin{aligned} \min_{\beta,\mu} \quad & f(\mu) + g(\mu) + p(\beta) \\ \text{s.t.} \quad & Z\beta - \mu = 0 \end{aligned} \tag{2}$$

  - $f(\cdot)$: convex and gradient Lipschitz continuous $L_f > 0$;
  - $g(\cdot)$: convex and non-differentiable with the separable structure and closed-form proximal mapping;

Key Ideas

- Golden section search for $\lambda^\star$ — establish a tight upper bound λ^U for the optimal $\lambda^\star$;
- iLP-ADMM for the resulting β-subproblem
  - operator splitting — prototypical form

$$\begin{aligned} \min_{\beta,\mu} \quad & f(\mu) + g(\mu) + p(\beta) \\ \text{s.t.} \quad & Z\beta - \mu = 0 \end{aligned} \tag{2}$$

  - $f(\cdot)$: convex and gradient Lipschitz continuous $L_f > 0$;
  - $g(\cdot)$: convex and non-differentiable with the separable structure and closed-form proximal mapping;
  - $p(\cdot) = \mathbb{I}_{\{\|\cdot\|_q \le \lambda\}}$;

Wasserstein DRO with Linear Hypothesis Space

Table 1: Three Representative Learning Models

Loss function $L(z)$	$f(\mu)$	$g_i(\mu_i)$
$\log(1+\exp(-z))$	$\frac{1}{N}\sum_{i=1}^{N}(L(\mu_i)+\frac{1}{2}(\mu_i-\lambda\kappa))$	$\frac{1}{2}\lvert z_i-\lambda\kappa\rvert$
$\max(1-z,0)$	0	$\max(1-\mu_i, 1+\mu_i-\lambda\kappa, 0)$
$\begin{cases} \frac{1}{2}-z & z\le 0 \\ \frac{1}{2}(1-z)^2 & 0<z<1 \\ 0 & z\ge 1 \end{cases}$	0	PLQ*

* piecewise linear-quadratic functions;

- Log-loss, hinge loss and smooth hinge loss;
- $g(\mu)=\frac{1}{N}\sum_{i=1}^{N} g_i(\mu_i)$;

Theoretical Upper Bound for $\lambda^\star$

Proposition

Suppose that $(\beta^\star, \lambda^\star, s^\star)$ is an optimal solution to Problem (Δ). Thus, we have

1. *If $L(z)$ is log-loss, we have $\lambda^\star \le \lambda^U = \frac{0.2785}{\epsilon}$.*
2. *If $L(z)$ is smooth hinge loss, we have $\lambda^\star \le \lambda^U = \frac{0.5}{\epsilon}$.*
3. *If $L(z)$ is hinge loss, we have $\lambda^\star \le \lambda^U = \frac{1}{\epsilon}$.*

- $q(\lambda) = \inf_{\beta} \Omega(\lambda, \beta)$ is a unimodal function on $\mathbb{R}$.
- $\Omega(\lambda, \beta) = \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} \max\left\{L(\beta^T \hat{z}_i), L(-\beta^T \hat{z}_i) - \lambda\kappa\right\} + \mathbb{I}_{\{\|\beta\|_q \le \lambda\}}$.

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

- **Ad-hoc linearized technique for μ-update**

$$\mu^{k+1} = \arg\min_{\mu}\left\{\nabla f(\mu^k)^T\mu + g(\mu) - \langle w^k, Z\beta^{k+1} - \mu\rangle + \frac{\rho}{2}\|\mu - Z\beta^{k+1}\|^2\right\};$$

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

- **Ad-hoc linearized technique for μ-update**

$$\mu^{k+1} = \arg\min_{\mu}\left\{\nabla f(\mu^k)^T\mu + g(\mu) - \langle w^k, Z\beta^{k+1} - \mu\rangle + \frac{\rho}{2}\|\mu - Z\beta^{k+1}\|^2\right\};$$

  - closed-form proximal mapping for $g(\mu)$;

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

- **Ad-hoc linearized technique for μ-update**

$$\mu^{k+1} = \arg\min_{\mu} \left\{ \nabla f(\mu^k)^T \mu + g(\mu) - \langle w^k, Z\beta^{k+1} - \mu \rangle + \frac{\rho}{2}\|\mu - Z\beta^{k+1}\|^2 \right\};$$

