Toward resilient, robust, responsible predictions/decisions

(a gentle introduction to optimal-transport-based distributionally robust optimization)

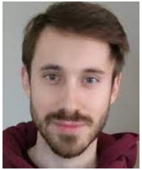
Jérôme MALICK

Learning and Optimization in Côte d'Azur (LOCA) – Sept. 2024

Based on joint work with

Waïss Azizian

Franck Iutzeler



Yassine Laguel

Tam Le

Florian Vincent

Deep learning can be impressive

Spectacular success of deep learning, in many fields/applications... E.g. in generation

Ex: illustrations generated from the title "towards resilient, robust, responsible decisions"

Deep learning can be impressive

Spectacular success of deep learning, in many fields/applications... E.g. in generation

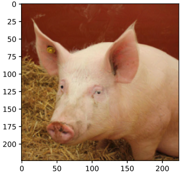
Ex: illustrations generated from the title "towards resilient, robust, responsible decisions"

with `stablediffusionweb.com`
(in sept 2023)

with `chatGPT`
(yesterday)

Don't forget how fragile deep learning can be !

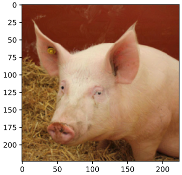
Example 1: Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)



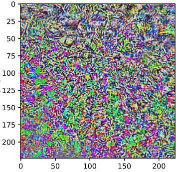
pig (99%)

Don't forget how fragile deep learning can be !

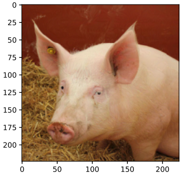
Example 1: Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)



$+\ \mathcal{E}\times$



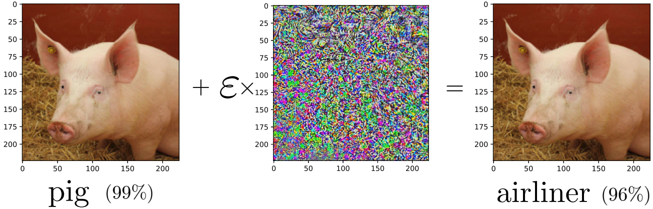
$=$



pig (99%)

Don't forget how fragile deep learning can be !

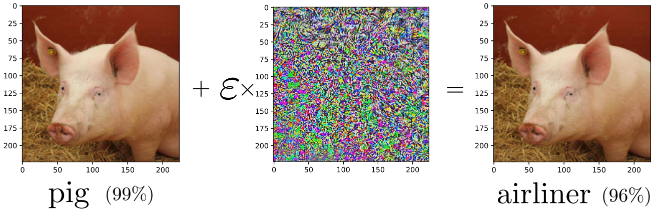
Example 1: Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)



"ML is a wonderful technology: it makes pigs fly" [Kolter, Madry '18]

Don’t forget how fragile deep learning can be !

Example 1: Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)

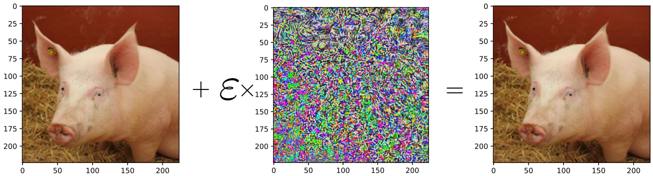


“ML is a wonderful technology: it makes pigs fly” [Kolter, Madry ’18]

Example 2: Attacks against self-driving cars [@ CVPR ’19]

Don't forget how fragile deep learning can be !

Example 1: Flying pigs (notebooks of NeurIPS 2018, tutorial on robustness)



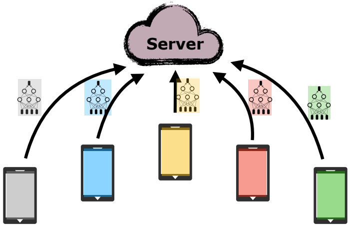
pig (99%) airliner (96%)

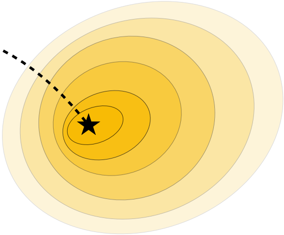
"ML is a wonderful technology: it makes pigs fly" [Kolter, Madry '18]

Example 2: Attacks against self-driving cars [@ ICLR '19]

ML may also perform poorly for some people

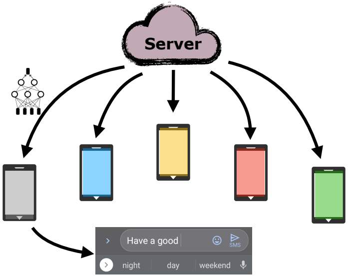
Example: Global model is trained on *average distribution* across clients (ERM)

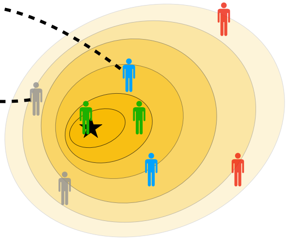




ML may also perform poorly for some people

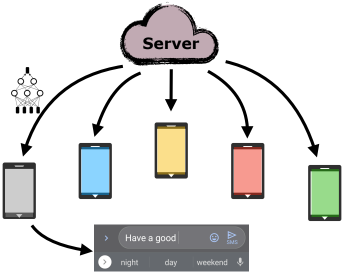
Example: Global model is deployed on *individual* clients

