Optimization beyond minimization:

spurious GANs, Wasserstein robustness, and other applications in machine learning*

Jérôme MALICK

CNRS, Lab. Jean Kuntzmann & MIAI

Thoth Seminar – Inria Grenoble – May 2022

*based on joint work with good people from DAO...

Optimization for machine learning

Optim. is at the core of ML, playing a fundamental role behind the scenes

(model training, hyperparameter tuning, feature selection,...)

$$\min_x \; F(x) \;=\; \mathbb{E}_{\xi\sim\mathbb{P}}[f(x,\xi)] \;\text{ or }\; \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$$

e.g. least-squares regression: $\xi_i = (a_i, y_i)$ feature, label

$$f\left(x,(a,y)\right) = (x^\top a - y)^2$$

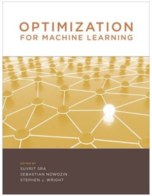
Optimization for machine learning

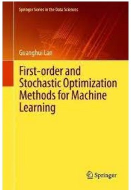
Optim. is at the core of ML, playing a fundamental role behind the scenes
(model training, hyperparameter tuning, feature selection,...)

$$\min_x \ F(x) \ = \ \mathbb{E}_{\xi\sim\mathbb{P}}[f(x,\xi)] \ \text{ or } \ \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$$

e.g. least-squares regression: $\xi_i = (a_i, y_i)$ feature, label
$f(x,(a,y)) = (x^\top a - y)^2$

E.g. optim. workshops at NeurIPS/ICML... multiple books...





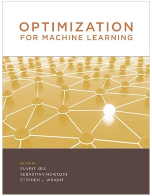
Optimization for machine learning

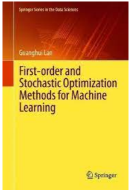
Optim. is at the core of ML, playing a fundamental role behind the scenes
(model training, hyperparameter tuning, feature selection,...)

$$\min_x \; F(x) \;=\; \mathbb{E}_{\xi\sim\mathbb{P}}[f(x,\xi)] \;\text{ or }\; \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$$

e.g. least-squares regression: $\xi_i = (a_i, y_i)$ feature, label
$f(x,(a,y)) = (x^\top a - y)^2$

E.g. optim. workshops at NeurIPS/ICML... multiple books...





E.g. Test of Time Awards

NeurIPS 2019 [Xiao '09]

Dual Averaging Method for Regularized Stochastic Learning and Online Optimization

Lin Xiao
Microsoft Research, Redmond, WA 98052
lin.xiao@microsoft.com

Abstract

We consider regularized stochastic learning and online optimization problems, where the objective function is the sum of two convex terms: one is the loss function of the learning task, and the other is a simple regularization term such as

ICML 2019 [Mairal *et al* '09]

Online Dictionary Learning for Sparse Coding

Julien Mairal JULIEN.MAIRAL@INRIA.FR
Francis Bach FRANCIS.BACH@INRIA.FR
INRIA,[1] 45 rue d'Ulm 75005 Paris, France
Jean Ponce JEAN.PONCE@ENS.FR
Ecole Normale Supérieure,[1] 45 rue d'Ulm 75005 Paris, France
Guillermo Sapiro GUILLE@UMN.EDU
University of Minnesota - Department of Electrical and Computer Engineering, 200 Union Street SE, Minneapolis, USA

Abstract

Sparse coding—that is, modelling data vectors as sparse linear combinations of basis elements—is widely used in machine learning, neuroscience, signal processing, and statistics. This paper focuses on *learning* the basis set, also called dic-

like decompositions based on principal component analysis and its variants, these models do not impose that the basis vectors be orthogonal, allowing more flexibility to adapt the representation to the data. While learning the dictionary has proven to be critical to achieve (or improve upon) state-of-the-art results, effectively solving the corresponding optimization problem is a significant compu-

NeurIPS 2020 [Recht *et al* '10]

HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent

Feng Niu leonn@cs.wisc.edu
Benjamin Recht brecht@cs.wisc.edu
Christopher Ré chrisre@cs.wisc.edu
Stephen J. Wright swright@cs.wisc.edu
Computer Sciences Department
University of Wisconsin-Madison
Madison, WI 53706

Abstract

Stochastic Gradient Descent (SGD) is a popular algorithm that can achieve state-of-the-art performance on a variety of machine learning tasks. Several researchers have recently proposed schemes to parallelize SGD, but all require performance-

ICML 2021 [Seeger *et al* '09]

Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design

Niranjan Srinivas NIRANJAN@CALTECH.EDU
Andreas Krause KRAUSEA@CALTECH.EDU
California Institute of Technology, Pasadena, CA, USA
Sham Kakade SKAKADE@WHARTON.UPENN.EDU
University of Pennsylvania, Philadelphia, PA, USA
Matthias Seeger MSEEGER@MMCI.UNI-SAARLAND.DE
Saarland University, Saarbrücken, Germany

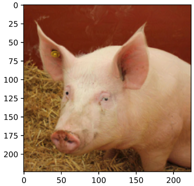
Abstract

Many applications require optimizing an unknown, noisy function that is expensive to evaluate. We formalize this task as a multi-armed bandit problem, where the payoff function is either sampled from a Gaussian process (GP) or has low RKHS norm. We resolve the important open problem of deriving regret bounds for this setting, which imply novel convergence rates

as possible, for example by maximizing information gain. The challenge in both approaches is twofold: we have to estimate an unknown function f from noisy samples, and we must optimize our estimate over some high-dimensional input space. For the former, much progress has been made in machine learning through kernel methods and Gaussian process (GP) models (Rasmussen & Williams, 2006), where smoothness

Beyond minimization? Two classical illustrative examples

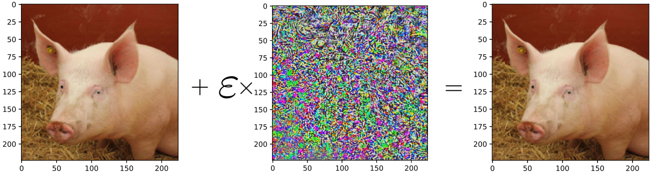
Flying pigs



pig (99%)