  - closed-form proximal mapping for $g(\mu)$;
  - exempt from the step size selection procedure;

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

- **Ad-hoc linearized technique for μ-update**

$$\mu^{k+1} = \arg\min_{\mu} \left\{ \nabla f(\mu^k)^T \mu + g(\mu) - \langle w^k, Z\beta^{k+1} - \mu \rangle + \frac{\rho}{2}\|\mu - Z\beta^{k+1}\|^2 \right\};$$

  - closed-form proximal mapping for $g(\mu)$;
  - exempt from the step size selection procedure;

- **Dynamically adjusting the penalty parameter**

Inexact Linearized Proximal ADMM (iLP-ADMM)

The augmented Lagrangian function is defined by

$$\mathcal{L}_\rho(\beta, \mu; w) = f(\mu) + g(\mu) + p(\beta) - w^T(Z\beta - \mu) + \frac{\rho}{2}\|Z\beta - \mu\|^2,$$

where $w \in \mathbb{R}^N$ is the multipliers and ρ is the penalty parameter.

- **Ad-hoc linearized technique for μ-update**

$$\mu^{k+1} = \arg\min_{\mu} \left\{ \nabla f(\mu^k)^T \mu + g(\mu) - \langle w^k, Z\beta^{k+1} - \mu \rangle + \frac{\rho}{2}\|\mu - Z\beta^{k+1}\|^2 \right\};$$

  - closed-form proximal mapping for $g(\mu)$;
  - exempt from the step size selection procedure;

- **Dynamically adjusting the penalty parameter**
  - $\rho_{k+1} \geq \rho_k$, e.g., geometrically increasing the penalty parameter;

iLP-ADMM (Cont'd)

- **Solving the β-subproblem in an inexact way**

$$\beta^{k+1} \approx \arg\min_{\beta \in \mathbb{R}^n} \left\{ \mathcal{L}_{\rho_{k+1}}(\beta, \mu^k; w^k) + \frac{1}{2}\|\beta - \beta^k\|_S^2 \right\};$$

iLP-ADMM (Cont'd)

- **Solving the β-subproblem in an inexact way**

$$\beta^{k+1} \approx \arg\min_{\beta \in \mathbb{R}^n} \left\{ \mathcal{L}_{\rho_{k+1}}(\beta, \mu^k; w^k) + \frac{1}{2}\|\beta - \beta^k\|_S^2 \right\};$$

  - select a positive semidefinite matrix S such that $[S; Z]$ has full column rank;

iLP-ADMM (Cont'd)

- **Solving the β-subproblem in an inexact way**

$$\beta^{k+1} \approx \arg\min_{\beta \in \mathbb{R}^n} \left\{ \mathcal{L}_{\rho_{k+1}}(\beta, \mu^k; w^k) + \frac{1}{2}\|\beta - \beta^k\|_S^2 \right\};$$

  - select a positive semidefinite matrix S such that $[S; Z]$ has full column rank;
  - convex quadratic problem with an ℓ_q-ball constraint — accelerated projected gradient descent;

iLP-ADMM (Cont'd)

- **Solving the β-subproblem in an inexact way**

$$\beta^{k+1} \approx \arg\min_{\beta \in \mathbb{R}^n} \left\{ \mathcal{L}_{\rho_{k+1}}(\beta, \mu^k; w^k) + \frac{1}{2}\|\beta - \beta^k\|_S^2 \right\};$$

  - select a positive semidefinite matrix S such that $[S; Z]$ has full column rank;
  - convex quadratic problem with an ℓ_q-ball constraint — accelerated projected gradient descent;
  - the error condition $\|d^{k+1}\| \leq \xi^{k+1}$,

$$d^{k+1} \in \partial_\beta \mathcal{L}_{\rho_{k+1}}(\beta^{k+1}, \mu^k; w^k) + S(\beta^{k+1} - \beta^k);$$

Convergence Analysis of iLP-ADMM

- The residual function we utilized to conduct the analysis,

$$r_{\mathsf{KKT}}(\beta,\mu,w) := d^2(0,\nabla f(\mu)+\partial g(\mu)+w)+d^2(0,\partial p(\beta)-Z^T w)+\|Z\beta-\mu\|^2.$$