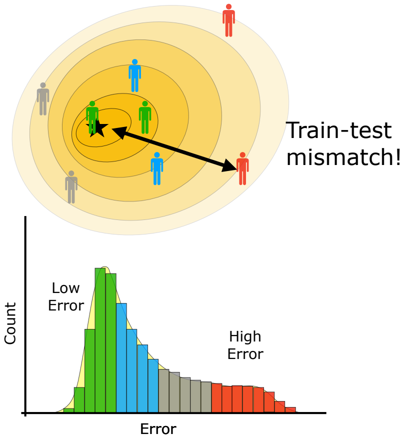




ML may also perform poorly for some people

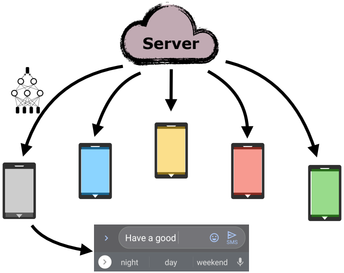
Example: Global model is deployed on *individual* clients

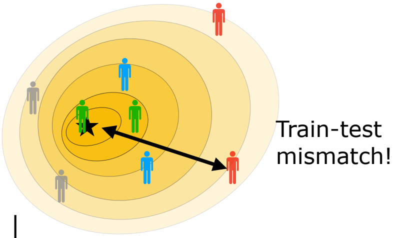




ML may also perform poorly for some people

Example: Global model is deployed on *individual* clients





Fairness issues, e.g.

Amazon : l'intelligence artificielle qui n'aimait pas les femmes

Fil info

09:55 Le cri de révolte de N

09:55 Municipales à Paris: l'éponge après des "a

09:40 En Inde, des applicat intimité précieuse au

Tout

Accélérer le recrutement en faisant analyser les CV par une IA : l'idée semblait prometteuse à Amazon. Mais elle s'est mise à sous-noter les femmes candidates à des postes tech.

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

The Washington Post

Democracy Dies in Darkness

THE ACCENT GAP

We tested Amazon's Alexa and Google's Home to see how people with accents are getting left behind in the smart-speaker revolution.

Upcoming legislation

European Union has recently considered the issue

- April ’19 : “Ethics Guidelines for Trustworthy AI”
- June ’24 : EU Artificial Intelligence Act passed
- July ’26 : High-risk AI will be required

  “Accuracy & Robustness consistently throughout their life cycle”

Upcoming legislation

European Union has recently considered the issue

- April '19 : "Ethics Guidelines for Trustworthy AI"
- June '24 : EU Artificial Intelligence Act passed
- July '26 : High-risk AI will be required

  "Accuracy & Robustness consistently throughout their life cycle"

In this context, our take :

distributionally robust optimization (topic of this talk ☺)

- is an answer to these issues and future requirements
- could be a pillar of trustworthy machine learning and decision-making

This talk: gentle introduction to WDRO

(Wasserstein) distributionally robust optimization (WDRO)

produces resilient, robust, responsible predictions/decisions

Very attractive:

- Natural in many applications (e.g. fairness [Pillutla, Laguel, M., Harchaoui '22]) back to [Scarf 1958] ! + (...) + recent trend in learning, e.g. [Kuhn *et al.* '20]
- Statistical/theoretical properties e.g. [Blanchet *et al.* '18] and [Blanchet and Shapiro '23]
- Computable in usual cases e.g. [Kuhn *et al.* '18], [Zhao Guan '18]...
- Interprets up to first-order as a penalization by $\|\nabla_\xi f(x, \xi)\|$ e.g. [Gao *et al.* '18]

This talk: gentle introduction to WDRO

(Wasserstein) distributionally robust optimization (WDRO)
produces resilient, robust, responsible predictions/decisions

Very attractive:

- Natural in many applications (e.g. fairness [Pillutla, Laguel, M., Harchaoui '22]) back to [Scarf 1958] ! + (...) + recent trend in learning, e.g. [Kuhn *et al.* '20]
- Statistical/theoretical properties – warning: dimensionality ! (spotlight #1) e.g. [Blanchet *et al.* '18] and [Blanchet and Shapiro '23]
- Computable in usual cases – in fact in many cases ! (spotlight #2) e.g. [Kuhn *et al.* '18], [Zhao Guan '18]...
- Interprets up to first-order as a penalization by $\|\nabla_\xi f(x,\xi)\|$ e.g. [Gao *et al.* '18]

Gentle introduction to WDRO: Outline

1. Basics of WDRO: setting, optimal transport, and duality
2. Spotlight $\#1$: Dimension-free statistical guarantees of WDRO
3. Spotlight $\#2$: Solving WDRO with `skWDRO`

Gentle introduction to WDRO: Outline

1. Basics of WDRO: setting, optimal transport, and duality
2. Spotlight #1: Dimension-free statistical guarantees of WDRO
3. Spotlight #2 : Solving WDRO with `skWDRO`

Math. setting

- Training data: $\xi_1, \ldots, \xi_N$ (in theory: sampled from $\mathbb{P}_{\text{train}}$ unknown)
  e.g. in supervised learning: labeled data $\xi_i = (a_i, y_i)$ feature, label
- Train model: $f(x, \cdot)$ the loss function with x the parameter/decision $(\omega, \beta, \theta, \ldots)$
  e.g. least-square regression: $f\left(x, (a, y)\right) = (x^\top a - y)^2$
- Compute x via empirical risk minimization (a.k.a SAA)
  (minimize the average loss on training data)

$$\min_{x} \frac{1}{N} \sum_{i=1}^{N} f(x, \xi_i)$$

Math. setting

- Training data: $\xi_1, \ldots, \xi_N$ (in theory: sampled from $\mathbb{P}_{\text{train}}$ unknown)

  e.g. in supervised learning: labeled data $\xi_i = (a_i, y_i)$ feature, label

- Train model: $f(x, \cdot)$ the loss function with x the parameter/decision $(\omega, \beta, \theta, \ldots)$

  e.g. least-square regression: $f\left(x, (a, y)\right) = (x^\top a - y)^2$

- Compute x via empirical risk minimization (a.k.a SAA)