Beyond minimization? Two classical illustrative examples

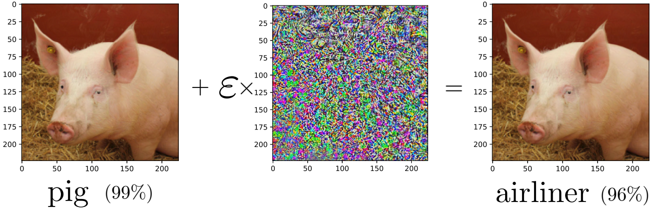
Flying pigs



pig (99%)

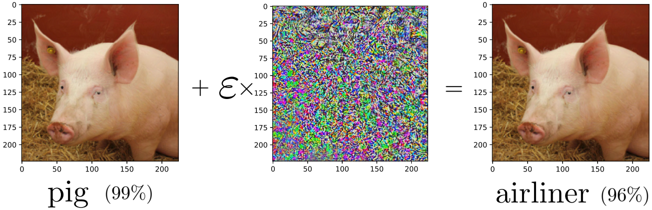
Beyond minimization? Two classical illustrative examples

Flying pigs



Beyond minimization? Two classical illustrative examples

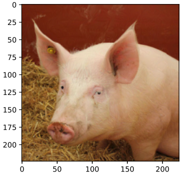
Flying pigs – robust/adversarial training (from notebooks of NeurIPS 2018, tutorial on robustness)



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

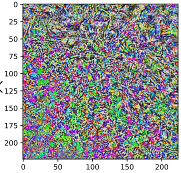
Beyond minimization? Two classical illustrative examples

Flying pigs – robust/adversarial training (from notebooks of NeurIPS 2018, tutorial on robustness)

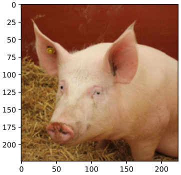


pig (99%)

$+ \ \mathcal{E} \times$



$=$



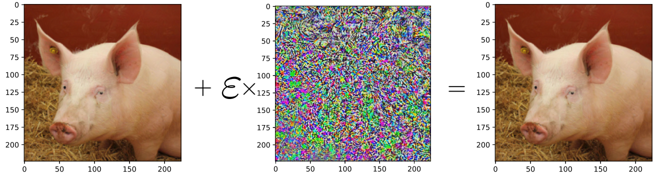
airliner (96%)

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

$$\min_x \ \mathbb{E}_{(a,y)\sim \text{data}} \left[\max_{\|a'-a\|_\infty \leqslant \rho} f(x,(a',y)) \right]$$

Beyond minimization? Two classical illustrative examples

Flying pigs – robust/adversarial training (from notebooks of NeurIPS 2018, tutorial on robustness)



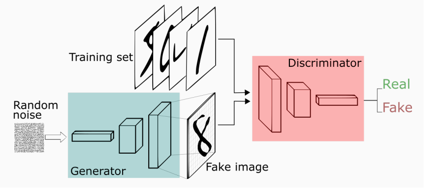
pig (99%) airliner (96%)

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

$$\min_{x} \mathbb{E}_{(a,y)\sim\text{data}}\left[\max_{\|a'-a\|_\infty \leqslant \rho} f(x,(a',y))\right]$$

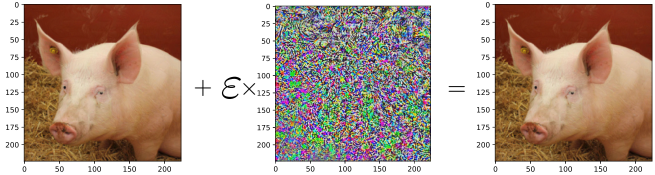
GANs training [Goodfellow *et al* '14]

$$\min_{\theta}\max_{\omega} \mathbb{E}_{\xi\sim\mathbb{P}_{\text{data}}}[\log D_\omega(\xi)] + \mathbb{E}_{\xi'}[\log(1-D_\omega(G_\theta(\xi')))]$$



Beyond minimization? Two classical illustrative examples

Flying pigs – robust/adversarial training (from notebooks of NeurIPS 2018, tutorial on robustness)



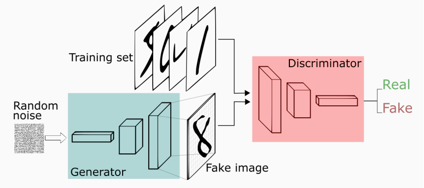
pig (99%)   airliner (96%)

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

$$\min_x \; \mathbb{E}_{(a,y)\sim \text{data}} \left[\max_{\|a'-a\|_\infty \leqslant \rho} f(x,(a',y)) \right] = \min_x \max_{\mathbb{P}_{\text{adv}} \in \mathcal{U}} \mathbb{E}_{(a,y)\sim \mathbb{P}_{\text{adv}}} \left[f(x,(a,y)) \right]$$

GANs training [Goodfellow *et al* '14]

$$\min_\theta \max_\omega \mathbb{E}_{\xi\sim\mathbb{P}_{\text{data}}} [\log D_\omega(\xi)] + \mathbb{E}_{\xi'} [\log(1 - D_\omega(G_\theta(\xi')))]$$



In this talk

Part I – about stochastic algorithms for min-max problems

Part II – about robust models in learning

,

In this talk

Part I – about stochastic algorithms for min-max problems

- illustrate spurious convergence – even for toy example
- present a simple fix and its theoretical guarantees
  [Hsieh, Iutzeler, M., Mertikopoulos, '20] – spotlight NeurIPS ☺

Part II – about robust models in learning

Yu-Guan Hsieh,

In this talk

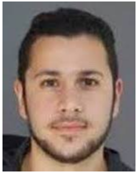
Part I – about stochastic algorithms for min-max problems

- illustrate spurious convergence – even for toy example
- present a simple fix and its theoretical guarantees
  [Hsieh, Iutzeler, M., Mertikopoulos, '20] – spotlight NeurIPS ☺