Theorem (Informal Statement)

If $\sup_{k\geq 1}\rho_k \in (3L_f,+\infty)$ *and the error condition* $\sum_{k=1}^{\infty}\xi^k < \infty$ *holds, we have*

1. *The sequence* $\{(\beta^{k+1},\mu^{k+1},w^{k+1})\}_{k\geq 0}$ *converges to a KKT point of Problem* (2).
2. *The KKT squared residual* $r_{KKT}(\beta^K,\mu^K,w^K)$ *converges with rate* $o(\frac{1}{K})$, *i.e.,*

$$\min_{1\leq k\leq K}\left\{r_{KKT}(\beta^k,\mu^k,w^k)\right\} = o\left(\frac{1}{K}\right).$$

Numerical Results

Wall-clock Time Comparison with the YALMIP

Table 2: Wall-clock Time Comparison on UCI Adult Datasets: Log-loss, ℓ_∞-norm, $\kappa = 1, \epsilon = 0.1$

Dataset	Data Statistics		Wall-clock Time (s)		Ratio
	Sample	Feature	YALMIP	GS-ADMM [2]	
a1a	1605	123	47.98	3.12	15
a2a	2265	123	67.08	3.78	18
a3a	3185	123	112.64	4.82	23
a4a	4781	123	222.78	4.91	45
a5a	6414	123	449.76	4.63	91
a6a	11220	123	1282.32	7.27	176
a7a	16100	123	2509.61	8.11	309
a8a	22696	123	4887.58	8.52	574
a9a	32561	123	10835.75	9.31	1164

[2] GS-ADMM denotes the proposed first-order algorithmic framework.

Efficiency of iLP-ADMM for β-subproblem

- Consider a representative model[3]— log-loss with $q = \infty$,

$$\min_{\beta} \frac{1}{N}\sum_{i=1}^{N}\left(\log(1+\exp(-\beta^T\hat{z}_i) + \frac{1}{2}(\beta^T\hat{z}_i - \lambda\kappa)\right) + \frac{1}{2N}\|Z\beta - \lambda\kappa e_N\|_1$$

$$\text{s.t.} \quad \|\beta\|_\infty \leq \lambda.$$

[3] e_N denotes the all-ones vector in $\mathbb{R}^N$.

Efficiency of iLP-ADMM for β-subproblem

- Consider a representative model[3]— log-loss with $q = \infty$,

$$\min_{\beta} \frac{1}{N}\sum_{i=1}^{N}\left(\log(1+\exp(-\beta^T\hat{z}_i) + \frac{1}{2}(\beta^T\hat{z}_i - \lambda\kappa)\right) + \frac{1}{2N}\|Z\beta - \lambda\kappa e_N\|_1$$

$$\text{s.t.} \quad \|\beta\|_\infty \le \lambda.$$

- Baseline methods:
  - Two-block Standard ADMM (cf. **SADMM**): For both β- and μ-updates, we used the accelerated projected gradient descent and semi-smooth Newton method respectively.
  - Primal-Dual Hybrid Gradient (cf. **PDHG**);
  - Linearized-ADMM (cf. **LADMM**): compared with iLP-ADMM, we add the term $\frac{L_f}{4}\|\mu - \mu^k\|^2$ for the μ-update.
  - Projected Subgradient Method (cf. **Subgradient**);

[3] e_N denotes the all-ones vector in $\mathbb{R}^N$.

Efficiency of iLP-ADMM for β-subproblem

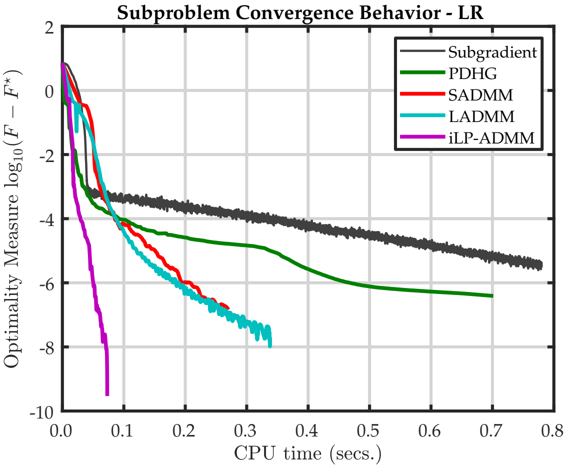


Figure 2: Synthetic Data — $(N, n) = (500, 100)$

$$F(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left(\log(1 + \exp(-\beta^T \hat{z}_i) + \frac{1}{2}(\beta^T \hat{z}_i - \lambda\kappa) \right) + \frac{1}{2N} \| Z\beta - \lambda\kappa e_N \|_1.$$