  (minimize the average loss on training data)

$$\min_x \ \frac{1}{N} \sum_{i=1}^{N} f(x, \xi_i)$$

- Prediction with x for different data ξ
  - Adversarial attacks, **e.g.** flying pigs, driving cakes...
  - Presence of bias, **e.g.** heterogeneous data
  - Distributional shifts: $\mathbb{P}_{\text{train}} \neq \mathbb{P}_{\text{test}}$
  - Generalization: computations with $\widehat{\mathbb{P}}_N$ and guarantees on $\mathbb{P}_{\text{train}}$

- Solution: take possible variations into account during training

Math. setting

- Training data: $\xi_1, \dots, \xi_N$ (in theory: sampled from $\mathbb{P}_{\text{train}}$ unknown)
  e.g. in supervised learning: labeled data $\xi_i = (a_i, y_i)$ feature, label

- Train model: $f(x, \cdot)$ the loss function with x the parameter/decision $(\omega, \beta, \theta, \dots)$
  e.g. least-square regression: $f\left(x, (a, y)\right) = (x^\top a - y)^2$

- Compute x via empirical risk minimization (a.k.a SAA)
  (minimize the average loss on training data)

$$\min_x \frac{1}{N} \sum_{i=1}^{N} f(x, \xi_i) = \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x, \xi)] \qquad \text{with } \widehat{\mathbb{P}}_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\xi_i}$$

- Prediction with x for different data ξ
  - Adversarial attacks, **e.g.** flying pigs, driving cakes...
  - Presence of bias, **e.g.** heterogeneous data
  - Distributional shifts: $\mathbb{P}_{\text{train}} \neq \mathbb{P}_{\text{test}}$
  - Generalization: computations with $\widehat{\mathbb{P}}_N$ and guarantees on $\mathbb{P}_{\text{train}}$

- Solution: take possible variations into account during training

(Distributionally) robust optimization

Optimize expected loss for the worst probability in a set of perturbations

rather than $\min_{x} \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x, \xi)]$ solve instead $\min_{x} \max_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$

with $\mathcal{U}$ a neighborhood of $\widehat{\mathbb{P}}_N$ (called ambiguity set)

(Distributionally) robust optimization

Optimize expected loss for the worst probability in a set of perturbations

rather than $\min_{x} \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$ solve instead $\min_{x} \max_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$

with $\mathcal{U}$ a neighborhood of $\widehat{\mathbb{P}}_N$ (called ambiguity set)

Trade-off between modeling vs. computational tractability

- $\mathcal{U} = \left\{\widehat{\mathbb{P}}_N\right\}$: $\min_{x} \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$ standard ERM
- $\mathcal{U}$ defined by moments e.g. [Delage, Ye, '10] [Jegelka *et al.* '19]
- $\mathcal{U} = \left\{\mathbb{Q} : d(\widehat{\mathbb{P}}_N,\mathbb{Q}) \leqslant \rho\right\}$ for various distances or divergences
  E.g. KL-div., χ_2-div., max-mean-discrepancy... e.g. [Namkoong, Duchi '17]
- $\mathcal{U} = \left\{\mathbb{Q} : W(\widehat{\mathbb{P}}_N,\mathbb{Q}) \leqslant \rho\right\}$ Wasserstein distance [Kuhn *et al.* '18] – focus of this talk

Optimal transport comes into play

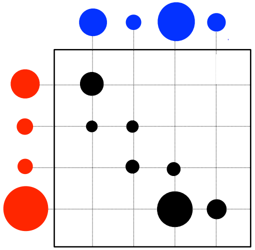
Wasserstein distance (given a cost function c)

$$W(\mathbb{P},\mathbb{Q}) \;=\; \min_{\boldsymbol{\pi}} \{\, \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P},\ [\boldsymbol{\pi}]_2 = \mathbb{Q} \,\}$$

Optimal transport comes into play

Wasserstein distance (given a cost function c)

$$W(\mathbb{P}, \mathbb{Q}) \;=\; \min_{\boldsymbol{\pi}} \{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi, \xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P},\ [\boldsymbol{\pi}]_2 = \mathbb{Q} \}$$

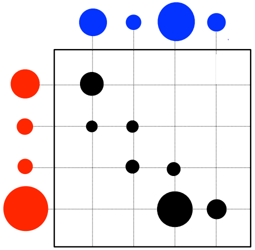


Discrete case

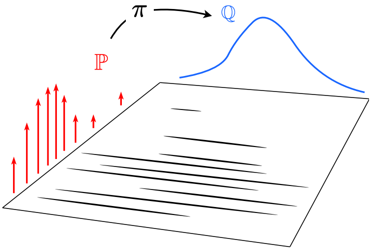
Optimal transport comes into play

Wasserstein distance (given a cost function c)

$$W(\mathbb{P},\mathbb{Q}) \;=\; \min_{\boldsymbol{\pi}} \left\{\, \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P},\ [\boldsymbol{\pi}]_2 = \mathbb{Q} \,\right\}$$



Discrete case



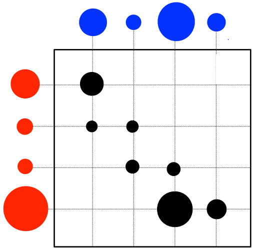
Semi-discrete case

$$\mathcal{U} = \{\, \mathbb{Q} : W(\widehat{\mathbb{P}}_N,\mathbb{Q}) \leqslant \rho \,\}$$