Part II – about robust models in learning

- introduce (distributionally) robust optimization, applied to learning problems
  [Laguel, Pillutla, M., Harchaoui '21]

Yu-Guan Hsieh, Yassine Laguel,



In this talk

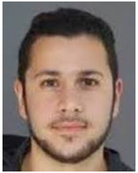
Part I – about stochastic <u>algorithms</u> for min-max problems

- illustrate spurious convergence – even for toy example
- present a simple fix and its theoretical guarantees
  [Hsieh, Iutzeler, M., Mertikopoulos, '20] – spotlight NeurIPS ☺

Part II – about robust <u>models</u> in learning

- introduce (distributionally) robust optimization, applied to learning problems
  [Laguel, Pillutla, M., Harchaoui '21]
- derive some nice duality/approximation results
  [Azizian, Iutzeler, M. '22]

Yu-Guan Hsieh, Yassine Laguel, Waïss Azizian



Part I – About stochastic algorithms for min/max

Success of Generative Adversarial Networks...

Question: who is real, who isn't ?

Success of Generative Adversarial Networks...

Question: who is real, who isn't ?

Answer: both are fake !
[https://thispersondoesnotexist.com]

Much technology... and some maths ☺

Optimization plays a role during training to compute equilibrium Generator/Discriminator

Success of Generative Adversarial Networks...

Question: who is real, who isn't ?

Answer: both are fake !
[https://thispersondoesnotexist.com]

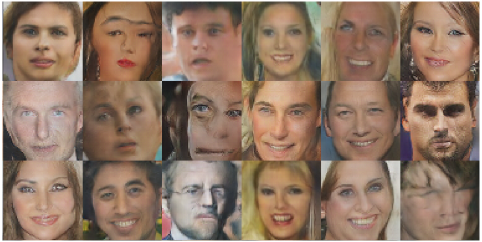
Much technology... and some maths ☺

Optimization plays a role during training to compute equilibrium Generator/Discriminator

Issue: Convergence of training algorithms ?

Coupling of two neural networks gives rise to strange behaviors and phenomena

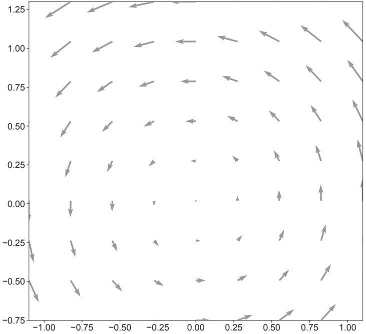
Even when solved with state-of-the-art stochastic gradient (extra-gradient variants)



Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very basic problems

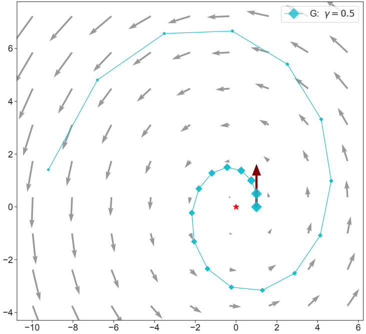
Example: $\min_x \max_y \; x\,y$ of solution/equilibrium $= (0,0)$ (arrows: gradient flows $V(x,y) = (-y,x)$)



Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very basic problems

Example: $\min_x \max_y \; x\,y$ of solution/equilibrium $= (0, 0)$ (arrows: gradient flows $V(x, y) = (-y, x)$)

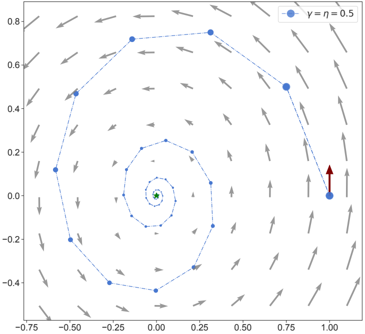


- Gradient algorithm diverges...

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very basic problems

Example: $\min_x \max_y \; x\,y$ of solution/equilibrium $= (0,0)$ (arrows: gradient flows $V(x,y) = (-y, x)$)

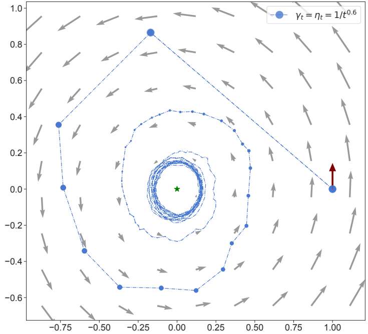


- Gradient algorithm diverges...
- Extra-gradient algorithm converges (thanks to its additional correction step)

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very basic problems

Example: $\min_x \max_y \; x\, y$ of solution/equilibrium $=(0,0)$ (arrows: gradient flows $V(x,y)=(-y,x)$)

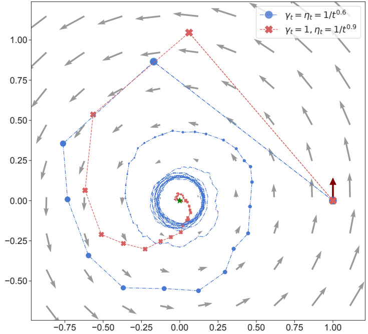


- Gradient algorithm diverges...
- Extra-gradient algorithm converges (thanks to its additional correction step)
- Stochastic extra-gradient never converges...