Efficiency of iLP-ADMM for β-subproblem

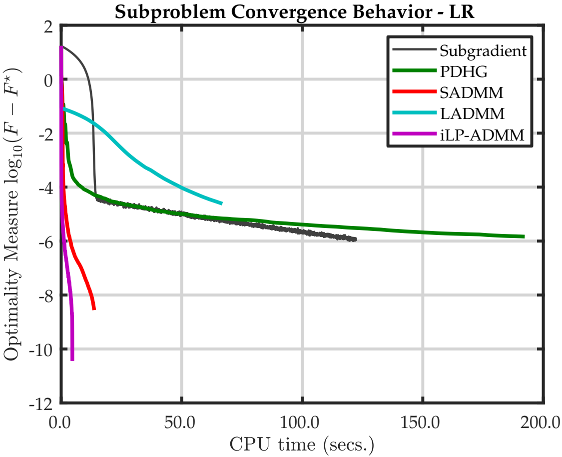


Figure 3: Synthetic Data — $(N, n) = (10000, 500)$

Efficiency of iLP-ADMM for β-subproblem

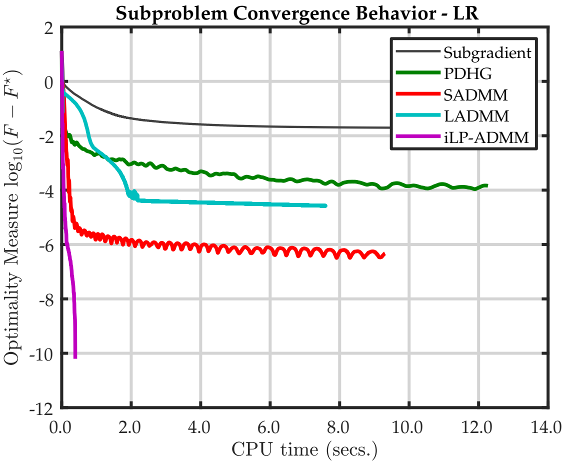


Figure 4: UCI Adult Dataset — a2a

Outline

Introduction and Motivation

Tractable Conic Reformulation

ADMM-based First-Order Algorithmic Framework

Conclusion and Future Directions

Conclusion and Future Directions

Summary

- Propose an **exceptionally efficient** first-order algorithmic framework for solving Wasserstein DRO problems with a linear hypothesis space;

Future Diection

Develop **provable** and **efficient** algorithms to tackle the distributionally robust formulation of deep neural network?

Conclusion and Future Directions

Summary

- Propose an **exceptionally efficient** first-order algorithmic framework for solving Wasserstein DRO problems with a linear hypothesis space;
- Produce **new computational tools** into the DRO community;

Future Diection

Develop **provable** and **efficient** algorithms to tackle the distributionally robust formulation of deep neural network?

Reference

Jiajin Li, Caihua Chen, Anthony Man-Cho So, and Sen Huang. "Towards a First-Order Algorithmic Framework for Wasserstein Distributionally Robust Risk Minimization." In Preparation.

The short version has been accepted in NeurIPS 2019.

Thank you! Questions?

References I

Zac Cranko, Zhan Shi, Xinhua Zhang, Richard Nock, and Simon Kornblith. Generalised lipschitz regularisation equals distributional robustness. In *International Conference on Machine Learning*, pages 2178–2188. PMLR, 2021.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein Distributional Robustness and Regularization in Statistical Learning. *arXiv preprint arXiv:1712.06050*, 2017.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.

Hongseok Namkoong and John C Duchi. Variance-based Regularization with Convex Objectives. In *Advances in Neural Information Processing Systems*, pages 2971–2980, 2017.

References II

Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103): 1–68, 2019.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.