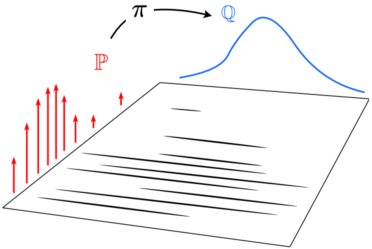
Optimal transport comes into play

Wasserstein distance (given a cost function c)

$$W(\mathbb{P}, \mathbb{Q}) \;=\; \min_{\boldsymbol{\pi}} \{\, \mathbb{E}_{\boldsymbol{\pi}}[c(\xi, \xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P},\ [\boldsymbol{\pi}]_2 = \mathbb{Q} \,\}$$



Discrete case



Semi-discrete case

$$\mathcal{U} = \{\, \mathbb{Q} : W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \,\}$$

Many ways to choose c (square distance, ℓ_p-distance...) – originality of our work: general c

E.g. classification tasks $c(\xi, \xi') = \|x - x'\|_2^2 + \kappa \mathbf{1}_{\{y \neq y'\}}$ with $\xi = (x, y)$

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\begin{cases} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{cases} \Leftrightarrow \begin{cases} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{cases}$$

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\left\{\begin{array}{c} \max_{\mathbb{Q}} \; \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\mathbb{Q},\boldsymbol{\pi}} \; \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\boldsymbol{\pi}} \; \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right.$$

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\left\{\begin{array}{c} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} \max_{\boldsymbol{\pi}} \ \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right.$$

$$\Leftrightarrow \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}\left[\max_{\xi'} \{f(x,\xi') - \lambda c(\xi,\xi')\}\right]$$

to be compared with $\mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\left\{\begin{array}{c} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\boldsymbol{\pi}} \ \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right.$$

$$\Leftrightarrow \ \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}\left[\max_{\xi'} \ \{f(x,\xi') - \lambda c(\xi,\xi')\} \right]$$

to be compared with $\quad \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$

...does not involve explicitly the transport plan

...computable in some (specific) cases [Kuhn *et al.* '18]

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\left\{\begin{array}{c} \max_{\mathbb{Q}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\mathbb{Q},\boldsymbol{\pi}} \ \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} \max_{\boldsymbol{\pi}} \ \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right.$$

$$\Leftrightarrow \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}[\max_{\xi'} \{f(x,\xi') - \lambda c(\xi,\xi')\}]$$

to be compared with $\mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$

...does not involve explicitly the transport plan

...computable in some (specific) cases [Kuhn *et al.* '18]

...actually many more; see spotlight #2

...does it worth it ? see spotlight #1

WDRO objective function

for given x, $\widehat{\mathbb{P}}_N$, ρ

$$\left\{\begin{array}{c} \max_{\mathbb{Q}} \; \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\mathbb{Q},\boldsymbol{\pi}} \; \mathbb{E}_{\mathbb{Q}}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N, [\boldsymbol{\pi}]_2 = \mathbb{Q} \\ \min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} \max_{\boldsymbol{\pi}} \; \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(x,\xi)] \\ [\boldsymbol{\pi}]_1 = \widehat{\mathbb{P}}_N \\ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \end{array}\right.$$

$$\Leftrightarrow \min_{\lambda \geqslant 0} \; \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}\left[\max_{\xi'} \; \{f(x,\xi') - \lambda c(\xi,\xi')\}\right]$$

to be compared with $\quad \mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)]$

...does not involve explicitly the transport plan

...computable in some (specific) cases [Kuhn *et al.* '18]

...actually many more; see spotlight #2

...does it worth it ? see spotlight #1

BTW: robustness brings nonsmoothness ♡

Illustration 1: the gain in robustness

Toy example: basic classification (linear, 2D, 2 classes...)

- Training data: $\xi_i = (a_i, y_i) \in \mathbb{R}^2 \times \{-1, +1\}$
  sampled from two Gaussian distributions with variances $\sigma = 1$ and $\sigma = 5$
- Testing data: reverse variance $\sigma = 5$ and $\sigma = 1$
- Compute standard separator by min logistic loss $f(x, \xi) = \log(1 + \exp(-y\, a^\top x))$

$$\min_{x} \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i\, a_i^\top x))$$

- Compute a robust separator by Wassertein DRO

Illustration 1: the gain in robustness

Toy example: basic classification (linear, 2D, 2 classes...)

- Training data: $\xi_i = (a_i, y_i) \in \mathbb{R}^2 \times \{-1, +1\}$
  sampled from two Gaussian distributions with variances $\sigma = 1$ and $\sigma = 5$
- Testing data: reverse variance $\sigma = 5$ and $\sigma = 1$
- Compute standard separator by min logistic loss $f(x, \xi) = \log(1 + \exp(-y\, a^\top x))$

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i\, a_i^\top x))$$

- Compute a robust separator by Wassertein DRO

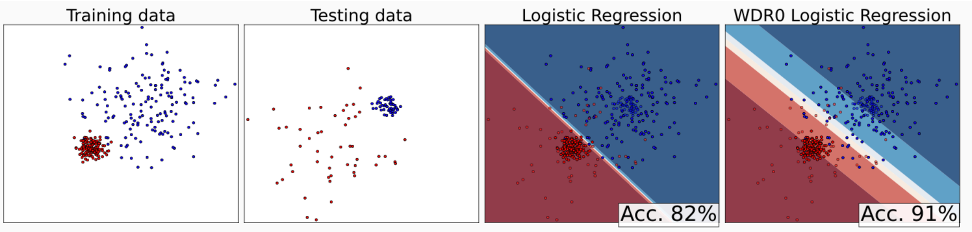


Illustration 2: gain in fairness

Federated learning framework with heterogeneous users (...) [Pillutla, Laguel, M., Harchaoui '22]