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very basic problems

Example: $\min_x \max_y \; x\,y$ of solution/equilibrium $= (0, 0)$ (arrows: gradient flows $V(x, y) = (-y, x)$)



- Gradient algorithm diverges...
- Extra-gradient algorithm converges (thanks to its additional correction step)
- Stochastic extra-gradient never converges...
- A remedy: use double stepsize strategy [Hsieh, Iutzeler, M., Mertikopoulos '20]

General set-up and simple new strategy

To compute a solution of $V(X) = 0$ from stochastic oracle ($\mathbb{E}[\hat{V}_s] = V(X_s)$ and bounded variance)

We propose to explore aggressively and update conservatively, in the stoc. extra-gradient

$$\begin{cases} X_{t+\frac{1}{2}} = X_t - \gamma_t \hat{V}_t \\ X_{t+1} = X_t - \gamma_t \hat{V}_{t+\frac{1}{2}} \end{cases} \rightsquigarrow \begin{cases} X_{t+\frac{1}{2}} = X_t - \gamma_t \hat{V}_t \\ X_{t+1} = X_t - \eta_t \hat{V}_{t+\frac{1}{2}} \end{cases} \quad \text{with } \eta_t/\gamma_t \to 0$$

Theorem [last-iterate convergence rate] [Hsieh, Iutzeler, M., Mertikopoulos '20]

1. Let V be monotone and affine. With stepsizes $\gamma_t \equiv \gamma$ and $\eta_t \simeq 1/t$,

$$\mathbb{E}[\|X_t - X^\star\|^2] \leqslant O\left(\frac{1}{t}\right)$$

2. Let V be variationally stable* and satisfy the error bound* condition. With stepsizes of the form $\gamma_t = \gamma/(t+b)^{1/3}$ and $\eta_t = \eta/(t+b)^{2/3}$,

$$\mathbb{E}[\|X_t - X^\star\|^2] \leqslant O\left(\frac{1}{\sqrt[3]{t}}\right)$$

$^*\langle V(X), X - X^\star\rangle \geqslant 0$ for all X

$^*\exists \tau > 0 : \|V(X)\| \geqslant \tau \|X - X^\star\|^2$ e.g. affine, strongly monotone...

Conclusions, perspectives on Part I

Many extensions, variations, improvements,...

- We also have local convergence results... beyond monotonicity ! (a bit technical)
- The constrained case is more complicated... still 13 days before deadline ;-)

Suggestion: invite Yu-Guan, who the ultimate expert on these topics...

Bottomline

- We propose a simple modification of the stochastic extragradient scheme to make its last iterate converge in a large spectrum of problems including all monotone games.
- Explicit convergence rates under additional assumptions (+ local convergence results)

Conclusions, perspectives on Part I

Many extensions, variations, improvements,...

- We also have local convergence results... beyond monotonicity ! (a bit technical)
- The constrained case is more complicated... still 13 days before deadline ;-)

Suggestion: invite Yu-Guan, who the ultimate expert on these topics...

Bottomline

- We propose a simple modification of the stochastic extragradient scheme to make its last iterate converge in a large spectrum of problems including all monotone games.
- Explicit convergence rates under additional assumptions (+ local convergence results)

small break for questions before Part II ?

Part II – About robust optimization and learning

Robustness...

we do not want machine-learned systems to fail when used in real-word

Robustness...

we do not want machine-learned systems to fail when used in real-word

Example 1:

keep in mind how fragile can be deep learning techniques

[@ NeurIPS '18]

Teapot(24.99%)
Joystick(37.39%)

Robustness...

we do not want machine-learned systems to fail when used in real-word

Example 1:

keep in mind how fragile can be deep learning techniques

[@ NeurIPS '18]

Teapot(24.99%)
Joystick(37.39%)

Example 2: Attacks against self-driving cars [@ CVPR '18]

Robustness...

we do not want machine-learned systems to fail when used in real-word

Example 1:

keep in mind how fragile can be deep learning techniques

[@ NeurIPS '18]

Teapot(24.99%)
Joystick(37.39%)

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

Robust ML

we do not want machine-learned systems to fail when used in real-word

Example 3: Data heterogeneity

Robust ML

we do not want machine-learned systems to fail when used in real-word

Example 3: Data heterogeneity

E.g. in federated learning

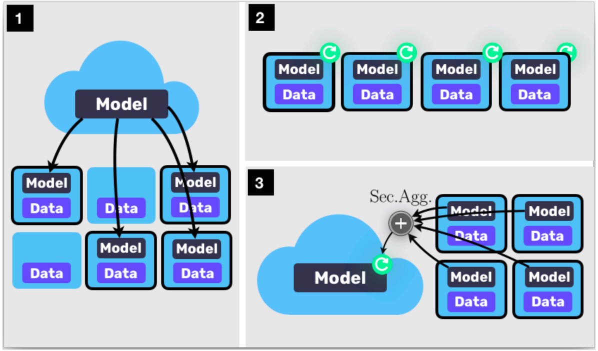
Google, hospital consortiums...

What about non-conforming users ?

Many issues !

(service quality? fairness?...)

More later...



remember the talk of Yassine Laguel in November...

Set-up: data-driven optimization under uncertainty

- Training data: $\xi_1, \ldots, \xi_N \sim \mathbb{P}$ (unknown)
  e.g. in supervised learning: $\xi_i = (a_i, y_i)$ feature, label
- Train model: x the parameter $f(x, \cdot)$ the objective function
  e.g. least-square regression: $f(x, (a, y)) = (x^\top a - y)^2$
- 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 ? (generalisation, data shifts, adversarial examples,...)
  Take possible variations into account during training (= when optimizing ☺)

Set-up: data-driven optimization under uncertainty

- Training data: $\xi_1, \dots, \xi_N \sim \mathbb{P}$ (unknown)
  e.g. in supervised learning: $\xi_i = (a_i, y_i)$ feature, label
- Train model: x the parameter $f(x, \cdot)$ the objective function
  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 ? (generalisation, data shifts, adversarial examples,...)
  Take possible variations into account during training (= when optimizing ☺)

Set-up: data-driven optimization under uncertainty

- Training data: $\xi_1, \ldots, \xi_N \sim \mathbb{P}$ (unknown)
  e.g. in supervised learning: $\xi_i = (a_i, y_i)$ feature, label
- Train model: x the parameter $f(x, \cdot)$ the objective function
  e.g. least-square regression: $f\big(x, (a, y)\big) = (x^\top a - y)^2$
- 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 ? (generalisation, data shifts, adversarial examples,...)
  Take possible variations into account during training (= when optimizing ☺)
- (Distributionally) robust optimization
  (optimize expected loss for the worst probability in a set of perturbations)

$$\min_x \max_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(x, \xi)]$$

Modeling issues

E.g. ambiguity/incertainty set $\mathcal{U}$: $\min_{x} \max_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(x,\xi)]$

- $\mathcal{U} = \left\{\widehat{\mathbb{P}}_N\right\}$: $\min_{x} \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$ standard ERM
- $\mathcal{U} = \{\mathbb{Q} : \mathrm{supp}(\mathbb{Q}) \subset U\}$: $\min_{x} \max_{\xi\in U} f(x,\xi)$ standard robust optimization
- $\mathcal{U}$ defined by moments e.g. [Delage, Ye, '10]
- $\mathcal{U} = \left\{\mathbb{Q} : \ d(\mathbb{Q},\widehat{\mathbb{P}}_N) \leqslant \rho\right\}$ for various distances or divergences
  E.g. KL-div., χ_2-div., max-mean-discrepancy... e.g. [Namkoong, Duchi '17]
- $\mathcal{U} = \left\{\mathbb{Q} : \ W(\mathbb{Q},\widehat{\mathbb{P}}_N) \leqslant \rho\right\}$ Wasserstein distance from optimal transport (OT) (in this talk)
  Good statistical/practical properties... e.g. [Kuhn *et al.* '18]
  Interprets up to first-order as a penalization by $\|\nabla_\xi f(x,\xi)\|$ e.g. [Gao *et al.* '18]

DRO in action #1 : toy example

Least-square linear regression

Data : $\xi_1, \xi_2, \ldots, \xi_N$ with $\xi_i = (a_i, y_i)$ in two groups (majority vs. minority)

$y_i = \bar{x}^\top a_i + \varepsilon_i$ with $\varepsilon_i \sim \beta\mathcal{N}^{\text{major}} + (1-\beta)\mathcal{N}^{\text{minor}}$

Compute from data:

standard regression x^{ERM} vs. DRO regression x^{DRO} (KL-regularized)

DRO in action #1 : toy example

Least-square linear regression

Data : $\xi_1, \xi_2, \ldots, \xi_N$ with $\xi_i = (a_i, y_i)$ in two groups (majority vs. minority)

$y_i = \bar{x}^\top a_i + \varepsilon_i$ with $\varepsilon_i \sim \beta\mathcal{N}^{\text{major}} + (1-\beta)\mathcal{N}^{\text{minor}}$

Compute from data:

standard regression x^{ERM} vs. DRO regression x^{DRO} (KL-regularized)

Generate new data $\xi'_1, \ldots, \xi'_M$

Test the regression errors given by x^{ERM} vs x^{DRO}

DRO in action #1 : toy example

Least-square linear regression

Data : $\xi_1, \xi_2, \ldots, \xi_N$ with $\xi_i = (a_i, y_i)$ in two groups (majority vs. minority)

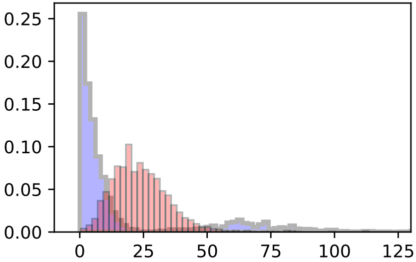
$y_i = \bar{x}^\top a_i + \varepsilon_i$ with $\varepsilon_i \sim \beta \mathcal{N}^{\text{major}} + (1-\beta)\mathcal{N}^{\text{minor}}$

Compute from data:

standard regression x^{ERM} vs. DRO regression x^{DRO} (KL-regularized)

Generate new data $\xi'_1, \ldots, \xi'_M$

Test the regression errors given by x^{ERM} vs x^{DRO}



Histogram of the test regression errors $(r_i = |x^\top a_i - y_i|)$

DRO in action #1 : toy example

Least-square linear regression

Data : $\xi_1, \xi_2, \dots, \xi_N$ with $\xi_i = (a_i, y_i)$ in two groups (majority vs. minority)

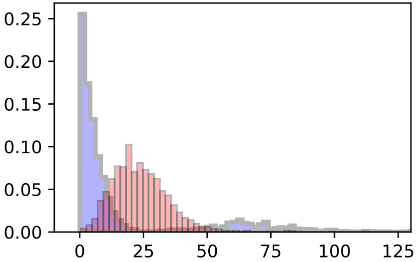
$y_i = \bar{x}^\top a_i + \varepsilon_i$ with $\varepsilon_i \sim \beta \mathcal{N}^{\text{major}} + (1-\beta)\mathcal{N}^{\text{minor}}$

Compute from data:

standard regression x^{ERM} vs. DRO regression x^{DRO} (KL-regularized)

Generate new data $\xi'_1, \dots, \xi'_M$

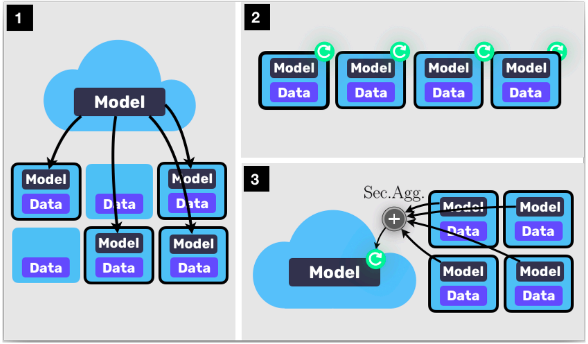
Test the regression errors given by x^{ERM} vs x^{DRO}



Histogram of the test regression errors $(r_i = |x^\top a_i - y_i|)$

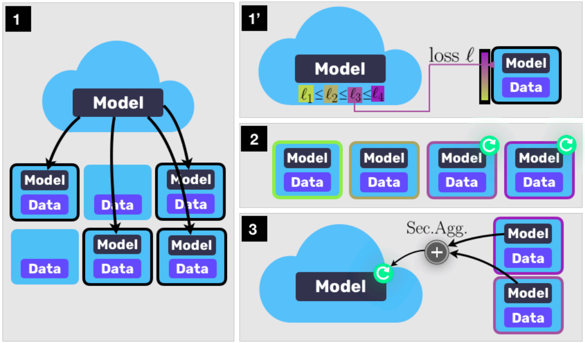
DRO re-shapes histograms towards more fairness ☺