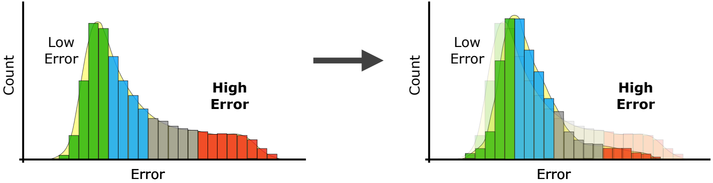
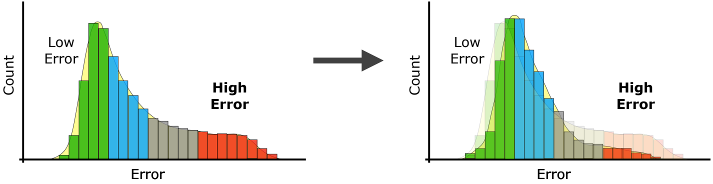


Illustration 2: gain in fairness

Federated learning framework with heterogeneous users (...) [Pillutla, Laguel, M., Harchaoui '22]



Experiments: (federated) classification task

ConvNet with EMNIST dataset

(1730 users, 179 images/users)

Histogram over users of test misclassif. error

Models: standard vs. robust

(dashed lines: 10%/90%-quantiles)

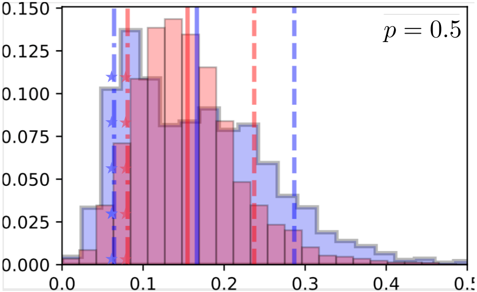
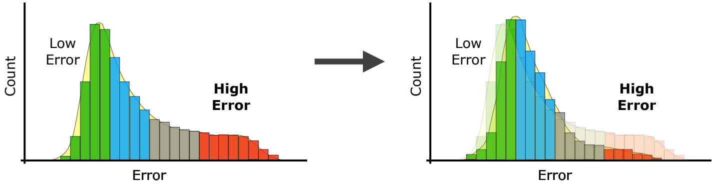


Illustration 2: gain in fairness

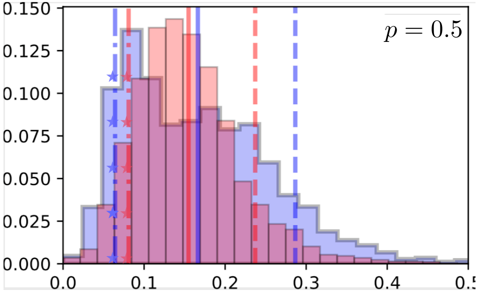
Federated learning framework with heterogeneous users (...) [Pillutla, Laguel, M., Harchaoui '22]



Experiments: (federated) classification task

ConvNet with EMNIST dataset
(1730 users, 179 images/users)

Histogram over users of test misclassif. error
Models: standard vs. robust
(dashed lines: 10%/90%-quantiles)



(W)DRO reshapes test histograms – towards more fairness

Gentle introduction to WDRO: Outline

1. Basics of WDRO: setting, optimal transport, and duality
2. Spotlight #1: Dimension-free statistical guarantees of WDRO
3. Spotlight #2 : Solving WDRO with `skWDRO`

Existing statistical guarantees of WDRO

- Suppose $\xi_1, \ldots, \xi_N \sim \mathbb{P}_{\text{train}}$ (where $\xi \in \mathbb{R}^d$)
- Computations with $\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i}$ and guarantees with $\mathbb{P}_{\text{train}}$?
- We manipulate the WDRO risk : $R_\rho(x) = \max_{W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$
- Obviously, if ρ, N large enough such that $W(\mathbb{P}_{\text{train}}, \widehat{\mathbb{P}}_N) \leqslant \rho$, then

$$\underbrace{R_\rho(x)}_{\text{can compute \& optimize}} \geqslant \underbrace{\mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x, \xi)]}_{\text{cannot access}}$$

- To be compared with $\mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x, \xi)] \geqslant \mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x, \xi)] + O\left(\frac{1}{\sqrt{N}}\right)$

Existing statistical guarantees of WDRO

- Suppose $\xi_1, \ldots, \xi_N \sim \mathbb{P}_{\text{train}}$ (where $\xi \in \mathbb{R}^d$)
- Computations with $\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i}$ and guarantees with $\mathbb{P}_{\text{train}}$?
- We manipulate the WDRO risk : $R_\rho(x) = \max_{W(\widehat{\mathbb{P}}_N, \mathbb{Q}) \leqslant \rho} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$
- Obviously, if ρ, N large enough such that $W(\mathbb{P}_{\text{train}}, \widehat{\mathbb{P}}_N) \leqslant \rho$, then

$$\underbrace{R_\rho(x)}_{\text{can compute \& optimize}} \geqslant \underbrace{\mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x,\xi)]}_{\text{cannot access}}$$

- To be compared with $\mathbb{E}_{\widehat{\mathbb{P}}_N}[f(x,\xi)] \geqslant \mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x,\xi)] + O\left(\frac{1}{\sqrt{N}}\right)$
- It requires $\rho \propto 1/\sqrt[d]{N}$ [Fournier and Guillin '15] (issue)
- Not optimal: $\rho \propto 1/\sqrt{N}$ suffices
  - asymptotically [Blanchet *et al* '22]
  - in particular cases [Shafieez-Adehabadeh *et al* '19]
  - or with error terms [Gao '22]