DRO in action #2 : federated learning with heterogeneous users



Federated Learning by Google = FedAvg

DRO in action #2 : federated learning with heterogeneous users



Federated Learning by Google = FedAvg vs. DRO FedAvg [Laguel, Pillutla, M., Harchaoui '21]

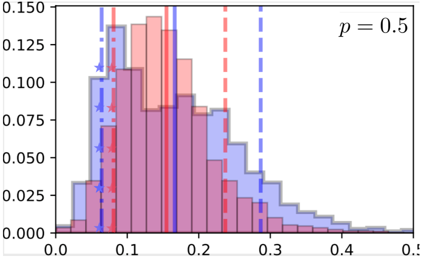
Illustration:

Classification task by ConvNet

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

Histogram over users of test misclassification error

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



Current research topic: extend the (W)DRO toolkit

- DRO works well ☺
- Trade-off in practice : modeling vs. computational tractability
- Wasserstein-DRO is popular...
  Good statistical/practical properties, e.g. [Kuhn *et al.* '18]
- ...but has some limitations ! news results
- We propose: Regularized WDRO [Azizian, Iutzeler, M. '22]
- Why regularizing ? it helps computationnally !
  One of the main reasons of the popularity of OT in ML [Cuturi '13]
- On-going research... (try to import and adapt the techniques of OT for WDRO)

DRO with Wasserstein balls as ambiguity sets

Def: Wasserstein distance (given a cost function c)

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

DRO with Wasserstein balls as ambiguity sets

Def: Wasserstein distance (given a cost function c)

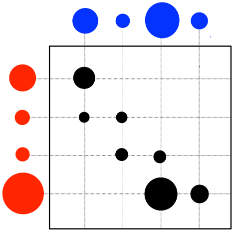
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}} \{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \}$$

Demystification: in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^N c_{i,j}\,\boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\dots,N \\ \quad \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\dots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\dots,N \end{cases}$$

linear assignment !



DRO with Wasserstein balls as ambiguity sets

Def: Wasserstein distance (given a cost function c)

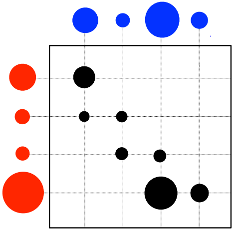
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}}\{\,\mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q}\}$$

Demystification: in the discrete case

e.g. $\mathbb{P} = (p_1,\dots,p_N)$ and $\mathbb{Q} = (q_1,\dots,q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} & \sum_{i,j=1}^{N} c_{i,j}\,\boldsymbol{\pi}_{i,j} \\ \sum_{j=1}^{N}\boldsymbol{\pi}_{i,j} = p_i & i=1,\dots,N \\ \sum_{i=1}^{N}\boldsymbol{\pi}_{i,j} = q_j & j=1,\dots,N \\ \boldsymbol{\pi}_{i,j} \geqslant 0 & i,j=1,\dots,N \end{cases}$$

linear assignment !



Wasserstein-DRO (WDRO) objective for given $\mathbb{P}$ and ρ

$$\begin{cases} \max_{\mathbb{Q}} & \mathbb{E}_{\mathbb{Q}}[f(\xi)] \\ & W(\mathbb{P},\mathbb{Q}) \leqslant \rho \end{cases}$$

DRO with Wasserstein balls as ambiguity sets

Def: Wasserstein distance (given a cost function c)

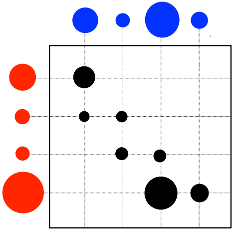
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}} \{ \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q} \}$$

Demystification: in the discrete case

e.g. $\mathbb{P} = (p_1, \dots, p_N)$ and $\mathbb{Q} = (q_1, \dots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^N c_{i,j}\, \boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\dots,N \\ \quad \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\dots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\dots,N \end{cases}$$

linear assignment !



Wasserstein-DRO (WDRO) objective for given $\mathbb{P}$ and ρ

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

DRO with Wasserstein balls as ambiguity sets

Def: Wasserstein distance (given a cost function c)

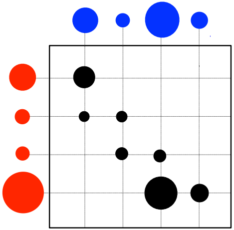
$$W(\mathbb{P},\mathbb{Q}) = \min_{\boldsymbol{\pi}}\{\, \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] : \boldsymbol{\pi} \text{ with marginals } [\boldsymbol{\pi}]_1 = \mathbb{P} \text{ and } [\boldsymbol{\pi}]_2 = \mathbb{Q}\}$$

Demystification: in the discrete case

e.g. $\mathbb{P} = (p_1, \ldots, p_N)$ and $\mathbb{Q} = (q_1, \ldots, q_N)$ in the simplex

$$\begin{cases} \min_{\boldsymbol{\pi}} \quad \sum_{i,j=1}^N c_{i,j}\, \boldsymbol{\pi}_{i,j} \\ \quad \sum_{j=1}^N \boldsymbol{\pi}_{i,j} = p_i \quad i = 1,\ldots,N \\ \quad \sum_{i=1}^N \boldsymbol{\pi}_{i,j} = q_j \quad j = 1,\ldots,N \\ \quad \boldsymbol{\pi}_{i,j} \geqslant 0 \quad i,j = 1,\ldots,N \end{cases}$$

linear assignment !