Extended exact generalization guarantees of WDRO

Our approach: a direct "optim." approach (work to get a concentration on the dual function)

Theorem ([Azizian, Iutzeler, M. '23], [Le, M. '24])

Assumptions: parametric family $f(x,\cdot)$ + compactness on x + compactness on ξ + non-degeneracy

For $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right)$ *then w.p.* $1-\delta$,

Generalization guarantee: $R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

Extended exact generalization guarantees of WDRO

Our approach: a direct "optim." approach (work to get a concentration on the dual function)

Theorem ([Azizian, Iutzeler, M. '23], [Le, M. '24])

Assumptions: parametric family $f(x,\cdot)$ + compactness on x + compactness on ξ + non-degeneracy

For $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right) = \rho_n$ *then w.p.* $1-\delta$,

Generalization guarantee: $R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

Distribution shifts:

$$W(\mathbb{P}_{train}, \mathbb{Q})^2 \leqslant \rho(\rho - \rho_n) \quad \textit{it holds} \quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

Extended exact generalization guarantees of WDRO

Our approach: a direct "optim." approach (work to get a concentration on the dual function)

Theorem ([Azizian, Iutzeler, M. '23], [Le, M. '24])

Assumptions: parametric family $f(x,\cdot)$ + compactness on x + compactness on ξ + non-degeneracy

For $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right) = \rho_n$ *then* *w.p.* $1-\delta$,

Generalization guarantee: $R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

Distribution shifts:

$$W(\mathbb{P}_{train}, \mathbb{Q})^2 \leqslant \rho(\rho - \rho_n) \quad \textit{it holds} \quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

Asymptotic tightness:

$$W(\mathbb{P}_{train}, \mathbb{Q})^2 \leqslant \rho(\rho + \rho_n) \quad \textit{it holds} \quad R_\rho(x) \leqslant \max_{\mathbb{Q}} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

Extended exact generalization guarantees of WDRO

Our approach: a direct "optim." approach (work to get a concentration on the dual function)

Theorem ([Azizian, Iutzeler, M. '23], [Le, M. '24])

Assumptions: parametric family $f(x,\cdot)$ + compactness on x + compactness on ξ + non-degeneracy

For $\delta \in (0,1)$, *if* $\rho \geqslant O\left(\sqrt{\frac{\log 1/\delta}{N}}\right) = \rho_n$ *then* *w.p.* $1-\delta$,

Generalization guarantee: $R_\rho(x) \geqslant \mathbb{E}_{\mathbb{P}_{train}}[f(x,\xi)]$

Distribution shifts:

$$W(\mathbb{P}_{train},\mathbb{Q})^2 \leqslant \rho(\rho-\rho_n) \quad \text{it holds} \quad R_\rho(x) \geqslant \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

Asymptotic tightness:

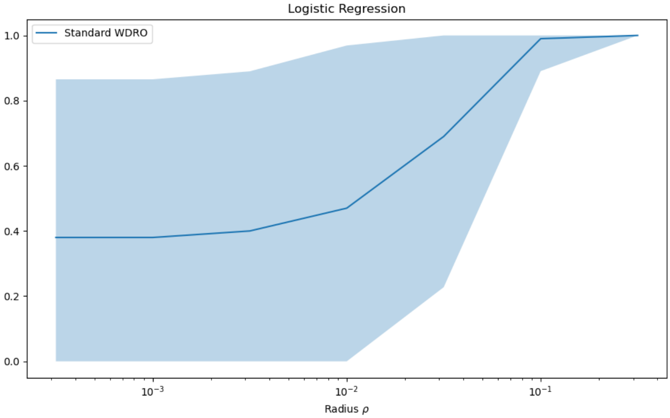
$$W(\mathbb{P}_{train},\mathbb{Q})^2 \leqslant \rho(\rho+\rho_n) \quad \text{it holds} \quad R_\rho(x) \leqslant \max_{\mathbb{Q}} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$$

- Universal result: deep learning, kernels, family of invertible mappings (e.g. normalizing flows)
- Retrieve existing results in linear/logistic regressions [Shafieez-Adehabadeh *et al* '19]

Theorem illustrated

On logistic regression:

- for each ρ, sample 200 training datasets
- solve the WDRO problem on each of them [Blanchet *et al* '22]
- plot the proba of $R_\rho(x) - \mathbb{E}_{\mathbb{P}_{\text{train}}}[f(x)] \geqslant 0$ (average, standard deviation)
- the training robust loss is indeed an upper-bound on the true loss



Robustness illustrated

Logistic regression again:

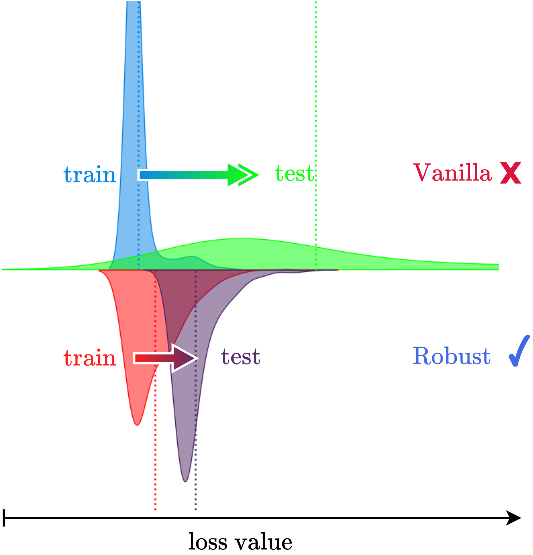
(train/test histograms)

Vanilla (ERM) model

- over-promises
- under-performs

Robust (WDRO) model

- (too?) conservative
- (way!) better testing loss



Robustness illustrated

Logistic regression again:

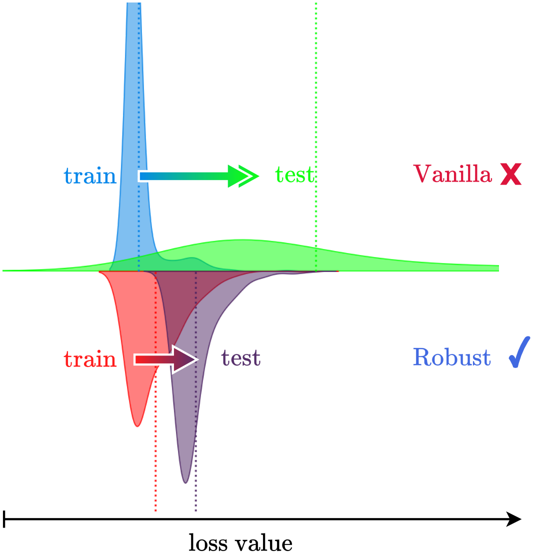
(train/test histograms)

Vanilla (ERM) model

- over-promises
- under-performs

Robust (WDRO) model

- (too?) conservative
- (way!) better testing loss



Great ! But how to compute such models !?

Gentle introduction to WDRO: Outline

1. **Basics of WDRO: setting, optimal transport, and duality**

2. **Spotlight #1: Dimension-free statistical guarantees of WDRO**

3. **Spotlight #2 : Solving WDRO with `skWDRO`**

Solving WDRO ?

Recall: dual WDRO objective is nonsmooth (in ℓ_2 case)

$$R_\rho(f) = \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}\left[\max_{\xi'} \{f(\xi') - \lambda\|\xi - \xi'\|^2\}\right]$$

Specific cases can still be formulated as convex optimization [Kuhn *et al.* '18]

Solving WDRO ?

Recall: dual WDRO objective is nonsmooth (in ℓ_2 case)

$$R_\rho(f) = \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}[\max_{\xi'} \{f(\xi') - \lambda\|\xi - \xi'\|^2\}]$$

Specific cases can still be formulated as convex optimization [Kuhn *et al.* '18]

Our approach: approximation by smoothing ! Smoothed WDRO counterpart:

$$R_\rho^\varepsilon(f) = \min_{\lambda \geqslant 0} \ \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N} \ \varepsilon \log\left(\mathbb{E}_{\xi' \sim \mathcal{N}(\xi,\sigma^2)} \exp\left(\frac{f(\xi') - \lambda\|\xi - \xi'\|^2}{\varepsilon}\right)\right)$$

Nice interpretation as entropy-regularized WDRO (similar but still different from Sinkhorn...)

Solving WDRO ?

Recall: dual WDRO objective is nonsmooth (in ℓ_2 case)

$$R_\rho(f) = \min_{\lambda \geqslant 0} \; \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N}\big[\max_{\xi'} \{f(\xi') - \lambda\|\xi - \xi'\|^2\}\big]$$

Specific cases can still be formulated as convex optimization [Kuhn *et al.* '18]

Our approach: approximation by smoothing ! Smoothed WDRO counterpart:

$$R_\rho^\varepsilon(f) = \min_{\lambda \geqslant 0} \; \lambda\rho + \mathbb{E}_{\widehat{\mathbb{P}}_N} \, \varepsilon \log\left(\mathbb{E}_{\xi' \sim \mathcal{N}(\xi,\sigma^2)} \exp\left(\frac{f(\xi') - \lambda\|\xi - \xi'\|^2}{\varepsilon}\right)\right)$$

Nice interpretation as entropy-regularized WDRO (similar but still different from Sinkhorn...)

Nice approximation results, e.g. :

Theorem (approximation bounds for WDRO [Azizian, Iutzeler, M. '21])

Under mild assumptions (non-degeneracy, f lipschitz), then

$$0 \;\leqslant\; R_\rho(f) - R_\rho^\varepsilon(f) \;\leqslant\; \left(C\,\varepsilon \,\log\frac{1}{\varepsilon}\right) d$$

Numerical optimization

Smoothed dual WDRO problem: minimizing a differentiable objective function

$$\min_{x} \min_{\lambda \geqslant 0} \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \varepsilon \log\left(\mathbb{E}_{\xi' \sim \mathcal{N}(\xi_i, \sigma^2)} \exp\left(\frac{f(x,\xi') - \lambda\|\xi - \xi'\|^2}{\varepsilon}\right)\right)$$

Our approach: use Pytorch tools (automatic backward diff. & adaptive SDG-like methods)

Numerical optimization

Smoothed dual WDRO problem: minimizing a differentiable objective function

$$\min_{x} \min_{\lambda \geqslant 0} \quad \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \varepsilon \log \left(\mathbb{E}_{\xi' \sim \mathcal{N}(\xi_i, \sigma^2)} \exp\left(\frac{f(x, \xi') - \lambda \|\xi - \xi'\|^2}{\varepsilon} \right) \right)$$

Our approach: use Pytorch tools (automatic backward diff. & adaptive SDG-like methods)

Not so easy, because of the inner expectation...

Numerical optimization

Smoothed dual WDRO problem: minimizing a differentiable objective function

$$\min_{x} \min_{\lambda \geqslant 0} \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \varepsilon \log\left(\mathbb{E}_{\xi' \sim \mathcal{N}(\xi_i, \sigma^2)} \exp\left(\frac{f(x,\xi') - \lambda\|\xi - \xi'\|^2}{\varepsilon}\right)\right)$$

Our approach: use Pytorch tools (automatic backward diff. & adaptive SDG-like methods)

Not so easy, because of the inner expectation...