Wasserstein-DRO (WDRO) objective for given $\mathbb{P}$ and ρ

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

WDRO: better duals by regularization

Let's write its dual

Primal WDRO

$$\begin{cases} \max_{\boldsymbol{\pi}} \quad \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(\xi)] \\ \quad [\boldsymbol{\pi}]_1 = \mathbb{P} \\ \quad \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] \leqslant \rho \qquad \leftarrow \lambda \geqslant 0 \end{cases}$$

Dual WDRO

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

WDRO: better duals by regularization

Let's write its dual

Primal WDRO regularized (with two convex functions R, S)

$$\begin{cases} \max_{\boldsymbol{\pi}} \quad \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(\xi)] - R(\boldsymbol{\pi}) \\ \quad [\boldsymbol{\pi}]_1 = \mathbb{P} \\ \quad \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] + S(\boldsymbol{\pi}) \leqslant \rho \qquad \leftarrow \lambda \geqslant 0 \end{cases}$$

Dual WDRO when regularized

$$\min_{\lambda \geqslant 0} \min_{\varphi} \lambda\rho + \mathbb{E}_{\mathbb{P}}[\, \max_{\xi'} \{f(\xi') - \lambda c(\xi,\xi') - \varphi(\xi,\xi')\}\,] + (R + \lambda S)_*(\varphi)$$

WDRO: better duals by regularization

Let's write its dual

Primal WDRO regularized (with two convex functions R, S)

$$\left\{\begin{array}{l} \max_{\boldsymbol{\pi}} \quad \mathbb{E}_{[\boldsymbol{\pi}]_2}[f(\xi)] - R(\boldsymbol{\pi}) \\ \quad [\boldsymbol{\pi}]_1 = \mathbb{P} \\ \quad \mathbb{E}_{\boldsymbol{\pi}}[c(\xi,\xi')] + S(\boldsymbol{\pi}) \leqslant \rho \qquad \leftarrow \lambda \geqslant 0 \end{array}\right.$$

Dual WDRO when regularized

$$\min_{\lambda \geqslant 0} \min_{\varphi} \lambda\rho + \mathbb{E}_{\mathbb{P}}[\max_{\xi'}\{f(\xi') - \lambda c(\xi,\xi') - \varphi(\xi,\xi')\}] + (R + \lambda S)_*(\varphi)$$

Quite abstract... but more concrete expressions when specialized

e.g. with $R(\pi) = \varepsilon\,\mathrm{KL}(\pi|\pi_0)$ and $S(\pi) = \delta\,\mathrm{KL}(\pi|\pi_0)$ for a given π_0

$$\min_{\lambda \geqslant 0} \quad \lambda\rho + (\varepsilon + \lambda\delta)\mathbb{E}_{\mathbb{P}} \log\left(\mathbb{E}_{\xi' \sim \pi_0(\cdot|\xi)} e^{\frac{f(\xi') - \lambda c(\xi,\xi')}{\varepsilon + \lambda\delta}}\right)$$

WDRO: approximation result

Dual WDRO: $(P) \quad \min_{\lambda \geqslant 0} \lambda\rho + \mathbb{E}_{\mathbb{P}}[\max_{\xi'} f(\xi') - \lambda c(\xi, \xi')]$

Dual WDRO regularized by $R(\pi) = \varepsilon \, \mathrm{KL}(\pi|\pi_0)$ and $S(\pi) = \delta \, \mathrm{KL}(\pi|\pi_0)$

$$(P_{\varepsilon,\delta}) \quad \min_{\lambda \geqslant 0} \lambda\rho + (\varepsilon + \lambda\delta)\mathbb{E}_{\mathbb{P}} \log \left(\mathbb{E}_{\xi' \sim \pi_0(\cdot|\xi)} e^{\frac{f(\xi') - \lambda c(\xi, \xi')}{\varepsilon + \lambda\delta}} \right)$$

Theorem ([Azizian, Iutzeler, M. '22])

Under mild assumptions (non-degeneracy, lipschitz, $c = \|\cdot\|^p$, special form of π_0), if the support of $\mathbb{P}$ is contained in a compact convex set $\Xi \subset \mathbb{R}^d$, then

$$0 \leqslant \mathrm{val}(P) - \mathrm{val}(P_{\varepsilon,\delta}) \leqslant C \, d \, (\varepsilon + \overline{\lambda}\delta) \log \frac{1}{\varepsilon + \overline{\lambda}\delta}$$

where $\overline{\lambda} = \frac{2 \sup_{\Xi} |f|}{\rho - \mathbb{E}_{\pi_0} c}$ an explicit dual bound.

WDRO: approximation result

Dual WDRO: $$(P) \quad \min_{\lambda \geqslant 0} \lambda\rho + \mathbb{E}_{\mathbb{P}}[\max_{\xi'} f(\xi') - \lambda c(\xi,\xi')]$$

Dual WDRO regularized by $R(\pi) = \varepsilon\, \mathrm{KL}(\pi|\pi_0)$ and $S(\pi) = \delta\, \mathrm{KL}(\pi|\pi_0)$

$$(P_{\varepsilon,\delta}) \quad \min_{\lambda \geqslant 0} \; \lambda\rho + (\varepsilon + \lambda\delta)\mathbb{E}_{\mathbb{P}} \log\left(\mathbb{E}_{\xi' \sim \pi_0(\cdot|\xi)} e^{\frac{f(\xi') - \lambda c(\xi,\xi')}{\varepsilon + \lambda\delta}}\right)$$

Theorem ([Azizian, Iutzeler, M. '22])

Under mild assumptions (non-degeneracy, lipschitz, $c = \|\cdot\|^p$, special form of π_0), if the support of $\mathbb{P}$ is contained in a compact convex set $\Xi \subset \mathbb{R}^d$, then

$$0 \;\leqslant\; \mathrm{val}(P) - \mathrm{val}(P_{\varepsilon,\delta}) \;\leqslant\; C\, d\, (\varepsilon + \overline{\lambda}\delta) \log \frac{1}{\varepsilon + \overline{\lambda}\delta}$$

where $\overline{\lambda} = \frac{2 \sup_{\Xi} |f|}{\rho - \mathbb{E}_{\pi_0} c}$ an explicit dual bound.

We control the error... Next steps:

- solve $(P_{\varepsilon,\delta})$ efficiently
- establish generalization bounds

another story...

Conclusion

Main take-aways

- min-max optimization is a rich/subtle field with many applications in ML
- In general: more work is needed on robustness (shifts, nonconvexity, stability, extreme cases...)
- Our current work: extend the toolkit of DRO by regularization (towards scalable algorithms...)
  general duality, approximation results, worst-case distribution... statistical guarantees ?