Requires some (hard) work on computational aspects, e.g.

- Control the biais of the lower bound, after sampling $\xi'_j \sim \mathcal{N}(\xi_i, \sigma^2)$

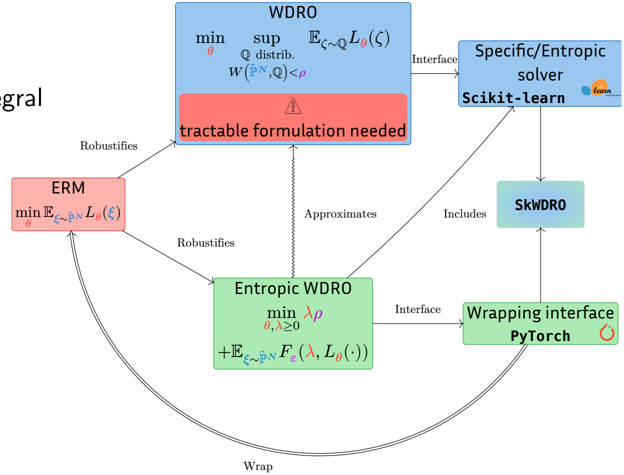
$$\min_{x} \min_{\lambda \geqslant 0} \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \varepsilon \log\left(\frac{1}{M}\sum_{j=1}^{M} \exp\left(\frac{f(x,\xi'_j) - \lambda\|\xi - \xi'_j\|^2}{\varepsilon}\right)\right)$$

  Objective still sharply peaked (so high variance in the gradient estimate...)

- Use importance sampling: sample the ξ'_j shifted towards the gradient

Python Toolbox `skWDRO`

- Control on the approximations
- Importance sampling for the inner integral
- Careful logsumexp
- Numerically stable backward pass
- Heuristics to set ε and σ
- Efficient heuristic to set starting λ
- All-in-one API
- User-friendly interfaces (Pytorch and Scikitlearn)



Try it out !

More (to come) in [Vincent, Azizian, Iutzeler, M. '24]

Easy to use, with few lines of code

Scikitlearn

```
from sklearn.linear_model import LogisticRegression # scikit-learn's standard version
from skwdro.linear_models import LogisticRegression as WDROLogisticRegression # WDRO version
```

Pytorch

```
def main():
    device = "cuda" if pt.cuda.is_available() else "cpu"
    model = MyShallowNet([1, 50, 30, 10, 1]).to(device)

    rho = pt.tensor(1e-1).to(device)

    x = pt.sort(pt.flatten(
        pt.linspace(0., 1., 10, device=device).unsqueeze(0)\
        + pt.randn(10000, 10, device=device) * 1e-1
    ))[0]
    y = f(x) + pt.randn(100000, device=device) * 2e-2
    dataset = DataLoader(TensorDataset(x.unsqueeze(-1), y.unsqueeze(-1)), batch_size=5000, shuffle=True)

    # New line: "dualize" the loss
    dual_loss = dualize_primal_loss(
            nn.MSELoss(reduction='none'),
            model,
            rho,
            x.unsqueeze(-1),
            y.unsqueeze(-1)
        )

    model = train(dual_loss, dataset, 1000) # type: ignore
    model.eval()
```

You can easily robustify your own models with skWDRO !

To sum up, in one slide...

Main take-aways

- ML works well, unless it does not. Work needed. Optimization is in the game
- Distributionally robust optimization is rich, active topic
- Spotlight #1: WDRO has nice generalization properties
- Spotlight #2: WDRO in practice with `skWDRO` (via scitkitlearn + Pytorch wrappers)

To sum up, in one slide...

Main take-aways

- ML works well, unless it does not. Work needed. Optimization is in the game
- Distributionally robust optimization is rich, active topic
- Spotlight #1: WDRO has nice generalization properties
- Spotlight #2: WDRO in practice with `skWDRO` (via scitkitlearn + Pytorch wrappers)

What's next ?

- Beyond Wasserstein neighborhoods... new models, new applications !
- How to deal with difficult constraints ? (0-1 variables, mixed-integer sets...)
- (after heterogeneous federated learning) real-life applications with impact ? More fairness ?

To sum up, in one slide...

Main take-aways

- ML works well, unless it does not. Work needed. Optimization is in the game
- Distributionally robust optimization is rich, active topic
- Spotlight #1: WDRO has nice generalization properties
- Spotlight #2: WDRO in practice with `skWDRO` (via scitkitlearn + Pytorch wrappers)

What's next ?

- Beyond Wasserstein neighborhoods... new models, new applications !
- How to deal with difficult constraints ? (0-1 variables, mixed-integer sets...)
- (after heterogeneous federated learning) real-life applications with impact ? More fairness ?

thank you all ☺

Work presented here

Y. Laguel, K. Pillutla, J. Malick, Z. Harchaoui
Federated Learning with Heterogeneous Data: A Superquantile Optimization Approach
Machine Learning Research, 2022

A. Waiss, F. Iutzeler, J. Malick
Regularization for Wasserstein distributionnally robust optimization
ESAIM: Control, Optimization, and Calculus of Variations, 2023

W. Azizian, F. Iutzeler, J. Malick
Exact Generalization Guarantees for (Regularized) Wasserstein Distributionally Robust Models
Advances in Neural Information Processing Systems (NeurIPS), 2023

T. Le, J. Malick
Universal generalization guarantees for Wasserstein distributionally robust models
Still an hope for NeurIPS, 2024

F. Vincent, W. Azizian, F. Iutzeler, J. Malick
`skwdro`: a library for Wasserstein distributionally robust machine learning
To be submitted, 2024