Work advertized today

1. Last-iterate convergence of stochastic min/max algorithms
   [Hsieh, Iutzeler, M., Mertikopoulos '20]
2. Improvements for non-conforming users in federated learning
   [Laguel, Pillutla, M., Harchaoui '21]
3. Regularization of distributionally robust optimization
   [Azizian, Iutzeler, M. '22]

Conclusion

Main take-aways

- min-max optimization is a rich/subtle field with many applications in ML
- In general: more work is needed on robustness (shifts, nonconvexity, stability, extreme cases...)
- Our current work: extend the toolkit of DRO by regularization (towards scalable algorithms...)
  general duality, approximation results, worst-case distribution... statistical guarantees ?

Work advertized today

1. Last-iterate convergence of stochastic min/max algorithms
   [Hsieh, Iutzeler, M., Mertikopoulos '20]
2. Improvements for non-conforming users in federated learning
   [Laguel, Pillutla, M., Harchaoui '21]
3. Regularization of distributionally robust optimization
   [Azizian, Iutzeler, M. '22]

thank you all !

Existing results extragradient in the stochastic setting

V is L-Lipschitz continuous

Stochastic	Hypothesis	Convergence type	rate
[JNT '11]	Monotone	Ergodic	$O(1/\sqrt{t})$
[KS '19]	Strongly monotone	Last iterate	$O(1/t)$
[MLZF+ '19]	Strictly coherent	Last iterate	-

Last-iterate convergence for stochastic monotone operators?

- Regularization with vanishing weight
- Variance reduction with increasing batch size
- Finite sum: SVRG-like variance reduction
- Second-order: stochastic Hamiltonian descent
- Different stepsizes for the two steps of EG!

Beyond monotonicity: Local convergence

Theorem

Assumptions:

(i) Locally variational stable and locally Lipschitz around a soultion $x^\star$.
(ii) V is differentiable at $x^\star$ and $\mathrm{Jac}\, V(sol)$ is invertible.

Beyond monotonicity: Local convergence

Theorem

Assumptions:

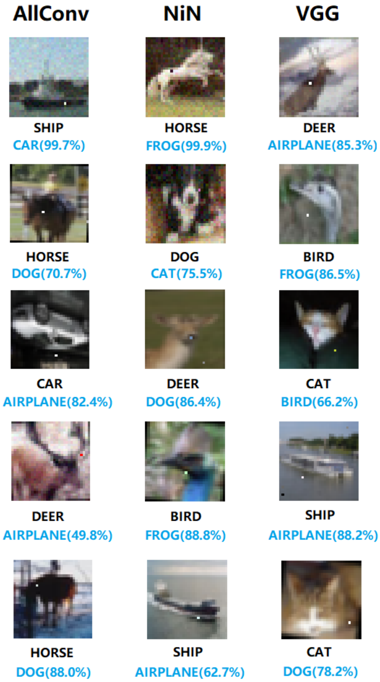
(i) Locally variational stable and locally Lipschitz around a soultion $x^\star$.
(ii) V is differentiable at $x^\star$ and $\mathrm{Jac}\, V(sol)$ is invertible.

Guarantee:

For any tolerance level $\delta > 0$, there exists a stepsize policy for double stepsize extra-gradient such that if the algorithm is initialized close enough to $x^\star$, there exists an event with probability at least $1 - \delta$ and, conditioned on this event:

- Under (i), the iterates converge to $x^\star$.
- Under (i) and (ii), X_t converges to $x^\star$ at a rate $\mathrm{O}\left(1/\sqrt[3]{t}\right)$ in mean square error.

One-pixel attack



From [Su, Vargas, Sakurai '18]

SFL comparison w. state-of-the-art

From [Laguel, Pillutla, M., Harchaoui '21]

		90^{th} Percentile		Average	
		Linear	ConvNet	Linear	ConvNet
	Δ-FL $p = 0.5$	$\mathbf{46.48} \pm 0.38$	$\mathbf{23.69} \pm 0.94$	35.02 ± 0.20	$\mathbf{15.49} \pm 0.30$
$\mathbb{E}$	FedAvg	49.66 ± 0.67	28.46 ± 1.07	34.38 ± 0.38	16.64 ± 0.50
prox	FedProx	49.15 ± 0.74	27.01 ± 1.86	$\mathbf{33.82} \pm 0.30$	16.02 ± 0.54
$\Vert \cdot \Vert_q^q$ $(q > 1)$	q-FFL	49.90 ± 0.58	28.02 ± 0.80	34.34 ± 0.33	16.59 ± 0.30
max	AFL	51.62 ± 0.28	45.08 ± 1.00	39.33 ± 0.27	33.01 ± 0.37

Regularized WDRO

From [Azizian, Iutzeler, M. '22]

- Recall : KL (Kullback-Lieber divergence)

$$\mathrm{KL}(\mu|\nu) = \begin{cases} \int \log \frac{\mathrm{d}\mu}{\mathrm{d}\nu} \, \mathrm{d}\mu & \text{if } \mu,\nu \geqslant 0 \text{ and } \mu \ll \nu \\ +\infty & \text{otherwise} \end{cases}$$

  In the discrete case: $\mathbb{P} = (p_1, \ldots, p_N)$ and $\mathbb{Q} = (q_1, \ldots, q_N)$

$$\mathrm{KL}(\mathbb{P}|\mathbb{Q}) = \sum_{i=1}^{N} p_i \log \frac{p_i}{q_i}$$

- Explicit reference measure

$$\pi_0(\mathrm{d}\xi, \mathrm{d}\xi') \;\propto\; \mathbb{P}(\mathrm{d}\xi)\, \mathbb{I}_{\xi' \in \Xi}\, e^{-\frac{\|\xi - \xi'\|^p}{2^{p-1}\sigma}}\, \mathrm{d}\xi'$$

- Worst-case distribution

$$\mathbb{P}^* = (\ldots) \text{ supported on the whole space}$$

  vs. WDRO where the worst-case is finitely supported...

  (WDRO hedges against wrong set of distributions ?)